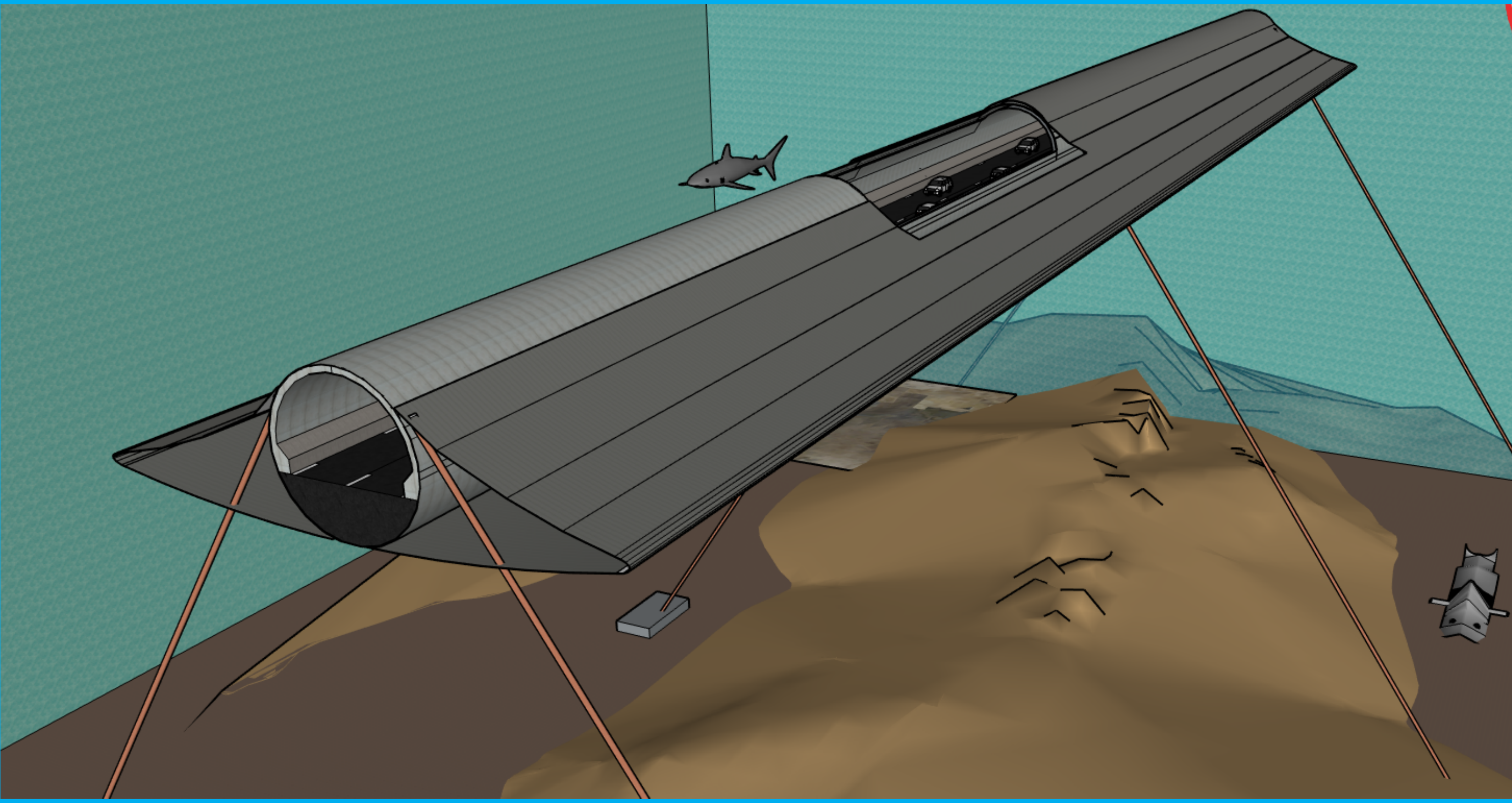
Form Finding for a Submerged Floating Tunnel

The Clever Cross-section for Coastal Crossings

C.J.F. van Marrewijk



TU Delft

Form Finding for a Submerged Floating Tunnel

The Clever Cross-section for Coastal Crossings

by

C.J.F. van Marrewijk

as the Thesis of the Master of Science in Civil Engineering
at the Delft University of Technology

Student number:	4349946	
Project duration:	September 3, 2019 – June 2, 2020	
Supervisors:	Prof. dr. ir. S.N. Jonkman,	Chairman
	Dr. ir. D.J. Peters,	Daily supervisor
	Ir. C.M.P. ’t Hart,	3rd Committee member
	Ir. P. Eigenraam,	4th Committee member

Preface

C.J.F. van Marrewijk
Delft, June 2, 2020

This report is a MSc-thesis and is written in order to fulfill the requirements of the MSc degree in Civil Engineering at the Delft University of Technology, at the faculty of Civil Engineering and Geo sciences. The track within the MSc chosen is Hydraulic Engineering, with a specialisation in Hydraulic Structures.

The concept of a Submerged Floating Tunnel (SFT) reached my awareness during one of the MSc-courses on bored and immersed tunnels. Especially since it has never been constructed yet, I think it is a challenging concept. It is a typical structure which meets the different interfaces within my education so far. On the one hand it really touches upon the hydrodynamic aspect, for which knowledge on waves, currents and their interaction with structures is required. On the other hand, it includes the structural aspect as well. The SFT is still a 'futuristic' concept on which I hope to contribute to the eventual realization.

Although the construction of a SFT has never been constructed, the first thoughts on it originate from many years ago. Apparently, the structure is still too challenging from a technical and a cost-effective perspective. Despite the many thought experiments on a SFT, barely any literature is available on other cross-sections than a circle or a rectangle. Obviously both cross-sections have their benefits, but is it the optimal one? By understanding the current challenges in the design of a SFT, I hope to find 'the optimal cross-section'.
I started this research in September 2019 and now I can proudly present my results. In order to obtain these results, I would like to thank the committee members. First of all I would like Bas Jonkman for being my Chairman and guiding me through the process. Next, I would like to make a special thanks to Dirk Jan Peters and Marcel 't Hart, with whom this adventure started in the Summer of 2019. Next to coming up with the research proposal, they helped me a lot during the process. Moreover, I would like to thank my final committee member, Peter Eigenraam. I used software (Grasshopper) which I was not familiar with and Peter helped me understand it. Finally, I'd like to thank Pengxu Zou who was willing to set his CFD-model available in order to accurately analyze some of my results.

I also would like to thank my family and friends for the support throughout my entire education. I can look back at 6 amazing years of study, for which I am very grateful. Now I am looking forward to a new episode in my life where all this knowledge can be converted to actual 'Engineering'!

Abstract

Although the concept of a Submerged Floating Tunnel (SFT) originates from the early 1900's, the cross-section has always been assumed to be circular (or rectangular). In this research, it is investigated whether this assumption is valid. What is the optimal cross-section for a SFT? This optimization process is split into two targets. The first target aims for the optimization of material use, while the second target guarantees a tensile force in the tethers that moor the tunnel to the seabed or to pontoons at the surface.

It is not a coincidence that a circle is assumed to be the cross-section, since a circle is well known for its good performance in absorbing large (uniform) pressures. A SFT is located at a typical depth of 20m to 30m. Hence, large hydrostatic water pressures are expected. However, the water pressure is not uniform but increases linearly by depth. Therefore, a circle is not necessarily the optimal shape.

The concept of Form Finding (FF) is applied in order to find the optimal shape regarding minimal material use. Within the FF-process, the external loads are applied on a material without any bending stiffness. The output is a shape which absorbs these external loads axially. No shear forces or bending moments are present in the cross-sectional plane. In this way, all fibres of the material are used equally and optimally.

The FF-process is executed in 'Grasshopper', which is a programming environment within the 'Rhino' software. The optimal shape deviates most from a circle close to the water surface, because the relative pressure difference between the top and the bottom of the SFT is maximal here. The output converges to an 'egg-shape'. This 'egg'-shape is slightly more efficient, reducing the maximum normal force by 3.7%. For increasing depths, the relative pressure difference decreases. The optimal shape according to FF quickly converges to a circle.

Secondly, the optimal shape should generate a tensile force in the tethers. Since the steel tethers have no compressive strength, a compressive force in the tethers would result in large deformations of the structure. This is unacceptable in terms of both comfort and safety.

The tensile force in the tethers in current design thoughts is achieved by a conservative Buoyancy Weight Ratio (BWR), which means that the tunnel structure strongly wants to float (or sink). The tethers prevent this, resulting in a high level of pretension. This is unfavourable in terms of material use for both the tethers and the tunnel structure itself. Therefore, it is desired to keep the BWR as close to 1.0 as safely possible.

Next to the BWR, the hydrodynamic loads determine the signal of the tether force F_{tether}. Some more insight is given on the fluid-structure interaction between the tunnel structure and these hydrodynamic loads. The hydrodynamic loads considered are the currents and the waves, where the currents turn out to have the most significant interaction with the shape of the SFT. The (horizontal) drag force and the (vertical) lift force are evaluated by the cross-section.

A regular circle does not generate any lift force from the approaching flow, excluding the turbulent effects. This lift force can be expressed by the lift coefficient C_L, which is used to convert a flow velocity to a force. In general, $\overline{C_L} = 0$ for a circle. Moreover, the drag force on a circle is relatively large ($\overline{C_D} \approx 0.6 - 1.0$) and a lot of turbulence can be expected at the trailing edge due to its blunt shape. Hence, some space for improvement is visible here.

The horizontal approaching flow can generate a lift force in the desired direction, contributing to the tensile tether force and a BWR closer to 1.0. Bernoulli's theorem states that fluids with higher velocities have a lower static pressure. Therefore, a differential velocity profile between SFT-top and SFT-bottom is demanded to generate a lift force. An analogy with an aeroplane wing is made, where a more convexly shaped top yields a positive lift force (upward). These shapes are retrieved by Form-Finding in the same Grasshopper code. The output looks more like an ellipse, with a convexly shaped top and a flatter bottom.

Consequently, these shapes are evaluated with the Source Panel Method (SPM), which is coded in Python. The SPM solves for the velocity potential of a uniform around a submerged body. The submerged body is divided into a set of panels, where each panel has its own source/sink term λ. This source/sink term λ describes the geometrical influence of that panel on the flow characteristics at any arbitrary point. The output of this part is the flow velocity profile along the SFT, which can be converted to a pressure field and a lift force.

The cross-sections generated in Grasshopper can be imported into Python. The velocity and pressure field for any random geometry can now be solved for with the SPM.
The optimal shape regarding this target looks like an aeroplane wing, symmetrical over the vertical. When the SFT is anchored to the bottom and a positive lift force is desired, the top of the SFT is more convexly shaped. This results in larger velocities at the top and a lower pressure. The SFT is basically 'hanging' in the water flow. Moreover, the shape now appears more like an ellipse, reducing the turbulence effects and reducing the drag coefficient $\overline{C_D}$ to approximately 0.1.

The 'optimal shape' is based on two optimization targets with each its own optimal cross-section. One can choose to optimize for both targets simultaneously by constructing a composite cross-section with an inner concrete tube (circular) with a steel exoskeleton. The exoskeleton can be manually designed in Grasshopper. It has got a convex shape at the top of the cross-section, when anchored to the seabed. Moreover, the bottom of the exoskeleton is flat in order to generate maximum tension. This exoskeleton generates a lift force on the total structure, which is calculated by running xFoil from Python. xFoil also solves for the Panel Method, but is also able to deal with viscous flow for sharp edged structures. The lift coefficient can reach up to $C_L \approx 3.5$, which is a significant amount of lift. The water pressure is absorbed by the inner tube, constructed of concrete.
One can also choose for a compromise between the two optimization targets. This can be achieved by Form-Finding for both optimization targets simultaneously. This cross-section consists of a concrete tube only. The top is still more convexly shaped to generate a lift force, but it also has some appearances of a circle. The lift coefficient for this solution reaches up to $C_L \approx 0.6$.
To determine which of the two solutions presented above is 'the optimal one', a cost-benefit analysis must be executed. This depends per design situation.

In conclusion, with this research some insight is presented in the optimization of the cross-section of a SFT. Especially in terms of the reduction of drag and the generation of lift, the cross-section can be improved significantly. In further research, it is recommended to perform physical experiments with the cross-sections presented here.

Contents

List of Abbreviations

- BWR = Buoyancy Weight Ratio
- CFD = Computational Fluid Dynamics
- FF = Form Finding
- GF = Geometry Factor
- GH = Grasshopper
- HWP = Hydrostatic Water Pressure
- PM = Panel Method
- SFT = Submerged Floating Tunnel
- SPM = Source Panel Method
- SW = Self Weight

1

Introduction

1.1. Background

Imagine yourself being on one side of a fjord, looking at your destination at the other side. The crossing covers only a couple of 100 meters as the crow flies. The water in between you and your destination is the only obstacle. There might be a ferry which offers you and your car the crossing, but this can take a significant amount of time. Obviously, you can make the detour by following the road and driving all the way along the coastline of the fjord. But how efficient is it to drive all this way, whereas your destination is in clear sight? Usually, in these kinds of situations, a bridge would be present, or a submerged tunnel. However, as mentioned before, you are at a fjord. These natural phenomena are characterised by a rocky environment including steep slopes and deep ocean beds. The span of these coastal crossings can be quite large, which makes it unfeasible to construct a traditional bridge. The steep slopes and deep ocean beds result in the impossibility of constructing a submerged tunnel.

Therefore, a solution can be found in the form of a Submerged Floating Tunnel (SFT). This type of structure is also known as an 'Archimedes Bridge'. A SFT is a tunnel structure submerged in water, supported by its own buoyancy and another structure. As the name already suggests, it floats in the water column. It can be supported in two different ways. Firstly, the tunnel can be fixed to the bottom with tethers, which are connected to heavy anchors resting at the bottom (type A). The tunnel structure itself wants to 'float' in this situation, meaning that the SFT weighs less than the volume of displaced water by the SFT (Archimedes' Law). Secondly, the SFT can be heavier than the displaced amount of water, meaning that the structure wants to sink (type B). In this situation, the SFT is linked to large buoys at the water surface via tethers. For an overview of both types, see Figure 1.1.
A submerged floating tunnel is relatively expensive as well. However, a lot of others costs made by commuters, lorries and individual people can be saved if the alternative is to make the detour. Besides, the environment will benefit as well, since less exhaust gasses will vanish in the atmosphere. On the long term, a SFT can be a good solution from both an economic and an environmental point of view.

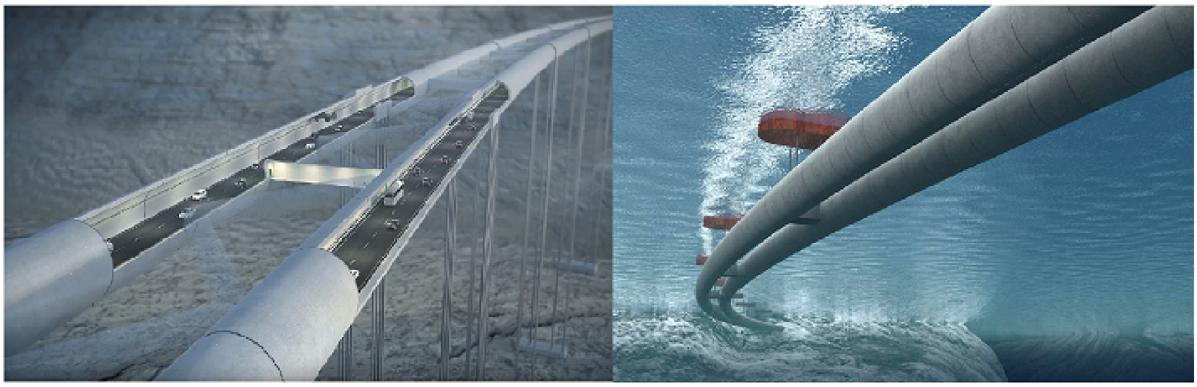


Figure 1.1: The SFT type A (left) [37] and type B (right) [28]

Either way, the tunnel is supported via tethers made of steel. Similar to a cable or a wire, a tether does not have any compressive strength. A compressive force, possibly generated by the flow or waves in the ocean, can easily result in large deformations of the tethers due to buckling effects. The occurrence of a compressive force in the tethers leading to buckling is a main failure mode of a submerged floating tunnel. [27].

The compressive forces in the tethers should therefore be avoided. In current designs, the tensile force in the tethers is guaranteed by adapting the buoyancy-weight ratio (BWR). For now, type A from figure 1.1 is considered. If the buoyancy force is much larger than the weight of the SFT, tension is present for all load situations. This situation can be achieved by enlarging the cross-section of the SFT. Main deficits of this solution are the larger cross-section than actually required and the large pretension in the tethers, being present all the time. Moreover, the structural stresses in longitudinal direction reach up to outrageous magnitudes, which is unnecessary. All of these negative effects can be prevented by looking at the dynamic interaction between SFT and hydrodynamic loads.

The shape of the tunnel can convert the flow into tensile forces in the tethers. Contrary to its compressive strength, the steel tethers do have a high stiffness related to tension. A large tensile force does not result in large deformations, meaning that the tunnel compartments stay relatively close to each other. The gasket doesn't fail.

Finally, an optimal shape for the cross-section is desired in terms of material use. Therefore, a second optimization issue is related to the internal force transfer which can reduce costs for material use. For both objectives regarding the optimization, see section 1.2.

1.2. Objective

Given the information and problem above, the objective of this research can be described as:

"Develop a parametric design method for the optimal shape of the cross-section of a Submerged Floating Tunnel, which:

1. *Converts the external hydrodynamic forces into dominant tension forces in the tethers*
2. *Optimizes the material use of the SFT."*

The objective consists of two parts. Both parts are related to an optimization problem.

The design should be parametric. This means that the optimal shape can be found for any situation. However, there is not one single solution, but a range of solutions to the problem. This is due to the presence of multiple parameters and multiple optimization targets as stated above.

Some sub questions need to be answered in order to achieve the objective of this thesis.

1. What are the existing types of cross-sections and what impact do the loads have on the system?
2. What loads are acting on the Submerged Floating Tunnel system?
3. What is the influence of the shape on the lift and drag forces in both a vertical and a lateral way?
4. How to guarantee a tensile force in the tethers, given the external hydrodynamic loads? (**1**)
5. How are the loads optimally diverted according to a Form-Finding method, which optimizes material use? (**2**)
6. Which cross-sectional shape is a compromise between the two optimization criteria **1** and **2** and what role does the BWR play?
7. How can both optimization criteria be met for one single situation?

1.3. Scope and limitations

Within the extent of this research, some simplifications must be made and some items must be excluded, in order to converge to a result in an acceptable amount of time. Therefore, the following items are excluded and/or simplified:

- The focus lies within the cross-section of the SFT. The BWR is assumed to be 1.0 at all times, to ensure that it is purely the hydrodynamic loads which cause tension in the tethers. Obviously, this benefits for the longitudinal direction as well. This is because the closer the BWR is to 1.0, the more economic the design can be.

- Only the permanent loads (Hydrostatic Water Pressure and Self-Weight) and hydrodynamic loads (Currents and Waves) are included in this research. The live-loads due to traffic are not included as well as the accidental loads due to earthquakes, anchor collision etc.

- This research does not consider the hydrodynamic loads on the tethers themselves. These are subjected and sensible to movements and vibrations as well. Here, the focus lies on the cross-section itself.

- Within the research, no probabilistic calculations are considered. Only the deterministic values are used to determine certain quantities. In an actual design, the failure probability must be accounted for accordingly by applying a probabilistic design.

1.4. Outline

In Chapter 2, an analysis is made where the existing design is first described. Then, the loads and forces due to these loads are listed and elaborated in paragraph 2.2 and 2.3. Then, the concept of Form Finding is briefly elaborated including its history. At the end of this chapter, the software applied is listed. In Chapter 3 the Grasshopper model is described which mainly focuses on the hydrostatic loads. This Chapter emphasizes the optimization of material use. Chapter 4 covers the optimization criterion which demands for tension in the tethers. First the currents are elaborated, which do determine the shape of the cross-section The waves do exert a load on the SFT, but are not taken into account within the Form-Finding process. Consequently, in Chapter 5, the previous two chapters are combined to come up with **the** optimal solution. Here, a distinction is made between one cross-section as a compromise between the two optimization criteria and a solution which fits both criteria well. The latter is a solution of a concrete SFT with an exoskeleton made of steel. In Chapter 6, a test case is worked out to give an impression on what the ideal cross-section of a SFT might look like. Finally, some conclusions and recommendations are made.

2 Analysis

In Chapter 2 an analysis is made to collect all the data and information to proceed with the actual research on how to optimize the cross-section for a SFT. First, 'the existing situation' is described, showing what the current developments are in constructing a SFT and what components are present. Secondly, a more extensive view is given on the loads acting on the tunnel structure. In paragraph 2.3 the interaction between these loads and the structure is elaborated. What are the forces to be expected and how to express them? Fourthly, some background and the definition of 'Form-Finding' is given. It is important to understand the underlying concepts before working with it. Then, the software used for this research is briefly listed and explained. Finally, some assumptions are shown on how the problem is simplified in order to be able to proceed with the research.

2.1. Existing situation

In this paragraph, the existing situation regarding the submerged floating tunnel is explained more extensively. What is the present situation regarding this revolutionary design? First 'the why' is described further, after which the support of the SFT is explained. Finally, the current ideas about a cross-section are shown.

2.1.1. Why a Submerged Floating Tunnel?

As mentioned before, the concept of the Submerged Floating Tunnel was created because of the demand of crossing a water body, which could not be crossed by a regular immersed tunnel or a bridge. This has got to do with the circumstances described above in Chapter 1. The crossing is too large and deep to make such a structure feasible. Such natural environments like fjords is what Norway is well-known for, for instance. To illustrate the issue here, see figure 2.1 [37].

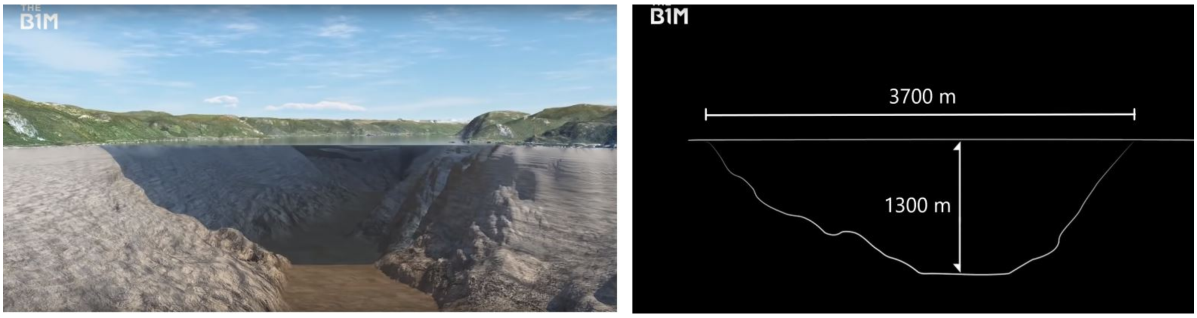


Figure 2.1: The longitudinal profile of the Sogne fjord [37]

A regular fjord can reach up to a depth of 500 m and a span between 250 m and 2500 m [12]. In figure 2.1 above, the so-called 'King of the Fjords' is shown. This is the Sognefjord in Norway with an incredible depth of 1.300 m.

The span is too large, while there is no possibility to support the bridge halfway. The column supporting the bridge would have to reach very deep into the ocean, being subject to all kinds of currents and wave loads.

Besides, due to the large length of the column, it would be very sensible to dynamic vibrations and to failure as a consequence. The large depth of such a fjord also disables the immersed and bored tunnel from being a good alternative. The steep slopes of the fjords often exceed the maximum allowed slopes for traffic. Especially lorries would experience a lot of difficulty climbing up 'the hill'. Moreover, the steep slopes would be unfavourable in terms of sustainability and traffic congestion as well.

Again, the submerged floating tunnel is the revolutionary solution to avoid these issues. The SFT is a structure floating at a certain immersion depth, which assures a minimal clearance to allow (navigation) vessels of passing safely [29]. In this way, the slope of the approach structure can be made within the safety and transport limits.

The demand for a SFT has got to do with both economic and environmental reasons. The transport of goods and people requires a crossing over the water body. If the alternative is to make the detour all the way around a fjord, a lot of time and fuel can be saved. Especially at locations where transport of goods is important, like a harbour, the trucks can save significant amounts of time and money. Moreover, the minimization of the fuel also reduces the amount of exhaust gases ending up in the atmosphere. Although the construction of the SFT also induces a lot of exhaust gases, on the long term it might be beneficial in environmental terms as well.

2.1.2. Anchoring the SFT - BWR

The submerged floating tunnel can be fixed in two different ways, as shown already in figure 1.1. The tunnel is permanently subject to the buoyancy force and to its self weight [29]. If the buoyant force exceeds the self weight, the SFT wants to float and is attached to the bottom with anchors and steel tethers. Contrary, if the self weight exceeds the buoyant force and the tunnel wants to sink, it is attached to the water surface with a set of pontoons and tethers. The main disadvantage of this variant is its sensibility to the waves' motions. The SFT also undergoes these motions. The disadvantage of the other variant, where the SFT is attached to the bottom, is the possible high mooring costs at a rocky seabed [11]. In current design thoughts on the SFT, these forces (BWR) should always have a significant net force to guarantee tension in the tethers.

The BWR determines the amount of pretension in the tethers. Since the tethers do not have any compressive strength, this pretension is very important. When no pretension would be present (BWR = 1,0), any wave or current might result in a compressive force in the tethers for a standard cross-section. Such a compressive force would result in 'dancing' of the SFT with fatal accidents as a consequence. Hence, current designs thoughts always have a $BWR \neq 1,0$. In terms of safety this is required. From an economic point of view, however, this is undesirable. The tethers are basically 'over-dimensioned', resulting in a lot of additional costs. This is explained further in paragraph 2.3.7.
Concluding, the guarantee of tensile forces in the tethers is in current designs achieved by 'over-dimensioning' the SFT, which costs a lot of money. These tension forces can also be achieved by designing the cross-sectional shape.

For an overview of the two possible anchoring methods, see figure 2.2

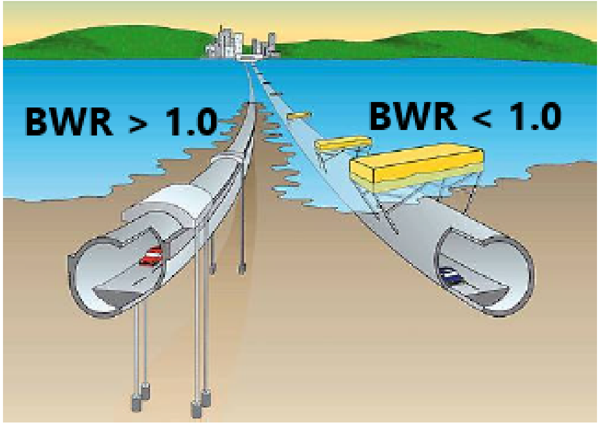


Figure 2.2: The two types of anchoring [35]

The current ideas already include a certain capacity in absorbing horizontal forces by constructing the cables under an angle (only visible at the right tunnel in figure 2.2). The tethers can only deliver a normal force and not a shear force. By constructing the tethers under an angle, the normal force also has got a horizontal component, which resists for the horizontal drag forces due to water flows. However, these loads can also have a vertical component due to the oscillating behaviour of waves for instance. This may result in compressive forces in the tethers and hence in large deformations. More information about these loads is given in paragraph 2.2.5. In figure 2.3 a vector overview is shown of the situation described above.

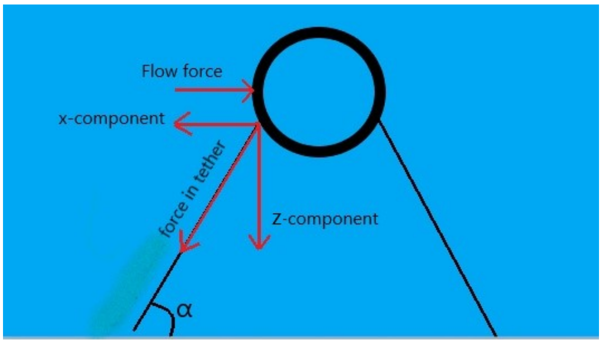


Figure 2.3: Horizontal capacity of tether under an angle

The smaller the angle α, the lower the normal force in the tether, but the longer the tether has to be. In this situation, a flow force with a downward vertical component can result in deformations of the tethers, depending on the BWR of the SFT.

From now, the focus is on the attachment type with anchors at the bottom of the seabed. **Concluding, the aim is to find a lift force being positive (upward).**

2.1.3. The Cross-Section

An SFT has to deal with high compressive forces due to the hydrostatic water pressure. A circular cross-section is very popular as it performs well in absorbing compressive forces. However, it is not the optimal solution regarding its 'hydrodynamics'. Later on, this is explained further.

Besides, the cross-section has to facilitate traffic and installations. A kind of flowchart is generated by Giulio [29], which shows an iterative process in defining the cross-sectional dimensions. First, the cross-section has to facilitate the traffic and installations, resulting in a minimal inner diameter. Accordingly, the total cross-section can be determined based on strength calculations, depending on the materials chosen and the loads present. The cross-section's volume results in a certain buoyant force, according to Archimedes' law. When this buoyant force is subtracted by the weight of the structure including ballast and live loads, a certain net force with a minimum upward value has to remain for a SFT anchored to the bottom. The exact value of this 'minimum' has never been determined yet. For an overview of this flowchart, see figure 2.4.

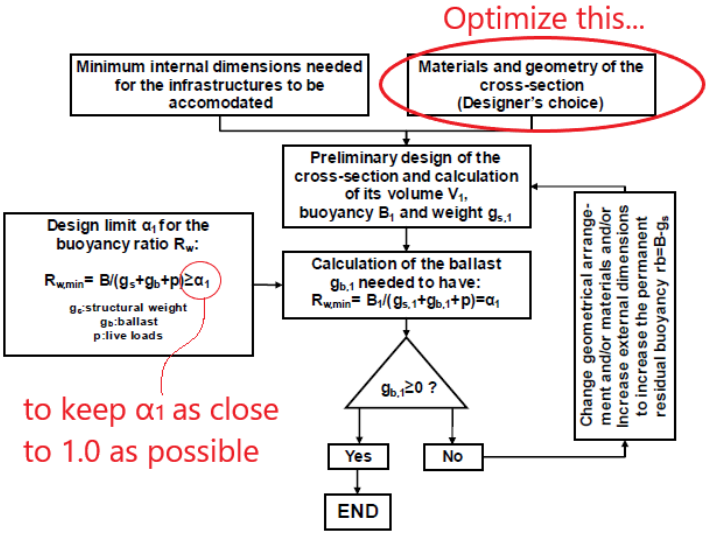

Optimize this...
Minimum internal dimensions needed for the infrastructures to be accomodated
Materials and geometry of the cross-section (Designer's choice)
Preliminary design of the cross-section and calculation of its volume V_1, buoyancy B_1 and weight $g_{s,1}$
Design limit α_1 for the buoyancy ratio R_w:
$R_{w,min} = B/(g_s+g_b+p) \geq \alpha_1$
g_s:structural weight
g_b:ballast
p:live loads
Calculation of the ballast $g_{b,1}$ needed to have: $R_{w,min} = B_1/(g_{s,1}+g_{b,1}+p) = \alpha_1$
$g_{b,1} \geq 0$?
Yes
No
END
Change geometrical arrangement and/or materials and/or Increase external dimensions to increase the permanent residual buoyancy rb=B-gs
to keep α1 as close to 1.0 as possible

Figure 2.4: Flowchart in preliminary design of cross-section [29]

With this idea in mind, a first impression of the cross-section can be made. As already mentioned, the cross-section should result in enough net buoyant force to prevent any compressive forces in the tethers. In order to do so, one could make a conservative design by increasing the safety parameter α_1, see figure 2.4. The buoyancy force would increase more than the weight of the structure, resulting in a larger upward net force. A downward external live load, due to waves or currents for example, would still not result in a compressive tether force. This means that the objective is reached, where only tensile forces are present. Hence, the tunnel structure would be stiff enough.

However, it would be optimal to keep safety factor α_1 from figure 2.4 as close to 1.0 as possible. The cross-section is, in that case, not unnecessarily large.

Types of Cross-Sections

Several types of cross-sections are possible for a SFT, according to Giulio [29]. This is mainly based on the facilitation of the traffic and the resisting capacity against the loads. In this research, the aim is to find an optimum shape of the cross-section to deal with the loads and to minimize deformations of the total structure. Before doing so, the existing thoughts on a possible cross-sectional shape are listed:

1. **Circular:** The advantage of a circular cross-section in SFT's is the ability to cope with the (compressive) hydrostatic water pressures [6]. The shape diverts the water pressures into (compressive) normal forces, while the bending moments in the cross-sectional plane are very small. Since concrete is often used for this type of structure and this type of environment, it is preferred to have limited bending moments. This is because concrete has a very small tension strength compared to its compressive strength. Cracks due to tension might endanger the water tightness of the tunnel.
   A disadvantage of a circular shape is the inefficient division of space. The space required to host the traffic has got a rather rectangular shape, which means that all the space between the circle and the inner rectangle is 'lost'.
   Finally, the circular shape does not guarantee a tension force in the tethers. If the flow is perfectly horizontal and the tunnel is perfectly circular, the net vertical force on the tunnel would be zero in theory. However, any deviation in either the cross-section or the flow might result in a compressive force in the tethers. Besides, the turbulent lift coefficient $C_L' \neq 0$ as becomes clear in paragraph 2.3.3. This coefficient might cause tension in the tethers as well.

2. **Rectangular:** The rectangular shape is the most efficient regarding use of space. As a matter of fact,

most immersed tunnels are rectangular. This mainly concerns situations where the external live loads are very limited, as well as the hydrostatic water pressure [29]. These immersed tunnels are often buried, meaning that the flows don't act as a load on the structure. If the SFT would have a rectangular shape, one can expect a lot of turbulent motions of the water flow resulting in a dynamic response of the SFT. Obviously, these motions are not preferred with respect to comfort and safety. This cross-sectional shape has the largest drag coefficient C_D.

3. **Elliptical in the horizontal direction elongated:** This shape looks like a compromise between the circular shape and the rectangular shape. It is already more efficient in terms of use of space. This also causes benefits regarding the material costs. However, the elliptical shape does not cope as well with compressive forces as the circular shape would. The outer shape still guides the flow along the tunnel, preventing heavy turbulent motions of the flow. The drag coefficient is lower, which becomes clear in paragraph 2.3.3.

4. **Double-tube circular with a streamlined cover:** This final type of cross-section is very similar to the first one, the circular shape. The main difference is the extra stiffness of the structure by connecting both tunnels to each other via a steel truss for instance. The whole set of tubes can also be covered by a larger structure, which should gradually divert the flow along the SFT. The cover should prevent the flow from turbulent behaviour.

See figure 2.5 for an overview of the types.

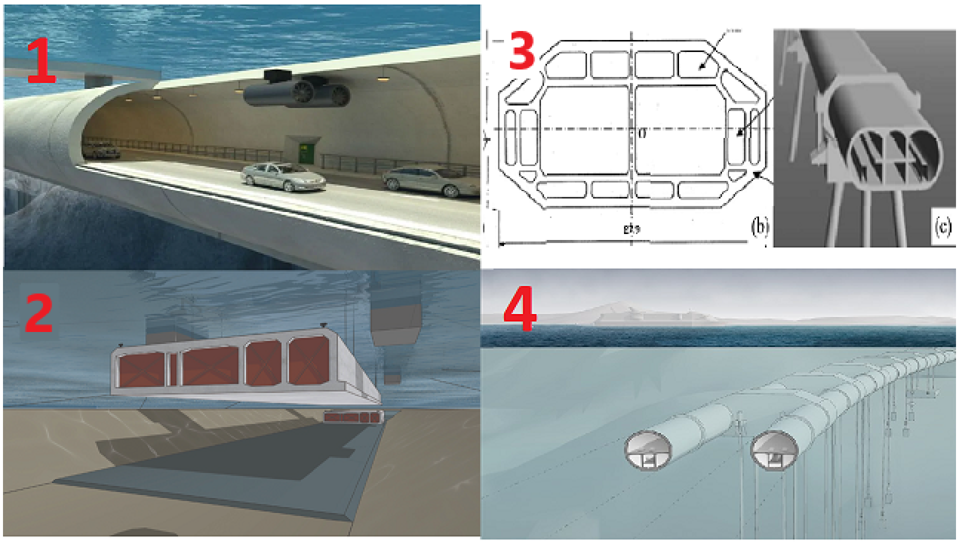


Figure 2.5: Different types of cross-section

2.1.4. Joints

Similar to regular immersed tunnels, submerged floating tunnels also consist of elements. It would be physically impossible to construct such a tunnel as a whole for both the fabrication and the transportation. However, it is possible to make the connections monolithic afterwards as planned in the Norwegian designs for fjord crossings. The elements are individually constructed and transported to the construction site. Consequently, immersed tunnels are connected via a so-called Gina gasket [33]. The Gina gasket is a smart solution in linking the individual elements, since it simply connects two elements by a large pressure difference. The gaskets assure the water tightness of the tunnel. In current design thoughts, the SFT is post-tensioned where the gaskets prevent water ingression. However, this is out of the scope of this research.

First, the element to be linked is brought in position. The water pressures left and right of the tunnel element are in equilibrium. Then, the water in the area between the two elements, the so-called bulkhead area, is pumped away. The water pressure at the outside (right) of the tunnel now pushes the element tightly to the previous element. The Gina gasket is not allowed to fail due to the 'dancing' of the SFT. Otherwise, the joint would fail and the structure would fill up with water and sink to the bottom.

2.2. Loads

In this paragraph, the loads acting on the submerged floating tunnel are enumerated. First, the permanent loads are described. These include the hydrostatic water pressure and the self-weight. Secondly, the loads due to wind and tidal currents are elaborated. Then, the wave loads are described and finally, some accidental loads are listed. In general, the permanent loads are noted with the symbol G and the variable loads are noted with the symbol Q.

2.2.1. Hydrostatic pressure (G)

The water imposes a permanent hydrostatic pressure on the tube of the SFT. This pressure depends linearly on the depth. The general formulation for the hydrostatic water pressure is:

$$p = \rho_{water} \cdot g \cdot d \tag{2.1}$$

Where

- p = hydrostatic water pressure $[\frac{N}{m^2}]$ or $[Pa]$
- ρ_{water} = fluid density $[\frac{kg}{m^3}]$
- $g = 9.81$ $[\frac{m}{s^2}]$
- d = water depth $[m]$

One should take a possible variation in the density into account for a stratified flow. For now, a uniform fluid density is assumed to be valid. The hydrostatic pressure induces a stress regime in the cross-sectional plane [29]. The load acts perpendicular to the structures' surface.
The water pressure also acts as a buoyant force on the SFT. This is elaborated further in the next section on the self-weight.

2.2.2. Self-weight (G)

The self-weight of the structure consists of multiple parts. First, the weight of the SFT itself obviously. But next to that, the structure is also subject to the weight of the road, the traffic and the required installations. From now on, this is called the functional self-weight. Finally, some marine growth can add up on the self-weight. However, due to the fact that the SFT is not at the seabed, it is neglected. At the seabed these loads can be of significant magnitude [42]. The total self-weight then becomes:

$$G_{total} = G_{self-weight} + G_{functional} \tag{2.2}$$

The functional self-weight differs for each SFT and its function. The self-weight per meter length of a circular SFT consists of the tunnel lining and the ballast. It can be calculated as follows:

$$G_{self-weight} = \rho_{concrete} \cdot g \cdot \frac{\pi \cdot (D_{ext}^2 - D_{int}^2)}{4} + G_{Ballast} \tag{2.3}$$

Where

- $G_{self-weight}$ = weight of the structure $[kN]$
- $\rho_{concrete}$ = density of concrete = 24 $[kN/m^3]$
- D_{ext} and D_{int} = External/Internal diameter $[m]$
- $G_{Ballast}$ = Ballast load $[kN]$

The ballast load $G_{Ballast}$ is the variable to be tweaked in order to reach a certain desired BWR. See figure 2.4 again.
As mentioned in the previous section, the buoyancy force has to be subtracted from the total self-weight. According to Archimedes' law, the buoyancy force is equal to the weight of the displaced amount of water:

$$G_{buoyancy} = \rho_{water} \cdot g \cdot \frac{\pi \cdot D_{ext}^2}{4} \tag{2.4}$$

For a total overview, see figure 2.6

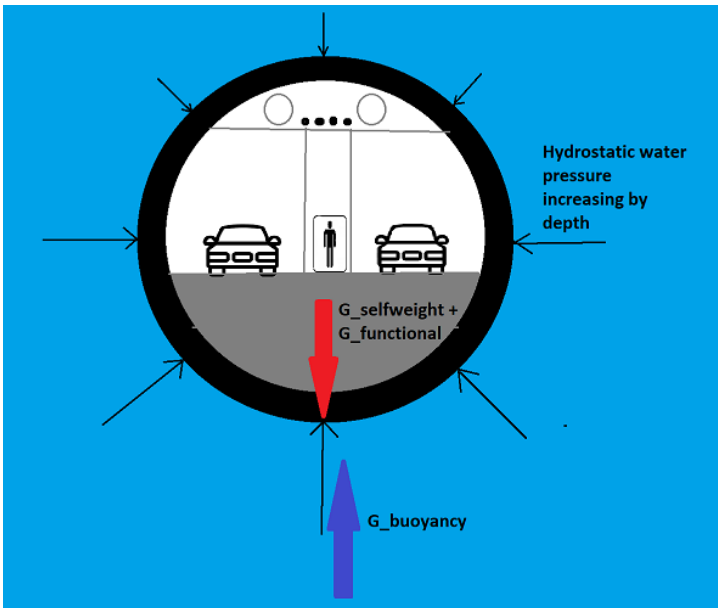

Hydrostatic water pressure increasing by depth
G_selfweight + G_functional
G_buoyancy

Figure 2.6: An overview of the permanent loads on the SFT

Note that the ratio between $G_{Self-Weight}$ and $G_{Buoyancy}$ is the earlier mentioned BWR and it can be reached by adapting $G_{Ballast}$.

2.2.3. Tidal currents (Q)

Tidal currents exist due to the interaction between the celestial bodies, like the earth, moon and the sun. Depending on the location on Earth, the tidal wave has got its own amplitude a and phase ϕ for each tidal constituent. Both diurnal (K1, O1) and semi-diurnal (M2, S2) tide constituents exist with each their own frequency ω [19]. The general form of such an harmonic wave is:

$$\eta(x,t) = a \cdot cos(\omega t - kx) \tag{2.5}$$

The actual tidal wave present at a location is the sum of multiple constituents. The tidal wave induces mainly horizontal currents which act on the SFT. The tidal waves have a relatively large period compared to wind waves for example. This means that variations in the flow velocity over time are relatively small. Therefore, one can model the tidal wave as a horizontal steady flow varying over depth. The deeper in the water column, the lower the flow velocities are. An analytical expression for the flow velocity at depth z is equal to [29]:

$$V_t(z) = V_T(0) \cdot \left(\frac{z+d}{d}\right)^{1/7} \tag{2.6}$$

Here, z is the vertical coordinate originating at the water surface and positive upward. Note that the final step is the conversion from flow velocity to an actual force. However, since this conversion is not as easy as for the permanent loads, a more extended view upon this conversion is given in the next section 2.3. The same holds for the wind currents, which are discussed next.

2.2.4. Wind currents (Q)

The wind blowing over the water surface also induces a current in the water column. Due to the difference in velocity and density between two fluids, a transfer of energy is generated [19]. Usually, the current induced by the wind speed is of more importance than the horizontal tidal flow velocity. However, it dampens out faster over depth than the tidal flow. Again, the current can be assumed to be horizontal with a dependence on water depth z. The contribution of the wind speed to the flow velocity at depth z is equal to:

$$V_W(z) = V_W(0) \cdot \left(\frac{z+d}{d}\right) \tag{2.7}$$

The total current velocity due to the tidal wave and the wind induced flow can now be calculated by simply adding up the two different components. An example of the velocity profile over a depth of 100 m looks as in figure 2.7.

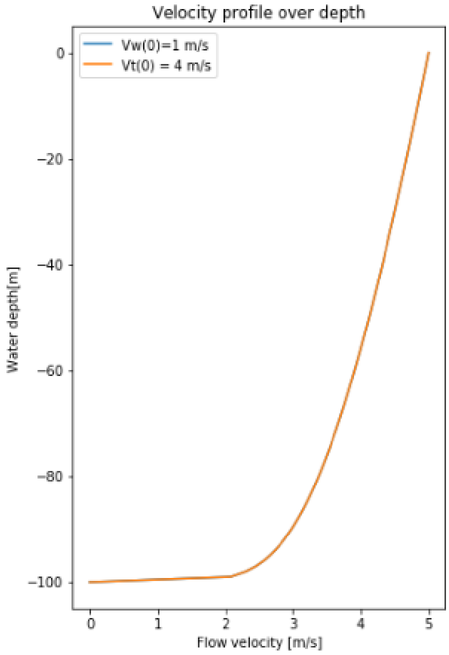


Figure 2.7: An example of the flow velocity profile over depth

2.2.5. Wave loads (Q)

Waves are the next external load on the SFT. Waves can have a significant impact on a structure and waves will always be present, either big or small. The tidal wave acts more or less like a current due to its large period, as explained before.

However, a lot of other types of waves exist. Mainly with a lot smaller period than the already mentioned tidal wave. Think of swell waves, seiches, wind generated waves, internal waves etc. Despite that all these types of waves have a different origin, they all have a similar behaviour. Waves have an oscillating character by definition, with a velocity potential parallel to its own motion [19]. The wave particles experience an orbital motion, which decreases in magnitude with increasing depth. See figure 2.8 left. The main question regarding this research is how the oscillating motion affects the SFT?

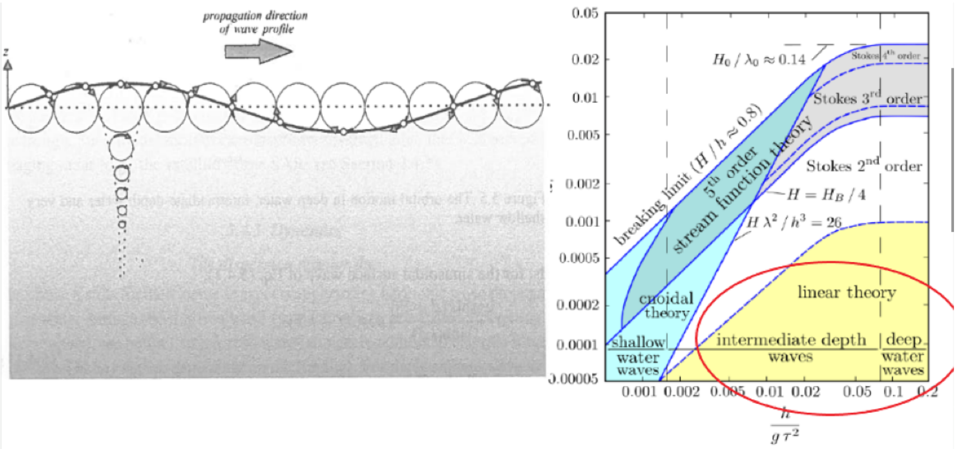


Figure 2.8: The orbital motion of a wave (left) and the validity of linear wave theory (right) [19]

In order to understand what effect the waves have on the structure, it is important to understand how the

waves themselves can be described. Linear wave theory describes the wave behaviour of surface gravity waves quite accurate in deep and intermediate water depth. See figure 2.8 right [25]. It is plausible to think that the SFT's are located in these wave regimes. To describe linear wave theory, some assumptions are required. The water is assumed to be incompressible with only the Earth's gravitation inducing the forces controlling the particles' motions. Given that waves are not too steep, it turns out this assumption is valid [19]. Besides, during the derivations it is assumed that the density is constant over space and time and that the water has no viscosity.
The start of the derivation starts, as often, from some balance equations. Namely, the *mass balance* equations and the *momentum balance* equation. For a volume box $\Delta x \Delta y \Delta z$, with the mass quantity ρ, the continuity equation can be derived as shown below.

$$\frac{\delta u_x}{\delta x} + \frac{\delta u_y}{\delta y} + \frac{\delta u_z}{\delta z} = 0 \tag{2.8}$$

In a similar way, the momentum balance equation can be derived. Note, the advective terms from the usual equation are removed because they are quadratic, whereas this is linear wave theory. Also, friction is ignored in the linear wave theory. The counteracting force to the momentum of the fluid is the pressure gradient. For x, y and z-direction the momentum equation becomes:

$$\frac{\delta u_x}{\delta t} = -\frac{1}{\rho} \cdot \frac{\delta p}{\delta x} \tag{2.9}$$

$$\frac{\delta u_y}{\delta t} = -\frac{1}{\rho} \cdot \frac{\delta p}{\delta y} \tag{2.10}$$

$$\frac{\delta u_z}{\delta t} = -\frac{1}{\rho} \cdot \frac{\delta p}{\delta z} - g \tag{2.11}$$

It is assumed the reader has the basic knowledge regarding these two balance equations, meaning no further explanation on this derivation is given.
These equations can be solved given some boundary conditions. First, it is assumed the waves have an infinitely long crest in the y-direction, meaning that the wave characteristics only depends on the x- and the z-coordinate. Furthermore, the particle velocity at the surface has to be equal to the speed of the water surface. Besides, the water particles at the bottom can not penetrate through the bottom itself. Finally, the pressure at the surface is equal to the atmospheric pressure.
This set of boundary conditions, together with the balance equations listed above, can be solved with the *velocity potential function*. The beauty of such a function is that only one function covers the velocities in any direction by taking the corresponding partial derivative. Concluding, the spatial derivatives of this velocity potential function $\phi(x, y, z, t)$ are the water particles velocity in the direction it is derived. For example, in the x-direction holds:

$$u_x = \frac{\delta \phi}{\delta x} \tag{2.12}$$

If this equation would be replaced in the continuity equation 2.8, the ***Laplace Equation*** is generated:

$$\frac{\delta^2 \phi}{\delta x^2} + \frac{\delta^2 \phi}{\delta y^2} + \frac{\delta^2 \phi}{\delta z^2} = 0 \tag{2.13}$$

When the boundary conditions are expressed in terms of ϕ as well, the Laplace equation can be solved. The analytical solution is a long-crested harmonic wave, propagating in the positive x-direction [19].

$$\eta(x, t) = a \cdot sin(\omega t - kx) \tag{2.14}$$

The velocity potential function belonging to the solution, describing the dynamic aspects of the wave behaviour is:

$$\phi = \hat{\phi} \cdot cos(\omega t - kx) \tag{2.15}$$

Where

$$\hat{\phi} = \frac{\omega a}{k} \frac{cosh[k(d+z)]}{sinh(kd)}$$

The following step is to make the move from the velocity potential function to the velocity motions in x and z direction. This can be done with equation 2.12 and equation 2.15.

The solution for u_x and u_z are harmonic functions as well, since the derivative of a harmonic function is a harmonic function itself. This follows from basic math and is in line with the expected orbital motion as shown in figure 2.8. Below, the total expression is shown for the flow velocities in x- and z-direction. The bold parts in equations 2.16 and 2.17 represent the amplitude of the harmonic function [19].

$$u_x = \omega \mathbf{a} \frac{\mathbf{cosh[k(d+z)]}}{\mathbf{sinh(kd)}} \cdot sin(\omega t - kx) \tag{2.16}$$

$$u_z = \omega \mathbf{a} \frac{\mathbf{sinh[k(d+z)]}}{\mathbf{sinh(kd)}} \cdot cos(\omega t - kx) \tag{2.17}$$

Now, there is a solution for the magnitude of the flow velocity due to ocean waves in x- and z-direction over time. This solution depends on the wave characteristics at the water surface, $a, \omega\ and\ k$, the local water depth and the depth of the SFT which is represented by z. These are all the variables likely to be of influence. The flow velocity u_y is of a negligible importance. It was assumed that the wave crests were infinitely long in the y-direction, resulting in zero y-velocities. Although a wave crest is not infinite in reality, a wave pattern usually has a significant length in the y-direction. This means that velocities in this direction are relatively small. Therefore, forces in this direction have less influence on the possible failure of the SFT. Furthermore, the SFT is likely to be located in a estuary. This means that the waves encounter the SFT perpendicular to its own longitudinal direction due to refraction mechanisms. Hence, this is another reason to safely assume that the wave velocities in y-direction are negligible.

