university of groningen

faculty of science and engineering

Multi-chamber Suction Anchor Design Optimization

Bachelor Integration Project

OCEAN GRAZER

Supervisors

prof. dr. A. Vakis

M. Mohebbi

Author

Leonard Vos | S3209016

Rijksuniversiteit Groningen

Ocean Grazer

Abstract

Ocean Grazer B.V. is developing an energy storage system for offshore purposes. The so-called Ocean Battery system must be installed on the seafloor to function properly since it uses the hydrostatic pressure from the ocean to store energy in potential form in flexible bladders. The system must be able to withstand all forces acting upon it. This report focuses on the foundation of the Ocean Battery that makes use of suction anchor technique. Suction anchors are widely used as offshore foundations for deep water purposes. The current design of the foundation consists of nine chambers and suction can be applied to all chambers. Last semester analytical research was performed on the current design and the results were promising. However, the design is not optimal, full installation cannot be guaranteed for all types of soils. Therefore, the analytical research was continued and design changes were investigated. Finally, a proposal is given to increase the skirt height of one of the chambers and increase the diameter of the cylindrical caissons to further guarantee installation and increase the final holding capacity.

Table of Contents

1. Introduction

Countering global warming by reducing CO_2 emissions is one of the main challenges for humanity for the near future. Sustainable energy production is one of the solutions to reduce CO_2 emissions. Offshore renewable energy is on the rise due to its great potential (Bahaj, 2011) and includes both wave energy and offshore wind. However, one of the main issues of renewable energy production in general is that some sort of storage is necessary due to intermittency (Alami, 2020). Power generation from fossil fuels can be turned on and off to supply the exact amount of energy needed throughout the day. However, renewable energy sources can be available in abundance or not at all, depending on the weather conditions.

The Ocean Grazer B.V. focuses on offshore sustainable energy harvesting and storage solutions. One of which is called the Ocean Battery. The Ocean Battery is a system that stores energy in potential form and stores the surplus of energy by the principles of pumped hydroelectric energy storage (PHES). However, instead of storage in gravitational potential form (Nogueira, et al., 2020), the energy is stored in the form of hydrostatic pressure potential form (Ocean Grazer, 2020).

For the battery system to function properly, it must be safely anchored to the seabed. Suction anchor technology is the most common method used in deep sea applications (Haldar, et al., 2020). Suction caissons are the most eco-friendly foundation for offshore structures. Heavy equipment is not necessary for installation, which means less vibration, noise, and suspended sediment. Furthermore, this type of foundation is economic due to relative simple and fast installation (Oh, et al., 2018).

The suction foundation of the Ocean Battery is the subject of this research. By performing literature and analytical research, the current design and installation process is investigated. The current design is not optimal. Therefore, design changes are investigated and the deliverable of this report is a proposal of an improved design.

2. Problem Analysis

Firstly, the context in which this research is performed will be discussed, as well as a system description and stakeholder analysis. Then, the research problem and goal are defined.

2.1 Problem Context

Ocean Grazer started as a research group at the Rijksuniversiteit Groningen in 2014. The research group started with an innovative idea to produce sustainable energy with the use of a wave energy converter. The following years, Ocean Grazer research group supported many students in their final research projects not just on wave energy conversion but also on other sustainable offshore energy harvesting and storage projects (Ocean Grazer, 2020). The Ocean Grazer B.V. was founded in 2019 and focuses on commercialising the ideas from the research group.

Currently, the focus of the Ocean Grazer B.V. is their storage system called the Ocean Battery. The Ocean Battery stores energy in hydrostatic pressure potential form on the seafloor by inflating flexible bladders by powering pumps with the excess energy. The bladders can then be deflated through a turbine, and energy will be converted back into electrical form. Two versions are currently being developed by the company. In this research project, the deep water version of the Ocean Battery is considered. The foundation is an important aspect of the system since the Ocean Battery must be safely anchored to the seafloor to operate properly. Currently, prototype B of the Ocean Battery is being developed and made ready for testing purposes. The focus of this research project is on the design of the foundation of prototype B, see Figure 2.1.

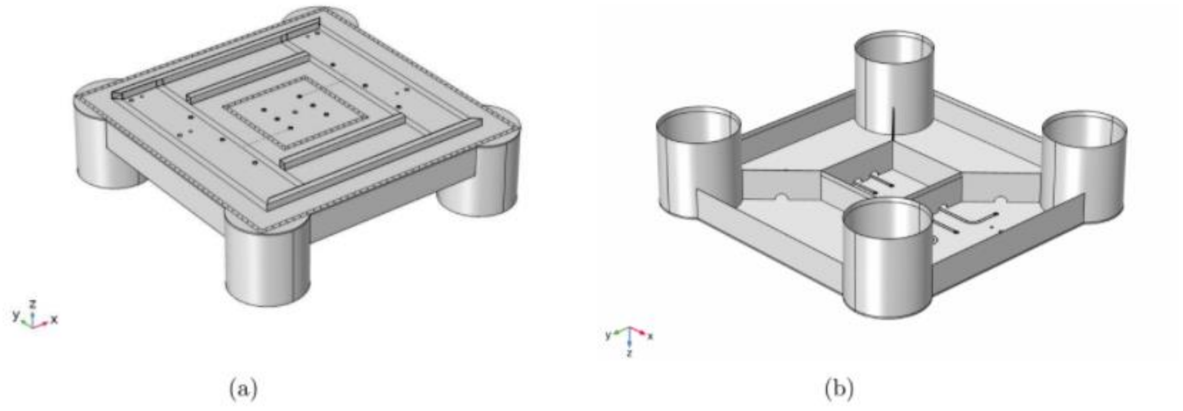


Figure 2.1. The foundation of Prototype B, (a) top view and (b) bottom view (Ocean Grazer, 2020).

Analytical research performed by (Hut, 2020) and (Van Tongeren, 2020) on Prototype B showed promising results. Both theses from Hut and Van Tongeren present models of the installation behaviour of the foundation of the Ocean Battery. The models are created in MATLAB and try to simulate the full installation from self-weight penetration phase to suction assisted penetration phase. Afterwards, the holding capacity is determined and will give an idea of the operating performance of the Ocean Battery.

The system was able to install under certain conditions and under certain assumptions. These assumptions are discussed in section 5.4. However, installation of the current design of the system cannot be guaranteed. Required suction for full installation approaches the estimated critical suction and when the required suction exceeds the critical suction, installation will fail. Further explanation on critical suction and failure phenomena will be discussed in section 4.2.6 and 4.3.6.

The model from Van Tongeren also includes three design changes to the anchor system that were investigated separately. These design changes will be discussed in section 2.2.3. These design adaptations effectively decrease the resistance acting on the system or decrease the calculated required suction. Results showed that with all the proposed changes, the installation performance improved (Van Tongeren, 2020). However, no final proposal was included to apply these changes to the design.

2.2 System Description and Scope

The company has worked on a design of single and multi-chambered suction anchors which resulted in prototype designs. Currently, prototype B of the system is being developed. The foundation of prototype B is a scale version of 2x2 [m] and the foundation consists of nine chambers, see Figure 2.1 (b). The battery system of prototype 2 has been fully developed and is being built and made ready for testing. The goal is to deploy this prototype in the Eemshaven early 2021. The suction anchor design is not yet physically developed. Due to the urgency of deploying the battery for testing purposes, the decision was made to use some other type of temporary foundation. Therefore, design changes can still be proposed for the foundation of the Ocean Battery.

The full system is the suction anchor with the Ocean Battery installed on top, however only the foundation lies within the scope of this research. Whether the system remains stable and fully installed on the seafloor can be determined from the calculated holding capacity which is an output of the model.

In the following sections, the Ocean Battery system and foundation are explained and the design adaptations that were considered in previously performed research will be briefly discussed. Furthermore, an additional design change will be presented.

2.2.1 Ocean Battery

The Ocean Battery stores energy in potential form that is generated by offshore wind turbines and wave harvesters. The battery is installed at the bottom of the sea. The surplus of energy being generated is used to power a pump within the battery. This pump will fill flexible bladders on the outside of the system. Once an energy deficit occurs, the bladders can be deflated by the hydrostatic pressure of the ocean water and the fluid flows through a turbine to regenerate energy (Ocean Grazer, 2020).

2.2.2 Ocean Foundation

The current design consists of nine chambers. Four cylindrical chambers in the corners, one square chamber in the middle and four trapezoidal chambers connecting the square and cylindrical chambers. All nine of these chambers can be used as suction caissons. The installation process proceeds as follows for the nine suction chambers, see also Figure 2.2. After the initial penetration depth is reached, the chambers will be activated in a sequence. First suction will be applied to the cylindrical chambers, then the to square chamber and finally to the trapezoidal chambers. Further explanation on the installation process will be discussed in section 4.1 for suction caissons in general and chapter 5 for the Ocean Battery foundation.

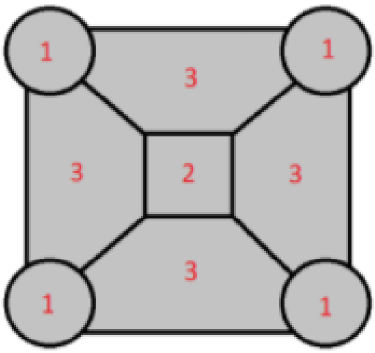


Figure 2.2. Installation sequence of prototype B of the Ocean Battery foundation (Van Tongeren, 2020).

2.2.3 Design changes

Three design changes for prototype B were previously investigated analytically (Van Tongeren, 2020). These changes can be seen in Figure 2.3 (a), (b) and (c). These three changes all showed benefits in the installation phase. Making the tip of the chambers slanted increased the initial penetration significantly and slightly decreased the required suction in the suction assisted phase. It should be noted that simulating the behaviour of slanted tips is difficult and depends on many parameters, see also section 4.4. Presumably, the effect of adding slanted tips is beneficial to the installation process. However, the calculation method in the model has not been validated. Decreasing the plate thickness and adding additional weight on top of the system also showed similar improvements. However, both design changes have drawbacks. Decreasing the plate thickness also has an effect on the overall strength of the system. Strength analysis was not part of the investigation and should be considered before implementation. Then, adding additional weight to the system is not desirable. Preferably, the system itself is adjusted to a design for which installation can be guaranteed without external attachments to the system. Therefore, adding weight to the system will only be considered when the other changes cannot improve the foundation sufficiently.

The fourth design change, Figure 2.3 (d) adjusting the skirt height of the square chamber, has not been investigated but has been proposed by the stakeholders. The hypothesis is that suction can be applied to the area of the square chamber for a longer period and installation performance will improve.

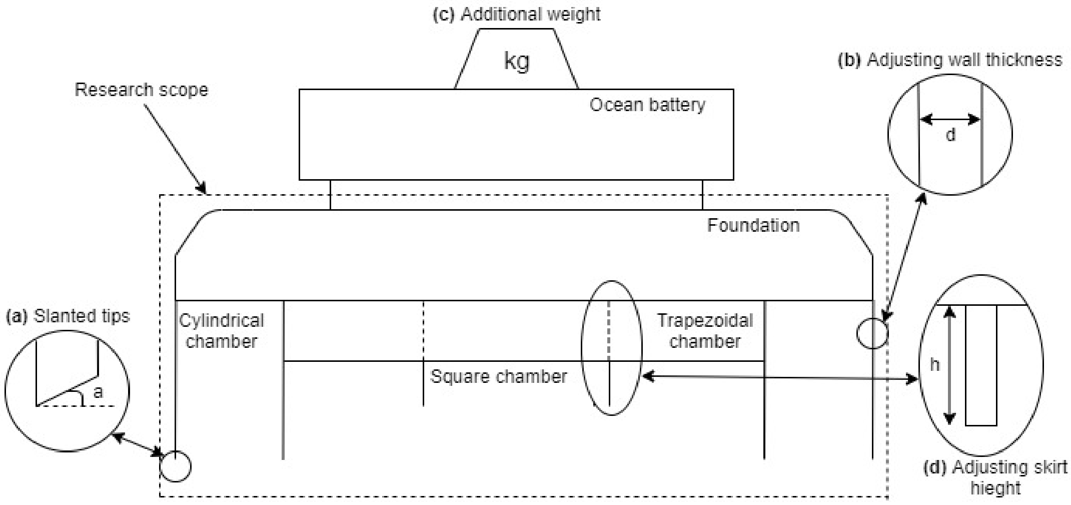


Figure 2.3. Schematic side view of the Ocean Battery with the considered design changes, (a) slanted tips, (b) adjusted wall thickness, (c) additional weight and (d) adjusting the skirt height of the square chamber.

2.3 Stakeholder Analysis

In this section the primary stakeholders and problem owner are identified. Their power or influence and interest is described, and their requirements are stated.

The problem owner is the CTO of the Ocean Grazer, Marijn van Rooij. Being the CTO, he is involved in all the technical aspects of the Ocean Grazer projects. Deployment of (a prototype of) the ocean battery system currently has the highest priority for the company. The company requires a working design that can fully install, and current design does not fulfil this requirement.

Other stakeholders that can be identified are Drs. Wout Prins and Prof. dr. Antonis Vakis. Wout Prins is a co-owner of the Ocean Grazer B.V. and co-inventor of the Ocean Battery system. Wout Prins is heavily engaged in the design of the system and therefore has high interest and high influence on the outcome of this research. Antonis Vakis is a co-founder and Scientific Advisor of the Ocean Grazer B.V. Antonis Vakis is also engaged in the technical aspects of the system and is the first supervisor of this thesis. Therefore, he has high interest in the outcome and great influence on the process of this research project.

2.4 Problem and Goal Statement

Sections 2.1 and 2.2 have stated the current state of the Ocean Battery foundation. The problem statement of this research project can be defined as follows:

Installation and operation of the Ocean Battery cannot be guaranteed with the current multi-chambered design of the suction anchor and the current state of the installation models.

Successful installation is defined as the system being levelled and fully penetrated in the sea floor. Operation of the system is successful if the system is fully able to cope with the external forces acting upon it and functions as a safe anchoring point for the Ocean Battery, equivalently: the holding capacity must be sufficient.

From the problem statement it can be concluded that the design must be improved. To achieve this, design changes will be proposed and the installation process will be investigated. The main idea of improving the installation process, is by decreasing the penetration resistance or increasing the area and timespan over which suction can be applied, but not at the cost of a lower holding capacity. Furthermore, the installation rules and conditions can be adapted to be more robust. These design changes will be investigated analytically. The goal statement is as follows:

To continue from previously performed analytical research, improve the current models of the installation procedure, and simultaneously propose adaptations for the current design that improve overall performance of the suction anchor.

The overall performance means performance of installation (guarantee of installation) and performance of operation, sufficient holding capacity.

3. Research Questions

In this section, the main research question is defined. To answer this question, sub questions are defined and the combination of answers to these questions will give an answer to the main research question. The main research question is:

*[**RQ**]: To what extent can the current multi-chambered suction design anchor and installation process be improved to guarantee installation and operation?*

Sub questions are then defined per design adaptation and are formulated below. For efficiency purposes, the design changes should be investigated in order of ease of implementation.

> [**SQ1**]: To what extent can slanted cylindrical chamber tips improve installation of the system?
>
> [**SQ2**]: To what extent can adaptations to skirt height of the square chamber improve the installation of the system?
>
> [**SQ3**]: What is the influence of the wall thickness of the plates on the installation phase of the system?
>
> [**SQ4**]: To what extent can the dimensions of the chambers of the foundation be changed to improve the installation of the system?

Sub questions SQ1 and SQ3 investigate whether the penetration resistance on the foundation can be decreased and consequently improve the installation process. Sub questions SQ2 will investigate whether the benefits of increasing the time over which suction can be applied are larger than the presumably increased penetration resistance. Sub question SQ4 focuses on whether changes to the areas of different chambers benefit the installation process. The overall area to which suction will be applied remains equal as the total width and length does not change.

The design adaptations focus on improving the installation phase of the system. However, the new design must ultimately function as an anchor, being able to cope with the forces of the Ocean Battery and external forces.

> [**SQ5**]: What is the influence of the design changes on the operation phase of the system?

Finally, when all answers combined answer the main research question, a proposal will be presented whether the current design can be improved with implementing above-mentioned adaptations.

4. Literature Research

To understand the full concept of suction anchors, literature research is required. Firstly, the modus operandi of suction anchors is presented. Then, calculations used in the model are described. Finally, literature research is performed to answer research sub question 1, whether slanted tips of the caisson improve installation behaviour.

4.1 Suction Anchor principles

The general concept of suction anchor is already discussed in previous sections. In this section a more comprehensive description of suction anchors in general will be given.

A suction anchor is a reversed bucket like structure with a closed top. This so-called caisson can be lowered on the seabed and once settled and a sufficient seal is present, Figure 4.1 (a), a suction force can be applied in the caisson. An under pressure is created in the caisson by means of pumping water out of the chamber, Figure 4.1 (b), and the caisson will sink further into the seabed. Once installed, Figure 4.1 (c), the suction force and the friction that occurs between the caisson wall and the soil ensures that the caisson can function as a safe anchor to the seabed. The installation process can be decomposed in two phases, the self-weight installation phase, and the suction assisted penetration phase.

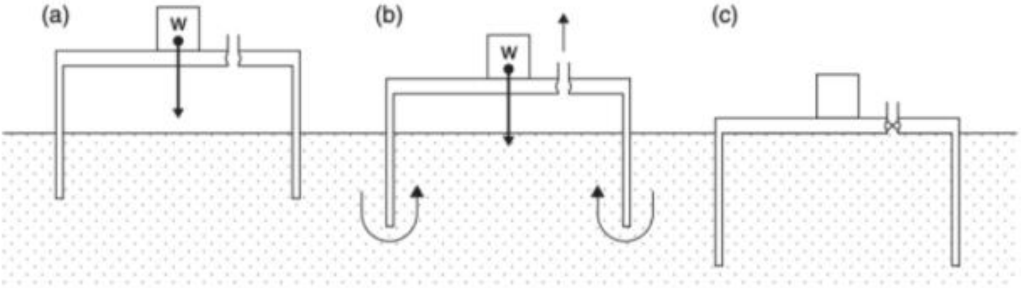


Figure 4.1. *Stages of installation of suction caisson (a) self-weight (b) suction assisted, and (c) installed* (Bhattacharya, 2019).

Successful installation must meet two requirements:

1. Initial penetration depth is sufficient to create a seal and suction forces can be applied.

2. The then required suction forces stay below the maximum or critical suction forces and the anchor can fully penetrate the seabed.

Once these two requirements are met, the system can install and will function as anchor point on the seafloor to which the Ocean Battery can be attached. Once installed, the system must be able to withstand all the external forces and the forces that the attached system applies on the anchor. The so-called holding capacity of the anchor must be sufficient.

Suction foundations have several benefits over other conventional offshore foundation methods. Heavy equipment is not required for installation and minimizes the disturbance of the ocean environment at the deployment location (Van Dijk, 2018).

Furthermore, compared to other foundation types, suction foundations are economic since construction can be done onshore and installation is relatively simple and fast (Oh, et al., 2018).

The calculation methods used in the model originate from the articles of Houlsby & Byrne (2005a and 2005b). The articles describe the necessary formulas and provide validation for these formulas for two types of soils. Generally, seafloor soils can be distinguished into two types: cohesionless and cohesive soils. The formulas required differ significantly and are therefore separately discussed in section 4.2 and 4.3. Besides the calculations required for the self-weight and suction assisted phase, the limitations of the installation process for both soils are given. Finally, an approximation for calculating the holding capacity of the system is presented.

4.2 Cohesionless soils

The following seven sections discuss the soil characteristics, formulas for the installation phase, limitations to the installation and calculations required for the operation phase for cohesionless soils.

4.2.1 Soil characteristics

Cohesionless soils are sand and gravel like soils that run freely. The strength of cohesionless soils depends on the internal friction angle [φ]. Previous research from Hut (2020) indicated four main types of cohesionless soils: Redhill 110 Sand, Silica Sand, Statoil Sand and Luce Bay Sand (Hut, 2020). Each type of soils has different soil parameters. In Table 4.1 below, the relevant parameters are given for each type of soil.

Parameter	**Symbol**	**Unit**	**Redhill 110 Sand**	**Silica Sand**	**Statoil Sand**	**Luce Bay Sand**
Effective soil weight	γ'	kN/m^3	7.82	5.99	8.50	10.30
Lateral earth pressure coefficient	K	$-$	0.41	0.80	0.8	0.29
Angle of friction	φ	°	36	36	45	45
Permeability ratio	k_f	$-$	5	1	3	5

Table 4.1. Soil parameters of Redhill 110 Sand, Silica Sand, Statoil Sand and Luce Bay Sand (Houlsby & Byrne, 2005a, Houlsby, et al., 2006, Kelly, et al., 2007).

4.2.2 Variable explanation

In Table 4.2, all the variables required for the calculations for cohesionless used in the model are given, together with the corresponding units and a brief definition.

Variable	Unit	Definition
a	$-$	Ratio of excess pore pressure at caisson tips
C, C_i, C_o	m	Average, inner, and outer circumference of chamber
D, D_i, D_o	m	Average, inner, and outer diameter of caisson
h	m	Penetration depth of the caisson
K	$-$	Lateral earth pressure coefficient
k_f	$-$	Permeability ratio inside to outside of caisson
m	$-$	Multiple of diameter for enhanced stress
N_q	$-$	Bearing capacity factor (overburden)
N_γ	$-$	Bear capacity factor (self-weight)
s	Pa	Suction within caisson with respect to ambient seawater pressure
s_{crit}	Pa	Critical suction with respect to ambient seawater pressure at which piping failure occurs
Q_b	N	Tensile bearing capacity of the soil
Q_h	N	Holding capacity of the caisson
Q_{so}	N	Side shear on the outside wall
t	m	Thickness of the caisson wall
V'	N	Effective vertical load
W_c	N	Submerged weight of the caisson
W_s	N	Submerged weight of the soil
Z_i	$-$	Formula simplification factor inside caisson
Z_o	$-$	Formula simplification factor outside caisson
γ'	kN/m^3	Effective unit weight of the soil
δ	°	Interface friction angle
σ'_{end}	MPa	End bearing term
σ'_{vi}	MPa	Effective vertical stress at the inside of the caisson
σ'_{vo}	MPa	Effective vertical stress at the outside of the caisson
φ	°	Internal friction angle of the soil

Table 4.2. Variable explanation of the presented formulas 1-16 (Houlsby & Byrne, 2005a, Iskander, 2002, Hung, et al., 2017).

4.2.3 Self-weight installation

Self-weight penetration is relatively straightforward but essential for installation. It provides an initial seal necessary for the suction to create a downward force (Alluqmani, et al., 2019). The self-weight penetration depth can be estimated by first calculating the friction between the inside and outside of the caisson with the soil and the end bearing on the tip of the caisson (Oh, et al., 2018). The total penetration resistance can be compared to the total downward force or the weight of the system. Once the total resistance exceeds the total downward force, the anchor will not penetrate further.

Houlsby & Byrne (2005a) propose the following formula for calculating the total penetration resistance:

$$V' = \gamma' Z_o^2 \left(\exp\left(\frac{h}{Z_o}\right) - 1 - \frac{h}{Z_o}\right)(K\tan\delta)_o(\pi D_o) + \gamma' Z_i^2 \left(\exp\left(\frac{h}{Z_i}\right) - 1 - \frac{h}{Z_i}\right)(K\tan\delta)_i(\pi D_i) + \sigma'_{end}(\pi D t) \quad (1)$$

Formula 1 consists of three parts, friction along the inside wall of the caisson, friction along the outside wall of the caisson and the tip resistance of the caisson. In this formula, also the enhancement of vertical stress to the pile is included. Z_i Z_o and σ'_{end} are functions themselves and defined as follows:

$$Z_i = \frac{D_i}{(4(K\tan\delta)_i)} \quad (2)$$

$$Z_o = \frac{D_o(m^2-1)}{(4(K\tan\delta)_o)} \quad (3)$$

Then, an inequality must be solved to calculate σ'_{end}:

$$\text{If } \sigma'_{vi} - \sigma'_{vo} < \frac{2tN_\gamma}{N_q}, \text{ then } \sigma'_{end} = \sigma'_{vo}N_q + \gamma'^{\left(t-\frac{2x^2}{t}\right)}N_\gamma \text{ where } x = \frac{t}{2} + \frac{(\sigma'_{vo}-\sigma'_{vi})N_q}{4\gamma' N_\gamma} \quad (4)$$

$$\text{If } \sigma'_{vi} - \sigma'_{vo} \geq \frac{2tN_\gamma}{N_q}, \text{ then } \sigma'_{end} = \sigma'_{vo}N_q + \gamma' t N_\gamma \quad (5)$$

Where σ'_{vi}, σ'_{vo}, N_q and N_γ are calculated as follows:

$$\sigma'_{vi} = \gamma' Z_i \left(\exp\left(\frac{z}{Z_i}\right) - 1\right) \quad (6)$$

$$\sigma'_{vo} = \gamma' Z_o \left(\exp\left(\frac{z}{Z_o}\right) - 1\right) \quad (7)$$

$$N_q = \frac{\exp(2\pi(0.75-(\varphi/360))\tan\varphi)}{2\cos^2(45+(\varphi/2))} \quad (8)$$

$$N_\gamma = \frac{2(N_q+1)\tan\varphi}{1+0.4\sin 4\varphi} \quad (9)$$

As stated before, now the total resistance force can be compared to the total weight of the system and once the total resistance exceeds the weight of the system, the initial penetration depth is reached.

4.2.4 Suction assisted installation

Houlsby and Byrne (2005a) also provide formulas to calculate the maximum penetration depth after suction. Their proposed formula for the suction assisted phase is as follows:

$$V' + s\left(\frac{\pi D_i^2}{4}\right) = \left(\gamma' + \frac{as}{h}\right) Z_o^2 \left[\exp\left(\frac{h}{Z_o}\right) - 1 - \left(\frac{h}{Z_o}\right)\right] (K \tan\delta)_o (\pi D_o) +$$

$$\left(\gamma' + \frac{(1-a)s}{h}\right) Z_i^2 \left[\exp\left(\frac{h}{Z_i}\right) - 1 - \left(\frac{h}{Z_i}\right)\right] (K \tan\delta)_i (\pi D_i) +$$

$$\left\{\left[\gamma' - \frac{(1-a)s}{h}\right] Z_i \left[\exp\left(\frac{h}{Z_i}\right) - 1\right] N_q + \gamma' t N_\gamma\right\} (\pi D t) \qquad (10)$$

The formula is similar to formula 1, however, a suction component is added to the right side as the suction aids in the process of further penetration. The friction terms are the same as before, however, γ' is replaced by $\gamma' + as/h$ or $\gamma' + (1-a)s/h$. It is assumed that pore pressure inside and outside the caisson is linear with depth and that stress is uniform. Formula 10 is a linear equation in s and can therefore be solved to find penetration depth h (Houlsby & Byrne, 2005a). Factor a used in formula 10 is the flow factor and is calculated as follows:

$$a = \frac{a_1 k_f}{(1-a_1) + a_1 k_f} \qquad (11)$$

Where a_1 is:

$$a_1 = c_0 - c_1\left(1 - \exp\left(-\frac{h}{c_2 D}\right)\right) \qquad (12)$$

According to Houlsby & Byrne (2005) the values for constants c_0, c_1 and $c_2$are as follows: $c_0 = 0.45$, $c_1 = 0.36$ and $c_2 = 0.48$.

4.2.5 Non-cylindrical chambers

The current design of the multi-chambered suction anchor also has non-cylindrical chambers. Literature lacks on calculations for non-cylindrical chambers. However, previous empirical research has shown that holding capacity values calculated with the circumference of the chamber do not significantly differ from holding capacity values calculated with the radius of a cylindrical chamber (Van de Loo, 2020). Therefore, the assumption is made that also for resistance and suction calculations, formulas 1 and 10, the circumference can be used for non-cylindrical chambers. Hence, in formulas 1 and 10 the diameters D, D_i and D_o can be replaced with circumferences C, C_i and C_o for non-cylindrical chambers.

4.2.6 Limitations to installation process

Limitations of the suction assisted phase do exist and have an effect of on the maximum suction force that can be applied. For soils consisting mainly out of sand, the "piping failure" phenomenon might occur, which causes the vertical effective stress inside the caisson at the tip to fall to zero (Houlsby & Byrne, 2005a). Further increasing the suction might cause major inflow of water in the caisson without significant further penetration. Houlsby & Byrne (2005a) propose a condition at which piping failure occurs if it holds:

$$\gamma' h - (1 - a)s = 0 \text{ or equivalently, } s_{crit} = \frac{\gamma' h}{(1-a)} \tag{13}$$

Formula 13 determines the critical suction over penetration depth h. The applied or calculated suction cannot exceed this critical suction, otherwise the installation will fail. The critical suction calculations are based on cylindrical shaped caissons. As discussed before, the critical suction values for chambers with sharp edges are lower since piping failure occurs faster (Van de Loo, 2020).

4.2.7 Operation

Suction foundations can resist vertical, horizontal, moment and torsional actions. However, torsional actions are often small (Van Dijk, 2018). One way to determine the operation performance of the suction foundation is to calculate the holding capacity. The capacity determines the force that the anchor can withstand and if this force exceeds the holding capacity, the anchor will fail and will be pulled out of the seabed (Iskander, 2002). A general formula proposed by Iskander (2002) for holding capacity of cylindrical caisson in undrained conditions is as follows:

$$Q_h = Q_{so} + Q_b + W_c + W_s \tag{14}$$

Where Q_{so} is the side shear on the outside wall of the caisson, Q_b is the tensile bearing capacity and W_c and W_s are the submerged weights of the caisson and the soil plug respectively (Iskander, 2002). Q_{so} and Q_b are calculated as follows (Hung, et al., 2017):

$$Q_{so} = Q_b = \frac{\gamma' h}{2}(K \tan \delta)(\pi D h) \tag{15}$$

The weight of the caisson is already determined, and the weight of the soil plug is calculated as follows:

$$W_s = \left(\frac{\pi D_i^2}{4}\right) h\gamma' \tag{16}$$

4.3 Cohesive soils

The following seven sections discuss the soil characteristics, formulas for the installation phase, limitations to the installation and calculations required for the operation phase for cohesive soils.

4.3.1 Soil characteristics

Cohesive soils are clay like soils and tend to adhere to one another. Particle size is significantly smaller compared to cohesionless particle size. Furthermore, cohesive soils are of low strength and are easily deformable (Bobrowsky & Marker, 2018). Previous research from Hut (2020) indicated three main types of cohesive soils: Kaolin Clay, Nkossa Clay and Qiantang River Silt (Hut, 2020). Each type of soils has different soil parameters. In Table 4.3 below, the relevant parameters are given for each type of soil.

Parameter	**Symbol**	**Unit**	**Kaolin Clay**	**Nkossa Clay**	**Qiantang River Silt**
Effective soil weight	γ'	kN/m^3	6.5	6.0	8.8
Lateral earth pressure coefficient	K	$-$	0.8	0.8	0.8
Angle of friction	φ	$°$	26	19	3.5
Adhesion factor	α	$-$	0.50	0.45	0.50
Mudline shear strength	s_{u0}	kPa	4.0	5.0	6.0
Shear strength rate	ρ	kPa/m	1.50	1.67	1.35

Table 4.3. Soil parameters of Kaolin Clay, Nkossa Clay and Qiantang River Silt (Bobrowsky & Marker, 2018, Houlsby & Byrne, 2005b, Colliat, et al., 1998, Wang, et al., 2014).

4.3.2 Variable explanation

In Table 4.4, all the variables required for the calculations for cohesive soils used in the model are given, together with the corresponding units and a brief definition.

Variable	Unit	Definition
C, C_i, C_o	m	Average, inner, and outer circumference of chamber
D, D_i, D_o	m	Average, inner, and outer diameter of caisson
D_m	m	Diameter multiple for enhanced stress
f	$-$	Bearing capacity correction factor
h	m	Penetration depth of the caisson
N_c	$-$	Bearing capacity factor (cohesion)
N_c^*	$-$	Bearing capacity factor (uplift)
N_q	$-$	Bearing capacity factor (overburden)
s	Pa	Suction within caisson with respect to ambient seawater pressure
s_{crit}	Pa	Critical suction with respect to ambient seawater pressure at which piping failure occurs
s_{u0}	kPa	Shear strength at the mudline
s_{u1}	kPa	Average shear strength over depth of the skirt
s_{u2}	kPa	Shear strength at caisson skirt tip
Q_b	N	Tensile bearing capacity of the soil
Q_h	N	Holding capacity of the caisson
Q_{so}	N	Side shear on the outside wall
t	m	Thickness of the caisson wall
V'	N	Effective vertical load
W_c	N	Submerged weight of the caisson
W_s	N	Submerged weight of the soil
α	$-$	Adhesion factor
γ'	kN/m^3	Effective unit weight of the soil
ρ	kPa/m	Rate of change of shear strength with depth
φ	°	Internal friction angle of the soil

Table 4.4. Variable explanation of the presented formulas 17-25 (Houlsby & Byrne, 2005b, Iskander, 2002, Yuqi, et al., 2020).