Depending on the local water depth and the wave length/number at the SFT, the waves are propagating in deep water, shallow water or in intermediate water. For both deep and shallow water, the expressions for the amplitude of the wave velocities can be simplified. The condition for a wave to be in deep or shallow water is:

$$kd > 3\ for\ deep\ water$$

$$kd < 0.3\ for\ shallow\ water$$

When being in one of these two regimes, the expressions reduce to the following expressions below. This can make life a bit easier.

$$\hat{u_x} = \hat{u_z} = \omega a e^{kz}\ for\ deep\ water \tag{2.18}$$

$$\hat{u_x} = \frac{\omega a}{kd}\ and\ \hat{u_z} = \omega a(1 + \frac{z}{d})\ for\ shallow\ water \tag{2.19}$$

An overview of the situation is shown in figure 2.9 below. Here, the **velocities** of the orbital motions are shown in the figure. Note that these velocities need to be converted to forces yet. Besides, the governing situation must be searched for. The total motion can be analysed with the help of Matlab as executed by Hemel [18]. This is further elaborated in Chapter 4, section 4.6.

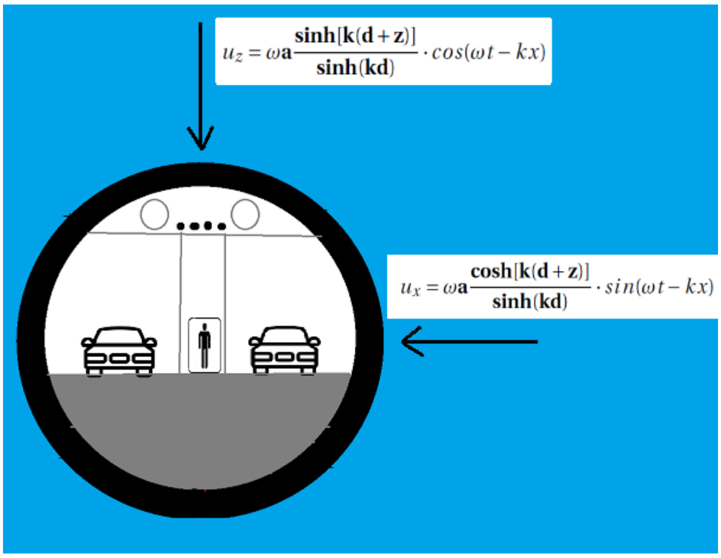

$u_z = \omega \mathbf{a} \frac{\mathbf{sinh[k(d+z)]}}{\mathbf{sinh(kd)}} \cdot cos(\omega t - kx)$
$u_x = \omega \mathbf{a} \frac{\mathbf{cosh[k(d+z)]}}{\mathbf{sinh(kd)}} \cdot sin(\omega t - kx)$

Figure 2.9: The wave velocities acting on the SFT

Similarly as for the current loads, the final step is the conversion from flow velocity to force. Again, this is a rather complicated process which requires some more background. This is given in section 2.3.

2.2.6. Anchoring force

The SFT is attached to the surroundings by tethers, being either the water surface or the seabed. Note that the anchoring force is the support reaction from the SFT to the surroundings. Therefore, it is not a typical load on the structure. The attachment of these tethers to the structure thus induce a support reaction for a BWR $\neq 1.0$. Although this research focuses less on the internal forces within the longitudinal direction, it is good to already think about them. The anchor suppport reaction in combination with the BWR always results in bending and shear. This is explained further in paragraph 2.3.7.
The attachment is via a hook constructed in the concrete of the SFT, see figure 2.10. This anchoring force is at a certain angle and when optimizing this angle, the tether force is within the 'centroid axis' or neutral line of the cross-section. This would mean that the tether force does not induce any bending moments in the cross-section, except for the (small) effect of eccentricity of the hook.

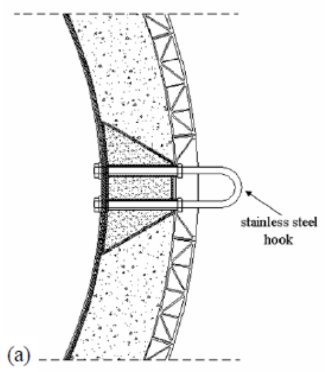


Figure 2.10: The attachment of the tethers to the SFT [29]

If the angle α at the location of attachment is equal to the tangent of the cross-section at that location, one can assume the anchoring force does not induce significant internal tension forces. This is not investigated further within this research, but just a side note.

2.2.7. Accidental loads (Q)

Next to the loads which are permanent (G) or variable but likely to be present, there are also accidental loads. These loads are, contrary to the current or wave loads, unlikely to happen. However, despite the small probability of occurring, the consequences of the accidental loads are usually quite large. Since the risk is the multiplication of the probability and the consequences, the risk of the accidental loads can be significant. Some examples [42]:

- Tsunamis
- Earthquakes
- Collision with a submarine
- Dropping anchor
- Sinking ship

The impact of such an accidental load is not accounted for as this goes beyond the scope of this research. However, it is important to be aware of the presence of these types of loads in further research and design.

2.3. Fluid-Structure interaction

As mentioned above, the hydrodynamic loads are known and expressed in terms of a velocity. Next, they have to be converted to actual forces. This requires some additional information which is shown in this section. First of all, an overview is given on different types of flows because they also result in different forcing. Then, the actual forcing is described for the already known velocities from section 2.2.

2.3.1. Types of flow

Different types of flow exist which have a significant influence on the forcing of the flow on a submerged structure. The type of flow is often characterised by the dimensionless 'Reynolds Number' [13]. This number represents the ratio between the inertia forces and the viscous forces, which can be seen as the load versus the resistance respectively. The inertia force is generated by the momentum of the fluid mass, encountering the submerged structures. The viscous force acts as resistance due to the cohesion between the fluid particles itself. In formula, the Reynolds number looks like [4]:

$$Re = \frac{u \cdot D}{\nu} \tag{2.20}$$

Where:

- Re = Reynolds Number [-]
- u = flow velocity [m/s]
- D = Characteristic length object [m]
- ν = Kinematic viscosity = dynamic viscosity η/mass density ρ $[m^2/s]$

The magnitude of the Reynolds Number determines if the flow is called laminar or turbulent. The higher the Reynolds Number, the more turbulent the flow is. In general, a flow is called turbulent if $Re > 1000$.
A laminar flow has got a clear stratification. The fluid particles basically only move in direction of the streamlines, meaning that there is no exchange of momentum perpendicular to the direction of the flow. This is a rather simple type of flow to describe, see the bottom part of figure 2.11. For a turbulent flow it is more complex to describe its behaviour. The water particles start moving perpendicular to the streamline and they separate along the body. See figure 2.11 for an illustration.

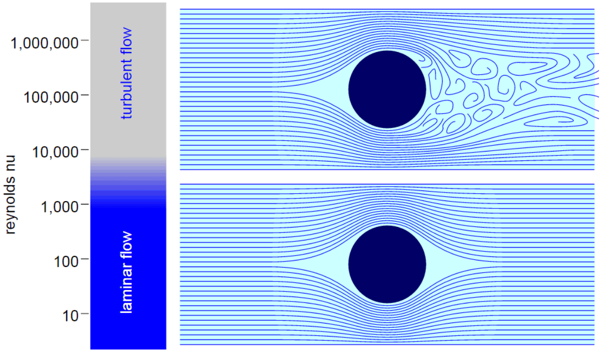


Figure 2.11: The laminar and turbulent flow [7]

The location where the flow separates from the body depends on the Reynolds number for curved cross-sections. Small differences in flow velocity can therefore result in different locations of separation. This causes an unsteady flow. An object with sharp edges has got its separation at the location of the edges by definition. Concluding, sharp edges are preferable regarding the prevention of an unsteady flow. In the next section, the turbulent behaviour is explained further.

2.3.2. Flow separation and vortex induced vibrations

Vortex induced vibrations are motions exerted by the fluid that interacts with a submerged body. Fluids have, in reality, always a certain amount of viscosity. In case of a relative velocity/acceleration between the fluid and the submerged body, this viscosity slows the fluid down along the surface of the body. The area, or length, over which the fluid is slowed down but still attached to the body, is called the boundary layer. Over this area, the pressure increases. From a certain point, the curvature of the body is too large for the fluid to follow it and the flow separates from the body. Now, the wake region is entered, where vortices are originated. See figure 2.12 [11]. The part after the separation point is called the 'wake region'.

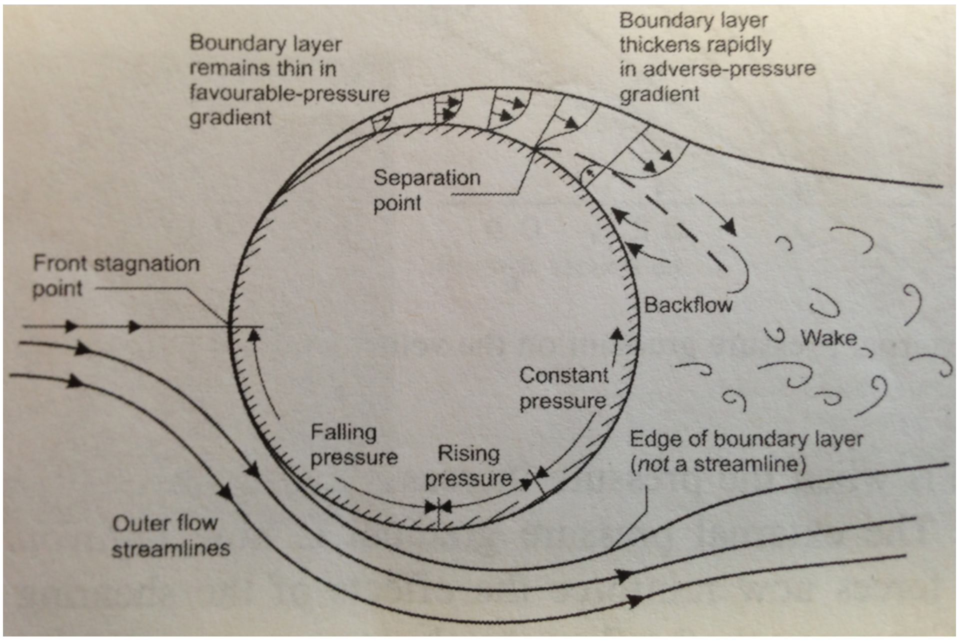

Boundary layer remains thin in favourable-pressure gradient
Boundary layer thickens rapidly in adverse-pressure gradient
Separation point
Front stagnation point
Backflow
Wake
Constant pressure
Falling pressure
Rising pressure
Edge of boundary layer (*not* a streamline)
Outer flow streamlines

Figure 2.12: The turbulent flow around a submerged body [17]

At the lee side of the submerged body, an adverse pressure gradient is present which can also be seen in the figure above. This causes a shear layer in the wake region and the boundary layer to separate from the body [38]. The vorticity in the boundary layer is transported to the shear layer. Then this layer starts to 'roll up' and a vortex field is generated in the wake region. Again, this is visible in figure 2.12. These vortices can have a significant impact on the forcing at a submerged body. Especially for a cylindrical shape, which doesn't have sharp edges. This is because the separation point is not predetermined. The flow separates at several locations, also causing vortices at several locations. In its turn, the vortices exert a (lift) force on the body. Since the vortices are not necessarily symmetrically distributed over the mid-line of the cross-section, differences in lift forces are likely to occur. This can cause unsteady forcing on the body resulting in (large) vibrations. Therefore, they are called vortex induced vibrations [32].
Concluding, a horizontal flow over a symmetrical submerged body can still cause (vertical) lift forces due to a turbulent flow. These forces are expressed with a turbulent lift coefficient C_L'. This is further explained in paragraph 2.3.3.

The turbulence at the lee side of the submerged body must be translated into a force, which depends on the flow that encounters the body in the first place. Basically, an analogy can be made with the wind loads for (high-rise) buildings. The 'suction force' at the lee side of a building due to an entering wind speed is modelled as an additional component being equal to a factor times the actual load. It is assumed to be correct when applying this analogy for fluids. This is required to model asymmetrical flow, as becomes clear in Chapter 4. The next question is what this factor should be. For wind loads in a similar situation, the load factor for the 'suction force', or the turbulence in this case, can be determined as follows [36].

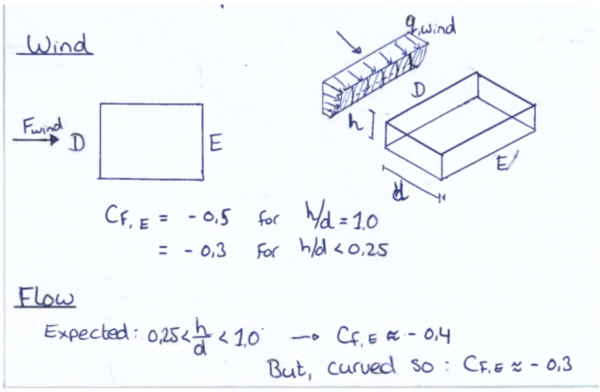

Wind
q_{wind}
D
h
E
d
F_{wind}
D
E
$C_{f,E}$ = -0,5 for h/d = 1,0
= -0,3 for h/d < 0,25
Flow
Expected: 0,25 < $\frac{h}{d}$ < 1,0 → $C_{f,E}$ ≈ -0,4
But, curved so : $C_{f,E}$ ≈ -0,3

Figure 2.13: The VIV-factor for wind

The expected ratio between the height of the SFT 'h' and the width of the SFT 'd' is expected to be in between 0,25 and 1,0, meaning that a VIV-factor of -0,4 can be assumed. However, the cross-sectional shape of the SFT is also expected to be more curved, meaning that the 'suction force' is expected to be lower. Finally, the VIV-factor is set to -0,3 and assumed to be valid. Again, this is only required for asymmetrical flow. The focus within this research lies on symmetrical flow, but an annotation on asymmetrical flow is given in Chapter 4 and Appendix C.

2.3.3. Flow to force for currents

The flow exerts a force on the submerged structure, because the flow has to curve around it. The directional change of the flow requires a force from the body on the flow, Then, according to Newtons' 3rd law, the flow also excites a force on the body. This force can be decomposed in a component perpendicular to the body's surface and a component parallel to the body's surface. The component perpendicular to the surface acts as a pressure gradient p, similar to the hydrostatic water pressure. The component parallel to the surface acts as a friction component τ_0. The direction of these two forces depends on the location along the body with angle Φ. See figure 2.14.

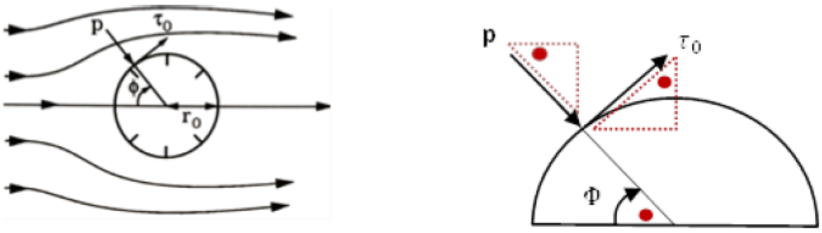


Figure 2.14: The forces perpendicular and parallel to a body [38].

Especially this pressure force is important regarding the Form Finding process, which is elaborated further in section 2.4.

Then, the forces on the SFT have to be expressed. In order to do so, some general empirical formula's are known. The force in the horizontal direction is called the drag force, whereas the vertical component is called the lift force. These two forces are usually expressed as follows:

$$F_{Drag,x} = \frac{1}{2} \cdot \rho \cdot D \cdot U^2 \cdot C_D \tag{2.21}$$

$$F_{Lift,z} = \frac{1}{2} \cdot \rho \cdot D \cdot U^2 \cdot C_L \tag{2.22}$$

The main deficit of these empirical formula's are the unknown parameters C_D and C_L which must be determined experimentally. These values depend, among others, on the shape of the cross-section And since the shape of the cross-section is highly variable during this research, it would be unfeasible to execute experiments for all shapes found. Therefore, another method is introduced in Chapter 4. For now, it is a good first

estimation to set C_D to 1.0.
For both equations, the velocity U represents the component in the direction of the force. Since the flow is assumed to be horizontal, as stated in paragraph 2.2.3, the velocity U only has a horizontal component. For a symmetrical cross-section over the mid-line and a perfectly horizontal current, the lift force would be equal to zero. However, due to asymmetrical instantaneous forces the net lift force F_L is not equal to zero. [11]. See figure 2.15 for a sketch of a graph with the development of C_D and C_L over time. The C'_L coefficient represent the turbulent behaviour for the lift force over time as described in paragraph 2.3.2. The mean lift coefficient $\overline{C}_L$ is equal to 0, since the encountering flow is horizontal.

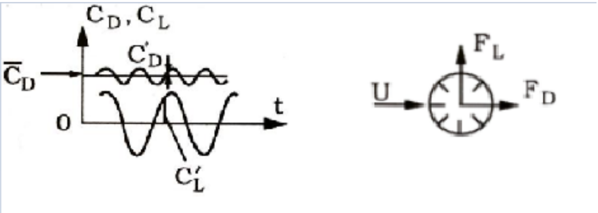


Figure 2.15: The drag and lift coefficients over time [38].

The drag coefficient also consists of a mean part $\overline{C}_D$ and a turbulent part C'_D. The mean part is here not equal to 0, because the flow is in the direction of the drag force. To get some insight in the magnitude of the mean coefficient $\overline{C}_D$ for different cross-sections, see figure 2.16.

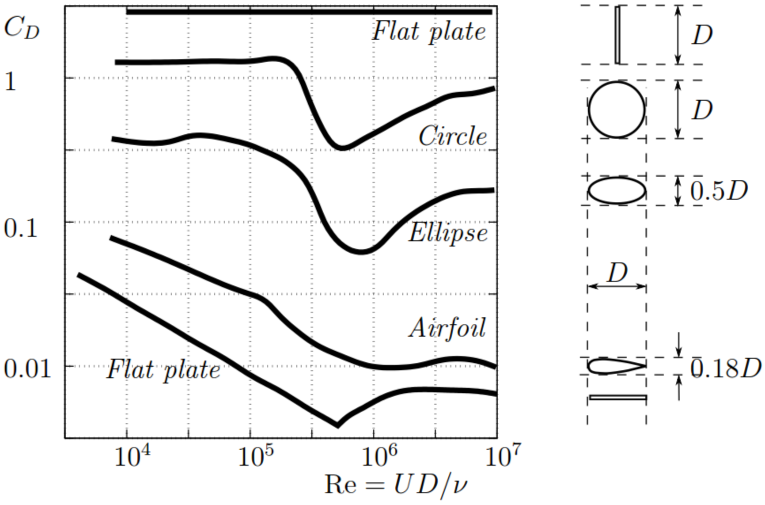


Figure 2.16: Experimental results for C_D for different cross-sections.

Note: the mean lift coefficient $\overline{C}_L$ is equal to 0 for a horizontal flow for all cross-section shown above. This is due to their symmetry over the horizontal. The aim of this research is to find a lift coefficient larger than 0 for a horizontal flow.

The starting point in Form Finding for the cross-sectional shape is a circle. A quick estimation of the Reynolds number for a circle with $D = 15\ m$ and flow velocity $U = 2\ m/s$ yields:

$$Re = \frac{U \cdot D}{\nu} = \frac{2.0 \cdot 15}{1.787 \cdot 10^{-6}} = 1.7 \cdot 10^7$$

Resulting in a drag coefficient about equal to 1.0 for a circle. See figure 2.16. This value is used within the Form-Finding process, which becomes clear in Chapter 4.

Now, the force from the current on the SFT can be expressed. In figure 2.17, a random segment of a discretized circle is shown. This segment has got an angle equal to β with respect to the horizontal.

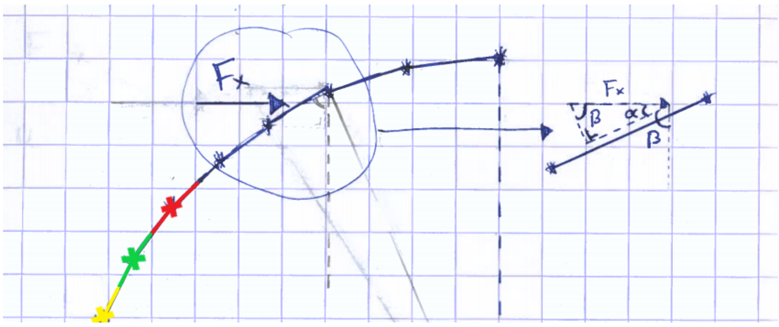


Figure 2.17: Locations of angle β

The next step is to decompose the horizontal force $F_{Drag,x}$ into a component perpendicular to the segment p and a component tangent to the segment τ according to figure 2.14. The tangential component does not influence the deformations of the SFT in the FF-process. The perpendicular component however, does. This perpendicular component is therefore inserted in the FF-model and can be calculated as:

$$p = F_{Drag,x} \cdot cos(\beta) \tag{2.23}$$

The current load is now expressed in a force which is used within the Form-Finding process. To determine the actual lift force on the structure, a more advanced method is applied as becomes clear from Chapter 4 and Appendix E.

2.3.4. Flow to force for waves

Here, the additional step is made to convert the oscillatory wave motion to a force on the SFT. Remind that the oscillatory motion of the wave was already explained in paragraph 2.2.5.
The behaviour of a wave is different from the behaviour of a (steady) current. Therefore, the conversion from velocity to force is different as well. The behaviour of a wave is an oscillatory motion, see figure 2.8. In order to determine the force due to a wave, a semi-empirical function was developed by Morison [30]. Sometimes the equation is called the 'MOJS equation', named after all the authors Morison, O'brien, Johnson and Schaaf [34]. This function consists of both a drag force, similar as in equation 2.21, and an inertia force. These two parts basically represent the force exerted by the velocity and by the acceleration, respectively. It yields the force in the in-line direction of the oscillatory flow.

Drag force component
The drag force component is similar for waves as it was for currents. The formula for the drag force component can be seen in equation 2.21.

Inertia force component
The inertia force component consists in its turn also of two parts. The Froude-Krylov force and the hydrodynamic mass force.
The Froude-Krylov force is caused by the dynamic pressure field, related to the undisturbed flow. This pressure field would have accelerated a certain amount of water which would be displaced by the submerged body [29]. For an overview, see figure 2.18 [11]. This dynamic pressure field can be expressed in terms of the acceleration of the flow:

$$\frac{dp}{dx} = \rho \cdot \frac{du}{dt}$$

When multiplying this pressure field with the total cross-sectional area A of the submerged body, the total force is:

$$F_{Krylov} = \rho AU'$$

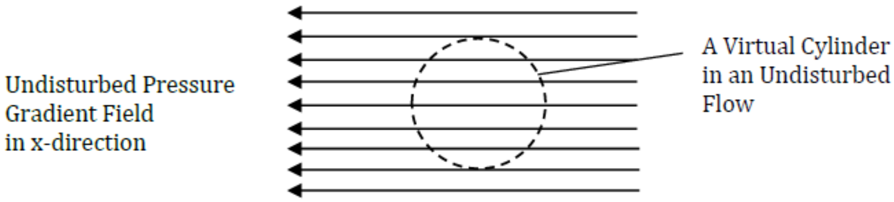


Figure 2.18: An undisturbed flow with a horizontal pressure field [11]

The second contribution is due to the hydrodynamic mass component, or also called the 'added mass contribution'. The trajectory of the water particles is deviated due to the presence of the submerged structure. This generates again a dynamic pressure field. The resultant of this field yields the hydrodynamic mass component. The physical meaning of this contribution might be hard to understand, but one can understand it better if the process is reversed [18]. If the submerged body was to accelerate in static water, the submerged body would have to accelerate some water particles as well in order to pass them. The additional amount of water particles to be displaced by the accelerated body is accounted for as 'added mass' and is expressed with a factor as shown in the formula below [9]. In formula:

$$F_{Hydro} = m' * \frac{du}{dt}$$

Where:

- $m' = hydrodynamicmassperunitlength = \rho \cdot C_a \cdot A$ [kg/m]
- C_a = Coefficient for added mass [-]
- A = cross-sectional area of submerged body [m^2]

These two components can be added up and together they form the inertia force. Below the complete Morison's formula is shown which can be used to calculate the total force due to waves.

The Morison's Formula

Now the two contributions to the Morison's Formula are known, the total equation can be composed as shwon below:

$$\frac{dF}{dz} = p(t,z) = p_I + p_D = C_I \cdot \rho \cdot A \frac{du}{dt} + C_D \cdot \frac{1}{2} \rho \cdot D \cdot U|U| \tag{2.24}$$

Where:

- p(t,z) = force on object per meter length at location z [N/m]
- p_I = Inertia force component [N/m]
- p_D = Drag force component [N/m]
- C_I = Inertia coefficient = $C_a + 1 \approx 2$ [-]
- C_D = Drag coefficient ≈ 1.0 for low flow velocities [-]
- u = horizontal velocity of water particles [m/s]
- $\frac{du}{dt}$ = horizontal acceleration of water particles [m/s^2]

The horizontal acceleration of the water particles hadn't been derived yet. However, given the expression for the horizontal velocity as in equation 2.16, it is an easy step to derive the term $\frac{du}{dt}$ by differentiating this equation with respect to time. The horizontal acceleration term then becomes:

$$\frac{du}{dt} = \omega^2 a \frac{cosh[k(d+z)]}{sinh(kd)} \cdot cos(\omega t - kx) \tag{2.25}$$

Similarly as for the flow load, note that values approximated for the coefficients C_I and C_D are an estimation. In reality, these two coefficients should be determined by experiments and they depend on the surface roughness, the Reynolds Number and the Keulegan-Carpenter number. This last term is a dimensionless number describing the relative importance for the drag force over the inertia of the oscillatory flow [22]. In formula:

$$K_C = \frac{VT}{L} \tag{2.26}$$

Where:

- V = amplitude of the flow velocity oscillation [m/s]
- T = period of the oscillation [s]
- L = characteristic length of the object [m]

Still, the values as approximated and shown in equation 2.24 are considered to be conservative and are used. The total force in horizontal direction can now be calculated in preliminary design situations with [11]:

$$F_x = \underbrace{\rho C_I A \frac{du_x}{dt}}_{Inertia\,component} + \underbrace{\frac{1}{2}\rho D C_{D,x} U_x \sqrt{U_x^2 + U_z^2}}_{Drag\,component} \tag{2.27}$$

Lift force
Now, the in-line force on the submerged body for an oscillatory flow can be calculated. However, similar to the forces due to currents, a lift force is also exerted on the submerged body. This is the force perpendicular to the direction of the wave. In most cases, this force is of the same order of magnitude as the drag force and it is generated by fluctuations in the vortexes around the submerged body [30]. Especially regarding this research, the force perpendicular to the wave direction is important since this force can generate a compressive force in the tethers. For preliminary design, this force can be approximated by:

$$F_z = \underbrace{\rho C_I A \frac{du_z}{dt}}_{Inertia\,component} + \underbrace{\frac{1}{2}\rho D C_{D,z} U_z \sqrt{U_x^2 + U_z^2}}_{Drag\,component} \tag{2.28}$$

Where:

- $C_{D,z}$ = C_L = Lift coefficient [-]

The magnitude of C_L has to be determined by experiments as well. However, for preliminary design one can say that $0 < C_L < C_D$. Hence, a conservative design assumes C_L being equal to C_D. For a circle, $C_D = C_L = 1.0$ as just concluded.

2.3.5. Net force on the structure

The shape is highly variable during this research, meaning that the coefficients C_D and C_L are as well. Hence, the determination of the net force on the structure is not as easy as it seems according to the equations above. This holds especially for the hydrodynamic loads, which must be quantified in order to assess the optimization criterion which demands for a tensile force in the tethers. Besides, the curved flow profile along the submerged body causes local velocity increases. The uniform flow velocity is not applicable everywhere along the SFT.

The net forces are determined based on the pressure distribution within the fluid with the help of Bernoulli's principle. This principle states that the energy along a streamline is constant and can be expressed by:

$$H = z + \frac{p}{\rho g} + \frac{u^2}{2g} \tag{2.29}$$

Where:

- H = Energy height [m]

- z = Local height [m]
- p = Pressure [Pa]
- u = Flow velocity $[m/s^2]$

A higher flow velocity causes a pressure drop in the fluid, since the total energy height [H] must remain constant along the streamline. In order to solve for the velocity (and pressure) field along the SFT, the Panel Method can be applied. This is a method solving for the velocity potential by modelling a geometry as a set of panels with each its own source/sink terms [15]. This principle is further elaborated in Chapter 4 and specifically in section 4.1.

2.3.6. Overview of loads

An overview of all loads acting on the SFT are graphically shown in figure 2.19. Obviously, more forces are present during the lifetime of a SFT. Think of the accidental loads as described before. However, for this research these are the main loads which are considered.

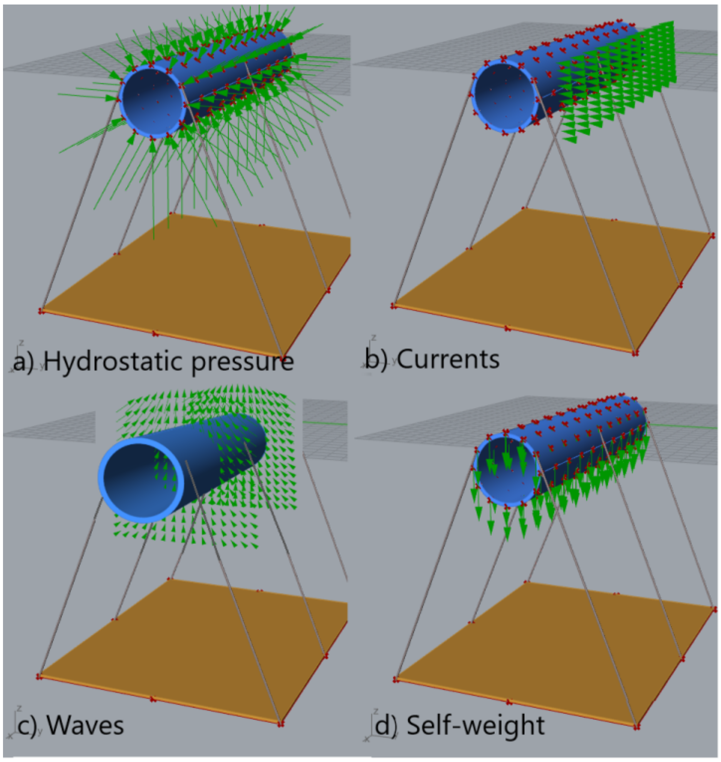


Figure 2.19: The main loads acting on the SFT

For the currents, an additional component at the lee-side of the structure is present as explained in paragraph 2.3.2. This component is not shown in the figure above. Moreover, note that the structure is segmented in order to model the loads.

- Hydrostatic water pressure: $F_{HWP} = \rho \cdot g \cdot z \cdot A$
- Currents: $F_{x,D} = \frac{1}{2} \cdot \rho \cdot D \cdot U^2 \cdot C_D$
- Waves - x: $F_x = \rho C_I A \frac{du_x}{dt} + \frac{1}{2}\rho D C_{D,x} U_x \sqrt{U_x^2 + U_z^2}$
- Waves - z: $F_z = \rho C_I A \frac{du_z}{dt} + \frac{1}{2}\rho D C_{D,z} U_z \sqrt{U_x^2 + U_z^2}$
- Self-Weight: $G_{total} = \rho_{concrete} \cdot g \cdot \frac{\pi \cdot (D_{ext}^2 - D_{int}^2)}{4} + G_{Ballast} + G_{functional}$

2.3.7. Tether forces

The required tension force in the tethers is in current designs achieved by a $BWR \neq 1,0$. In case when the SFT is anchored to the bottom, the BWR should be larger than 1,0 for instance.

In figure 2.20 three situations are shown for the tether force F_{tether} over time. In figure 2.20a, the situation is shown for a regular circular cross-section with a BWR equal to 1,0. The force signal enters the compressive zone due to the wave load or due to the turbulent force from the currents, as just described. In figure 2.20b, the situation is sketched for current design thoughts. The BWR > 1.0, meaning that a certain level of pretension is present. The buoyancy exceeds the weight, resulting in a structure that strongly wants to float. Although the waves and currents push the cross-section towards the compressive zone, the net result is still a tensile force in the tethers. Objective achieved, one might say. However, the 'level of pretension' is not economical as explained in the next part. Figure 2.20c is the objective of this research. A BWR equal to 1.0, where the external hydrodynamic loads cause tension in the tethers based on the fluid-structure interaction.

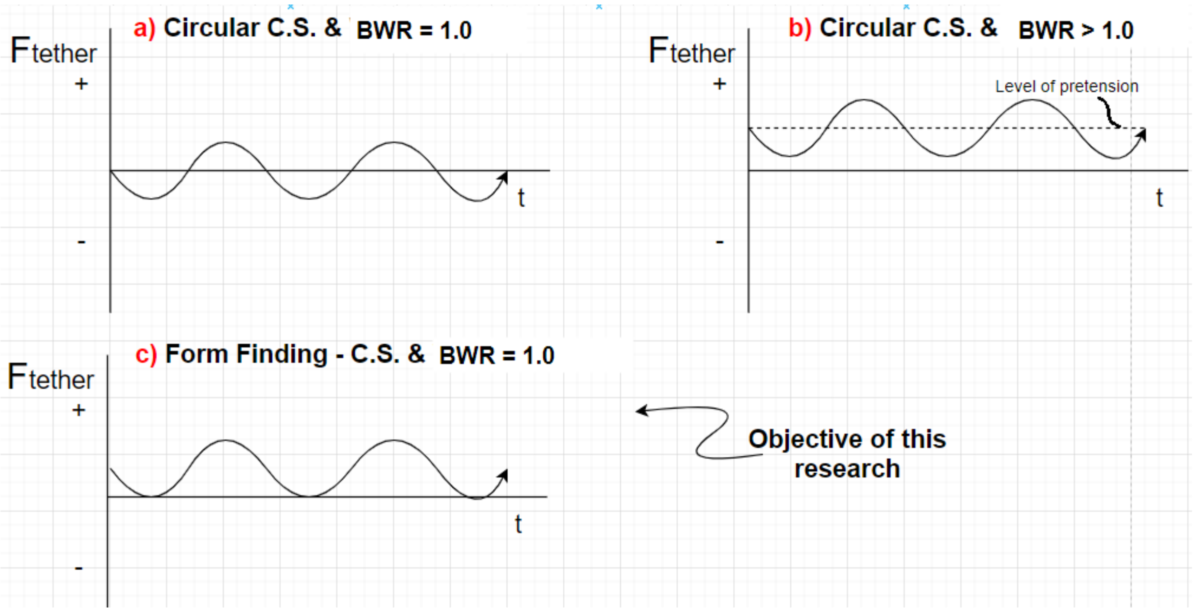


Figure 2.20: The forces in the tethers for different situations over time.

A very brief analysis is executed to show the importance of a BWR close to 1.0. The current design thoughts include a BWR equal to 1.3 or $\frac{1}{1.3}$, depending on the anchoring method. Assuming a circle with radius $R = 7.5\ m$ and the corresponding BWR equal to 1.3, the net upward water pressure is equal to:

$$Buoyancy = \pi \cdot R^2 \cdot \gamma_w = \pi \cdot 7.5^2 \cdot 10 = 1767\ kN\ per\ m\ length$$

$$Weight = \frac{Buoyancy}{BWR} = \frac{1767}{1.3} = 1359\ kN\ per\ m\ length$$

$$q_{upward} = Buoyancy - Weight = 1767 - 1359 = 408\ kN/m$$

See figure 2.21 for a clarification. Note that q_{upward} is the resultant of the Weight and the Buoyancy.

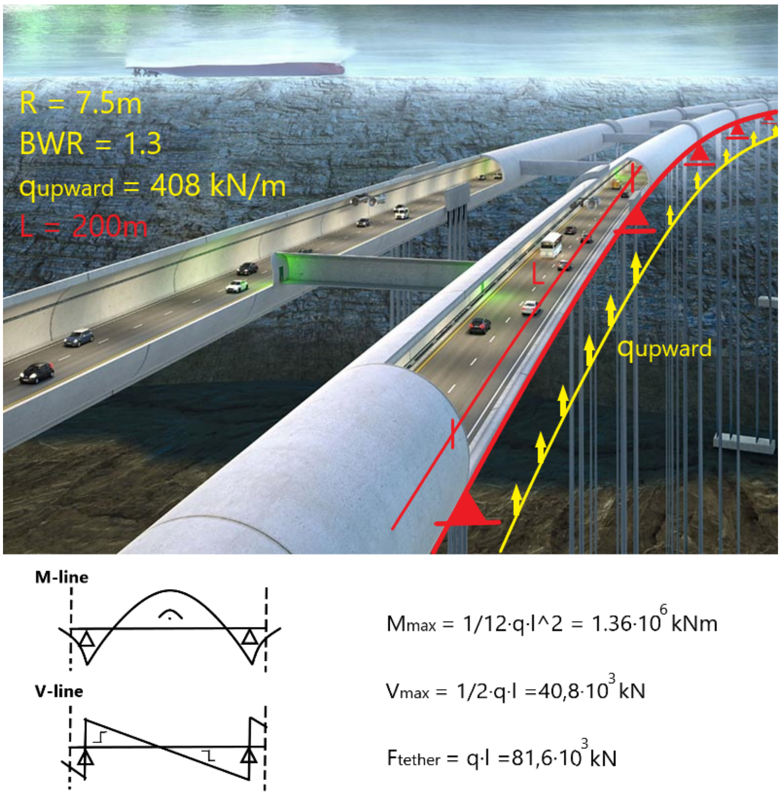

R = 7.5m
BWR = 1.3
qupward = 408 kN/m
L = 200m
L
qupward
M-line
V-line
M_{max} = 1/12·q·l^2 = 1.36·10^{6} kNm
V_{max} = 1/2·q·l =40,8·10^{3} kN
F_{tether} = q·l =81,6·10^{3} kN

Figure 2.21: Sketch showing the presence of shear and bending for $BWR \neq 1.0$.

The span between the tethers is typically around 200 m. When this is the case, the tether forces can be calculated per support and the moments and shear forces in longitudinal direction as well. From basic mechanics:

- $F_{tether} = q \cdot L = 81.6\ MN$
- $M_{Ed} = \frac{1}{12} q l^2 = 1353\ MNm$
- $V_{Ed} = \frac{ql}{2} = 40.8\ MN$

Concluding, some real significant loads/stresses occur for a BWR far from 1.0. The tether diameter can be quickly estimated, by assuming that the support reaction is absorbed via normal force only. Note that this value becomes even larger for tethers under an angle.

$$\sigma = \frac{N}{A} = \frac{40.8 \cdot 10^6}{A} = 235\ N/mm^2$$

Note that each support is anchored to two tethers, meaning that the force F_{tether} as listed above is divided by 2. The cross-sectional area of each tether would almost reach up to $D = 500\ mm$, which is large. A quick check of the reinforcement in longitudinal direction would also result in significant amounts of steel.

$$D_{tether} = 2 \cdot \sqrt{\frac{A}{\pi}} = 2 \cdot \sqrt{\frac{40,8 \cdot 10^6/235}{\pi}} = 470\ mm$$

In the next section, the concept of Form-Finding is explained. This design process searches for a shape which diverts the loads axially **within the cross-section** for a BWR equal to 1.0. It is important to understand the difference with respect to the actual structure in 3D, because shear forces and bending moments are present in longitudinal direction as stated by the example above. For the cross-section, just the tube is considered without the effects of the buoyancy-weight ratio and the tether forces.

2.4. Form finding

In this section, the concept of form finding is explained. This way of engineering is applied for the SFT in this research. Therefore, it is important to understand the underlying thoughts. First, a brief history of the form finding method is given where Gaudí is one of the architects who is well known for his form finding methods. Then, an analytical view on the deformations due to hydrostatic water pressures is shown.

2.4.1. General history of Form Finding

Form finding is sometimes known as a relatively new concept, whereas it was actually applied ages ago already. Maybe one of the first engineers, or an architect actually, who got well known due to form finding was Antoni Gaudí [20]. Form finding, or 'form follows force', is a way of structural engineering in which the forces acting on a structure determine the shape of the structure itself.

The idea is brilliant in its simplicity. By exerting a certain force on a structure or material without any stiffness, the structure will deflect in such a way that the forces are diverted in an axial, or in-plane way [39]. This is because the material does not have a (significant amount of) bending capacity, because it doesn't have any stiffness as just mentioned. This means that the forces have to be diverted axially or within the plane and that no shear forces or bending moments can occur. Especially in the earlier times, where masonry was the main construction element, this concept in engineering was very important. Masonry also has only a limited amount of bending or shear capacity, whereas the axial capacity (compressive strength) is rather large. By constructing buildings in a 'form finding' way, the masonry was used in an optimal way. Moreover, materials like reinforced concrete or steel were not widely available back then. This means that engineers had to find a way to use the available materials in an 'axial way'.

As just mentioned, Antoní Gaudi was a Spanish architect who was born at the end of the 19th century. Probably, he is most famous for his design of the Sagrada Familia in Barcelona. Gaudi is sometimes called the founder of these 'form follows force' methods. He used items like chains to find the optimal form for his great works. See figure 2.22

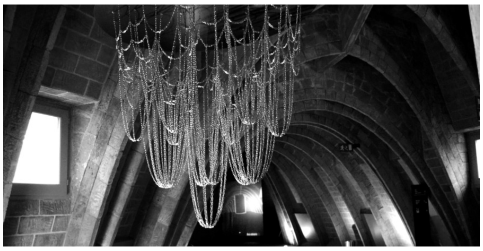


Figure 2.22: The hanging chain applied for 'Form Finding'

As visible from the figure, the chain takes forms like an arc, but upside down. This is not surprising, since almost all (old) buildings show shapes like these. Besides, arcs are indeed well known for their ability in diverting forces in an axial way. This was also already mentioned in section 2.1.3. In figure 2.23 a picture is shown of the house of Milá, with the structure made of diaphragm arcs [20]. The vaults are clearly visible in the form of the chain from figure 2.22.

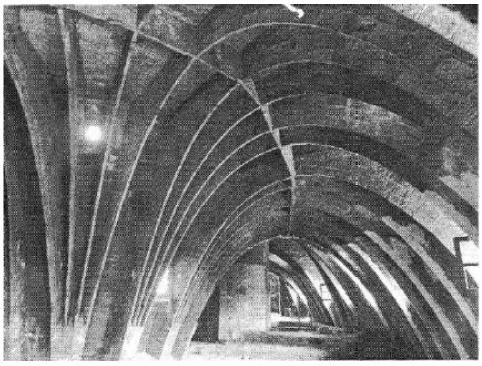


Figure 2.23: The vaults with the 'form finding' shape in one of Gaudi's works

Since the submerged floating tunnel has to deal with high compressive forces due to the hydrostatic water pressure, plus other forces due to waves and currents, form finding is a good way in determining the optimal shape. The materials are optimally used when they mainly have to transfer axial loads. Since the SFT is not subject to a uniform hydrostatic water pressure, a circle is probably not the optimal shape.
Rhino/Grasshopper has got several packages built-in, like Karamba3D and Kangaroo, which are capable of 'form finding'. Therefore, these software programs are used in determining what ***is*** the optimal shape of a SFT.
Concluding:

"Form-Finding is a design process where the (optimal) shape of a form-active structure is found for a static equilibrium, in which the structure diverts the loads axially or in-plane [26]"

2.4.2. Load profile due to the Water Pressure

Here, the deformations due to hydrostatic water pressure are analytically derived. In this way, the output of the form finding method in Grasshopper/Rhino can be verified. The tunnel structure is subject to the hydrostatic water pressure which increases linearly over depth. The total profile over the cross-section looks like in figure 2.6.
This water pressure can be divided into two parts. One constant part for the whole cross-section and one variable part which differs by depth. This variable component can be expressed in terms of the radial angle θ. In figure 2.24 an overview is shown for a situation where the top of the SFT is just below the water surface. The first component, p_0, is the water pressure at the local water depth of the center point. This means a water depth equal to $r - meters$. In case the SFT is at a deeper level, this value has to be enlarged by the water depth between water surface and top SFT. Consequently, the variable part has to be added up to the constant component in order to reach the total pressure at the cross-section. This pressure depends on the cosine function of the radial angle θ, since it is only the vertical component which affects the unidirectional water pressure.
The variable component has an amplitude equal to the radius of the SFT. The total difference in pressure between the top and bottom of the SFT is then equal to twice the radius, which is in line with the expectations. Also bear in mind that the variable component changes from sign at the locations where $\theta = \frac{1}{2}\pi$ and $\theta = \frac{3}{2}\pi$, as the cosine function changes from sign at these values. This is also what should happen in order to achieve the correct water pressure distribution.

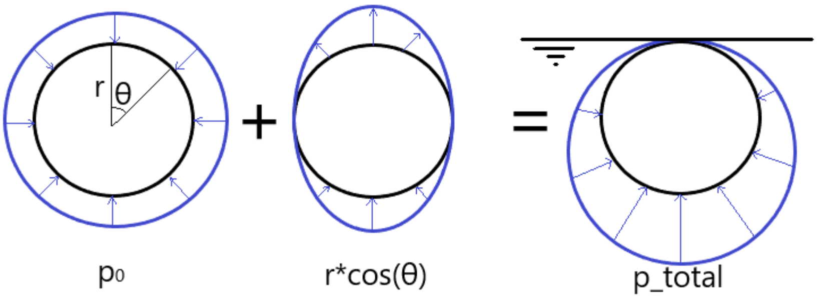


Figure 2.24: The water pressure decomposed in two parts

In formula, the water pressure distribution can now be defined as:

$$p(\theta) = \rho g \cdot ((r + d) - r * cos(\theta)) \tag{2.30}$$

Where:

- p = hydrostatic water pressure [Pa]
- ρ = density of fluid $[kg/m^3]$
- g = gravitational acceleration $[m/s^2]$

- r = radius of the SFT [m]
- d = depth of the top of the SFT [m]
- θ = angle from the vertical along the cross-section [rad]

The hydrostatic pressure changes over depth, as can be observed by the variable component in the pressure distribution. Due to these relative pressures, the circular cross-section wants to deform. The form finding method lets the cross-section deform in such a way, that the shape diverts the loads in an axial way. This is optimal because all the fibres of the material are used equally.

The shape that is expected is the one with the total load acting on it, as shown by p_{total} in figure 2.24. It is this shape since the form finding method searches for the shape defined by the load profile (form follows force).

2.5. The model and software

Several software programs and plug-ins are used to model the Form-Finding process for the SFT. In order to understand the output of the model, it is good to know how the software programs function. The main components are mentioned and briefly elaborated below.

2.5.1. Rhinoceros

Rhinoceros is a computer software especially designed for 3D computer programming. The link between programming and a 3D graphical interface is a special feature of this software. Basically, Rhinoceros is the graphical interface, where other plug-ins are responsible for the programming. It is also possible to make 3D-models or drawings in Rhinoceros itself. Consequently, these drawings or elements can be coupled to the programming language 'Grasshopper'. However, these designs are not parametric, which is favourable in the concept of this thesis. To make a parametric design, some programming tool is required named 'Grasshopper'. See the next paragraph.

2.5.2. Grasshopper

Grasshopper is the main programming 'environment' which is used within Rhinoceros software. However, one can hardly call it a language as it is build up like a flowchart consisting of a lot of different components. Therefore, Grasshopper is very 'accessible' language for beginners. Within Grasshopper, plug-ins like Karamba and Kangaroo are available in order to make constructive calculations. It is also possible to make FEM calculations within Grasshopper. As just mentioned, the main benefit of using Grasshopper is that it enables one to make parametric designs. As an optimal shape for a random SFT is searched for in this research, it is important to have the ability to change the input parameters.

2.5.3. Kangaroo

One of the plug-ins in Grasshopper is 'Kangaroo'. Kangaroo can be combined with any of the other components of Grasshopper and it functions on all geometries generated in either Grasshopper or Rhinoceros. Basically, this plug-in is a live physics engine in the process of form-finding, working with discrete elements and a local stiffness based solver [5]. A form of dynamic relaxation (DR) method for iteratively solving structural problems is applied in this plug-in, where the forces on each node are accumulated to calculate accelerations, velocities and displacements for each node. Basically, one can describe the Kangaroo solver as a mass-spring like system. It is important, however, to realize that Kangaroo doesn't work with physical quantities. A bending stiffness for instance is not described by EI with the units kNm^2, but with local stiffness S. Therefore, all deformations are relative with respect to each other, the form-found shape is in line with the actual physics. To calculate the resulting forces on the SFT, another tool is used namely Karamba3D.

2.5.4. Karamba3D

Karamba3D is another plug-in that can be inserted into Grasshopper and which can be used for structural engineering. Moreover, it is also fully incorporated in Grasshopper. It provides an accurate analysis on spatial frames, trusses and shells. Moreover, Karamba3D is also able to use in a parametric design. The important benefit of this tool is that it enables the user to work with the physical quantities as known and applied in practice.

An overview of all software programs is shown in figure 2.25 below.

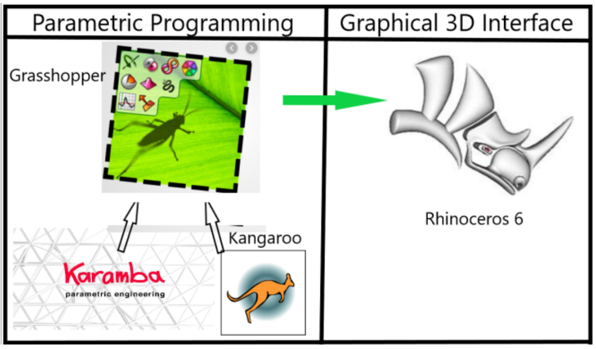


Figure 2.25: The software programs used in relation to each other

2.6. Assumptions

In this section, the assumptions made to proceed with the research are listed. At the end of the research, it is checked whether the assumptions made are valid.

- Only the permanent loads and the hydrodynamic loads due to currents and waves are included in this analysis. No accidental loads are included, but must be considered in further research.

- In reality, the tethers themselves would be subject to forces due to currents and waves as well. However, since the surface of the tether is negligible compared to the surface of the SFT, these forces are not taken into account.

- Turbulent flow around the structure can be of large importance regarding the stiffness and the stability of the SFT. The turbulent flow causes a dynamic response of the tunnel structure, which should be limited in amplitude to guarantee both comfort and safety. Moreover, the turbulent flow exhibits a force on the structure that might cause torsion. Although closed cross-sections are performing generally well on torsion, the effects have to be investigated.
  The modelling of a turbulent flow is quite complicated and can barely be done by an analytical solution. Therefore, a computational fluid dynamics (CFD) model should be inserted. The effect of turbulence is neglected in the extent of this report. However, one should not forget these effects in reality.

- It is assumed that the accuracy of the model is high enough to analyse the hydrostatic pressure on each segment to be linear. This also results in a hydrostatic force acting perpendicular to the tangent at this node. This can be checked by comparing the models output with the analytical solution for the normal forces.

- When evaluating the optimal shape for the external loads, just the tubular cross-section is considered without the effects of the BWR or the tether forces. Besides, this is not of influence on the movement of the SFT due to hydrodynamic loads. However, when Form-Finding for a shape which diverts the loads axially within the cross-section, one has to remind that shear and bending are inevitable for the 3D-structure due to a BWR not equal to 1.0. See figure 2.21.

3 Model - Hydrostatic loads

In this chapter, the permanent loads are modelled and the form finding process in Grasshopper is elaborated further. The permanent loads consist of the hydrostatic water pressure and the self-weight of the SFT. The tether force F_{tether} is zero, since the BWR is taken as 1.0 in this Chapter. The main thoughts of the process are elaborated in this Chapter of the main report, whereas the actual built-up of the model is shown more extensively in Appendix A.
It is important to understand to which optimization target this Chapter applies. The hydrostatic pressure is not of any influence on the dynamic behaviour of the SFT. Neither is the self-weight of the SFT. Hence, it has got nothing to do with the optimization target which aims for tension forces in the tethers. On the contrary, it is usually the main load acting on the structure causing structural stresses due to the large water pressures. Especially at larger depths, the water pressure dominates over the hydrodynamic loads. Therefore, this Chapter focuses on the second optimization target as listed in section 1.2. The optimization target:

$$Optimize\ the\ material\ use\ of\ the\ SFT$$

Where the aim is to find the **shape** of the cross-section which absorbs the external loads most efficiently. This implies a **minimization** of material use. This process is executed with the help of the 'Form-Finding principle', as explained in section 2.4. Moreover, the cross-sectional resultant forces must be quantified.

First, a short introduction is given on the hydrostatic water pressure. Then, the variables which are desired to be kept parametric are listed, followed by a short explanation for the model set-up in Grasshopper. Then, the looping process is described, which is required to account for geometrical effects and rotational effects of the segments. Fourthly, a reference case is worked out for a circle. Finally, the output of the FF-shape is evaluated, validated and compared to the reference case.

3.1. Introduction - Hydrostatic water pressure

By analyzing the model with common physical sense, one can already validate the model until a certain extent. Since the Form Finding method, or 'Form follows Force' method is based on deformations of the structure lead by the loads, it is expected that higher loads result in higher deformations. For a SFT being subject to an increasing water pressure over depth, one might expect an oval rather than an usual circle for the cross-sectional shape. The higher loads at the bottom of the SFT should result in larger deformations. See figure 2.6. Besides, the oval shape has a larger curvature at the bottom, meaning that the normal force is more efficient in absorbing the external water pressure. Hypotheses like this can be answered by modelling the hydrostatic water pressure in Rhinoceros/Grasshopper.

3.2. Parameters

The design should be parametric, as was already stated by the objective of this research. Therefore, the parameters which can change for different designs have to be inventoried in advance. They are:

- Depth of the SFT [m]

- Required diameter of SFT [m]
- Fluid density $\rho_{water} [kg/m^3]$

Here, the depth of the SFT represents the **depth on top** of the SFT. Together with the diameter of the structure, the local depth z can be calculated. The self-weight is determined by the buoyancy of the SFT. As already mentioned in the scope of this research in section 1.3, the BWR is assumed to be 1.0 in order to focus purely on the cross-section Eventually, the aim is to keep the BWR as close to 1.0 as possible for an economic design. The self-weight is equally distributed over the tunnel lining. This is further elaborated in paragraph 3.3.2.

3.3. Model set-up

In this section, the model set-up is briefly described with the main underlying thoughts. For a total view on the model set-up, one is referred to Appendix A.

3.3.1. Geometry and hydrostatic load

First, a simple circle is generated as a starting point. A circle is chosen since it is expected to be close to the 'optimal shape' according to FF. The circle is created within a 2 dimensional plane, with an imaginary extrusion of 1 m in the longitudinal direction. Hence, in the remaining of this report, all loads are normalised *per meter length*.

Consequently, the circle is being divided into N sections. This defines the accuracy of the model. The denser this new 'grid', the more accurate the results but the higher the computational time required. A polyline is drawn through these N nodes, after which it is exploded again. The result is now a geometry consisting of 'N' nodes and 'N' segments. It is important to understand these definitions, as they return often in describing and defining the model. For an overview of the nodes and the segments, see figure 3.1. Now, the starting geometry is generated.

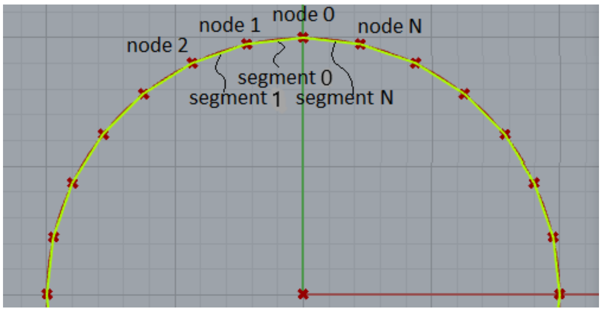


Figure 3.1: The definition of nodes and segments

Secondly, the force per node is calculated. This force is calculated by determining the water pressure at a node, multiplied by the length/area of the segment which diverts its load to the node. The water pressure is calculated according to equation 2.1, where z is the z-coordinate of the node. The segments adjacent to each node divert half of their length to the node. For the first loop, where all segments are equally long, each node takes one segment length. The output of this part is a list of forces per node, depending on the local water depth and the length absorbed by the node. Note that the force is equal to the hydrostatic pressure, multiplied by the segment length *and* the imaginary depth of $1m$.

Thirdly, these forces are positioned correctly on the nodes. The list of forces represents the *magnitude* of the vectors, because the hydrostatic water pressure is unidirectional. Therefore, the forces are acting perpendicular to the circles' surface. It is important to first generate a new plane at the nodes. Each node has got a new local plane (u,v,w), where the w-coordinate represents the perpendicular component to the circles surface. The creation of the new coordinate systems is rather easy for a circle in Grasshopper. First, one draws a line from the circle centre point to each node. Then, a new local plane can be generated perpendicular to this base line in the origin of the node. Again, see Appendix A for a more detailed view upon this.

Now, a local vector can be created with the magnitude of the force vector acting in the w-direction. For a clarification of this local coordinate system, see figure 3.2. Note that the water pressures directed *inward* are modelled as forces *outward*. This is because load and resistance are contrarily directed and to find the optimal shape, the opposite direction of the force has to be modelled. Again, the model is a slice of the SFT, where all quantities are normalised 'per meter length'.

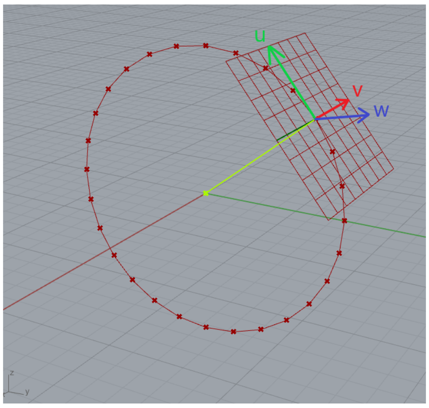


Figure 3.2: A locally generated plane, tangential to the circle

Note: theoretically the force doesn't work perpendicular to the tangent of the circle. This would be true if the hydrostatic pressure is uniform over the cross-section. However, as mentioned often before, the hydrostatic pressure increases with depth. Therefore, the 'tangent of the resultant force' is tilted a little to the deeper parts as the pressure here is higher. Since the force acts perpendicular to this 'working line', the force on each node also is tilted a little to the deeper parts. This effect is ignored by increasing the accuracy of the model, meaning that the assumption of a uniform pressure over the segments becomes more and more valid. For a clarification of this effect, see figure 3.3.

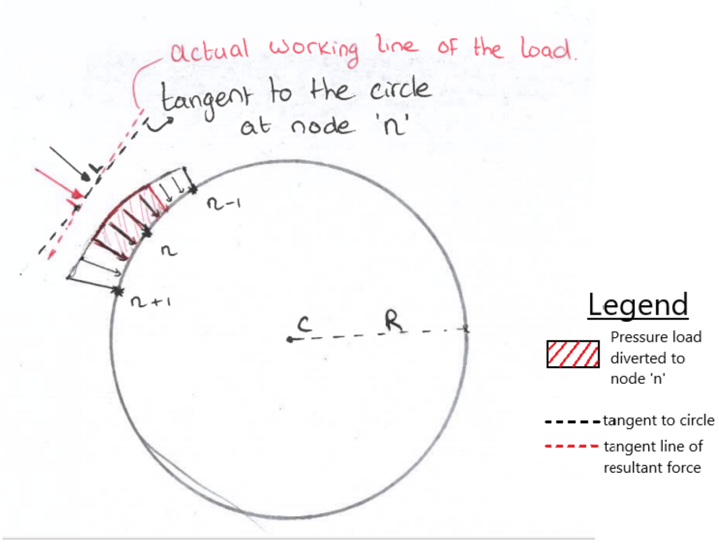

actual working line of the load.
tangent to the circle
at node 'n'
n-1
n
n+1
c
R
Legend
Pressure load diverted to node 'n'
tangent to circle
tangent line of resultant force

Figure 3.3: The tilted force due to non-uniform pressures

Both the segments and the loads are inserted in the Kangaroo Plug-in within Grasshopper. The segments are inserted into a 'Length Line' component, meaning that is has no bending stiffness. This is required according to the definition of Form Finding. The load vectors belonging to each node are inserted in the plug in as well.

3.3.2. Self Weight and anchor point

The anchor point is very important in the model. The model functions well as long as the loads on the geometry are uniform in each direction. As explained in section 2.5, the plug-in Kangaroo functions as a mass-spring like system. Each load results in an acceleration, velocity and displacement per node according to the spring stiffness 'S'. This means that no relative movements are present, given that the sum of all the forces is equal to zero for each direction. Concluding, Kangaroo is holding the geometry at its original position. Once the loads

are not uniform for each direction, the geometry starts moving because the acceleration is not zero (Newton's 2nd law). If there is no anchor point, the geometry keeps 'falling' in the direction of the net force. Therefore, an anchor point is required, since the forces acting on the SFT are not uniform. This can be done in two ways, explained in Appendix A.
Concluding from the statement in the Appendix, it is favourable to use a random point at the shapes' surface. The other option, an anchor point in the centre of the shape, is unfavourable because additional elements are inserted into the model then. These additional elements require an additional loop which is unnecessary. Applying the anchor point to a random point at the surface only requires a stabilizing force, holding the shape at its original position but not affecting the FF-shape. See figure 3.4 for the anchor point and the stabilizing forces shown in yellow. This additional force basically represents the self-weight of the SFT. Since it stabilizes the vertical force from the buoyancy, it can be compared to the self-weight of the SFT in a situation where the BWR is equal to 1.0. Note that all the self-weight is now assumed to be in the tunnel lining.

These forces do not affect the shape, but do affect the N-line as found by Karamba3D. Later on, this is explained further. In the figure below, both the hydrostatic water pressure load and the stabilizing force (self-weight) are shown.

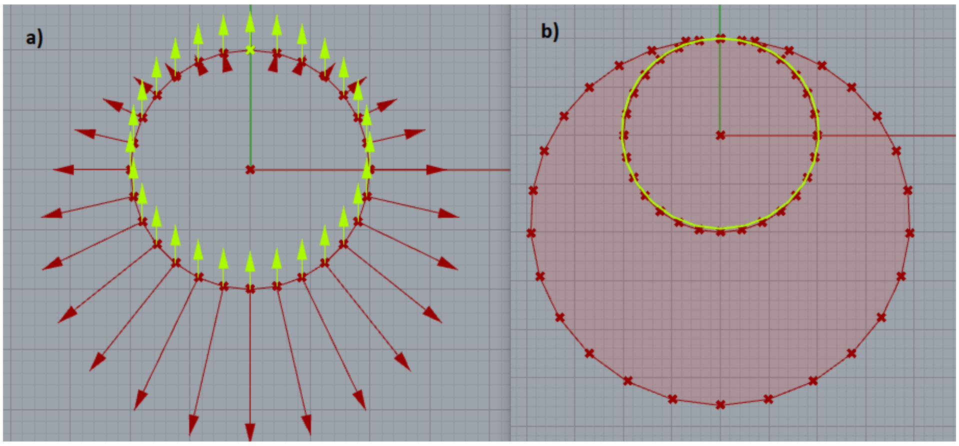


Figure 3.4: a) the geometry with loads, vertical stabilization loads in yellow. b) the form-found shape, scaled back in yellow

3.3.3. Loops

The next step is to loop the model in order to converge to a stable solution. Due to the deformed shape, the nodes absorb different segment lengths (surfaces). This geometrical effect has to be taken into account. Moreover, the orientation of the segments and nodes has changed as well, resulting in a different orientation of the loads and the necessity of a new local plane.

Geometrical effect

For a sketch of the geometrical effect described above, see figure 3.5.

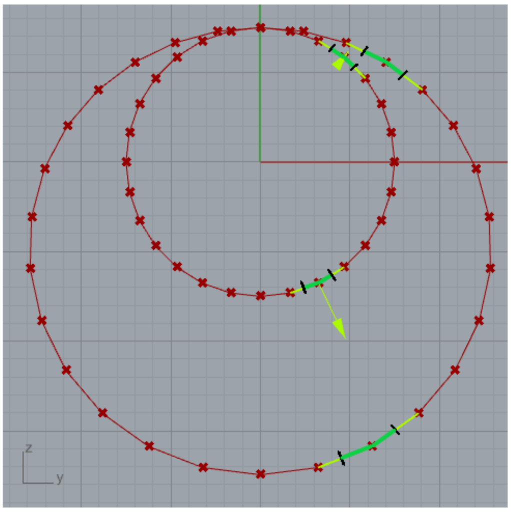


Figure 3.5: The geometrical effect due to deformations

Here, the original circle (starting geometry) and the deformed shape (outer 'circle') are shown. A random node at the right top and at the right bottom are highlighted.
The N nodes of the shape each absorb the pressure forces from half the lengths of the adjacent segments. Note that the load is a pressure, multiplied by the area of the subdivided section. This area is equal to the length of the segment, multiplied by the (imaginary) length of 1 m. In this way, a force is generated again.
Since the plug-in 'Kangaroo' functions as a mass-spring system, the load acting on the nodes results in a deformation of the node. Consequently, the area which transfers its pressure to nearest node changes as well. This means that the total force on all of these N nodes changes. See figure 3.5. The yellow lines are the adjacent segments of the marked nodes with the highlighted load vector on it. The green lines represent the both halves, which are absorbed by the node. For the original circle (starting geometry), the green lines are equally long. This is simply because the circle was equally subdivided into N segments. For the deformed shape, however, this length (surface) is *relatively* larger for the node at the bottom than for the node at the top. This is because of the higher pressure loads deeper in the water column. Hence, the higher loads result in larger deformations. After the first loop, the nodes at the bottom absorb a relatively larger pressure force. This geometrical effect has to be taken into account by *looping* the Form-Finding method. The form-found shape is scaled back to its original circumference and then subjected to the loads again.

Defining new local planes

The pressure forces act perpendicular to the structures' surface, as explained in section 3.3.1. These forces can be modelled by introducing a new coordinate system. Defining this new plane, perpendicular to the tangent of the shape, is rather easy for a circle. This process is already explained. Now, the shape is not a perfect circle anymore, which complicates the process of introducing a new Cartesian coordinate system (u,v,w) at each node in the right directions. However, another solution is found and described below. See figure 3.6 to help understand the description.

1. Start with node 'N'
2. Select nodes 'N-1' and 'N+1'
3. Generate a new circle through these three points
4. Select the centre of this new local circle
5. Draw a base line between this new local circle centre and node 'N'
6. Generate a new local coordinate system (u,v,w) perpendicular to the just created base line, in node 'N'
7. Repeat this process for all nodes 'N'

When executing this process, each node 'N' has got a new local Cartesian coordinate system which allows the user to apply a force perpendicular to the structures' surface. Again, it is important to be able to apply such a pressure force on the SFT, since the water pressure is unidirectional.

The orientation of this new local plane depends on the curvature/orientation of the two adjacent segment per node. The benefit of creating a new circle through the three points 'N', 'N-1' and 'N+1' is that this circle is generated based on the curvature/orientation of these two adjacent segments. This means that the new local plane is always oriented by an angle, average of the two adjacent segments. This should be the case.

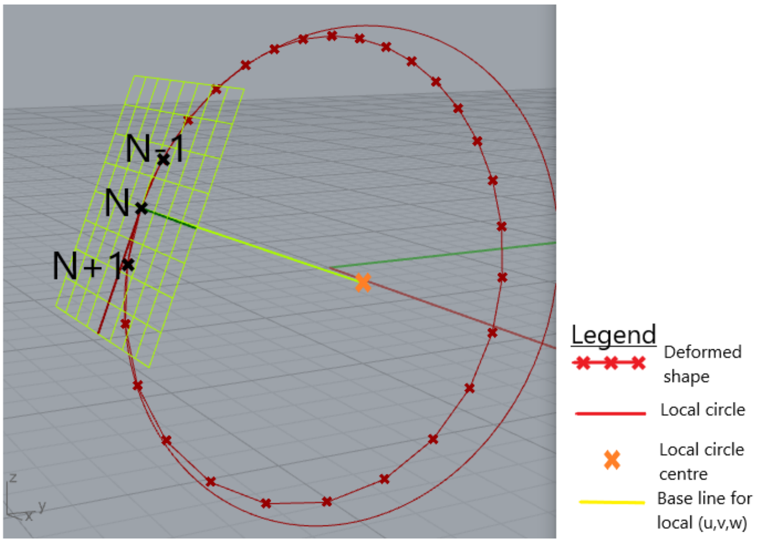


Figure 3.6: the underlying elements to help generating the new u,v,w, coordinate system.

The deformed geometry is now ready to be processed by an additional loop. The shape converges to a stable solution. This is shown in the next paragraph.

3.3.4. External looping via file

The looping process is executed externally in Grasshopper, because the canvas in Grasshopper becomes messy when looping internally. Internal looping would mean that the output of loop 'i' is used as input for loop 'i+1', while the whole process mentioned above has to be copied in Grasshoppers' canvas. Another benefit of externally looping is the accessibility of the output of the model.

This means that a starting geometry is generated and written to an external file. A circle with radius R is created and subdivided into N nodes, as explained before. These nodes, consisting of coordinates, are exported to this file. Then, the same file is imported to Grasshopper again. A polyline is drawn through these coordinates, yielding the starting shape of the SFT within the looping process. Consequently, the cross-section undergoes all the steps explained before, until the deformed shape is scaled back to its original circumference. The coordinates of the scaled geometry, again a list of N coordinates, is now written to the same file again. Grasshopper now recognizes the new input and starts to execute the process again, until the relative movements are practically zero again (Kangaroo's converge demand). In this way, an automatic loop is created. However, for every loop one has to press the 'loop' button, which actually implies the 'write to file' command. The renewed coordinates are then inserted in the model again. Typically, the shape is converged after 5 to 10 loops. An overview of this process is shown in the flowchart below in figure 3.7.

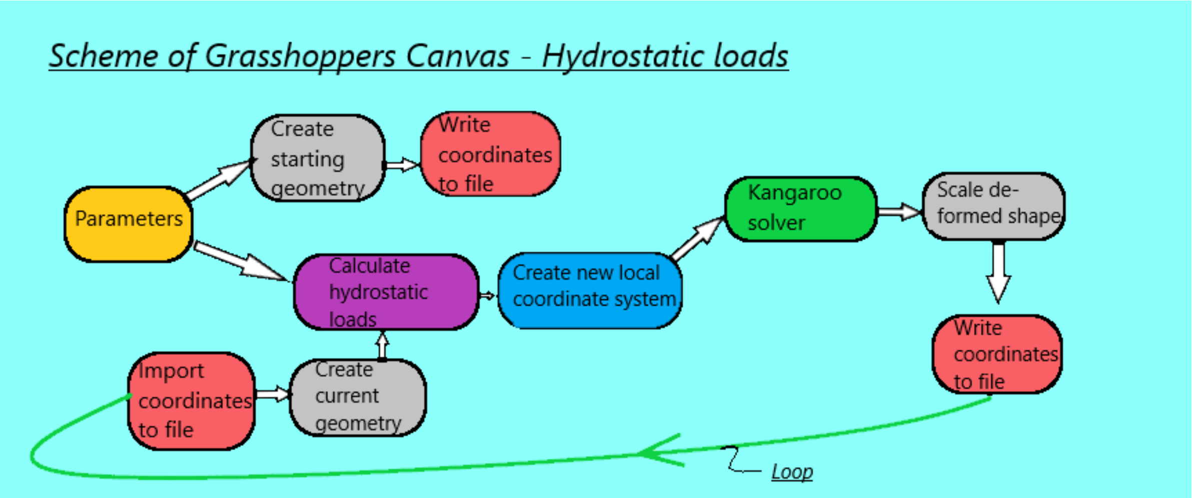

Scheme of Grasshoppers Canvas - Hydrostatic loads
Parameters
Create starting geometry
Write coordinates to file
Calculate hydrostatic loads
Create new local coordinate system
Kangaroo solver
Scale de-formed shape
Write coordinates to file
Import coordinates to file
Create current geometry
Loop

Figure 3.7: the external looping process in Grasshopper.

3.4. Reference case - Circle

Before analyzing the output of the model described above, a reference case is worked out for a circle. It is expected that a circle already performs quite well in absorbing the hydrostatic pressures.

The normal forces acting on each node of the model should be in equilibrium with the external loads. These external loads can be determined easily by integrating the hydrostatic water pressure over the area 'belonging' to the node. The curvature of the deformed shape at the location of that node determines if the normal forces can balance the external load.

The formula for the hydrostatic water pressure over the cross-section is shown in equation 2.30, as elaborated in paragraph 2.4.2. This can be converted to a formula for the normal force (N-line) over the cross-section for a *circular* cross-section, according to [33]:

$$N_0(\theta) = -\rho \cdot g \cdot (R+d) * R + \rho \cdot g \cdot R^2 \cdot cos(\theta) \tag{3.1}$$

Where:

- N_0 = 1st order normal force [N]
- ρ = fluid density $[kg/m^3]$
- g = 9.81 $[m/s^2]$
- R = radius of SFT [m]
- d = depth on top of SFT [m]
- θ = angle from the vertical [-]

The profile of the N-line, according to the analytical solution, can be drawn for the situation where the fluid density is equal to $\rho = 1050\ kg/m^3$, radius $R = 7.5\ m$ and depth on top of SFT $d = 0\ m$. The accuracy of the model here is $N = 28$. See figure 3.8 for the analytical N-line over the cross-section for a perfect circle.

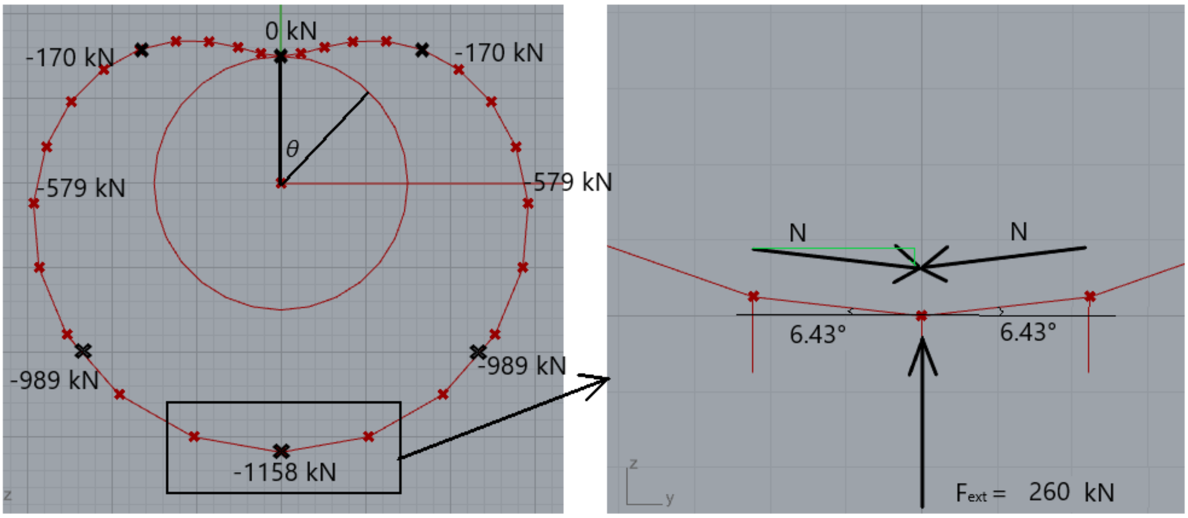


Figure 3.8: The N-line according to the analytical solution (left) and the equilibrium in a random node (right)

The concept where the curvature of the shape enables the equilibrium with the external forces is explained now for this circle. On the right in the figure above, a detail is shown of the lowest node in the cross-section. The adjacent segments of this node each have a normal force N. Since both segments are under an angle with respect to each other, they have the ability to absorb the external load.
The external load is positioned perpendicular to the geometry's surface, just how the pressure acts on the SFT. The external load can be calculated with:

$$F_{ext} = \rho g z \cdot A = 1050 \cdot 9.81 \cdot 15 \cdot 1.68 = 260\ kN$$

Where:

- $A = 2\pi R/N \cdot 1m = 2\pi \cdot 7.5/28 \cdot 1m = 1.68m^2$
- $z = 2R + d = 2 \cdot 7.5 + 0 = 15\ m$

The angle of the segments with respect to the tangent at the node determine the 'efficiency' of the normal force N to balance the external load. The higher this angle, the larger the component of the normal force which balances the external load. The component of N which makes equilibrium with the external load is equal to:

$$N_{eq} = N \cdot sin(\theta) = N \cdot sin(6.43)$$

Note that the circle is divided into N = 28 segments. The 'curvature' or rotational angle of each segment is then $\frac{360°}{28}$ = 12.68°. The angle with respect to the tangent is then equal to half this curvature, namely 6.43 °. Assuming that the external load is completely absorbed by the normal force, the load N can be calculated by:

$$N = \frac{F_{ext}}{2 \cdot sin(6.43)} = 1160\ kN$$

This is in line with the analytical solution as shown on the left in figure 3.8. Concluding, the shape absorbs the external load indeed with the normal forces. The result is no shear or bending in the cross-section. The maximum normal force is equal to N_{max} = 1160 kN. This value must be compared with the maximum value for the FF-shape to conclude whether the FF-shape is indeed more efficient. Now, the models' results are elaborated.

3.5. Output - Form Finding

The main objective of this part was to generate the most efficient shape in absorbing the external hydrostatic water pressure. Hence, that is what is focused on in this part. In figure 3.9 below, one can see the converged shape. This shape is the result after some loops which are required for earlier mentioned reasons. The intermediate results (loops) are also shown in black in figure 3.9. In green, the eventual shape is shown which indeed start showing some appearances of an 'oval' or an 'egg'.

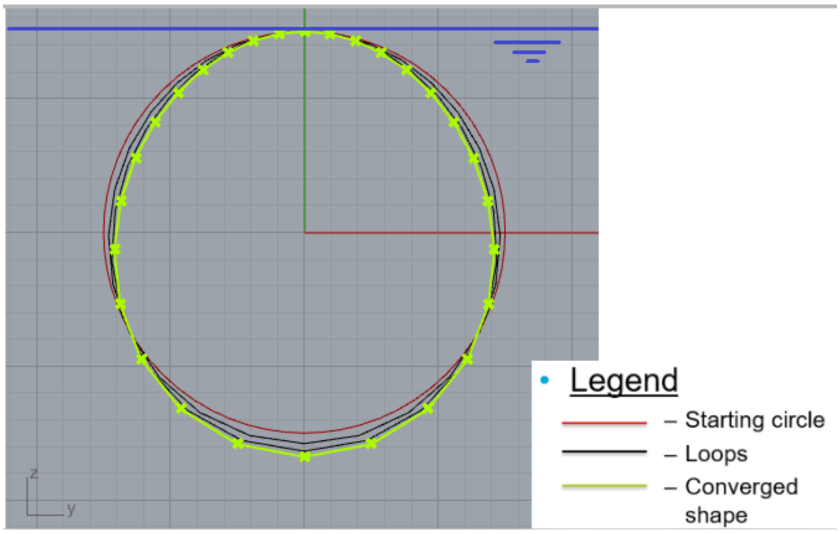


Figure 3.9: the output for the water pressure.

Now the output is known, it is possible to evaluate and compare it to the reference case as mentioned in section 3.4.
Moreover, the influence of the starting geometry has been evaluated as well as can be read in Appendix A.7. When starting with a rectangle as geometry and applying the same method, the same shape is the output. Concluding, the starting geometry does not have any influence on the final shape found by Form Finding.

3.6. Validation

Here, the model described above is validated for the hydrostatic loads. According to the Form-Finding principles, the model should return a shape which diverts the external water pressure into a normal force (N) only. No shear forces and bending moments should be present. Besides, this should be the most efficient shape in absorbing the hydrostatic water pressure. Hence, the N-line of the FF-shape must be compared to the N-line of the reference case, which is a circle.

The N-line for the deformed shape will be different than for a perfect circle. Therefore, a different plug-in in Grasshopper is used to calculate the normal forces N in all segments. This plug-in is Karamba3D. When applying the same conditions for ρ, R and d, the N-line can be calculated again. Note that the modelling of the SFT with a stabilizing vertical force equally distributed over all N nodes *does* influence the N-line in Karamba, as mentioned before in paragraph 3.3.2. Therefore, one has to be aware of this effect.
The lowest node is considered again, where the normal force according to Karamba3D is equal to:

$$N_{lowest-node} = 941.1\ kN$$

The external load, being 413.8 kN, has to be balanced by the normal forces. Note that the external load has increased due to geometrical effects and a larger depth. As just mentioned, the normal force should be equal to 941.1 kN. The vertical stabilizing force due to the anchoring method affects the N-line distribution. An overview of the loads at the lowest node is shown in figure 3.10. Here, the Form-Found shape retrieved from Kangaroo is also visible. This is the shape inserted into Karamba3D.

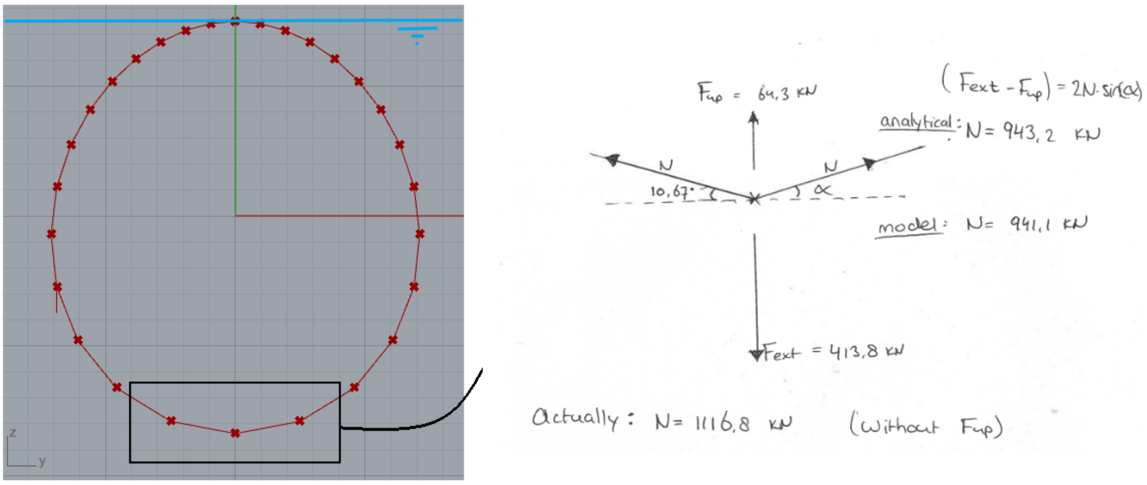


Figure 3.10: The equilibrium of the lowest node for the deformed shape

On the right, all the forces acting on the node are shown and the only unknown 'N' has been calculated. The forces are shown opposite to the way they act to simulate the Form-Finding principle where the loads are reversed. The important values here are:

- $F_{external} = 413.8\ kN$
- $F_{up} = 64.3\ kN$
- $\alpha = 10.67°$

The value α is retrieved from the output in Grasshopper. According to equilibrium, the normal force in the segments that should be present in order to absorb the external load fully axial is equal to:

$$N = \frac{F_{ext} - F_{up}}{2 \cdot sin(\alpha)} = \frac{413.8 - 64.3}{2 \cdot sin(10.67)} = 943.8\ kN \tag{3.2}$$

This value is close to the value found by Karamba, which was equal to 941 kN. Hence, this proves that the model indeed converges to a shape that absorbs the external loads via normal forces only.
However, the force F_{up} influences the actual value for N. Without this 'balancing' force F_{up}, the normal force would be equal to:

$$N = \frac{F_{ext}}{2 \cdot sin(\alpha)} = \frac{413.8}{2 \cdot sin(10.67)} = 1117\ kN \tag{3.3}$$

This value is already relatively close to the analytical solution for a circle, which was $N = 1158\ kN$. The Form-Found shape is slightly more efficient in absorbing the external load due to the larger curvature at the bottom. The order of magnitude is close to each other, which also proves that the process seems to be valid. Finally, the shape is also slightly more efficient as was expected.

The same can be done for another node, where F_{up} does not have the same working line as F_{ext}. Node 4 is the next one to be investigated. See figure 3.11

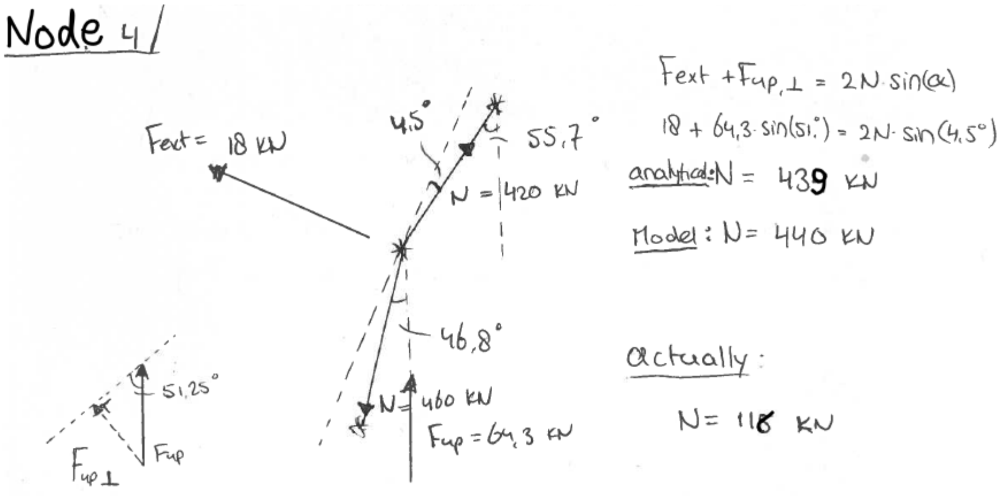


Figure 3.11: The equilibrium of node 4

The angles as shown in the figure are again retrieved from the output in Kangaroo. Besides, the loads are again reversely shown as is required for the FF-method.
First, the tangent at the node has to be calculated, which is the average of the two adjacent segments. The angles of these segments are $\alpha_1 = 55.7°$ w.r.t. the vertical and $\alpha_2 = 46.8°$. These values are output of the model in Kangaroo. This means that the angle of the tangent at node 4 is equal to the average, being $\alpha_{avg} = 51.3°$.
Then, the component of F_{up} in the same direction as F_{ext} has to be found. This is shown at the left bottom of figure 3.11. The component that has to be added up to F_{ext} is equal to:

$$F_{up}\perp = F_{up} \cdot sin(\alpha_{avg}°) = 64.3 \cdot sin(51.3) = 50.1\ kN$$

The external force at node 4 is equal to $F_{ext} = 18\ kN$. Now, the equilibrium in the perpendicular direction to the tangent at node 4 requires:

$$F_{ext} + F_{up}\perp = 2N \cdot sin(\alpha_{avg} - \alpha_1)$$

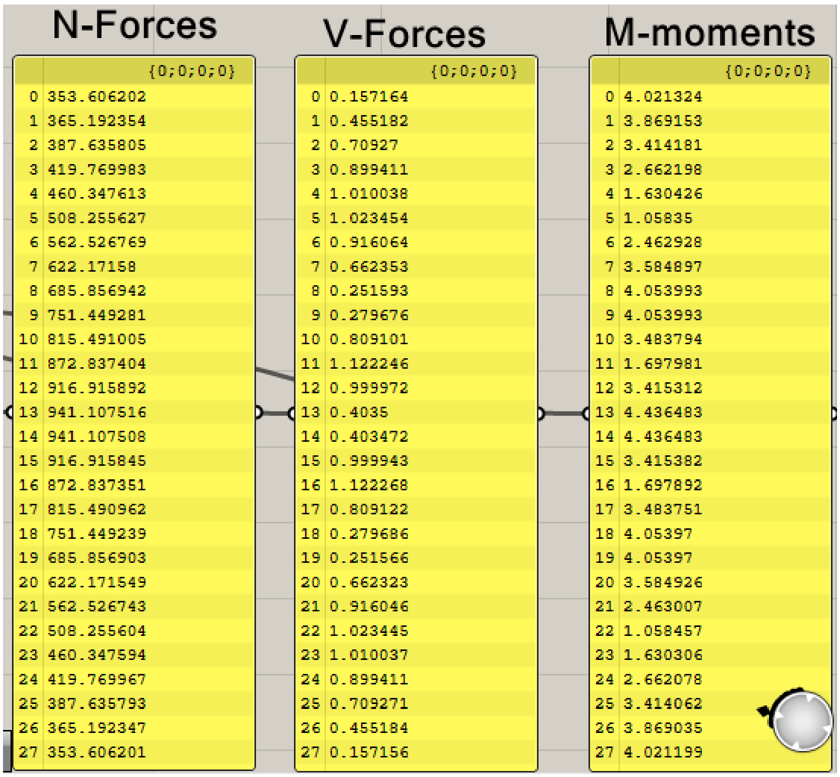
$$18 + 50.1 = 2N \cdot sin(4.45°)$$

$$N = 439\ kN$$

Concluding, also for node 4 the analytical solution is very close to the output of the model $N = 440\ kN$. But again, this value is influenced by the stabilizing force F_{up}. The actual value can also be calculated by using the same shape and angles, but without this vertical stabilizing force. The result for N is then:

$$N = \frac{F_{ext}}{2 \cdot sin(\alpha_{avg} - \alpha_1)} = \frac{18}{2 \cdot sin(4.45} = 116kN$$

Next to the N-line, Karamba3D also generates the V-line and the M-line. When the Form-Found shape of Kangaroo is inserted into Karamba3D and the same loads are applied, the following is the result.



	N-Forces		V-Forces		M-moments
	{0;0;0;0}		{0;0;0;0}		{0;0;0;0}
0	353.606202	0	0.157164	0	4.021324
1	365.192354	1	0.455182	1	3.869153
2	387.635805	2	0.70927	2	3.414181
3	419.769983	3	0.899411	3	2.662198
4	460.347613	4	1.010038	4	1.630426
5	508.255627	5	1.023454	5	1.05835
6	562.526769	6	0.916064	6	2.462928
7	622.17158	7	0.662353	7	3.584897
8	685.856942	8	0.251593	8	4.053993
9	751.449281	9	0.279676	9	4.053993
10	815.491005	10	0.809101	10	3.483794
11	872.837404	11	1.122246	11	1.697981
12	916.915892	12	0.999972	12	3.415312
13	941.107516	13	0.4035	13	4.436483
14	941.107508	14	0.403472	14	4.436483
15	916.915845	15	0.999943	15	3.415382
16	872.837351	16	1.122268	16	1.697892
17	815.490962	17	0.809122	17	3.483751
18	751.449239	18	0.279686	18	4.05397
19	685.856903	19	0.251566	19	4.05397
20	622.171549	20	0.662323	20	3.584926
21	562.526743	21	0.916046	21	2.463007
22	508.255604	22	1.023445	22	1.058457
23	460.347594	23	1.010037	23	1.630306
24	419.769967	24	0.899411	24	2.662078
25	387.635793	25	0.709271	25	3.414062
26	365.192347	26	0.455184	26	3.869035
27	353.606201	27	0.157156	27	4.021199

Figure 3.12: The resultant forces in the form-found cross-section

Note that all these N-values still have to be corrected for the stabilizing force F_{Up} as shown above.
The shear forces V and the bending moments M are very small compared to the normal force N. The fact that they are not exactly equal to zero can be explained by the fact that the model with 28 nodes is not accurate enough yet. However, it shows that the model searches for the right shape.

Finally, when the method shown above is applied for all nodes and the effect of the vertical stabilizing force is accounted for, the actual N-line of the deformed shape can be generated. When comparing this one to the one in figure 3.8, one can conclude that the Form-Found shape is indeed slightly more efficient in absorbing the external load, as the maximum normal force has decreased.

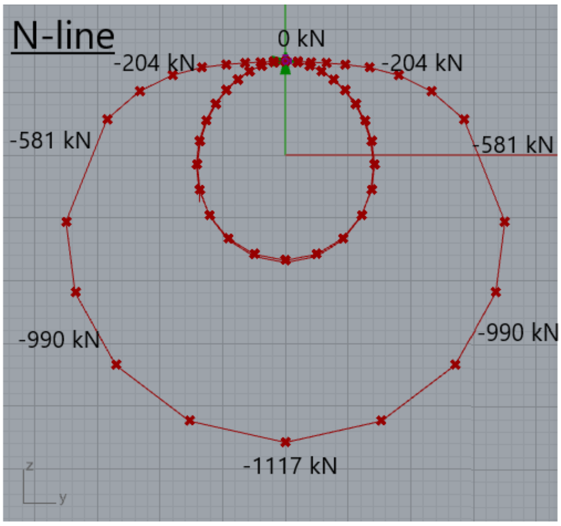


Figure 3.13: The N-line in the CS for the deformed shape

3.7. Conclusion

Concluding from this Chapter, the Form Finding method is applied for the hydrostatic water pressure. The shape found by FF is indeed slightly more efficient in absorbing the external hydrostatic loads. The maximum normal force for the reference shape, which is a circle, is equal to $N = 1160\ kN$. The maximum normal force for the FF-shape by Kangaroo is equal to $N = 1117\ kN$. No shear forces V or bending moments M are present, which is in line with the Form Finding definition. Note that the differences are relatively small, which can be explained by the fact that a circle is already a good performer in absorbing pressure loads. But still, the larger loads at the bottom of the SFT caused a higher local curvature within the FF-process. The larger curvature enables the normal force to absorb the hydrostatic water pressure more efficiently. A larger component of the normal force can now counteract the water pressure. So, the ideal shape for a SFT just under the water surface is not a circle but an 'egg'- or an 'oval'-like shape. This effect is known from the bored-tunnels discipline, expressed as the ovalization. The ovalization is the horizontal distance divided by the vertical distance of the cross-section. This effect decreases by increasing depth, as turns out from figure 3.14.

The numbers mentioned above were for a SFT just below the water surface. In this situation, the *relative* difference in water pressure between top SFT and bottom SFT are the largest, meaning that the largest differences are visible for the curvature of the shape. When the SFT is located deeper and deeper, this effect gradually vanishes out until a circle is retrieved again from the model.

Finally, one can say that a circle is not the perfect shape in absorbing the water pressure, but that the optimal shape comes closer and closer to a circle for a deeper located SFT. See figure 3.14.

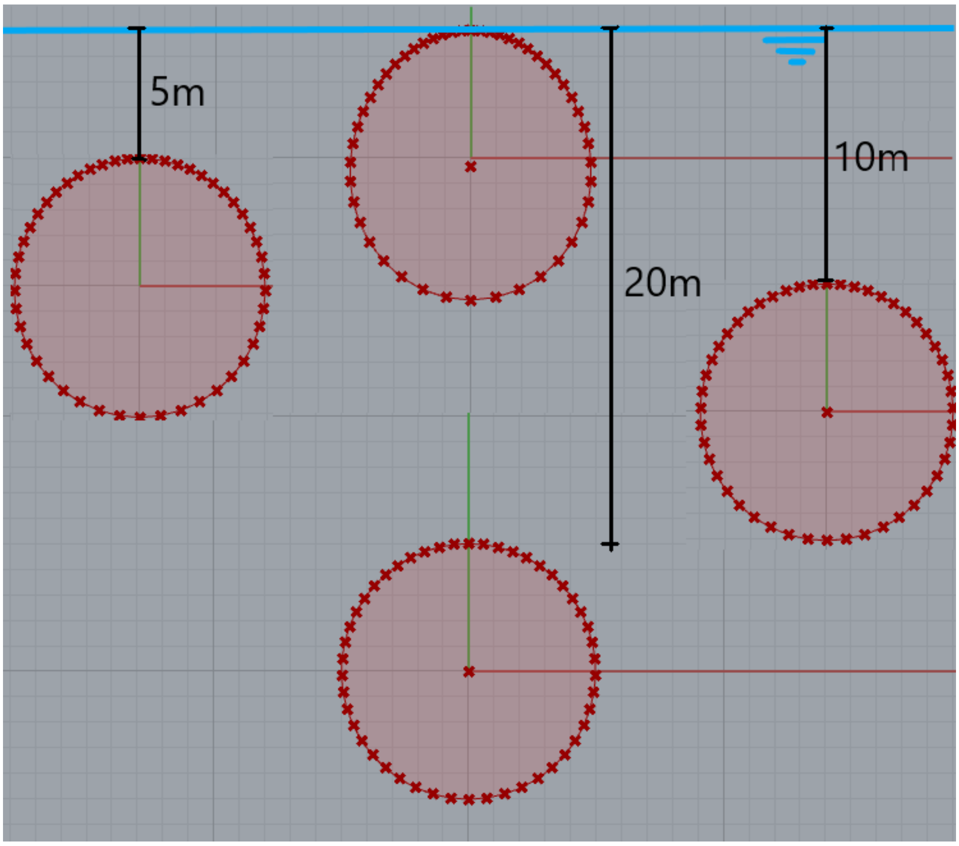


Figure 3.14: The optimal shape according to FF per depth

4
Model - Hydrodynamic loads

In this chapter, the hydrodynamic loads are elaborated. These consist of the loads due to currents and the loads due to waves. The hydrodynamic loads are the ones causing the SFT to move in a certain direction with consequences for F_{Tether}. Hence, this chapter concerns the $1st$ objective as shown in section 1.2:

$$Convert\ the\ external\ hydrodynamic\ loads\ into\ tension\ in\ the\ tethers\ of\ the\ SFT.$$

As became clear from Chapter 2, the lift coefficient for a regular circle is equal to $\overline{C}_L = 0$, when excluding turbulence effects. The goal of this Chapter is to find a shape which has a $\overline{C}_L$ value larger than 0, to generate a net vertical force from a horizontal approaching flow. This lift force must be upward in case the SFT is anchored to the sea bed. The opposite holds for a SFT connected to buoys at the water surface.

The $2nd$ objective, optimizing the material use, is also influenced by the hydrodynamic loads. However, especially at larger depths, the stresses due to currents and waves are only a small fraction of the hydrostatic water pressure. Concluding, the hydrodynamic loads are purely allocated to the $1st$ objective of this research and the hydrostatic loads are allocated to the $2nd$ optimization target, as was done in Chapter 3.

First, Bernoulli's principle is explained and an analogy is made with an aeroplane. This knowledge is required to understand what cross-sectional shape is required. Then, this principle is related to a circle. Thirdly, the actual shapes are generated for a horizontal approaching flow which meet the requirements according to Bernoulli. The output is a cross-section depending on a Geometry Factor. Consequently, the net vertical force must be determined. First, the lift force is determined by making use of a CFD, after which the same process is done with the 'the Source Panel Method'. This includes an analytical solution to determine the lift force on the SFT. Finally, some remarks are given on the wave loads.

4.1. Bernoulli's principle

Daniel Bernoulli was a Swiss mathematician and physician, born in Groningen [8]. He is probably most known for his theorem, which states that moving fluids and gases experience a decrease in static pressure given an increase in flow velocity. He worked narrowly with Leonhard Euler, who is the one who eventually derived the Bernoulli equation as we know it nowadays [21]. The Bernoulli equation, rewritten to length-units, is:

$$H = h + \frac{p}{\rho g} + \frac{v^2}{2g} \tag{4.1}$$

Where:

- H = Energy height [m]
- p = pressure in the fluid [Pa]
- ρ = fluid density [kg/m^3]
- v = flow velocity [m/s]

Furthermore, it is stated that this energy height is constant **along the streamline**. This means that the velocity of a fluid also determines the pressure within the fluid. And in its turn, the pressure determines in which direction the submerged body is pushed or pulled. This principle is also the explanation why an aeroplane flies. This object can be used as a reference project which might induce some first ideas on the eventual shape. See figure 4.1 for an overview of an aeroplane's wing and why it flies [23].

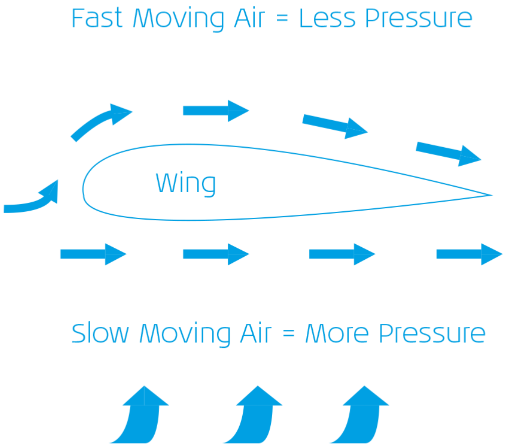

Fast Moving Air = Less Pressure
Wing
Slow Moving Air = More Pressure

Figure 4.1: Bernoulli's principle applied on an aeroplane

The top side of the wing has got a more convexly shape than the bottom. An air particle (or water particle in case of the SFT) has to travel a larger distance over the top compared to the bottom. Combined with the fact that continuity in the domain is required, this results in a larger flow velocity at the top of the wing. The faster moving air results in a lower pressure. Hence, the lift force is directed upward and the aeroplane flies. Concluding, a longer distance to be covered by an air/water particle results in a lower pressure.
This theory can be applied to the concept of a SFT and it already gives a first impression/hypotheses on the eventual shape of the SFT. In case the SFT is anchored to the bottom of the seabed, it is desired to have a lift force directed upward to generate tension in the tethers. Given a laminar flow, the top side of the SFT should have a more convex shape in order to have faster moving water here. The higher flow velocities corresponds to a lower fluid pressure, as stated by Bernoulli (equation 4.1). Consequently, the lower fluid pressure at the top results in a lift force upward as is desired.

4.2. Bernoulli's theorem for a Circle

The goal of form-finding was already stated in section 2.4, which is to find the shape that diverts the external loads in an axial way. Obviously, this method suits well with the first optimization target in optimizing the material use. When the loads are transferred via a normal force, all fibres of the material are loaded equally. However, the objective of this part of the research is to guarantee tension forces in the tethers. Why can form-finding be applied here?

According to Bernoulli's theorem, differential flow velocities between top and bottom are required in order to generate a certain lift force. These differential flow velocities can be achieved by moving the **deflection point** of a cross-section. The deflection point separates the flow from passing the SFT via either the top or the bottom. Hence, the deflection point directly influences the distance to be covered by a water particle. A water particle that needs to travel a larger distance will have a lower static pressure in the fluid. This is, as explained, due to the fact of continuity and Bernoulli's principle.
The deflection point is the location at the geometry where the angle between the approaching flow and the geometry's surface is equal to 90°. For a simple circle, this point is at the midline of the circle. See figure 4.2. The flow will locally converge at the leading edge of the submerged body and diverge at the trailing edge again. This causes a local flow increase which mainly determines the fluid pressure. Since the deflection point is at the midline, the convergence of flow at the top and the bottom is equal. For both the top and the bottom, a maximum flow velocity of $U_{max} = 2 \cdot U_{flow}$ [16]. Again, see figure 4.2. The flow velocity profile is symmetric, meaning that the pressure distribution along the SFT is as well. Concluding, a net lift force equal to $F_{lift} = 0\ kN$ (no turbulence accounted). This is in line with the lift coefficient $\overline{C}_L = 0$ for a circle according

to figure 2.15.

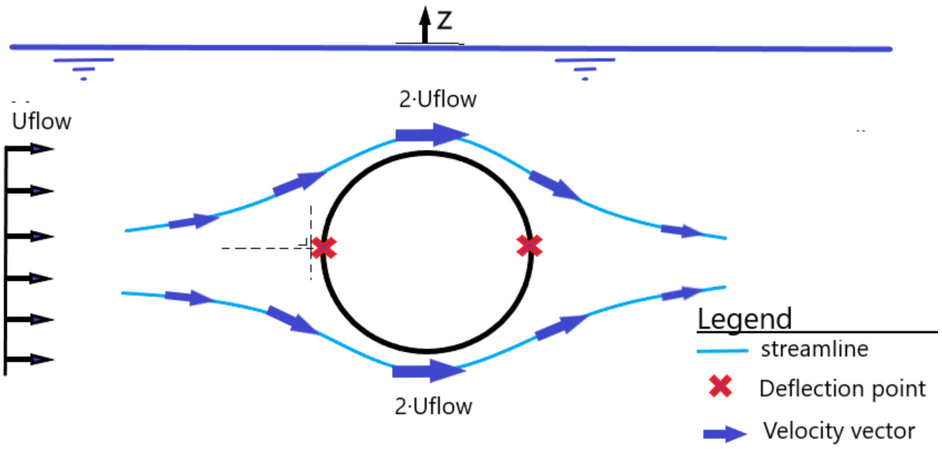


Figure 4.2: The velocity profile of a circular tube in a flow

In order to create differential velocities between the top and the bottom of the SFT, the deflection point must be moved. This can be done with the help of Form Finding which is explained in section 4.3.

4.3. Form Finding - Currents

In this paragraph, the form-finding method is applied for the hydrodynamic load due to currents. As mentioned in Chapter 2 already, the currents exert a certain flow force on the SFT, consisting of a part due to wind currents and a part due to the tidal wave. Besides, a force at the trailing edge of the submerged body is present as well. See paragraph 2.3.2. These flow velocities can be translated to a force by analyzing them as a drag force on the structure. Then, the pressure component must be taken to include in the FF-process. See figure 2.14.

Before starting with the Form Finding, some boundary conditions are applied to the problem. First of all, the (a)symmetry of the flow must be dealt with. The flow always encounters the SFT from one side at a time, whereas it can encounter the SFT from multiple sides during its lifetime. This is due to the motion of the tidal wave for instance. Therefore, it is desired to have a symmetrical shape of the SFT. However, since Form-Finding is directly related to the load profile, it is impossible to demand for a symmetric shape combined with an asymmetric load profile. Hence, the boundary conditions to the problem are:

- The flow load is applied as it would encounter the SFT from both sides **simultaneously**. This is done to gain a symmetrical shape. The evaluation of the lift force is determined as if the flow approaches the SFT from one side only, as it does in reality.