4.3.3 Self-weight installation

Self-weight installation is also straightforward for cohesive soils. The adhesion terms on the outside and inside of the wall and the end bearing at the tips are added together. Houlsby and Byrne (2005b) provide a formula for which the self-weight installation forces can be calculated (Houlsby & Byrne, 2005b). The formula is as follows:

$$V' = h\alpha_0 s_{u1}(\pi D_o) + h\alpha_i s_{u1}(\pi D_i) + \left(\gamma' h N_q + s_{u2} N_c\right)(\pi D t) \qquad (17)$$

The adhesion terms are calculated by applying a factor α to the value of the undrained strength. The end bearing is calculated with a N_q and N_c term. For undrained cases, N_q is 1 and can be left out of the equation. s_{u1} and s_{u2} are calculated as follows:

$$s_{u1} = s_{u0} + \rho\frac{h}{2} \qquad (18)$$

$$s_{u2} = s_{u0} + \rho h \qquad (19)$$

These are the average undrained shear strengths between the mudline and depth h and at depth h, respectively.

4.3.4 Suction assisted installation

The suction assisted phase starts when a sufficient seal is present. The calculation process is similar to the cohesionless case. Suction together with the effective vertical load V' cause the caisson to further penetrate the seabed. The formula is as follows (Houlsby & Byrne, 2005b):

$$V' + s(\pi D_0^2/4) = h\alpha_o s_{u1}(\pi D_o) + h\alpha_i s_{u1}(\pi D_i) + \ (\gamma' h + s_{u2} N_c)(\pi D t) \qquad (20)$$

Formula 20 gives a relationship between suction and depth and can be solved for s to determine the required suction for the according penetration depth (Houlsby & Byrne, 2005b). It should be noted that, contrary to the cohesionless case, formulas 17 and 20 do not consider any enhancement of the stress inside or outside the caisson due to frictional terms. Taking it into account makes the formulas more complex and in practice these changes make small differences to the calculations (Houlsby & Byrne, 2005b).

4.3.5 Non-cylindrical chambers

Calculations for non-cylindrical chambers can be performed by using the circumference instead of the diameter (Van de Loo, 2020). D can be replaced by C in formulas 17 and 20 to determine the initial penetration depth and required suction for non-cylindrical chambers.

4.3.6 Limitations to the installation process

The limitation to the installation process for cohesive soils is the “reverse bearing capacity” phenomenon. Essentially, due to the adhering tendency of cohesive soils, the soil will flow into the caisson instead of the caisson further penetrating the seabed. If the difference between vertical stress inside and outside the caisson at the tip level exceeds a certain value, local plastic failure might occur (Houlsby & Byrne, 2005b). Houlsby and Byrne (2005b) propose a formula for which the reverse bearing phenomenon occurs:

$$-s+\gamma' h+\frac{\pi D_i h \alpha_i s_{u1}}{\pi D_i^2/4}=\gamma' h+\frac{\pi D_o h \alpha_o s_{u1}}{\pi\left(D_m^2-D_o^2\right)/4}-N_c^* s_{u2} \tag{21}$$

N_c^* is a bearing capacity factor that is appropriate for the uplift of a buried circular footing. N_c^* is likely to be in the region of 12 times α_o but can vary significantly (Houlsby & Byrne, 2005b). Critical suction for cohesive soils is further investigated by Yuqi et al. (2020). The following formula for critical suction is presented (Yuqi, et al., 2020):

$$s_{crit}=\left(\gamma' h+\frac{4h\alpha_i s_{u1}}{D_i}\right)-\left(\gamma' h+\frac{(1+1.5/D_o)^2\alpha_o s_{u1}}{1+3h/d_o}\right)+$$

$$\left(2+\pi+\sin^{-1}\alpha_o+\sin^{-1}\alpha_i+\sqrt{1-\alpha_o^2}+\sqrt{1-\alpha_i^2}\right)s_{u2} \tag{22}$$

In practice, the method used by Yuqi et al. (2020) gives a lower value of critical suction and will therefore be used in the model.

4.3.7 Operation

Operation performance is determined by calculating the holding capacity of the system. The calculations are similar to the cohesionless case. The general formula is the same as formula 14 (Iskander, 2002):

$$Q_h=Q_{so}+Q_b+W_c+W_s \tag{23}$$

Where Q_{so} is the side shear on the outside wall of the caisson, Q_b is the tensile bearing capacity and W_c and W_s are the submerged weights of the caisson and the soil plug respectively (Iskander, 2002). Q_{so} and Q_b are calculated as follows:

$$Q_{so}=\alpha s_{uo}\pi D_o h \tag{24}$$

$$Q_b=s_{uo}N_c f\left(\frac{\pi D_o^2}{4}\right) \tag{25}$$

The term f is denotes as a correction factor and is determined empirically at a value of 0.7 (Van Dijken, 2020). W_c and W_s are determined in the same way as the cohesionless case, for W_s, formula 16 can be used.

4.4 Slanted Tips

The decision to perform only literature research to answer research sub question SQ1, was made since modelling correct installation behaviour for different tip angles is rather complex. Time could be spent more efficiently during the timespan of this research by performing literature research rather than trying to model the effect of tip angle. Furthermore, due to the complex design of the foundation and inability of performing empirical experiments, it would be difficult to validate the implementation afterwards.

The effect of the tip angle on soil penetration is relatively well understood for cone penetration tests. Cone penetrometers have been used to investigate soil properties for many years (Johnson, 2003). The penetrometers are pushed in the soil and by measuring the resistance soil parameters can be determined. Experiments have indicated that penetration resistance varies as a function of cone angle (Gill, 1968). The effect of the cone angle varies for different soil characteristic as well as the dimensions of the cone (Dorgunoglu & Mitchell, 1974) (Johnson, 2003) (Koolen & Vaandragen, 1984). Figure 4.4 shows the angle that is considered below, note that this angle is different from the angle depicted in Figure 2.3.

The results of performed research on the effects of the angle of cones can be interpreted and used for this research project to determine the optimal angle of the caisson tips. To find the optimal angle, two articles of Dorgunoglu and Mitchell (1974) and Koolen and Vaandragen (1984) are used. Figure 4.2 shows for different tips of a cone of perfectly rough, semi rough, and perfectly smooth surfaces the penetration resistance factors N_c and $N_{\gamma q}$ for different semi-apex tip angles. Assuming the roughness of the tip of the suction anchor is semi-rough, the optimal penetration angle, where the penetration factors are minimal, is for angle $\alpha = 25$ [°] (Dorgunoglu & Mitchell, 1974). The cone that is considered has a total angle of $2\alpha = 50$ [°]. The graph is created for soil with an internal friction angle of 30 [°].

The article of Koolen and Vaandragen (1984) measured the cone index values or penetration resistance for several tip angles with equal base area on 67 different agricultural fields. Each cone index was the mean of 5 measurements and the authors present a graph, Figure 4.3, where the results are shown. For a cone angle of 30 [°] the cone index (CI) is minimal (Koolen & Vaandragen, 1984).

It should be noted that Figure 4.2 only gives an approximation for the optimal tip angle of the caisson. The graph is made for cone penetration and the suction caisson is not a cone but an annulus with an angle. Also, as mentioned, the graphs presented in Figure 4.2 hold for soils with an internal friction angle of 30 [°]. The internal friction angles of the soils that are considered are between 36 and 45 [°]. Finally, the roughness of the caisson tip material is unknown. Empirical experiments can be conducted to find the optimal angle, however, an angle between 30 and 50 [°] is assumed to be optimal.

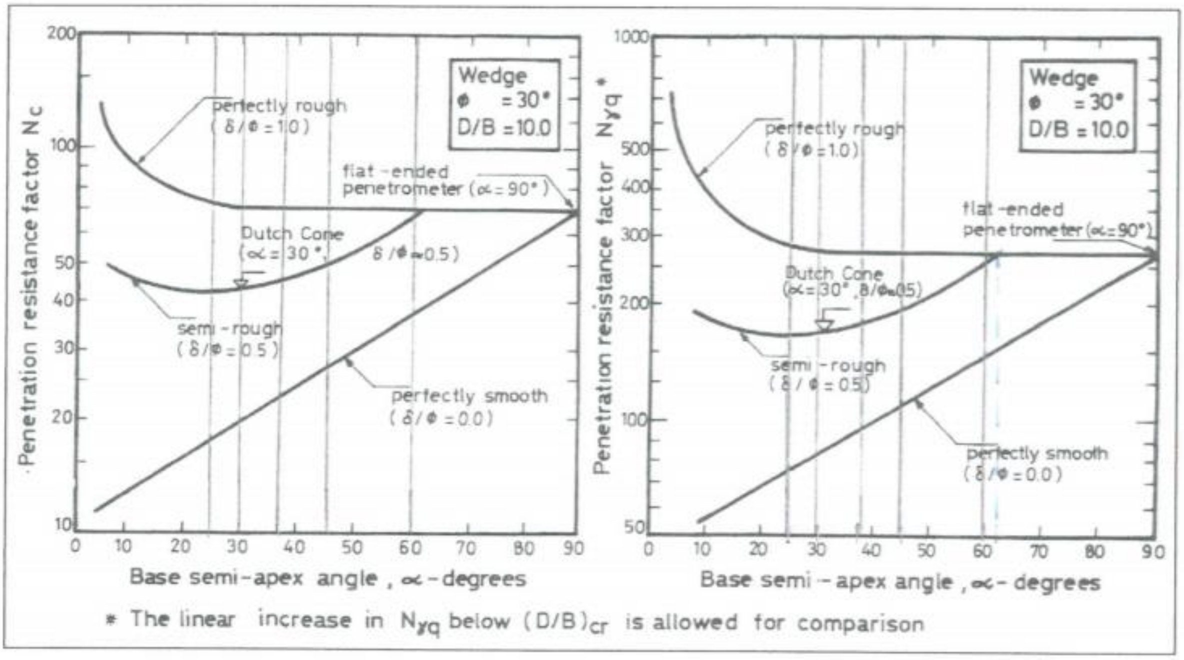


Figure 4.2. Variation of penetration resistance factors with base semi-apex angle (Dorgunoglu & Mitchell, 1974).

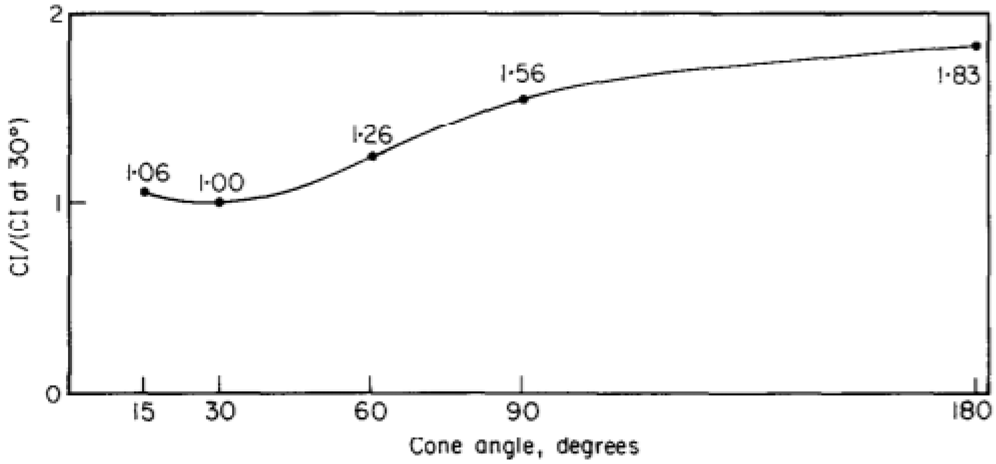


Figure 4.3. Cone index values divided by the cone index at a cone angle of 30 [°] (Koolen & Vaandragen, 1984).

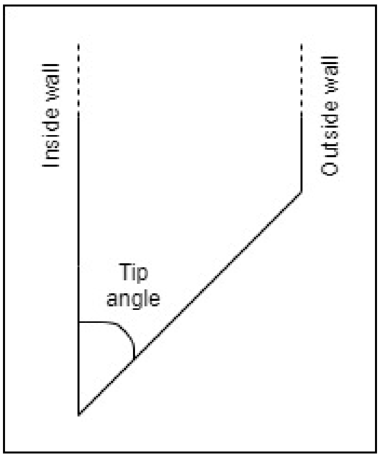


Figure 4.4. The tip angle of the Ocean Battery design, schematically represented.

5. Model setup

The first design change to the system is not a physical change to the system but rather a change for the installation process itself. Currently, suction will be applied to the square chamber or trapezoidal chambers when the critical suction is reached for the previous installation step. Critical suction is a theoretically calculated value. When the real critical suction is lower than the calculated critical suction and applied suction is based on the calculated critical suction, the installation would fail. Piping failure for cohesionless soils, see section 4.2.6, or plug upheave for cohesive soils, see section 4.3.6, would occur.

Exceeding the real critical suction could be prevented by starting the next installation step when a sufficient seal is present for the square or trapezoidal chambers. As discussed in section 4.1, suction can be applied when an initial seal is present that prevents flow of water in the chamber. When a seal is present for the square or trapezoidal chambers, suction can directly be applied and over a larger area.

5.1 Adjusting skirt height

To answer research sub question SQ2, the effect of the height of the square chamber is investigated. The hypothesis of increasing the height of the square skirts is that suction can be applied more gradually and installation would be more likely to succeed by increasing the timespan over which suction can be applied. Once the square skirt height is increased, the installation of the system works as a three-stage rocket instead of only consisting out of two steps. Furthermore, increasing the skirt heights are expected to increase the holding capacity of the foundation.

5.3 Adjusting wall thickness

The next design change that is considered is adjusting the thickness of the walls, to answer research sub question SQ3. Decreasing the thickness of the walls reduces the tip area, the tip resistance force, and therefore total resistances. Lower resistance leads to increased initial penetration depth and lower required suction. However, the total volume and weight of the chambers decrease likewise. Lower weight of the system decreases the total weight of the system and leads to higher required suction and higher initial penetration depth. To cancel out the effect of decreased weight, the system is set to a constant weight so that only the effect of decreased tip resistance is captured.

Additionally, the thickness of the plates ensure structural strength and stiffness and should of course not be compromised to obtain the best possible installation performance. However, strength analysis of the system is not included in this research but should be a point of consideration.

5.3 Chamber Dimensions

Chamber dimensions adaptations are also investigated to see what their relative effect is on the installation process, research sub question SQ4 will then be answered. Two changes are investigated, increasing, and decreasing the radius of the cylindrical chambers and the width of the square chamber. Both changes are investigated separately. When the radius of the cylindrical chambers is changed, the rest of the dimensions of the foundation are kept at their original lengths. The total length and width of the foundation does not change, however, due to the increasing or decreasing radius, the connecting skirts between two cylindrical chambers or a cylindrical chamber and the square chamber adapt accordingly. The results should show whether the increased or decreased area over which suction can be applied at different stages, improves the installation process. The overall area will remain approximately the same, however, the areas of the different chambers will change.

5.4 Assumptions

5.4.1 Flat tips

Slanted tips are included in this research, however, these will not be implemented in the model, as discussed in section 4.4. The current design of prototype 2 incorporates slanted tips but this will not be included in the model. The slanted tips will improve installation, hence, the calculated suction resulting from the model is expected to be an overestimation.

5.4.2 Non-cylindrical shaped chambers

For non-circular shapes, the circumference is used instead of the radius for cylindrical chambers as proposed by Houlsby & Byrne (2005). Empirical experiments showed that the self-weight penetration depth and the holding capacity of square and cylindrical chambers are similar (Van de Loo, 2020). The results of these experiments suggest that using the circumference for non-circular chambers is justified. The circumference is calculated by adding the terms of average circumference separately and for the small part where the outside of the cylindrical chamber is part of the trapezoidal chamber, the formula for calculating a segment is used.

5.4.3 Weight distribution

Weight distribution is assumed to be equal over all the chambers. Although the trapezoidal chambers are larger, it might carry a larger portion of the weight. However, the specific weight distribution of the system is difficult to determine and the weight each chamber holds is therefore not known.

5.4.4 Equal suction for all chambers

The suction that is applied to each chamber is assumed to be equal for all chambers. The system will gradually install in the seabed. Applying different suction to each chamber might be beneficial in real world installation, to overcome seabed unevenness for example, however, in this model it is assumed the system is level after the initial penetration phase and the suction can be applied equally over all the chambers.

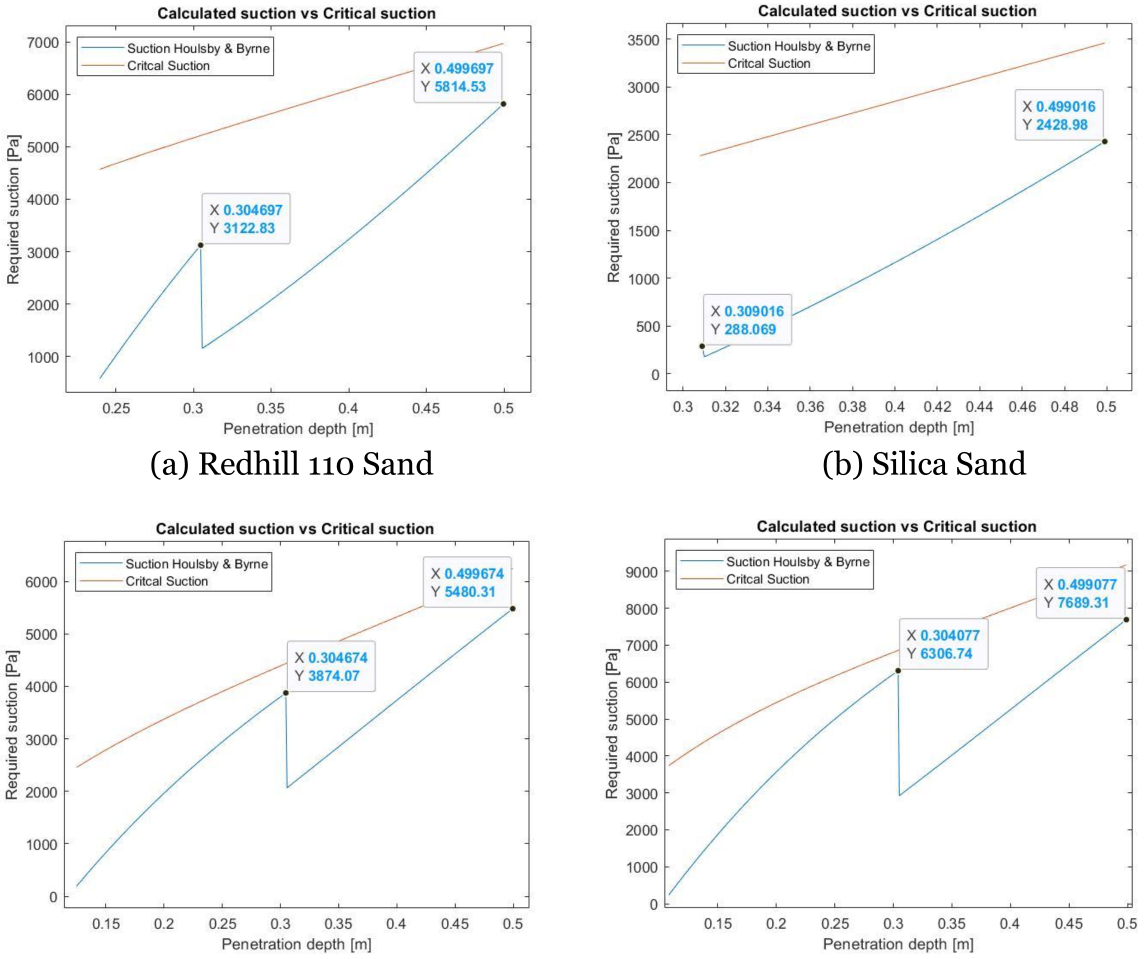
6. Results

This chapter shows all the results from running the MATLAB model. Both type of soils, cohesionless and cohesive, will be investigated.

6.1 Cohesionless Soils

6.1.1 Square skirt height

The square skirt height is increased in increments of 50 [mm] while the heights of the other chambers remain the same, 0.50 and 0.21 [m] for the cylindrical and trapezoidal chambers, respectively. Figure 6.1 shows the starting position where the square skirt height has the same height as the trapezoidal chambers. For all four cohesionless soils, the system can install.



(a) Redhill 110 Sand

(b) Silica Sand

(c) Statoil Sand

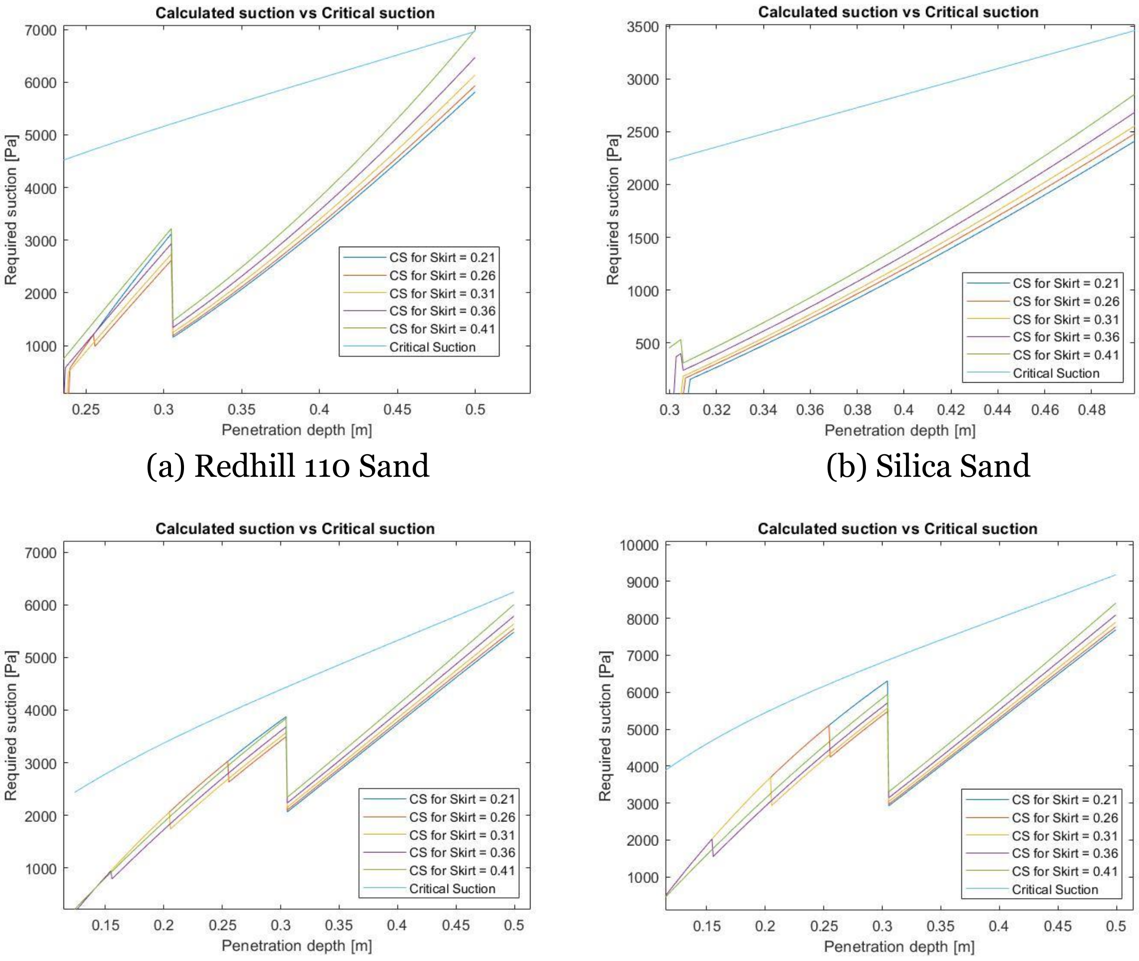
(d) Luce Bay Sand

Figure 6.1. Calculated vs. Critical suction for cohesionless soils, original design with square skirt height of 0.21 [m].

To see the full effect of increasing the square skirt height, different heights are plotted in one figure, see Figure 6.2. The final required suction is maximum when the skirt height is 0.41 [m], for all soils. Especially for installation on Redhill 110 Sand and Silica Sand, increasing the square skirt height does not seem beneficial. Ultimately, installation would fail for a skirt height of 0.41 [m] on Redhill Sand. However, for

Statoil and Luce Bay Sand, benefits of the increased skirt height can be seen, Figure 6.2 (c) and (d). When the skirt heights are equal for the square and trapezoidal chambers, required suction approaches critical suction. Increasing the square skirt height creates a split of that peak and causes the required suction to stay well below the critical suction. This effect validates the hypothesis of the increased skirt height to ensure a more gradual installation process.

This benefit does not hold for Redhill 110 Sand and Silica Sand. The initial penetration depth for these two soils is relatively deep, being approximately 0.24 and 0.31 [m] for Redhill and Silica Sand, respectively. The square chamber already has a sufficient seal after the self-weight penetration phase or reaches the seafloor relatively soon in the suction assisted phase. Therefore, the benefit of the square chamber being an in between step in the installation process does not hold.



(a) Redhill 110 Sand

(b) Silica Sand

(c) Statoil Sand

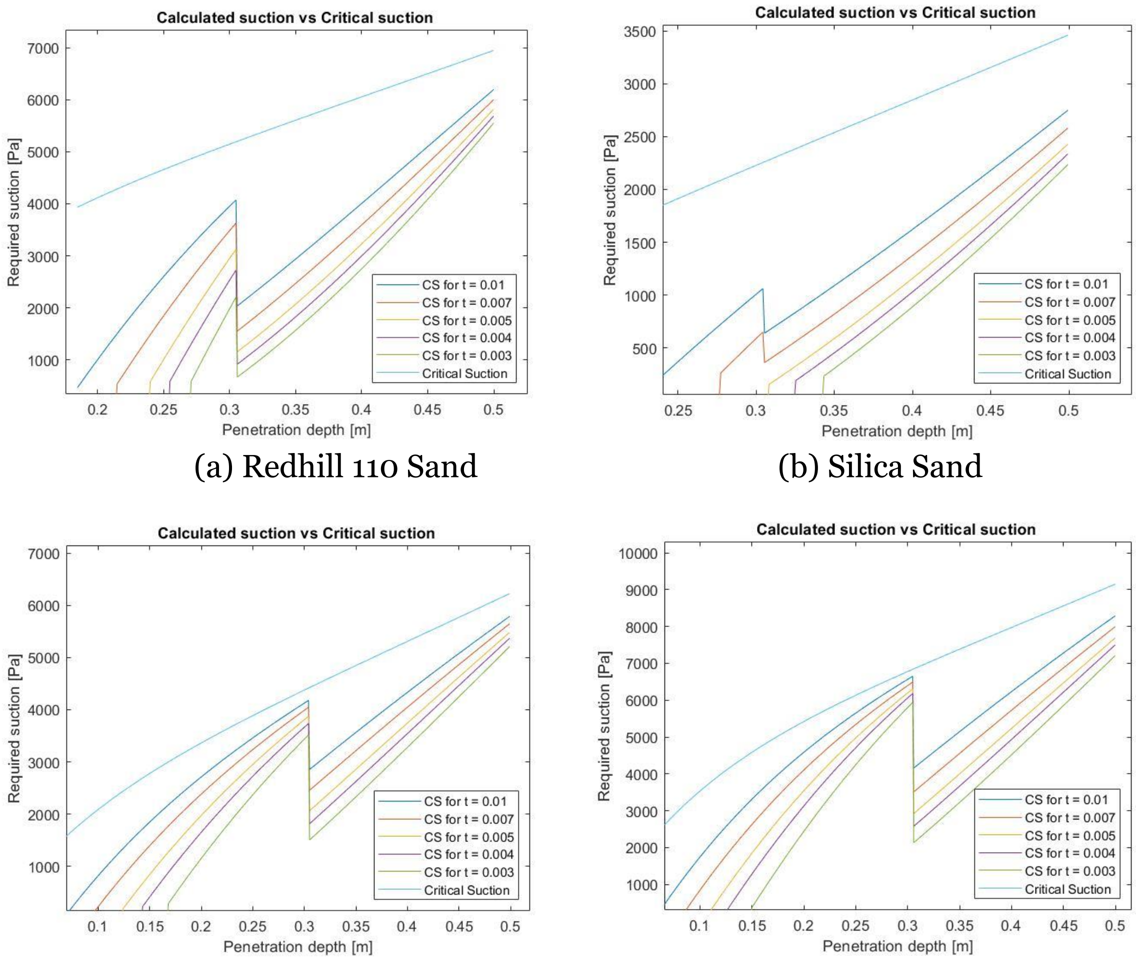
(d) Luce Bay Sand

Figure 6.2. Calculated vs. Critical suction for cohesionless soils with increasing height of the square skirt height, ranging from 0.21 to 0.41 [m].

Hence, for Redhill 110 Sand and Silica Sand it is not beneficial to increase the skirt height, although up to 0.36 [m] the system is still able to install for both soils. However, for Statoil Sand and Luce Bay Sand it is beneficial to increase the square skirt height. The system can install for all heights, however, when the square skirt height is increased, the distance to the critical suction line is smaller.

6.1.2 Plate thickness

The plates in the current design have a thickness of 5 [mm]. The results below, Figure 6.3, show the required suction for thickness of the plates from 3 to 10 [mm] for the four cohesionless soils. Decreasing the thickness of the wall decreases the calculated suction for all four cohesionless soils. Furthermore, the initial penetration depth significantly increases. The calculated suction lines move further to the right for a decrease in thickness, which means that suction starts at a further penetration depth.



(a) Redhill 110 Sand

(b) Silica Sand

(c) Statoil Sand

(d) Luce Bay Sand

Figure 6.3. Calculated vs. Critical suction for cohesionless soils for different thicknesses of the chamber wall, ranging from 3 to 10 [mm].

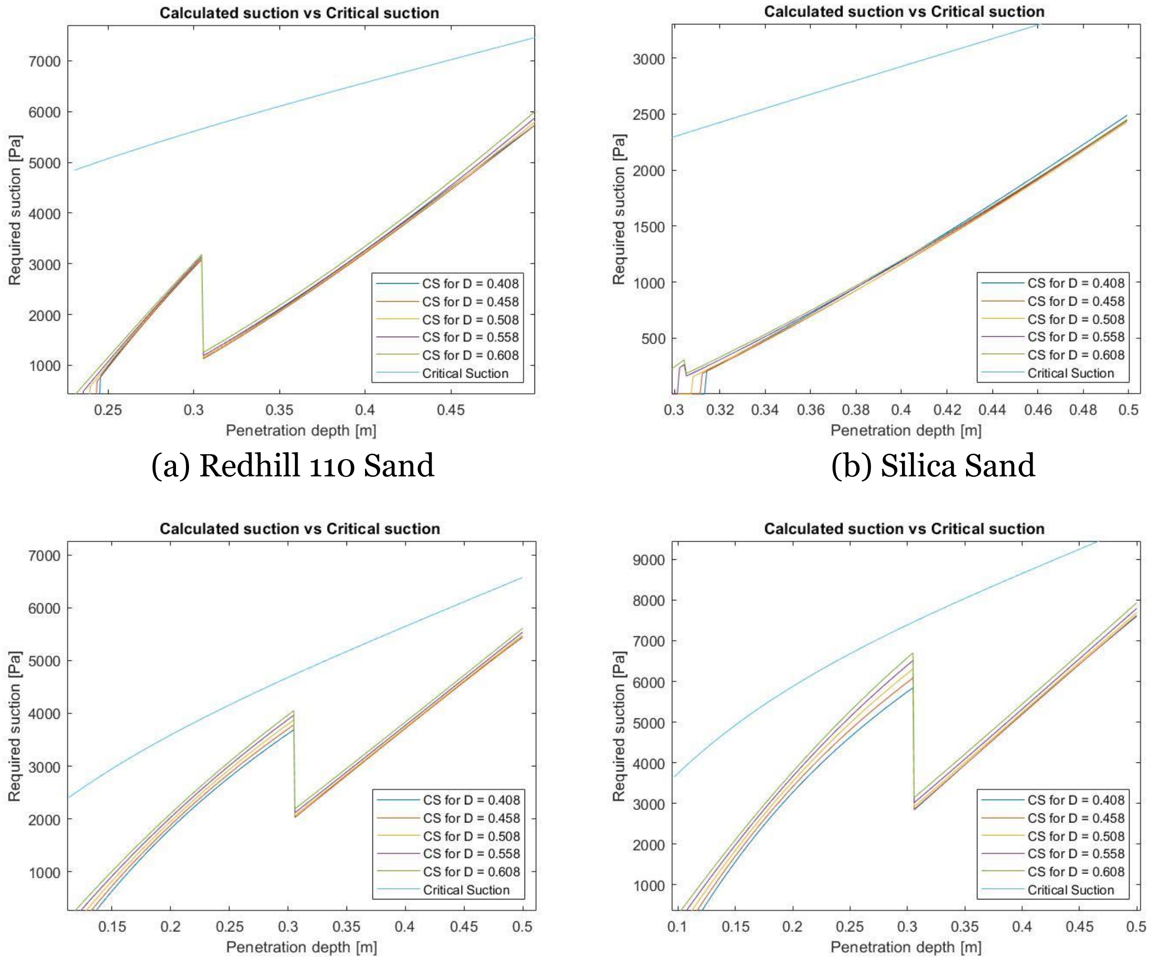
Decreasing the thickness of the plates is beneficial to the installation process. However, decreasing the thickness of the plates influences the overall strength of the system and should be considered.