- The flow velocity is assumed to be uniform over depth. The small rate of decay over depth makes this assumption plausible. Moreover, an uniform flow velocity is required to solve for the velocity potential as becomes clear in section 4.5.

For some further illustrations on an asymmetrical flow profile, one is referred to Appendix C.

First, the model set-up is described. For a full elaboration on the model set-up for the current load, see Appendix B. Then, the output is analyzed and checked on validity by both a CFD and by an analytical solution written in Python. This analytical solution is based on the Panel Method [14].

Hypotheses

Before starting with the model, it is wise to think about a possible solution applying common physical sense. As the objective related to this part states, tension forces are required. In case of a SFT anchored to the bottom, the structure has to be pushed 'upward'. The load profile is a horizontal flow, which is assumed to be uniform. A possible shape which diverts the external loads into tension forces looks like the wing of an aeroplane as was explained in section 4.1. See figure 4.1. This is because the flow velocities at the top are expected to be larger, resulting in a lower pressure. Concluding, a shape with a flat bottom and a convexly shaped top is expected. Note that the use of internal space must be efficient as well, which means that the eventual shape is expected to be less flat than an aeroplane wing.

4.3.1. Parameters

This part of the model is also parametric, meaning that the parameters are listed first. The variables which are kept parametric are similar to the ones for the hydrostatic water pressure, supplemented by some parameters depending on the current loads. This is the flow velocity due to the tide, due to the wind and the total depth. Next to these parameters, the dimensionless drag coefficient C_D is also a parameter to bear in mind. However, this parameter depends on the shape of the structure. Hence, this is not a 'direct' parameter which has to be set in advance. It has to be changed during the form finding process, however.

4.3.2. Description of the model

Again, for a full description of the model, see Appendix B. Here, only a brief description is given on the model for the hydrodynamic loads. The beginning of the built-up of this model is basically the same as for the hydrostatic water pressure. The main difference is the type of load.

1. The geometry is created in the same way. A circle with a variable radius is the starting point, being divided into 'N' segments and nodes with index i. Note that the starting geometry is not of influence on the final shape, as proved by figure A.12.

2. The flow velocity is assigned to each node i. Because it is uniform, no differences have to be considered for different local depths along the SFT.

3. The loads on each node can be calculated according to the theory as described in paragraph 2.3.3.

$$F_{D,i} = \frac{1}{2} \cdot \rho \cdot C_D \cdot D_i \cdot u^2$$

4. Each segment diverts half of its 'length' to the adjacent node. The 'effective length' that each node absorbs is equal to variable D_i in the equation above.

5. Finally, the total loads which are required within FF are calculated by taking the pressure component from the horizontal drag force. See figure 2.14 and equation 2.23.

$$F_{Di,p} = F_{D,i} \cdot cos(\beta)$$

6. All elements are now inserted into the Kangaroo plug-in. This includes the geometry, the anchor point and the pressure component of the flow load $F_{Di,p}$.

7. Contrary to the model from Chapter 3, loops are not executed here. This is not required due to the different objective of this Chapter. Appendix B elaborates this further.

4.3.3. Output

First of all, the loads on the SFT can be visualized by displaying the load vectors acting on the nodes. Eventually, the shape will deform according to this load because Kangaroo functions as a mass-spring system with equal stiffness for the segments. The shape should therefore deform likewise the load profile. An overview of the load profile is shown in figure 4.3a.

It is clearly visible that the shape extends over its width. This is as expected, since the flow load is horizontal and acts in the direction of the width of the cross-section. Note again that only the pressure gradient of the horizontal drag force influences the deformed shape. The angle of rotation of each segment is of large influence on the actual load due to this drag force, as stated in equation 2.23. This results in the load profile as shown above.

When Form Finding is applied for the Hydrodynamic flow load, an elliptical shape is the result. Notice that the loads are reversely placed on the geometry. Since the approaching flow is assumed to be uniform, the load profile is symmetrical over the vertical. Therefore, the deflection point is still at the midline and no

differential velocities are expected between top and bottom SFT. The lift force is still equal to $F_{Lift} = 0\ kN$. On the contrary, the drag force (coefficient) is strongly reduced in this way. Consider the difference for C_D between a circle and a ellipse in figure 2.16.
For the load profile in Grasshoppers Kangaroo and the output of the shape, see figure 4.3 below.

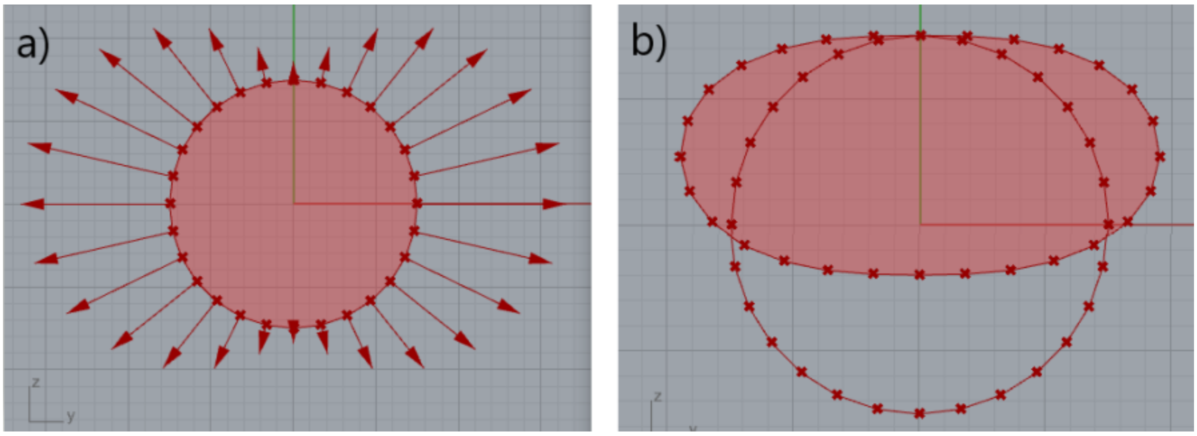


Figure 4.3: a) The load vectors due to currents on the SFT - b) the form-found shape due to this load

Although the large benefits considering the drag force, this still doesn't meet the requirements considering the objective of this Chapter. The lift force is still equal to 0 kN as just mentioned. The flow velocities at the top and the bottom of the SFT are equal (but lower than $2 \cdot U_{flow}$), resulting in a symmetric pressure distribution over the vertical. In the next section (section 4.4), an adapted version of this FF-process is shown which does result in differential velocities.

4.4. Geometry factor to guarantee tension

In this section, the model as described above is manipulated in such a way that tension in the tethers is generated. The model as described above still generates a shape with a lift force $F_{Lift} = 0\ kN$. However, the lift force is desired to be larger than zero in order to generate tension in the tether. The main question is how?

This has got to do with the deflection point of the cross-section. As already explained in section 4.1, the deflection point determines the distance to be travelled by water particles via the top and the bottom of the SFT respectively. Due to continuity, this distance is directly related to the velocity of the water particles. Consequently, the velocity determines the static pressure in the fluid according to Bernoulli's principle. Finally, the pressure distribution along the SFT's surface determines the lift force. In order to have a **positive** lift force, the deflection point must be moved **downward**.

The load vectors manipulate the cross-section in such a way, that a state of equilibrium is reached. The larger loads at the lower half of the (initial) circle cause larger deformations of the bottom. The deflection point moves downward, meaning that a larger distance is to be covered by the flow passing the SFT through the top. As a consequence, the flow velocities at the top increase and the local fluid pressure decreases.
In general, it can be said that:
A lift force directed upward is achieved by lowering the deflection point. A lift force directed downward is achieved by heightening the deflection point. This all has got to do with the local flow velocities and corresponding fluid pressures.

How this can be achieved is explained in the paragraph below.

4.4.1. Selecting the right load vectors

The load vectors that contribute to a positive lift force on the SFT can be amplified. First, these loads have to be selected. The vectors which cause a lowering of the deflection point are the ones required.
In the case considered here, the SFT is anchored to the seabed. The pressure component of the loads at the bottom half of the (initial) circle have a negative z-value. Hence, these loads are the ones to be amplified as

they cause a lowering of the deflection point. The process in Grasshopper is shown in figure 4.4 below. The load profile is as shown in figure 4.3a.

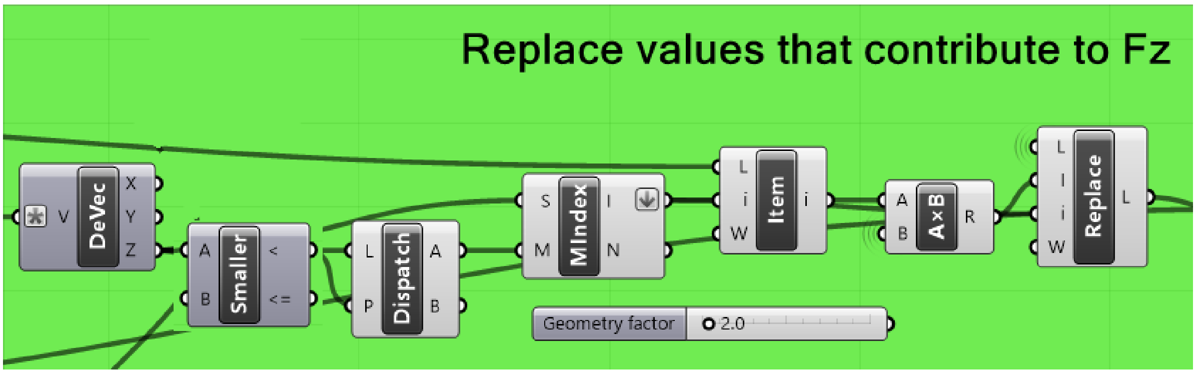


Figure 4.4: The amplification of the load vectors to move the deflection point in Grasshopper

First, the vectors are decomposed in a x-, y- and z-direction. The z-value is selected and subjected to a 'smaller than' gate. All vectors with a z-value smaller than 'zero' get a value 'True'. The load vectors with positive z-component get a 'False' value. With the 'Dispatch' gate, the indices of the values with a 'True-value' are selected. Remind that the 'True-values' are the ones which lower the deflection point and therefore positively contribute to the objective here. Then, these load vectors are selected and amplified with a **Geometry factor** which is set to 2.0 in this case. Finally, the original load profile list is adapted by replacing the load vectors with the 'positive' effect by the renewed, amplified load vectors. See figure 4.5 for an overview of the manipulated load profile.

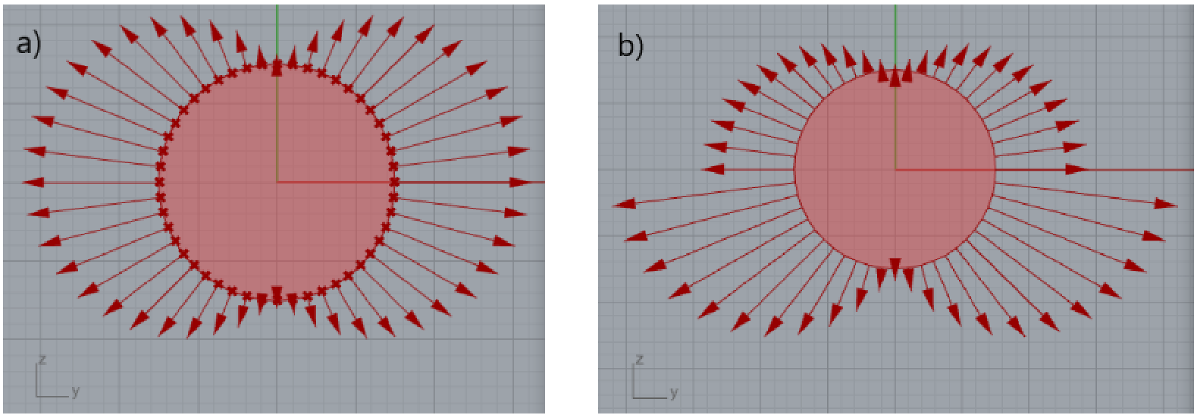


Figure 4.5: a) the actual load profile in FF b) the manipulated load profile with factor 2.0

As just mentioned, the aim of the manipulated load profile is to move the deflection point down. In the next figure, it is checked whether the deflection point is indeed moved downward compared to the original load case. To have a clear visual confirmation, the geometry factor is set to 5.0.

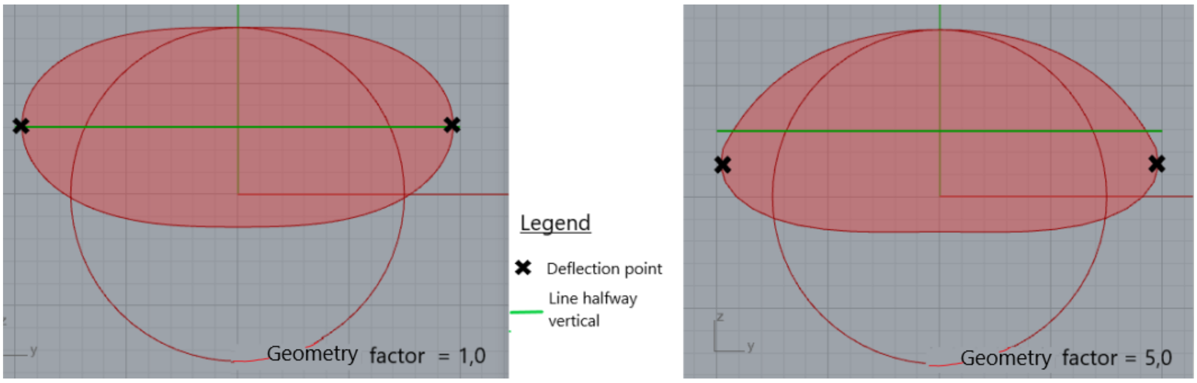


Figure 4.6: The deformed shapes from FF with amplification factors 1.0 (left) and 5.0 (right)

On the left an amplification factor of 1.0 is applied. In other words, this is just the original load profile as it would converge to a lift force of 0 kN. The deflection points, marked with black crosses, are basically halfway the vertical. This means that the flow velocity at the top and the bottom are equal. See figure 4.2.
On the right in figure 4.6 an amplification factor of 5,0 is applied. The deflection point is indeed lower than for the base case with factor 1,0. The green line separates the vertical halfway again. Now the deflection point is lower located within the cross-section and the water particles passing via the top require a larger velocity. A net upward force (tension in tethers) is expected. Moreover, note that the shape starts to look like described as in the hypotheses at the start of section 4.3. The top of the cross-section has a more convexly shape, meaning that the flow velocities along the top are expected to be larger. This corresponds to a lower pressure and a positive lift force.
In figure 4.7 a sketch is made which can be compared to figure 4.2. The main idea is that $X > Y$ such that the pressure at the top $p_{top} < p_{bottom}$. Therefore, a positive lift force is expected.

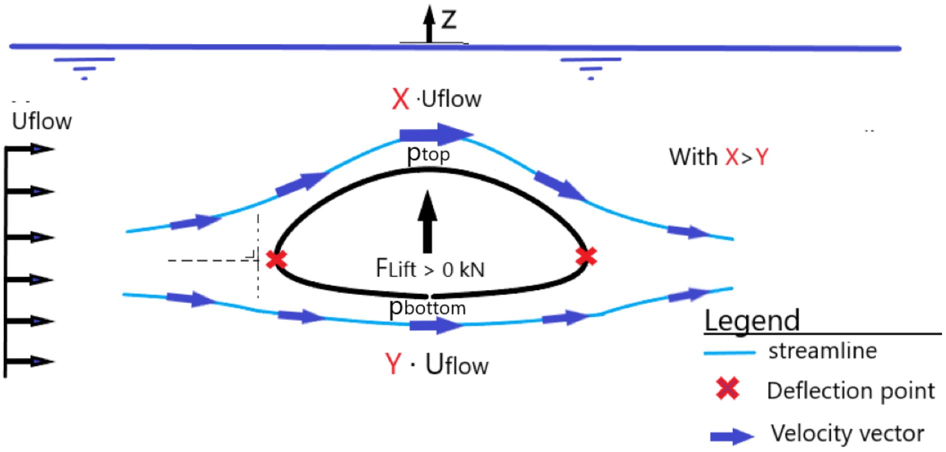


Figure 4.7: A sketch of the flow profile along the newly found cross-section

The magnitude of the lift force for the different geometry factors is discussed in section 4.5. But first, some elaborations are given on asymmetrical flows and shapes.

4.4.2. Asymmetrical flow

In the paragraphs above, the flow was modelled symmetrically. This implies that the flow encounters the submerged body from both sides simultaneously. Obviously, this cannot be the case since the water flows in one direction. However, the flow changes in direction every now and then. Especially the flow due to the tidal wave changes two or four times a day, depending on the location of the SFT (diurnal vs semi-diurnal tide). Hence, it is favourable to have a cross-section which is symmetric. When this is the case, the cross-section performs equally well on both load situations.
However, the demand of a symmetrical cross-section is prominently applicable in case of a single tube. When

a double tube SFT is desired, the main flow does encounter the structure from one side only. The asymmetrical load profile, where the flow encounters the SFT from one side with a suction factor of that same load at the lee side, might be of use then. For some additional information on the asymmetrical load profile, see Appendix C. Another situation where the asymmetrical shape might be desired is for certain estuaries, where a dominant flow is present from one side. The return flow can be much less significant, depending on the geographical circumstances.

4.5. Calculation of F_{Lift} and C_L

The next step is to determine the magnitude of the lift force on the SFT due to the flow. The velocity profile must be determined along the cross-section, after which the pressure distribution can be calculated using Bernoulli's principle. By integrating the pressure over the surface, a net force (both drag and lift) can be calculated.

This step can be executed in multiple ways. First of all, a CFD simulation is executed to calculate the velocity and pressure distribution along the SFT's surface. The results are quite accurate and give a good approximation of turbulence effects. However, the computation time is quite demanding. Therefore, a second option is applied. This is an analytical solution with the Panel Method. This method is already mentioned in paragraph 2.3.5. A large benefit of the analytical solution is that it is modelled within the open source programming code of Python and that it is computationally very efficient. However, the deficit of such an analytical solution is the incapability of dealing with viscous flow.

Eventually, the analytical solution for the panel method is calibrated such that it approximates the CFD-simulation accurately.

The shapes that are evaluated are generated by the model in Grasshopper as described in section 4.4. Again, these cross-sections are characterised by a GF. See figure B.6 in Appendix B for a relation between the cross-section and the GF.

4.5.1. CFD in Flow3D

First, the cross-sections are analyzed by a CFD model which has already been validated and verified. Therefore, it is assumed that these simulations give a trustworthy representation of reality. In the next section, the SPM is validated based on these outputs.

Concluding, this is a more advanced method to determine the lift force. But as just mentioned, the main deficit of a CFD is that it is computationally very demanding. Therefore, it is desired to tweak the Python script based on the CFD output. The result would be a very efficient model which approximates F_{Lift} and C_L accurately.

First, the CFD model must be set up which is already done by Zou. For a more detailed description of the set-up of the CFD, one is referred to [43]. For here, the settings of the model are:

- Shape with geometry factor 5.0 (see figure 4.6b)
- Uniform flow velocity = $V_\infty = 2\ m/s$
- Turbulence model = $\kappa\epsilon$-model

The $\kappa\epsilon$-model is used because it is known for its wide-range applicability and good convergence properties [41]. The output of the velocity field (figure 4.8) and the pressure field (figure 4.9) around the SFT are shown for the CFD-simulation below:

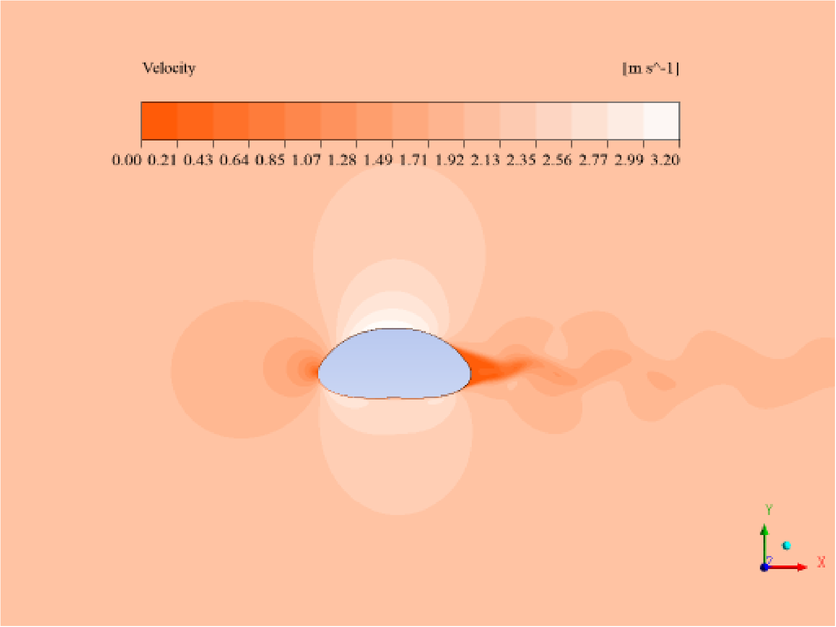


Figure 4.8: The velocity field from the CFD-simulation [43]

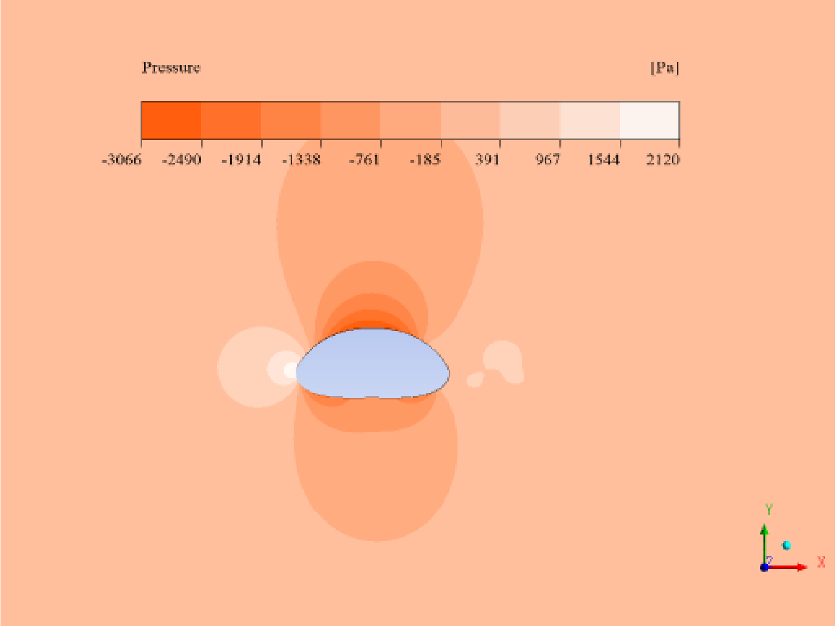


Figure 4.9: The pressure field from the CFD-simulation [43]

First of all, it is visible from figure 4.8 that the flow velocities over the top are larger than the flow velocities over the bottom. Concluding, $X > Y$ when making the connection to figure 4.7. It turns out that lowering the deflection point indeed results in larger flow velocities at the top. Bernoulli's principle, which states that larger flow velocities result in a lower static pressure in the fluid, turns out to be applicable here as well. When considering figure 4.9, the reference pressures at the top are indeed lower than at the bottom. The flow is 'pulling the tunnel up', resulting in tension in the tethers. Note that the reference pressure at the bottom is also negative, meaning that the flow at the bottom pulls the SFT down. This is because the flow velocities increased at the bottom as well, but in a lesser extent. Therefore, the net lift force F_{Lift} is still positive. See figure 4.10. Next the to the lift force, the drag force F_{Drag} is shown as well. The objective of this Chapter, a tensile force in the tethers, is achieved. Besides, the drag force is small, which is also beneficial.

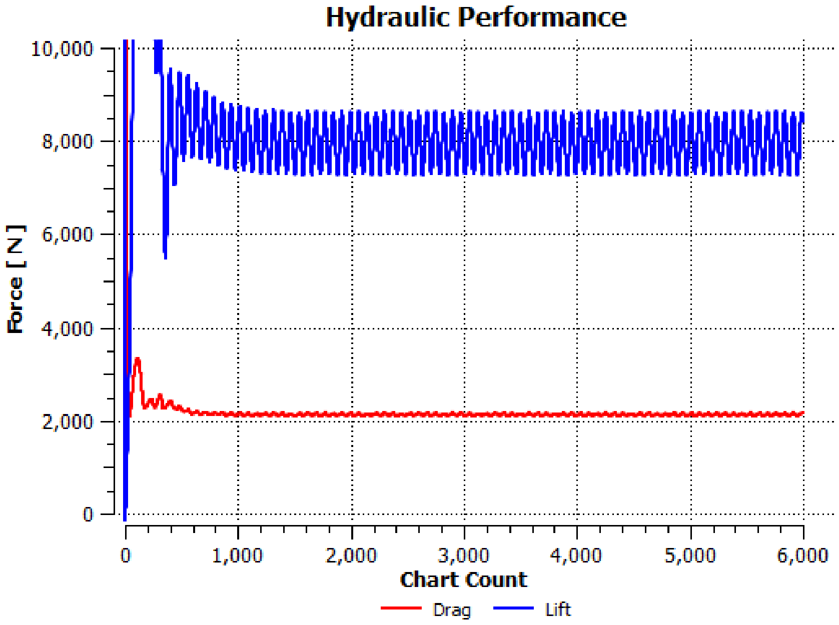


Figure 4.10: F_{Lift} and F_{Drag} from the CFD-simulation [43]

- $F_{Lift} = 8.0\ kN$
- $F_{Drag} = 2.1\ kN$

These values can be converted to a lift coefficient and a drag coefficient. This is done by the formula 2.21 which is applicable for both drag and lift. Note that for a circular cross-section and a horizontal approaching flow, the (mean) lift coefficient is equal to 0.

$$\overline{C_L} = \frac{F_{Lift}}{\frac{1}{2}\cdot\rho\cdot D\cdot V_\infty^2} = 0.43 \tag{4.2}$$

$$\overline{C_D} = \frac{F_{Drag}}{\frac{1}{2}\cdot\rho\cdot D\cdot V_\infty^2} = 0.11 \tag{4.3}$$

Where:

- $\rho = 1050\ kg/m^3$
- D = length unit perpendicular to flow = 8.8 m
- $V_\infty = 2\ m/s$

Some additional simulations from this CFD model are available and used to validate and verify the Panel Method in paragraph 4.5.3. If one is interested in the exact output of these CFD-simulations, see Appendix G.

4.5.2. Panel Method in Python

In this paragraph, the lift force F_{Lift} is determined by analytically solving the velocity potential for any arbitrary point P around or along the SFT. In the main report, only the main thoughts are shown behind this 'Source Panel Method (SPM)'. In Appendix E a full derivation is given for the velocity potential.
The velocity potential is a general function which can be used to calculate the velocity in any direction, by taking the corresponding partial derivative to this function. For instance:

$$\phi_p = velocity\ potential$$

$$V_x = \frac{\partial\phi_p}{\partial x} \tag{4.4}$$

$$V_y = \frac{\partial \phi_p}{\partial y} \tag{4.5}$$

This general velocity potential consists of multiple parts. Since all parts are linear functions, the principle of superposition can be applied to obtain the total solution for the velocity potential. In this research, the velocity potentials are considered for the uniform flow and the submerged body. The submerged body consists of so-called 'panels', which can be compared to the earlier mentioned segments in the Grasshopper model. The total velocity potential at any point p is equal to:

$$\phi_p = V_\infty \cdot cos(\alpha) \cdot x + V_\infty \cdot sin(\alpha) \cdot y + \sum_{j=1}^{N} \frac{\lambda_j}{2 * \pi} \cdot \int ln(r_{pj}(s_j) \cdot ds_j \tag{4.6}$$

Where:

- V_∞ = Uniform flow velocity
- α = angle of incoming flow w.r.t. horizontal
- λ_j = source/sink term of panel j
- r_{pj} = distance between point p and panel j
- s_j = panel length of panel j

The unknowns in velocity potential equation are the source/sink terms λ_j. Each panel, or segment as was defined earlier, has got its own source/sink term. These unknowns can be solved by setting the normal velocity at each panel j to zero. This can be physically explained by the fact that the flow can not go through the submerged body. The normal velocity at panel j can be found by taking the partial derivative of the velocity potential w.r.t. the normal direction for each panel. Again, for the full derivation see Appendix E. The system of equations that needs to be solved for λ_j is:

$$V_{n,i} = \begin{pmatrix} \frac{I_{11}}{2\pi} & \frac{I_{12}}{2\pi} & \cdots & \cdots & \frac{I_{1n}}{2\pi} \\ \frac{I_{21}}{2\pi} & \frac{1}{2} & & & \\ \vdots & & \ddots & & \\ \vdots & & & \ddots & \\ \frac{I_{n1}}{2\pi} & & & & \frac{I_{nn}}{2\pi} \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} -V_\infty \cdot cos(\beta_1) \\ -V_\infty \cdot cos(\beta_2) \\ \vdots \\ \vdots \\ -V_\infty \cdot cos(\beta_n) \end{pmatrix}$$

The integrals I_{ij} represent the geometrical influence on the flow pattern of panel j on control point i. The definition of these integrals is visible in Appendix E.

The output of this set of equations is a list of λ_j's, which can consequently be used to calculate the tangential velocity at the SFT's surface. Besides, the velocity potential for any arbitrary point p can be solved now. Note that these equations hold for laminar flow and do not take turbulence into account. Moreover, β is the angle between the approaching flow and the outward normal of a panel.

The solution for V_x and V_y at point p can be calculated according to:

$$V_x = V_\infty \cdot cos(\alpha) + \sum_{j=1}^{N} \frac{\lambda_j}{2\pi} \cdot \frac{s_j(x_p - x_j)}{(x_p - x_j)^2 + (y_p - y_j)^2} \tag{4.7}$$

$$V_y = V_\infty \cdot sin(\alpha) + \sum_{j=1}^{N} \frac{\lambda_j}{2\pi} \cdot \frac{s_j(y_p - y_j)}{(x_p - x_j)^2 + (y_p - y_j)^2} \tag{4.8}$$

Output

The output of the analytical derivation mentioned above can be visualized with a script in Python. To have a view upon this script, see Appendix F. The case to be visualized here is:

- Shape SFT is generated with geometry factor = 5.0
- Uniform flow velocity $V_\infty = 2.0\ m/s$

The output for the velocity profile is:

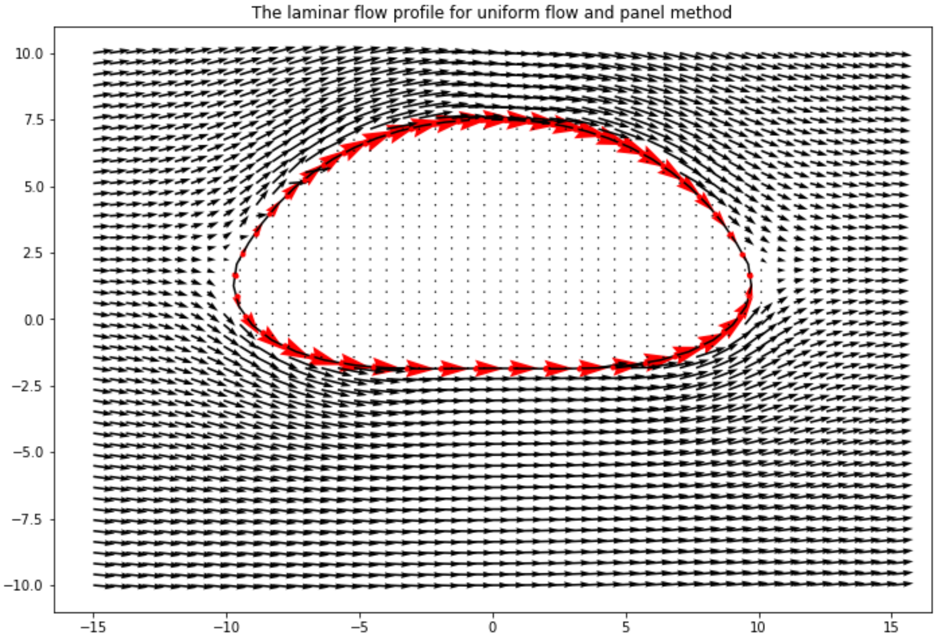


Figure 4.11: The flow velocity quiver plot along SFT with geometry factor 5.0

What can be seen from figure 4.11 is that the boundary condition for $V_n = 0$ is solved correctly. The flow does not go through the SFT as was imposed. What is also nicely visible is that the flow starts curving around the SFT well before it 'attacks' the structure. This is also in line with physical expectations.

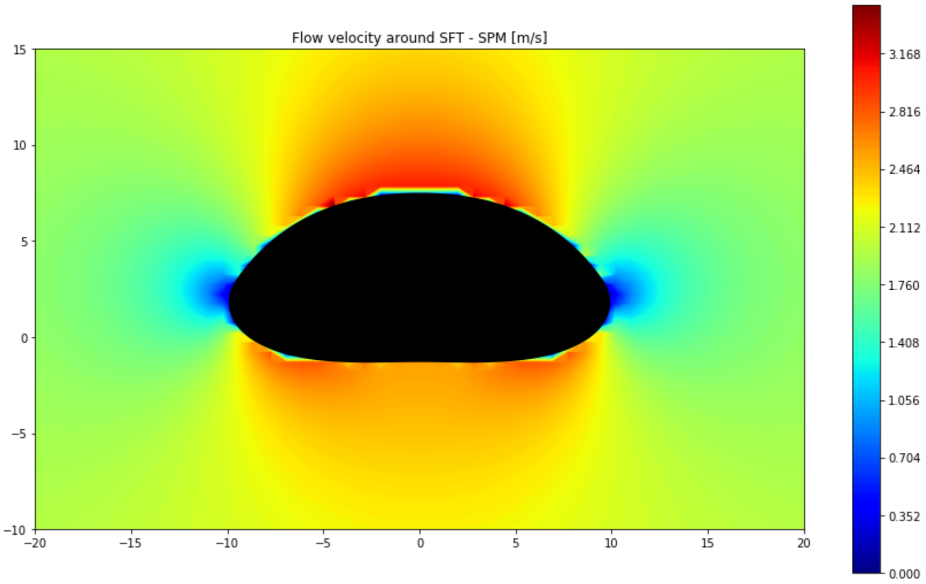


Figure 4.12: The flow velocity contour plot along SFT with geometry factor 5.0

In figure 4.12, a clear flow velocity increase at the top of the SFT is visible. This is in line with the expectations and the objective of this chapter.

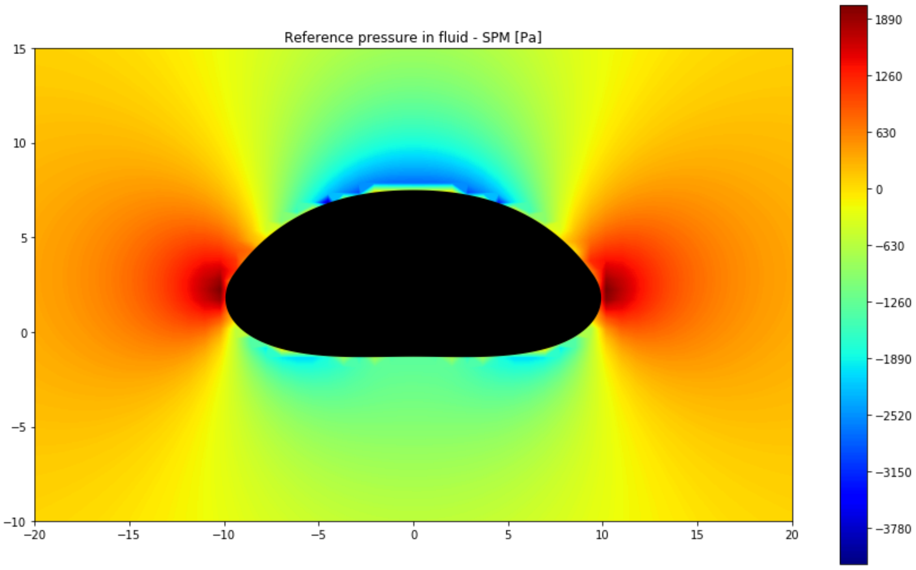


Figure 4.13: The pressure distribution contour plot along SFT with geometry factor 5.0

Finally, the reference pressure distribution can be calculated using Bernoulli's principle as stated in equation 4.1. What becomes visible is that the higher flow velocities at the top indeed cause a larger 'under pressure' at the top of the SFT. A positive lift force is expected.

When comparing figure 4.12 and 4.13 with figure 4.8 and 4.9 respectively, two striking things are visible. First of all, some similarities are visible in both the magnitude and the distribution of the velocity/pressure field. The maximum flow velocity for both models is about $V_{max} = 3.2\ m/s$ and located at the top of the SFT (as expected). Besides, the flow velocity at the bottom has increased as well, but with a lower factor. Notice that this is the condition $X > Y$ from figure 4.7 again. The flow velocity at the top has increased with 55% with respect to V_∞, while the velocity at the bottom has increased with 28%.

Secondly, it is clearly visible that the model in Python does not include turbulence effects. These occur at the trailing edge of the SFT.

The lift force and the drag force is in the SPM equal to 0 kN since the flow is perfectly laminar. It doesn't separate from the body which is physically not the case here. However, it gives a good approximation of the velocity/pressure field at **the leading edge** of the SFT. This is used in the next section where the SPM is calibrated.

4.5.3. Application of Panel Method

In this paragraph, the SPM as written in Python is evaluated in such a way that it can be applied as a quick and good estimation of the lift force for random geometries. First, the velocity distribution is analyzed more accurately to see where the major differences are present between the SPM and the CFD-model. See figure 4.14 for a comparison. Note that each x-coordinate is represented twice. One time for the velocity at the top of the SFT and one time for the velocity at the bottom. The line with the highest flow velocities corresponds to the top of the SFT, as indicated in figure 4.14. Note that the uniform flow velocity is still $V_\infty = 2\ m/s$ and that this is still the geometry with GF = 5.0. This is a more accurate comparison between figures 4.8 and 4.12.

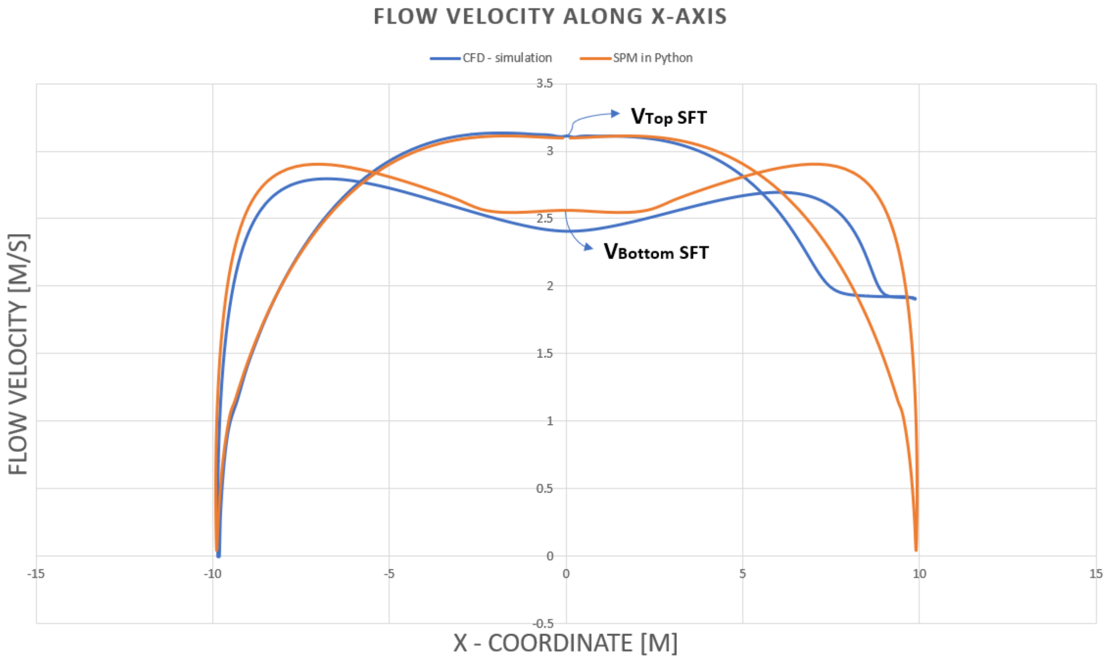


Figure 4.14: The velocity profile along the SFT for SPM and CFD-model

When evaluating the leading edge of the SFT, which is the left half in figure 4.14, it can be noted that the SPM is almost a perfect approximation of the CFD-model. Especially at the top of the SFT. Although the SPM starts deviating from the CFD-model at the left bottom corner of the SFT, it still approximates the CFD quite well. All values at the leading edge of the SFT, i.e. the left half, are within a 5% error margin. This is assumed to be a valid representation. However, at the trailing edge some strange things happen. The high intensity turbulence region is at the trailing edge, as could be noted from figure 4.8 as well. Since the SPM is not able to take turbulence into account, this is where the largest deviations take place. Moreover, the laminar flow profile is clearly visible by the symmetry of the pressure distribution for the SPM.

Concluding, the leading edge of the SPM gives trustworthy output to work further with. In the next part, the velocity at the top of the SFT and at the bottom of the SFT is retrieved from the SPM to base the lift force on. These values represent X and Y from figure 4.7. The ratio between the two is used to determine the lift force

$\mathbf{V_{Top}/V_{Bottom}}$ to determine $\mathbf{F_{Lift}}$

The ratio between the velocity at the top and the bottom respectively is used to determine the lift force. The lift force depends quadratically on the flow velocity, which means that a quadratic relation is expected between $\frac{V_{top}}{V_{bottom}}$ and C_L. A typical quadratic expression has got 3 unknowns with a shape of $y = a \cdot x^2 + b \cdot x + c$. In order to obtain the relation between F_{Lift} or C_L and $\frac{V_{top}}{V_{bottom}}$, three equations are required. As known, the mean lift force is equal to 0 kN for a circle. As became clear from figure 4.2 already, V_{top} and V_{bottom} are equal for a circle. This equation together with two more CFD-simulations are required to obtain the relationship between $\frac{V_{top}}{V_{bottom}}$ and F_{Lift}. For the output of these CFD-simulations, one is referred to Appendix G. In this Appendix, the values from table 4.1 are worked out. The 'equations' known now are listed in table 4.1 below. Besides, the lift coefficient is calculated as well based on equation 4.2. The force F_{Lift} is normalised per meter length in the longitudinal direction.

Geometry	Vtop [m/s]	Vbottom [m/s]	Ratio [-]	Flift [kN]	C_L [-]
1 (Circle)	4	4	1	0	0
2 (GF = 5)	3.10	2.56	1.21	8.0	0.43
3 (GF = 3)	2.99	2.59	1.15	5.6	0.30

Table 4.1: The three equations for the relation between V_{top}/V_{bottom} and C_L for $V_\infty = 2\ m/s$

Here, V_{Top} and V_{Bottom} are calculated with the SPM at the center of the SFT. These values are assumed to be valid as turned out by figure 4.14. The lift force is retrieved from the CFD-model, since the SPM is not able to compute these forces. Now, three calibration points (equations) are available. Consequently, a relation can be created between the flow velocity ratio and the lift coefficient. Again, a lift coefficient is chosen rather than a lift force as this is applicable to multiple situations. Finally, a fourth CFD-simulation is available to verify the relation just described. See figure 4.15 for an overview of the relation, the calibration points and the verification point. For a full overview of all values, see Appendix G. The CFD-output of the verification point is shown there as well.

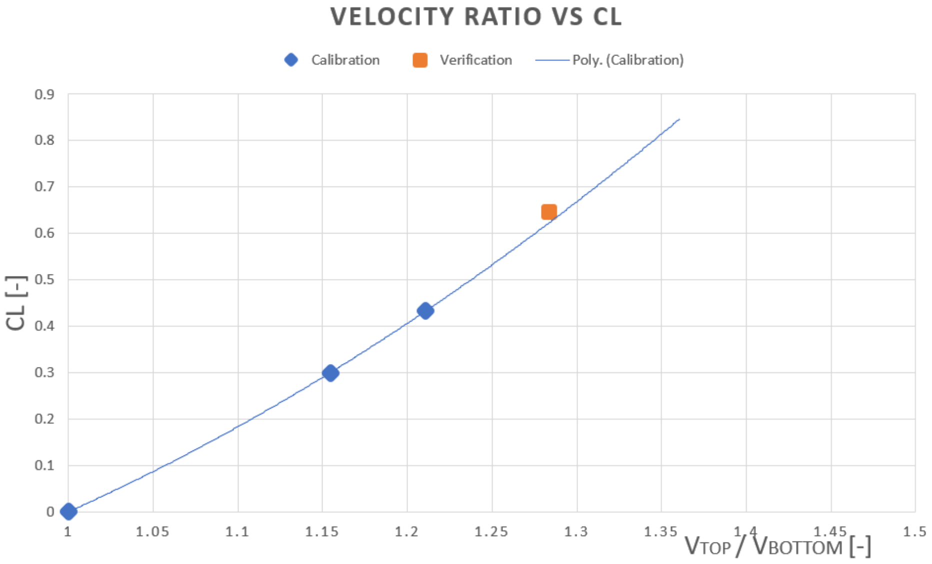


Figure 4.15: The relation between $\frac{V_top}{V_{bottom}}$ and C_L

The polyline which is shown in figure 4.15 is based on the blue dots, which are the calibration points as shown in table 4.1. The orange dot is a fourth point from the CFD-simulation. This point is required in order to assess the just found relation between the velocity ratio and the lift coefficient C_L. For this fourth point holds:

- $V_{Top} = 3.26\ m/s$
- $V_{Bottom} = 2.54\ m/s$
- $Ratio = 1.28\ [-]$
- $F_{Lift} = 11.7\ kN$
- $C_L = 0.648\ [-]$

The lift coefficient according to the fitted polynomial as plotted in the figure above is equal to $C_L = 0.622$. Concluding, this relation slightly underestimates the lift force. However, the differences are very small. Hence, this method is assumed to give reliable estimations of the lift coefficient based on the ratio between the velocity at the top and bottom of the SFT. The lift force can be plotted as a function of the Geometry Factor as well. This curve has a logarithmic character, because the shape converges to a stable solution for an increasing Geometry Factor. See figure 4.16.

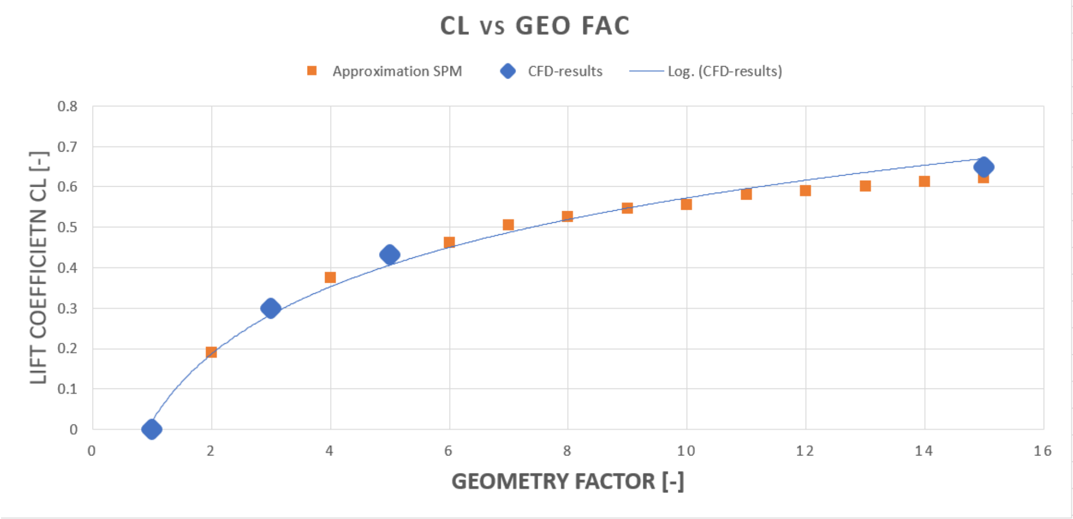


Figure 4.16: The relation between C_L and the Geometry Factor

Here, the blue dots again represent the lift coefficients as found by the CFD-simulations. As already mentioned, these values are assumed to be realistic since the CFD-model has been validated and verified already by Zou, P. [43]. The blue dotted line is a fit through these results. It is expected that the lift coefficient C_L for different geometry factors is on this line. In orange, the lift coefficients are shown as calculated by the relation found in figure 4.15. Again, for each GF the V_{Top} and V_{Bottom} are determined by the SPM. Then, this ratio is plugged into the new found relation with a lift coefficient as a result. As becomes visible from figure 4.16, all points are quite close to expected profile based on the CFD-results.

Conclusion
Concluding, the CFD-results are compared to the SPM script in Python. What became visible is that the leading edge of the SFT is modelled quite accurately by the SPM. On the contrary, the trailing edge lacked a lot of accuracy due to the incapability of dealing with turbulence by the SPM. Hence, it is decided to retrieve the velocities at the top and bottom of the SFT, at the center. These values are within a 5% error margin of the CFD.
The relation between the lift coefficient and the ratio of these velocities, $\frac{V_{Top}}{V_{Bottom}}$, is assumed to be quadratic according to equation 4.2. Once this relation was found and calibrated, it turned out that the verification point was valid as well. The result is a very quick estimation of C_L by applying the Source Panel Method for any random geometry.

4.6. Form Finding - Waves

Next to the hydrodynamic load due to currents, the waves also exert a dynamic force on the SFT. For a reference to the theoretical description of the wave behaviour, see section 2.3.4. With an increasing depth, the importance of waves decreases rapidly. This has got to do with the velocity profile of an orbital wave motion compared to the velocity profile of a flow. Compare figures 2.7 and 2.8a. However, the dynamic motion of the SFT due to wave forces must be investigated.

4.6.1. Wave loads - FF

As was done for the flow load, the form-finding process can be executed for the wave load as well. However, a wave load is a little more complex than a flow load.
The flow load could be modelled as a horizontal velocity profile, coming either from the left or the right in the governing situations. Concluding, two load profiles are present and still easy to deal with. The loads were applied on both sides of the SFT to gain a symmetric cross-section. On the contrary, the wave profile has a lot more load situations.

When the linear wave theory as described in section 2.3.4 is solved analytically in Matlab [18], the velocity and the acceleration profile can be generated over the cross-section. See figure 4.17 below for an overview of both the velocity and the acceleration at $t = 0\ s$ for a wave with characteristics:

1. Wave height $H = 4\ m$
2. Wave period $T = 7.0\ s$
3. Total depth $d = 100\ m$

Note, variable k from equations 2.16 and 2.17 is solved with the dispersion relationship:

$$\omega^2 = gk \cdot tanh(kd) \tag{4.9}$$

The local depth is shown along the y-axis in the figure below.

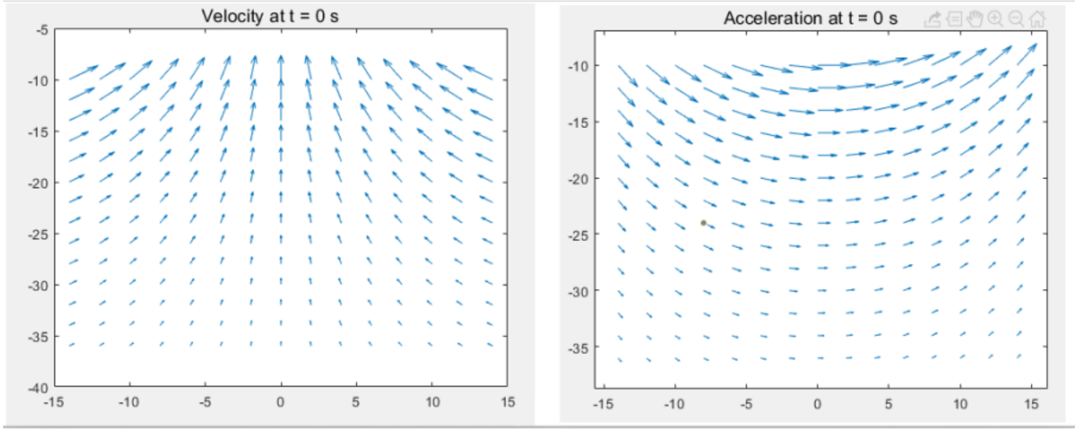


Figure 4.17: The velocity and acceleration profile over the cross-section

For this particular situation, the cross-section can be placed in the velocity and acceleration profile. Note that both the velocity and the acceleration have a contribution to the wave load, where the velocity is responsible for the drag force and the acceleration for the inertia force.
The profiles are imported in Grasshopper, after which it searches for the closest velocity/acceleration vector belonging to each node. Then, the closest vectors are used to calculated the wave loads according to equations 2.27 and 2.28. Again, only the pressure component of the loads are used in the form-finding process. In figure 4.18 below, the vectors including the starting geometry are shown on the left. On the right, the form-found shape is shown given the load profile.