It should be noted that critical suction also depends on the thickness of the plates, however, the changes are insignificant. Furthermore, the critical suction is already an estimation and only serves as a comparison purpose.

6.1.3 Cylindrical chamber diameter

The overall area to which suction is applied through the whole installation process remains approximately equal, however, at the different stages the area to which suction can be applied changes by changing the area of the cylindrical and square chambers.

The cylindrical chambers are located in the corners of the foundation and will penetrate the soil first. Figure 6.4 shows the results of five different values of the cylindrical chamber diameter.



(a) Redhill 110 Sand

(b) Silica Sand

(c) Statoil Sand

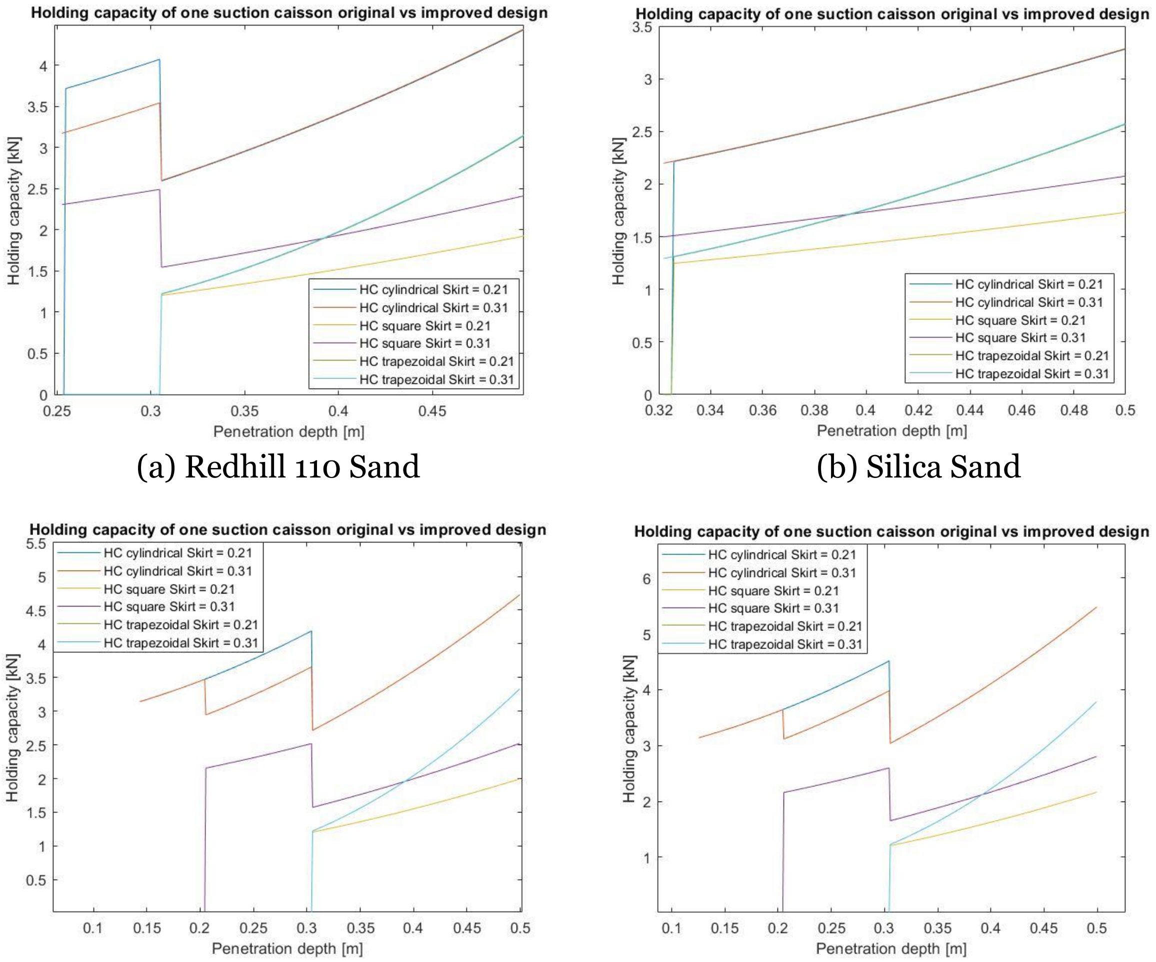
(d) Luce Bay Sand

Figure 6.4. Calculated vs. Critical suction for cohesionless soils for different diameters ranging from 0.408 to 0.608 [m] of the cylindrical chambers.

Increasing the diameter decreases the initial penetration depth of the system, which is due to the increase in inside and outside wall area and tip surface. The system is however quite robust to the investigated change in the suction assisted phase. Especially, Redhill 110 Sand and Silica Sand, Figure 6.4 (a) and (b) do not show significant changes in calculated suction for different diameters. For Statoil Sand and Luce Bay Sand the changes are more significant and the calculated suction increases with the diameter of the cylindrical chamber. Decreasing the diameter is beneficial to the installation process, however, the changes are of minor influence.

6.1.4 Holding Capacity

The before-mentioned design changes also influence the final holding capacity of the system. An improvement in installation does not necessarily mean that the overall performance of the system improves. The holding capacity is now investigated by means of a sensitivity analysis. For each design change, the original holding capacity is compared to the holding capacity of the system with an implemented design change. Figure 6.5 shows the comparison for the square skirt height of 0.21 versus 0.31 [m].



(a) Redhill 110 Sand

(b) Silica Sand

(c) Statoil Sand

(d) Luce Bay Sand

Figure 6.5. Comparison of the holding capacities of a single cylindrical chamber, the square chamber, and a single trapezoidal chamber for a square skirt height of 0.21 and 0.31 [m] for cohesionless soils.

The cylindrical chamber holding capacity is lower during the first or second phase of the suction assisted penetration phase for all soils except Silica Sand. However, the final holding capacity is equal for both heights of the square chamber. The trapezoidal chamber holding capacity does not change. The holding capacity of the square chamber increases for the whole installation process, due to the increase in inside volume and wall area of the square chamber. Thus, increasing the square chamber height is beneficial for the operation phase of the system for cohesionless soils. Figure 6.6 shows the sensitivity analysis for the holding capacity when the thickness of the plates is changed. Only the plot for Redhill 110 Sand is shown.

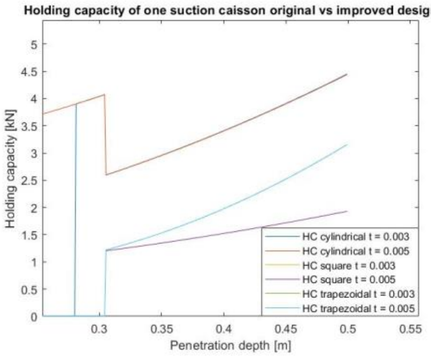
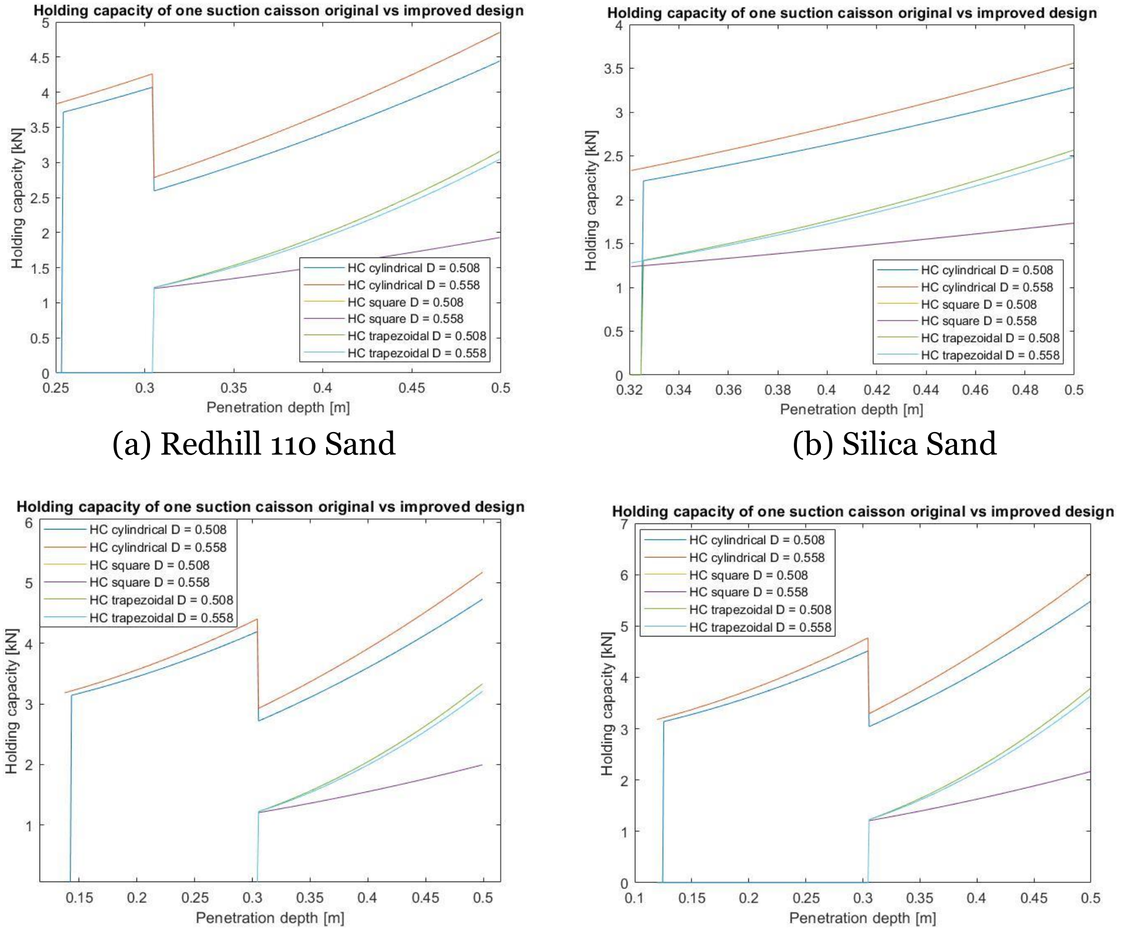


Figure 6.6. Comparison of the holding capacities of a single cylindrical chamber, the square chamber, and a single trapezoidal chamber for a plate thickness of 3 and 5 [mm] for Redhill 110 Sand.

The plate thickness has no effect on the holding capacity. The final holding capacity does not change when the plate thickness is increased or decreased. For all soils, this behaviour is the same and will not be further investigated. Figure 6.7 shows the effect of an increased diameter of the cylindrical chambers on the holding capacity.



(a) Redhill 110 Sand

(b) Silica Sand

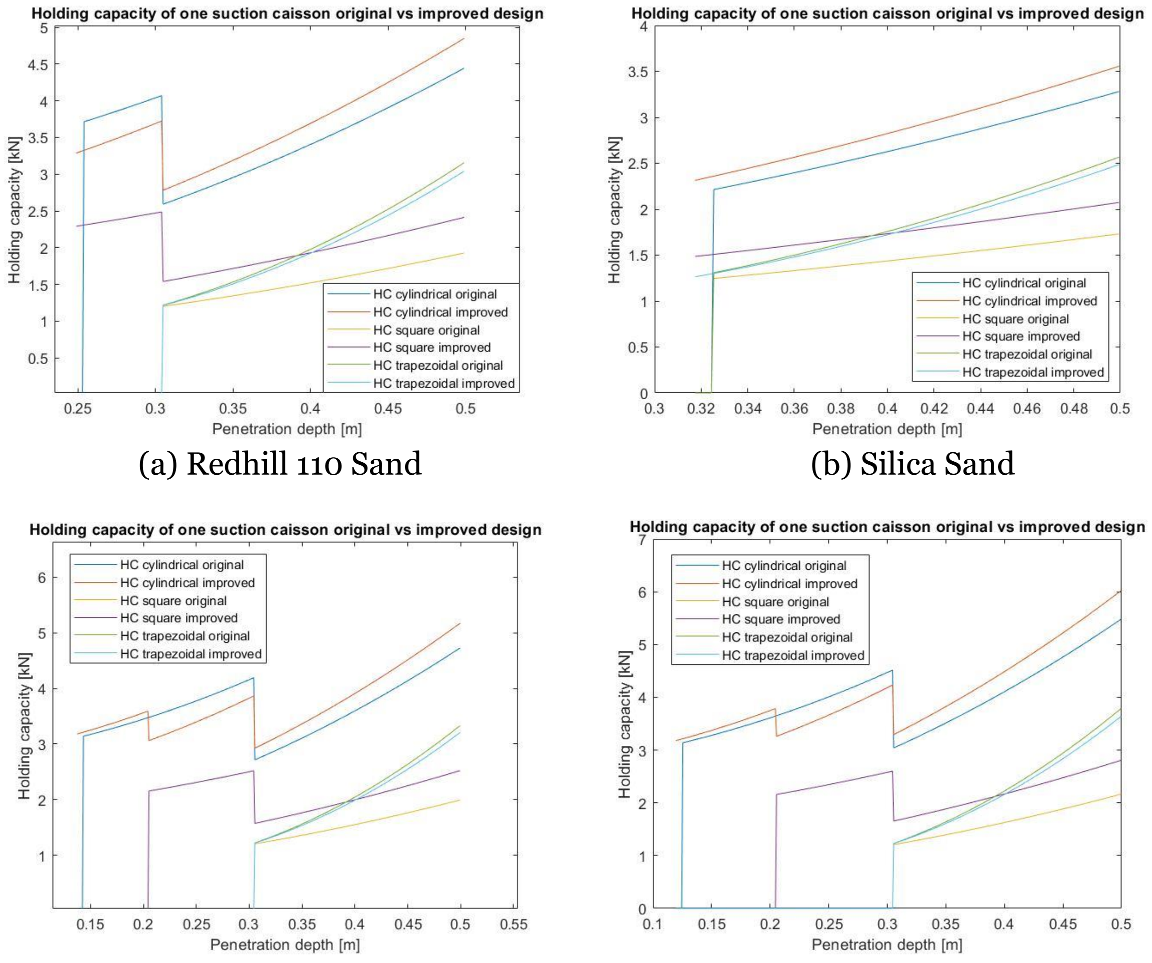
(c) Statoil Sand

(d) Luce Bay Sand

Figure 6.7. Comparison of the holding capacities of a single cylindrical chamber, the square chamber, and a single trapezoidal chamber for a diameter of the cylindrical chamber of 0.508 and 0.558 [m] for cohesionless soils.

An increased diameter shows an increase in the final holding capacity for the cylindrical chamber. The square holding capacity remains equal and the trapezoidal holding capacity decreases slightly. As stated before, the total width and length of the system does not change when the diameter of the cylindrical chambers increases. Therefore, the area of the trapezoidal chambers are decreased. However, the increase in the holding capacity of the cylindrical chambers is greater than the decrease of the holding capacity of the trapezoidal chamber.

Then, the two design changes can be combined to show the final improvement of the holding capacity. The results can be seen in Figure 6.8.



(a) Redhill 110 Sand

(b) Silica Sand

(c) Statoil Sand

(d) Luce Bay Sand

Figure 6.8. Comparison of the holding capacities of a single cylindrical chamber, the square chamber, and a single trapezoidal chamber of the original design versus the improved design with a square skirt height of 0.31 [m] and a cylindrical chamber diameter of 0.558 [m] for cohesionless soils.

The total final capacity of the system can be calculated and compared. The holding capacity is depicted for a singular chamber. Hence, the cylindrical and trapezoidal capacities should be multiplied by four. The results are shown in Table 6.1 below.

		Cylinder HC $[kN]$	Square HC $[kN]$	Trap. HC $[kN]$	Total HC $[kN]$	Percentage Increase
Redhill 110 Sand	Improved	4.849	2.414	3.041	33.974	5.1
	Original	4.443	1.929	3.158	32.333	
Silica Sand	Improved	3.556	2.073	2.486	26.241	4.5
	Original	3.281	1.731	2.567	25.123	
Statoil Sand	Improved	5.172	2.522	3.207	36.038	5.3
	Original	4.727	1.993	3.331	34.225	
Luce Bay Sand	Improved	6.018	2.806	3.637	41.426	5.6
	Original	5.478	2.165	3.786	39.221	

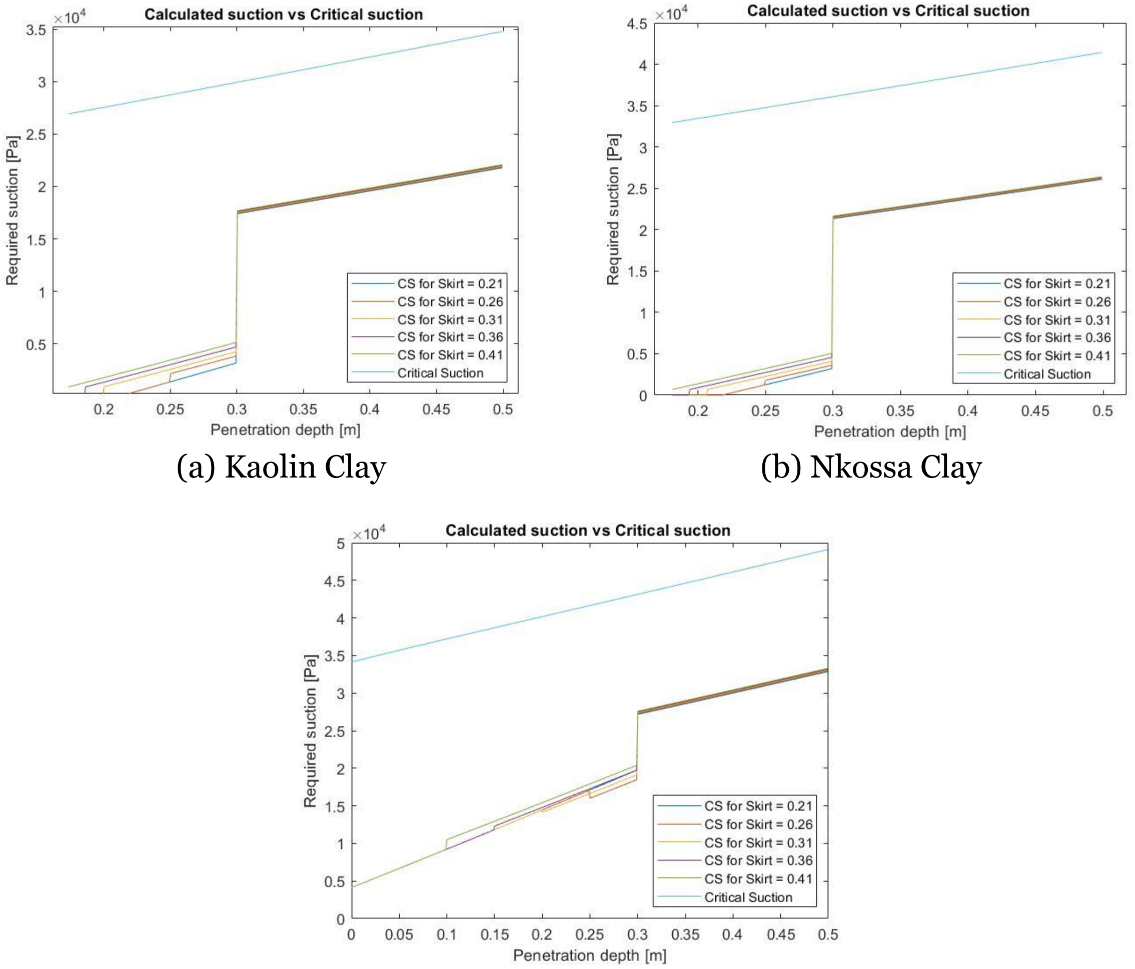
Table 6.1. Comparison of the holding capacity values of the original and improved design where the square chamber increased to 0.31 [m] and cylindrical chamber diameter increased to 0.558 [m] for cohesionless soils.

The improved design has an increase in total holding capacity of approximately 5 percent. The increased diameter of the cylindrical chambers and the increased square skirt height do improve the operation phase of the system.

6.2 Cohesive Soils

6.2.1 Square skirt height

The effect of different square skirt heights is also investigated for cohesive soils. The same method as for the cohesionless soils is used, the square skirt height is increased in increments of 50 [mm]. Figure 6.9 shows the results for the three cohesive soils. For all three soils and all heights of the square chamber, the system can install.



(a) Kaolin Clay

(b) Nkossa Clay

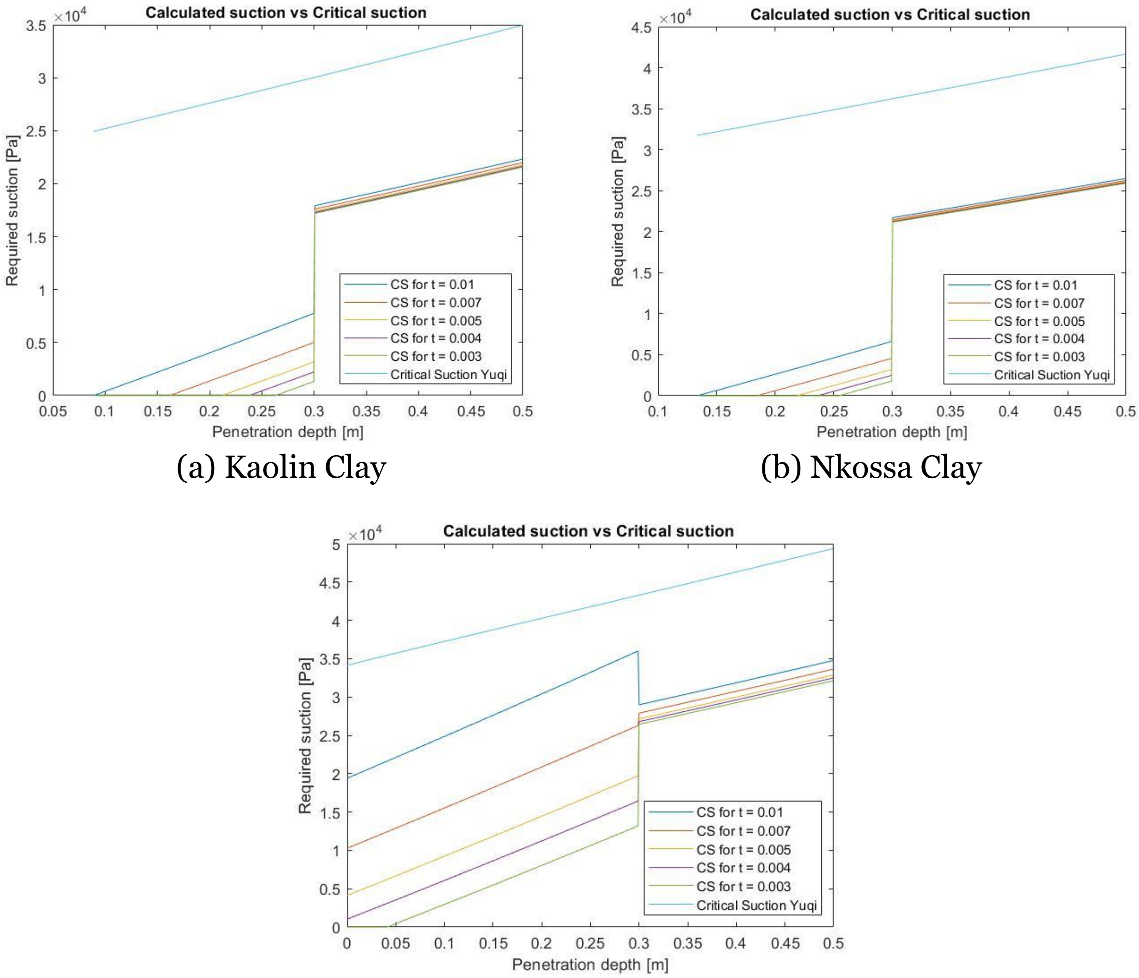
(c) Qiantang River Silt

Figure 6.9. Calculated vs. Critical suction for cohesive soils with increasing height of the square skirt height, ranging from 0.21 to 0.41 [m].

Increasing the square skirt height decreases the initial penetration depth. The square chamber reaches the seabed sooner and increases the penetration resistance when suction is not applied. For the suction assisted phase, the increased square chamber height increases the calculated suction for Kaolin Clay and Nkossa Clay. However, for Qiantang River Silt soil, increasing the square skirt height up until 0.31 [m] decreases the calculated suction. The in between step of applying suction to the square chamber before applying suction to the trapezoidal chamber is beneficial for the installation process for Qiantang river silt. However, the critical suction value is much higher than the calculated suction for all cases and therefore should not impose a problem to installation of the original design.

6.2.2 Plate thickness

Plate thickness is investigated for cohesive soils as well. The thickness is adjusted from 3 to 10 [mm]. The original design uses a thickness of 5 [mm]. Figure 6.10 shows the results for the three cohesive soils.



(a) Kaolin Clay

(b) Nkossa Clay

(c) Qiantang River Silt

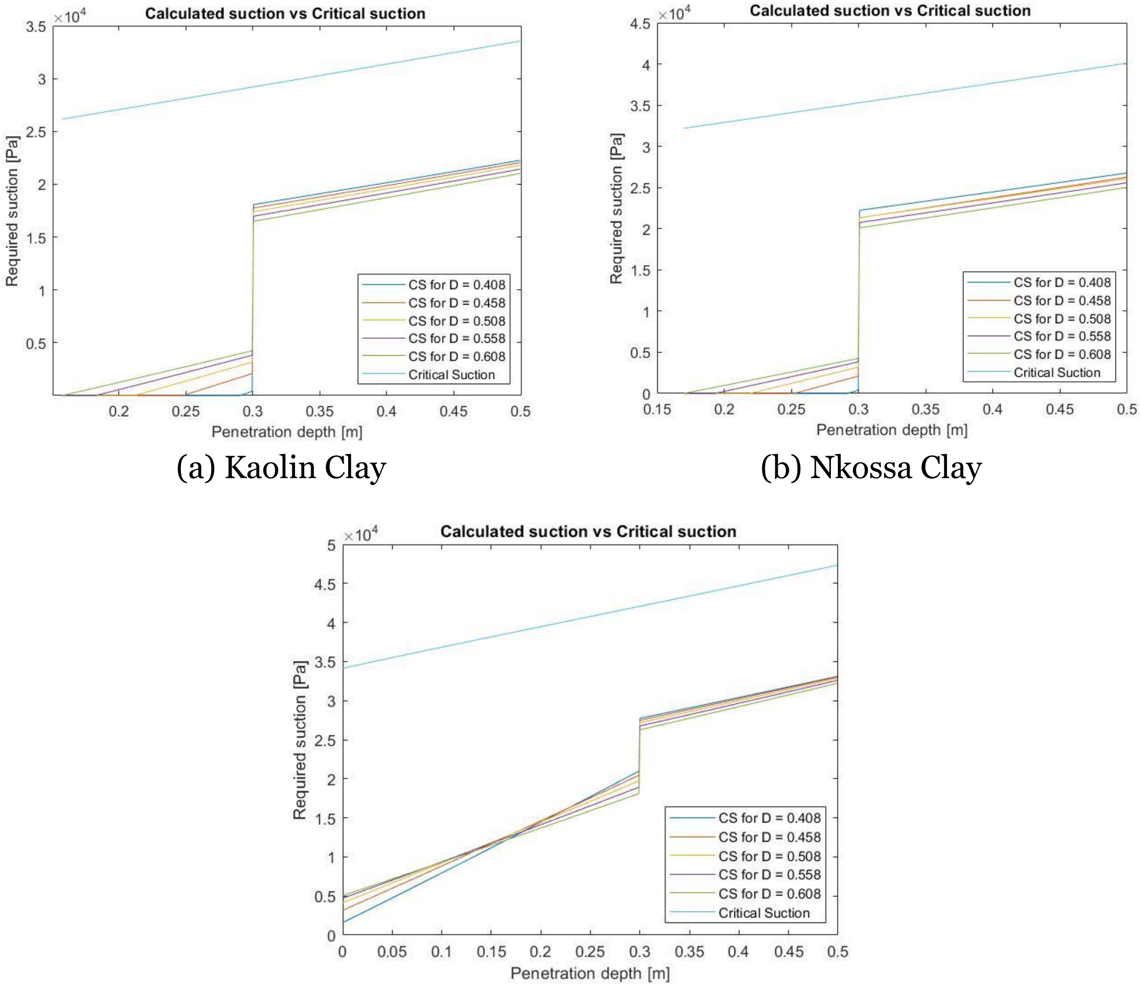
Figure 6.10. Calculated vs. Critical suction for cohesive soils for different thicknesses of the chamber wall, ranging from 3 to 10 [mm].

Like the cohesionless soils, reducing the thickness increases the initial penetration depth and decreases the calculated suction. Especially for Qiantang river silt, the calculated suction decreases significantly. However, all soils can install for all thicknesses. Strength analysis is not part of this project. However, it should be noted that decreasing the wall thickness reduces the overall strength of the system.

The critical suction is also dependant on the thickness of the plates. However, changes are insignificant, and the critical suction is calculated for a thickness of 5 [mm] to keep the graph clear.

6.2.3 Cylindrical chamber diameter

The diameter of the cylindrical chambers is investigated for a range of 0.408 to 0.608 [m]. The total length of the system remains equal to the original design which means that the dimensions of the diagonal skirts and the plates between the cylindrical chambers change accordingly. The results can be seen in Figure 6.11.



(a) Kaolin Clay

(b) Nkossa Clay

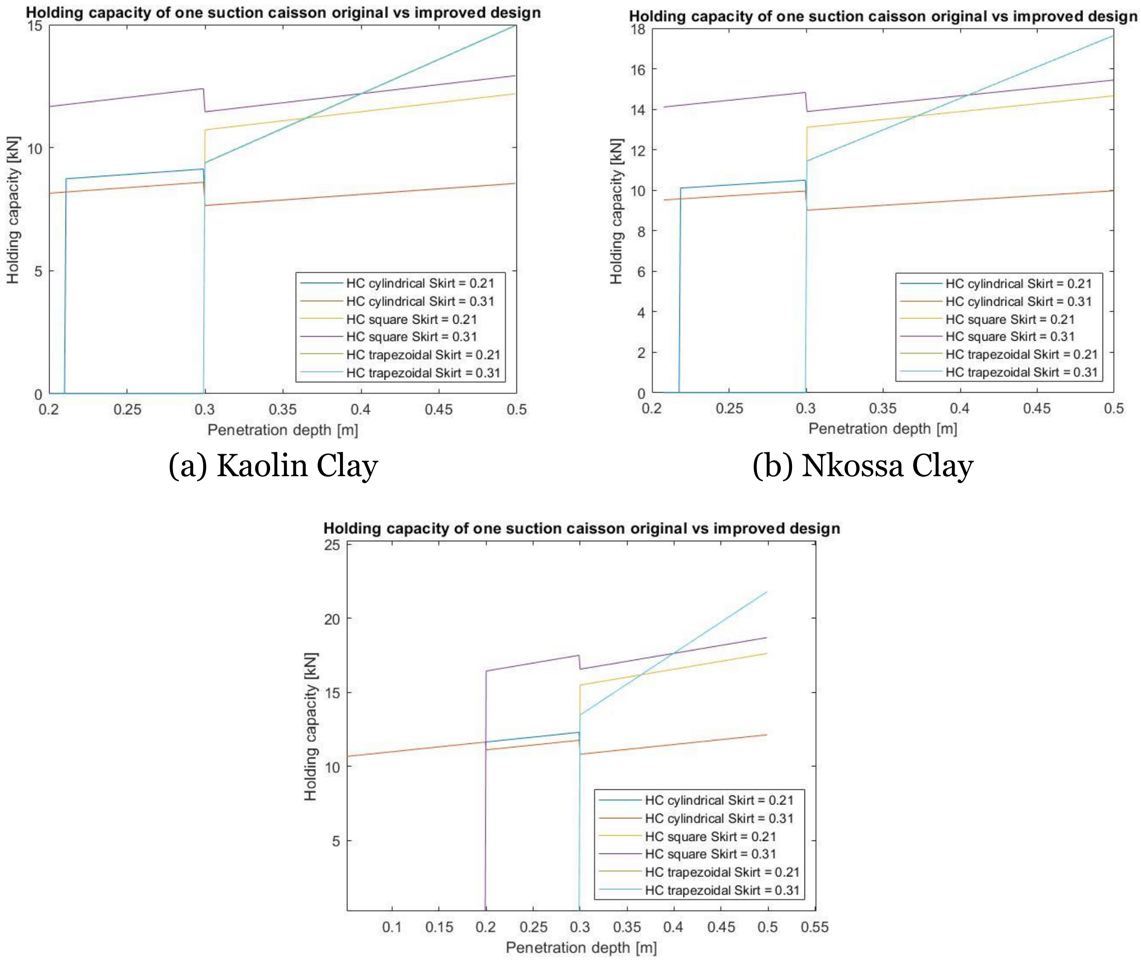
(c) Qiantang River Silt

Figure 6.11. Calculated vs. Critical suction for cohesive soils for different diameters ranging from 0.408 to 0.608 [m] of the cylindrical chambers.