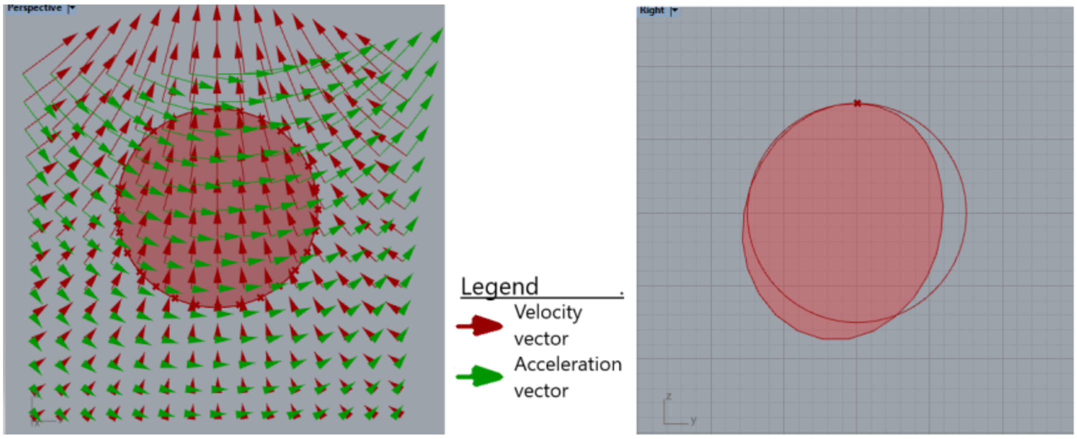


Figure 4.18: a) The starting geometry including load profile vectors, b) The FF-shape due to the load profile

Every split-second the velocity and acceleration profile changes, until the total wave has passed. It is not feasible to try and find an optimal shape which is suitable for the load situation which is present only a very small fraction of time. Hence, it is decided to exclude the wave load in the determination of the optimal cross-sectional shape. On the contrary, the wave loads are included in the calculation of F_{tether}, obviously. For a small impression of the 'optimal shape' during a fraction of the wave with characteristics mentioned above, see figure 4.19.

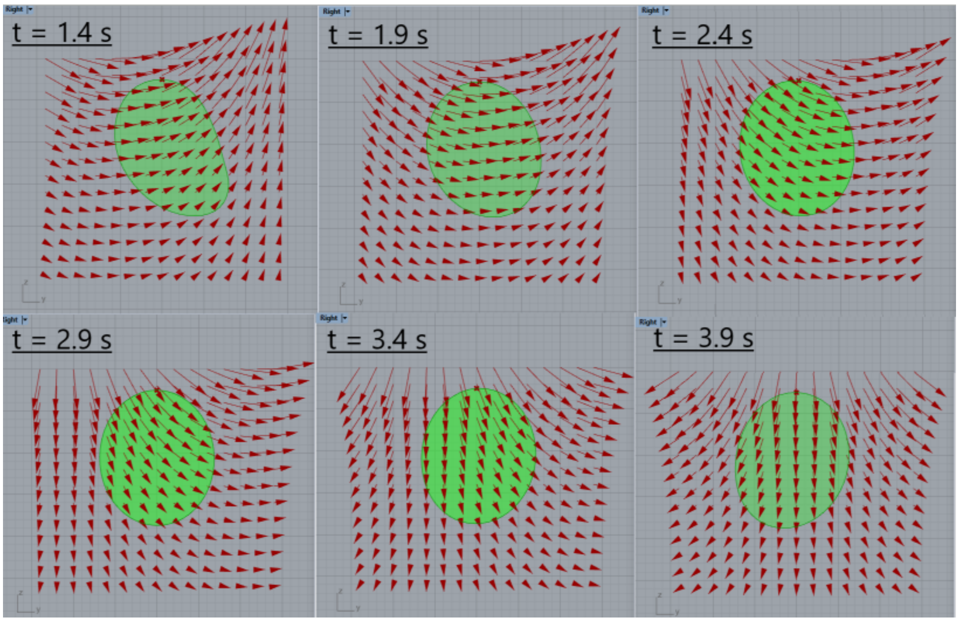


Figure 4.19: The form-found shape during several snaps of the passing wave

In the figure above, only the *velocity* vectors are shown.
Concluding, the wave load is not considered in the Form-Finding process for the SFT. On the contrary, it must be considered in the determination of the lift force on the structure. The wave force consists of an inertia component and a drag component, as was already presented by equations 2.27 and 2.28 in paragraph 2.3.4.

4.6.2. Lift force due to waves

The wave load depends among others on the cross-sectional properties. Since the (optimal) cross-section is not known yet, the wave load can not be determined as well. However, the magnitude of the wave forces are predominantly determined by the inertia component within the Morison equation. See equation 2.27 and 2.28. This is also visible in figure 4.21. The cross-sectional dependence within this inertia component is the cross-sectional area A. Since this value is bound within certain limits due to the required use of internal space, a fair estimation can be made for the order of magnitude of the wave force. **Important to emphasize here is that the wave load does not depend on the shape of the cross-section, but most dominantly on the cross-sectional area A.**
The magnitude of the wave force is required since it mainly determines the minimal required BWR. This is explained further in section 5.1. The following situation is used.

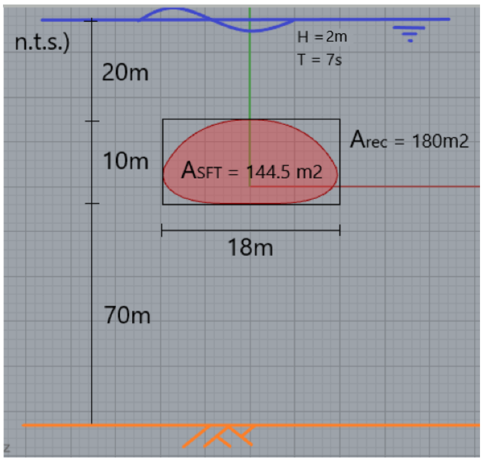


Figure 4.20: The situation for which the order of magnitude of the wave load is determined.

Besides, the script of Hemel [18] is used to get a first insight in the magnitude of the wave loads. Since this script in Matlab only handles rectangular cross-sections, a rectangle is fit around the FF-cross-section from section 4.4. The actual value for the wave load on the SFT is expected to be lower, since the cross-sectional area of the SFT is lower and the drag coefficients are expected to be lower as well. For the **rectangular shape** and the conditions shown in figure 4.20, the following is the result for the wave loads.

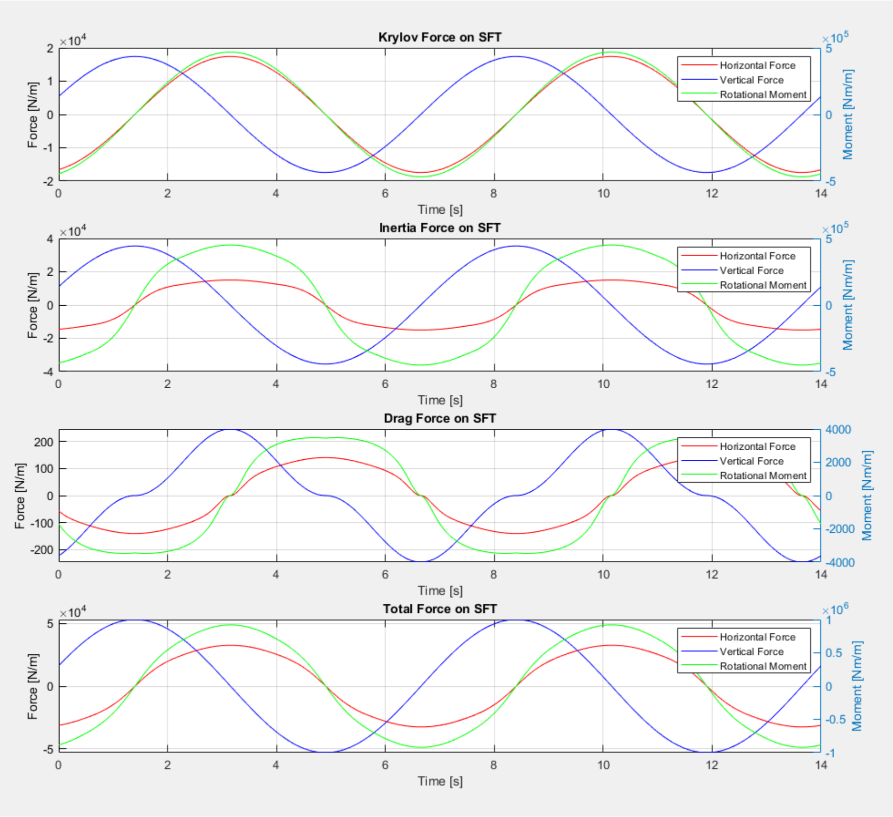


Figure 4.21: The wave loads on a representative rectangle at 20m depth according to Hemel [18]

The maximum vertical force on the rectangle is equal to:

- $F_{z,max} = 52.9 kN/m$

Note the unity of the force is equal to kN/m, which represents the force per meter length. Moreover, the dominance of the inertia component with respect to the drag component is strongly visible in figure 4.21.

The inertia component is represented by the Krylov force and the inertia force.
Similarly, an estimation of the wave load for the SFT can be made by making use of equation 2.28.

$$F_z = \rho C_I A \frac{du_z}{dt} + \frac{1}{2}\rho D C_{D,z} U_z \sqrt{U_x^2 + U_z^2} \tag{4.10}$$

Where:

- $\rho = 1050\ kg/m^3$
- $\omega = \frac{2\pi}{T} = \frac{2\pi}{7.0} = 0.90\ rad/s$
- $a = \frac{H}{2} = 1\ m$
- $C_I = 2.0$
- $A = 144.5\ m^2$
- $D = 18\ m$
- $C_D = 1.2$
- $d = 100\ m,\ z = -20\ m$
- $u_x = \omega a \frac{cosh[k(d+z)]}{sinh(kd)} \cdot sin(\omega t - kx)$
- $u_z = \omega a \frac{sinh[k(d+z)]}{sinh(kd)} \cdot cos(\omega t - kx)$
- $\frac{\partial u_z}{\partial t} = -\omega^2 a \frac{sinh[k(d+z)]}{sinh(kd)} \cdot sin(\omega t - kx)$

Such that:

$$F_z = -47.237 \cdot sin(\omega t - kx) + 341 \cdot cos(\omega t - kx)[N]$$

$$F_x = 47.237 \cdot cos(\omega t - kx) + 341 \cdot sin(\omega t - kx)[N]$$

When plotting this wave signal (at the same location 'x' within the wave) over time, the following is the result which can be compared to figure 4.21. Again, notice that the load is normalised per meter length of the SFT in longitudinal direction.

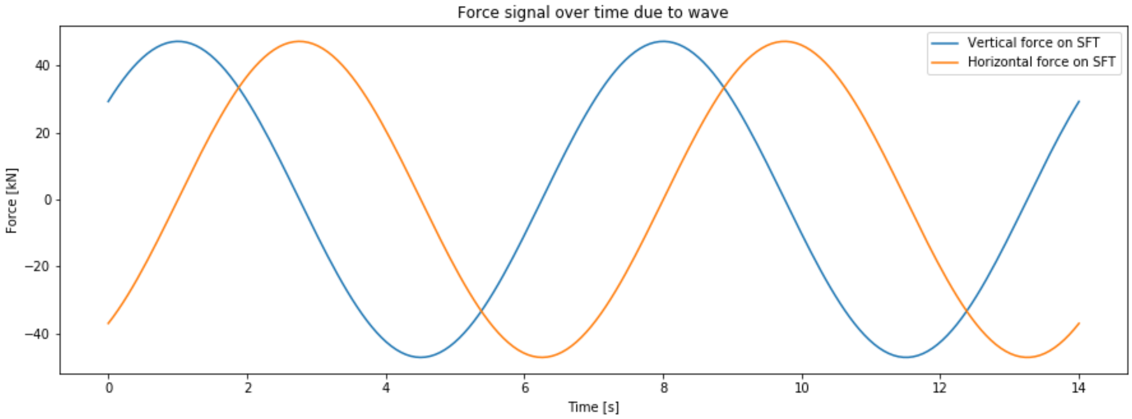


Figure 4.22: The wave load on the SFT according to LWT at 20m depth

From the figure and formula above, the maximum (vertical) force on the SFT is equal to:

- $F_{Lift,max} = F_{Drag,max} = 47.2\ kN/m$

As expected, the maximum wave load is slightly smaller due to a smaller cross-sectional area. This has a direct influence on the inertia component of the wave load. The order or magnitude of the wave load in this particular case is equal for both figures 4.22 and 4.21. The order of magnitude of the wave load is rather important, as becomes clear in section 5.1 and 5.2.

Dependence on depth
The values shown above are applicable for a local depth of 20 meter on top of the SFT. However, the velocity profile of a wave decreases strongly with depth. This is already visible in the orbital motion as shown by figure 2.8. Therefore, it might be beneficial to locate the SFT deeper. The wave loads decrease significantly. Contrarily, the hydrostatic water pressure increases which is unfavourable. The ideal depth of the SFT is always a consideration to be made by the engineer. For an overview of the **magnitude of the wave force** over depth, see the figure below.

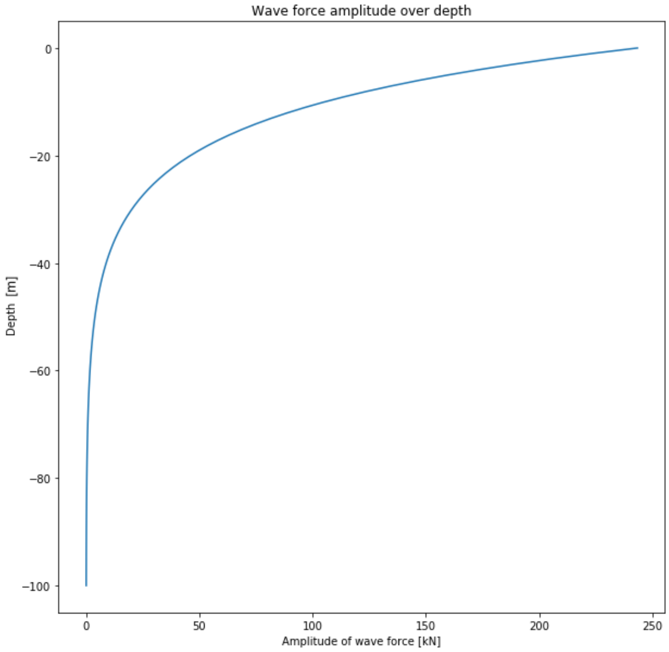


Figure 4.23: The amplitude of the wave force on the SFT vs depth

Concluding, the wave load is not taken into account within the form-finding process. This can be simply explained by the fact that the wave motion differs every split-second, meaning that the optimal shape does as well. However, the wave load is a very important load acting on the SFT, meaning that it cannot be ignored. The order of magnitude of the wave load can be determined per situation as shown above. The wave load must be absorbed by the BWR to guarantee tension in the tethers, as becomes clear in section 5.1.

4.7. Conclusion

As a conclusion of this chapter, some items are summarized. The objective of this chapter was to obtain a tension force in the tethers at all time. The hydrodynamic loads are responsible for the dynamic motion of the SFT. Therefore, these loads directly influence the tether force and it being tensile or compressive. The (hydro)dynamic loads taken into account in this research are the currents and the waves. The hydrostatic water pressure is of no (direct) influence on the tether force.

Currents
The current load is assumed to be uniform and horizontal. Therefore, a horizontal flow must generate a (vertical) lift force on the SFT. A comparison is made with the wing of an aeroplane, on which Bernoulli's principle is applicable. The idea is to generate larger flow velocities at the top of the SFT compared to the bottom of the SFT. Consequently, the static pressure in the fluid drops **more** at the top than at the bottom (Bernoulli). The result is a positive lift force.

In order to obtain larger flow velocities at the top, the **deflection point** of the cross-section is lowered. In case of a low deflection point, the water particles that travel via the top need to cover a larger distance with a lower pressure as a consequence. This is done using a **geometry factor** within the Form Finding process. The larger this GF, the lower the deflection point.

The current can encounter the SFT from both sides. Therefore, a symmetrical shape is desired. Within the FF-process, a symmetrical shape can only be generated for a symmetrical load profile. Hence, the load profile due to the currents is mirrored over the vertical. Note that the analysis of the drag/lift force is executed for a flow which encounters the submerged body from only one side, as would happen in reality.

Then, the flow velocity profile along the SFT (and lift force) can be analyzed with the Panel Method which is worked out in a Python script. Since the Panel Method only handles laminar flow, the output of the trailing edge is unreliable. The SFT is after all in the turbulent regime. The output from the Panel Method for the leading edge is reliable. Hence, these results are used to create a relation between the flow velocities $\frac{V_{Top}}{V_{Bottom}}$ and the lift coefficient C_L. This relation is calibrated and verified with CFD-results. The Panel Method can now quickly estimate the lift force for any random geometry. An optimization process can be executed very efficiently now.

Waves
As just mentioned, the FF-shape is directly related to the load profile. Since the wave load changes every split second, it is found to be unfeasible to take the wave loads into account in this process. The output would be a shape which is optimal during a very small fraction of its lifetime.
However, the wave loads can have a significant impact on the tether forces. This strongly depends on the depth of the SFT, as becomes visible from figure 4.23.

The magnitude of the wave force at the depth of a SFT is practically solely determined by the inertia component within the Morison equation. The cross-sectional dependence of this term becomes visible with variable A, the cross-sectional area. This area is mainly determined by the 'use of internal space'. How many lanes does the SFT need to accommodate? What is the minimum height required? Etc... Concluding, the wave load is barely dependent on the **shape** of the SFT but rather on the total cross-sectional area. This is another reason not to take the wave load into account during the FF-process.

However, it can not be ignored at all. The lift force due to waves determines the minimum BWR as becomes clear in the next chapter in section 5.1.

5
Combined model

In this Chapter, the two optimization targets are combined to find the optimal shape. Again, it is important to emphasize the difference between the hydrostatic and the hydrodynamic loads which each represent an optimization target. The hydrostatic water pressure, which is covered in Chapter3, aims for the optimization of the material use. Since the water pressure is dominant at larger depths, this load is solely used within this optimization target. The SFT is typically situated at depths ranging from 20 to 30 meter, making this assumption plausible. In short, the output of Chapter 3 covers the target:

$$Optimize\, use\, of\, material$$

On the contrary, the hydrodynamic loads are responsible for the second optimization target. This target demands for tensile forces in the tethers at all time. The hydrostatic water pressure, as the name already implies, doesn't influence the dynamic movement of the SFT. Therefore, it can be excluded in this part of the research. This was covered in Chapter 4. The loads due to currents are used within the FF-process, where a Geometry Factor is applied to obtain differential velocities between the top and the bottom of the SFT. A lift force being non-zero for a horizontal flow is the result. The wave loads are not included within the FF-process due to its large variability in time. Concluding, the output of Chapter 4 covered the target:

$$\Sigma F_{Lift} \geq 0$$

Given that the SFT is anchored to the seabed.

The output of Chapters 3 and 4 must be combined to obtain the 'optimal' cross-section which meets both optimization targets. This can be done in two ways:

1. Integrate the output for both targets in one solution, consisting of a concrete tube with an exoskeleton made of steel.
2. Make a compromise between the two targets. The design consists of only a concrete tube, which' shape is determined by a compromise within the FF-process.

The main benefit of the first option is that both optimization targets can be fully achieved, whereas the second option must relinquish some to both targets. The main deficit of the first option is the higher costs, since the final design is a composite of two different structures. Obviously, this doesn't apply for the second option which is a benefit. What option is desired in an actual design, is cost-dependent and must be determined by the client. In this research, both options are elaborated.
Finally, there is another requirement considering the cross-sectional shape. The SFT must be able to accommodate a 'x-way road' or a railroad regarding its internal space. This last criterion can be met by scaling the cross-section.

First, some load combinations are shown. Then, the wave loads as discussed in section 4.6 are handled and accounted for by a minimum BWR. Consequently, the two possible solutions as just mentioned are discussed. Finally, the cross-section is scaled considering its internal space.

5.1. Load combinations

All the loads mentioned do not occur at the same time continuously. Since different loads have different impacts on the optimization targets, it is important to create some insight in the different load combinations and which ones are governing. In table 5.1 below, the load combinations are shown for the loads taken into account in this research. As already mentioned in Chapter 2, the accidental loads are not taken into account here. For further research they must be considered.

Load / Load combination	**1**	**2**	**3**	**4**
Hydrostatic Water Pressure	X	X	X	X
Currents		X		X
Waves			X	X
Accidental loads	-	-	-	-

Table 5.1: The different load combinations

The first optimization criterion mentioned at the start of this chapter, where the material use is optimized, is mainly dependent on the hydrostatic water pressure. As becomes clear from table 5.1, this load is always present. Note, the deeper the SFT is located, the more dominant this load is. Concluding, the optimization of material use is assumed to be unaffected by different load combinations.

On the contrary, the optimization target for $\Sigma F_z \geq 0$ is affected by the load combinations. In load combination 1, where no hydrodynamic loads are present, the net vertical force should be solely determined by the BWR. No effects of currents or waves are expected. In load combination 2, only currents are present. As was the aim of this research, the currents cause a tension force in the tethers. Therefore, no additional measures have to be taken for this load combination. Load combination 4 includes both hydrodynamic forces, where the wave load may induce a negative impact on the lift force. However, the current load is present as well, which positively contributes to the requirement of a tension force in the tethers by its shape. Hence, load combination 3 is the governing one where the load due to currents is not present, while the design wave is passing over the SFT. The magnitude of this load can be determined according to section 4.6. In the example shown there, the amplitude of the wave is equal to 47.2 kN/m. The motion of the wave oscillates around 'zero', meaning that both the maximum and the minimum are equal to plus and minus the amplitude, respectively.

5.2. Required BWR

For the situation sketched in section 4.6 with a wave with wave height $H = 2\ m$ and period $T = 7.0\ s$, the force is equal to a sinusoidal signal with an amplitude equal to approximately $F_{z,max} = 47.2\ kN/m$. See figure 4.22. Note that this value changes per situation, obviously. Although the cross-section is optimized to convert the horizontal flow over the cross-section to a tension force in the tethers, this does not have any effect in this particular load combination. Therefore, the BWR is responsible to guarantee tension in the tethers and to prevent the SFT from making large movements. The BWR is desired to be as close to 1,0 as possible, to reduce the amount of material needed. See paragraph 2.3.7 for more information. The minimum BWR can now be calculated by taking the maximum wave load as a fraction of the buoyancy force, which depends on the cross-sectional area of the SFT. For example, the situation from figure 4.20 is used with an area $A = 144.5\ m^2$. The required BWR is then equal to:

$$BWR_{min} = \frac{B}{W} = \frac{A \cdot \rho_w}{A \cdot \rho_w - F_{z,max}} = \frac{144.5 \cdot 10}{144.5 \cdot 10 - 47.2} = 1.034$$

In this way, the minimum required BWR can be determined by first calculating the cross-sectional area of the found cross-section and the maximum wave load acting on this cross-section The Weight of the SFT is easily adaptable in terms of the amount of ballast. Note that the value for the BWR is still rather close to 1,0 as is desired. In current design thoughts, the BWR is approximately 1.3 (or 0.77 for the reverse anchoring method). Still, this is a main advantage compared to the former design thoughts on Submerged Floating Tunnels. Regardless of the solution type (option 1 or 2 from the introduction of this Chapter), the minimum BWR must be determined according to the method shown above with the inclusion of safety parameters.

5.3. Composite cross-section with exoskeleton

This section covers the first solution as described by the introduction of this Chapter. Both optimization targets are solved for separately, as elaborated by Chapter 3 and 4. The optimization of material use searches for the concrete cross-section which absorbs the water pressure axially. In this way, all fibres of the concrete material are used equally.

The target which aims for tension in the tethers is achieved by constructing an exoskeleton. The exoskeleton diverts the flow around the SFT, causing an under pressure at the top. A positive lift force on the exoskeleton is the result. Note that some small holes are created in the exoskeleton, meaning that it does not need to absorb the water pressure. These small holes are assumed to be of no influence on the flow along the SFT. The inner tube is attached to the exoskeleton, meaning that it is pulled in the 'right' direction. The exoskeleton can be manually designed in Grasshopper in terms of the following parameters:

- Chord length - the length of the 'wing'
- Tip height - the height of the tips at the leading and trailing edge
- Smoothness - Determines how gradually the exoskeleton curves around the inner concrete tube.

For an overview of these parameters and some changes between them, see figure 5.1.

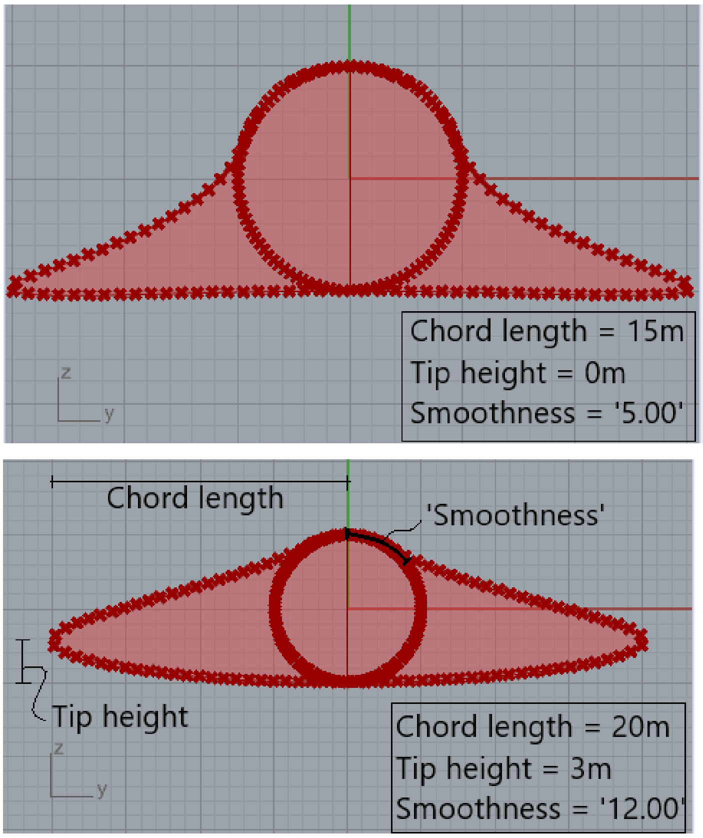


Figure 5.1: The definition of the parameters of the exoskeleton

The shapes as found in Chapter 4 still have blunt edges at the leading and the trailing edge. This would be required if the cross-section was constructed out of concrete. However, the exoskeleton can be created from steel, meaning that sharp edges are no issue. The sharp edges drastically reduce the turbulence effects, which is beneficial for the dynamic behaviour of the SFT.

The lift force on such a composite structure can be calculated more accurately with xFoil. See the next paragraph for the determination of the lift force over the exoskeleton.

For an overview of a total solution, see figure 5.2

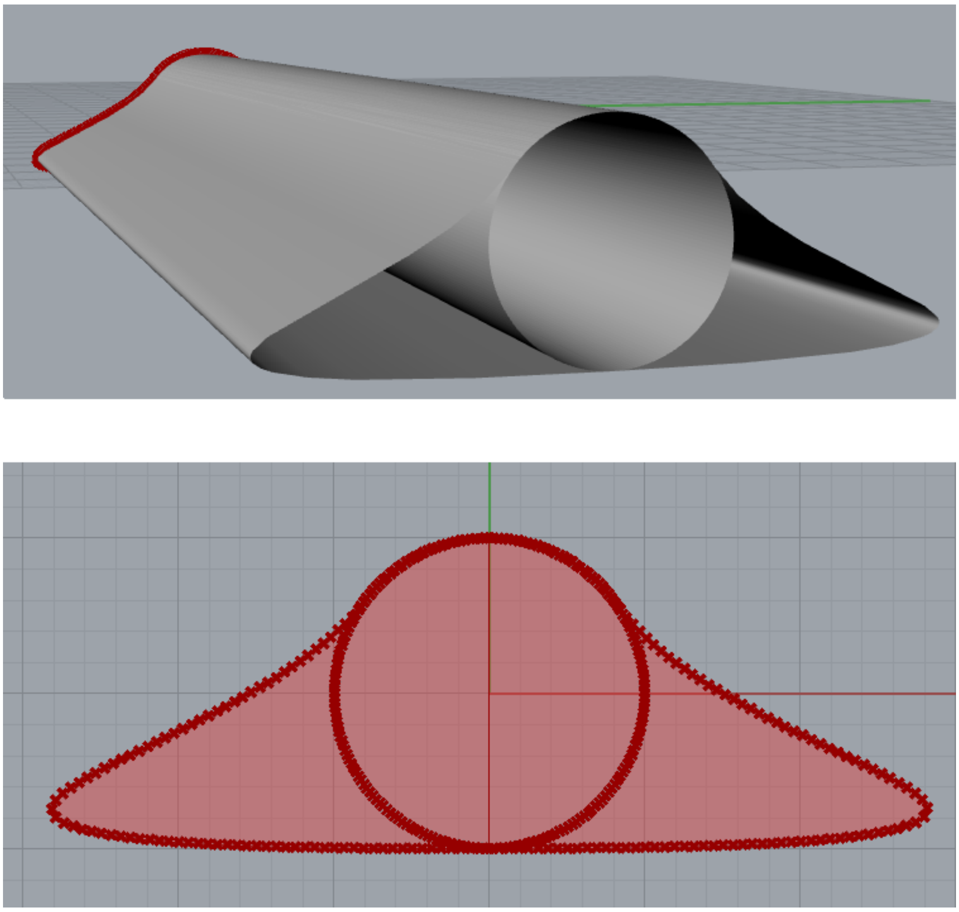


Figure 5.2: The integrated solution for both targets consisting of a concrete tube with an exoskeleton

5.3.1. xFoil - program

In this paragraph, the lift force due to the flow over the exoskeleton is calculated by applying the program 'xFoil'. The exoskeleton is of a different shape as shown in Chapter 4. The blunt edges can now be adapted to sharp edges, since the exoskeleton is made of steel while the tube is made of concrete. The sharp edges are desired to reduce turbulence effects and restrict the dynamic behaviour of the SFT. Moreover, it can reduce the drag force on the SFT.

xFoil is an interactive program, firstly designed by Mark Drela at MIT, to analyze the pressure distribution over isolated subsonic airfoils [40]. The program solves for the Panel Method as was described in paragraph 4.5.2 and Appendix E. Moreover, it can solve for the boundary layer equations by an approximation. The boundary layer as it builds up along a rough surface was already shown in figure 2.12. These equation are required to calculate the separation point from the flow, In this way, it can quickly estimate the pressure distribution for viscous flow over any random geometry. This approximation for the boundary layers and the wake are based on a two equation lagged dissipation integral and an envelope e^n transition [10].
Note, this approximation of the boundary layer equations is only valid for 'airfoil' like geometries, with sharp edges. Therefore, xFoil can not be applied to cross-sections as presented in Chapter 4.

The set-up of a xFoil model is described by Drela and Youngren [10]. The geometry of the exoskeleton can be exported from Grasshopper. Consequently, this geometry is imported in the Python script. Note that xFoil runs externally from Python. The 'xfoil.exe' file must be in the same folder as the Python Notebook. Besides, the same holds for all the files exported from grasshopper including the geometrical properties. The script to run xFoil from Python is retrieved externally [31].
The output from the xFoil function is most importantly a pressure coefficient. This coefficient is defined as:

$$CP = 1 - \frac{V_t{}^2}{V_\infty} \tag{5.1}$$

Where:

- CP = Pressure coefficient [-]
- V_t = tangential velocity at surface panel [m/s]
- V_∞ = Uniform flow velocity [m/s]

These pressure coefficients can be used to calculated the tangential velocity at the SFT's surface. Once the velocity is known, the pressure can be determined applying Bernoulli's principle. The final step is to decompose these pressure forces in a lift component (vertical) and a drag component (horizontal). The results are shown in the next paragraph.

5.3.2. xFoil - Flow over exoskeleton

In the previous paragraph, some background on xFoil is given. In this paragraph, the software is applied to the actual exoskeletons generated by Grasshopper. Moreover, it is validated to a CFD-simulation from Zou (BRONVERMELDING). The exoskeleton shown below has relatively long chords and thin ends. This is just for validation purposes to check whether the CFD-output and the xFoil output match. Important to note is that xFoil also runs within a couple of seconds, whereas the CFD-model in Flow3D requires a couple of hours. First, the output from the CFD is shown, after which the xFoil output is presented. The approaching flow is horizontal with an uniform velocity equal to $V_\infty = 2\ m/s$.

CFD-results

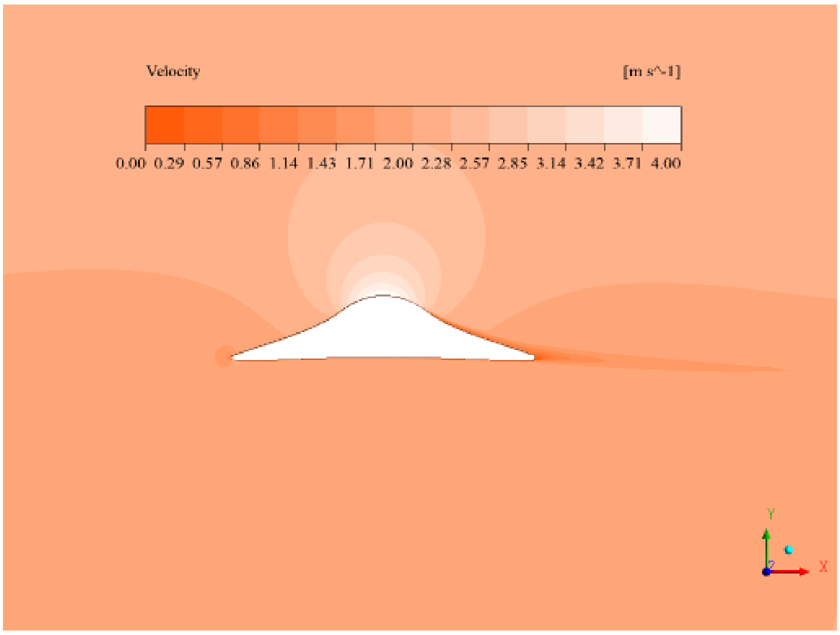


Figure 5.3: CFD - results for exoskeleton - Velocity profile[43]

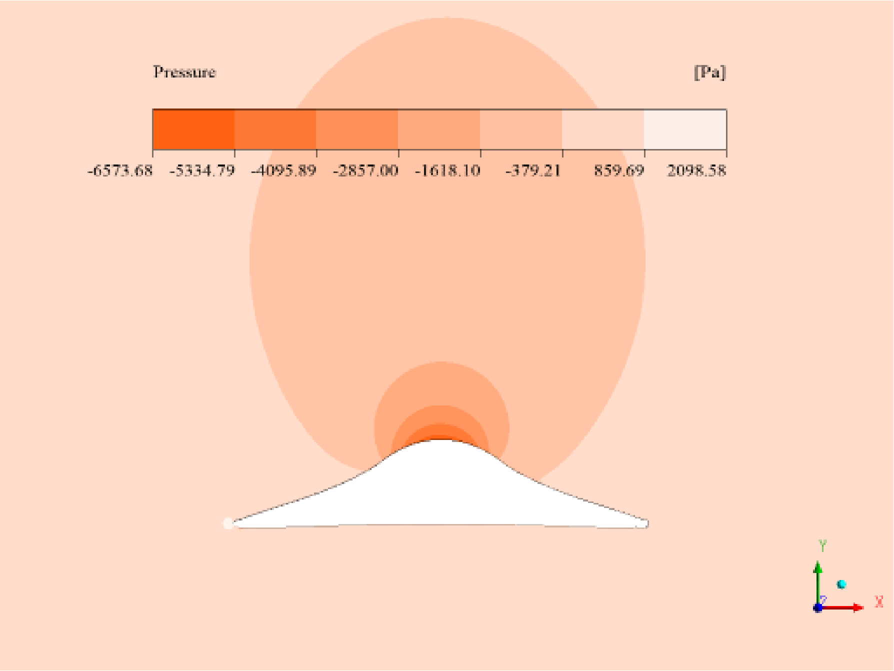


Figure 5.4: CFD - results for exoskeleton - Reference pressure profile[43]

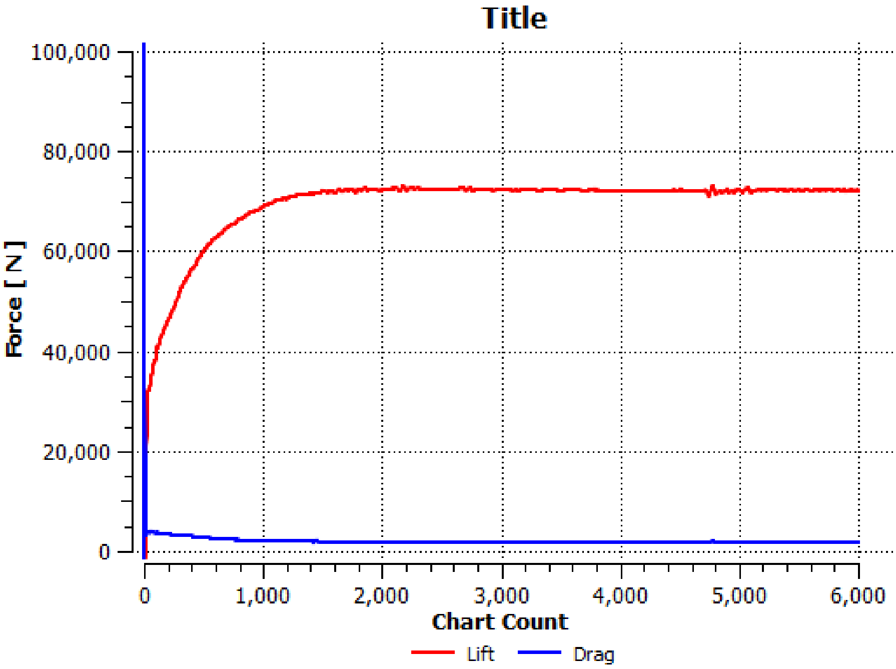


Figure 5.5: CFD - results for exoskeleton - Fdrag and Flift [43]

The first thing which becomes strongly visible is the reduction in turbulence. This was the aim of the exoskeleton with sharp edges. This is visible in the velocity profile, when one compares the trailing edge from figure 5.3 with figure 4.8. Moreover, the reduction in turbulence behaviour of the flow is also visible in figure 5.5. The forces F_{Drag} and F_{Lift} have only a small variation in time. This means that the turbulent component of the force is relatively small. Note that this is related to coefficient C'_L from figure 2.15.
When reading figure 5.5 above, the CFD-results for the lift force and drag force can be approximated by:

- $F_{Drag} = 2.5\ kN$

- $F_{Lift} = 72\ kN$

xFoil results

As already mentioned, xFoil runs externally from the Python code. If one is interested in the set-up of the xFoil simulation, one is referred to Appendix F. The definition of xFoil is shown in figure F.1, F.2 and F.3. In short, the most important settings to know:

- The exoskeleton from GH is imported as geometry
- The number of panels is set to 300
- The bunching parameter is 2, which means that the leading edge and trailing edge have a denser panel distribution.
- Viscous mode is on with $Re = 1e7$
- Angle of attack = 0 °
- $V_\infty = 2\ m/s$

As mentioned, the output of xFoil is a pressure coefficient per panel. The pressure coefficient can be converted to a velocity magnitude by equation 5.1. Then, the velocity can be related to the reference pressure according to Bernoulli by equation 4.1. Finally, these pressures are converted to a lift force and drag force by multiplying the pressure with the panel length and taking the correct component for lift and drag. The output is shown below.

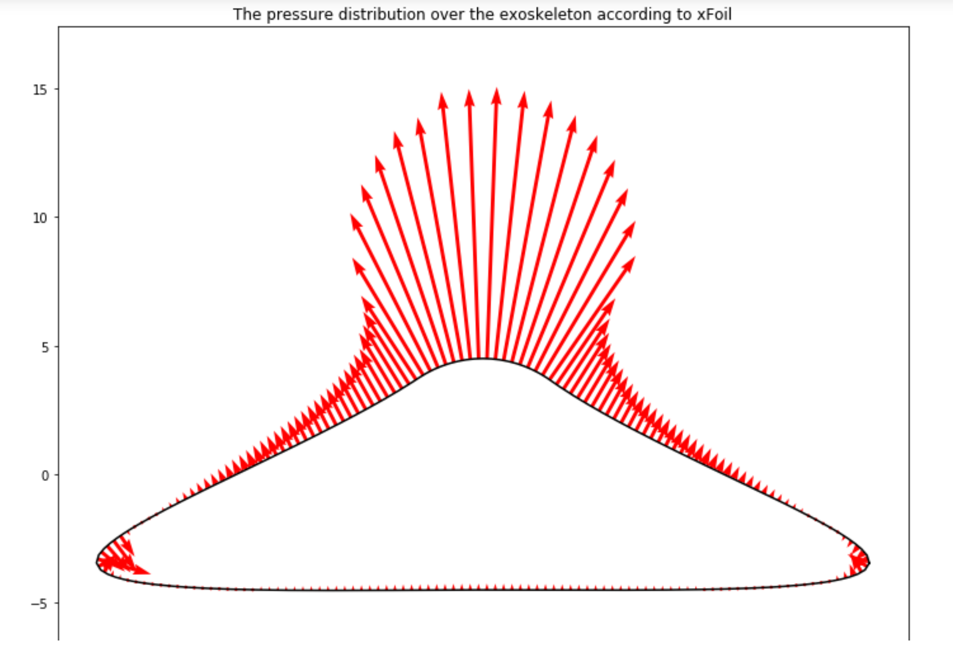


Figure 5.6: xFoil output for pressure distribution

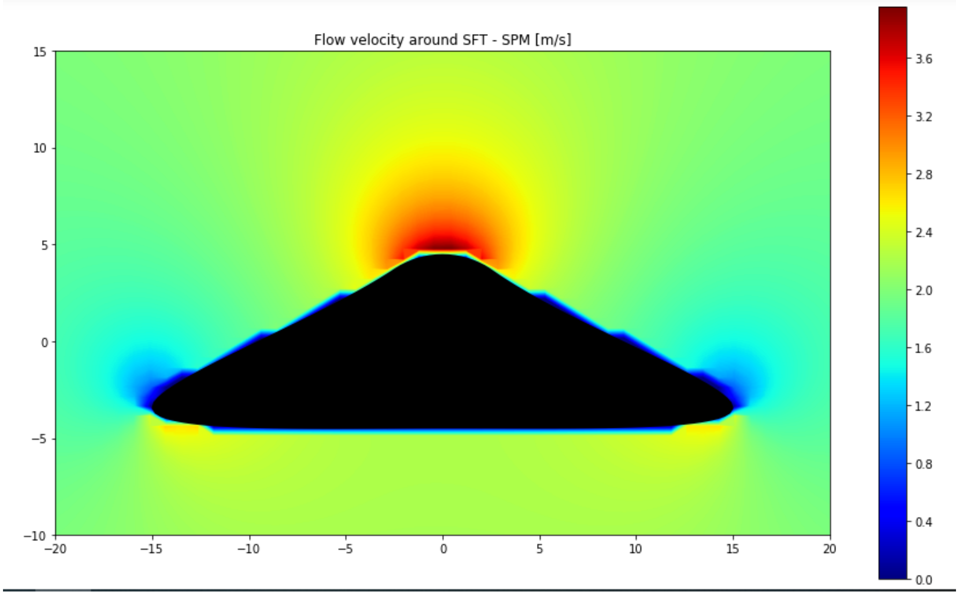


Figure 5.7: Velocity field - Panel Method

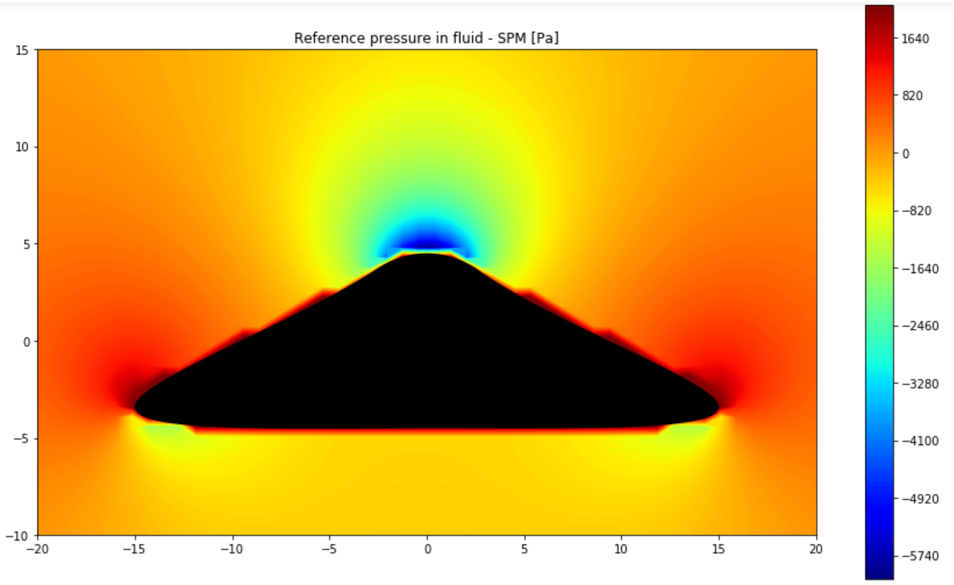


Figure 5.8: Pressure field - Panel Method

Note that only figure 5.6 is output from xFoil. It does not solve for the velocity potential for any arbitrary point, like the general PM does in the Python code. However, figure 5.7 and 5.8 are required to validate the xFoil output. xFoil might give unstable solutions for certain geometries. Hence, the user must always use its common physical sense to evaluate the xFoil output. The xFoil-results for the lift force and drag force are:

- $F_{Drag} = 2.66\ kN$
- $F_{Lift} = 69\ kN$

5.3.3. Conclusion

Concluding, both optimization criteria can be met by constructing a composite SFT. This composite is made of a steel exoskeleton which is responsible for the tensile force in the tethers. Secondly, the composite is made of a concrete tube within this exoskeleton being responsible for absorbing the water pressure and for the accommodation of the infrastructure. At larger depths, the optimal cross-section for the inner concrete tube tends to converge towards a circle. This is the output from the FF-process in GH. Within GH, the exoskeleton can be manually designed as well for parameters 'chord length', 'tip height' and 'smoothness'. See figure 5.1. Consequently, the flow over this exoskeleton is calculated in a Python code which externally runs the software xFoil. The output is a lift force and drag force which are validated with a detailed CFD-model. However, the xFoil simulation might give errors, which is why the output must always be compared with the pressure distribution according to the ordinary Panel Method. This can all be done within the same Python code.

5.4. Single cross-section - Combined Form Finding

In this section, the second option as mentioned in the introduction of this Chapter is worked out. The cross-section is searched which meets both optimization criteria as good as possible while still constructing only a concrete tube. Note that the cross-sections found in this section are always a compromise between the two different criteria.

5.4.1. Single cross-section

Within the FF-process, both targets are included. This means that both loads are inserted into the Kangaroo solver, which searches for a cross-section based on the deformations due to these loads. Since the water pressure load is too dominant over the flow loads, only a fraction of the water pressure is included. This is simply a percentage of the original water pressure load profile, where 100% is the actual load profile. The actual load profile for the water pressure within FF is visible in figure 3.4, for example. For the optimization target aiming for tension in the tethers, the GF is included similar to Chapter 4. Concluding, the two variables within the FF-process for a single cross-section are:

- Fraction of water pressure [%] - Optimize material use
- Geometry Factor GF [-] - Tension in tethers

For instance, the geometry for a fraction of water pressure equal to 10 % and a GF equal to 15 looks as below. The size of this cross-section is based on a circle with radius $R = 7.5\ m$. The eventual FF-shape is scaled back to the original perimeter of the starting circle.

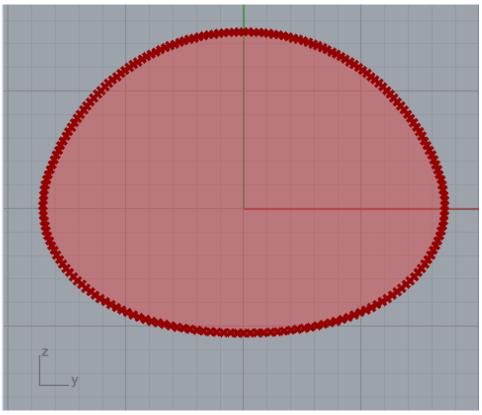


Figure 5.9: cross-section for a combined load profile

To evaluate the lift force on the cross-section, the relation between C_L and $\frac{V_{Top}}{V_{Bottom}}$ is applied as shown in Chapter 4. Note that xFoil can not be used here, since the cross-section has blunt edges. The approximation for the boundary layer equations in xFoil does not apply for this situation. The output of the Panel Method is as follows:

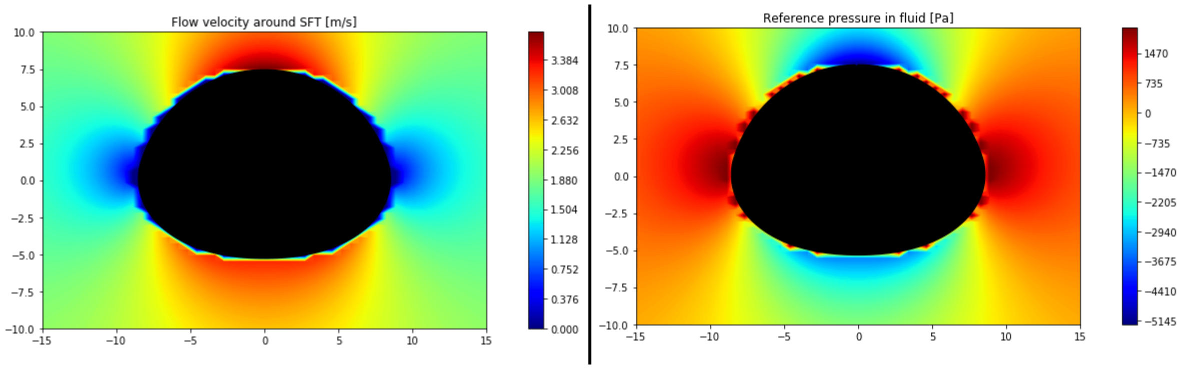


Figure 5.10: cross-section for a combined load profile

With:

- $\frac{V_{Top}}{V_{Bottom}} = \frac{3.75}{3.24} = 1.14$
- $C_L = 0.286$
- $F_{Lift} = 5.22 kN/m$

While the cross-section still tends to be quite circular with the characteristic of performing well at the water pressure. Hence, this might be an efficient solution as well.

The plug-in Karamba3D within GH is able to calculate the structural stresses within the cross-section Consequently, it optimizes for the thickness of the concrete lining based on the water pressure loads. For the cross-section shown above, the required thickness is equal to 82 cm. This value indicates how efficient a certain cross-section performs in terms of material use. This value can be used within an optimization process, as executed in the next Chapter.

5.4.2. Conclusion

Concluding, a single cross-section can be generated which is a compromise between the two optimization targets. The shape is found by form finding with the GF and a fraction of the water pressure, which represent the generation of a lift force and the optimization of material use respectively. By analyzing the cross-section with the panel method as coded in Python, the lift coefficient can be determined. Besides, Karamba3D is a tool which evaluates the efficiency of the cross-section in absorbing the water pressure. The two optimization targets can now be quantified to use in an optimization process. All of these steps can be executed within a couple of seconds, making the optimization process efficient. This is executed in the next Chapter 6.

6 Test case

In order to show what the result of a random case would be, a quick optimization process is executed here. The input parameters are taken as realistic values to give an impression of what the ideal cross-section of an actual SFT might look like. First, the steps to be executed are enumerated. Then, the single cross-section is presented after which the composite cross-section is shown, as elaborated in Chapter 5. For the actual design, a cost-benefit analysis must be executed to determine which option is 'the best'. Since this goes beyond the scope of this research, only the cross-sections are delivered here. Finally, some impressions are shown on what the SFT might look like in reality.

6.1. Action plan

In this section, the steps to be performed for the optimization process are listed.

1. Select input parameters, including a certain desired lift coefficient C_L
2. Single cross-section
   (a) Select a range for the Geometry Factors to be investigated
   (b) Select the step size of the $\%HWP$ from 0% to 100%
   (c) Evaluate each combination of $\%HWP$ and GF on lift coefficient C_L by the Panel Method in Python
   (d) Evaluate each combination of $\%HWP$ and GF on structural efficiency by Karamba3D (thickness concrete lining)
   (e) Select the optimal combination and check for use of internal space. If not sufficient, scale the cross-section
   (f) Calculate cross-sectional area A
   (g) Determine maximum wave load and minimum BWR_{min}
3. Composite cross-section
   (a) Loop the FF-process for the water pressure to retrieve the optimal shape of the inner concrete tube.
   (b) Manually design the exoskeleton with respect to chord length, tip height and smoothness.
   (c) Evaluate the lift force in Python with xFoil. If not sufficient, manually adapt exoskeleton.
   (d) Calculate cross-sectional area A
   (e) Determine maximum wave load and minimum BWR_{min}
4. Perform a cost-benefit analysis to select 'the optimal solution'.

6.2. Input parameters

The input parameters are listed below

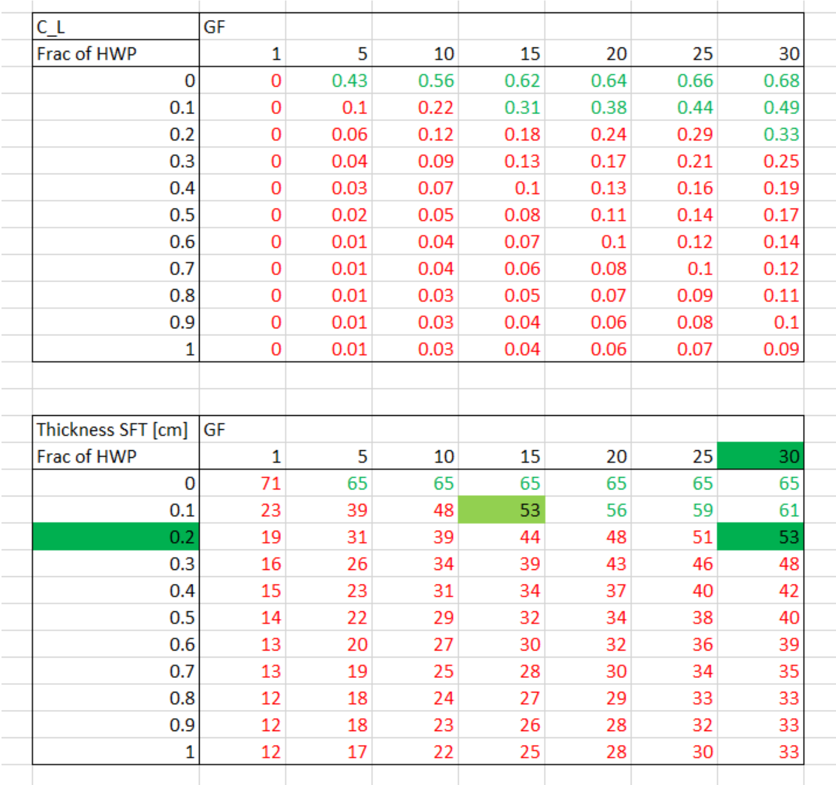
1. The SFT must accommodate a 2-lane highway A1 CAT with 3.5m lane width. For bored tunnels, a rule of thumb for the indicative diameter of the tunnel would be $D = 11.7\ m$ [33]. Concluding, for a circular cross-section the Radius $R = 5.9\ m$. This is a first indicative for the perimeter of the SFT. In the end, the total solution might have to be scaled according the lane width and internal height. This is because the eventual output is not necessarily a circle.
2. Fluid density $\rho = 1050\ kg/m^3$
3. The depth of the SFT is 20 meters
4. Total depth = 500 m
5. Flow velocity due to wind at surface $V_w(0) = 1\ m/s$
6. Flow velocity due to tide at surface $V_t(0) = 1\ m/s$
7. Maximum wave height $H_{max} = 4\ m$
8. Peak period $T_p = 8\ s$
9. **Objective: Create a cross-section with a lift coefficient C_L equal to 0.3, whilst optimizing material use**

The required lift coefficient can be retrieved from a safety analysis within a probabilistic design. As already noted in the scope (section 1.3), this research does not include probabilistic design. Hence, this value is assumed to be within the safety limits. The value of $C_L = 0.3$ is chosen because it is significantly larger than the expected drag coefficient C_D, which limits rotational effects of the SFT. The value of C_D is expected to be around 0.1 according to figure 2.16. Again, this case is just for illustrational purposes to show what the optimal cross-section approximately looks like.

6.3. Single Cross-Section

First, the single cross-section is worked out as shown in section 5.4. This is a solution compromising between the two optimization targets. These targets are the optimization of the material use and the tensile force in the tethers, represented by the fraction of water pressure and the Geometry Factor (GF) respectively. Note that a larger GF results in a larger C_L, whereas a larger fraction of the water pressure within the FF-process yields a more efficient shape with respect to the material use. The efficiency of material use is expressed with the 'required thickness' of the concrete lining. The Karamba plug-in within Grasshopper is able to calculate the stresses on the SFT and to optimize this to the thickness of the concrete lining. This thickness is not the 'actual thickness' required, because safety factors must be included in the design. However, it does indicate the efficiency of the cross-section relatively. The lift coefficient C_L can be calculated with the Panel Method as explained in paragraph 4.5.2. This code runs from Python.

Within the optimization process, steps of 5 are taken between the GF's in order to highlight the differences and to execute the process faster. The range within this analysis between the GF's is from 1 to 30. The fraction of water pressure taken into account is from 0 to 100%, with steps equal to 10%. The output of the optimization process is shown in figure 6.1 below. The C_L table represents the optimization for tension in the tethers, whereas the thickness of the SFT lining represents the use of material.



C_L	GF						
Frac of HWP	1	5	10	15	20	25	30
0	0	0.43	0.56	0.62	0.64	0.66	0.68
0.1	0	0.1	0.22	0.31	0.38	0.44	0.49
0.2	0	0.06	0.12	0.18	0.24	0.29	0.33
0.3	0	0.04	0.09	0.13	0.17	0.21	0.25
0.4	0	0.03	0.07	0.1	0.13	0.16	0.19
0.5	0	0.02	0.05	0.08	0.11	0.14	0.17
0.6	0	0.01	0.04	0.07	0.1	0.12	0.14
0.7	0	0.01	0.04	0.06	0.08	0.1	0.12
0.8	0	0.01	0.03	0.05	0.07	0.09	0.11
0.9	0	0.01	0.03	0.04	0.06	0.08	0.1
1	0	0.01	0.03	0.04	0.06	0.07	0.09

Thickness SFT [cm]	GF						
Frac of HWP	1	5	10	15	20	25	30
0	71	65	65	65	65	65	65
0.1	23	39	48	53	56	59	61
0.2	19	31	39	44	48	51	53
0.3	16	26	34	39	43	46	48
0.4	15	23	31	34	37	40	42
0.5	14	22	29	32	34	38	40
0.6	13	20	27	30	32	36	39
0.7	13	19	25	28	30	34	35
0.8	12	18	24	27	29	33	33
0.9	12	18	23	26	28	32	33
1	12	17	22	25	28	30	33

Figure 6.1: The optimization process

First of all, note that an increasing 'fraction of HWP' included in the FF-process results in lower concrete thicknesses, as expected. Similarly, a larger 'GF' yields larger lift coefficients C_L. Both parameters support to their objective.

The main demand for the cross-section is to have a lift coefficient C_L equal to 0.3 at least. Given this requirement, only a few combinations are left. These combinations of GF and fraction of HWP are marked in green in the top table of figure 6.1. Hence, it must be one of these combinations. Consequently, the material use must be optimal given the lift coefficient of at least 0.3. When looking at the bottom of figure 6.1, the possible combinations are marked in green as well. From these combinations, two of them have the lowest required thickness of 53 cm according to the Karamba structural analysis. The one with $GF = 30$ and fraction of HWP = 0.2 has the largest lift coefficient of the two, meaning that this is considered as 'the optimal' in this case. Obviously, both options are very similar as shown below.

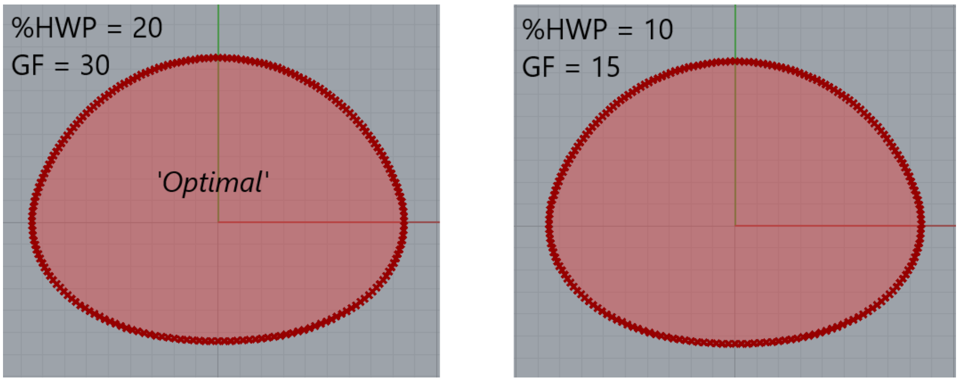


Figure 6.2: The two 'optimal' cross-sections

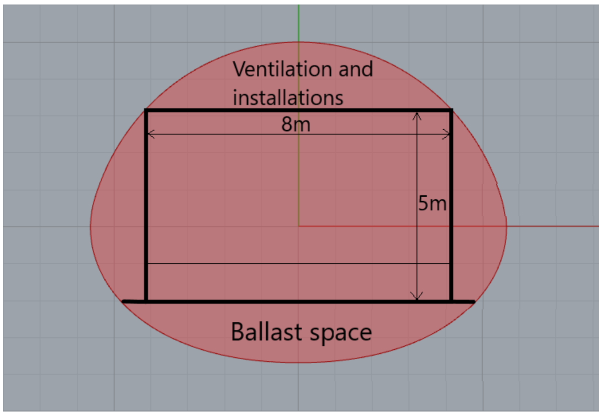


Figure 6.3: The optimal cross-section with use of internal space

The internal space is enough to construct a 2 lane highway with each a lane width of 3.5 meters and 0.5 meters safety at each side. Moreover, the minimum height of 4.5 m is also achieved with the 5 meter height. Concluding, this cross-section does not need to be scaled in order to accommodate the infrastructure for its purpose. The velocity and pressure profile of the selected cross-section are as shown below.

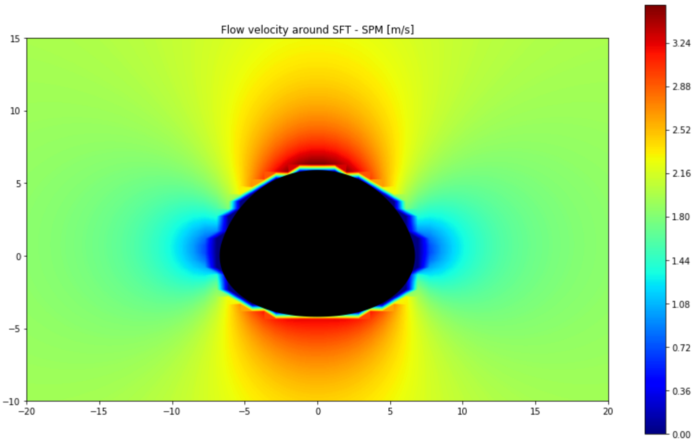


Figure 6.4: The velocity profile for the single cross-section

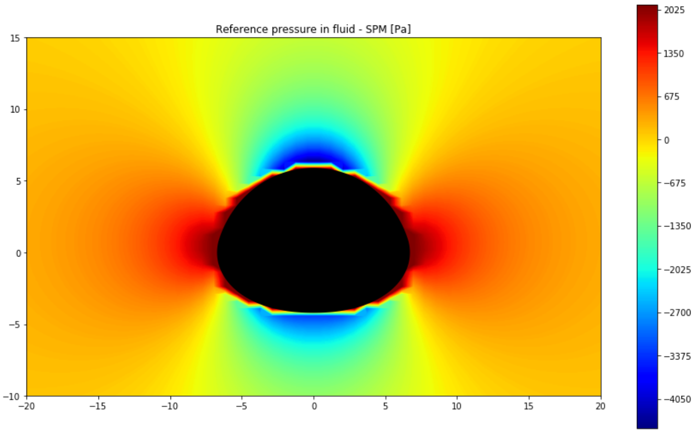


Figure 6.5: The pressure profile for the single cross-section

The Cross-Sectional area is required in order to define the minimum BWR, as explained in section 5.2. In order to do so, the maximum wave load is determined according to paragraph 4.6.2. The cross-sectional area

$A = 105.47\ m^2$. This yields a wave load equal to:

$$F_z = \rho C_I A \frac{du_z}{dt} + \frac{1}{2}\rho D C_{D,z} U_z \sqrt{U_x^2 + U_z^2} \tag{6.1}$$

$$F_z = 1050 \cdot 2 \cdot 105.5 \cdot 0.35 + \frac{1}{2} \cdot 1050 \cdot 11.5 \cdot 1.0 \cdot 0.45\sqrt{0.45^2 + 0.45^2} = 78\ kN/m$$

For an overview of all expressions above, see equation 4.10.
This wave force can be converted to the minimum BWR, which also determines the structural stress in longitudinal direction.

$$BWR_{min} = \frac{B}{W} = \frac{A \cdot \rho_w}{A \cdot \rho_w - F_{z,max}} = \frac{105.5 \cdot 10}{105.5 \cdot 10 - 78} = 1.08 \tag{6.2}$$

Concluding, a cross-section with $\%HWP = 20$ and $GF = 30$ is the optimal cross-section with a $BWR_{min} = 1.08$. The lift coefficient is 0.33 and the thickness of the concrete is 53 cm.

6.4. Composite Cross-Section

This solution is easier to generate, since both targets can be fully met separately. For a depth of 20 meter, the optimal cross-section of the inner tube can be determined according the method presented in Chapter 3. For a depth of 20 meter, the cross-section of the inner tube practically is a circle. The exoskeleton however, can be manually adjusted to create a lift coefficient of at least $C_L = 0.3$. Together with the maximum flow velocity, the fluid density and the height of the cross-section of the inner tube, this can be converted to a lift force:

$$F_{Lift} = \frac{1}{2} \cdot \rho \cdot C_L \cdot D \cdot V_\infty^2 = \frac{1}{2} \cdot 1050 \cdot 0.3 \cdot 11.7 \cdot 1.95^2 = 7\ kN/m \tag{6.3}$$

Where:

- D = Diameter of tunnel = 11.7 m
- $V_\infty = 1.95\ m/s$ according to equations 2.6 and 2.7 at the top of the SFT.

After tweaking the parameters for the exoskeleton, the lift force can be calculated using xFoil as explained in paragraph 5.3.1. The lift force is desired not to be unnecessarily large. However, it obviously must be at least the 7 kN/m as just calculated. Besides, the exoskeleton reduces the drag force on the SFT. For now, the following cross-section is found and assumed to be sufficient regarding the requirements. If one is interested, further tweaking can be executed from the model in Grasshopper.

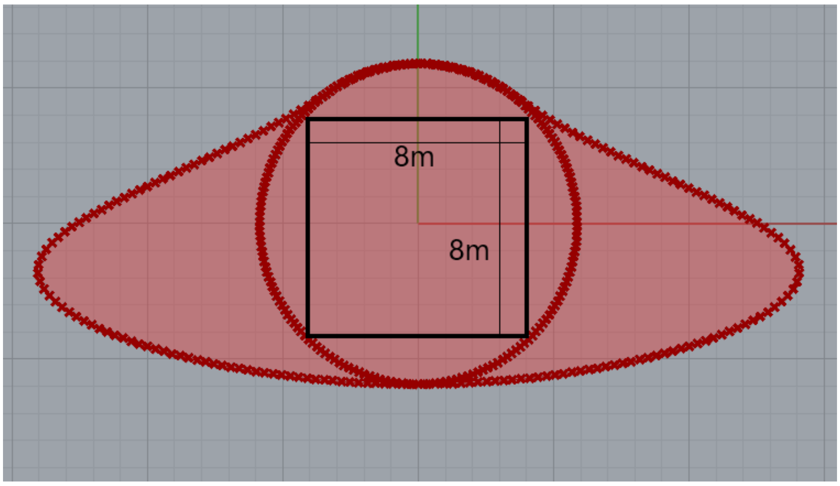
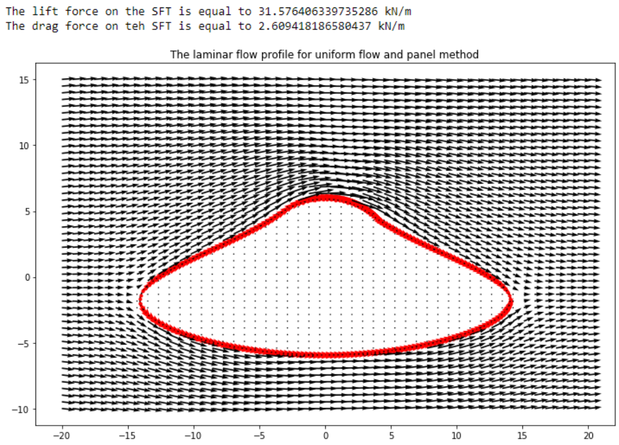


Figure 6.6: The composite cross-section



```
The lift force on the SFT is equal to 31.576406339735286 kN/m
The drag force on teh SFT is equal to 2.609418186580437 kN/m
```

Figure 6.7: The composite cross-section and its flow profile

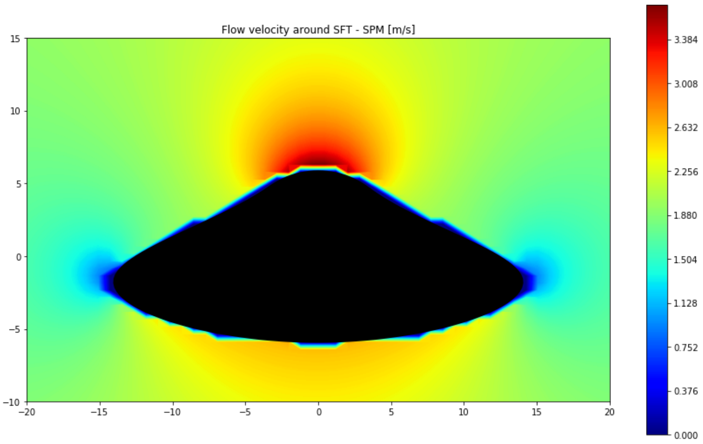


Figure 6.8: The velocity profile for the composite cross-section

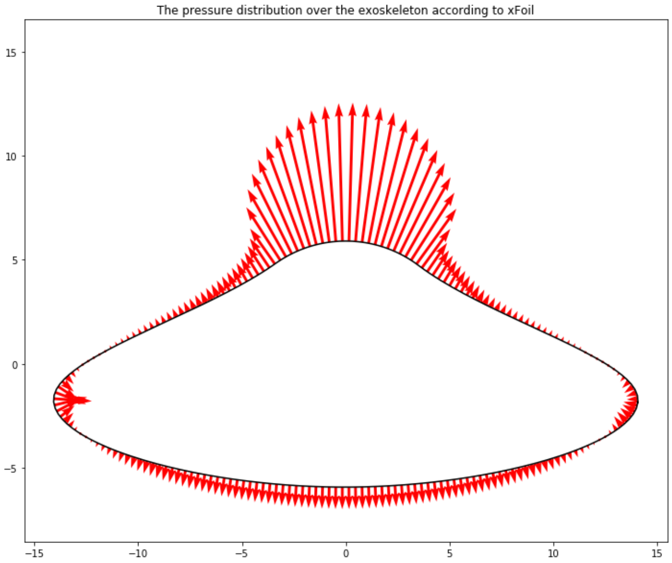


Figure 6.9: The pressure profile for the composite cross-section

With:

- $F_{Drag} = 2.6\ kN/m$, $C_D = 0.11$
- $F_{Lift} = 31.6\ kN/m$, $C_L = 1.35$
- Thickness concrete lining = 8 cm

Similar as for the single cross-section, the minimum BWR must be determined. This is expected to be larger due to the larger cross-sectional area. This area includes the exoskeleton, which also experiences the inertia force due to the waves from the Morison equation. The area of the total cross-section is $A_{total} = 214\ m^2$. Note that not all of this area experiences buoyancy. This is only the part of the inner tube. This area is required for the BWR-calculation and equal to $A_{inner} = 109\ m^2$:

$$F_z = 1050 \cdot 2 \cdot 214 \cdot 0.35 + \frac{1}{2} \cdot 1050 \cdot 28 \cdot 1.0 \cdot 0.45\sqrt{0.45^2 + 0.45^2} = 160\ kN/m$$

$$BWR_{min} = \frac{B}{W} = \frac{A \cdot \rho_w}{A \cdot \rho_w - F_{z,max}} = \frac{109 \cdot 10}{109 \cdot 10 - 160} = 1.17$$

6.5. Conclusion

Concluding, two solutions are presented which both meet the requirements listed at the start of this Chapter. The benefit of the first solution, the single cross-section, is the lower construction costs due to the fact that the cross-section is made of a concrete tube only. Besides, it has a smaller BWR_{min} which is beneficial for the structural stresses in longitudinal direction. Moreover, less tethers are required now which are expensive. The deficit is that it does not optimize for both targets to the fullest. This is visible in the lower lift coefficient and the larger concrete thickness compared to the composite cross-section. The second solution, the composite cross-section, does solve for both targets to the fullest. However, it is a composite structure which might be more expensive, despite the fact that the concrete thickness is lower. And as just mentioned, the BWR_{min} value is larger. To determine which of the two is 'the optimal one', a cost-benefit analysis must be performed. This is not included within this research, but the input for a cost-benefit analysis is presented here.

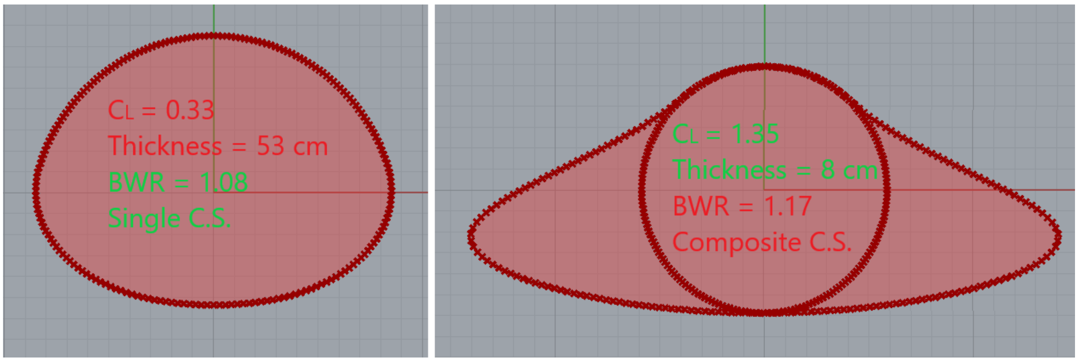


Figure 6.10: The two possible solutions

6.6. Impressions

Some general impressions of the submerged floating tunnel in its environment is shown below. Note that cross-sections below are not identical to the ones presented above.

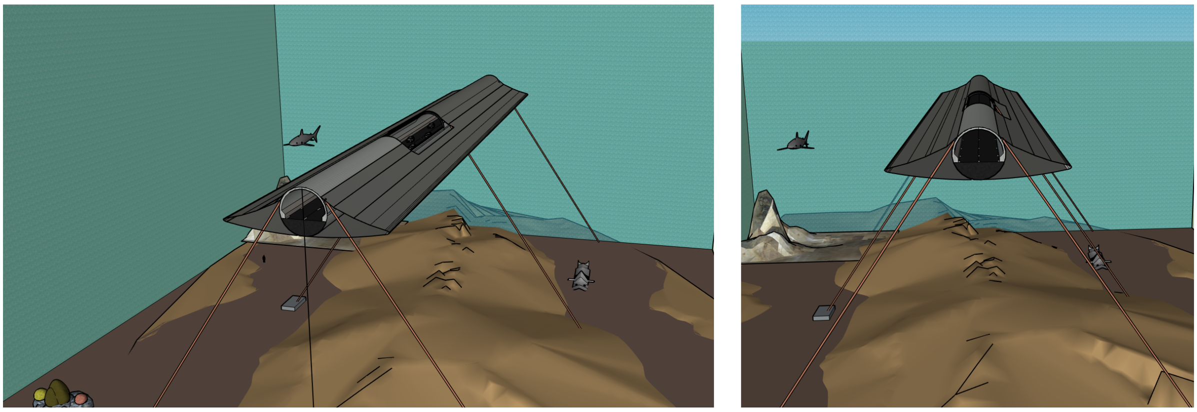


Figure 6.11: The composite cross-section - Impression

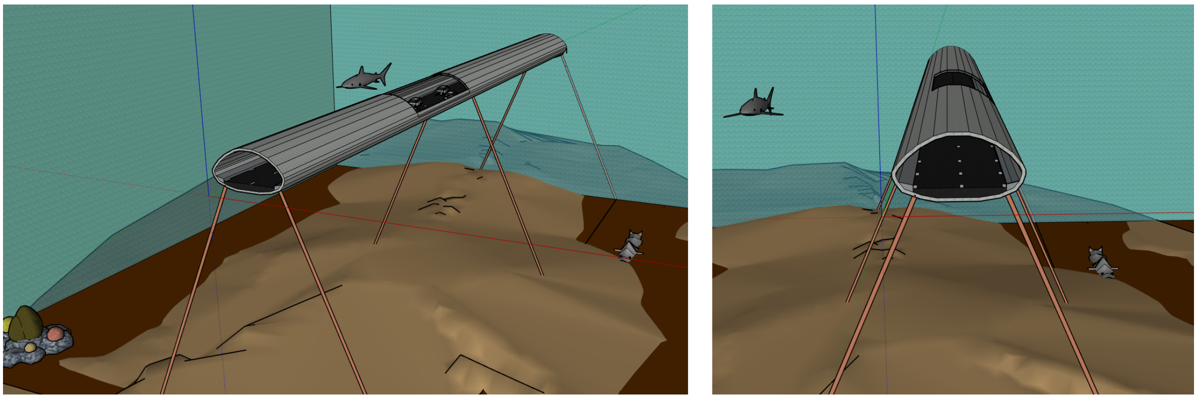


Figure 6.12: The single cross-section - Impression

7 Conclusions and Recommendations

The concept of a Submerged Floating Tunnel (SFT) can be improved by an optimization of the cross-section. First, a general conclusion is drawn given the objective. Then, the research questions as listed in section 1.2 are briefly answered to support this general conclusion. As elaborated in Chapter 5, two solution techniques for 'the optimal cross-section' are presented. In the discussion, the parametric influence on this optimal solution is described. Finally, some recommendations are listed for further research.

7.1. Conclusions

In general, it can be concluded that a parametric model is created in Grasshopper to find the optimal cross-section for a SFT. This optimization process is split into two targets. The first target aims for an optimization of material use, whereas the second target guarantees a tensile force in the tethers that moor the tunnel to the environment.

The optimization of material use yields a shape which appears like an 'egg' or an 'oval' close to the water surface, converging towards a circle in deeper water. This can be explained by the (relative) pressure difference between SFT-top and SFT-bottom. These shapes are obtained by Form-Finding within Grasshopper using the 'Kangaroo' plug-in.

Secondly, the cross-section can generate a lift force in the desired direction from the encountering flow. The lift force contributes to the tensile force in the tethers and a BWR closer to 1.0. This is achieved by creating a differential velocity profile between SFT-top and SFT-bottom. According to Bernoulli's theorem, these differential velocity fields yield a lift force on the total structure. This can be assured by constructing an exoskeleton. When the SFT is anchored to the sea bed, the exoskeleton has a flat bottom with a convexly shaped top. Besides, the edges are sharp in order to reduce the drag force and the turbulent motions. The lift (and drag) force are quantified in Python applying the Panel Method.

Ideally, both targets are fully met by constructing the SFT with a composite cross-section. This consists of an inner (concrete) tube which absorbs the water pressure and a steel exoskeleton which generates lift and reduces drag and turbulence. One can also choose for a compromise between the optimization targets, yielding to a single cross-section. This cross-section also has a more convexly shaped top to generate a lift force, but also has some appearances of a circle to absorb the water pressure efficiently. See section 7.2 for a more elaborate view upon this.

1. **What are the existing types of cross-sections?**
   Often, a regular circle is opted for. Most tunnels are constructed with a circular cross-section because it is well known for its good performance in absorbing (uniform) pressure loads. Another cross-section which shows up during several other researches is a rectangular shape. The rectangle is very common within the design of a regular 'Immersed Tunnel'. It is efficient with regard to its use of internal space. However, both solutions have some disadvantages regarding the interaction between the hydrodynamic loads and the structure.

2. **What loads are acting on the SFT?**
   The main loads considered are the hydrostatic water pressure (HWP), the self weight (SW), the load due to currents and the load due to waves. Consequently, these loads are subdivided into two groups,

where each group was allocated to an optimization target. The two targets with their allocated loads are:

Optimization target	Load group	Consists of:
1.) Optimize material use	Hydrostatic loads	HWP SW + Ballast
2.) Tension in tethers	Hydrodynamic loads	Currents Waves

Table 7.1: The optimization targets and the allocated loads

3. **What is the influence of the shape on F_{Lift} and F_{Drag}?**
First, the drag force is described. As turned out from figure 2.16, a circle is not optimal considering the fact that a low drag force is desired. From experimental data, the drag coefficient $C_D \approx 1.0$ varying slightly for different Reynolds numbers. Based on the same figure, an ellipse proves to be more efficient with a drag coefficient $C_D \approx 0.1$.
The static lift coefficient C_L is 0 (not considering C_L' from turbulence) for cross-sections which are symmetrical over the horizontal. Given the first objective as stated in section 1.2, it is desired to have a lift coefficient which contributes to a tensile force in the tethers. According to Bernoulli's principle, a larger local flow velocity causes a static pressure drop in the fluid. Hence, a differential velocity profile is required between SFT-top and SFT-bottom. This is achieved by moving the 'deflection point' of the cross-section with a Geometry Factor applied on the load profile within Form Finding. This is explained in Chapter 4. A more convexly shaped top results in larger local flow velocities and lower local fluid pressures. The result is a lift coefficient $C_L > 0$ to generate tension in the tethers. The magnitude of C_L can be determined with a relation found between $\frac{V_{top}}{V_{bottom}}$ and C_L. The ratio $\frac{V_{top}}{V_{bottom}}$ is calculated based on the Panel Method, which is coded in Python. Concluding, when one side (top or bottom) has a more convex shape than the other, some lift F_{Lift} can be generated.

4. **How to guarantee a tensile force in the tethers?**
The BWR is responsible for the tensile force in current design thoughts. However, a BWR far from 1.0 has tremendous consequences for the structural stresses in longitudinal direction and the tether system. Hence, it is desired to keep the BWR as close to 1.0 as possible.
The cross-section as described in Research Question 3 can generate a lift force from a horizontal flow. However, this flow is not always present. The governing load combination is when no flow is present, while the design wave is passing. The (lift) force of a wave is most dominantly determined by the inertia component within the Morison equation. The cross-sectional dependence in this inertia component is expressed by the area A and **not by the shape**. Therefore, waves are not included in finding the optimal cross-section and must be fully compensated for by the BWR to guarantee tension at all times. See section 5.2 for a more elaborate view upon this.

5. **How are the loads optimally diverted according to Form-Finding?**
Form Finding applies the external loads on a material with no bending stiffness. The deformed shape is the shape which absorbs these external loads axially. This is most efficient considering material use, because all material fibres are used equally.
When Form-Finding is applied for the water pressure loads, as executed in Chapter 3, an 'egg' or 'oval' turns out to be the optimal shape. The larger curvature at the bottom of the SFT enables the cross-section to absorb the (locally larger) external water pressure more efficiently via its normal force N. It is slightly more efficient compared to a circle with a maximum normal force reduction of 3.7% close to the water surface. Concluding, a circle is not 'the optimal shape' considering material use, but a very good approximation of 'the optimal one'. Especially at larger depths, where the optimal shape converges towards a circle.

6. **What is a compromise between the optimization targets and can they be met both?**
Two solution techniques are presented for 'the optimal cross-section'. The first one is a compromise between the two optimization targets. This shape is retrieved by Form Finding with a fraction of the water pressure load and a Geometry Factor. These represent the optimization for material use and the

generation of F_{Lift} respectively.
The second solution is to construct a composite cross-section made of a (concrete) inner tube with a steel exoskeleton. The concrete inner tube has a cross-section based on the output from Chapter 3. The exoskeleton can be manually created in Grasshopper. The magnitude of the lift and drag force are evaluated by xFoil, which runs from the same python code as mentioned earlier. The benefit is that both optimization targets are fully met.

7.2. Discussion

Here, some simplifications and assumptions are evaluated with their influence on the model output. Moreover, some expectations are given on the validity and applicability of the different types of cross-section per situation, depending on the 'Depth of SFT'.

First of all, the cross-section is optimized in terms of material use as shown in Chapter 3. The loads acting on the SFT are included in the FF-process to generate the shape which diverts all external loads axially. The loads considered in this Chapter are the water pressure and the self-weight, for a BWR equal to 1.0.
The water pressure is correctly applied on the geometry. However, the self-weight is here assumed to be equally divided over the perimeter of the SFT. In reality, this is probably not the case since some ballast is applied at the bottom of the SFT. This yields to a different load profile within the FF-process and a different cross-section as a result. However, at larger depths the water pressure increases. This demands for a thicker concrete tube, making it more plausible to equally divide the self-weight over the perimeter. Hence, it is expected that this effect becomes negligible for larger depths. This is the case for a SFT.
Secondly, the lift coefficients C_L and F_{Lift} as found by Chapter 4 are for both types of cross-section based on approximations for the turbulent effects. Although it is expected that the order of magnitude is right for both methods, the exact value might deviate in reality from these approximations. However, by Bernoulli's principle it is simply impossible that the sign of the lift coefficient is estimated wrong. And since the governing load combination does not include the flow load, these small deviations do not result in large design errors.

Finally, the objective of this research was to make a parametric model. Different (load)-situations results in different cross-sections. Besides, two types of cross-sections are presented within this research, where the optimal one must be chosen based on an explicit cost-benefit analysis.
However, some logical thinking already gives some more insight in this part. The deficit of the single cross-section is the fact that it must relinquish some to both optimization targets, where the optimization of material use suffers the most. This target is most important at larger depths, due to the larger water pressure. Concluding, the deeper the SFT, the more beneficial the composite cross-section becomes.
Moreover, the deficit of the composite cross-section is the larger cross-sectional area A. This area determines the wave (lift) force and therefore the value for BWR_{min}. This wave (lift) force decreases rapidly for increasing depths until a depth of approximately $35m-40m$ (figure 4.23). This deficit in terms of the BWR vanishes out for increasing depths, which is another reason to choose for the composite cross-section at larger depths.
For an overview of what the cross-section approximately looks like depending on the depth, see figure 7.1. Close to the surface, the single cross-section is beneficial with a low value for the fraction of water pressure included. This fraction of water pressure increases by depth. From about 35 meter, the composite cross-section is expected to be optimal, where the area A can increase by depth to reduce drag and

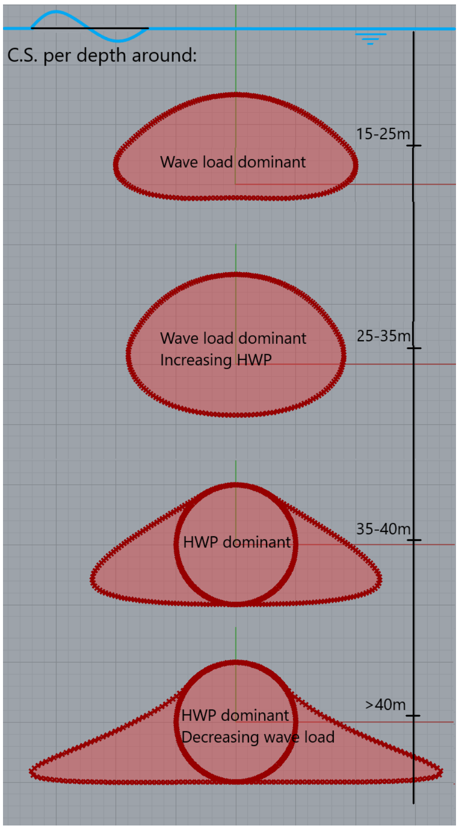


Figure 7.1: cross-section depending on depth

increase lift, while the consequences on BWR_{min} remain
small.
Either way, the two solutions presented are both more favourable regarding a smaller drag force F_{Drag} and a larger lift force F_{Lift} (in the right direction) compared to a regular circle.

7.3. Recommendations

The following recommendations are made:

1. The effect of turbulence on the SFT must be investigated further with its consequences for the lift force. Especially a relation for C_L' is desired to know how to guarantee the tensile force in the tethers. When the turbulent lift force dominates over the mean lift force, it can negatively contribute to the tension force in the tethers. A different (more conservative) BWR_{min} is required then. Therefore, the C_L'-value is important to know. The Panel Method only deals with laminar flow.

2. Some more verification points are required to confirm the found relation between $\frac{V_{top}}{V_{bottom}}$ and C_L. Only one CFD-run was available to verify the found relation.

3. The same holds for the xFoil calculations. This is verified on only one CFD-run, where more are required to confirm the correspondence. Since xFoil approximates the boundary layer equations for sharp edges, it is also interesting to know where the limit is. How blunt can the edges of the cross-section be for xFoil to still approximate it correctly with the e^n-relation?

4. The wave load must be analysed further. Here, the lift force due to waves is calculated based on the Morison equation and fully absorbed by the pretension in the tethers. However, the Morison equation might not be applicable when the SFT becomes too wide. Some interactions can occur if the SFT responds to the wavelength of the passing design wave.

5. An explicit cost-benefit analysis must be performed to choose whether the single cross-section is optimal or the composite cross-section.

6. All calculations as shown above must be executed with safety factors or another probabilistic analysis to come up with the actual design.

7. The rotational moments of the SFT are not accounted for in this research. A rotational moment might generate a compressive force in the tethers as well. However, the lift coefficient C_L is for most cross-sections found significantly larger than C_D. Hence, one might expect little influence of the rotational moment on the tether forces. Nevertheless, it must be investigated further. Moreover, rotational motions might interfere with the flow oscillations.

8. The accidental loads like tsunamis, earthquakes and ship collisions must be included in the actual design.

9. The constructability of the tunnel compartments must be taken into account for an actual design. All different kinds of curvatures might yield complications in the production process of the concrete tunnel compartments. It might be useful to set certain ranges of constant curvature.

10. The panel method as applied here turned out to be useful to analyze the flow behaviour around a submerged body. It might be useful in other disciplines as well, like for the wind flow around high-rise buildings. It is recommended to further explore the applications of the panel method in other disciplines.

Bibliography

[1] Abbott, I. H. and Von Doenhoff, A. E. *Theory of wing sections: including a summary of airfoil data.* Courier Corporation, 2012.

[2] Anderson, J. D. Fundamentals of aerodynamics, 2001. *Google Scholar*, pages 344–346, 2015.

[3] Anderson, J. D. and Anderson Jr, J. D. *A history of aerodynamics: and its impact on flying machines*, volume 8. Cambridge University Press, 1998.

[4] Battjes, J. Vloeistofmechanica. *Collegedictaat CT2100*, 2002.

[5] Bauer, A. M., Längst, P., La Magna, R., Lienhard, J., Piker, D., Quinn, G., Gengnagel, C., and Bletzinger, K.-U. Exploring software approaches for the design and simulation of bending active systems. In *Proceedings of IASS Annual Symposia*, volume 2018, pages 1–8. International Association for Shell and Spatial Structures (IASS), 2018.

[6] Brancaleoni, F., Castellani, A., and D'asdia, P. The response of submerged tunnels to their environment. *Engineering Structures*, 11(1):47–56, 1989.

[7] Connor, N. Laminar flow vs. turbulent flow, May 2019. URL https://www.reactor-physics.com/engineering/fluid-dynamics/laminar-flow-vs-turbulent-flow/.

[8] Dehling, H. and van Maanen, J. Kleine groninger en grote wetenschapper. *NIEUW ARCHIEF VOOR WISKUNDE*, 4:254–257, 2003.

[9] Dong, R. Effective mass and damping of submerged structures. Technical report, California Univ., Livermore (USA). Lawrence Livermore Lab., 1978.

[10] Drela, M. and Youngren, H. Xfoil 6.9 user primer, Nov 2001. URL http://web.mit.edu/aeroutil_v1.0/xfoil_doc.txt.

[11] Drost, L. The submerged floating tunnel: An experimental study on the hydrodynamics and kinematics of a submerged rectangular cylinder in a wave-current environment. *Engineering Structures*, 2019.

[12] Ecotoerisme, E. Geirangerfjord noorwegen, 2012. URL https://www.ecotoerisme.eu/noorwegen/geirangerfjord/.

[13] Elger, D., Williams, B., Crowe, C., and Roberson, J. Engineering fluid mechanics. *Energy*, 2:03–01, 2013.

[14] Engineer, J. t. Panel methods, 2020. URL https://www.joshtheengineer.com/panel-methods/.

[15] Erickson, L. L. Panel methods: An introduction. 1990.

[16] feed me, S. Advanced fluid dynamics - flow around a cylinder, 2014. URL https://spoonfeedme.com.au/course/295/super/flud102/advanced-fluid-mechanics/video/26608.

[17] Groh, R. Boundary layer separation and pressure drag, Jul 2018. URL https://aerospaceengineeringblog.com/boundary-layer-separation-and-pressure-drag/.

[18] Hemel, M.-J. Submerged floating tunnel: the dynamic response due to fluid structure interaction. *Engineering Structures*, 2019.

[19] Holthuijsen, L. H. *Waves in oceanic and coastal waters.* Cambridge university press, 2010.

[20] Huerta, S. Structural design in the work of gaudi. *Architectural science review*, 49(4):324–339, 2006.

[21] Johnson, R. W. *Handbook of fluid dynamics.* Crc Press, 2016.

[22] Keulegan, G. H. and Carpenter, L. H. Forces on cylinders and plates in an oscillating fluid. *Journal of research of the National Bureau of Standards*, 60(5):423–440, 1958.

[23] Kroon, M. Hoe vliegt een vliegtuig, Apr 2017. URL `https://blog.klm.com/nl/hoe-vliegt-een-vliegtuig/`.

[24] Kuethe, A. M. and Chow, C.-Y. *Fondations of aerodynamics*. John Wiley & Sons, 1976.

[25] Le Mehaute, B. *An introduction to hydrodynamics and water waves*. Springer Science & Business Media, 2013.

[26] Lewis, W. J. *Tension structures: form and behaviour*. Thomas Telford, 2003.

[27] Lim, S., Popov, D., Raza, N., Pervaiz, F., and Al-Qadi, M. Feasibility study of submerged floating crossing. *Engineering Structures*, 2018.

[28] Lynch, P. Submerged floating tunnels may be the solution to crossing norway's treacherous fjords, Jul 2016. URL `https://www.archdaily.com/792121/submerged-floating-tunnels-may-be-the-solution-to-crossing-norways-treacherous-fjords`.

[29] Martire, G. *The development of Submerged Floating Tunnels as an innovative solution for waterway crossings*. PhD thesis, PhD Thesis in Construction Engineering, University of Naples "Federico II ..., 2010.

[30] Molenaar, W. and Voorendt, M. Manual hydraulic structures. *Collegedictaat CIE3330*, 2016.

[31] Ozkol, U. Running xfoil from python, Feb 2019. URL `http://www.joshtheengineer.com/2019/02/06/running-xfoil-from-python/`.

[32] Placzek, A., Sigrist, J.-F., and Hamdouni, A. Numerical simulation of an oscillating cylinder in a cross-flow at low reynolds number: Forced and free oscillations. *Computers & Fluids*, 38(1):80–100, 2009.

[33] Reinders, K. *Bored and immersed tunnels*. TU Delft, 2017.

[34] Sarpkaya, T. Force on a circular cylinder in viscous oscillatory flow at low keulegan—carpenter numbers. *Journal of Fluid Mechanics*, 165:61–71, 1986.

[35] seminarppts Follow. Submerged floating tunnel, Apr 2018. URL `https://www.slideshare.net/seminarppts/submerged-floating-tunnel-93232692`.

[36] Soons, F. and Raaij, B. v. *Quick Reference*. TU Delft, 2014 edition, 2014.

[37] Staten, V. The e39 coastal highway route, Dec 2019. URL `https://www.vegvesen.no/en/roads/Roadsandbridges/Roadprojects/e39coastalhighwayroute`.

[38] Sumer, B. M. et al. *Hydrodynamics around cylindrical strucures*, volume 26. World scientific, 2006.

[39] Veenendaal, D. and Block, P. An overview and comparison of structural form finding methods for general networks. *International Journal of Solids and Structures*, 49(26):3741–3753, 2012.

[40] Waitz, I. A. and Litant, W. T. Aero-astro. *Google Scholar*, 2005.

[41] Wikipedia. K-epsilon turbulence model, Mar 2020. URL `https://en.wikipedia.org/wiki/K-epsilon_turbulence_model`.

[42] Zara, D. L. *Dynamic Fluid-Structure-Vehicle-Interaction Analysis for Submerged Floating Tunnels*. TU Delft, 2019.

[43] Zou, P., Bricker, J., and Uijttewaal, W. *Optimization of Submerged Floating Tunnel Cross Section based on Parametric Bézier Curves and Hybrid Backpropagation - Genetic Algorithm. Journal of Marine Structures. under review*. TU Delft, 2020.

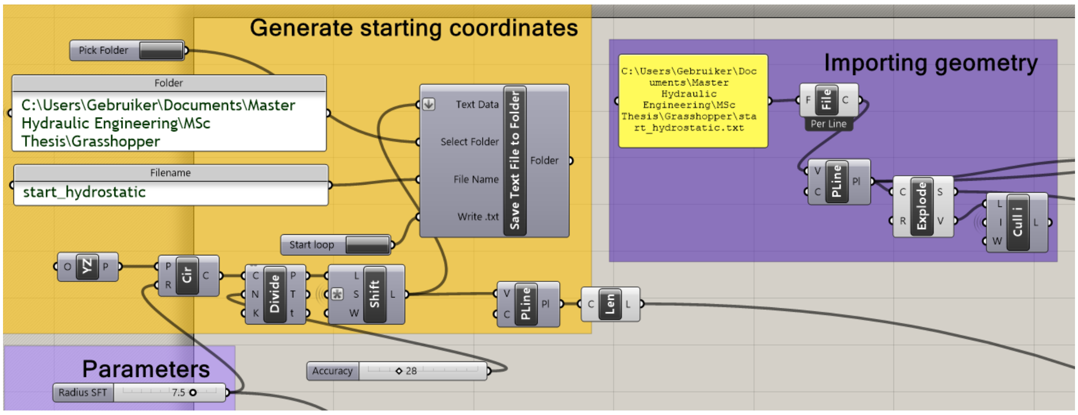
A

Hydrostatic Water pressure in Rhino-Grasshopper

In this Appendix, the model in Grasshopper for the hydrostatic water pressure is further elaborated. This Appendix coincides with the information presented in Chapter 3. It is assumed that the reader has the basic background knowledge of the modules used in the programming language of 'Grasshopper'. The focus here lies on the 'chain of modules' in Grasshopper, which all together form the script. This script yields the output as shown in Chapter 3. Moreover, note that the final model is folded into clusters to maintain overview. This model is shown in Appendix D. The content of these clusters is shown here.
First, some additional explanations are given on the geometry. Then the loads are discussed, which consist of the water pressure and the 'stabilizing force' (self-weight). Consequently, two options for the anchor point are elaborated. Finally, the influence per parameter on the output is considered.

A.1. Geometry

The starting geometry is a circle with a certain radius. This radius is important, since the deformed shape as found by Kangaroo is scaled back to the original perimeter. The generated circle is divided into N nodes, which defines the accuracy of the model. The output of this part is N coordinates, each defining a node on the geometry. In order to be able to loop the model, these coordinates are written to a file. The looping process is explained later on in this Appendix. Notice that the location of the external file can be manually adapted.
After that the starting geometry is written to an external file, it is imported to Grasshopper again. Then, a polyline is drawn through the N nodes. Finally, this polyline is exploded. The result is a geometry consisting of N nodes and N segments. The nodes and segments are identified as shown in figure 3.1. For an overview of the script, see figure A.1.



Folder
C:\Users\Gebruiker\Documents\Master Hydraulic Engineering\MSc Thesis\Grasshopper

Filename
start_hydrostatic

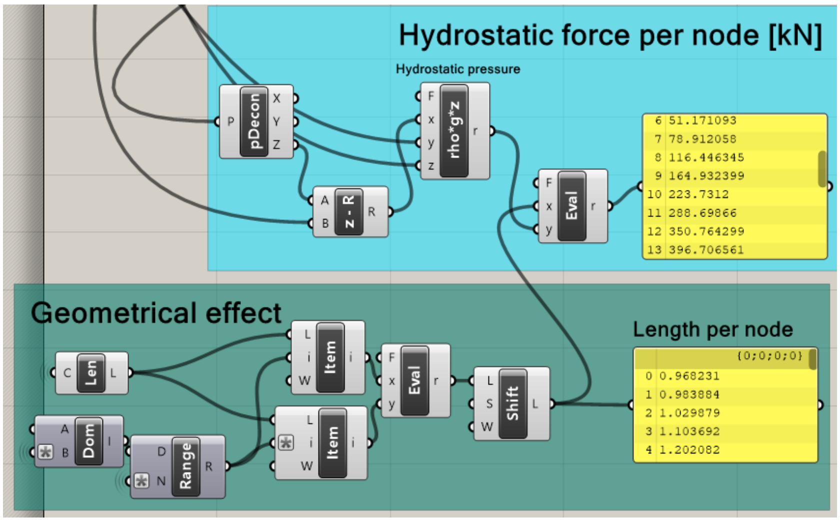
C:\Users\Gebruiker\Documents\Master Hydraulic Engineering\MSc Thesis\Grasshopper\start_hydrostatic.txt

Figure A.1: The script in GH for the generation of the geometry

Now, the starting point of the model is generated and the loads can be defined.

A.2. Loads - Hydrostatic water pressure

Here, the loads are calculated. The hydrostatic water pressure linearly depends on the local depth z, as was shown by equation 2.1. Therefore, the coordinates of the nodes are decomposed into a x,y and z-component. Logically, the z-component is here of interest. Before calculating the pressure at each node, the radius R is subtracted from the local height z. This must be done because the circle is generated around the origin of the coordinate system. Hence, the top of the SFT now has local height $+R$, whereas it is desired to be 0. Moreover, the depth on top of the SFT must be added to the local depth z. The output is now the correct local depth per node. The final step in determining the water pressure per node is simply working out equation 2.1.
The next step is to calculate the 'effective length per node'. The effective length represents the area which diverts its pressure to the closest node. This area is the half-length of the two adjacent segments, multiplied by the imaginary depth of $1m$. This is done in Grasshopper by constructing a domain from 0 to 'N-1'. This domain is then converted to a range (list) with 'N-1' steps. The output of this range module is a list which gradually increments from 0 to 'N-1' with steps equal to 1. This list is required to repeat a certain process. Consequently, the length of the segments are selected with a 'Len'-module. Then, item i and item $i+1$ from this length list are selected and averaged. By inserting the 'range module' as index for the 'list item' module, all N nodes get an averaged length assigned. The final step is to shift this list with -1, to have the values to the corresponding coordinates.
Finally, this effective area is multiplied with the pressure to achieve a hydrostatic force per node. The list of forces is also shown in figure A.2



```
 6 51.171093
 7 78.912058
 8 116.446345
 9 164.932399
10 223.7312
11 288.69866
12 350.764299
13 396.706561
```

```
         {0;0;0;0}
0 0.968231
1 0.983884
2 1.029879
3 1.103692
4 1.202082
```

Figure A.2: The script in GH for the generation of the hydrostatic loads

A.3. Plane definition

As mentioned before, the hydrostatic pressure is unidirectional. Therefore, pressure component is equal to the **magnitude** of the load itself. In order to locate the water pressure correctly, a new plane (u,v,w) must be defined. For the starting geometry, this is rather easy. A line is drawn between the circle center and the node (base line). Perpendicular to this base line, a new local coordinate system (u,v,w) is generated with its origin at the node. A similar approach is applied for a geometry which is not a circle. Note that this is the case later in the looping process. This underlying concept has already been explained in paragraph 3.3.3 and in figure 3.6. In figure A.3 the same process is shown, but then in GH's script. A similar range is created in order to repeat a certain step multiple times, but now from 1 to N with steps equal to 1. For each step, node 'N-1', node 'N' and node 'N+1' are selected using the 'list item' module. For these three nodes, a new circle is generated. The same base line is then drawn between the newly defined circle center and node 'N'. Perpendicular to this base line, a new coordinate system is introduced. The w-component represents the magnitude of the water pressure load. Hence, a new point is defined within the plane just created. The u and v-component are 0 and the w-component has got the magnitude of the load as was calculated in section A.2

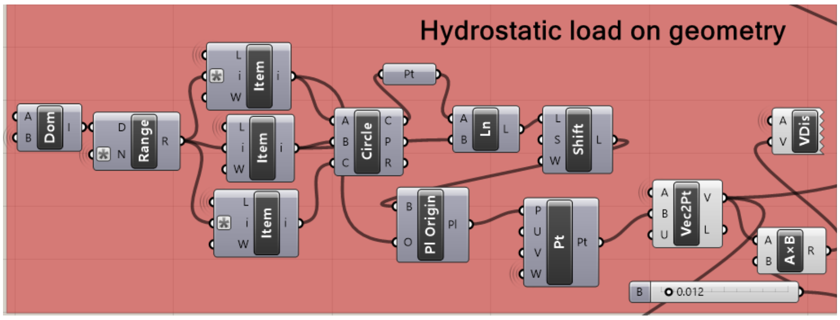


Figure A.3: The script in GH for the generation of the new (u,v,w)-coordinate systems

Finally, a vector between two points is created. These two points are the origin of the new coordinate system and the point 'w' as described above. The hydrostatic loads are now located correctly.