For the first part of the suction assisted installation phase, where only the cylindrical chambers penetrate the seabed, an increased diameter increases the calculated suction. For the second part, where the square and trapezoidal chambers penetrate the seabed, the calculated suction is lower for increasing diameters. For Qiantang river silt, the tip over point occurs even for lower depths, during the first part of the suction assisted phase.

6.2.4 Holding Capacity

The holding capacity is also investigated for cohesive soils. The results are presented in a similar way as for the cohesionless soils. First, the effects of design changes on the holding capacity will be presented. The holding capacities for different heights of the square chamber Figure 6.12, and different diameters of the cylindrical chamber, Figure 6.13, are visualised. The effect of plate thickness will be left out, as it has no effect on the holding capacity.



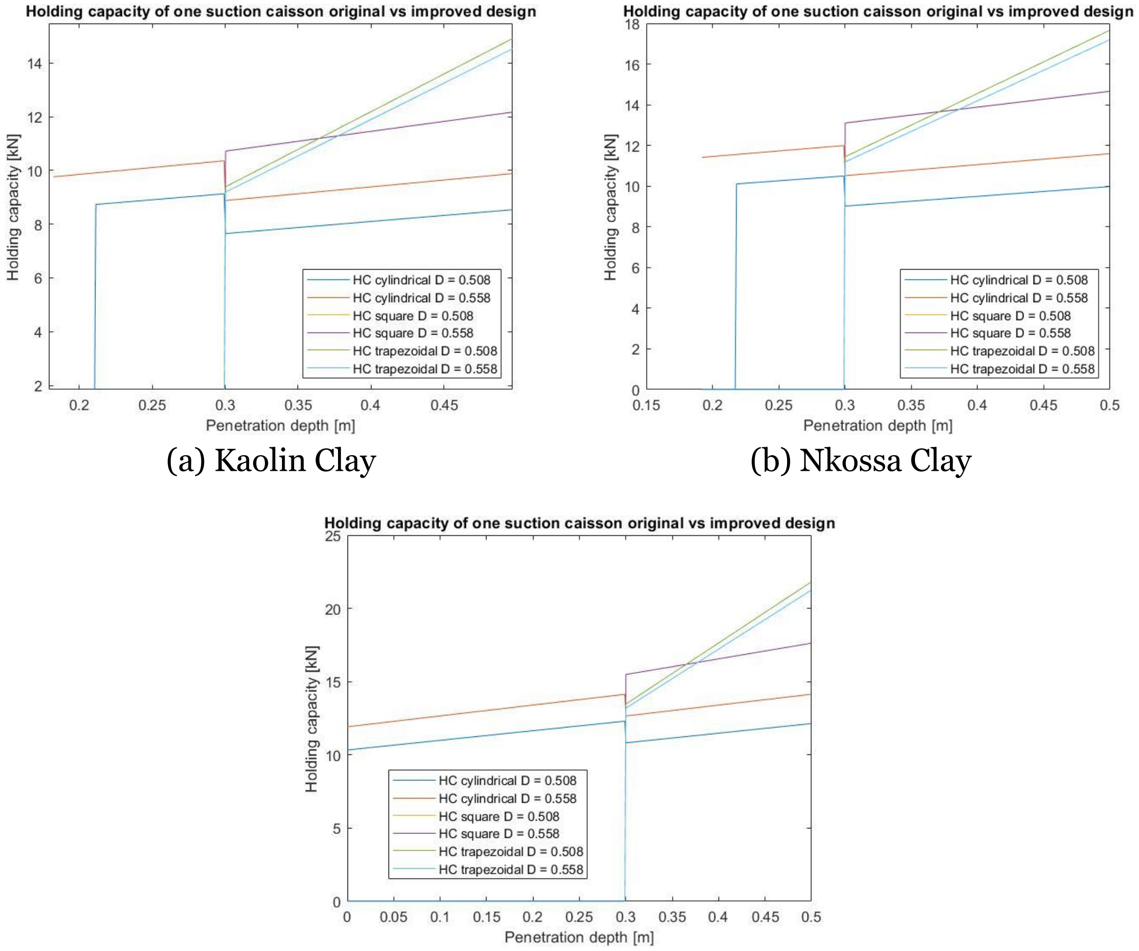
(a) Kaolin Clay

(b) Nkossa Clay

(c) Qiantang River Silt

Figure 6.12. Comparison of the holding capacities of a single cylindrical chamber, the square chamber, and a single trapezoidal chamber for a square skirt height of 0.21 and 0.31 [m] for cohesive soils.

Similar effects as for cohesionless soils can be seen. A small decrease in holding capacity for the cylindrical chamber can be seen when the square chamber reaches the seafloor. However, the final holding capacities is equal for both heights for the cylindrical and trapezoidal chambers. The holding capacity for the square chamber is logically increased for increase in square skirt height.



(a) Kaolin Clay

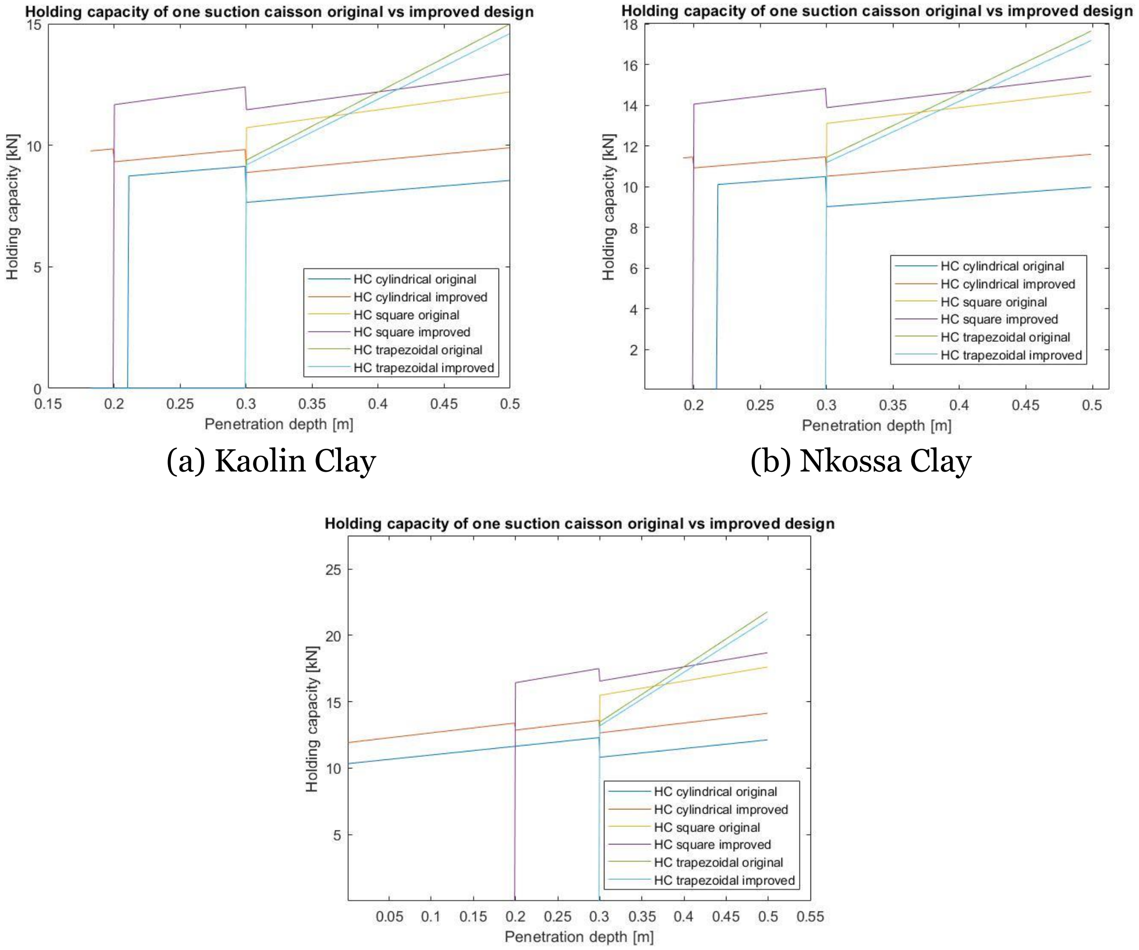
(b) Nkossa Clay

(c) Qiantang River Silt

Figure 6.13. Comparison of the holding capacities of a single cylindrical chamber, the square chamber, and a single trapezoidal chamber for a diameter of the cylindrical chamber of 0.508 and 0.558 [m] for cohesive soils.

The holding capacity of the cylindrical chamber increases for the whole installation process and the trapezoidal holding capacity decreases. However, similar to cohesionless soils, the decrease is smaller than the increase of the cylinder holding capacity.

Now the fully improved design can be compared to the original design for cohesive soils. The results can be seen in Figure 6.14. Table 6.2 shows the final holding capacities per chamber from which the total holding capacity can be calculated for both designs.



(a) Kaolin Clay

(b) Nkossa Clay

(c) Qiantang River Silt

Figure 6.14. Comparison of the holding capacities of a single cylindrical chamber, the square chamber, and a single trapezoidal chamber of the original design versus the improved design with a square skirt height of 0.31 [m] and a cylindrical chamber diameter of 0.558 [m] for cohesionless soils.

When the design changes are implemented, the overall holding capacity increases.

		Cylinder HC $[kN]$	Square HC $[kN]$	Trap. HC $[kN]$	Total HC $[kN]$	Percentage Increase
Kaolin Clay	Improved	9.895	12.927	14.589	110.863	4.3
	Original	8.549	12.192	14.973	106.280	
Nkossa Clay	Improved	11.592	15.436	17.177	130.512	4.3
	Original	9.975	14.661	17.641	125.125	
Qiantang River Silt	Improved	14.133	18.688	21.220	160.100	4.5
	Original	12.131	17.617	21.778	153.253	

Table 6.2. Comparison of the holding capacity values of the original and improved design where the square chamber increased to 0.31 [m] and cylindrical chamber diameter increased to 0.558 [m] for cohesive soils.

The percentage increase is approximately 4 percent, a fraction smaller than the increase for cohesionless soils. However, the operation phase still improves and Table 6.1 and Table 6.2 show improvement for all soils that were investigated.

7. Discussion

Firstly, changing the installation process from being dependant on the critical suction to being dependant on the penetration depth showed quite significant results. As discussed in chapter 5, the critical suction is an estimation and depends heavily on soil parameters. Presumably, it is preferred that the installation process is self-reliant and the critical suction estimation only serves as a check whether the system is able to install. Furthermore, when the square chamber height is increased, making the installation process dependant on penetration depth and the presence of a sufficient seal is more logical than the installation process being dependent on critical suction. The benefit of the in between step that the square chamber creates can be nullified when the installation process is independent of the penetration depth.

Comparing the results of this research to the previously performed analytical research by Hut (2020) and Van Tongeren (2020) and see whether improvement can be seen is difficult. The approaches of the installation process fundamentally differ as previous research in essence attempted to install the foundation with the minimal amount of chambers. However, for many soils, suction needed to be applied to all nine chambers for the system being able to install. Therefore, it seems more logical to assume that suction must be applied to all nine chambers and make the installation as effective as possible by applying suction when a sufficient seal is present. However, a literature gap exists for suction foundations with multiple chambers of different height. The optimal installation process, when the next chambers must be activated, should be investigated empirically.

The initial plan was to investigate the effect of slanted tips on the installation process in the MATLAB model. However, as stated before in section 4.4, due to the uncertainty of correct implementation and the inability of validation, the decision was made to perform literature research. The articles that are discussed are considering cone penetration and dry soil conditions. Therefore, validation is required whether the results of those articles are applicable to caisson penetration in wetted soils.

Increasing the height of the square chamber was proposed by the stakeholders and was therefore implemented in this research. The hypothesis that this design change leads to a more gradual installation process was only be validated with the MATLAB model for three soils; Statoil Sand and Luce Bay Sand, Figure 6.2 (c) and (d) and Qiantang River Silt, Figure 6.9 (c). Apparently, the soils with higher penetration resistance and shallower initial penetration depth seem to benefit from an increased square chamber height. However, as stated before, empirical research must further validate or disproof the hypothesis.

Plate thickness reduction has been investigated by Van Tongeren (2020) as well and showed similar results. However, due to the lack of strength analysis in both previous and this research and lack of improvement in holding capacity it is not advised to change the plate thickness of the design.

Changing the diameter of the cylindrical chambers were not investigated before. The total dimensions of the foundation is kept equal and therefore the total area to which suction can be applied does not change. It was therefore assumed that the holding capacity would not significantly change. However, since the cylindrical chambers are longer, the total wall area of the foundation increases with the diameter and therefore also the holding capacity.

7.1 Limitations

Covid-19 has had a significant impact on the course of this project. The initial goal was to perform empirical experiments on the multi-chambered suction anchor design of the Ocean Battery. Since the design was already investigated analytically, the logical next step would be to perform experiments and validate the models that are already present. However, since the outbreak of the virus, the water hall of the Rijksuniversiteit Groningen, where all experiments would be conducted, is closed for students. Hence, the initial goal had to be adapted such that it was possible to achieve without this physical component. Contributing with another project based on analytical research was challenging. However, results were still gathered from which conclusions can be drawn.

Due to the complexity of the multi-chambered design of the anchor assumptions were made in the model, see section 5.4. These assumptions are not validated currently. Furthermore, although enough literature exists on calculation methods for suction anchors, the combined formulas for the multi-chambered foundation are not validated. Essentially all literature is based on a single cylindrical caisson. Using the circumference instead of diameter for non-cylindrical chambers is validated to some extent with empirical research perform by Van de Loo (2020). However, combining the formulas to calculate required suction for up to nine different chambers of different heights should be validated to some extent before making impactful decisions on the final design.

7.2 Further research

First and foremost, the design and model should be validated with empirical experiments. As discussed in section 7.1, the model is currently not validated and the results of the model are more meaningful when the model is supported with actual proof of describing real world behaviour. The multi-chambered design should be tested with different chamber heights and shapes. Additionally, tests can be performed with different tip angles, to validate the use of the literature on effect of tip angles on cone penetration to determine optimal tip angles of suction foundations.

Furthermore, the installation behaviour on an uneven seabed must be investigated. The system is expected to be able to install on an uneven seabed, assuming suction can be regulated per chamber. The cylindrical chambers can then even the system in the first step of the suction assisted phase. However, applying different suction per chamber is assumed to generate higher stress concentrations in the foundation. Furthermore, previously performed research for Van de Loo (2020) concluded that certain designs of multi-chambered suction anchors showed worrisome results as piping failure occurred earlier than for singular chambered anchors.

8. Proposal

The results in chapter 6 answer research sub questions SQ2, SQ3, SQ4 and SQ5 and together with the answer for sub question SQ1 in section 4.4 the main research question can be answered. This answer will lead to a proposal to change the current design of the foundation of the Ocean Battery. It should be noted that empirical research is required to validate this proposal. However, purely based on the analytical research performed during this project the following design changes are proposed.

Firstly, the most significant improvement of the installation process was the change from activating the next chamber or chambers when critical suction was reached to activation when a sufficient seal is present. The system can install for all soils with this implementation. Additionally, design changes to the system can further improve both the installation and operation process.

From section 4.4 it follows that a tip angle between 30 and 50 [°] is recommended. To determine the optimal angle, empirical experiments are required, and the soil parameters must be investigated at the deployment location. Nonetheless, a tip angle between 30 and 50 [°] should be a good starting point.

Then, section 6.1.1 and 6.2.1 show that a slight benefit in installation behaviour can be gained from increasing the skirt height of the square chamber from 0.21 to 0.31 [m] for soils with a higher penetration resistance. Required suction for soils with a lower penetration increases when the square skirt height increases. However, the system is still able to install in these soils without complications. The holding capacity also increases when the square chamber height is increased, see section 6.1.4 and 6.2.4, which is an important justification to implement this design change.

Section 6.1.2 and 6.2.2 show that thinner plates significantly decrease required suction during the first phase of the suction assisted penetration phase of the installation. The reduction in final required suction is less noticeable. However, strength analysis was not a part of this project. The thesis of Lennard Hut (2020) included strength analysis of the original design and concluded that with a thickness of 5 [mm] the system can install without complications for all soils. However, the impact of reducing the thickness of the plates is not known and should be determined either with a strength analysis or empirically. Additionally, the effect on the holding capacity is insignificant, see section 6.1.4 and 6.2.4. Moreover, since the system is already able to install, a decrease in plate thickness is not recommended.

Finally, section 6.1.3 and 6.2.3 show the effect of different diameters for the cylindrical chamber on the installation process. For the two soil types the effects are different. Cohesionless soil show an increase in final required suction when the diameter is changed, and the opposite holds for cohesive soils. However, the effect on required suction is minimal and the system can install in all cases that were investigated for all soils. The substantial benefit of an increased diameter can be seen in the increase in holding capacity. Therefore, it is recommended to increase the diameter of the cylindrical chambers. Consequently, the plates connecting the cylindrical chambers and the diagonal plates connecting the square and cylindrical chambers decrease to keep the total length and width of the system equal.

To summarize, when the next step in the suction assisted phase starts when a sufficient seal is present and the above-mentioned changes are implemented, the square chamber height is increased to 0.31 [m] and the diameter of the cylindrical chamber is increased to 0.558 [m], the system is fully able to install for all soils that were investigated. Furthermore, the holding capacity of the system increases with approximately 5 and 4 percent for cohesionless and cohesive soils, respectively.

9. Conclusion

The subject of this research was the foundation of the deep water version of the Ocean Battery. Currently, prototype B is being developed which is a 2x2 [m] scale version. The foundation of the prototype is not physically developed yet and, therefore, design changes can still be implemented. By answering the design research questions, the goal of this research project was achieved. The goal reads as follows:

To continue from previously performed analytical research, improve the current models of the installation procedure, and simultaneously propose adaptations for the current design that improve overall performance of the suction anchor.

The MATLAB model was designed by using calculation methods proposed by Houlsby & Byrne (2005a and 2005b) and applying those formulas to the complex design of the Ocean Battery foundation. The model can predict the initial penetration depth, the required suction and the final holding capacity for cohesionless and cohesive soils.

Firstly, the already existing MATLAB model of the installation behaviour was redesigned. The condition to apply suction to the next chamber or chambers was shifted from reaching critical suction to reaching sufficient penetration depth. The new installation process is more robust and better able to install.

Then, through answering the research sub questions, several design changes were investigated with literature research and performing analytical research. The answers lead to a proposal to implement design changes in the current design of the foundation. The new design is able to install in all soils that were investigated with lower required suction than the current design. Furthermore, the new design has a higher holding capacity for all soils.

Unfortunately, due to the current situation with the Covid-19 virus, the original plan of this project, which was to perform empirical experiments, had to be changed to research that could be conducted without physical component. Additionally, the previous and current model are not validated. Validation by literature research is not possible due to the complexity of the design of the foundation and the lack of literature on multi-chambered suction anchors with different chamber heights. Therefore, the results must be analysed with this in mind.

10. Bibliography

Alami, A. H., 2020. *Mechanical energy storage for renewable and sustainable energy resources*. Zwitzerland: Springer Nature.

Alluqmani, A. E., Naqash, M. T. & Harireche, O., 2019. Standard Formulation for the Installation of Suction Caissons in Sand. *Journal of Ocean Engineering and Science,* 4(4), pp. 395-405.

Bahaj, A. B. S., 2011. Generating Electricity from the Oceans. *Renewable and Sustainable Energy Reviews,* 15(7), pp. 3399-3416.

Bhattacharya, A., 2019. *Design of Foundations for Offshore Wind Turbines*. Hoboken (NJ): John Wiley & Sons, Inc..

Bobrowsky, P. & Marker, B., 2018. *Encyclopedia of Engineering Geology*. Springer: Encyclopedia of Engineering Geology.

Dorgunoglu, H. T. & Mitchell, J. K., 1974. Influence of Penetrometer Characteristics on Static Penetration Resistance. *Proceedings of the European Symposium on Penetration Testing,* pp. 133-139.

Gill, W. R., 1968. Influence of compaction hardening of soil on penetration. *Transactions ASAE,* Volume 11, pp. 741-745.

Haldar, S., Patra, S. & Ghanekar, R. K., 2020. *Advances in Offshore Geotechnics: Proceedings of ISOG2019*. Volume 92 ed. Singapore: Springer.

Houlsby, G. T. & Byrne, B. W., 2005a. Design procedures for installation of suction caissons in sand. *Proceedings of the Institution of Civil Engineers-Geotechnical Engineering,* 158(GE3), pp. 135-144.

Houlsby, G. T. & Byrne, B. W., 2005b. Design procedures for installation of suction caissons in clay and other materials. *Proceedings of the Institution of Civil Engineers-Geotechnical Engineering,* 158(GE2), pp. 75-82.

Houlsby, G. T., Kelly, R. B., Huxtable, J. & Byrne, B. W., 2006. Field trials of suction caissons in sand for offshore wind turbine foundations. *Géotechnique,* 56(1), pp. 3-10.

Hung, L. C., Shihoon, L., Tran, N. X. & Kim, S.-R., 2017. Experimental investigation of the vertical pullout cyclic response of bucket foundations in sand. *Applied Ocean Research,* Volume Volume 68, pp. 325-335.

Hut, L. Y., 2020. Investigating the Structural Behaviour of the Ocean Battery. *MSc thesis, Faculty of Science and Engineering, Rijksuniversiteit Groningen.*

Iskander, M. E.-G. S. a. O. R., 2002. Performance of Suction Caissons in Sand and Clay. *Canadian Geotechnical Journal,* 39(3), pp. 576-584.

Johnson, J. B., 2003. *A Statistical Micromechanical Theory of Cone Penetration in Granular Materials,* Hanover, New Hampshire: US Army Corps of Engineers.

Koolen, A. J. & Vaandragen, P., 1984. Relationship between soil mechanical properties. *Journal of Agricultural Engineering Research,* Volume 29, pp. 313-319.

Nogueira, M. R., Flores, A. T. & Perrella Balestieri, J. A., 2020. Pumped hydro storage plants: a review. *Journal of the Brazilian Society of Mechenical Sciences and Engineering,* 42(8).

Ocean Grazer, 2020. *Ocean Energy for a Sustainable Future.* Available at: https://oceangrazer.com/ [Accessed 10-102020].

Oh, K.-Y.et al., 2018. A Review of Foundations of Offshore Wind Energy Convertors: Current Status and Future Perspectives. *Renewable & sustainable energy reviews,* Volume 88, pp. 16-36.

Van de Loo, K., 2020. Laboratory experiments on the technical feasibility of the Ocean Battery suction foundation in sand. *BSc thesis, Rijksuniversiteit Groningen, Faculty of Science and Engineering.*

Van Dijk, B., 2018. Design of Suction Foundations. *Journal of Zhejiang University-SCIENCE A : Applied Physics & Engineering,* 19(8), pp. 579-599.

Van Dijken, N., 2020. Seabed Conditions and its Relation to Suction Anchor Operation. *MSc thesis, Faculty of Science and Engineering, Rijksuniversiteit Groningen.*

Van Tongeren, D. H. J., 2020. Analytical Calculations on the Nine-Chambered Suction Anchor. *BSc thesis, Rijksuniversiteit Groningen, Faculty of Science and Engineering.*

Yuqi, W., Yu, Z. & Dayong, L., 2020. Solution to critical suction pressure of penetrating suction caissons into clay using limit analysis. *Applied Ocean Research,* Volume 101.

11. Appendices

11.1 Appendix A: MATLAB code cohesionless soils

```
%% Master Thesis %% Ocean Grazer's second prototype %%
% Authors: Lennard Hut (S2718960) and Leonard Vos (S3209016)
% Year: 2020

% About this script: simulates the installation behaviour of a suction
% anchor for cohesionless soils. Firstly, the initial penetration depth is
% determined and then the required suction is determined for the suction
% assisted phase. Critical suction is also determined to see whether the
% system can install. The design is fully parametrized, when the
% model is run all dimensions and soil type can be filled in and selected.
% The values of the current design are entered in advance but caan easily
% be altered.


% Case: Full functioning model for predicting penetration behaviour.

close all
clear all
clc

%% Designing the prompts %%
% These prompts will request different parameters of the user, which will
% be coupled to their appropriate matlab parameters.

%--------------------------------------------------------------------
% Save the values in their own variable name for the Caisson
%--------------------------------------------------------------------

% Calling a prompt for the design parameters of the caissons
prompt = {'Inner diameter of the Suction Caisson [m] :', ...
    'Outer diameter of the Suction Caisson [m]:',...
    'Inner height of the Suction Caisson [m]:', ...
    'Outer height of the Suction Caisson [m]:'...
    'Tip angle of the caisson:'};
dlg_title = 'Design Parameters Suction Caisson';
num_lines = 1;
defaultans = {'0.498','0.508','0.50','0.51','0'};
answer = inputdlg(prompt,dlg_title,[1 70],defaultans);


for i = 1:(size(answer))
  data(i) = str2num(answer{i});
end

D_inner = data(1);          % Inner diameter of the caisson [m]
D_outer = data(2);          % Outer diameter of the caisson [m]
H_inner = data(3);          % Inner height of the caisson [m]
H_outer = data(4);          % Outer height of the caisson [m]
Angle_caisson = data(5);    % Angle of the tip of the caisson [degree]

%--------------------------------------------------------------------
% Save the values in their own variable name for the skirts
%--------------------------------------------------------------------

% Calling a second prompt for the design parameters of the skirts
prompt2 = {'Length of the Caisson Connecting Skirts [m]:', ...
    'Length of the Middle Skirts [m]:', ...
    'Length of the Diagonal Connecting Skirts [m]:',...
    'Height of the Square Skirt [m]:',...
    'Height of the Trapezoidal Skirt [m]:',...
    'Thickness of the Skirt [m]:', ...
    'Length of the Platform Plate [m]:','Width of the Platform Plate [m]:',...
    'Thickness of the Platform Plate [m]:',...
    'Seal condition [m]:'};
dlg_title2 = 'Design Parameters Skirts';
num_lines = 1;
defaultans2 = {'1.28','0.61','0.48','0.21','0.21','0.005','1.998',...
    '1.998','0.01','0.005'};
answer2 = inputdlg(prompt2,dlg_title2,[1 80],defaultans2);


for i = 1:(size(answer2))
    data2(i) = str2num(answer2{i});
end

L_plate_outerskirts         = data2(1);     % Length of the outerskirts [m]
L_plate_innerskirts         = data2(2);     % Length of the inner skirts [m]
L_plate_diagonalskirts      = data2(3);     % Length of the diagonal skirts [m]
H_square                    = data2(4);     % Height of the square skirts [m]
H_trap                      = data2(5);     % Height of the trapezoidal skirts [m]
t_plates                    = data2(6);     % Thickness of the skirts [m]
L_plate_platform            = data2(7);     % Length of the platform [m]
W_plate_platform            = data2(8);     % Width of the platform [m]
t_plate_platform            = data2(9);     % Thickness of the platform [m]
seal                        = data2(10);    % Extra depth needed to create a seal
[m]

%-------------------------------------------------------------------------
% Save the values in their own variable name for the remaining
% parameters
%-------------------------------------------------------------------------

% Calling a third promt for other needed parameters
prompt3 = {'Density of the Foundation Material [kg/m^3]:', ...
    'Density of the Sea Water [kg/m^3]', ...
    'Total Weight of the Complete Structure [kg]',};
dlg_title3 = 'Additional Parameters';
num_lines = 1;
defaultans3 = {'7850','1020','1089.9'};
answer3 = inputdlg(prompt3,dlg_title3,[1 80],defaultans3);

for i = 1:(size(answer3))
    data3(i) = str2num(answer3{i});
end

rho_mat = data3(1);                 % Density of the foundation [kg/m^3]
rho_wat = data3(2);                 % Density of the water [kg/m^3]
Total_system_Weight=data3(3);       % Weight of the Ocean Battery [kg]

%-------------------------------------------------------------------------
% Save the values in their own variable name for the remaining
% parameters
%-------------------------------------------------------------------------

list = {'Silica Sand','Statoil Sand','Redhill 110 Sand',...
    'Luce Bay Sand'};
[indx,tf] = listdlg('PromptString',{'Select a soil type.',...
    'Only select one of the soils.'},'SelectionMode','single',...
    'ListString', list, 'ListSize', [300, 300], 'Name', ...
    'Select a soil type');

if indx == 1         % For Silica Sand
        y_komma         = 5993;     % Effective soil weight[kN/m^3]
        Ktandelta       = 0.63;     % [-]
        phi             = 36;       % Angle of friction [degree]
        k_f             = 1;        % Permeability ratio [-]

elseif indx == 2    % For Statoil Sand
        y_komma         = 8500;     % Effictive soil weight [kN/m^3]
        Ktandelta       = 0.8;      % [-]
        phi             = 45;       % Angle of friction [degree]
        k_f             = 3;        % Permeability ratio [-]
```

```
elseif indx == 3    % For Redhill 110 Sand
        y_komma         = 7820;     % Effective soil weight [kN/m^3]
        Ktandelta       = 0.8;      % [-]
        phi             = 36;       % Angle of friction [degree]
        k_f             = 5;        % Permeability ratio [-]

elseif indx == 4    % For Luce Bay Sand
        y_komma         = 10300;    % Effective soil weight [kN/m^3]
        Ktandelta       = 0.8;      % [-]
        phi             = 45;       % Angle of friction [degree]
        k_f             = 5;        % Permeability ratio [-]
else
end

% Calculating the bearing capacity factors
% Bearing capacity factor overburden [-] (Houlsby and Byrne)
Nq = (exp(2*pi*(0.75-(phi/360))*tand(phi)))/...
    (2*(cosd(45+(phi/2))^2));
% Bearing capacity factor self-weight [-] (Houlsby and Byrne)
Ny = (2*(Nq+1)*tand(phi))/(1+0.4*sind(phi));
% Bearing capacity factor cohesion [-]
Nc = (Nq-1)/tand(phi);

%% Calculating different basis parameters
gravity     = 9.81;     % Gravity acceleration constant [m/s^2]
m_SW        = 1.5;      % Multiple of the diameter for enhanced stress [-]

%-----------------------------------------------------------------------
% Calculation of parameters of the suction caissons
%-----------------------------------------------------------------------

% Thickness of the suction caisson walls [m]
t              = ((D_outer-D_inner)/2);
% Average Diameter of the suction caisson [m]
D_av           = ((D_outer+D_inner)/2);
% Height difference between plates and caisson
H_split1        = H_outer - H_square;
H_split2        = H_outer - H_trap;
% Volume calculation of the suction caisson [m^3]
Vol_SC         = (pi*((D_outer/2)^2)*H_outer)-(pi*((D_inner/2)^2)*H_inner);
% Weight calculation of the suction caisson [kg]
Weight_SC      = Vol_SC*rho_mat;
% Vertical load due to the weight of the suction caisson [N]
WeightForce_SC = gravity*Weight_SC;
% Buoyancy force of the submerged suction caisson [N]
Fbuoy_SC       = Vol_SC*rho_wat*gravity;

%-----------------------------------------------------------------------
% Calculations for tip angle
%-----------------------------------------------------------------------

% Tip angle coefficient for the cylindrical caissons
cof_tip        = 1;

%-----------------------------------------------------------------------
% Calculating the circumferences of the plates
%-----------------------------------------------------------------------

% The plates connecting the caissons
% Calculating the outer circumference of the plates between the caissons [m]
C_o_outerskirts  = 0.25*((2*L_plate_outerskirts)+2*(L_plate_outerskirts...
    +(2*t_plates)));
% Caclulating the inner circumference of the plates between the caissons [m]
C_i_outerskirts  = 0.25*((2*L_plate_outerskirts)+2*(L_plate_outerskirts...
    -(2*t_plates)));
% Calculating the average circumference of the plates between the caissons [m]
C_av_outerskirts = 0.25*((C_o_outerskirts+C_i_outerskirts)/2);

% The plates that form a square in the middle
% Calculating the outer circumference of the plates in the middle [m]
C_o_innerskirts  = 0.25*(2*L_plate_innerskirts + 2*(L_plate_innerskirts...
     +(2*t_plates)));
% Calculating the inner circumference of the plates in the middle [m]
C_i_innerskirts  = 0.25*(2*L_plate_innerskirts + 2*(L_plate_innerskirts...
    -(2*t_plates)));
% Calculating the average circumference of the plates in the middle [m]
C_av_innerskirts = 0.25*((C_o_innerskirts+C_i_innerskirts)/2);

% The diagonal plates connecting the middle plates to the skirts
% Calculating the outer circumference of the diagonal plates [m]
C_o_diagonalskirts = 0.25*(2*L_plate_diagonalskirts +...
    2*(L_plate_diagonalskirts+(2*t_plates)));
% Calculating the inner circumference of the diagonal plates [m]
C_i_diagonalskirts = 0.25*(2*L_plate_diagonalskirts +...
    2*(L_plate_diagonalskirts-(2*t_plates)));
% Calculating the average circumference of the diagonal plates [m]
C_av_diagonalskirts = 0.25*((C_o_diagonalskirts+C_i_diagonalskirts)/2);

%-------------------------------------------------------------------------
% Calculating the remaining parameters of the plates and the system
%-------------------------------------------------------------------------

% Calculation of parameters of the plates
% Volume calculation of the skirts considered for 1 caisson [m^3]
Vol_tot_plates          = (4*(L_plate_outerskirts*H_trap*t_plates)) +...
    (4*(L_plate_innerskirts*H_square*t_plates))+...
    (4*(L_plate_diagonalskirts*H_trap*t_plates));
% Weight calculation of the skirt [kg]
Weight_plates           = Vol_tot_plates*rho_mat;
% Vertical load due to the weight of the plate [N]
WeightForce_plates      = Weight_plates*gravity;
% Buoyancy force of the submerged plate [N]
Fbuoy_plates            = Vol_tot_plates*rho_wat*gravity;
% Volume calculation of the platform plate [m^3]
Vol_plate_platform      = (L_plate_platform*W_plate_platform)*t_plate_platform;
% Weight calculation of the platform plate [kg]
Weight_plate_platform   = Vol_plate_platform*rho_mat;
% Vertical load due to the weight of the platform plate [N]
WeightForce_platform    = Weight_plate_platform*gravity;
% Buoyancy force of the submerged platform plate [N]
Fbuoy_platform          = Vol_plate_platform*rho_wat*gravity;

% Calculation of the reservoir weight
Reservoir_weight        = Total_system_Weight-Weight_plate_platform-...
    Weight_plates-(4*Weight_SC);
Reservoir_weight_force  = Reservoir_weight*gravity;
Fbouy_Reservoir         = (Reservoir_weight/rho_mat)*rho_wat*gravity;

% Calculation of total forces
% Total Vertical load [N]
WeightForce_total       = WeightForce_SC + 0.25*WeightForce_plates + ...
     0.25*WeightForce_platform + 0.25*Reservoir_weight_force;
% Total buoyancy load [N]
Fbuoy_total             = Fbuoy_SC + 0.25*Fbuoy_plates + ...
     0.25*Fbuoy_platform + 0.25*Fbouy_Reservoir;

%-------------------------------------------------------------------------
% Calculating the  parameters of square caisson
%-------------------------------------------------------------------------

% Calculating the area of the square caisson
A_SQ_inner = L_plate_innerskirts*(L_plate_innerskirts - (2*t_plates));
A_SQ_outer = L_plate_innerskirts*(L_plate_innerskirts + (2*t_plates));
A_SQ_tip = (A_SQ_outer-A_SQ_inner);

%-------------------------------------------------------------------------
% Calculating the  parameters of trapezoidal caissons
%-------------------------------------------------------------------------
```

```
% Calculating dimensions of the trapezoidal shapes
h_trap = sqrt((L_plate_diagonalskirts)^2-((L_plate_outerskirts-...
    L_plate_innerskirts)/2)^2);
h_rect = (L_plate_platform/2)-(L_plate_innerskirts/2)-h_trap;
angle_skirtcaisson = 45; %degrees

% Calculating the areas of the trapezoidal shapes
A_circsegouter = 0.5*(((pi/180)*angle_skirtcaisson)-sind...
    (angle_skirtcaisson))*((D_inner/2)^2);
A_circseginner = 0.5*(((pi/180)*angle_skirtcaisson)-sind...
    (angle_skirtcaisson))*((D_outer/2)^2);

A_trap_outer = ((L_plate_outerskirts+L_plate_innerskirts)/2)*h_trap;
A_rect_outer = (L_plate_outerskirts*h_rect)-(2*A_circseginner);
A_outer_trapC = A_trap_outer + A_rect_outer;

A_trap_inner = (((L_plate_outerskirts-2*t_plates)+...
    (L_plate_innerskirts-2*t_plates))/2)*(h_trap-t_plates);
A_rect_inner = ((L_plate_outerskirts-2*t_plates)*(h_rect-t_plates))...
    -(2*A_circsegouter);
A_inner_trapC = A_trap_inner + A_rect_inner;

% Calculating the circumferences of the trapezoidal shapes
arc_inner = (angle_skirtcaisson/360)*2*pi*(D_outer/2);
arc_outer = (angle_skirtcaisson/360)*2*pi*(D_inner/2);

C_outer_trap = L_plate_outerskirts+(2*arc_outer)+(2*...
    L_plate_diagonalskirts)+L_plate_innerskirts;
C_inner_trap = (L_plate_outerskirts-2*t_plates)+(2*arc_inner)+(2*...
    (L_plate_diagonalskirts-2*t_plates))+ (L_plate_innerskirts-2*t_plates);
C_av_trap = (C_outer_trap + C_inner_trap)/2;

%% Calculations for the self-weight penetration depth (h_Self_Weight)

%-----------------------------------------------------------------------
% Calculating the Self-Weight penetration phase (Cohesionless case)
%-----------------------------------------------------------------------

Z_inner_SW = (D_inner)/(4*Ktandelta);
Z_outer_SW = ((D_outer)*((m_SW^2)-1))/(4*Ktandelta);

% The next while loop is based on the equations given by Houlsby and Byrne.
% Behind every equation an explanation is given. The obtained self-weight
% penetration depth will be used in a for loop to determine all the
% resistances.

% Initializing the values for all intermediate steps
Q_outer_SW    = 0;
Q_inner_SW    = 0;
Q_tip_SW      = 0;
h_Self_Weight = 0;
Qtot_SW       = 0;

while Qtot_SW < WeightForce_total
% Updating the depth for every step of the loop [m]
h_Self_Weight= h_Self_Weight+0.00000001;

%%% If the plates not reach the seabed in the first phase
if h_Self_Weight < H_split1
Q_outer_plates_SW = 0;
Q_inner_plates_SW = 0;
Q_tip_plates_SW = 0;

% ---------------------------------------------------------------------
% Calculating the outer resistance force [N]
% ---------------------------------------------------------------------
Q_outer_SW = y_komma*(Z_outer_SW^2)*((exp(h_Self_Weight/Z_outer_SW))...
    -1-(h_Self_Weight/Z_outer_SW))*(Ktandelta*pi*D_outer);
```

```
% -----------------------------------------------------------------------
% Calculating the inner resistance force [N]
% -----------------------------------------------------------------------
% Remains the same for flat tips and slanted tips
Q_inner_SW = y_komma*(Z_inner_SW^2)*((exp(h_Self_Weight/Z_inner_SW))...
    -1-(h_Self_Weight/Z_inner_SW))*(Ktandelta*pi*D_inner);

% -----------------------------------------------------------------------
% Calculation of the resistance force at the tips of the caisson [N]
% -----------------------------------------------------------------------
% Calculating the stresses to set the end term. When tips are
% slanted, the coefficient cof_tip is less than 1 and will thereforeq
% decrease the tip forces
% Determining the stress at the inside of the caisson [MPa]
sigma_vikom = y_komma*Z_inner_SW*((exp(h_Self_Weight/Z_inner_SW))-1);
% Determining the stress at the outside of the caisson [MPa]
sigma_vokom = y_komma*Z_outer_SW*((exp(h_Self_Weight/Z_outer_SW))-1);
% Factor for the inequality
Fac = (2*t*Ny)/Nq;
% Specification of the variable x
x = (t/2)+(((sigma_vokom - sigma_vikom)*Nq)/(4*y_komma*Ny));
% Initialization of sigma_end
sigma_end = 0;

if (sigma_vikom-sigma_vokom) < Fac
    sigma_end = sigma_vokom*Nq + y_komma*(t-((2*(x^2))/t))*Ny;
else
    sigma_end = sigma_vokom*Nq + y_komma*t*Ny;
end

% Calculation of the resistance force at the tip [N]
Q_tip_SW = sigma_end*pi*D_av*t*cof_tip;

% -----------------------------------------------------------------------
% Calculating the total resistance force [N]
% -----------------------------------------------------------------------
% Add all the terms
Qtot_SW = Q_outer_SW + Q_inner_SW + Q_tip_SW + Q_outer_plates_SW + ...
    Q_inner_plates_SW + Q_tip_plates_SW + Fbuoy_total;

%%% If the square chamber reaches the sea bed in the SW phase
elseif (H_split1 < h_Self_Weight) && (h_Self_Weight < H_split2)
% -----------------------------------------------------------------------
% Calculating the inner and outer resistance forces for the caisson [N]
% -----------------------------------------------------------------------
Q_outer_SW = y_komma*(Z_outer_SW^2)*((exp(h_Self_Weight/Z_outer_SW))...
    -1-(h_Self_Weight/Z_outer_SW))*(Ktandelta*pi*D_outer);
Q_inner_SW = y_komma*(Z_inner_SW^2)*((exp(h_Self_Weight/Z_inner_SW))...
    -1-(h_Self_Weight/Z_inner_SW))*(Ktandelta*pi*D_inner);

% -----------------------------------------------------------------------
% Calculation of the resistance force at the tips of the caisson [N]
% -----------------------------------------------------------------------
sigma_vikom = y_komma*Z_inner_SW*((exp(h_Self_Weight/Z_inner_SW))-1);
sigma_vokom = y_komma*Z_outer_SW*((exp(h_Self_Weight/Z_outer_SW))-1);
Fac = (2*t*Ny)/Nq;
x = (t/2)+(((sigma_vokom - sigma_vikom)*Nq)/(4*y_komma*Ny));
sigma_end = 0;

if (sigma_vikom-sigma_vokom) < Fac
   sigma_end = sigma_vokom*Nq + y_komma*(t-((2*(x^2))/t))*Ny;
else
   sigma_end = sigma_vokom*Nq + y_komma*t*Ny;
end

Q_tip_SW = sigma_end*pi*D_av*t*cof_tip;

% -----------------------------------------------------------------------
% Calculating the frictional terms of the square chamber [N]
% ----------------------------------------------------------------------
Q_outer_plates_SW = ((y_komma*((h_Self_Weight-H_split1))^2)/2)*...
    Ktandelta*C_o_innerskirts;
Q_inner_plates_SW = ((y_komma*((h_Self_Weight-H_split1))^2)/2)*...
    Ktandelta*C_i_innerskirts;
Q_tip_plates_SW   = ((y_komma*(h_Self_Weight-H_split1)*Nq)+...
    (y_komma*(t_plates/2)*Ny))*(C_av_innerskirts*t_plates);

% ----------------------------------------------------------------------
% Calculating the total resistance force [N]
% ----------------------------------------------------------------------
Qtot_SW = Q_outer_SW + Q_inner_SW + Q_tip_SW + Q_outer_plates_SW + ...
    Q_inner_plates_SW + Q_tip_plates_SW + Fbuoy_total;

%%% If the trapezoidal chambers reach the sea bed in the SW phase
else
% ----------------------------------------------------------------------
% Calculating the inner and outer resistance forces
% ----------------------------------------------------------------------
Q_outer_SW = y_komma*(Z_outer_SW^2)*((exp(h_Self_Weight/Z_outer_SW))...
    -1-(h_Self_Weight/Z_outer_SW))*(Ktandelta*pi*D_outer);
Q_inner_SW = y_komma*(Z_inner_SW^2)*((exp(h_Self_Weight/Z_inner_SW))...
    -1-(h_Self_Weight/Z_inner_SW))*(Ktandelta*pi*D_inner);

% ----------------------------------------------------------------------
% Calculation of the resistance force at the tips of the caisson [N]
% ----------------------------------------------------------------------
sigma_vikom = y_komma*Z_inner_SW*((exp(h_Self_Weight/Z_inner_SW))-1);
sigma_vokom = y_komma*Z_outer_SW*((exp(h_Self_Weight/Z_outer_SW))-1);
Fac = (2*t*Ny)/Nq;
x = (t/2)+(((sigma_vokom - sigma_vikom)*Nq)/(4*y_komma*Ny));
sigma_end = 0;

if (sigma_vikom-sigma_vokom) < Fac
   sigma_end = sigma_vokom*Nq + y_komma*(t-((2*(x^2))/t))*Ny;
else
   sigma_end = sigma_vokom*Nq + y_komma*t*Ny;
end

Q_tip_SW = sigma_end*pi*D_av*t*cof_tip;

% ----------------------------------------------------------------------
% Calculating the frictional terms of the plates combined [N]
% ----------------------------------------------------------------------
Q_outer_plates_SW = ((y_komma*((h_Self_Weight-H_split2))^2)/2)*...
    Ktandelta*(C_o_outerskirts + C_o_diagonalskirts)+...
    ((y_komma*((h_Self_Weight-H_split1))^2)/2)*...
    Ktandelta*C_o_innerskirts;
Q_inner_plates_SW = ((y_komma*((h_Self_Weight-H_split2))^2)/2)*...
    Ktandelta*(C_i_outerskirts + C_i_diagonalskirts)+...
    ((y_komma*((h_Self_Weight-H_split1))^2)/2)*...
    Ktandelta*C_i_innerskirts;
Q_tip_plates_SW   = ((y_komma*(h_Self_Weight-H_split2)*Nq)+...
    (y_komma*(t_plates/2)*Ny))*((C_av_outerskirts...
    +C_av_diagonalskirts)*t_plates)+...
    ((y_komma*(h_Self_Weight-H_split1)*Nq)+...
    (y_komma*(t_plates/2)*Ny))*(C_av_innerskirts*t_plates);

% ----------------------------------------------------------------------
% Calculating the total resistance force [N]
% ----------------------------------------------------------------------
Qtot_SW = Q_outer_SW + Q_inner_SW + Q_tip_SW + Q_outer_plates_SW + ...
    Q_inner_plates_SW + Q_tip_plates_SW + Fbuoy_total;

end
end

%% For loop for the Self-Weight penetration to determine force per depth
%Initializing all arrays, to be able to store the data.
```

```
h_depth = [0:0.00001:h_Self_Weight];

for i = 1:1:length(h_depth)
%%% If the plates not reach the seabed in the first phase
if h_depth(i) < H_split1
V_Q_outer_plates_SW(i) = 0;
V_Q_inner_plates_SW(i) = 0;
V_Q_tip_plates_SW(i) = 0;

% Calculating the inner and outer resistance forces [N]
V_Q_outer_SW(i) = y_komma*(Z_outer_SW^2)*((exp((h_depth(i)/Z_outer_SW)))...
    -1-(h_depth(i)/Z_outer_SW))*(Ktandelta*pi*D_outer);
V_Q_inner_SW(i) = y_komma*(Z_inner_SW^2)*((exp(h_depth(i)/Z_inner_SW))...
    -1-(h_depth(i)/Z_inner_SW))*(Ktandelta*pi*D_inner);

% Calculating the stresses at the inside and outside of the caisson [MPa]
V_sigma_vikom(i) = y_komma*Z_inner_SW*((exp(h_depth(i)/Z_inner_SW))-1);
V_sigma_vokom(i) = y_komma*Z_outer_SW*((exp(h_depth(i)/Z_outer_SW))-1);

Fac = (2*t*Ny)/Nq;
V_x(i) = (t/2)+(((V_sigma_vokom(i) - V_sigma_vikom(i))*Nq)/(4*y_komma*Ny));
V_sigma_end(i) = 0;

% Checking the stresses against the inequality factor
if (V_sigma_vikom(i)-V_sigma_vokom(i)) < Fac
V_sigma_end(i) = V_sigma_vokom(i)*Nq + y_komma*(t-((2*(V_x(i)^2))/t))*Ny;
else
V_sigma_end(i) = V_sigma_vokom(i)*Nq + y_komma*t*Ny;
end

% Calculating the resistance force at the tip and the total upward forces [N].
V_Q_tip_SW(i) = V_sigma_end(i)*pi*D_av*t*cof_tip;

V_Qtot_SW(i) = V_Q_outer_SW(i) + V_Q_inner_SW(i) + V_Q_tip_SW(i) + ...
    V_Q_outer_plates_SW(i) + V_Q_inner_plates_SW(i) + ...
    V_Q_tip_plates_SW(i) + Fbuoy_total;

% Holding Capacity Cylindrical Caisson
R_sos_SW(i) = ((y_komma*h_depth(i))/2)*Ktandelta*pi*D_av*h_depth(i);
R_sis_SW(i) = R_sos_SW(i);
W_c_SW(i) = WeightForce_total;
W_s_SW(i) = (((pi*(D_outer)^2)/4)*h_depth(i))*(y_komma);

H_c_SW(i) = R_sos_SW(i)+R_sis_SW(i)+W_c_SW(i)+W_s_SW(i);
H_c2_SW(i) = H_c_SW(i)/1000;

% Holding Capacity Square Caisson
H_c2_SQ_SW(i) =0;

% Holding Capacity Trapezoidal Caisson
H_c2_trap_SW(i) =0;

%%% If the square plates reach the seabed in the first phase
elseif (H_split1 < h_depth(i)) && (h_depth(i) < H_split2)
% Calculating the inner and outer resistance forces [N]
V_Q_outer_SW(i) = y_komma*(Z_outer_SW^2)*((exp((h_depth(i)/Z_outer_SW)))...
    -1-(h_depth(i)/Z_outer_SW))*(Ktandelta*pi*D_outer);
V_Q_inner_SW(i) = y_komma*(Z_inner_SW^2)*((exp(h_depth(i)/Z_inner_SW))...
    -1-(h_depth(i)/Z_inner_SW))*(Ktandelta*pi*D_inner);

% Calculating the stresses at the inside and outside of the caisson [MPa]
V_sigma_vikom(i) = y_komma*Z_inner_SW*((exp(h_depth(i)/Z_inner_SW))-1);
V_sigma_vokom(i) = y_komma*Z_outer_SW*((exp(h_depth(i)/Z_outer_SW))-1);

Fac = (2*t*Ny)/Nq;
V_x(i) = (t/2)+(((V_sigma_vokom(i) - V_sigma_vikom(i))*Nq)/(4*y_komma*Ny));
V_sigma_end(i) = 0;

% Checking the stresses against the inequality factor
if (V_sigma_vikom(i)-V_sigma_vokom(i)) < Fac
```

```
V_sigma_end(i) = V_sigma_vokom(i)*Nq + y_komma*(t-((2*(V_x(i)^2))/t))*Ny;
else
V_sigma_end(i) = V_sigma_vokom(i)*Nq + y_komma*t*Ny;
end
% Calculating the resistance force at the tip and the total upward forces [N]
V_Q_tip_SW(i) = V_sigma_end(i)*pi*D_av*t*cof_tip;

% Calculating the frictional terms of the square plate [N]
V_Q_outer_plates_SW(i) = ((y_komma*((h_depth(i)-H_split1))^2)/2)*...
    Ktandelta*C_o_innerskirts;
V_Q_inner_plates_SW(i) = ((y_komma*((h_depth(i)-H_split1))^2)/2)*...
    Ktandelta*C_i_innerskirts;
V_Q_tip_plates_SW(i)   = ((y_komma*(h_depth(i)-H_split1)*Nq)+...
    (y_komma*(t_plates/2)*Ny))*(C_av_innerskirts*t_plates);

V_Qtot_SW(i) = V_Q_outer_SW(i) + V_Q_inner_SW(i) + V_Q_tip_SW(i) + ...
    V_Q_outer_plates_SW(i) + V_Q_inner_plates_SW(i) +...
    V_Q_tip_plates_SW(i) + Fbuoy_total;

% Holding Capacity Cylindrical Caisson
R_sos_SW(i) = ((y_komma*h_depth(i))/2)*Ktandelta*pi*D_av*h_depth(i);
R_sis_SW(i) = R_sos_SW(i);
W_c_SW(i) = (4*WeightForce_total)/5;
W_s_SW(i) = (((pi*(D_outer)^2)/4)*h_depth(i))*(y_komma);

H_c_SW(i) = R_sos_SW(i)+R_sis_SW(i)+W_c_SW(i)+W_s_SW(i);
H_c2_SW(i) = H_c_SW(i)/1000;

% Holding Capacity Square Caisson
if h_depth(i) < H_split1
H_c2_SQ_SW(i) = 0;
else
R_sos_SQ_SW(i) = ((y_komma*(h_depth(i)-H_split1))/2)*Ktandelta*...
    (4*C_av_innerskirts)*(h_depth(i)-H_split1);
R_sis_SQ_SW(i) = R_sos_SQ_SW(i);
W_c_SQ_SW(i) = (4*WeightForce_total)/5;
W_s_SQ_SW(i) = (A_SQ_outer)*(h_depth(i)-H_split1)*(y_komma);

H_c_SQ_SW(i) = R_sos_SQ_SW(i)+R_sis_SQ_SW(i)+W_c_SQ_SW(i)+W_s_SQ_SW(i);
H_c2_SQ_SW(i) = H_c_SQ_SW(i)/1000;
end

% Holding Capacity Trapezoidal Caisson
H_c2_trap_SW(i) =0;

%%% If the plates reach the trapezoidal seabed in the first phase
else
% Calculating the inner and outer resistance forces [N]
V_Q_outer_SW(i) = y_komma*(Z_outer_SW^2)*((exp((h_depth(i)/Z_outer_SW)))...
    -1-(h_depth(i)/Z_outer_SW))*(Ktandelta*pi*D_outer);
V_Q_inner_SW(i) = y_komma*(Z_inner_SW^2)*((exp(h_depth(i)/Z_inner_SW))...
    -1-(h_depth(i)/Z_inner_SW))*(Ktandelta*pi*D_inner);

% Calculating the stresses at the inside and outside of the caisson [MPa]
V_sigma_vikom(i) = y_komma*Z_inner_SW*((exp(h_depth(i)/Z_inner_SW))-1);
V_sigma_vokom(i) = y_komma*Z_outer_SW*((exp(h_depth(i)/Z_outer_SW))-1);

Fac = (2*t*Ny)/Nq;
V_x(i) = (t/2)+(((V_sigma_vokom(i) - V_sigma_vikom(i))*Nq)/(4*y_komma*Ny));
V_sigma_end(i) = 0;

% Checking the stresses against the inequality factor
if (V_sigma_vikom(i)-V_sigma_vokom(i)) < Fac
V_sigma_end(i) = V_sigma_vokom(i)*Nq + y_komma*(t-((2*(V_x(i)^2))/t))*Ny;
else
V_sigma_end(i) = V_sigma_vokom(i)*Nq + y_komma*t*Ny;
end
% Calculating the resistance force at the tip and the total upward forces [N]
V_Q_tip_SW(i) = V_sigma_end(i)*pi*D_av*t*cof_tip;
```

```
% Calculating the frictional terms of the trapezoidal plates [N]
V_Q_outer_plates_SW(i) = ((y_komma*((h_depth(i)-H_split2))^2)/2)*...
    Ktandelta*(C_o_outerskirts + C_o_diagonalskirts)+...
    ((y_komma*((h_depth(i)-H_split1))^2)/2)*Ktandelta*C_o_innerskirts;
V_Q_inner_plates_SW(i) = ((y_komma*((h_depth(i)-H_split2))^2)/2)*...
    Ktandelta*(C_i_outerskirts + C_i_diagonalskirts)+...
    ((y_komma*((h_depth(i)-H_split1))^2)/2)*Ktandelta*C_i_innerskirts;
V_Q_tip_plates_SW(i)   = ((y_komma*(h_depth(i)-H_split2)*Nq)+...
    (y_komma*(t_plates/2)*Ny))*((C_av_outerskirts+...
    C_av_diagonalskirts)*t_plates)+...
    ((y_komma*(h_depth(i)-H_split1)*Nq)+...
    (y_komma*(t_plates/2)*Ny))*(C_av_innerskirts*t_plates);

V_Qtot_SW(i) = V_Q_outer_SW(i) + V_Q_inner_SW(i) + V_Q_tip_SW(i) + ...
    V_Q_outer_plates_SW(i) + V_Q_inner_plates_SW(i) +...
    V_Q_tip_plates_SW(i) + Fbuoy_total;

% Holding Capacity Cylindrical Caisson
R_sos_SW(i) = ((y_komma*h_depth(i))/2)*Ktandelta*pi*D_av*h_depth(i);
R_sis_SW(i) = R_sos_SW(i);
W_c_SW(i) = (4*WeightForce_total)/9;
W_s_SW(i) = (((pi*(D_outer)^2)/4)*h_depth(i))*(y_komma);

H_c_SW(i) = R_sos_SW(i)+R_sis_SW(i)+W_c_SW(i)+W_s_SW(i);
H_c2_SW(i) = H_c_SW(i)/1000;

% Holding Capacity Square Caisson
if h_depth(i) < H_split1
H_c2_SQ_SW(i) = 0;
else
R_sos_SQ_SW(i) = ((y_komma*(h_depth(i)-H_split1))/2)*Ktandelta*...
    (4*C_av_innerskirts)*(h_depth(i)-H_split1);
R_sis_SQ_SW(i) = R_sos_SQ_SW(i);
W_c_SQ_SW(i) = (4*WeightForce_total)/9;
W_s_SQ_SW(i) = (A_SQ_outer)*(h_depth(i)-H_split1)*(y_komma);

H_c_SQ_SW(i) = R_sos_SQ_SW(i)+R_sis_SQ_SW(i)+W_c_SQ_SW(i)+W_s_SQ_SW(i);
H_c2_SQ_SW(i) = H_c_SQ_SW(i)/1000;
end
% Holding Capacity Trapezoidal Caisson
if h_depth(i) < H_split2
H_c2_trap_SW(i) = 0;
else
R_sos_trap_SW(i) = ((y_komma*(h_depth(i)-H_split2))/2)*Ktandelta*...
    (C_av_trap)*(h_depth(i)-H_split2);
R_sis_trap_SW(i) = R_sos_trap_SW(i);
W_c_trap_SW(i) = (4*WeightForce_total)/9;
W_s_trap_SW(i) = (A_inner_trapC)*(h_depth(i)-H_split2)*(y_komma);

H_c_trap_SW(i) = R_sos_trap_SW(i)+R_sis_trap_SW(i)+W_c_trap_SW(i)+...
    W_s_trap_SW(i);
H_c2_trap_SW(i) = H_c_trap_SW(i)/1000;
end
end
end

%% Parameters for suction assisted penetration phase

%------------------------------------------------------------------------
% Calculating the Suction Assisted penetration phase (Cohesionless case)
%------------------------------------------------------------------------

c_0 = 0.45;       % coefficient 1 for term a1 (Houlsby & Byrne)
c_1 = 0.36;       % coefficient 2 for term a1 (Houlsby & Byrne)
c_2 = 0.48;       % coefficient 3 for term a1 (Houlsby & Byrne)

% Calculation of flow factor parameters (a and a1)
% maximum penetration depth of the suction caisson [m]
h_max = H_inner;
```

```
a1    = c_0 - c_1*(1-exp((-1)*(h_max/(c_2*D_av))));
% Flow factor form outside to inside of the caisson [-]
a     = (a1*k_f)/((1-a1)+a1*k_f);

%% Calculation of the maximum suction to be applied for 0.5m penetration
% Multiple of the diameter for enhanced stress [-] (A. Altarriba)
m_SA          = 1.5;

Z_inner_SA  = (D_inner)/(4*Ktandelta);
Z_outer_SA  = ((D_outer)*((m_SA^2)-1))/(4*Ktandelta);

%% For loops for the Suction Assisted penetration to determine force per depth
% Create an array for the total suction asssisted penetration phase
h_depth_SA = [h_Self_Weight:0.001:h_max];

if h_Self_Weight < H_seal1
% Only the cylindrical caissons have sufficient penetration depth and seal
% After the self-weight phase

%Determine depth arrays for the installation steps
H_SAC = length([h_Self_Weight:0.001:H_split1]);
H_SAS = length([H_split1+0.001:0.001:H_split2]);
h_CSSA = H_SAC + H_SAS;

% For loop for cylindrical caissons depth
for i = 1:1:H_SAC
% Initializing for the flow factors
a1_SA(i) = c_0 - c_1*(1-(exp((-1)*(h_depth_SA(i)/(c_2*D_av)))));
a_SA(i)  = (a1_SA(i)*k_f)/((1-a1_SA(i))+a1_SA(i)*k_f);
a2_SA = 0.5;

% Calculating the critical suction
s_crit(i) = (y_komma*h_depth_SA(i))/(1-a_SA(i));

% Part where the square plate does not yet reach the seabed
% Initializing the factors as done in the maximization before
V_x1(i) = ((Z_outer_SA)^2)*((exp(h_depth_SA(i)/Z_outer_SA))-1-...
    (h_depth_SA(i)/Z_outer_SA))*(Ktandelta)*(pi*D_outer);
V_x2(i) = ((Z_inner_SA)^2)*((exp(h_depth_SA(i)/Z_inner_SA))-1-...
    (h_depth_SA(i)/Z_inner_SA))*(Ktandelta)*(pi*D_inner);
V_x3(i) = (Z_inner_SA*((exp(h_depth_SA(i)/Z_inner_SA))-1)*Nq);
V_x4(i) = (y_komma*t*Ny);
V_x5(i) = (pi*D_av*t*cof_tip);
V_x6(i) = ((pi*(D_inner^2))/4);