A.4. Anchor point

The anchor point is very important in the model. The model functions well as long as the loads on the geometry are uniform in each direction. As explained in section 2.5, the plug-in Kangaroo functions as a mass-spring system. Each load results in an acceleration, velocity and displacement per node according to the spring stiffness 'S'. This means that no relative movements are present, given that the sum of all the forces is equal to zero for each direction. Concluding, Kangaroo is holding the geometry at its original position. Once the loads are not in equilibrium for each direction, the geometry starts moving as the acceleration is not zero (Newton's 2nd law). If there is no anchor point, the geometry keeps 'falling' in the direction of the net force. Therefore, an anchor point is required, since the forces acting on the SFT are not uniform.
Below, two options are considered as anchor point. In the end, after the pros and cons are weighed against each other, the final solution is chosen.

A.4.1. Centre point as anchor

The centre point can function as anchor point, but only once all the segments of the SFT are connected to the anchor point. This can be done by connecting all the nodes to the centre point via a line element. Obviously, these elements also have to be inserted into Kangaroo. Otherwise, the original segments are not connected to the anchor point and they will start moving anyway. This option works well. However, it has got one deficit. As mentioned before, the elements inserted in Kangaroo each have stiffness 'S'. Hence, the additional line elements in the system also exert a certain resistance on the deformation of the geometry. The additional force on the nodes, directed to the centre, is equal to:

$$F_{centre} = S \cdot (l_{new} - l_{old}) = S \cdot \Delta l \tag{A.1}$$

This is because the rest length of a line element is by default equal to the original length of the line element. The force is then dependent on the displacement of the node. This effect becomes stronger for the 'deeper' nodes of the SFT. The deeper the node, the higher the loads due to hydrostatic pressures. Consequently, a higher load results in a larger deformation. And a larger deformations means a larger additional force according to equation A.1. See figure A.4 for an overview of the loads. In red, the deformed shape is sketched.

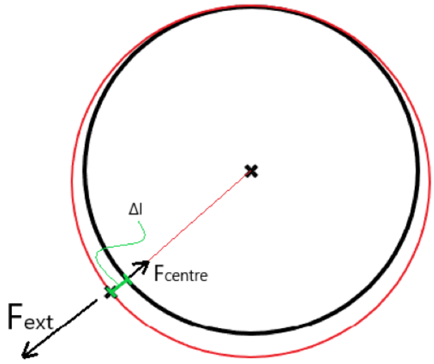


Figure A.4: The additional load towards the centre

Once the shape is deformed, the additional line elements are deformed as well. This means that the spring force in these line elements starts acting. The influence of these additional spring forces is noticeable as from the first loop. Kangaroo applies a default value of l_{rest} equal to the original length of each line element, as just mentioned. This means that the force in the additional line elements is linearly dependent on its own length variation Δl. Concluding, the effect of the additional line elements is directly dependent on the output of the process. The external force should be increased by:

$$S \cdot (l_{new} - l_{rest})$$

This implies that, on top of the required loops for geometrical effects and a change of curvature, an additional loop is required to account for the additional spring forces within this anchoring method. The output of one loop 'i', which is a displacement Δl, is required in a second loop 'j'. This means that the process of Form-Finding becomes more difficult, both physically as computationally. However, it is possible to model the SFT with this anchoring method. To clarify the required looping processes, see figure A.5. Both loop 'i' and 'j' stop when the process is converged and no relative movements are present anymore.

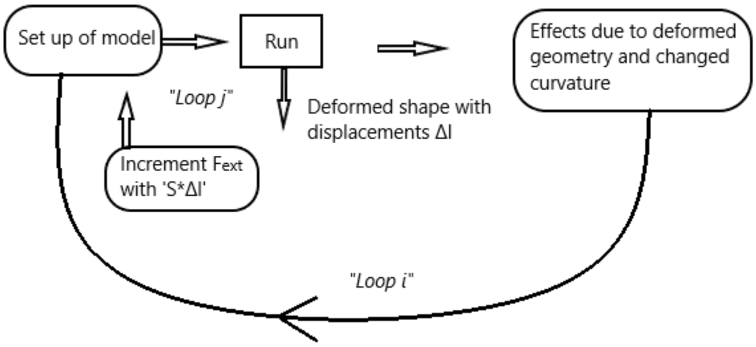

Set up of model
Run
Effects due to deformed geometry and changed curvature
"Loop j"
Deformed shape with displacements Δl
Increment Fext with 'S*Δl'
"Loop i"

Figure A.5: The double looping process

A.4.2. Top point as anchor

The top point can be used as anchor point as well. Or actually, any other random point at the circles perimeter. For the elaboration of this method, the top point is used throughout this explanation. But note that any other node is possible as well.

The use of the top point is beneficial, since the effect of the additional line elements can be prevented. (Indirectly), all segments are connected to the anchor point. However, a malformed shape is then the result, because the anchor point affects the deformations with its support reaction. This can be solved by compensating for the difference in forces for each direction, being only the z-direction for the hydrostatic water pressure.

By applying an additional force, which stabilizes the vertical movement, the geometry does not float away.

When this load is equally distributed over all the nodes, it also doesn't affect the form-found shape. This stabilizing force can be compared to the weight of the SFT in case the BWR is 1.0. The support reaction is then equal to zero. This additional load basically is a 'rigid body displacement'. Hence, it only causes equilibrium and not a malformed shape. The result is a FF-shape, which is not affected by the anchor point and does not want to float away. Besides, no additional elements are required which influence the Form-Finding shape. See figure A.6. Left, all the loads are shown. In yellow, the stabilizing vertical forces are highlighted, equally distributed over the nodes. On the right, the form-found shape is shown with the scaled shape in yellow again. Note: the additional loads *do not* influence the shape, but *do* influence the N-line distribution. This effect is discussed later in paragraph 3.6 about the validation of the model.

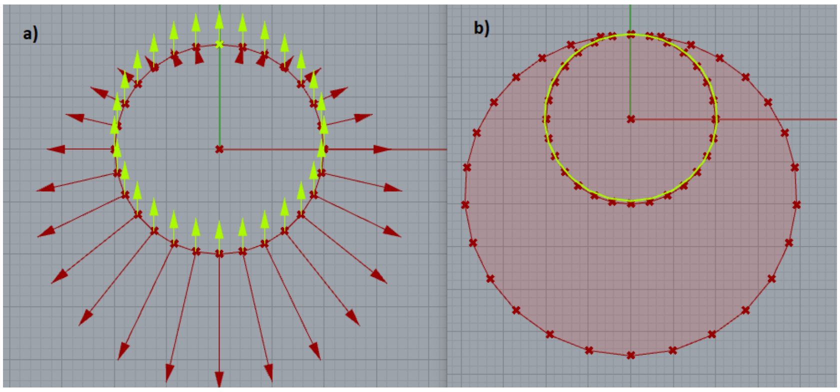


Figure A.6: a) the geometry with loads, vertical stabilization loads in yellow. b) the form-found shape, scaled back in yellow

For now, the second method with the top point as anchor point is applied in the remainder of this research. This is done because the modelling is easier as explained above. However, if one prefers the centre point as anchor point, this is possible given a compensation for the additional spring forces.

The stabilizing force in GH is modelled as shown below. The water pressure loads are decomposed in a x,y and z-direction. Since only the z-direction of the load has a net value not equal to zero, this net value is calculated. Consequently, this net value is divided by N, resulting in an equal distribution of this 'stabilizing force' over all N nodes. Finally, the load is placed on each node with the just found vector.

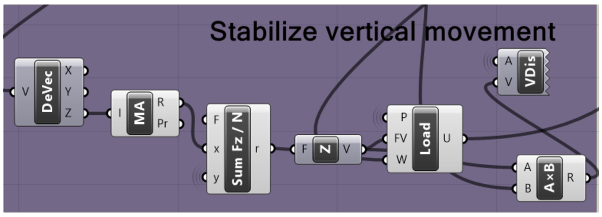


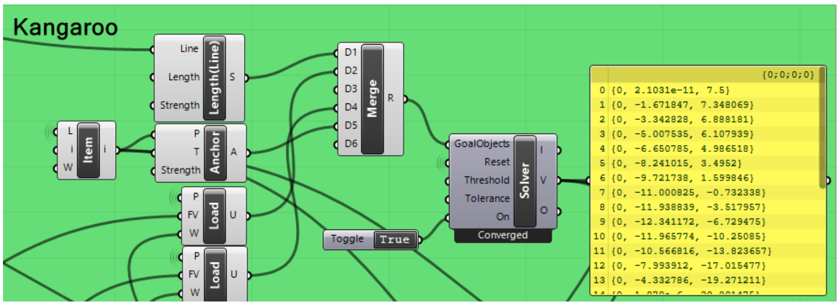
Figure A.7: The stabilizing force in GH's script

A.5. Kangaroo

Before starting the Form Finding process, everything is inserted into the Kangaroo plug in. This means:

- The geometry segments: Via a Length Line component, resulting in elements with no stiffness
- The anchor point: By selecting the right node (top)
- Hydrostatic Water Pressure load
- Stabilizing load

See figure A.8 for an overview in the GH's script.



```
                         {0;0;0;0}
 0 {0, 2.1031e-11, 7.5}
 1 {0, -1.671847, 7.348069}
 2 {0, -3.342828, 6.888181}
 3 {0, -5.007535, 6.107939}
 4 {0, -6.650785, 4.986518}
 5 {0, -8.241015, 3.4952}
 6 {0, -9.721738, 1.599846}
 7 {0, -11.000825, -0.732338}
 8 {0, -11.938839, -3.517957}
 9 {0, -12.341172, -6.729475}
10 {0, -11.965774, -10.25085}
11 {0, -10.566816, -13.823657}
12 {0, -7.993912, -17.015477}
13 {0, -4.332786, -19.271211}
```

Figure A.8: The elements inserted into the Kangaroo plug-in

Finally, the FF-shape from Kangaroo is scaled back to the original perimeter to have a usable and presentable shape.

A.6. Model overview

For an overview of the total model described above, see figure A.9

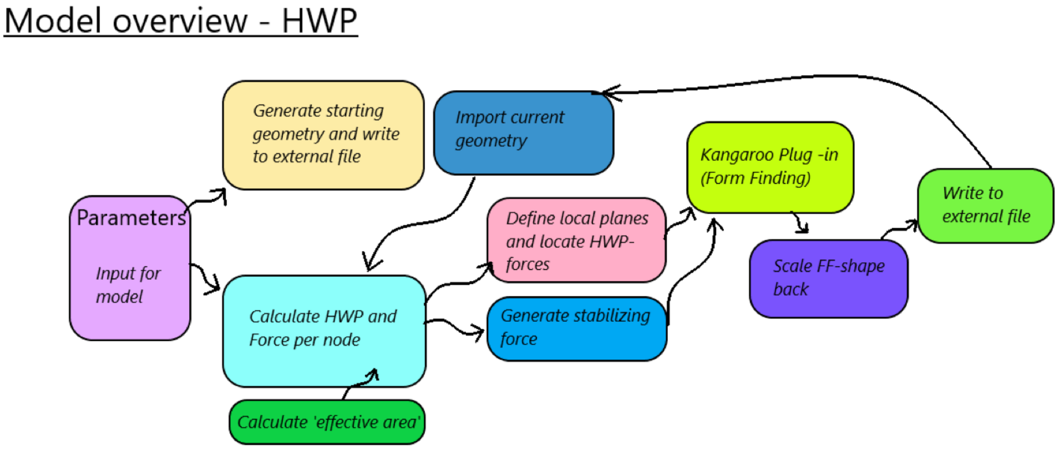

Model overview - HWP
Generate starting geometry and write to external file
Import current geometry
Kangaroo Plug -in (Form Finding)
Parameters
Input for model
Define local planes and locate HWP-forces
Write to external file
Scale FF-shape back
Calculate HWP and Force per node
Generate stabilizing force
Calculate 'effective area'

Figure A.9: A complete overview of the water pressure model in GH.

The loops that are required to take geometrical effects and changes of curvature into account are now automatically executed, since Grasshopper recognizes the renewed condition once the newly found coordinates are written to the same external file. By simply writing the renewed coordinates to the file, the looping process is being executed. Besides, this process is convergent meaning that one single solution is the output of the 'optimal shape'. This shape can be seen in section 3.5.

A.7. Model output

Here, the model output is shortly described. Especially some parameter influence is elaborated. For the full output of the water pressure model, one is referred to the main report in section 3.5. Moreover, a different starting geometry (rectangle) is used here to evaluate whether a circular like shape is indeed optimal according to Form Finding.

The reference situation sketched to compare changing parameters to is shown in figure A.10. The settings of this model are:

- $R = 7.5\ m$
- $\rho = 1050\ kg/m^3$
- $Depth\ top\ SFT = 0\ m$

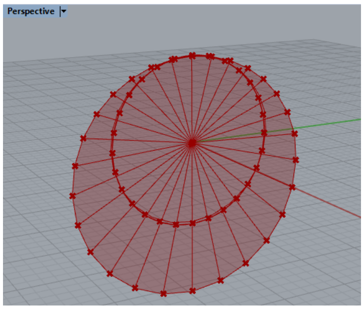


Figure A.10: The cross-section for a SFT just below water surface

The expectations seem to be valid, considering the figure above. The deformation at the top is practically zero, because there is no water pressure at the top of the SFT. Besides, note that the deformed shape is not scaled back yet in figure A.10. Moreover, the relative pressure difference results in a 'water drop' shape.
Other influences of the parameters on the cross-sectional shape are listed below, with the visual confirmations of the influences in figure A.11.

- **Radius:** An increasing radius of the SFT should not influence the shape of the Form-Finding method significantly. For a larger radius, the relative difference in water pressure between top and bottom also increases slightly. Therefore, the 'water drop' shape is exaggerated a little. However, the effects are expected to be negligible. This is visible in figure A.11a. The shape is comparable to the one in the reference situation.
- **Depth of the SFT:** An increasing depth of the SFT causes higher hydrostatic water pressures everywhere. Firstly, this results in deformations at the top of the SFT as well since it is now subject to a hydrostatic load. Besides, the relative water pressure difference decreases. Although the absolute pressure difference remains constant, as it depends on the tunnel diameter, the relative difference decreases. This means that the cross-sectional shape should converge more towards a circle rather than a water drop. Both effects are visible in figure A.11b.
- **Fluid density:** The fluid density results in higher pressures on the SFT. This means that a higher density causes higher deformations. This is visible in Rhino/Grasshopper, although the changes are negligible. Due to the higher loads, the relative difference in hydrostatic pressure increases as well. But again, the effects are negligible.

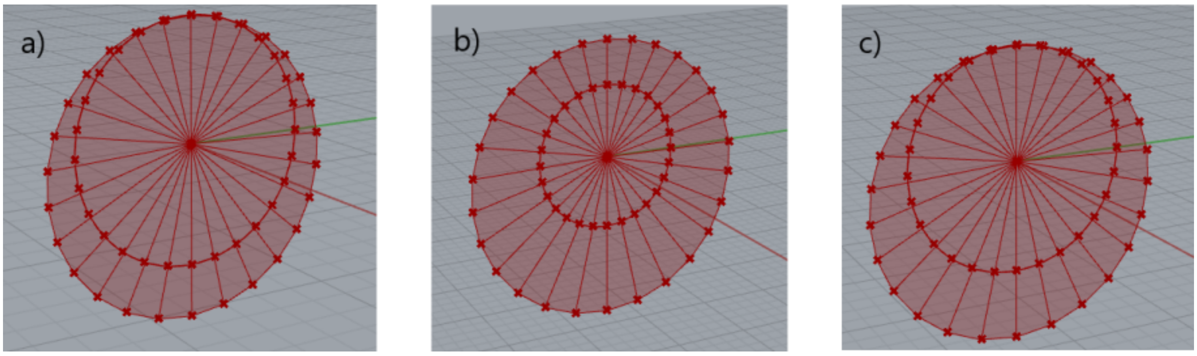


Figure A.11: The CS-shape for a) changing radius, b) higher water depth, c) changing fluid density

These are the results for one single loop, not taking into account the geometrical effects and changes in curvature yet. Now it is established that the first loop yields reasonable results, multiple loops can be generated. The looping process should converge towards one solution. This is because the deformed shape should, according the the Form-Finding method, be the optimal shape in diverting the loads in an axial way. If this is the case, no deformations should be there. It is only the geometrical effects and the change in curvature that causes small deviations. However, they should vanish during the looping process.

The looping process, as explained by figure 3.7, is repeated 3 times for visual confirmation of the converging iterative process. Furthermore, the starting geometry is changed from a circle to a rectangle. Although it is well known that a circle is a well performing shape in absorbing pressures, it is checked once again to see if the process indeed converges to a circular kind of shape. See figure A.12.

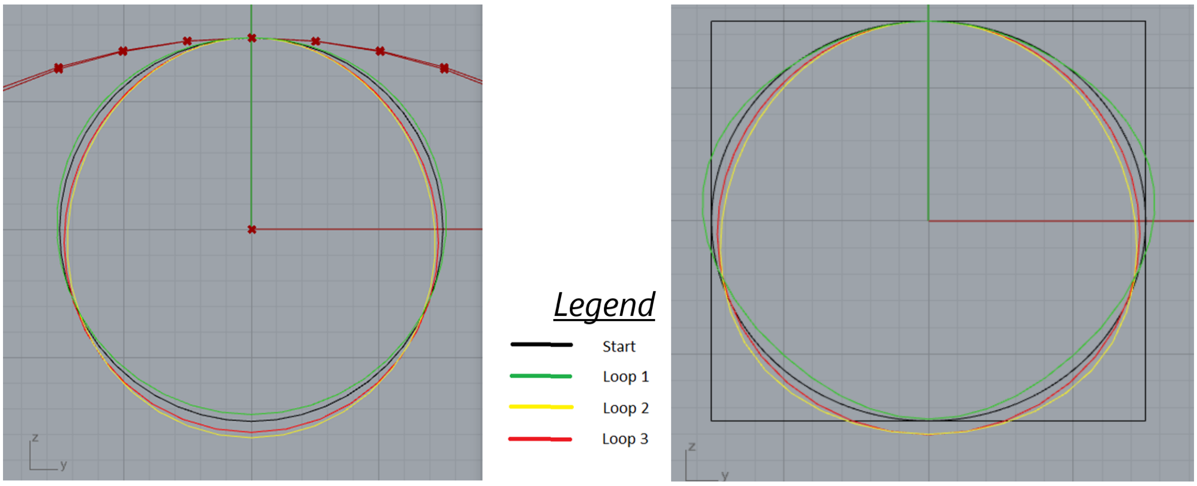


Figure A.12: The looping process with 3 loops for a circle (left) and a rectangle (right) as starting geometry

What can be concluded from the figure above is that the model indeed converges to a circular shape, independent of the starting geometry. A striking thing is that even after 3 loops the results are already almost identical. The geometry converges to a stable solution where the eventual shape looks more or less like a water drop. As explained before, this is in line with the hypotheses. The next step is to validate the model analytically and check whether all external forces are, indeed, absorbed axially. This is done in paragraph 3.6 in the main report.

B

Hydrodynamic currents in Rhino-Grasshopper

In this Appendix, the model set-up is described further for the hydrodynamic loads due to currents in Grasshopper. It is assumed that the reader has basic knowledge of the modules used in Grasshopper. The underlying concepts are explained. Besides, the flow loads are here placed symmetrically. This is done to obtain a symmetrical shape, which is desired since the flow may encounter the SFT from both sides. This is already explained in Chapter 4. In Appendix C, an additional view is given upon asymmetrical flow/shapes.
Similar to Chapter 4 from the main report, this Appendix governs the optimization criterion:

$$Convert\ the\ external\ hydrodynamic\ loads\ into\ tension\ in\ the\ tethers\ of\ the\ SFT.$$

As became clear from section 4.1 and 4.2, the aim is to create differential velocities between the top of the SFT and the bottom of the SFT. This is of influence on the corresponding pressures in the fluid according to Bernoulli's principle. Concluding, differential velocities result in a lift force F_{Lift} being non-zero. How these differential velocities are obtained, is explained in this Appendix.
First, the additional parameters and the geometry are shown. Then, the (manipulated) loads are explained after which some output is shown.

B.1. Input Parameters and Geometry

This part is mainly the same as section A.1. Therefore, for a more detailed view on the creation of the geometry, one is referred to Appendix A.1. In short, a circle is used as starting geometry. It is subdivided into 'N' nodes and 'N' segments. Variable 'N' represents the accuracy of the model. The additional parameters compared to water pressure model are:

- V_{wind} = flow velocity due to wind at surface
- V_{tide} = flow velocity due to tidal wave at surface
- Total depth at location of SFT

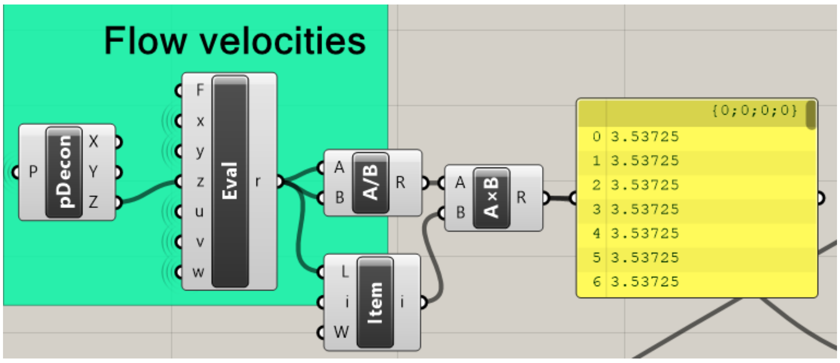
Concluding, barely any differences compared to the water pressure model from Appendix A so far.

B.2. Hydrodynamic loads

Here, the loads due to the hydrodynamic current is calculated and placed on the nodes which are just created. The magnitude of the horizontal flow load is a drag force, as expressed in equation 2.21. The equation is shown below again:

$$F_{x,D} = \frac{1}{2} \cdot \rho \cdot D \cdot V^2 \cdot C_D$$

The pressure component of these force is required within Form Finding. See figure 2.14. Besides, note that the flow velocity V is assumed to be uniform. Therefore, the flow velocity is not exactly calculated based on the profile shown in equations 2.6 and 2.7. However, the flow velocity at the **top of the SFT** is calculated according the the corresponding formulas and assumed to be uniform with that magnitude. This saves a lot of complexity within the construction of both the Panel Method and the CFD. Still, it approximates reality quite well. See figure B.1



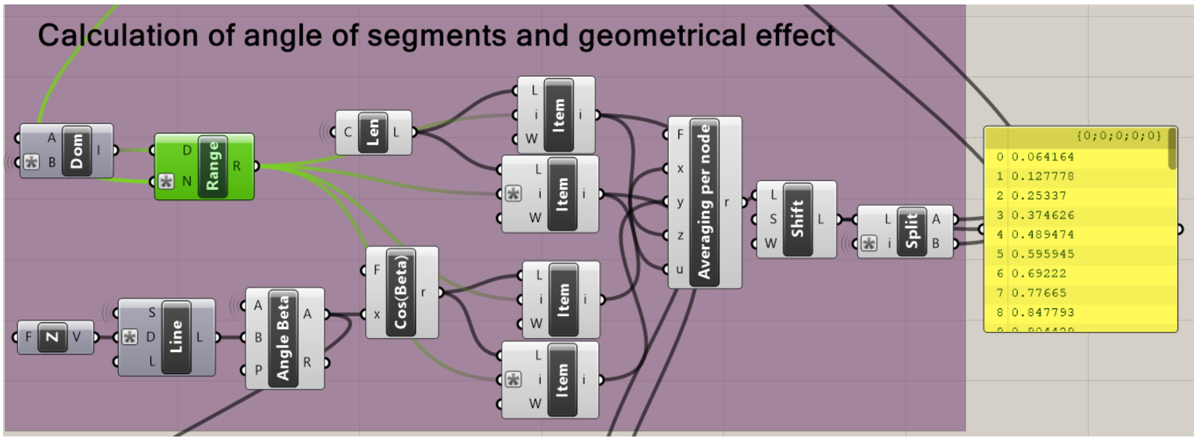
```
{0;0;0;0}
0 3.53725
1 3.53725
2 3.53725
3 3.53725
4 3.53725
5 3.53725
6 3.53725
```

Figure B.1: The uniform flow velocities at each node

Here, the input is for $V_{tide} = 2\ m/s$ and $V_{wind} = 2\ m/s$. At the top of the SFT, which is 20 meter below the water surface here, the local flow velocity is $V_{total} = 3.53\ m/s$. The output from this part is a list of N values with each the uniform flow velocity corresponding to the velocity at the top.

Another parameter from the equation above is the drag coefficient C_D. This value depends on the shape and continuously changes during the FF-process. Within the FF-process, the magnitude of C_D is of barely any influence. A larger C_D-value results in larger loads. Consequently, the larger loads result in larger deformations in the FF-process. However, the deformed shape as found by Kangaroo is scaled back to its original perimeter. Concluding, the relative differences for the C_D-value are practically zero. This value is set to $C_D = 1.0$ according to figure 2.16.

The next step is to determine the pressure component of the drag force, which is required within the FF-process. This can be calculated with equation 2.23. Moreover, the 'effective length' absorbed by each node is determined in the same way as done for the water pressure model. These two steps are combined and visible in figure B.2.



```
{0;0;0;0;0}
0 0.064164
1 0.127778
2 0.25337
3 0.374626
4 0.489474
5 0.595945
6 0.69222
7 0.77665
8 0.847793
```

Figure B.2: The calculation of the pressure component of the 'effective length' per node

On the top left, a domain and range component are visible again. This is to create a list from 0 to 'N-1' with incremental steps equal to 1. This list is required to execute a loop within Grasshopper, as was explained in Appendix A already. On the left bottom, angle β is calculated for each segment. For the definition of angle beta, see figure 2.17. Consequently, the cosine of this angle is taken to determine the pressure component. At the top of the figure above, a 'Len'-module is visible. This module shows the length of each segment.

Then, both the $cos(\beta)$ and the length values are selected for node 'N' and node 'N+1' simultaneously with the 'List Item' modules. This is followed by the 'Averaging per node' module. As the name already implies, this averages the values for the length and the $cos(\beta)$ per node. Finally, this list is shifted with '-1' to have the correct values with the corresponding node again. The output is shown on the right, a list of 'the pressure component of the effective length'. This might sound like a complex term, but it is nothing more than the length of the adjacent half lengths per node, multiplied by $cos(\beta)$ to retrieve the pressure component of the drag force. For a clarification, see figure B.3

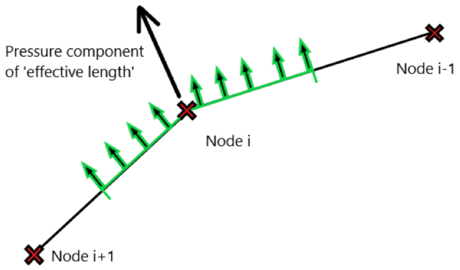


Figure B.3: The visualization of the pressure component of the 'effective length' per node

This 'pressure component of the effective length' can be multiplied with the imaginary length of 1m in the longitudinal direction, to obtain a surface area $[m^2]$. The drag force that must be inserted into the Kangaroo plug-in is then:

$$F_{D\perp} = p \cdot A$$

Where:

- $p = \frac{1}{2} \cdot \rho \cdot V_{total}^2$
- A = pressure component of effective length $\cdot 1\ m$

Furthermore, the flow loads are placed symmetrically as mentioned in the beginning of this Appendix. Physically, this would look like the left half of the SFT being subject to a uniform flow encountering from the left and the right half of the SFT being subject to a uniform flow from the right. This is in reality not possible, but required in order to obtain a symmetrical output. The Form-Finding process is directly related to the load profile, meaning that an asymmetrical load profile can never result in a symmetrical shape. Since the latter is preferred, this unrealistic load profile is applied anyway. The drag forces must be placed perpendicular to the structures' surface again. In order to do so, the same process is repeated as was explained in Appendix A and supported by figures 3.6 and A.3. The output of the FF-process as was already shown in figure 4.3 is shown here again.

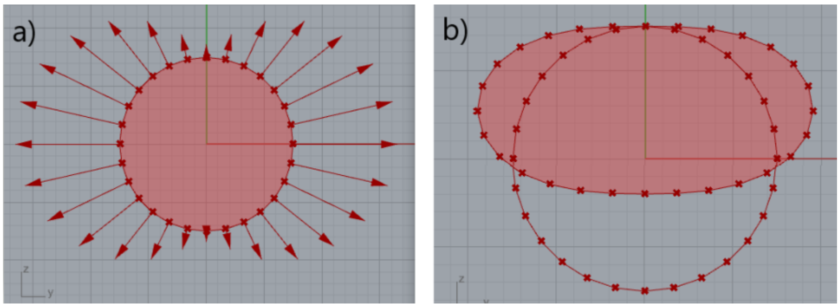


Figure B.4: The load profile and Kangaroo's output for the hydrodynamic current

However, the load profile and the shape are still symmetrical over the horizontal. This means that no differential velocities between top SFT and bottom SFT are obtained yet. This is treated in the next paragraph.

B.2.1. Movement of the deflection point

The model as described so far, is here manipulated in such a way that the deflection point is moved. In case the SFT is anchored to the seabed, a positive lift force is desired. The result is a tension force in the tethers. In order to achieve this, the deflection point must be moved **downward**. Then, the distance via the top increases meaning that the flow velocities increase locally as well. This goes combined with a local pressure decrease and a positive lift force.

The load profile from figure B.4 must be manipulated to move the deflection point downward. The loads from figure B.4a with a **negative** z-component are amplified with a certain **Geometry Factor**. The process in grasshopper is shown below.

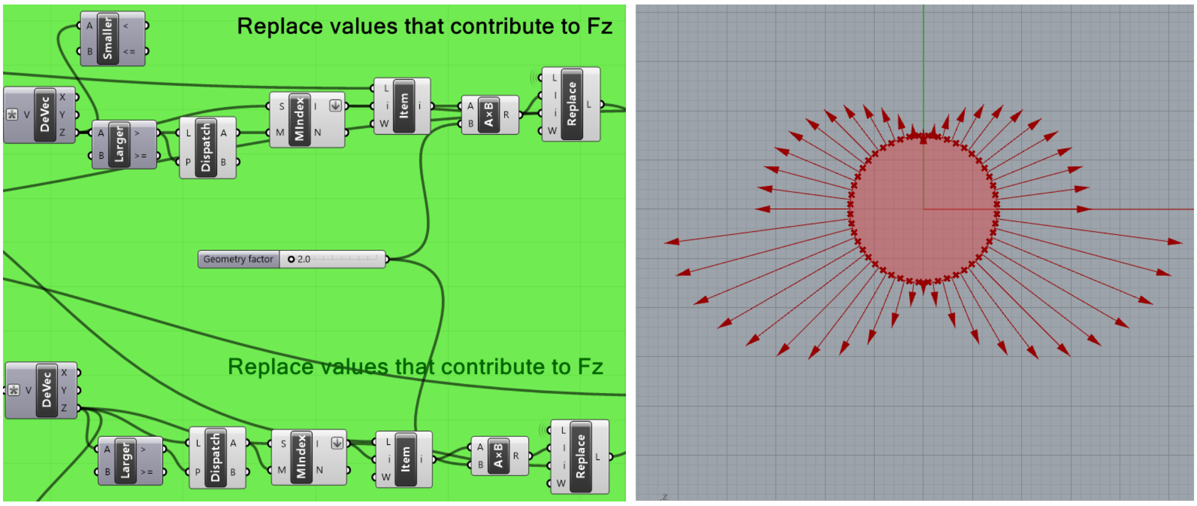


Figure B.5: The amplified load profile and GH's output

In short, the load vectors are selected and decomposed in a x,y- and z-component. When the z-component is smaller than zero, the original load vector is selected with the 'Dispatch' module, the 'MIndex' module and the 'List Item' module. These vectors are consequently being multiplied with a Geometry Factor and replaced in the original list of load vectors. The manipulated load profile is shown on the right in figure B.5.

B.3. Kangaroo and Output

Again, all items are inserted into the Kangaroo plug-in again, which performs the Form Finding process. This is not shown and elaborated further, since this is already done in Appendix A. The item inserted into Kangaroo are:

- Geometry (segments without stiffness)
- Anchor point (top point)
- Manipulated load profile as shown in figure B.5
- Stabilizing forces to compensate for anchor location and get $\Sigma F_{x,y,z} = 0$

Note, loops are not necessarily required here because the objective was to create differential velocities between top and bottom. This is achieved by executing the process described above only once. When looping multiple times, the consequences regarding the internal space are very unfavourable. Hence, one loop is assumed to be enough here. The final step is to scale back the deformed geometry to its original perimeter. The output of this model is a shape with differential velocities between top and bottom of the SFT, depending on the **Geometry Factor**. For an overview of different shapes depending on the geometry factor, see the figure below.

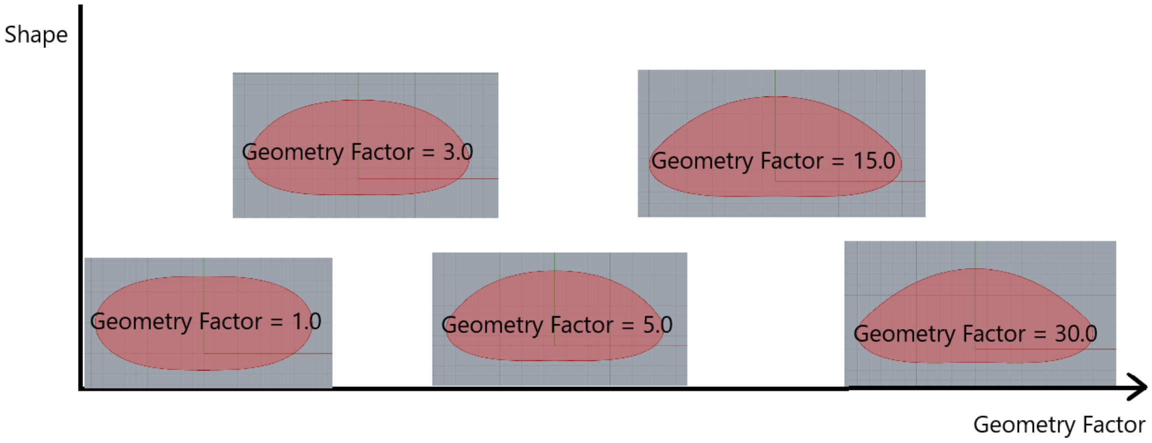


Figure B.6: The different shapes for varying geometry factor

Concluding, the output of the model described above is a shape with differential velocities between the top of the SFT and the bottom of the SFT, as was the objective. The magnitude of the differential velocities depends on the Geometry Factor. This factor is an amplification of the loads that move the deflection point of the geometry in the desired direction.
The exact difference between these velocities is calculated later on in the report. In those parts, it becomes clear that the shapes retrieved with this model indeed result in differential velocities and with a positive lift force F_{Lift}. Finally, a model overview is shown in the next section.

B.4. Model overview - currents

Here, an overview of the model described above is shown in figure B.7.

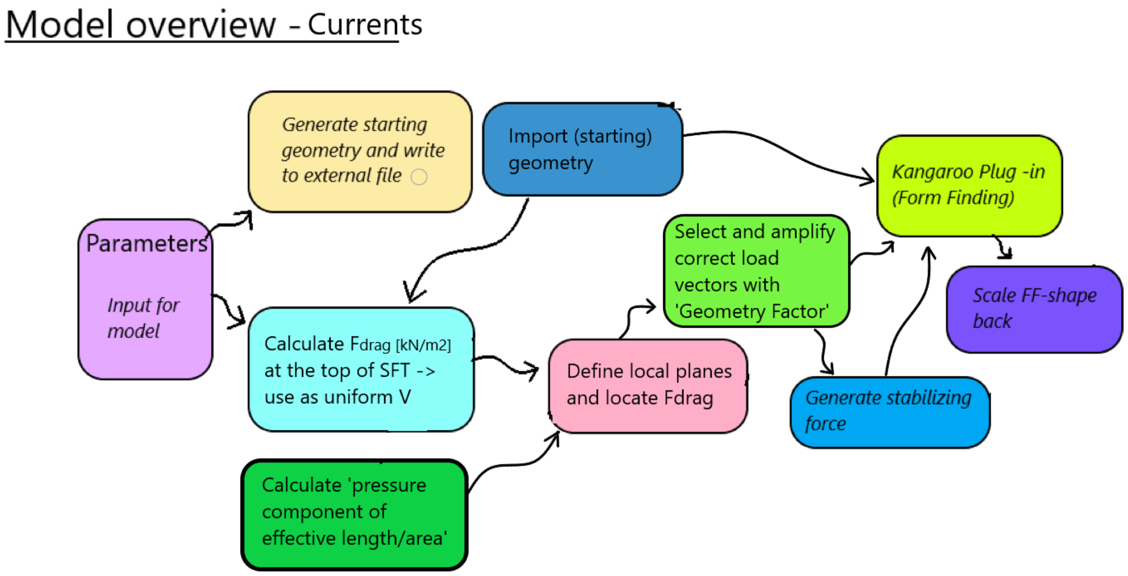

Model overview - Currents
Generate starting geometry and write to external file
Import (starting) geometry
Kangaroo Plug -in (Form Finding)
Parameters
Input for model
Select and amplify correct load vectors with 'Geometry Factor'
Scale FF-shape back
Calculate Fdrag [kN/m2] at the top of SFT -> use as uniform V
Define local planes and locate Fdrag
Generate stabilizing force
Calculate 'pressure component of effective length/area'

Figure B.7: A schematic overview of the hydrodynamic model for currents in Grasshopper

C

Model - Currents - Asymmetrical FF

In Chapter 4 the form-finding process for the currents is discussed. The loads are placed symmetrically, which can never be the case in reality. Since form-finding is directly coupled to a single load situation, it is not possible to demand a symmetrical shape out of an asymmetrical load case. As mentioned before, it is desired to have a symmetric cross-section which performs equally well for both flow directions.
However, this is the case for a single tube SFT on which the emphasis is in this research. A small 'side-road' can be entered by looking at a double tube SFT, where the asymmetrical load profile might be useful. In case of a double tube, each tube absorbs the main loads of the flow coming from that side. When the flow encounters from the other side, it is the other tube which has got to deal with the main load.

So what does the shape look like for an asymmetrical load profile. As stated in paragraph 2.3.2, a suction factor of 0.3 was assumed to be valid for the load at the lee side of the SFT. This value was based on a comparison with the factors for wind loads. For an overview of the loads acting on the starting geometry (circle), see figure C.1 below.

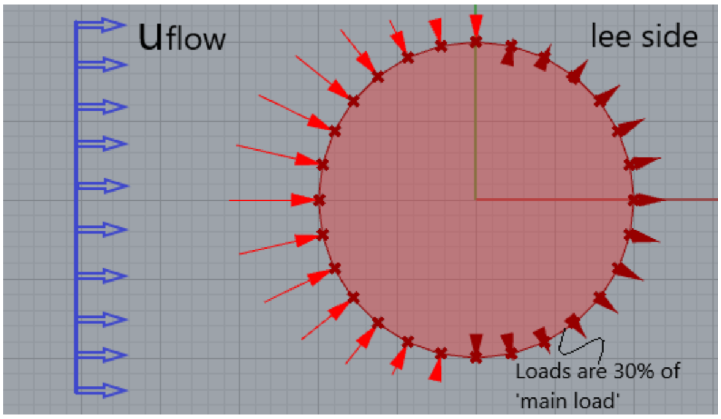


Figure C.1: The load profile for currents

The loads shown in the figure above are reversed in direction within the form finding process. Moreover, the pressure components of the drag force are shown and applied. Again, the difference in horizontal and vertical force is compensated by equally distributing the difference over all N nodes. The result is a rigid body displacement which keeps the shape at its original place, without influencing the shape. This step is explained extensively throughout the report.
The shape found by Kangaroo is shown in the left figure below. The result is shown for only one iteration.

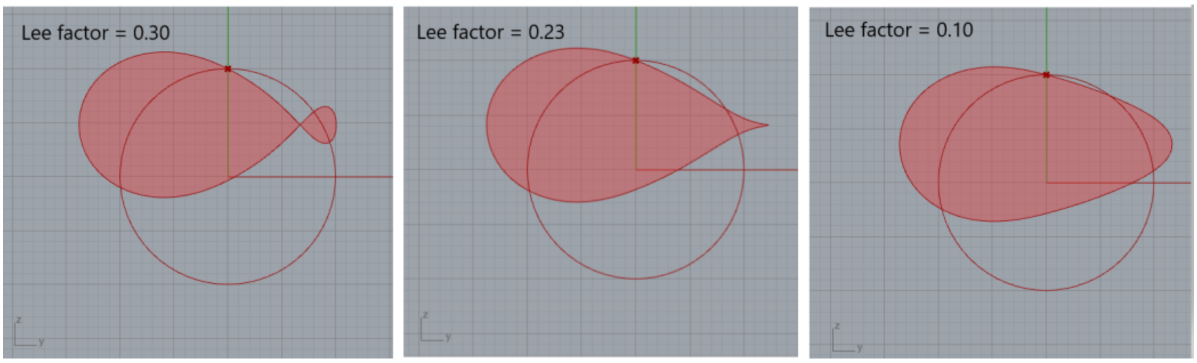


Figure C.2: The FF-shapes for different lee factors

A striking thing happens when a lee factor of 0.30 is applied for the load situation. The shape starts to 'curl up'. This has got to do with the numerical solver in Kangaroo. The mass-spring system can only find equilibrium when there is tension in the members. For a factor of 0.30, the stabilizing loads (directed to the left) are too large to guarantee tension in the members. Hence, the shape 'curls' and reaches a state with tension again. Obviously, this is not a shape which can be applied in practice. This is a constraint of the solver 'Kangaroo'.
In the center image of figure C.2, the boundary situation is found where the shape is an exact water drop, but then rotated. The load in this particular situation is a horizontal flow, which can be represented by a gravitational force rotated with 90°. A natural phenomena is mimicked with the Kangaroo solver here. On the right in figure C.2, a lee factor of 0.10 is applied. In this situation, no compression is found in the members meaning that the form-finding method performs well.

The constraint of Kangaroo can be solved by implementing a certain bending stiffness in the members. Remember the definition of form-finding:

*"Exert the loads on a material with **no** bending stiffness, to find the shape which diverts the loads axially or in-plane.*
This is also the reason why Kangaroo cannot solve a situation with compression in the members, since they have no (bending) stiffness. By using rods, with a certain axial/bending stiffness, the compressive forces can be absorbed. Although this is actually out of range for the form-finding definition, it is a solution to find a usable shape. And since all members have the same stiffness, the differences are relative. With some stiffness added, the FF-shape looks like in figure C.3 below. An additional note on the stiffness used is given at the end of this chapter.

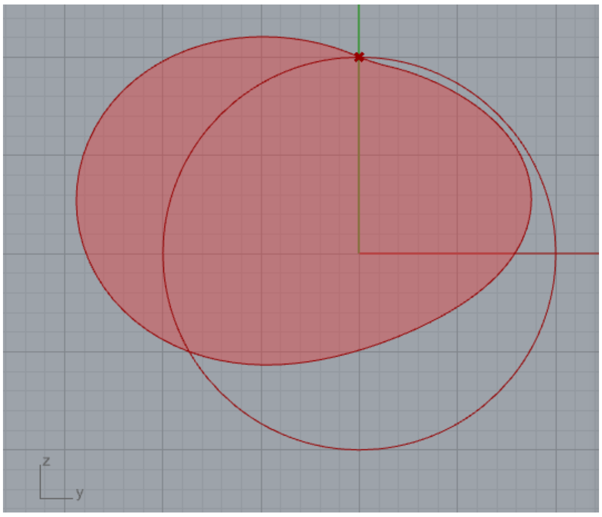


Figure C.3: The FF-shapes with added bending stiffness and lee factor = 0.30

This shape can be used in a double tube design. The shape shown is the one which should reach towards a net vertical force equal to zero. To guarantee tension, the load profile must be manipulated with a Geometry Factor again, as shown in Appendix B.

C.1. Double tube design

The shape found for asymmetrical loading conditions, which is basically always the case in reality, can be used for a double tube design. The demand for a symmetrical shape, which is desired since the flow can encounter the SFT from both sides, can be achieved by mirroring the asymmetrical shape around the vertical. See figure C.4 for an impression of such a double tube design.

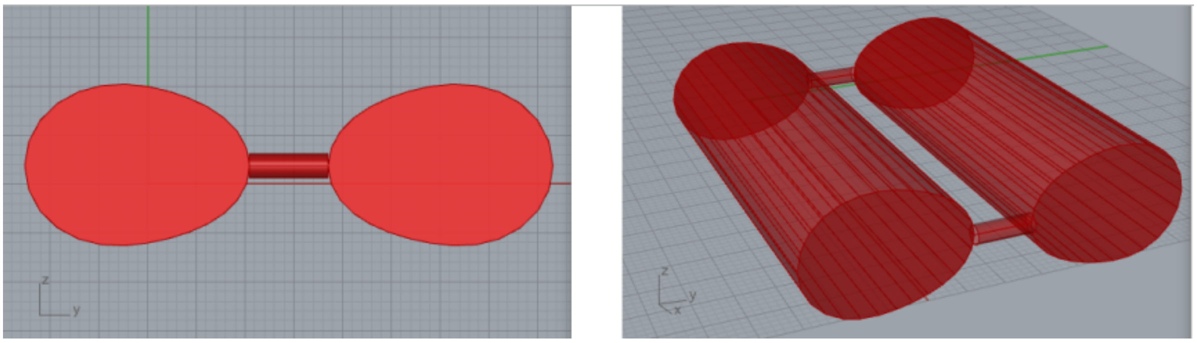


Figure C.4: The asymmetrical shape applied in double tube design

The flow enters the submerged body from one side and it separates from the lee side on. The separating flow loses energy meaning that the 'main load' has already been absorbed by the structure.
However, the interaction between the flow and the 'second tube' is dependent on the flow characteristics and the eventual shape. This is complex and not realistic to assess without a CFD or experimental knowledge. Hence, the application of a double tube design with asymmetrical load profiles goes beyond the extent of this research. It is recommended to do some further research on the behaviour of such a double tube in a flow with help of a CFD or with physical experiments.

C.2. Bending/axial stiffness

The amount of stiffness allocated to the rods strongly influences the shape. As already stated in section 2.5, the Kangaroo solver does not work with physical quantities. Both bending and axial stiffness have got a unity $'S'$, rather than $'kN/m'$ for instance.
The relative stiffness might do the work. The slice of the SFT was segmented into N elements, representing a 1 m-wide section of the actual structure. Therefore, the ratio between the bending stiffness and the axial stiffness can be simply determined for a rectangular cross-section, i.e. a segment from the SFT.

$$R = \frac{EI}{EA} = \frac{I}{A} \tag{C.1}$$

with:

$$I = \frac{1}{12} \cdot b \cdot h^3$$

$$A = b \cdot h$$

These expressions follow from basic mechanics. The eventual ratio can then be expressed as:

$$R = \frac{\frac{1}{12} bh^3}{bh} = \frac{1}{12} \cdot h^2$$

Still, the disadvantage of the Kangaroo solver remains, as the unity of the ratio R is in $[L^2]$. The question remains what the unity of this length parameter is (m, cm, mm?). When choosing for cm, the most realistic results are the output. In the figure below, some output is given for different axial stiffnesses. The bending stiffness is then automatically determined by the ratio shown above. The thickness of the SFT 'h' is kept at 80 cm. See figure C.5 for the influence of different stiffnesses.

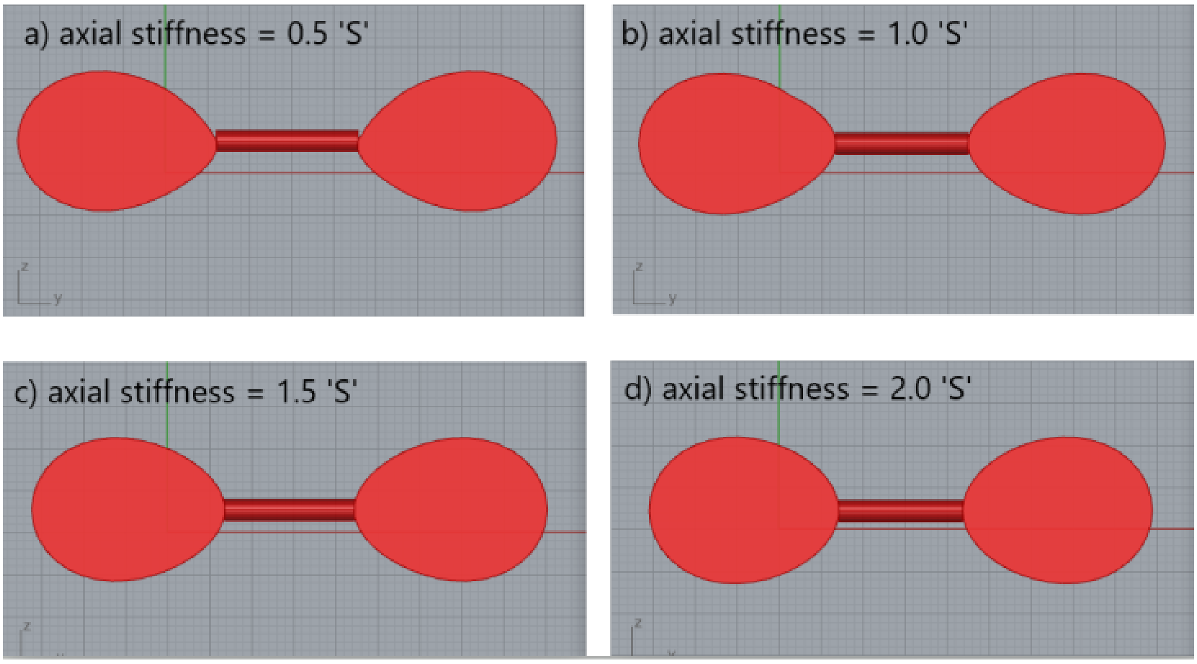

a) axial stiffness = 0.5 'S'
b) axial stiffness = 1.0 'S'
c) axial stiffness = 1.5 'S'
d) axial stiffness = 2.0 'S'
z
y

Figure C.5: The asymmetrical shape applied in double tube designs with different stiffnesses

What can be seen from the figure is the larger the stiffness, the lower the influence of the Form Finding and the closer the shape will remain to its starting geometry, a circle.
Concluding, the asymmetrical shape can be used in double tube design. However, it is strongly recommended to do some further research with physical experiments and a CFD-simulation. Moreover, the Kangaroo solver might not be the right one as it does not calculate with physical units and it is hard to express the axial/bending stiffness in a justified manner. Moreover, if a non-zero lift force is desired, apply a GF on the load profile as discussed in Appendix B.