% Initializing the friction forces at the plate
V_outer_plates(i) = 0;
V_inner_plates(i) = 0;
V_tip_plates(i)   = 0;

% Calculating the variable s [Pa]
syms p2
eqn = p2*(V_x6(i)) + WeightForce_total == (y_komma + ((a_SA(i)*p2)/...
    h_depth_SA(i)))*V_x1(i) + (y_komma - (((1-a_SA(i))*p2)/h_depth_SA(i)))...
    *V_x2(i)+ ((((y_komma - (((1-a_SA(i))*p2)/h_depth_SA(i)))*V_x3(i))+...
    V_x4(i))*V_x5(i)) + V_outer_plates(i)+ V_inner_plates(i)+...
    V_tip_plates(i) + Fbuoy_total;

s2(i) = double (solve(eqn,p2));

% Calculating the forces acting within the system
V_Force_SA(i)   = WeightForce_total + s2(i)*(V_x6(i));
V_Q_outer_SA(i) = (y_komma + ((a_SA(i)*s2(i))/h_depth_SA(i)))*V_x1(i);
V_Q_inner_SA(i) = (y_komma - (((1-a_SA(i))*s2(i))/h_depth_SA(i)))*V_x2(i);
V_Q_tip_SA(i)   = ((((y_komma - (((1-a_SA(i))*s2(i))/h_depth_SA(i)))...
    *V_x3(i))+V_x4(i))*V_x5(i));
V_Qtot_SA(i)      = V_Q_outer_SA(i) + V_Q_inner_SA(i) + V_Q_tip_SA(i) ...
    + V_outer_plates(i) + V_inner_plates(i) + V_tip_plates(i)...
    + Fbuoy_total;
```

```
% Holding Capacity Cylindrical Caisson
R_sos_SA(i) = ((y_komma*h_depth_SA(i))/2)*Ktandelta*pi*D_av*h_depth_SA(i);
R_sis_SA(i) = R_sos_SA(i);
W_c_SA(i) = WeightForce_total;
W_s_SA(i) = (((pi*(D_outer)^2)/4)*h_depth_SA(i))*(y_komma);

H_c_SA(i) = R_sos_SA(i)+R_sis_SA(i)+W_c_SA(i)+W_s_SA(i);
H_c2_SA(i) = H_c_SA(i)/1000;

% Holding Capacity Square Chamber
H_c2_SQ_SA(i) = 0;

% Holding Capacity Trapezoidal Chamber
H_c2_trap_SA(i) = 0;
end

% For loop also including square chambers
for i = H_SAC+1:1:h_CSSA
% Initializing for the flow factors
a1_SA(i) = c_0 - c_1*(1-(exp((-1)*(h_depth_SA(i)/(c_2*D_av)))));
a_SA(i)  = (a1_SA(i)*k_f)/((1-a1_SA(i))+a1_SA(i)*k_f);
a2_SA = 0.5;

% Calculating the critical suction
s_crit(i) = (y_komma*h_depth_SA(i))/(1-a_SA(i));
% Determining the multiple of the area
Am_SQ_outer = m_SA*A_SQ_outer;

Z_inner_SQ = (A_SQ_inner/(Ktandelta*(4*C_i_innerskirts)));
Z_outer_SQ = ((Am_SQ_outer-A_SQ_outer)/(Ktandelta*(4*C_o_innerskirts)));

% Setting the factors for the round caisson
V_x1(i) = ((Z_outer_SA)^2)*((exp(h_depth_SA(i)/Z_outer_SA))-1-...
    (h_depth_SA(i)/Z_outer_SA))*(Ktandelta)*(pi*D_outer);
V_x2(i) = ((Z_inner_SA)^2)*((exp(h_depth_SA(i)/Z_inner_SA))-1-...
    (h_depth_SA(i)/Z_inner_SA))*(Ktandelta)*(pi*D_inner);
V_x3(i) = (Z_inner_SA*((exp(h_depth_SA(i)/Z_inner_SA))-1)*Nq);
V_x4(i) = (y_komma*t*Ny);
V_x5(i) = (pi*D_av*t*cof_tip);
V_x6(i) = ((pi*(D_inner^2))/4);

% Same factors as for the round suction caisson, but
% determined for the square one
V_x7(i) = ((Z_outer_SQ)^2)*((exp((h_depth_SA(i)-H_split1)/Z_outer_SQ))...
    -1-((h_depth_SA(i)-H_split1)/Z_outer_SQ))*(Ktandelta)*...
    (4*(C_o_innerskirts));
V_x8(i) = ((Z_inner_SQ)^2)*((exp((h_depth_SA(i)-H_split1)/Z_inner_SQ))...
    -1-((h_depth_SA(i)-H_split1)/Z_inner_SQ))*(Ktandelta)*...
    (4*(C_i_innerskirts));
V_x9(i) = (Z_inner_SQ*((exp((h_depth_SA(i)-H_split1)/Z_inner_SQ))-1)*Nq);
V_x10(i) = (y_komma*t_plates*Ny);
V_x11(i) = (A_SQ_outer-A_SQ_inner);
V_x12(i) = A_SQ_inner;

V_outer_plates(i) = 0;
V_inner_plates(i) = 0;
V_tip_plates(i)   = 0;

% Calculating the new variable s [Pa], based on the full system
% In this equation, the additional terms for the square
% caisson are considered
syms p2
eqn = (p2*V_x12(i)) + 4*(p2*(V_x6(i))) + 4*(WeightForce_total) == 4*...
    ((y_komma + ((a_SA(i)*p2)/h_depth_SA(i)))*V_x1(i) + (y_komma - ...
    (((1-a_SA(i))*p2)/h_depth_SA(i)))*V_x2(i) ...
    + ((((y_komma - (((1-a_SA(i))*p2)/h_depth_SA(i)))*V_x3(i))+V_x4(i))...
    *V_x5(i))) + V_outer_plates(i) + ...
    ((y_komma + ((a2_SA*p2)/(h_depth_SA(i)-H_split1)))*V_x7(i) + ...
    (y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-H_split1)))*V_x8(i) + ...
    ((((y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-H_split1)))*V_x9(i))...
    +V_x10(i))*V_x11(i)))+ V_tip_plates(i) + V_inner_plates(i)...
    + 4*(Fbuoy_total);

s2(i) = double (solve(eqn,p2));

V_Force_SA(i)   = WeightForce_total + s2(i)*(V_x6(i));
V_Q_outer_SA(i) = (y_komma + ((a_SA(i)*(s2(i)))/h_depth_SA(i)))*V_x1(i);
V_Q_inner_SA(i) = (y_komma - (((1-a_SA(i))*(s2(i)))/h_depth_SA(i)))*V_x2(i);
V_Q_tip_SA(i)   = ((((y_komma - (((1-a_SA(i))*(s2(i)))/h_depth_SA(i)))...
    *V_x3(i))+V_x4(i))*V_x5(i));

% Calculating the terms for the square suction caisson
V_Q_outer_SA_SQ(i) = (y_komma + ((a2_SA*(s2(i)))/(h_depth_SA(i)...
    )))*V_x7(i);
V_Q_inner_SA_SQ(i) = (y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x8(i);
V_Q_tip_SA_SQ(i)   = ((((y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x9(i))+V_x10(i))*V_x11(i));

% Calculating the total frictional forces
V_Qtot_SA(i)   = V_Q_outer_SA(i) + V_Q_inner_SA(i) + V_Q_tip_SA(i)...
   + V_outer_plates(i) + V_Q_outer_SA_SQ(i) + V_Q_inner_SA_SQ(i) +...
    V_Q_tip_SA_SQ(i) + V_inner_plates(i) +V_tip_plates(i)+ 4*Fbuoy_total;

% Holding Capacity Cylindrical Caisson
R_sos_SA(i) = ((y_komma*h_depth_SA(i))/2)*Ktandelta*pi*D_av*h_depth_SA(i);
R_sis_SA(i) = R_sos_SA(i);
W_c_SA(i) = (4*WeightForce_total)/5;
W_s_SA(i) = (((pi*(D_outer)^2)/4)*h_depth_SA(i))*(y_komma);

H_c_SA(i) = R_sos_SA(i)+R_sis_SA(i)+W_c_SA(i)+W_s_SA(i);
H_c2_SA(i) = H_c_SA(i)/1000;

% Holding Capacity Square Chamber
R_sos_SQ_SA(i) = ((y_komma*(h_depth_SA(i)-H_split1))/2)*Ktandelta*...
    (4*C_av_innerskirts)*(h_depth_SA(i)-H_split1);
R_sis_SQ_SA(i) = R_sos_SQ_SA(i);
W_c_SQ_SA(i) = (4*WeightForce_total)/5;
W_s_SQ_SA(i) = (A_SQ_outer)*(h_depth_SA(i)-H_split1)*(y_komma);

H_c_SQ_SA(i) = R_sos_SQ_SA(i)+R_sis_SQ_SA(i)+W_c_SQ_SA(i)+W_s_SQ_SA(i);
H_c2_SQ_SA(i) = H_c_SQ_SA(i)/1000;

% Holding Capacity Square Chamber
H_c2_trap_SA(i) = 0;
end

% For loop also including square and trapezoidal chambers
for i = h_CSSA+1:1:length(h_depth_SA)
% Initializing for the flow factors
a1_SA(i) = c_0 - c_1*(1-(exp((-1)*(h_depth_SA(i)/(c_2*D_av)))));
a_SA(i)  = (a1_SA(i)*k_f)/((1-a1_SA(i))+a1_SA(i)*k_f);
a2_SA = 0.5;

% Calculating the critical suction
s_crit(i) = (y_komma*h_depth_SA(i))/(1-a_SA(i));
% Determining the multiple of the area
Am_SQ_outer = m_SA*A_SQ_outer;
Am_trap_outer = m_SA*A_outer_trapC;

Z_inner_SQ = (A_SQ_inner/(Ktandelta*(4*C_i_innerskirts)));
Z_outer_SQ = ((Am_SQ_outer-A_SQ_outer)/(Ktandelta*(4*C_o_innerskirts)));

Z_inner_trap = (A_inner_trapC/(Ktandelta*C_inner_trap));
Z_outer_trap = ((Am_trap_outer-A_outer_trapC)/(Ktandelta*C_outer_trap));


% Setting the factors for the round caisson
V_x1(i) = ((Z_outer_SA)^2)*((exp(h_depth_SA(i)/Z_outer_SA))-1-...
    (h_depth_SA(i)/Z_outer_SA))*(Ktandelta)*(pi*D_outer);
V_x2(i) = ((Z_inner_SA)^2)*((exp(h_depth_SA(i)/Z_inner_SA))-1-...
    (h_depth_SA(i)/Z_inner_SA))*(Ktandelta)*(pi*D_inner);
V_x3(i) = (Z_inner_SA*((exp(h_depth_SA(i)/Z_inner_SA))-1)*Nq);
V_x4(i) = (y_komma*t*Ny);
V_x5(i) = (pi*D_av*t*cof_tip);
V_x6(i) = ((pi*(D_inner^2))/4);

% Same factors as for the round suction caisson, but
% determined for the square one
V_x7(i) = ((Z_outer_SQ)^2)*((exp((h_depth_SA(i)-H_split1)/Z_outer_SQ))...
    -1-((h_depth_SA(i)-H_split1)/Z_outer_SQ))*(Ktandelta)*...
    (4*(C_o_innerskirts));
V_x8(i) = ((Z_inner_SQ)^2)*((exp((h_depth_SA(i)-H_split1)/Z_inner_SQ))...
    -1-((h_depth_SA(i)-H_split1)/Z_inner_SQ))*(Ktandelta)*...
    (4*(C_i_innerskirts));
V_x9(i) = (Z_inner_SQ*((exp((h_depth_SA(i)-H_split1)/Z_inner_SQ))-1)*Nq);
V_x10(i) = (y_komma*t_plates*Ny);
V_x11(i) = (A_SQ_outer-A_SQ_inner);
V_x12(i) = A_SQ_inner;

% Same factors as for the round caisson, but for the trapezoidal chambers
V_x13(i) = ((Z_outer_trap)^2)*((exp((h_depth_SA(i)-H_split2)/Z_outer_trap))...
    -1-((h_depth_SA(i)-H_split2)/Z_outer_trap))*(Ktandelta)*C_outer_trap;
V_x14(i) = ((Z_inner_trap)^2)*((exp((h_depth_SA(i)-H_split2)/Z_inner_trap))...
    -1-((h_depth_SA(i)-H_split2)/Z_inner_trap))*(Ktandelta)*C_inner_trap;
V_x15(i) = (Z_inner_trap*((exp((h_depth_SA(i)-H_split2)/Z_inner_trap))-1)*Nq);
V_x16(i) = (y_komma*t_plates*Ny);
V_x17(i) = (A_outer_trapC-A_inner_trapC);
V_x18(i) = A_inner_trapC;

V_outer_plates(i) = 0;
V_inner_plates(i) = 0;
V_tip_plates(i)   = 0;

% Calculating the new variable s [Pa], based on the full system
% In this equation, the additional terms for the square and
% trapezoidal caisson are considered

syms p2
eqn = 4*(p2*V_x18(i)) +(p2*V_x12(i)) + 4*(p2*(V_x6(i))) + 4*...
    (WeightForce_total) == 4*((y_komma + ((a_SA(i)*p2)/h_depth_SA(i)))...
    *V_x1(i) + (y_komma - (((1-a_SA(i))*p2)/h_depth_SA(i)))*V_x2(i) ...
    + ((((y_komma - (((1-a_SA(i))*p2)/h_depth_SA(i)))*V_x3(i))+V_x4(i))...
    *V_x5(i))) +((y_komma + ((a2_SA*p2)/(h_depth_SA(i)-H_split1)))*V_x7(i)...
    + (y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-H_split1)))*V_x8(i) + ...
    ((((y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-H_split1)))*V_x9(i))+...
    V_x10(i))*V_x11(i)))+ (4*((y_komma + ((a2_SA*p2)/(h_depth_SA(i)...
    -H_split2)))*V_x13(i) + (y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-...
    H_split2)))*V_x14(i) +((((y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)...
    -H_split2)))*V_x15(i))+V_x16(i))*V_x17(i)))) + 4*(Fbuoy_total);

s2(i) = double (solve(eqn,p2));

V_Force_SA(i)   = WeightForce_total + s2(i)*(V_x6(i));
V_Q_outer_SA(i) = (y_komma + ((a_SA(i)*(s2(i)))/h_depth_SA(i)))*V_x1(i);
V_Q_inner_SA(i) = (y_komma - (((1-a_SA(i))*(s2(i)))/h_depth_SA(i)))*V_x2(i);
V_Q_tip_SA(i)   = ((((y_komma - (((1-a_SA(i))*(s2(i)))/h_depth_SA(i)))...
    *V_x3(i))+V_x4(i))*V_x5(i));

% Calculating the terms for the square suction caisson
V_Q_outer_SA_SQ(i) = (y_komma + ((a2_SA*(s2(i)))/(h_depth_SA(i)...
    )))*V_x7(i);
V_Q_inner_SA_SQ(i) = (y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x8(i);
V_Q_tip_SA_SQ(i)   = ((((y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x9(i))+V_x10(i))*V_x11(i));

% Calculating the terms for the trapezoidal suction caissons
```

```
V_Q_outer_SA_trap(i) = (y_komma + ((a2_SA*(s2(i)))/(h_depth_SA(i)...
    )))*V_x13(i);
V_Q_inner_SA_trap(i) = (y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x14(i);
V_Q_tip_SA_trap(i)   = ((((y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x15(i))+V_x16(i))*V_x17(i));

% Calculating the total frictional forces
V_Qtot_SA(i)   = V_Q_outer_SA(i) + V_Q_inner_SA(i) + V_Q_tip_SA(i) +...
    V_Q_outer_SA_trap(i) +V_Q_inner_SA_trap(i) + V_Q_tip_SA_trap(i)+...
    V_Q_outer_SA_SQ(i) + V_Q_inner_SA_SQ(i) + V_Q_tip_SA_SQ(i)...
    + 4*Fbuoy_total;

% Holding Capacity Cylindrical Caisson
R_sos_SA(i) = ((y_komma*h_depth_SA(i))/2)*Ktandelta*pi*D_av*h_depth_SA(i);
R_sis_SA(i) = R_sos_SA(i);
W_c_SA(i) = (4*WeightForce_total)/9;
W_s_SA(i) = (((pi*(D_outer)^2)/4)*h_depth_SA(i))*(y_komma);

H_c_SA(i) = R_sos_SA(i)+R_sis_SA(i)+W_c_SA(i)+W_s_SA(i);
H_c2_SA(i) = H_c_SA(i)/1000;

% Holding Capacity Square Caisson
R_sos_SQ_SA(i) = ((y_komma*(h_depth_SA(i)-H_split1))/2)*Ktandelta*...
    (4*C_av_innerskirts)*(h_depth_SA(i)-H_split1);
R_sis_SQ_SA(i) = R_sos_SQ_SA(i);
W_c_SQ_SA(i) = (4*WeightForce_total)/9;
W_s_SQ_SA(i) = (A_SQ_outer)*(h_depth_SA(i)-H_split1)*(y_komma);

H_c_SQ_SA(i) = R_sos_SQ_SA(i)+R_sis_SQ_SA(i)+W_c_SQ_SA(i)+W_s_SQ_SA(i);
H_c2_SQ_SA(i) = H_c_SQ_SA(i)/1000;

% Holding Capacity Trapezoidal Caisson
R_sos_trap_SA(i) = ((y_komma*(h_depth_SA(i)-H_split2))/2)*Ktandelta*...
    (C_av_trap)*(h_depth_SA(i)-H_split2);
R_sis_trap_SA(i) = R_sos_trap_SA(i);
W_c_trap_SA(i) = (4*WeightForce_total)/9;
W_s_trap_SA(i) = (A_inner_trapC)*(h_depth_SA(i)-H_split2)*(y_komma);

H_c_trap_SA(i) = R_sos_trap_SA(i)+R_sis_trap_SA(i)+W_c_trap_SA(i)+...
    W_s_trap_SA(i);
H_c2_trap_SA(i) = H_c_trap_SA(i)/1000;
end

elseif (H_seal1 <= h_Self_Weight) && (h_Self_Weight < H_seal2)
% The cylindrical and square caissons have sufficient penetration depth and seal
% After the self-weight phase

%Determine depth arrays for the installation steps
H_SAS = length([h_Self_Weight:0.001:H_split2]);

% For loop also including square chambers
for i = 1:1:H_SAS
% Initializing for the flow factors
a1_SA(i) = c_0 - c_1*(1-(exp((-1)*(h_depth_SA(i)/(c_2*D_av)))));
a_SA(i)  = (a1_SA(i)*k_f)/((1-a1_SA(i))+a1_SA(i)*k_f);
a2_SA = 0.5;

% Calculating the critical suction
s_crit(i) = (y_komma*h_depth_SA(i))/(1-a_SA(i));
% Determining the multiple of the area
Am_SQ_outer = m_SA*A_SQ_outer;

Z_inner_SQ = (A_SQ_inner/(Ktandelta*(4*C_i_innerskirts)));
Z_outer_SQ = ((Am_SQ_outer-A_SQ_outer)/(Ktandelta*(4*C_o_innerskirts)));

% Setting the factors for the round caisson
V_x1(i) = ((Z_outer_SA)^2)*((exp(h_depth_SA(i)/Z_outer_SA))-1-...
    (h_depth_SA(i)/Z_outer_SA))*(Ktandelta)*(pi*D_outer);
V_x2(i) = ((Z_inner_SA)^2)*((exp(h_depth_SA(i)/Z_inner_SA))-1-...
    (h_depth_SA(i)/Z_inner_SA))*(Ktandelta)*(pi*D_inner);
V_x3(i) = (Z_inner_SA*((exp(h_depth_SA(i)/Z_inner_SA))-1)*Nq);
V_x4(i) = (y_komma*t*Ny);
V_x5(i) = (pi*D_av*t*cof_tip);
V_x6(i) = ((pi*(D_inner^2))/4);

% Same factors as for the round suction caisson, but
% determined for the square one
V_x7(i) = ((Z_outer_SQ)^2)*((exp((h_depth_SA(i)-H_split1)/Z_outer_SQ))...
    -1-((h_depth_SA(i)-H_split1)/Z_outer_SQ))*(Ktandelta)*...
    (4*(C_o_innerskirts));
V_x8(i) = ((Z_inner_SQ)^2)*((exp((h_depth_SA(i)-H_split1)/Z_inner_SQ))...
    -1-((h_depth_SA(i)-H_split1)/Z_inner_SQ))*(Ktandelta)*...
    (4*(C_i_innerskirts));
V_x9(i) = (Z_inner_SQ*((exp((h_depth_SA(i)-H_split1)/Z_inner_SQ))-1)*Nq);
V_x10(i) = (y_komma*t_plates*Ny);
V_x11(i) = (A_SQ_outer-A_SQ_inner);
V_x12(i) = A_SQ_inner;

V_outer_plates(i) = 0;
V_inner_plates(i) = 0;
V_tip_plates(i)   = 0;

% Calculating the new variable s [Pa], based on the full system
% In this equation, the additional terms for the square
% caisson are considered
syms p2
eqn = (p2*V_x12(i)) + 4*(p2*(V_x6(i))) + 4*(WeightForce_total) == 4*...
    ((y_komma + ((a_SA(i)*p2)/h_depth_SA(i)))*V_x1(i) + (y_komma - ...
    (((1-a_SA(i))*p2)/h_depth_SA(i)))*V_x2(i) ...
    + ((((y_komma - (((1-a_SA(i))*p2)/h_depth_SA(i)))*V_x3(i))+V_x4(i))...
    *V_x5(i))) + V_outer_plates(i) + ...
    ((y_komma + ((a2_SA*p2)/(h_depth_SA(i)-H_split1)))*V_x7(i) + ...
    (y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-H_split1)))*V_x8(i) + ...
    ((((y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-H_split1)))*V_x9(i))...
    +V_x10(i))*V_x11(i)))+ V_tip_plates(i) + V_inner_plates(i)...
    + 4*(Fbuoy_total);

s2(i) = double (solve(eqn,p2));

V_Force_SA(i)   = WeightForce_total + s2(i)*(V_x6(i));
V_Q_outer_SA(i) = (y_komma + ((a_SA(i)*(s2(i)))/h_depth_SA(i)))*V_x1(i);
V_Q_inner_SA(i) = (y_komma - (((1-a_SA(i))*(s2(i)))/h_depth_SA(i)))*V_x2(i);
V_Q_tip_SA(i)   = ((((y_komma - (((1-a_SA(i))*(s2(i)))/h_depth_SA(i)))...
    *V_x3(i))+V_x4(i))*V_x5(i));

% Calculating the terms for the square suction caisson
V_Q_outer_SA_SQ(i) = (y_komma + ((a2_SA*(s2(i)))/(h_depth_SA(i)...
    )))*V_x7(i);
V_Q_inner_SA_SQ(i) = (y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x8(i);
V_Q_tip_SA_SQ(i)   = ((((y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x9(i))+V_x10(i))*V_x11(i));

% Calculating the total frictional forces
V_Qtot_SA(i)   = V_Q_outer_SA(i) + V_Q_inner_SA(i) + V_Q_tip_SA(i)...
   + V_outer_plates(i) + V_Q_outer_SA_SQ(i) + V_Q_inner_SA_SQ(i) +...
    V_Q_tip_SA_SQ(i) + V_inner_plates(i) +V_tip_plates(i)+ 4*Fbuoy_total;

% Holding Capacity Cylindrical Caisson
R_sos_SA(i) = ((y_komma*h_depth_SA(i))/2)*Ktandelta*pi*D_av*h_depth_SA(i);
R_sis_SA(i) = R_sos_SA(i);
W_c_SA(i) = (4*WeightForce_total)/5;
W_s_SA(i) = (((pi*(D_outer)^2)/4)*h_depth_SA(i))*(y_komma);

H_c_SA(i) = R_sos_SA(i)+R_sis_SA(i)+W_c_SA(i)+W_s_SA(i);
H_c2_SA(i) = H_c_SA(i)/1000;

% Holding Capacity Square Chamber
```

```
R_sos_SQ_SA(i) = ((y_komma*(h_depth_SA(i)-H_split1))/2)*Ktandelta*...
    (4*C_av_innerskirts)*(h_depth_SA(i)-H_split1);
R_sis_SQ_SA(i) = R_sos_SQ_SA(i);
W_c_SQ_SA(i) = (4*WeightForce_total)/5;
W_s_SQ_SA(i) = (A_SQ_outer)*(h_depth_SA(i)-H_split1)*(y_komma);

H_c_SQ_SA(i) = R_sos_SQ_SA(i)+R_sis_SQ_SA(i)+W_c_SQ_SA(i)+W_s_SQ_SA(i);
H_c2_SQ_SA(i) = H_c_SQ_SA(i)/1000;

% Holding Capacity Square Chamber
H_c2_trap_SA(i) = 0;
end

% For loop also inclding square and trapezoidal chambers
for i = H_SAS+1:1:length(h_depth_SA)
% Initializing for the flow factors
a1_SA(i) = c_0 - c_1*(1-(exp((-1)*(h_depth_SA(i)/(c_2*D_av)))));
a_SA(i)  = (a1_SA(i)*k_f)/((1-a1_SA(i))+a1_SA(i)*k_f);
a2_SA = 0.5;

% Calculating the critical suction
s_crit(i) = (y_komma*h_depth_SA(i))/(1-a_SA(i));
% Determining the multiple of the area
Am_SQ_outer = m_SA*A_SQ_outer;
Am_trap_outer = m_SA*A_outer_trapC;

Z_inner_SQ = (A_SQ_inner/(Ktandelta*(4*C_i_innerskirts)));
Z_outer_SQ = ((Am_SQ_outer-A_SQ_outer)/(Ktandelta*(4*C_o_innerskirts)));

Z_inner_trap = (A_inner_trapC/(Ktandelta*C_inner_trap));
Z_outer_trap = ((Am_trap_outer-A_outer_trapC)/(Ktandelta*C_outer_trap));


% Setting the factors for the round caisson
V_x1(i) = ((Z_outer_SA)^2)*((exp(h_depth_SA(i)/Z_outer_SA))-1-...
    (h_depth_SA(i)/Z_outer_SA))*(Ktandelta)*(pi*D_outer);
V_x2(i) = ((Z_inner_SA)^2)*((exp(h_depth_SA(i)/Z_inner_SA))-1-...
    (h_depth_SA(i)/Z_inner_SA))*(Ktandelta)*(pi*D_inner);
V_x3(i) = (Z_inner_SA*((exp(h_depth_SA(i)/Z_inner_SA))-1)*Nq);
V_x4(i) = (y_komma*t*Ny);
V_x5(i) = (pi*D_av*t*cof_tip);
V_x6(i) = ((pi*(D_inner^2))/4);

% Same factors as for the round suction caisson, but
% determined for the square one
V_x7(i) = ((Z_outer_SQ)^2)*((exp((h_depth_SA(i)-H_split1)/Z_outer_SQ))...
    -1-((h_depth_SA(i)-H_split1)/Z_outer_SQ))*(Ktandelta)*...
    (4*(C_o_innerskirts));
V_x8(i) = ((Z_inner_SQ)^2)*((exp((h_depth_SA(i)-H_split1)/Z_inner_SQ))...
    -1-((h_depth_SA(i)-H_split1)/Z_inner_SQ))*(Ktandelta)*...
    (4*(C_i_innerskirts));
V_x9(i) = (Z_inner_SQ*((exp((h_depth_SA(i)-H_split1)/Z_inner_SQ))-1)*Nq);
V_x10(i) = (y_komma*t_plates*Ny);
V_x11(i) = (A_SQ_outer-A_SQ_inner);
V_x12(i) = A_SQ_inner;

% Same factors as for the round caisson, but for the trapezoidal chambers
V_x13(i) = ((Z_outer_trap)^2)*((exp((h_depth_SA(i)-H_split2)/Z_outer_trap))...
    -1-((h_depth_SA(i)-H_split2)/Z_outer_trap))*(Ktandelta)*C_outer_trap;
V_x14(i) = ((Z_inner_trap)^2)*((exp((h_depth_SA(i)-H_split2)/Z_inner_trap))...
    -1-((h_depth_SA(i)-H_split2)/Z_inner_trap))*(Ktandelta)*C_inner_trap;
V_x15(i) = (Z_inner_trap*((exp((h_depth_SA(i)-H_split2)/Z_inner_trap))-1)*Nq);
V_x16(i) = (y_komma*t_plates*Ny);
V_x17(i) = (A_outer_trapC-A_inner_trapC);
V_x18(i) = A_inner_trapC;

V_outer_plates(i) = 0;
V_inner_plates(i) = 0;
V_tip_plates(i)   = 0;
```

```
% Calculating the new variable s [Pa], based on the full system
% In this equation, the additional terms for the square and
% trapezoidal caisson are considered

syms p2
eqn = 4*(p2*V_x18(i)) +(p2*V_x12(i)) + 4*(p2*(V_x6(i))) + 4*...
    (WeightForce_total) == 4*((y_komma + ((a_SA(i)*p2)/h_depth_SA(i)))...
    *V_x1(i) + (y_komma - (((1-a_SA(i))*p2)/h_depth_SA(i)))*V_x2(i) ...
    + ((((y_komma - (((1-a_SA(i))*p2)/h_depth_SA(i)))*V_x3(i))+V_x4(i))...
    *V_x5(i))) +((y_komma + ((a2_SA*p2)/(h_depth_SA(i)-H_split1)))*V_x7(i)...
    + (y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-H_split1)))*V_x8(i) + ...
    ((((y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-H_split1)))*V_x9(i))+...
    V_x10(i))*V_x11(i)))+ (4*((y_komma + ((a2_SA*p2)/(h_depth_SA(i)...
    -H_split2)))*V_x13(i) + (y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-...
    H_split2)))*V_x14(i) +((((y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)...
    -H_split2)))*V_x15(i))+V_x16(i))*V_x17(i)))) + 4*(Fbuoy_total);

s2(i) = double (solve(eqn,p2));

V_Force_SA(i)   = WeightForce_total + s2(i)*(V_x6(i));
V_Q_outer_SA(i) = (y_komma + ((a_SA(i)*(s2(i)))/h_depth_SA(i)))*V_x1(i);
V_Q_inner_SA(i) = (y_komma - (((1-a_SA(i))*(s2(i)))/h_depth_SA(i)))*V_x2(i);
V_Q_tip_SA(i)   = ((((y_komma - (((1-a_SA(i))*(s2(i)))/h_depth_SA(i)))...
    *V_x3(i))+V_x4(i))*V_x5(i));

% Calculating the terms for the square suction caisson
V_Q_outer_SA_SQ(i) = (y_komma + ((a2_SA*(s2(i)))/(h_depth_SA(i)...
    )))*V_x7(i);
V_Q_inner_SA_SQ(i) = (y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x8(i);
V_Q_tip_SA_SQ(i)   = ((((y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x9(i))+V_x10(i))*V_x11(i));

% Calculating the terms for the trapezoidal suction caissons
V_Q_outer_SA_trap(i) = (y_komma + ((a2_SA*(s2(i)))/(h_depth_SA(i)...
    )))*V_x13(i);
V_Q_inner_SA_trap(i) = (y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x14(i);
V_Q_tip_SA_trap(i)   = ((((y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    )))*V_x15(i))+V_x16(i))*V_x17(i));

% Calculating the total frictional forces
V_Qtot_SA(i)   = V_Q_outer_SA(i) + V_Q_inner_SA(i) + V_Q_tip_SA(i) +...
    V_Q_outer_SA_trap(i) +V_Q_inner_SA_trap(i) + V_Q_tip_SA_trap(i)+...
    V_Q_outer_SA_SQ(i) + V_Q_inner_SA_SQ(i) + V_Q_tip_SA_SQ(i)...
    + 4*Fbuoy_total;

% Holding Capacity Cylindrical Caisson
R_sos_SA(i) = ((y_komma*h_depth_SA(i))/2)*Ktandelta*pi*D_av*h_depth_SA(i);
R_sis_SA(i) = R_sos_SA(i);
W_c_SA(i) = (4*WeightForce_total)/9;
W_s_SA(i) = (((pi*(D_outer)^2)/4)*h_depth_SA(i))*(y_komma);

H_c_SA(i) = R_sos_SA(i)+R_sis_SA(i)+W_c_SA(i)+W_s_SA(i);
H_c2_SA(i) = H_c_SA(i)/1000;

% Holding Capacity Square Caisson
R_sos_SQ_SA(i) = ((y_komma*(h_depth_SA(i)-H_split1))/2)*Ktandelta*...
    (4*C_av_innerskirts)*(h_depth_SA(i)-H_split1);
R_sis_SQ_SA(i) = R_sos_SQ_SA(i);
W_c_SQ_SA(i) = (4*WeightForce_total)/9;
W_s_SQ_SA(i) = (A_SQ_outer)*(h_depth_SA(i)-H_split1)*(y_komma);

H_c_SQ_SA(i) = R_sos_SQ_SA(i)+R_sis_SQ_SA(i)+W_c_SQ_SA(i)+W_s_SQ_SA(i);
H_c2_SQ_SA(i) = H_c_SQ_SA(i)/1000;

% Holding Capacity Trapezoidal Caisson
R_sos_trap_SA(i) = ((y_komma*(h_depth_SA(i)-H_split2))/2)*Ktandelta*...
    (C_av_trap)*(h_depth_SA(i)-H_split2);
```

```
R_sis_trap_SA(i) = R_sos_trap_SA(i);
W_c_trap_SA(i) = (4*WeightForce_total)/9;
W_s_trap_SA(i) = (A_inner_trapC)*(h_depth_SA(i)-H_split2)*(y_komma);

H_c_trap_SA(i) = R_sos_trap_SA(i)+R_sis_trap_SA(i)+W_c_trap_SA(i)+...
    W_s_trap_SA(i);
H_c2_trap_SA(i) = H_c_trap_SA(i)/1000;
end

else
% All chambers will be activated at once when all plates have a sufficient
% seal after the self-weight penetration phase

for i = 1:1:length(h_depth_SA)
% Initializing for the flow factors
a1_SA(i) = c_0 - c_1*(1-(exp((-1)*(h_depth_SA(i)/(c_2*D_av)))));
a_SA(i)  = (a1_SA(i)*k_f)/((1-a1_SA(i))+a1_SA(i)*k_f);
a2_SA = 0.5;

% Calculating the critical suction
s_crit(i) = (y_komma*h_depth_SA(i))/(1-a_SA(i));
% Determining the multiple of the area
Am_SQ_outer = m_SA*A_SQ_outer;
Am_trap_outer = m_SA*A_outer_trapC;

Z_inner_SQ = (A_SQ_inner/(Ktandelta*(4*C_i_innerskirts)));
Z_outer_SQ = ((Am_SQ_outer-A_SQ_outer)/(Ktandelta*(4*C_o_innerskirts)));

Z_inner_trap = (A_inner_trapC/(Ktandelta*C_inner_trap));
Z_outer_trap = ((Am_trap_outer-A_outer_trapC)/(Ktandelta*C_outer_trap));


% Setting the factors for the round caisson
V_x1(i) = ((Z_outer_SA)^2)*((exp(h_depth_SA(i)/Z_outer_SA))-1-...
    (h_depth_SA(i)/Z_outer_SA))*(Ktandelta)*(pi*D_outer);
V_x2(i) = ((Z_inner_SA)^2)*((exp(h_depth_SA(i)/Z_inner_SA))-1-...
    (h_depth_SA(i)/Z_inner_SA))*(Ktandelta)*(pi*D_inner);
V_x3(i) = (Z_inner_SA*((exp(h_depth_SA(i)/Z_inner_SA))-1)*Nq);
V_x4(i) = (y_komma*t*Ny);
V_x5(i) = (pi*D_av*t*cof_tip);
V_x6(i) = ((pi*(D_inner^2))/4);

% Same factors as for the round suction caisson, but
% determined for the square one
V_x7(i) = ((Z_outer_SQ)^2)*((exp((h_depth_SA(i)-H_split1)/Z_outer_SQ))...
    -1-((h_depth_SA(i)-H_split1)/Z_outer_SQ))*(Ktandelta)*...
    (4*(C_o_innerskirts));
V_x8(i) = ((Z_inner_SQ)^2)*((exp((h_depth_SA(i)-H_split1)/Z_inner_SQ))...
    -1-((h_depth_SA(i)-H_split1)/Z_inner_SQ))*(Ktandelta)*...
    (4*(C_i_innerskirts));
V_x9(i) = (Z_inner_SQ*((exp((h_depth_SA(i)-H_split1)/Z_inner_SQ))-1)*Nq);
V_x10(i) = (y_komma*t_plates*Ny);
V_x11(i) = (A_SQ_outer-A_SQ_inner);
V_x12(i) = A_SQ_inner;

% Same factors as for the round caisson, but for the trapezoidal chambers
V_x13(i) = ((Z_outer_trap)^2)*((exp((h_depth_SA(i)-H_split2)/Z_outer_trap))...
    -1-((h_depth_SA(i)-H_split2)/Z_outer_trap))*(Ktandelta)*C_outer_trap;
V_x14(i) = ((Z_inner_trap)^2)*((exp((h_depth_SA(i)-H_split2)/Z_inner_trap))...
    -1-((h_depth_SA(i)-H_split2)/Z_inner_trap))*(Ktandelta)*C_inner_trap;
V_x15(i) = (Z_inner_trap*((exp((h_depth_SA(i)-H_split2)/Z_inner_trap))-1)*Nq);
V_x16(i) = (y_komma*t_plates*Ny);
V_x17(i) = (A_outer_trapC-A_inner_trapC);
V_x18(i) = A_inner_trapC;

V_outer_plates(i) = 0;
V_inner_plates(i) = 0;
V_tip_plates(i)   = 0;

% Calculating the new variable s [Pa], based on the full system
% In this equation, the additional terms for the square and
% trapezoidal caisson are considered

syms p2
eqn = 4*(p2*V_x18(i)) +(p2*V_x12(i)) + 4*(p2*(V_x6(i))) + 4*...
    (WeightForce_total) == 4*((y_komma + ((a_SA(i)*p2)/h_depth_SA(i)))...
    *V_x1(i) + (y_komma - (((1-a_SA(i))*p2)/h_depth_SA(i)))*V_x2(i) ...
    + ((((y_komma - (((1-a_SA(i))*p2)/h_depth_SA(i)))*V_x3(i))+V_x4(i))...
    *V_x5(i))) +((y_komma + ((a2_SA*p2)/(h_depth_SA(i)-H_split1)))*V_x7(i)...
    + (y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-H_split1)))*V_x8(i) + ...
    ((((y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-H_split1)))*V_x9(i))+...
    V_x10(i))*V_x11(i)))+ (4*((y_komma + ((a2_SA*p2)/(h_depth_SA(i)...
    -H_split2)))*V_x13(i) + (y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)-...
    H_split2)))*V_x14(i) +((((y_komma - (((1-a2_SA)*p2)/(h_depth_SA(i)...
    -H_split2)))*V_x15(i))+V_x16(i))*V_x17(i)))) + 4*(Fbuoy_total);

s2(i) = double (solve(eqn,p2));

V_Force_SA(i)   = WeightForce_total + s2(i)*(V_x6(i));
V_Q_outer_SA(i) = (y_komma + ((a_SA(i)*(s2(i)))/h_depth_SA(i)))*V_x1(i);
V_Q_inner_SA(i) = (y_komma - (((1-a_SA(i))*(s2(i)))/h_depth_SA(i)))*V_x2(i);
V_Q_tip_SA(i)   = ((((y_komma - (((1-a_SA(i))*(s2(i)))/h_depth_SA(i)))...
    *V_x3(i))+V_x4(i))*V_x5(i));

% Calculating the terms for the square suction caisson
V_Q_outer_SA_SQ(i) = (y_komma + ((a2_SA*(s2(i)))/(h_depth_SA(i)...
    -H_split1)))*V_x7(i);
V_Q_inner_SA_SQ(i) = (y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    -H_split1)))*V_x8(i);
V_Q_tip_SA_SQ(i)   = ((((y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    -H_split1)))*V_x9(i))+V_x10(i))*V_x11(i));

% Calculating the terms for the trapezoidal suction caissons
V_Q_outer_SA_trap(i) = (y_komma + ((a2_SA*(s2(i)))/(h_depth_SA(i)-...
    H_split2)))*V_x13(i);
V_Q_inner_SA_trap(i) = (y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)-...
    H_split2)))*V_x14(i);
V_Q_tip_SA_trap(i)   = ((((y_komma - (((1-a2_SA)*(s2(i)))/(h_depth_SA(i)...
    -H_split2)))*V_x15(i))+V_x16(i))*V_x17(i));

% Calculating the total frictional forces
V_Qtot_SA(i)   = V_Q_outer_SA(i) + V_Q_inner_SA(i) + V_Q_tip_SA(i) +...
    V_Q_outer_SA_trap(i) +V_Q_inner_SA_trap(i) + V_Q_tip_SA_trap(i)+...
    V_Q_outer_SA_SQ(i) + V_Q_inner_SA_SQ(i) + V_Q_tip_SA_SQ(i)...
    + 4*Fbuoy_total;

% Holding Capacity Cylindrical Caisson
R_sos_SA(i) = ((y_komma*h_depth_SA(i))/2)*Ktandelta*pi*D_av*h_depth_SA(i);
R_sis_SA(i) = R_sos_SA(i);
W_c_SA(i) = (4*WeightForce_total)/9;
W_s_SA(i) = (((pi*(D_outer)^2)/4)*h_depth_SA(i))*(y_komma);

H_c_SA(i) = R_sos_SA(i)+R_sis_SA(i)+W_c_SA(i)+W_s_SA(i);
H_c2_SA(i) = H_c_SA(i)/1000;

% Holding Capacity Square Caisson
R_sos_SQ_SA(i) = ((y_komma*(h_depth_SA(i)-H_split1))/2)*Ktandelta*...
    (4*C_av_innerskirts)*(h_depth_SA(i)-H_split1);
R_sis_SQ_SA(i) = R_sos_SQ_SA(i);
W_c_SQ_SA(i) = (4*WeightForce_total)/9;
W_s_SQ_SA(i) = (A_SQ_outer)*(h_depth_SA(i)-H_split1)*(y_komma);

H_c_SQ_SA(i) = R_sos_SQ_SA(i)+R_sis_SQ_SA(i)+W_c_SQ_SA(i)+W_s_SQ_SA(i);
H_c2_SQ_SA(i) = H_c_SQ_SA(i)/1000;

% Holding Capacity Trapezoidal Caisson
R_sos_trap_SA(i) = ((y_komma*(h_depth_SA(i)-H_split2))/2)*Ktandelta*...
    (C_av_trap)*(h_depth_SA(i)-H_split2);
R_sis_trap_SA(i) = R_sos_trap_SA(i);
```

```
W_c_trap_SA(i) = (4*WeightForce_total)/9;
W_s_trap_SA(i) = (A_inner_trapC)*(h_depth_SA(i)-H_split2)*(y_komma);

H_c_trap_SA(i) = R_sos_trap_SA(i)+R_sis_trap_SA(i)+W_c_trap_SA(i)+...
    W_s_trap_SA(i);
H_c2_trap_SA(i) = H_c_trap_SA(i)/1000;
end

end

%% Plotting the behaviour of the foundation system

%------------------------------------------------------------------------
% Plotting the behaviour of the foundation system for individual phases
% (Cohesionless case)
%------------------------------------------------------------------------

figure(1)
plot(h_depth, V_Qtot_SW)
title('Total resistance force during the Self-weight penetration phase')
xlabel('Penetration depth [m]')
ylabel('Total resistance force [N]')

figure(2)
plot(h_depth_SA,s2)
title('Suction required for a total depth of 0.5 m')
xlabel('Penetration depth [m]')
ylabel('Required suction [Pa]')

figure(3)
plot(h_depth_SA,V_Qtot_SA)
title('Total resistance forces during the suction assisted phase')
xlabel('Penetration depth [m]')
ylabel('Total resistance force [N]')

figure(4)
plot(h_depth_SA,V_Q_tip_SA)
title('Resistance force at the tips of the caisson during the suction assisted
phase')
xlabel('Penetration depth [m]')
ylabel('Resistance force at the tips [N]')

figure(5)
plot(h_depth_SA,V_Q_inner_SA)
title('Resistance force at the inside of the caisson during the suction assisted
phase')
xlabel('Penetration depth [m]')
ylabel('Resistance force at the inside of the caisson [N]')

figure(6)
plot(h_depth_SA,V_Q_outer_SA)
title('Resistance force at the outside of the caisson during the suction assisted
phase')
xlabel('Penetration depth [m]')
ylabel('Resistance force at the outside of the caisson [N]')

figure(7)
plot(h_depth_SA, V_Q_outer_SA)
hold on
plot(h_depth_SA, V_Q_inner_SA)
hold on
plot(h_depth_SA, V_Q_tip_SA)
hold off

title('Resistance forces during the suction assisted penetration phase')
xlabel('Penetration depth [m]')
ylabel('Resistance force [N]')
legend('Outer resistance force', 'Inner resistance force',...
    'Tip resistance force', 'Location', 'Southeast')
```

```
%-----------------------------------------------------------------------
% Plotting the behaviour of the foundation system for all phases
% (Cohesionless case)
%-----------------------------------------------------------------------

h_tot  = [h_depth h_depth_SA];
V_Q_outer_tot = [V_Q_outer_SW V_Q_outer_SA];
V_Q_inner_tot = [V_Q_inner_SW V_Q_inner_SA];
V_Q_tip_tot = [V_Q_tip_SW V_Q_tip_SA];
V_Qtot = [V_Qtot_SW  V_Qtot_SA];
H_c2 = [H_c2_SW H_c2_SA];
H_c2_SQ = [H_c2_SQ_SW H_c2_SQ_SA];
H_c2_trap = [H_c2_trap_SW H_c2_trap_SA];

figure(8)
plot(h_tot,V_Qtot)
title('Total resistance force during the installation')
xlabel('Penetration depth [m]')
ylabel('Total resistance force [N]')

figure(9)
plot(h_tot, V_Q_outer_tot)
hold on
plot(h_tot, V_Q_inner_tot)
hold on
plot(h_tot, V_Q_tip_tot)
hold off

title('Resistance forces for the caisson over the penetration depth')
xlabel('Penetration depth [m]')
ylabel('Resistance force [N]')
legend('Outer resistance force', 'Inner resistance force',...
    'Tip resistance force', 'Location', 'Northwest')

figure(10)
plot(h_depth_SA,a_SA)
title('Flow factor "a" over the penetration depth')
xlabel('Penetration depth [m]')
ylabel('Flow factor a [-]')

% critical suction
figure(11)
plot(h_depth_SA,s2)
hold on
plot(h_depth_SA,s_crit)
hold off
title('Calculated suction vs Critical suction ')
xlabel('Penetration depth [m]')
ylabel('Required suction [Pa]')
legend('Suction Houlsby & Byrne','Critcal Suction', 'Location', 'Northwest')

figure(12)
plot(h_tot,H_c2)
hold on
plot(h_tot,H_c2_SQ)
hold on
plot(h_tot,H_c2_trap)
hold off
title('Holding capacity of one suction caisson')
xlabel('Penetration depth [m]')
ylabel('Holding capacity [kN]')
legend('Circular Caissons','Square Caisson', 'Trapezoidal Caisson' ,...
     'Location', 'Northwest')
```

11.2 Appendix A: MATLAB code cohesive soils

```
%% Master Thesis %% Ocean Grazer's second prototype %%
% Authors: Lennard Hut (S2718960) and Leonard Vos (S3209016)
% Year: 2020

% About this script: simulates the installation behaviour of a suction
% anchor for cohesive soils. Firstly, the initial penetration depth is
% determined and then the required suction is determined for the suction
% assisted phase. Critical suction is also determined to see whether the
% system can install.
% The design is fully parametrized and when the model is run all dimensions
% and soil type can be filled in and selected. The values of the current
% design are entered in advance but caan easily be altered.

% Case: Full functioning model for predicting penetration behaviour.

close all
clear all
clc

%% Designing the prompts %%
% These prompts will request different parameters of the user, which will
% be coupled to their appropriate matlab parameters.

%----------------------------------------------------------------------
% Save the values in their own variable name for the Caisson
%----------------------------------------------------------------------

% Calling a prompt for the design parameters of the caissons
prompt = {'Inner diameter of the Suction Caisson [m] :', ...
    'Outer diameter of the Suction Caisson [m]:',...
    'Inner height of the Suction Caisson [m]:', ...
    'Outer height of the Suction Caisson [m]:'};
dlg_title = 'Design Parameters Suction Caisson';
num_lines = 1;
defaultans = {'0.498','0.508','0.50','0.51'};
answer = inputdlg(prompt,dlg_title,[1 70],defaultans);


for i = 1:(size(answer))
  data(i) = str2num(answer{i});
end

D_inner = data(1);          % Inner diameter of the caisson [m]
D_outer = data(2);          % Outer diameter of the caisson [m]
H_inner = data(3);          % Inner height of the caisson [m]
H_outer = data(4);          % Outer height of the caisson [m]

%----------------------------------------------------------------------
% Save the values in their own variable name for the skirts
%----------------------------------------------------------------------

% Calling a second prompt for the design parameters of the skirts
prompt2 = {'Length of the Caisson Connecting Skirts [m]:', ...
    'Length of the Middle Skirts [m]:', ...
    'Length of the Diagonal Connecting Skirts [m]:',...
    'Height of the Square Skirt [m]:',...
    'Height of the Trapezoidal Skirt [m]:',...
    'Thickness of the Skirt [m]:', ...
    'Length of the Platform Plate [m]:','Width of the Platform Plate [m]:',...
    'Thickness of the Platform Plate [m]:'};
dlg_title2 = 'Design Parameters Skirts';
num_lines = 1;
defaultans2 = {'1.28','0.61','0.48','0.21','0.21','0.005','1.998',...
    '1.998','0.01'};
answer2 = inputdlg(prompt2,dlg_title2,[1 80],defaultans2);
```

```
for i = 1:(size(answer2))
    data2(i) = str2num(answer2{i});
end

L_plate_outerskirts         = data2(1);     % Length of the outerskirts [m]
L_plate_innerskirts         = data2(2);     % Length of the inner skirts [m]
L_plate_diagonalskirts      = data2(3);     % Length of the diagonal skirts [m]
H_square                    = data2(4);     % Height of the square skirts [m]
H_trap                      = data2(5);     % Height of the trapezoidal skirts [m]
t_plates                    = data2(6);     % Thickness of the skirts [m]
L_plate_platform            = data2(7);     % Length of the platform [m]
W_plate_platform            = data2(8);     % Width of the platform [m]
t_plate_platform            = data2(9);     % Thickness of the platform [m]

%------------------------------------------------------------------------
% Save the values in their own variable name for the remaining
% parameters
%------------------------------------------------------------------------

% Calling a third promt for other needed parameters
prompt3 = {'Density of the Foundation Material [kg/m^3]:', ...
    'Density of the Sea Water [kg/m^3]', ...
    'Total Weight of the Complete Structure [kg]',};
dlg_title3 = 'Additional Parameters';
num_lines = 1;
defaultans3 = {'7850','1020','1089.9'};
answer3 = inputdlg(prompt3,dlg_title3,[1 80],defaultans3);

for i = 1:(size(answer3))
    data3(i) = str2num(answer3{i});
end

rho_mat = data3(1);                 % Density of the foundation [kg/m^3]
rho_wat = data3(2);                 % Density of the water [kg/m^3]
Total_system_Weight=data3(3);       % Weight of the Ocean Battery [kg]

%------------------------------------------------------------------------
% Save the values in their own variable name for the remaining
% parameters
%------------------------------------------------------------------------

list = {'Kaolin Clay','Nkossa Clay', 'Qiantang River Silt'};
[indx,tf] = listdlg('PromptString',{'Select a soil type.',...
    'Only select one of the soils.'},'SelectionMode','single',...
    'ListString', list, 'ListSize', [300, 300], 'Name', ...
    'Select a soil type');

if indx == 1    % For Kaolin Clay
        y_komma         = 6500;     % Effective soil weight [kN/m^3]
        Ktandelta       = 0.8;      % [-]
        phi             = 26;       % Angle of friction [degree]
        alpha           = 0.5;      % Adhesion factor [-]
        s_u0            = 4000;     % Mudline shear strength [Pa]
        rate_change     = 1500;     % Shear strength rate [Pa]

elseif indx == 2    % For Nkossa Clay
        y_komma         = 6000;     % Effective soil weight [kN/m^3]
        Ktandelta       = 0.8;      % [-]
        phi             = 19;       % Angle of friction [degree]
        alpha           = 0.45;     % Adhesion factor [-]
        s_u0            = 5000;     % Mudline shear strength [Pa]
        rate_change     = 1670;     % Shear strength rate [Pa]

elseif indx == 3    % For Qiantang River Silt
        y_komma         = 8823;     % Effective soil weight [kN/m3]
        Ktandelta       = 0.8;      % [-]
        phi             = 36.5;     % Angle of friction [degree]
        alpha           = 0.5;      % Adhesion factor [-]
        s_u0            = 6000;     % Mudline shear strength [Pa]
        rate_change     = 1350;     % Shear strength rate [Pa]
else
end

% Calculating the bearing capacity factors
% Bearing capacity factor overburden [-] (Houlsby and Byrne)
Nq = (exp(2*pi*(0.75-(phi/360))*tand(phi)))/...
    (2*(cosd(45+(phi/2))^2));
% Bearing capacity factor self-weight [-] (Houlsby and Byrne)
Ny = (2*(Nq+1)*tand(phi))/(1+0.4*sind(phi));
% Bearing capacity factor cohesion [-]
Nc = (Nq-1)/tand(phi);

%% Calculating different basis parameters
gravity     = 9.81;     % Gravity acceleration constant [m/s^2]
m_SW        = 1.5;      % Multiple of the diameter for enhanced stress [-]

%------------------------------------------------------------------------
% Calculation of parameters of the suction caissons
%------------------------------------------------------------------------

% Thickness of the suction caisson walls [m]
t              = ((D_outer-D_inner)/2);
% Average Diameter of the suction caisson [m]
D_av           = ((D_outer+D_inner)/2);
% Height difference between plates and caisson
H_split1        = H_outer - H_square;
H_split2        = H_outer - H_trap;
% Volume calculation of the suction caisson [m^3]
Vol_SC         = (pi*((D_outer/2)^2)*H_outer)-(pi*((D_inner/2)^2)*H_inner);
% Weight calculation of the suction caisson [kg]
Weight_SC      = Vol_SC*rho_mat;
% Vertical load due to the weight of the suction caisson [N]
WeightForce_SC = gravity*Weight_SC;
% Buoyancy force of the submerged suction caisson [N]
Fbuoy_SC       = Vol_SC*rho_wat*gravity;

%------------------------------------------------------------------------
% Calculations for tip angle
%------------------------------------------------------------------------

% Tip angle coefficient for the cylindrical caissons
cof_tip        = 1;

%------------------------------------------------------------------------
% Calculating the circumferences of the plates
%------------------------------------------------------------------------

% The plates connecting the caissons
% Calculating the outer circumference of the plates between the caissons [m]
C_o_outerskirts  = 0.25*((2*L_plate_outerskirts)+2*(L_plate_outerskirts...
    +(2*t_plates)));
% Caclulating the inner circumference of the plates between the caissons [m]
C_i_outerskirts  = 0.25*((2*L_plate_outerskirts)+2*(L_plate_outerskirts...
    -(2*t_plates)));
% Calculating the average circumference of the plates between the caissons [m]
C_av_outerskirts = 0.25*((C_o_outerskirts+C_i_outerskirts)/2);

% The plates that form a square in the middle
% Calculating the outer circumference of the plates in the middle [m]
C_o_innerskirts  = 0.25*(2*L_plate_innerskirts + 2*(L_plate_innerskirts...
     +(2*t_plates)));
% Calculating the inner circumference of the plates in the middle [m]
C_i_innerskirts  = 0.25*(2*L_plate_innerskirts + 2*(L_plate_innerskirts...
    -(2*t_plates)));
% Calculating the average circumference of the plates in the middle [m]
C_av_innerskirts = 0.25*((C_o_innerskirts+C_i_innerskirts)/2);

% The diagonal plates connecting the middle plates to the skirts
% Calculating the outer circumference of the diagonal plates [m]
C_o_diagonalskirts = 0.25*(2*L_plate_diagonalskirts +...
    2*(L_plate_diagonalskirts+(2*t_plates)));
% Calculating the inner circumference of the diagonal plates [m]
C_i_diagonalskirts = 0.25*(2*L_plate_diagonalskirts +...
    2*(L_plate_diagonalskirts-(2*t_plates)));
% Calculating the average circumference of the diagonal plates [m]
C_av_diagonalskirts = 0.25*((C_o_diagonalskirts+C_i_diagonalskirts)/2);

%-------------------------------------------------------------------------
% Calculating the remaining parameters of the plates and the system
%-------------------------------------------------------------------------

% Calculation of parameters of the plates
% Volume calculation of the skirts considered for 1 caisson [m^3]
Vol_tot_plates          = (4*(L_plate_outerskirts*H_trap*t_plates)) +...
    (4*(L_plate_innerskirts*H_square*t_plates))+...
    (4*(L_plate_diagonalskirts*H_trap*t_plates));
% Weight calculation of the skirt [kg]
Weight_plates           = Vol_tot_plates*rho_mat;
% Vertical load due to the weight of the plate [N]
WeightForce_plates      = Weight_plates*gravity;
% Buoyancy force of the submerged plate [N]
Fbuoy_plates            = Vol_tot_plates*rho_wat*gravity;
% Volume calculation of the platform plate [m^3]
Vol_plate_platform      = (L_plate_platform*W_plate_platform)*t_plate_platform;
% Weight calculation of the platform plate [kg]
Weight_plate_platform   = Vol_plate_platform*rho_mat;
% Vertical load due to the weight of the platform plate [N]
WeightForce_platform    = Weight_plate_platform*gravity;
% Buoyancy force of the submerged platform plate [N]
Fbuoy_platform          = Vol_plate_platform*rho_wat*gravity;

% Calculation of the reservoir weight
Reservoir_weight        = Total_system_Weight-Weight_plate_platform-...
    Weight_plates-(4*Weight_SC);
Reservoir_weight_force  = Reservoir_weight*gravity;
Fbouy_Reservoir         = (Reservoir_weight/rho_mat)*rho_wat*gravity;

% Calculation of total forces
% Total Vertical load [N]
WeightForce_total       = WeightForce_SC + 0.25*WeightForce_plates + ...
     0.25*WeightForce_platform + 0.25*Reservoir_weight_force;
% Total buoyancy load [N]
Fbuoy_total             = Fbuoy_SC + 0.25*Fbuoy_plates + ...
     0.25*Fbuoy_platform + 0.25*Fbouy_Reservoir;

%-------------------------------------------------------------------------
% Calculating the  parameters of square caisson
%-------------------------------------------------------------------------

% Calculating the area of the square caisson
A_SQ_inner = L_plate_innerskirts*(L_plate_innerskirts - (2*t_plates));
A_SQ_outer = L_plate_innerskirts*(L_plate_innerskirts + (2*t_plates));
A_SQ_tip = (A_SQ_outer-A_SQ_inner);

%-------------------------------------------------------------------------
% Calculating the  parameters of trapezoidal caissons
%-------------------------------------------------------------------------

% Calculating dimensions of the trapezoidal shapes
h_trap = sqrt((L_plate_diagonalskirts)^2-((L_plate_outerskirts-...
    L_plate_innerskirts)/2)^2);
h_rect = (L_plate_platform/2)-(L_plate_innerskirts/2)-h_trap;
angle_skirtcaisson = 45; %degrees

% Calculating the areas of the trapezoidal shapes
A_circsegouter = 0.5*(((pi/180)*angle_skirtcaisson)-sind...
    (angle_skirtcaisson))*((D_inner/2)^2);
```

```
A_circseginner = 0.5*(((pi/180)*angle_skirtcaisson)-sind...
    (angle_skirtcaisson))*((D_outer/2)^2);

A_trap_outer = ((L_plate_outerskirts+L_plate_innerskirts)/2)*h_trap;
A_rect_outer = (L_plate_outerskirts*h_rect)-(2*A_circseginner);
A_outer_trapC = A_trap_outer + A_rect_outer;

A_trap_inner = (((L_plate_outerskirts-2*t_plates)+...
    (L_plate_innerskirts-2*t_plates))/2)*(h_trap-t_plates);
A_rect_inner = ((L_plate_outerskirts-2*t_plates)*(h_rect-t_plates))...
    -(2*A_circsegouter);
A_inner_trapC = A_trap_inner + A_rect_inner;

A_tip_trap = (A_inner_trapC + A_outer_trapC)/2;

% Calculating the circumferences of the trapezoidal shapes
arc_inner = (angle_skirtcaisson/360)*2*pi*(D_outer/2);
arc_outer = (angle_skirtcaisson/360)*2*pi*(D_inner/2);

C_outer_trap = L_plate_outerskirts+(2*arc_outer)+(2*...
    L_plate_diagonalskirts)+L_plate_innerskirts;
C_inner_trap = (L_plate_outerskirts-2*t_plates)+(2*arc_inner)+(2*...
    (L_plate_diagonalskirts-2*t_plates))+ (L_plate_innerskirts-2*t_plates);
C_av_trap = (C_outer_trap + C_inner_trap)/2;



%% Calculations for the self-weight penetration depth (h_Self_Weight)
Nq=1; % Due to undrained conditions
f=0.7;
Nc_isk = 9;

%------------------------------------------------------------------------
% Calculating the Self-Weight penetration phase (Cohesive case)
%------------------------------------------------------------------------

% Initializing the values for all intermediate steps
Q_outer_SW    = 0;
Q_inner_SW    = 0;
Q_tip_SW      = 0;
h_Self_Weight = 0;
Qtot_SW       = 0;

while Qtot_SW < WeightForce_total
% Updating the depth for every step of the loop [m]
h_Self_Weight= h_Self_Weight+0.00000001;

%%% If the square and trapezoidal plates not reach the seabed in the first phase
if h_Self_Weight < H_split1
Q_outer_plates_SW   = 0;
Q_inner_plates_SW   = 0;
Q_tip_plates_SW     = 0;

% Initializing terms su1 and su2 (undrained shear strengths)
% Calculating the average undrained shear strength between
% mudline and depth h [Pa]
s_u1 = s_u0 + rate_change*(h_Self_Weight/2);
% Calculating the undrained shear strength at depth h [Pa]
s_u2 = s_u0 + rate_change*h_Self_Weight;

% Calculating the resistance forces
Q_outer_SW = h_Self_Weight*alpha*s_u1*(pi*D_outer);
Q_inner_SW = h_Self_Weight*alpha*s_u1*(pi*D_inner);
Q_tip_SW = (y_komma*h_Self_Weight*Nq + s_u2*Nc)*(pi*D_av*t);

Qtot_SW = Q_outer_SW + Q_inner_SW + Q_tip_SW + Q_outer_plates_SW ...
    + Q_inner_plates_SW + + Q_tip_plates_SW + Fbuoy_total;

%If the square plates reach the seabed in the first phase
elseif (H_split1 < h_Self_Weight) && (h_Self_Weight < H_split2)
```

```
% Initializing terms su1 and su2 (undrained shear strengths)
% Calculating the average undrained shear strength between
% mudline and depth h [Pa]
s_u1 = s_u0 + rate_change*(h_Self_Weight/2);
% Calculating the undrained shear strength at depth h [Pa]
s_u2 = s_u0 + rate_change*h_Self_Weight;

% Calculating the resistance forces
Q_outer_SW = h_Self_Weight*alpha*s_u1*(pi*D_outer);
Q_inner_SW = h_Self_Weight*alpha*s_u1*(pi*D_inner);
Q_tip_SW = (y_komma*h_Self_Weight*Nq + s_u2*Nc)*(pi*D_av*t);

% Calculating the frictional terms of the square plates
Q_outer_plates_SW = (h_Self_Weight-H_split1)*alpha*s_u1*(C_o_innerskirts);
Q_inner_plates_SW = (h_Self_Weight-H_split1)*alpha*s_u1*(C_i_innerskirts);
Q_tip_plates_SW = ((y_komma*(h_Self_Weight-H_split1)*Nq)+(s_u2*Nc))*...
    ((C_av_innerskirts)*t_plates);

Qtot_SW = Q_outer_SW + Q_inner_SW + Q_tip_SW + Q_outer_plates_SW ...
    + Q_inner_plates_SW + + Q_tip_plates_SW + Fbuoy_total;

%%% If all the plates reach the seabed in the first phase
else
% Initializing terms su1 and su2 (undrained shear strengths)
% Calculating the average undrained shear strength between
% mudline and depth h [Pa]
s_u1 = s_u0 + rate_change*(h_Self_Weight/2);
% Calculating the undrained shear strength at depth h [Pa]
s_u2 = s_u0 + rate_change*h_Self_Weight;

% Calculating the resistance forces
Q_outer_SW = h_Self_Weight*alpha*s_u1*(pi*D_outer);
Q_inner_SW = h_Self_Weight*alpha*s_u1*(pi*D_inner);
Q_tip_SW = (y_komma*h_Self_Weight*Nq + s_u2*Nc)*(pi*D_av*t);

% Calculating the frictional terms of the plates combined
Q_outer_plates_SW = (h_Self_Weight-H_split2)*alpha*s_u1*(C_o_outerskirts...
    + C_o_diagonalskirts)+...
    (h_Self_Weight-H_split1)*alpha*s_u1*(C_o_innerskirts);
Q_inner_plates_SW = (h_Self_Weight-H_split2)*alpha*s_u1*(C_i_outerskirts...
    + C_i_diagonalskirts)+...
    (h_Self_Weight-H_split1)*alpha*s_u1*(C_i_innerskirts);
Q_tip_plates_SW = ((y_komma*(h_Self_Weight-H_split2)*Nq)+(s_u2*Nc))*...
    ((C_av_outerskirts+C_av_diagonalskirts)*t_plates)+...
    ((y_komma*(h_Self_Weight-H_split1)*Nq)+(s_u2*Nc))*...
    ((C_av_innerskirts)*t_plates);

Qtot_SW = Q_outer_SW + Q_inner_SW + Q_tip_SW + Q_outer_plates_SW ...
    + Q_inner_plates_SW + + Q_tip_plates_SW + Fbuoy_total;
end
end

%% For loop for the Self-Weight penetration to determine force per depth
%Initializing all arrays, to be able to store the data.
h_depth = [0:0.00001:h_Self_Weight];

for i = 1:1:length(h_depth)
%%% If the plates not reach the seabed in the first phase
if h_depth(i) < H_split1

V_Q_outer_plates_SW(i) = 0;
V_Q_inner_plates_SW(i) = 0;
V_Q_tip_plates_SW(i) = 0;

% Initializing terms su1 and su2 (undrained shear strengths)
V_s_u1(i) = s_u0 + rate_change*(h_depth(i)/2);
V_s_u2(i) = s_u0 + rate_change*h_depth(i);

% Calculating the resistance forces
V_Q_outer_SW(i) = h_depth(i)*alpha*V_s_u1(i)*(pi*D_outer);
```

```
V_Q_inner_SW(i) = h_depth(i)*alpha*V_s_u1(i)*(pi*D_inner);
V_Q_tip_SW(i) = (y_komma*h_depth(i)*Nq + V_s_u2(i)*Nc)*(pi*D_av*t);

V_Qtot_SW(i) = V_Q_outer_SW(i) + V_Q_inner_SW(i) + V_Q_tip_SW(i) +...
    V_Q_outer_plates_SW(i)+ V_Q_inner_plates_SW(i) +...
    V_Q_tip_plates_SW(i) + Fbuoy_total;

% Holding Capacity Cylindrical Caisson
Q_su_SW(i) = alpha*s_u0*pi*D_outer*h_depth(i);
Q_b_SW(i) = s_u0*Nc_isk*f*((pi*(D_outer)^2)/4);
W_c_SW(i) = (4*WeightForce_total)/4;
W_s_SW(i) = (((pi*(D_outer)^2)/4)*h_depth(i))*(y_komma);

H_c_SW(i) = Q_su_SW(i)+Q_b_SW(i)+W_c_SW(i)+W_s_SW(i);
H_c2_SW(i) = H_c_SW(i)/1000;

% Holding Capacity Square Caisson
H_c2_SQ_SW(i) = 0;

% Holding Capacity Trapezoidal Caisson
H_c2_trap_SW(i) = 0;

elseif (H_split1 < h_depth(i)) && (h_depth(i) < H_split2)
% Part where the plate does reach the seabed
% Initializing terms su1 and su2 (undrained shear strengths)
V_s_u1(i) = s_u0 + rate_change*(h_depth(i)/2);
V_s_u2(i) = s_u0 + rate_change*h_depth(i);

% Calculating the resistance forces
V_Q_outer_SW(i) = h_depth(i)*alpha*V_s_u1(i)*(pi*D_outer);
V_Q_inner_SW(i) = h_depth(i)*alpha*V_s_u1(i)*(pi*D_inner);
V_Q_tip_SW(i) = (y_komma*h_depth(i)*Nq + V_s_u2(i)*Nc)*(pi*D_av*t);

V_Q_outer_plates_SW(i) = (h_depth(i)-H_split1)*alpha*s_u1*C_o_innerskirts;
V_Q_inner_plates_SW(i) = (h_depth(i)-H_split1)*alpha*s_u1*C_i_innerskirts;
V_Q_tip_plates_SW(i) = ((y_komma*(h_depth(i)-H_split1)*Nq)+(s_u2*Nc))...
    *((C_av_innerskirts)*t_plates);


V_Qtot_SW(i) = V_Q_outer_SW(i) + V_Q_inner_SW(i) + V_Q_tip_SW(i) +...
    V_Q_outer_plates_SW(i)+ V_Q_inner_plates_SW(i)+ V_Q_tip_plates_SW(i)...
    + Fbuoy_total;

% Holding Capacity Cylindrical Caisson
Q_su_SW(i) = alpha*s_u0*pi*D_outer*h_depth(i);
Q_b_SW(i) = s_u0*Nc_isk*f*((pi*(D_outer)^2)/4);
W_c_SW(i) = (4*WeightForce_total)/5;
W_s_SW(i) = (((pi*(D_outer)^2)/4)*h_depth(i))*(y_komma);

H_c_SW(i) = Q_su_SW(i)+Q_b_SW(i)+W_c_SW(i)+W_s_SW(i);
H_c2_SW(i) = H_c_SW(i)/1000;

% Holding Capacity Square Caisson
Q_su_SQ_SW(i) = alpha*s_u0*(4*C_o_innerskirts)*(h_depth(i)-H_split1);
Q_b_SQ_SW(i) = s_u0*Nc_isk*f*(A_SQ_outer);
W_c_SQ_SW(i) = (4*WeightForce_total)/5;
W_s_SQ_SW(i) = ((A_SQ_outer)*(h_depth(i)-H_split1))*(y_komma);

H_c_SQ_SW(i) = Q_su_SQ_SW(i)+Q_b_SQ_SW(i)+W_c_SQ_SW(i)+W_s_SQ_SW(i);
H_c2_SQ_SW(i) = H_c_SQ_SW(i)/1000;

% Holding Capacity Trapezoidal Caisson
H_c2_trap_SW(i) = 0;

else
% Part where the plate does reach the seabed
% Initializing terms su1 and su2 (undrained shear strengths)
V_s_u1(i) = s_u0 + rate_change*(h_depth(i)/2);
V_s_u2(i) = s_u0 + rate_change*h_depth(i);
```

```
% Calculating the resistance forces
V_Q_outer_SW(i) = h_depth(i)*alpha*V_s_u1(i)*(pi*D_outer);
V_Q_inner_SW(i) = h_depth(i)*alpha*V_s_u1(i)*(pi*D_inner);
V_Q_tip_SW(i) = (y_komma*h_depth(i)*Nq + V_s_u2(i)*Nc)*(pi*D_av*t);

V_Q_outer_plates_SW(i) = (h_depth(i)-H_split2)*alpha*s_u1*...
    (C_o_outerskirts + C_o_diagonalskirts)+...
    (h_depth(i)-H_split1)*alpha*s_u1*C_o_innerskirts;
V_Q_inner_plates_SW(i) = (h_depth(i)-H_split2)*alpha*s_u1*...
    (C_i_outerskirts + C_i_diagonalskirts)+...
    (h_depth(i)-H_split1)*alpha*s_u1*C_i_innerskirts;
V_Q_tip_plates_SW(i) = ((y_komma*(h_depth(i)-H_split2)*Nq)+(s_u2*Nc))...
    *((C_av_outerskirts+C_av_diagonalskirts)*t_plates)+...
    ((y_komma*(h_depth(i)-H_split1)*Nq)+(s_u2*Nc))...
    *((C_av_innerskirts)*t_plates);

V_Qtot_SW(i) = V_Q_outer_SW(i) + V_Q_inner_SW(i) + V_Q_tip_SW(i) +...
    V_Q_outer_plates_SW(i)+ V_Q_inner_plates_SW(i)+ V_Q_tip_plates_SW(i)...
    + Fbuoy_total;

% Holding Capacity Cylindrical Caisson
Q_su_SW(i) = alpha*s_u0*pi*D_outer*h_depth(i);
Q_b_SW(i) = s_u0*Nc_isk*f*((pi*(D_outer)^2)/4);
W_c_SW(i) = (4*WeightForce_total)/9;
W_s_SW(i) = (((pi*(D_outer)^2)/4)*h_depth(i))*(y_komma);

H_c_SW(i) = Q_su_SW(i)+Q_b_SW(i)+W_c_SW(i)+W_s_SW(i);
H_c2_SW(i) = H_c_SW(i)/1000;

% Holding Capacity Square Caisson
Q_su_SQ_SW(i) = alpha*s_u0*(4*C_o_innerskirts)*(h_depth(i)-H_split1);
Q_b_SQ_SW(i) = s_u0*Nc_isk*f*(A_SQ_outer);
W_c_SQ_SW(i) = (4*WeightForce_total)/9;
W_s_SQ_SW(i) = ((A_SQ_outer)*(h_depth(i)-H_split1))*(y_komma);

H_c_SQ_SW(i) = Q_su_SQ_SW(i)+Q_b_SQ_SW(i)+W_c_SQ_SW(i)+W_s_SQ_SW(i);
H_c2_SQ_SW(i) = H_c_SQ_SW(i)/1000;

% Holding Capacity Trapezoidal Caisson
Q_su_trap_SW(i) = alpha*s_u0*(4*C_outer_trap)*(h_depth(i)-H_split2);
Q_b_trap_SW(i) = s_u0*Nc_isk*f*(A_trap_outer);
W_c_trap_SW(i) = (4*WeightForce_total)/9;
W_s_trap_SW(i) = ((A_trap_outer)*(h_depth(i)-H_split2))*(y_komma);

H_c_trap_SW(i) = Q_su_trap_SW(i)+Q_b_trap_SW(i)+W_c_trap_SW(i)+...
    W_s_trap_SW(i);
H_c2_trap_SW(i) = H_c_trap_SW(i)/1000;
end
end

%% For loop for the Suction Assisted penetration to determine force per depth
%Setting parameters and creating the depth array for the suction assisted
%phase
m = 1;
h_max = H_inner;
h_depth_SA = [h_Self_Weight:0.001:h_max];

% Critical Suction
for i = 1:1:length(h_depth_SA)
s_u1_SA(i) = s_u0 + rate_change*(h_depth_SA(i)/2);
s_u2_SA(i) = s_u0 + rate_change*h_depth_SA(i);
Nc_star = 3*4*alpha;
D_m = m*D_av;

% Calculating the critical suction by Yuqi
s_crit_Y(i) = (y_komma*(h_depth_SA(i)) + ((4*(h_depth_SA(i))*alpha*...
    (s_u1_SA(i)))/D_inner))-(y_komma*(h_depth_SA(i)) +((((1+((1.5*...
    (h_depth_SA(i)))/D_outer))^2)*alpha*(s_u1_SA(i)))/(1+((3*...
    (h_depth_SA(i)))/D_outer)))) +(2 + pi + asin(alpha) + asin(alpha)...
    + sqrt(1-(alpha^2)) - sqrt(1-(alpha^2)))*(s_u2_SA(i));
% s_crit(i) = ((pi*D_inner*h_depth_SA(i)*alpha*s_u1_SA(i))/(0.25*pi*(D_inner^2)))-
...
%     ((pi*D_outer*h_depth_SA(i)*alpha*s_u1_SA(i))/(0.25*((D_m^2)-
(D_outer^2))))+...
%     Nc_star*s_u2_SA(i);
end

if h_Self_Weight < H_seal1
% Only the cylindrical caissons have sufficient penetration depth and seal
% After the self-weight phase

%Determine depth arrays for the installation steps
H_SAC = length([h_Self_Weight:0.001:H_split1]);
H_SAS = length([H_split1+0.001:0.001:H_split2]);
h_CSSA = H_SAC + H_SAS;

for  i = 1:1:H_SAC
% Initializing the resistance forces of the plates, which are zero in
% this part
V_Q_outer_plates_SA(i) = 0;
V_Q_inner_plates_SA(i) = 0;
V_Q_tip_plates_SA(i) = 0;

syms p2
eqn = WeightForce_total + p2*((pi*(D_inner)^2)/4) == h_depth_SA(i)*alpha*...
    s_u1_SA(i)*(pi*D_outer)+ h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_inner)...
    + (y_komma*h_depth_SA(i) - p2 + s_u2_SA(i)*Nc)*(pi*D_av*t) + ...
    V_Q_outer_plates_SA(i) + V_Q_inner_plates_SA(i) + V_Q_tip_plates_SA(i)...
    + Fbuoy_total;

s2(i) = double (solve(eqn,p2));
% Calculating the forces acting within the system
V_Force_SA(i) = WeightForce_total + s2(i)*((pi*(D_inner)^2)/4);
V_Q_outer_SA(i) = h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_outer);
V_Q_inner_SA(i) = h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_inner);
V_Q_tip_SA(i) = (y_komma*h_depth_SA(i)*Nq - s2(i) + s_u2_SA(i)*Nc)...
    *(pi*D_av*t);
V_Qtot_SA(i) = V_Q_outer_SA(i) + V_Q_inner_SA(i) + V_Q_tip_SA(i) + ...
    V_Q_outer_plates_SA(i) + V_Q_inner_plates_SA(i) + V_Q_tip_plates_SA(i)...
    + Fbuoy_total;

% Holding Capacity Cylindrical Caisson
Q_su_SA(i) = alpha*s_u0*pi*D_outer*h_depth_SA(i);
Q_b_SA(i) = s_u0*Nc_isk*f*((pi*(D_outer)^2)/4);
W_c_SA(i) = (4*WeightForce_total)/4;
W_s_SA(i) = (((pi*(D_outer)^2)/4)*h_depth_SA(i))*(y_komma);

H_c_SA(i) = Q_su_SA(i)+Q_b_SA(i)+W_c_SA(i)+W_s_SA(i);
H_c2_SA(i) = H_c_SA(i)/1000;

% Holding Capacity Square Caisson
H_c2_SQ_SA(i) = 0;

% Holding Capacity Trapezoidal Caisson
H_c2_trap_SA(i) = 0;
end

for i = H_SAC+1:1:h_CSSA
% The plating for that is not used as a caisson is now
% determined for a full system instead of a quarter
V_Q_outer_plates_SA(i) = 0;
V_Q_inner_plates_SA(i) = 0;
V_Q_tip_plates_SA(i) = 0;

% Calculating the new variable s [Pa], based on the full system
% In this equation, the additional terms for the square
% caisson are considered

syms p2
```

```
eqn = (4*WeightForce_total) + 4*(p2*((pi*(D_inner)^2)/4)) + (p2*A_SQ_inner)...
    == 4*(h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_outer)+ h_depth_SA(i)*...
    alpha*s_u1_SA(i)*(pi*D_inner) + (y_komma*h_depth_SA(i) - p2 + ...
    s_u2_SA(i)*Nc)*(pi*D_av*t))+...
    (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*(4*C_o_innerskirts)+...
    (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*(4*C_i_innerskirts) +...
    ( y_komma*(h_depth_SA(i)-H_split1) - p2 +s_u2_SA(i)*Nc)*(A_SQ_tip)+...
    + 4*(Fbuoy_total);

s2(i) = double (solve(eqn,p2));

% Calculating the forces acting within the system
V_Force_SA(i) = WeightForce_total + s2(i)*((pi*(D_inner)^2)/4);
V_Q_outer_SA(i) = h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_outer);
V_Q_inner_SA(i) = h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_inner);
V_Q_tip_SA(i) = (y_komma*h_depth_SA(i)*Nq - s2(i) + s_u2_SA(i)*Nc)...
    *(pi*D_av*t);

% Calculating the terms for the square suction caisson
V_Q_outer_SA_SQ(i) = (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*...
    (4*C_o_innerskirts);
V_Q_inner_SA_SQ(i) = (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*...
    (4*C_i_innerskirts);
V_Q_tip_SA_SQ(i) = (y_komma*(h_depth_SA(i)-H_split1)*Nq - s2(i)...
    + s_u2_SA(i)*Nc)*(A_SQ_tip);

V_Qtot_SA(i) = V_Q_outer_SA(i) + V_Q_inner_SA(i) + V_Q_tip_SA(i) + ...
    V_Q_outer_plates_SA(i)+ V_Q_outer_SA_SQ(i) + V_Q_inner_SA_SQ(i) ...
    + V_Q_tip_SA_SQ(i) + V_Q_inner_plates_SA(i) + V_Q_tip_plates_SA(i) ...
    + 4*Fbuoy_total;

% Holding Capacity Cylindrical Caisson
Q_su_SA(i) = alpha*s_u0*pi*D_outer*h_depth_SA(i);
Q_b_SA(i) = s_u0*Nc_isk*f*((pi*(D_outer)^2)/4);
W_c_SA(i) = (4*WeightForce_total)/5;
W_s_SA(i) = (((pi*(D_outer)^2)/4)*h_depth_SA(i))*(y_komma);

H_c_SA(i) = Q_su_SA(i)+Q_b_SA(i)+W_c_SA(i)+W_s_SA(i);
H_c2_SA(i) = H_c_SA(i)/1000;

% Holding Capacity Square Caisson
Q_su_SQ_SA(i) = alpha*s_u0*(4*C_o_innerskirts)*(h_depth_SA(i)-H_split1);
Q_b_SQ_SA(i) = s_u0*Nc_isk*f*(A_SQ_outer);
W_c_SQ_SA(i) = (4*WeightForce_total)/5;
W_s_SQ_SA(i) = ((A_SQ_outer)*(h_depth_SA(i)-H_split1))*(y_komma);

H_c_SQ_SA(i) = Q_su_SQ_SA(i)+Q_b_SQ_SA(i)+W_c_SQ_SA(i)+W_s_SQ_SA(i);
H_c2_SQ_SA(i) = H_c_SQ_SA(i)/1000;

% Holding Capacity Trapezoidal Caisson
H_c2_trap_SA(i) = 0;
end

for i = h_CSSA+1:1:length(h_depth_SA)
% Calculating the new variable s [Pa], based on the full system
% In this equation, the additional terms for the square and
% trapezoidal caisson are considered
syms p2
eqn = (4*WeightForce_total) + 4*(p2*((pi*(D_inner)^2)/4)) + (p2*A_SQ_inner)...
    + 4*(p2*A_inner_trapC) == 4*((h_depth_SA(i)*alpha*s_u1_SA(i))*(pi*D_outer)+...
    (h_depth_SA(i)*alpha*s_u1_SA(i))*(pi*D_inner) +...
    (y_komma*h_depth_SA(i) - p2 + s_u2_SA(i)*Nc)*(pi*D_av*t)) +...
    ((h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*(4*C_o_innerskirts)+...
    (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*(4*C_i_innerskirts) +...
    (y_komma*(h_depth_SA(i)-H_split1) - p2 + s_u2_SA(i)*Nc)*(A_SQ_tip))+
4*(Fbuoy_total)+...
    4*((h_depth_SA(i)-H_split2)*alpha*s_u1_SA(i)*(C_outer_trap)+...
    (h_depth_SA(i)-H_split2)*alpha*s_u1_SA(i)*(C_inner_trap)+...
    (y_komma*(h_depth_SA(i)-H_split2) - p2 + s_u2_SA(i)*Nc_isk)*(A_tip_trap));
```

```
s2(i) = double (solve(eqn,p2));

% Calculating the forces acting within the system
V_Force_SA(i) = WeightForce_total + s2(i)*((pi*(D_inner)^2)/4);
V_Q_outer_SA(i) = h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_outer);
V_Q_inner_SA(i) = h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_inner);
V_Q_tip_SA(i) = (y_komma*h_depth_SA(i)*Nq - s2(i) + s_u2_SA(i)*Nc)...
    *(pi*D_av*t);

% Calculating the terms for the square suction caisson
V_Q_outer_SA_SQ(i) = (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*...
    (4*C_o_innerskirts);
V_Q_inner_SA_SQ(i) = (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*...
    (4*C_i_innerskirts);
V_Q_tip_SA_SQ(i)   = (y_komma*(h_depth_SA(i)-H_split1)*Nq - s2(i) +...
    s_u2_SA(i)*Nc)*(A_SQ_tip);

% Calculating the terms for the trapezoidal suction caisson
V_Q_outer_SA_trap(i)= (h_depth_SA(i)-H_split2)*alpha*s_u1_SA(i)*...
    (C_outer_trap);
V_Q_inner_SA_trap(i)= (h_depth_SA(i)-H_split2)*alpha*s_u1_SA(i)*...
    (C_inner_trap);
V_Q_tip_SA_trap(i)= (y_komma*(h_depth_SA(i)-H_split2)*Nq - s2(i) +...
    s_u2_SA(i)*Nc)*(A_tip_trap);

V_Qtot_SA(i) = V_Q_outer_SA(i) + V_Q_inner_SA(i) + V_Q_tip_SA(i) + ...
    + V_Q_outer_SA_SQ(i) + V_Q_inner_SA_SQ(i) + V_Q_tip_SA_SQ(i) +...
    + 4*Fbuoy_total + V_Q_tip_SA_trap(i) + V_Q_inner_SA_trap(i) +...
    V_Q_outer_SA_trap(i);

% Holding Capacity Cylindrical Caisson
Q_su_SA(i) = alpha*s_u0*pi*D_outer*h_depth_SA(i);
Q_b_SA(i) = s_u0*Nc_isk*f*((pi*(D_outer)^2)/4);
W_c_SA(i) = (4*WeightForce_total)/9;
W_s_SA(i) = (((pi*(D_outer)^2)/4)*h_depth_SA(i))*(y_komma);

H_c_SA(i) = Q_su_SA(i)+Q_b_SA(i)+W_c_SA(i)+W_s_SA(i);
H_c2_SA(i) = H_c_SA(i)/1000;

% Holding Capacity Square Caisson
Q_su_SQ_SA(i) = alpha*s_u0*(4*C_o_innerskirts)*(h_depth_SA(i)-H_split1);
Q_b_SQ_SA(i) = s_u0*Nc_isk*f*(A_SQ_outer);
W_c_SQ_SA(i) = (4*WeightForce_total)/9;
W_s_SQ_SA(i) = ((A_SQ_outer)*(h_depth_SA(i)-H_split1))*(y_komma);

H_c_SQ_SA(i) = Q_su_SQ_SA(i)+Q_b_SQ_SA(i)+W_c_SQ_SA(i)+W_s_SQ_SA(i);
H_c2_SQ_SA(i) = H_c_SQ_SA(i)/1000;

% Holding Capacity Trapezoidal Caisson
Q_su_trap_SA(i) = alpha*s_u0*(4*C_outer_trap)*(h_depth_SA(i)-H_split2);
Q_b_trap_SA(i) = s_u0*Nc_isk*f*(A_trap_outer);
W_c_trap_SA(i) = (4*WeightForce_total)/9;
W_s_trap_SA(i) = ((A_trap_outer)*(h_depth_SA(i)-H_split2))*(y_komma);

H_c_trap_SA(i) = Q_su_trap_SA(i)+Q_b_trap_SA(i)+W_c_trap_SA(i)+...
                 W_s_trap_SA(i);
H_c2_trap_SA(i) = H_c_trap_SA(i)/1000;
end

elseif (H_seal1 <= h_Self_Weight) && (h_Self_Weight < H_seal2)
% The cylindrical and square caissons have sufficient penetration depth and seal
% After the self-weight phase

%Determine depth arrays for the installation steps
H_SAS = length([h_Self_Weight:0.001:H_split2]);

for i = 1:1:H_SAS
% The plating for that is not used as a caisson is now
% determined for a full system instead of a quarter
V_Q_outer_plates_SA(i) = 0;
V_Q_inner_plates_SA(i) = 0;
V_Q_tip_plates_SA(i) = 0;

% Calculating the new variable s [Pa], based on the full system
% In this equation, the additional terms for the square
% caisson are considered

syms p2
eqn = (4*WeightForce_total) + 4*(p2*((pi*(D_inner)^2)/4)) + (p2*A_SQ_inner)...
    == 4*(h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_outer)+ h_depth_SA(i)*...
    alpha*s_u1_SA(i)*(pi*D_inner) + (y_komma*h_depth_SA(i) - p2 + ...
    s_u2_SA(i)*Nc)*(pi*D_av*t))+(h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)...
    *(4*C_o_innerskirts)+ (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*...
    (4*C_i_innerskirts) + ( y_komma*(h_depth_SA(i)-H_split1) - p2 +...
    s_u2_SA(i)*Nc)*(A_SQ_tip) + 4*(Fbuoy_total);

s2(i) = double (solve(eqn,p2));

% Calculating the forces acting within the system
V_Force_SA(i) = WeightForce_total + s2(i)*((pi*(D_inner)^2)/4);
V_Q_outer_SA(i) = h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_outer);
V_Q_inner_SA(i) = h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_inner);
V_Q_tip_SA(i) = (y_komma*h_depth_SA(i)*Nq - s2(i) + s_u2_SA(i)*Nc)...
    *(pi*D_av*t);

% Calculating the terms for the square suction caisson
V_Q_outer_SA_SQ(i) = (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*...
    (4*C_o_innerskirts);
V_Q_inner_SA_SQ(i) = (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*...
    (4*C_i_innerskirts);
V_Q_tip_SA_SQ(i) = (y_komma*(h_depth_SA(i)-H_split1)*Nq - s2(i)...
    + s_u2_SA(i)*Nc)*(A_SQ_tip);

V_Qtot_SA(i) = V_Q_outer_SA(i) + V_Q_inner_SA(i) + V_Q_tip_SA(i) + ...
    V_Q_outer_plates_SA(i)+ V_Q_outer_SA_SQ(i) + V_Q_inner_SA_SQ(i) ...
    + V_Q_tip_SA_SQ(i) + V_Q_inner_plates_SA(i) + V_Q_tip_plates_SA(i) ...
    + 4*Fbuoy_total;

% Holding Capacity Cylindrical Caisson
Q_su_SA(i) = alpha*s_u0*pi*D_outer*h_depth_SA(i);
Q_b_SA(i) = s_u0*Nc_isk*f*((pi*(D_outer)^2)/4);
W_c_SA(i) = (4*WeightForce_total)/5;
W_s_SA(i) = (((pi*(D_outer)^2)/4)*h_depth_SA(i))*(y_komma);

H_c_SA(i) = Q_su_SA(i)+Q_b_SA(i)+W_c_SA(i)+W_s_SA(i);
H_c2_SA(i) = H_c_SA(i)/1000;

% Holding Capacity Square Caisson
Q_su_SQ_SA(i) = alpha*s_u0*(4*C_o_innerskirts)*(h_depth_SA(i)-H_split1);
Q_b_SQ_SA(i) = s_u0*Nc_isk*f*(A_SQ_outer);
W_c_SQ_SA(i) = (4*WeightForce_total)/5;
W_s_SQ_SA(i) = ((A_SQ_outer)*(h_depth_SA(i)-H_split1))*(y_komma);

H_c_SQ_SA(i) = Q_su_SQ_SA(i)+Q_b_SQ_SA(i)+W_c_SQ_SA(i)+W_s_SQ_SA(i);
H_c2_SQ_SA(i) = H_c_SQ_SA(i)/1000;

% Holding Capacity Trapezoidal Caisson
H_c2_trap_SA(i) = 0;
end

for i = H_SAS+1:1:length(h_depth_SA)
% Calculating the new variable s [Pa], based on the full system
% In this equation, the additional terms for the square and
% trapezoidal caisson are considered
syms p2
eqn = (4*WeightForce_total) + 4*(p2*((pi*(D_inner)^2)/4)) + (p2*A_SQ_inner)...
    + (4*p2*A_inner_trapC) == 4*((h_depth_SA(i)*alpha*s_u1_SA(i))*(pi*D_outer)+...
    (h_depth_SA(i)*alpha*s_u1_SA(i))*(pi*D_inner) +...
    (y_komma*h_depth_SA(i) - p2 + s_u2_SA(i)*Nc)*(pi*D_av*t)) +...
    ((h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*(4*C_o_innerskirts)+...
    (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*(4*C_i_innerskirts) +...
    (y_komma*(h_depth_SA(i)-H_split1) - p2 + s_u2_SA(i)*Nc)*(A_SQ_tip))+ ...
    4*((h_depth_SA(i)-H_split2)*alpha*s_u1_SA(i)*(C_outer_trap)+...
    (h_depth_SA(i)-H_split2)*alpha*s_u1_SA(i)*(C_inner_trap) +...
    (y_komma*(h_depth_SA(i)-H_split2) - p2 + s_u2_SA(i)*Nc_isk)*(A_tip_trap)) + ...
    4*(Fbuoy_total);

s2(i) = double (solve(eqn,p2));

% Calculating the forces acting within the system
V_Force_SA(i) = WeightForce_total + s2(i)*((pi*(D_inner)^2)/4);
V_Q_outer_SA(i) = h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_outer);
V_Q_inner_SA(i) = h_depth_SA(i)*alpha*s_u1_SA(i)*(pi*D_inner);
V_Q_tip_SA(i) = (y_komma*h_depth_SA(i)*Nq - s2(i) + s_u2_SA(i)*Nc)...
    *(pi*D_av*t);

% Calculating the terms for the square suction caisson
V_Q_outer_SA_SQ(i) = (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*...
    (4*C_o_innerskirts);
V_Q_inner_SA_SQ(i) = (h_depth_SA(i)-H_split1)*alpha*s_u1_SA(i)*...
    (4*C_i_innerskirts);
V_Q_tip_SA_SQ(i)   = (y_komma*(h_depth_SA(i)-H_split1)*Nq - s2(i) +...
    s_u2_SA(i)*Nc)*(A_SQ_tip);

% Calculating the terms for the trapezoidal suction caisson
V_Q_outer_SA_trap(i)= (h_depth_SA(i)-H_split2)*alpha*s_u1_SA(i)*...
    (C_outer_trap);
V_Q_inner_SA_trap(i)= (h_depth_SA(i)-H_split2)*alpha*s_u1_SA(i)*...
    (C_inner_trap);
V_Q_tip_SA_trap(i)= (y_komma*(h_depth_SA(i)-H_split2)*Nq - s2(i) +...
    s_u2_SA(i)*Nc)*(A_tip_trap);

V_Qtot_SA(i) = V_Q_outer_SA(i) + V_Q_inner_SA(i) + V_Q_tip_SA(i) + ...
    + V_Q_outer_SA_SQ(i) + V_Q_inner_SA_SQ(i) + V_Q_tip_SA_SQ(i) +...
    + 4*Fbuoy_total + V_Q_tip_SA_trap(i) + V_Q_inner_SA_trap(i) +...
    V_Q_outer_SA_trap(i);

% Holding Capacity Cylindrical Caisson
Q_su_SA(i) = alpha*s_u0*pi*D_outer*h_depth_SA(i);
Q_b_SA(i) = s_u0*Nc_isk*f*((pi*(D_outer)^2)/4);
W_c_SA(i) = (4*WeightForce_total)/9;
W_s_SA(i) = (((pi*(D_outer)^2)/4)*h_depth_SA(i))*(y_komma);

H_c_SA(i) = Q_su_SA(i)+Q_b_SA(i)+W_c_SA(i)+W_s_SA(i);
H_c2_SA(i) = H_c_SA(i)/1000;

% Holding Capacity Square Caisson
Q_su_SQ_SA(i) = alpha*s_u0*(4*C_o_innerskirts)*(h_depth_SA(i)-H_split1);
Q_b_SQ_SA(i) = s_u0*Nc_isk*f*(A_SQ_outer);
W_c_SQ_SA(i) = (4*WeightForce_total)/9;
W_s_SQ_SA(i) = ((A_SQ_outer)*(h_depth_SA(i)-H_split1))*(y_komma);

H_c_SQ_SA(i) = Q_su_SQ_SA(i)+Q_b_SQ_SA(i)+W_c_SQ_SA(i)+W_s_SQ_SA(i);
H_c2_SQ_SA(i) = H_c_SQ_SA(i)/1000;

% Holding Capacity Trapezoidal Caisson
Q_su_trap_SA(i) = alpha*s_u0*(4*C_outer_trap)*(h_depth_SA(i)-H_split2);
Q_b_trap_SA(i) = s_u0*Nc_isk*f*(A_trap_outer);
W_c_trap_SA(i) = (4*WeightForce_total)/9;
W_s_trap_SA(i) = ((A_trap_outer)*(h_depth_SA(i)-H_split2))*(y_komma);

H_c_trap_SA(i) = Q_su_trap_SA(i)+Q_b_trap_SA(i)+W_c_trap_SA(i)+...
                 W_s_trap_SA(i);
H_c2_trap_SA(i) = H_c_trap_SA(i)/1000;
end
end
%% Plotting the behaviour of the foundation system
```

```
%------------------------------------------------------------------------
% Plotting the behaviour of the foundation system for individual phases
% (Cohesive case)
%------------------------------------------------------------------------

% Plotting the total resistance force against the penetration depth
figure(1)
plot(h_depth, V_Qtot_SW)
title('Total resistance force during the Self-weight penetration phase')
xlabel('Penetration depth [m]')
ylabel('Total resistance force [N]')

figure(2)
plot(h_depth_SA,s2)
title('Suction required for a total depth of 0.5 m')
xlabel('Penetration depth [m]')
ylabel('Required suction [Pa]')

figure(3)
plot(h_depth_SA,V_Qtot_SA)
title('Total resistance forces during the suction assisted phase')
xlabel('Penetration depth [m]')
ylabel('Total resistance force [N]')

figure(4)
plot(h_depth_SA,V_Q_tip_SA)
title('Resistance force at the tips during the suction assisted phase')
xlabel('Penetration depth [m]')
ylabel('Resistance force at the tips [N]')

figure(5)
plot(h_depth_SA,V_Q_inner_SA)
title('Resistance force at the inside of the caisson during the suction assisted
phase')
xlabel('Penetration depth [m]')
ylabel('Resistance force at the inside of the caisson [N]')

figure(6)
plot(h_depth_SA,V_Q_outer_SA)
title('Resistance force at the outside of the caisson during the suction assisted
phase')
xlabel('Penetration depth [m]')
ylabel('Resistance force at the outside of the caisson [N]')

figure(7)
plot(h_depth_SA, V_Q_outer_SA)
hold on
plot(h_depth_SA, V_Q_inner_SA)
hold on
plot(h_depth_SA, V_Q_tip_SA)
hold off

title('Resistance forces during the suction assisted penetration phase')
xlabel('Penetration depth [m]')
ylabel('Resistance force [N]')
legend('Outer resistance force', 'Inner resistance force',...
    'Tip resistance force')

%------------------------------------------------------------------------
% Plotting the behaviour of the foundation system for all phases
% (Cohesive case)
%------------------------------------------------------------------------

h_tot  = [h_depth h_depth_SA];
V_Q_outer_tot = [V_Q_outer_SW V_Q_outer_SA];
V_Q_inner_tot = [V_Q_inner_SW V_Q_inner_SA];
V_Q_tip_tot = [V_Q_tip_SW V_Q_tip_SA];
V_Qtot = [V_Qtot_SW  V_Qtot_SA];
H_c2 = [H_c2_SW  H_c2_SA];
```

```
H_c2_SQ = [H_c2_SQ_SW  H_c2_SQ_SA];
H_c2_trap = [H_c2_trap_SW H_c2_trap_SA];

% Total penetration resistance for the full depth
figure(8)
plot(h_tot,V_Qtot)
title('Total resistance force during the installation')
xlabel('Penetration depth [m]')
ylabel('Total resistance force [N]')

% Inner, outer and tip resistance for the full depth
figure(9)
plot(h_tot, V_Q_outer_tot)
hold on
plot(h_tot, V_Q_inner_tot)
hold on
plot(h_tot, V_Q_tip_tot)
hold off
%
title('Resistance forces over the penetration depth')
xlabel('Penetration depth [m]')
ylabel('Resistance force [N]')
legend('Outer resistance force', 'Inner resistance force',...
    'Tip resistance force', 'Location', 'Southeast')

% critical suction
figure(10)
plot(h_depth_SA,s2)
hold on
plot(h_depth_SA,s_crit_Y)
hold off

title('Calculated suction vs Critical suction ')
xlabel('Penetration depth [m]')
ylabel('Required suction [Pa]')
legend('Calculated Suction', 'Critical Suction Yuqi',...
    'Location', 'Southeast')

% Holding capacity per chamber
figure(11)
plot(h_tot,H_c2)
hold on
plot(h_tot,H_c2_SQ)
hold on
plot(h_tot,H_c2_trap)
hold off

title('Holding capacity of one suction caisson')
xlabel('Penetration depth [m]')
ylabel('Holding capacity [kN]')
legend('Circular Caissons','Square Caisson', 'Trapezoidal Caisson' ,...
    'Location', 'Southeast')
```