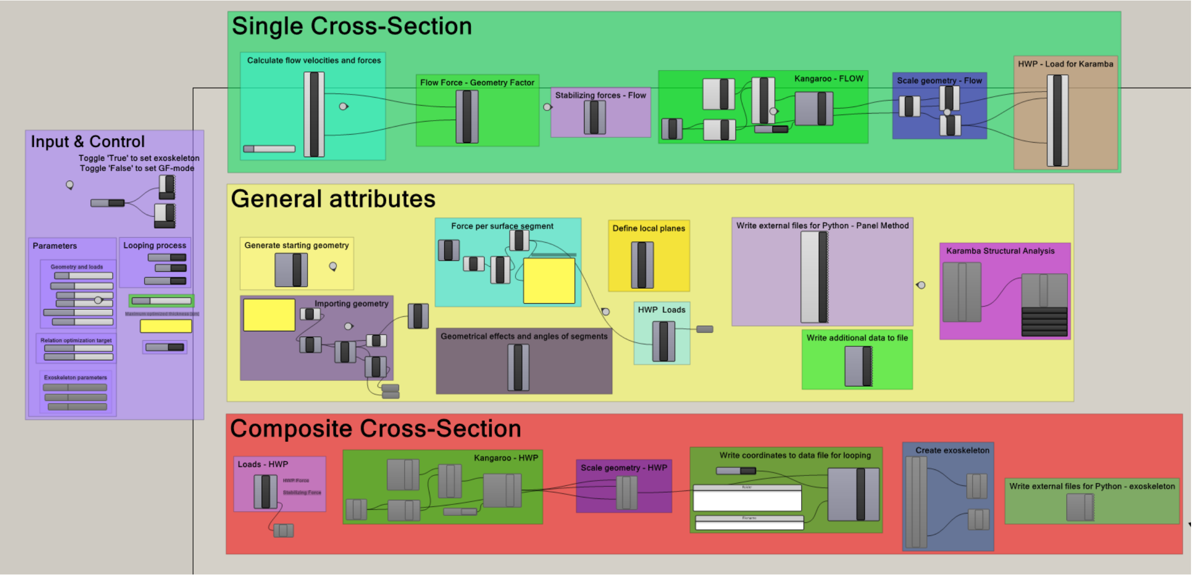
D

Total model in Grasshopper

In this Appendix, the complete model in Grasshopper is worked out. It is assumed that the reader has the basic knowledge of Grasshopper, meaning that the focus is not on the individual components and their functioning. On the contrary, the focus lies on the underlying thoughts and the connection between certain components.

The canvas in Grasshopper as shown in this Appendix is rather compact, due to the fact that many components are clustered into single ones. This is favourable in terms of the overview of the model. For most clusters, it is not shown what is inside. First of all, because the content of most clusters is already covered in Appendix A and B. Secondly, the aim/output of each cluster is shown in order to understand the **process** in Grasshopper.

The model is split into four parts. First, the 'control center' is shown where one can set the input for the model and loop for the water pressure. Within this 'control center', one also chooses for the single solution or for the composite solution, as described in Chapter 5. For both solutions, a part of 'general attributes' is required. This part is covered secondly. Thirdly and fourthly, the parts are explained for the 'single solution' and the 'composite solution' respectively. An overview of the total model and these four parts is shown below in figure D.1.



Single Cross-Section
Calculate flow velocities and forces
Flow Force - Geometry Factor
Stabilizing forces - Flow
Kangaroo - FLOW
Scale geometry - Flow
HWP - Load for Karamba
Input & Control
Toggle 'True' to set exoskeleton
Toggle 'False' to set GF-mode
Parameters
Geometry and loads
Relation optimization target
Exoskeleton parameters
Looping process
Maximum optimized thickness [cm]
General attributes
Generate starting geometry
Importing geometry
Force per surface segment
Geometrical effects and angles of segments
Define local planes
HWP Loads
Write external files for Python - Panel Method
Write additional data to file
Karamba Structural Analysis
Composite Cross-Section
Loads - HWP
Kangaroo - HWP
Scale geometry - HWP
Write coordinates to data file for looping
Create exoskeleton
Write external files for Python - exoskeleton

Figure D.1: The total model overview in GH including the four parts

D.1. Part 1: Input and Control

In this section, the ’Input and Control’ part from figure D.1 is elaborated further. First of all, one has to select which type of cross-section is generated. This is done by making use of a toggle, being either ’True’ or ’False’. This component basically enables/disables certain other components within the GH canvas, resulting in the desired output. For an overview of the toggle including the output, see figure D.2 below.

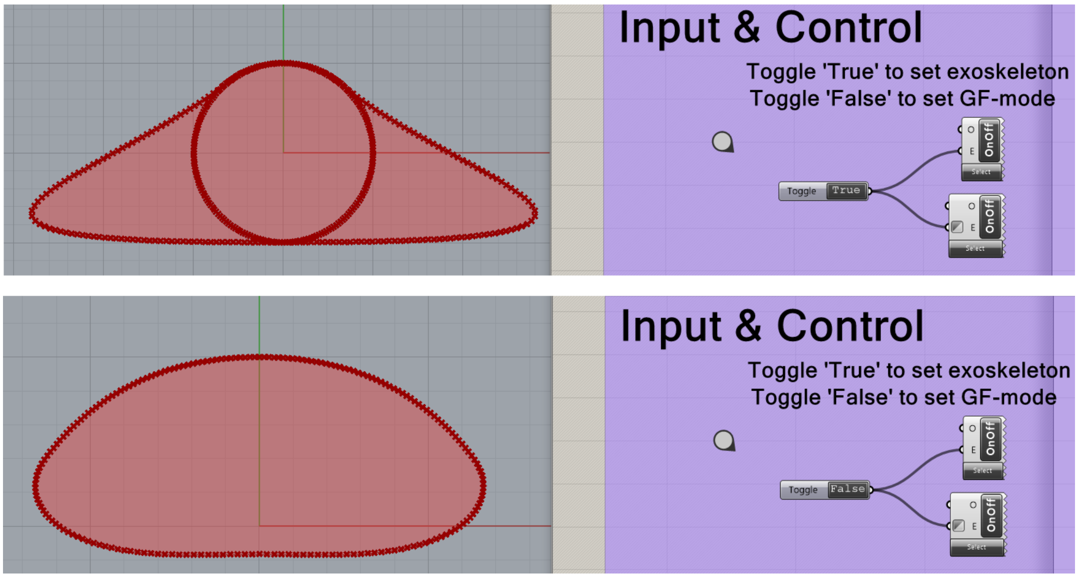

Input & Control
Toggle 'True' to set exoskeleton
Toggle 'False' to set GF-mode
Toggle True
Toggle False
OnOff
Select

Figure D.2: The selection for the single cross-section or the composite cross-section.

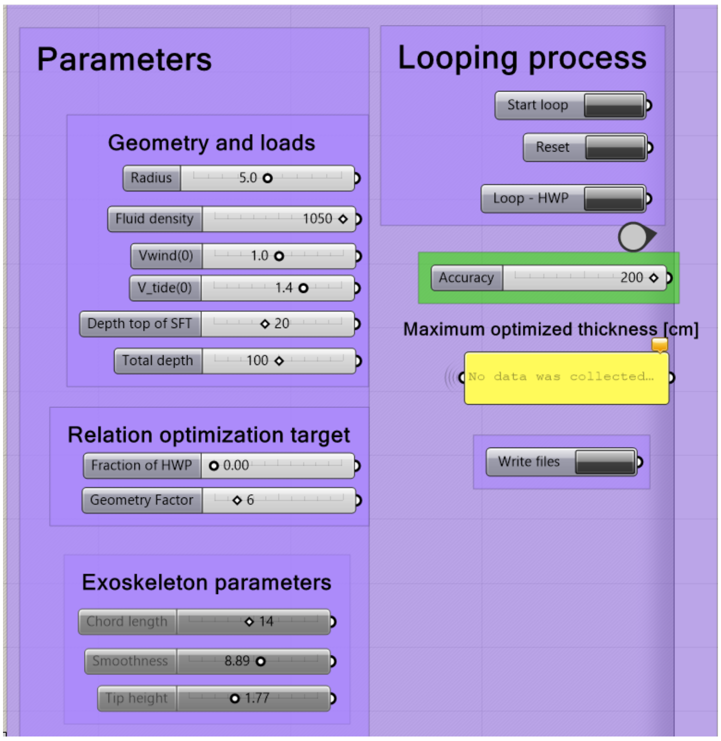
The input parameters under the header ’Geometry and Loads’ are quite self-evident. The radius determines how large the cross-section must be. Note that for the single cross-section, the shape is scaled back to the perimeter of the starting circle with this radius R. The ’fluid density’ and ’Depth on top of SFT’ are required to calculated the water pressure. $V_{wind}(0)$, $V_{tide}(0)$ and the total depth are required to calculate the flow velocity at the top of the SFT by equations 2.6 and 2.7. This flow velocity is taken as uniform within the Panel Method.

Under the header ’Relation optimization target’, the GF and the fraction of water pressure is set. These values determine the load profile which is inserted into **the same** Kangaroo Solver. Therefore, these settings belong to ’the single solution’. Concluding, the distribution of these parameters determines whether the cross-section optimizes most for material use or for the tensile force in the tethers. This is the compromise between the output from Chapters 3 and 4.

The next header ’Exoskeleton parameters’ is only required for the composite solution. Therefore, it is disabled in figure D.3 since there is chosen for the single cross-section. However, these parameters represent the ’chord length’, the ’smoothness’ and ’the tip height’. For the explicit definition of these parameters, one is referred to section 5.3.

After setting all the parameters, one goes to the right part within the ’Control center’. Before starting the looping process, one can set the accuracy of the model, shown in green. The accuracy is the number of segments and nodes in which Grasshopper divides the geometry. The larger the accuracy, the better the output but the longer the computational time will be. Then, the ’start loop’ and ’reset’ button must be pressed, in that order. The renewed starting geometry is now created and all loads as just defined are applied on the starting geometry. Notice that only if chosen for the composite cross-section, one must press the ’Loop - HWP’ button in order to take the geometrical effects into account. For more information, see Appendix A.

Finally, the ’write files’ button must be pressed to export all the geometrical properties. These files can be imported into the Python code which determines the lift force consequently. This button works for both types of solution.



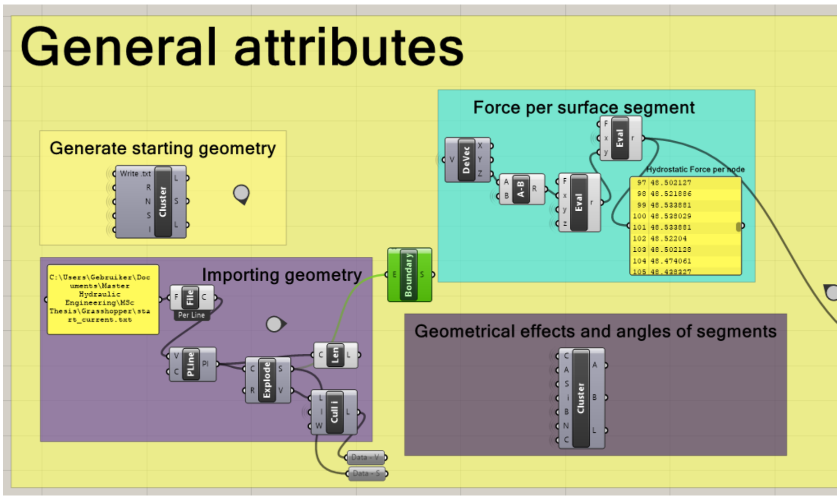
Radius 5.0
Fluid density 1050
Vwind(0) 1.0
V_tide(0) 1.4
Depth top of SFT 20
Total depth 100
Fraction of HWP 0.00
Geometry Factor 6
Chord length 14
Smoothness 8.89
Tip height 1.77
Accuracy 200

```
No data was collected…
```

Figure D.3: The other input parameters

D.2. Part 2: General Attributes

In this part, some general attributes are shown which are required for both solutions. First, the starting geometry is created based on the just defined input parameters. This is a circle with radius R, being subdivided into 'N' nodes as defined by the accuracy. Consequently, this geometry is imported in Grasshopper again in order to execute loops. This is similar as explained in Appendix A. The hydrostatic water pressure force is calculated based on the local depth of a node and by the area absorbed by each node. The area absorbed by each node is calculated within the 'geometrical effects and angles of segments' cluster. The pressure at each node is calculated according to equation 2.1. The water pressure is required for both solutions, which is why it is represented in the 'General Attributes' part.



```
C:\Users\Gebruiker\Documents\Master Hydraulic Engineering\MSc Thesis\Grasshopper\start_current.txt
```

```
 97 48.502127
 98 48.521886
 99 48.533881
100 48.538029
101 48.533881
102 48.52204
103 48.502128
104 48.474061
105 48.438327
```

Figure D.4: The general attributes - 1

Then, the local planes must be created in order to place the loads perpendicular to the geometry's surface. This is executed by generating new circles for each node, as explained in paragraph 3.3.3. Consequently, the water pressure-load is generated by applying the load vectors to the corresponding nodes. Now, all general attributes are known in order to switch to the chosen solution. These attributes are applied in either the 'single cross-section' or the 'composite cross-section', which are covered in the next sections. Once the geometries are created by these parts, they are returned to the 'General Attributes' part. Then, they are written to external files. Moreover, some additional data is written to a file. This data includes:

- Distance D perpendicular to the flow
- The Geometry Factor
- The chosen solution: single vs composite
- The (uniform) flow velocity at the top of the SFT

This additional data is required for the Python code in order to calculate F_{Lift} and C_L.

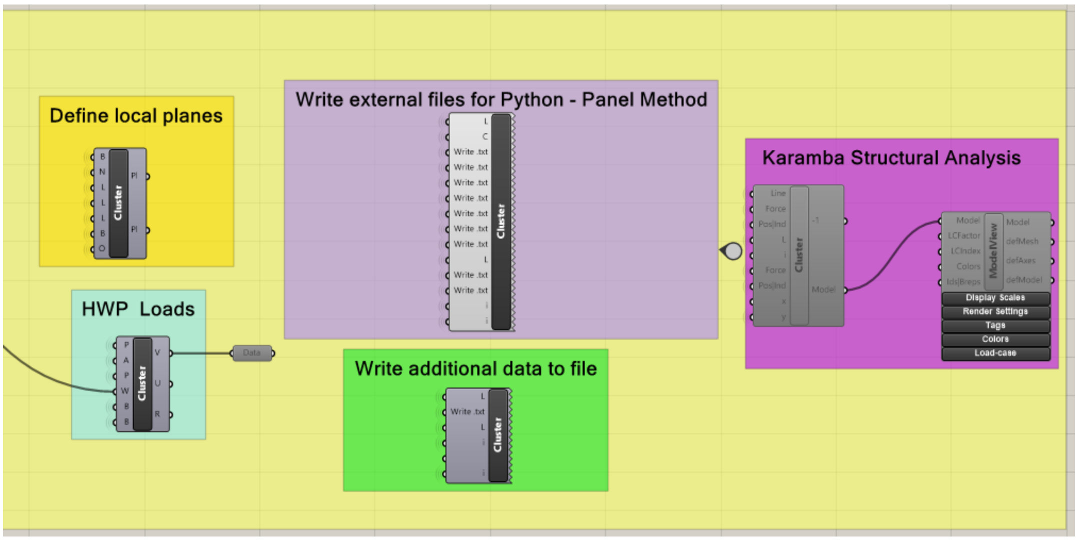


Figure D.5: The general attributes - 2

Finally, Karamba structural analysis is applied to calculate the stresses in the cross-section. The geometry is converted to structural beams. Some supports are inserted in the model at any random location. Note that the BWR is assumed to be 1.0, meaning that the location of the support is of no influence as the support reaction is zero. Then, the water pressure loads and the 'stabilizing force' are inserted into the model as well. Karamba can now calculate the cross-sectional properties (N,M and V) and optimize for the thickness of the cross-section. Moreover, the material strength can be chosen as well by setting a certain type of concrete. For now, concrete C50/60 is chosen.

D.3. Part 3: Single cross-section

As just mentioned, some general attributes are required to obtain 'the single C.S', like the local planes and the water pressure. However, this solution also includes the loads due to the flow. Hence, at the beginning of this part, the flow velocity at the top of the SFT is determined first. Consequently, these flow velocities are converted to a load according to equation 2.21. Note that these loads are equal for the left half and the right half in order to obtain a symmetrical cross-section. Then, some of these loads are amplified by the Geometry Factor. All of this is exactly the same as shown in Chapter 4 and Appendix B. The (adapted) load profile due to the loads is then used to calculate the 'stabilizing forces'. Within the latter component, also a fraction of the water pressure is included now. Note that this is different from Chapter 4. Once the fraction of water pressure is set to 0.0, the output is similar as for Chapter 4. For an overview of all these components, see figure D.6. Finally, all these loads are inserted into the Kangaroo solver as becomes clear from figure D.7.

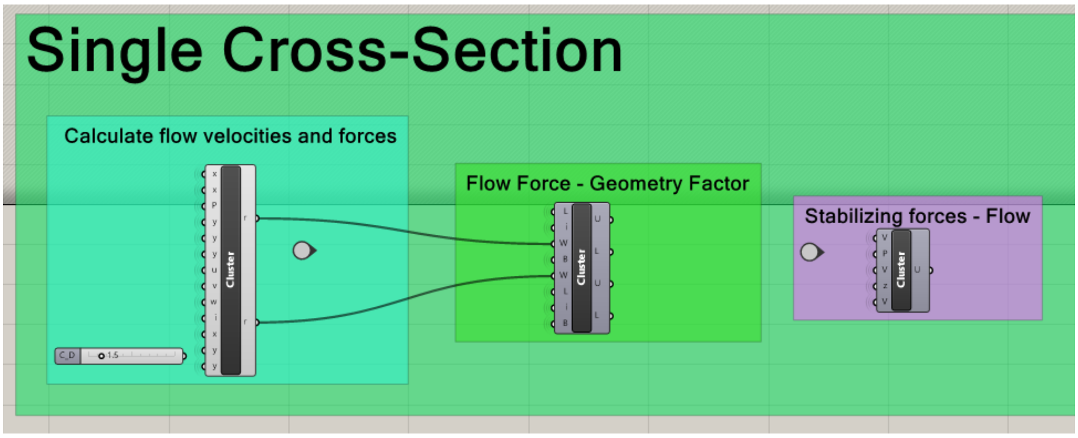

Single Cross-Section
Calculate flow velocities and forces
Cluster
C_D
1.5
Flow Force - Geometry Factor
Cluster
Stabilizing forces - Flow
Cluster

Figure D.6: The single cross-section - 1

Then, as just mentioned, the loads and geometry are inserted into the Kangaroo solver. This solver executes the Form Finding process. The deformed shape is scaled back to the original perimeter of the starting circle. Then, this renewed geometry is written to external files as shown in the 'General Attributes' part. Finally, the water pressure loads on the renewed geometry are created in order to execute the structural analysis in Karamba. This is no different from the water pressure loads on the starting geometry, but on a different cross-section.

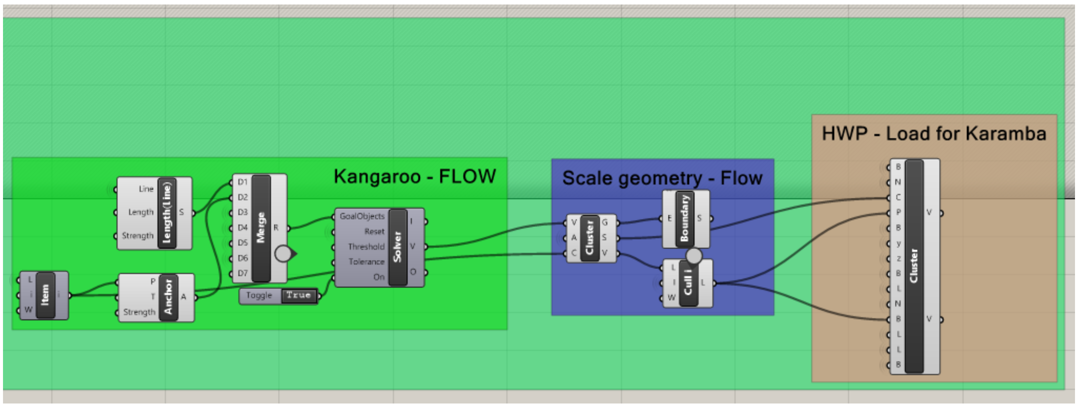


Figure D.7: The single cross-section - 2

D.4. Part 4: Composite cross-section

Similar for Part 3, some general attributes were already generated in Part 2. The water pressure load is now the only load inserted into the Kangaroo Solver. Together with the already defined local planes, the water pressure force can be inserted correctly. Moreover, the geometry is included with the 'Lenght (Line)' component. This implies that the geometry has no bending stiffness, which is by definition required for Form Finding. Consequently, the deformed shape is scaled back to its original perimeter.

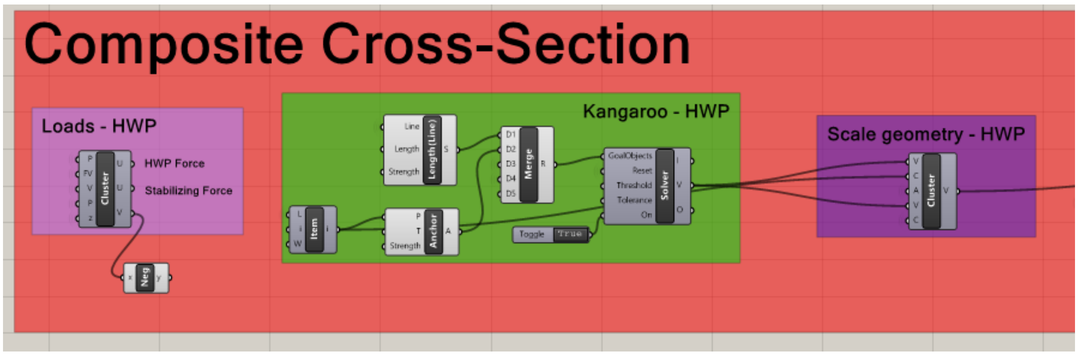
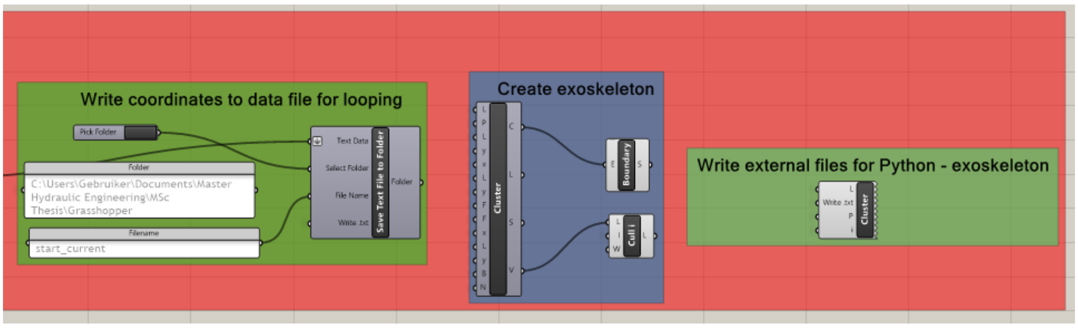


Figure D.8: The composite cross-section - 1

Another difference between the single cross-section and the composite cross-section is that the inner tube can be optimized with respect to the material use. Hence, the geometrical effects and changes in curvature should be considered here. Similar as for Chapter 3, a looping process can be executed. The scaled geometry is written to the same external file. Now Grasshopper recognizes the new output and executes the same process again. In order to execute the loop, one must press the 'Loop - HWP' button from the 'control center' as shown in figure D.3.

Now, the optimal cross-section for the inner tube is generated based on the water pressure. Note that this part is identical to the model as described in Chapter 3. Secondly, the exoskeleton is created. This exoskeleton can be manually adapted from the control center. Finally, when pressing the 'write to file' button, the exoskeleton is exported. Now, the Python code can be used to quickly estimate the lift force generated by the exoskeleton. Depending on the requirements, one can decide to accept the shape of the exoskeleton or to manually adapt it.



Folder
C:\Users\Gebruiker\Documents\Master Hydraulic Engineering\MSc Thesis\Grasshopper

Filename
start_current

Figure D.9: The composite cross-section - 2

E

Analytical derivation of velocity potential

The flow around a submerged body can be determined using a CFD. However, this requires often a lot of computational time whereas it is desired to have a quick view upon the flow velocities around the submerged body for an optimization process. The flow velocities around the submerged body determine the local fluid pressure. The fluid pressure is required to calculate the lift force on the submerged body.
In order to calculate the flow around a submerged body, the velocity potential can be used. The velocity potential was first introduced by Joseph Louis Lagrange [3]. The velocity potential ϕ_p is a general equation which describes a scalar potential of the potential flow. To start, it is important to understand that the equations hold for a 2D-situation where the x-coordinate represents the horizontal direction and the y-coordinate represents the vertical direction. Note that this is different from the rest of the report, where the z-coordinate represents the vertical direction. The equation has got the following characteristics for any point.

$$\phi_p = velocity\, potential$$

$$V_x = \frac{\partial \phi_p}{\partial x} \tag{E.1}$$

$$V_y = \frac{\partial \phi_p}{\partial y} \tag{E.2}$$

The velocity potential can be derived for any situation to calculate the velocity at any arbitrary point P. Some general expressions for this velocity potential are known for a cylinder [16]. For now, no further elaboration is given on the derivation of this velocity potential since it will not be used in this research. However, it is used as a first validation of the Panel Method which is discussed in the rest of this Appendix. In figure E.1 the flow velocity around a cylinder is shown based on polar coordinates.

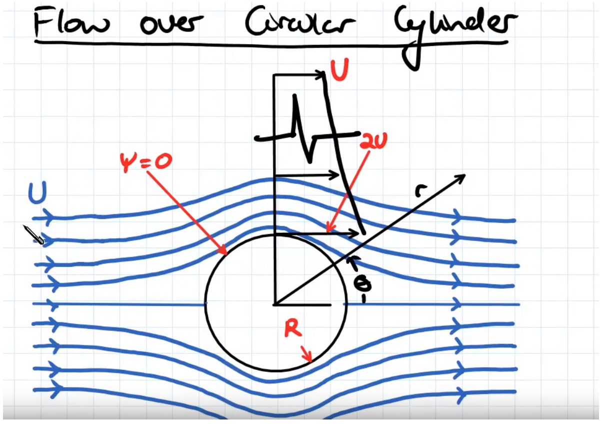


Figure E.1: The analytical flow velocity around a cylinder [16]

The flow velocity at the top and the bottom of the cylinder is equal to $2 \cdot u$ and the flow velocity right at the leading and the trailing edge are equal to zero. These values are kept in mind for now.

The main difference is that the desired geometry is not a perfect cylinder as shown in Chapter 4. The desired higher velocities at the top of the SFT require a more convexly shaped geometry at the top. See figure 4.6b for instance. Furthermore, it is now difficult to apply polar coordinates since the radius R has become a function of angle θ.
In the rest of this Appendix, the velocity potential is derived for a uniform flow over an arbitrary shape (airfoil) using the panel method. Note that the equations are linear, meaning that superposition can be applied here. The total result is the sum of the individual results.

E.1. Velocity potential - SFT in a Uniform Flow

In this section, the velocity potential for a uniform flow flowing over the SFT is derived. First, the velocity potential for a uniform flow is shown after which the same is done for the SFT. Finally, the velocity potential is elaborated and solved in terms of an actual velocity.

E.1.1. Uniform flow

For uniform flow, the velocity potential is rather easy to generate. When no submerged body would be present, the flow velocity at point P is equal to the incoming flow velocity at the boundary of the domain. Since the flow velocity in both x- and y- direction are the derivative of the velocity potential, the function can be written as:

$$\phi_{u,p} = V_\infty \cdot cos(\alpha) \cdot x + V_\infty \cdot sin(\alpha) \cdot y \tag{E.3}$$

Where:

- V_∞ = uniform flow velocity
- α = angle between flow and horizontal

This is a rather simple equation and represents the input of flow in the 'domain'. When taking the derivative of the velocity potential to x and y, V_x and V_y are retrieved respectively.

E.1.2. SFT - Panel Method Geometry

The next part of the total velocity potential function is due to the submerged body. The submerged body functions as a source/sink term to the local flow velocity. However, the submerged body can not be modelled as one point source, since it would not be able to describe the flow close to the structure accurately. Hence, the panel method is applied here [14] [1].
The idea is to discretize the geometry into sections, just as was done for the SFT before in Chapter 3 and 4. The earlier called segments are here called 'Panels'. Hence, this method is called the 'Panel Method'.
The velocity of a point source can be described as [2] [24]:

$$\phi_{p,point} = \frac{\lambda}{2 * \pi} \cdot ln(r_p) \tag{E.4}$$

The velocity potential of such a panel must be described differently. The length of the panel should be included by integrating the term over the length of the panel s. This results in the source/sink term for a panel being equal to:

$$\phi_{p,panel} = \int \frac{\lambda}{2\pi} \cdot ln(r_p) \cdot ds \tag{E.5}$$

Where:

- λ = source/sink constant of the panel
- r_p = distance between point p and the panel
- s = length of the panel

Since the geometry consists of multiple panels and the velocity potential functions are still linear, the total source term can be described as a summation of all the individual panels. Moreover, the source/sink term λ is a constant for each panel and not dependent on distance along the panel (s_j). This means that this term can be excluded from the integral. Concluding, the velocity potential for the submerged body is:

$$\phi_{p,panels} = \sum_{j=1}^{N} \frac{\lambda_j}{2*\pi} \cdot \int ln(r_{pj}(s_j) \cdot ds_j \tag{E.6}$$

Where:

- j = index of panel
- N = amount of panels (accuracy of the geometry)

Now, the total velocity potential is known for a uniform flow over the SFT. This is equal to:

$$\phi_p = V_\infty \cdot cos(\alpha) \cdot x + V_\infty \cdot sin(\alpha) \cdot y + \sum_{j=1}^{N} \frac{\lambda_j}{2*\pi} \cdot \int ln(r_{pj}(s_j) \cdot ds_j \tag{E.7}$$

E.1.3. Solving the velocity potential

The function shown in equation E.7 now must be solved. This can be done for any arbitrary point P to calculate the local flow velocity. However, there are still some unknowns in the equation above, namely the set of $\lambda' s$. When these values are not solved correctly, the flow would be able to go straight through the structure, which is physically impossible or at least undesirable. Hence, the panels each have a boundary condition. The flow velocity right at the surface can not have a normal component, because it would then flow through the submerged body. The tangential flow velocity at the surface is a certain constant C. The boundary conditions applicable for each panel is:

$$V_n = 0 \; m/s$$

Where n is the component normal to the panel's surface. Again, the benefit of the general expression for the velocity potential can be used to find the normal velocity component at the panels' surface.
Now, there are N unknowns and N boundary conditions, meaning that the set of equations is determinant and can be solved.

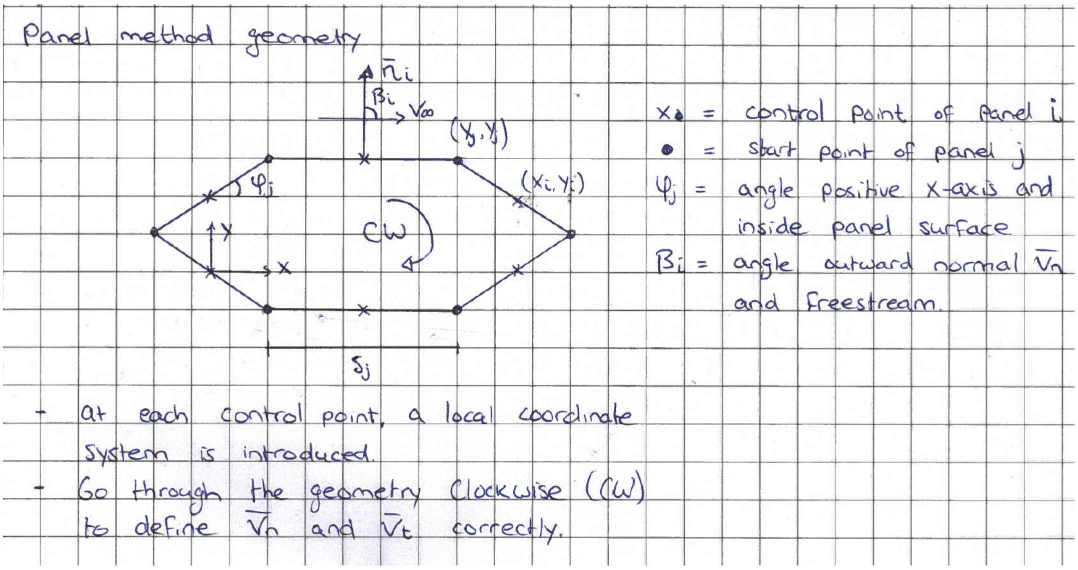
$$V_{n,i} = \frac{\partial \phi_i}{\partial n_i} = \underbrace{V_\infty \cdot cos(\alpha) \cdot \frac{\partial x_i}{\partial n}}_{1} + \underbrace{V_\infty \cdot sin(\alpha) \cdot \frac{\partial y_i}{\partial n}}_{2} + \underbrace{\sum_{j=1}^{N} \frac{\lambda_j}{2*\pi} \cdot \int \frac{\partial}{\partial n_i}(ln(r_{pj}(s_j)) \cdot ds_j}_{3} = 0 \tag{E.8}$$

Concluding, the unknown set of λ's is solved by evaluating the velocity in the normal direction at the panels' surface. This is done later on in this section.
Note, in the exact same manner the tangential velocity at the surface can be determined. The right-hand side of the equation is not zero then, but the required $\lambda' s$ are known now. Therefore, that expression can be solved too.

$$V_{t,i} = \frac{\partial \phi_i}{\partial t_i} = \underbrace{V_\infty \cdot cos(\alpha) \cdot \frac{\partial x_i}{\partial t}}_{1} + \underbrace{V_\infty \cdot sin(\alpha) \cdot \frac{\partial y_i}{\partial t}}_{2} + \underbrace{\sum_{j=1}^{N} \frac{\lambda_j}{2*\pi} \cdot \int \frac{\partial}{\partial t_i}(ln(r_{pj}(s_j)) \cdot ds_j}_{3} = C \tag{E.9}$$

Geometry definitions

Before actually solving the equations above, it is necessary to show some definitions which repeatedly show up during the derivation. Especially have a careful look at the definition of the control points x_i, y_i, which are located at the middle of the panel and the starting points of the panel x_j, y_j. It is important to make a distinction between the control points index i and the panel index j. The normal flow velocity at each panel i must be equal to zero according to the boundary condition. However, all panels j have an influence on the normal velocity component at control point i. Hence, a double loop is required later on and it is important to identify the panels with the two different indices during this double loop. Moreover, the definition of the angles is also important. See figure E.2



x• = control point of panel i

● = start point of panel j

φ_j = angle positive x-axis and inside panel surface

β_i = angle outward normal $\bar{V}_n$ and freestream.

- at each control point, a local coordinate system is introduced.
- Go through the geometry clockwise (CW) to define $\bar{V}_n$ and $\bar{V}_t$ correctly.

Figure E.2: The geometry definitions during the derivation

Finally, it is important to loop **clockwise** when going from index j to index $j+1$. This has got to do with the definition of angles and the outward normal on a panels surface.

Normal velocity

Now, the equations can be solved. As might become clear from equations E.8 and E.9, it is term 3 in both equations which causes some trouble. The terms 1 and 2 can be easily combined into one single expression depending on angle β. As shown in figure E.2, angle β is the angle between the outward normal vector of the panel and the free stream. When solving terms 1 and 2, the intermediate result for the solved equation is:

$$V_{n,i} = \underbrace{V_\infty \cdot cos(\beta_i)}_{1\, and\, 2} + I_{i,j}$$

$$V_{t,i} = \underbrace{V_\infty \cdot sin(\beta_i)}_{1\, and\, 2} + J_{i,j}$$

Where:

- $I_{i,j}$ = term 3 in equation E.8
- $J_{i,j}$ = term 3 in equation E.9

Now, the summation of the integral terms is evaluated in order to solve for the normal and tangential velocity components. The summation term Σ can be simply 'solved' in the end by adding up the solution for the integral for all panels j. Hence, the integral needs to be solved first. Note that this integral solely depends on the panel geometry. The integral to be solved in the following is shown below. Besides, the constant $\frac{\lambda_j}{2\pi}$ is excluded for the derivation and added in the end again.

$$I_{i,j} = \int_j \frac{\partial}{\partial n_i} ln(r_{i,j}) \cdot d_{sj}$$

Where:

- $r_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ = distance between control point i and panel coordinate j

Moreover, the control points of the panels are now used as the earlier mentioned 'arbitrary points' to solve for the boundary condition. Applying the chain rule and the derivative for the natural logarithm yields:

$$I_{i,j} = \int_j \frac{\partial r_{i,j}}{\partial n_i} \frac{1}{r_{i,j}} \cdot d_{sj}$$

The next step is to find the derivative of distance $r_{i,j}$ to the normal direction n.

$$\frac{\partial r_{i,j}}{\partial n_i} = \frac{1}{2} \cdot \left[(x_i - x_j)^2 + (y_i - y_j)^2)\right]^{-0.5} \cdot \left[2(x_i - x_j)\frac{\partial x_i}{\partial n_i} + 2(y_i - y_j)\frac{\partial y_i}{\partial n_i}\right]$$

When bringing down the first term in the expression above, because of the power '-0.5', and multiplying this with $\frac{1}{r_{i,j}}$, the square roots of $r_{i,j}$ are multiplied with each other which means they cancel each other out. Moreover, the $\frac{1}{2}$-term and the 2 from the expression above cancel each other out as well. The following is the result:

$$\frac{1}{r_{i,j}} \frac{\partial r_{i,j}}{\partial n_i} = \frac{(x_i - x_j) \cdot \frac{\partial x_i}{\partial n_i} + (y_i - y_j) \cdot \frac{\partial y_i}{\partial n_i}}{(x_i - x_j)^2 + (y_i - y_j)^2}$$

The next step is to eliminate the partial derivatives of x_i and y_i over n_i. For a clarification, see figure E.3. The partial derivatives can be solved applying simple trigonometry.

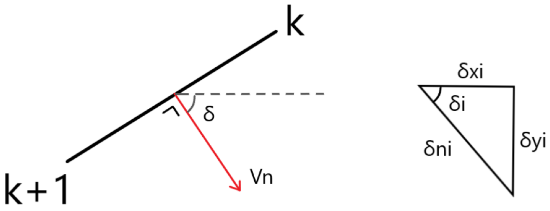


Figure E.3: The partial derivatives of xi and yi over ni

This leads to:

- $\frac{\partial x_i}{\partial n_i} = cos(\delta_i)$
- $\frac{\partial y_i}{\partial n_i} = sin(\delta_i)$
- $\delta_i = \phi_i + 90°$

And finally a total intermediate result equal to:

$$I_{i,j} = \int_j \frac{(x_i - x_j) \cdot cos(\delta_i) + (y_i - y_j) \cdot sin(\delta_i)}{(x_i - x_j)^2 + (y_i - y_j)^2} \cdot ds_j \qquad \text{(E.10)}$$

The next step is to define x_j and y_j as a function along the panels length s_j. In order to do so, the starting point of the panel is defined as X_j and Y_j. Note the difference in the capital X and Y.

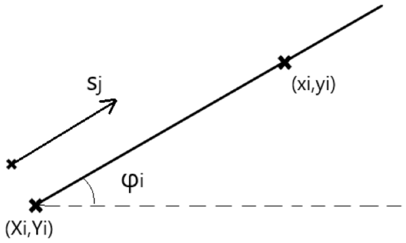


Figure E.4: The definition of x_i and y_i along panel length s_j

With:

- $x_j = X_j + s_j \cdot cos(\phi_i)$
- $y_j = Y_j + s_j \cdot sin(\phi_i)$

Consequently, the new expressions for x_j and y_j are inserted in equation E.10. Moreover, the new expression for δ_i is used in the following. For the full derivation, see [14]. The result for the numerator of the fraction in the integral is equal to:

$$sin(\phi_i - \phi_j) \cdot s_j - (x_i - X_j) \cdot sin(\phi_i) + (y_i - Y_j) \cdot cos(\phi_i)$$

Which can be written as:

$$numerator = C \cdot s_j + D$$

With:

- $C = sin(\phi_i - \phi_j)$
- $D = -(x_i - X_j) \cdot sin(\phi_i) + (y_i - Y_j) \cdot cos(\phi_i)$

Now the same can be done for the denominator. First, the terms are expanded which means the parentheses are worked out. Secondly, the new expressions for x_j and y_j are inserted resulting in the following. Again, for the full derivation see [14]. The result is an expression for the denominator in a quadratic form in terms of variable s_j. Note, this is important since the total term has to be integrated with respect to variable s_j. The result is:

$$denominator = s_j^2 + 2 \cdot s_j \left[-(x_i - X_j) \cdot cos(\phi_j) - (y_i - Y_j) \cdot sin(\phi)j)\right] + \left[(x_i - X_j)^2 + (y_i - Y_j)^2\right]$$

Again, mind the difference between indices i and j and the difference between capital X,Y and small x,y. Now, each term is ready to be integrated over variable s_j. This term can be rewritten in the form of:

$$denumerator = s_j^2 + A \cdot s_j + B$$

Where:

- $A = -(x_i - X_j) \cdot cos(\phi_j) - (y_i - Y_j) \cdot sin(\phi_j)$
- $B = (x_i - X_j)^2 + (y_i - Y_j)^2$

The following is now the intermediate result for the integral $I_{i,j}$

$$I_{i,j} = \int_0^{s_j} \frac{C \cdot s_j + D}{s_j^2 + 2A \cdot s_j + B} \tag{E.11}$$

With A, B, C and D as mentioned above.
Now, another substitution is executed for convenience purposes. The denominator can be rewritten as:

$$s_j^2 + 2A \cdot s_j + B = (s_j + A)^2 + (B - A^2)$$

With additional parameter $E = \sqrt{B - A^2}$ which yields:

$$I_{i,j} = \int_0^{s_j} \frac{C \cdot s_j + D}{(s_j + A)^2 + E^2}$$

Then, after some substitutions in the derivation of this integral the result is an expression for $I_{i,j}$ without any difficult terms to solve. Since the derivation is just math and does not include any fundamental steps in defining some variables or terms, it is excluded here. For the full derivation, again see [14]. The result for the integral is:

$$I_{i,j} = \frac{C}{2} \cdot ln\left[\frac{s_j^2 + 2As_j + B}{B}\right] + \frac{D - AC}{E}\left[tan^{-1}(\frac{s_j + A}{E}) - tan^{-1}(\frac{A}{E})\right] \tag{E.12}$$

With:

- $A = -(x_i - X_j) \cdot cos(\phi_j) - (y_i - Y_j) \cdot sin(\phi_j)$
- $B = (x_i - X_j)^2 + (y_i - Y_j)^2$
- $C = sin(\phi_i - \phi_j)$

- $D = -(x_i - X_j) \cdot sin(\phi_i) + (y_i - Y_j) \cdot cos(\phi_i)$
- $E = \sqrt{B - A^2}$

The final step in solving the integral is adding up all the different panels which influence the velocity potential at an arbitrary point p.

$$I_{i,j} = \sum_{j=1}^{N} \left(\frac{\lambda_j}{2 * \pi} \cdot \frac{C}{2} \cdot ln \left[\frac{s_j^2 + 2As_j + B}{B} \right] + \frac{D - AC}{E} \left[tan^{-1}(\frac{s_j + A}{E}) - tan^{-1}(\frac{A}{E}) \right] \right)$$

Now the difficult term 3 from equation E.8 is solved, the total normal velocity at the panels' surface can be expressed and set to zero. The N equations and N unknowns can be solved for unknown λ, which is the source term for each panel. Remind that this solves for a situation where the flow can not pass through the body.

An additional explanation for the physical meaning of integral $I_{i,j}$ is: the geometrical influence of panel j on control point i. Concluding, all N panels have an influence on the normal velocity at control point i. So $I_{2,5}$ would show the geometrical influence of panel 5 on the normal velocity at panel 2. Since all panels j have an influence on the normal velocity at panel i, the I-terms are written in matrix shape. The total set of equations looks as:

$$V_{n,i} = \begin{pmatrix} \frac{I_{11}}{2\pi} & \frac{I_{12}}{2\pi} & \cdots & \cdots & \frac{I_{1n}}{2\pi} \\ \frac{I_{21}}{2\pi} & \frac{1}{2} & & & \\ \vdots & & \ddots & & \\ \vdots & & & \ddots & \\ \frac{I_{n1}}{2\pi} & & & & \frac{I_{nn}}{2\pi} \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} -V_\infty \cdot cos(\beta_1) \\ -V_\infty \cdot cos(\beta_2) \\ \vdots \\ \vdots \\ -V_\infty \cdot cos(\beta_n) \end{pmatrix}$$

Note that the diagonal has got a value of $\frac{1}{2}$. This is in line with mathematical theory which is not elaborated further here. The diagonal represents the influence of a panel on its own normal velocity.
The output is now a set of $\lambda' s$ belonging to the panel geometry.

Tangential velocity
The next step is to calculate the tangential velocity at the surface, since this is of interest when determining the net lift force on the structure. A similar-like derivation can be done in determining the integral $J_{i,j}$ which is required for the tangential velocity. This derivation is not shown here. The output is:

$$J_{i,j} = \frac{C}{2} \cdot ln \left[\frac{s_j^2 + 2As_j + B}{B} \right] + \frac{D - AC}{E} \left[tan^{-1}(\frac{s_j + A}{E}) - tan^{-1}(\frac{A}{E}) \right] \tag{E.13}$$

With:

- $A = -(x_i - X_j) \cdot cos(\phi_j) - (y_i - Y_j) \cdot sin(\phi_j)$
- $B = (x_i - X_j)^2 + (y_i - Y_j)^2$
- $C = -cos(\phi_i - \phi_j)$
- $D = (x_i - X_j) \cdot cos(\phi_i) + (y_i - Y_j) \cdot sin(\phi_i)$
- $E = \sqrt{B - A^2}$

Note that only the expressions for C and D are slightly different. The tangential velocity at the panels surface can now be calculated by:

$$V_{t,i} = \begin{pmatrix} \frac{J_{11}}{2\pi} & \frac{J_{12}}{2\pi} & \cdots & \cdots & \frac{J_{1n}}{2\pi} \\ \frac{J_{21}}{2\pi} & 0 & & & \\ \vdots & & \ddots & & \\ \vdots & & & \ddots & \\ \frac{J_{n1}}{2\pi} & & & & \frac{J_{nn}}{2\pi} \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \vdots \\ \lambda_n \end{pmatrix} + \begin{pmatrix} V_\infty \cdot sin(\beta_1) \\ V_\infty \cdot sin(\beta_2) \\ \vdots \\ \vdots \\ V_\infty \cdot sin(\beta_n) \end{pmatrix}$$

Velocity potential at arbitrary point P

In order to calculate the total velocity profile throughout the cross-section, the original velocity potential must be solved as was shown in equations E.7. The horizontal and vertical velocities can be retrieved by differentiating the velocity potential to the corresponding direction, as stated in the start of this Appendix. This is done in a similar way as for the normal and tangential velocity. However, this derivation is easier since the terms can be simply derived to the Cartesian coordinate system.

$$V_{x,p} = \frac{\partial \phi_p}{\partial x} = V_\infty \cdot cos(\alpha) + \frac{\lambda_j}{2\pi} \sum_{j=1}^{N} \frac{\partial}{\partial x} \int_j ln(r_{pj}(s_j)) \cdot ds_j \tag{E.14}$$

With:

- $r_{pj} = \sqrt{(x_p - x_j)^2 + (y_p - y_j)^2}$

Assuming that the partial lengths are relatively small, which is the case for the SFT as the accuracy is set to a large number, the distance along the panel to arbitrary point P is averaged. This means that variables x_j and y_j are independent of s_j. The integral term then simply becomes:

$$\int_j ln(r_{pj}) \cdot ds_j = ln(r_{pj}) \cdot s_j$$

The next step is to take the partial derivative of this term to x. When applying the derivative rule for the natural log and applying the chain rule (2 times), the result is:

$$\frac{\partial}{\partial x} ln(\sqrt{(x_p - x_j)^2 + (y_p - y_j)^2}) \cdot s_j =$$

$$\frac{s_j}{\sqrt{(x_p - x_j)^2 + (y_p - y_j)^2}} \cdot \frac{1}{2 \cdot \sqrt{(x_p - x_j)^2 + (y_p - y_j)^2}} \cdot 2(x_p - x_j) =$$

$$\frac{s_j(x_p - x_j)}{(x_p - x_j)^2 + (y_p - y_j)^2}$$

This results in the total expression for V_x at point P being equal to:

$$V_x = V_\infty \cdot cos(\alpha) + \sum_{j=1}^{N} \frac{\lambda_j}{2\pi} \cdot \frac{s_j(x_p - x_j)}{(x_p - x_j)^2 + (y_p - y_j)^2} \tag{E.15}$$

The same thing can be done for the vertical velocity at point P, which results in:

$$V_y = V_\infty \cdot sin(\alpha) + \sum_{j=1}^{N} \frac{\lambda_j}{2\pi} \cdot \frac{s_j(y_p - y_j)}{(x_p - x_j)^2 + (y_p - y_j)^2} \tag{E.16}$$

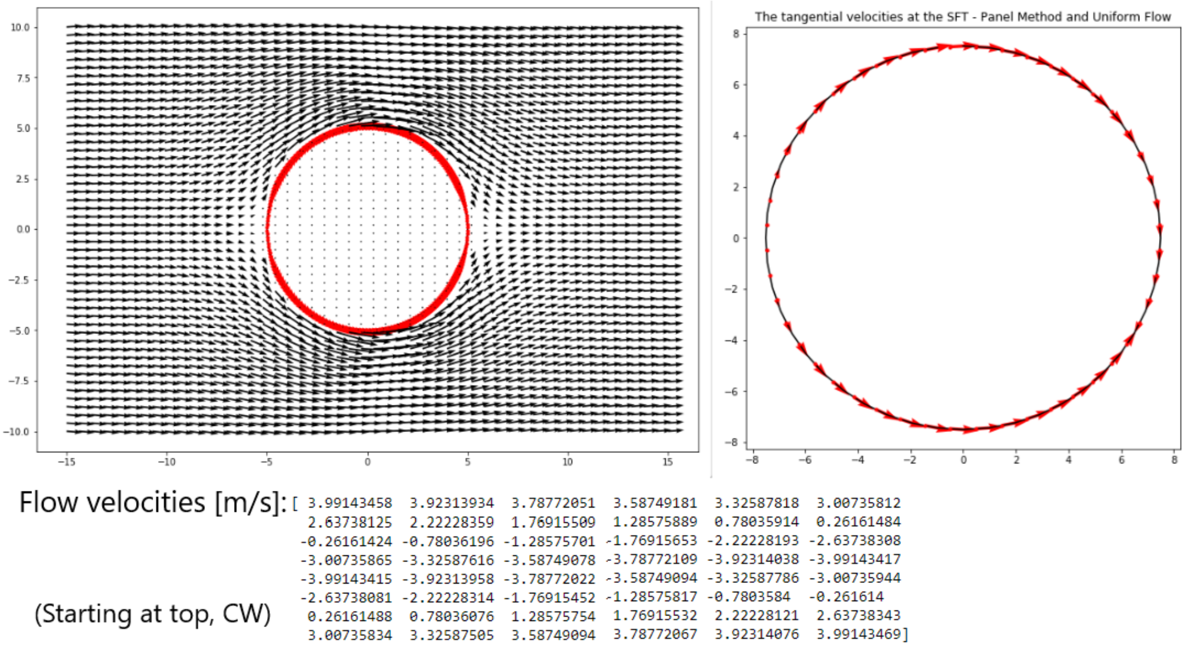
Conclusion

Concluding, the total velocity field can now be solved by means of the velocity potential. First, the unknown source/sink terms of the submerged geometry are solved by setting a boundary condition at the geometry's surface. The normal velocity must be zero, meaning that the flow can not flow through the body. This is also physically understandable. After the source/sink terms (λ) are solved, the tangential velocity at the surface can be calculated as well as the velocity at any arbitrary point. The output is shown in section E.2.

E.2. Velocity potential - Total result

Some results of the analytical derivation of the flow along a geometry are shown in this section. First, a circular cross-section is tested in order to check the theory as was shown in figure E.1. Here, it was predicted that the flow velocity at the top and the bottom of a cylinder is equal to $2 \cdot U$ or in this case $2 \cdot V_\infty$. The case tested here is:

- $V_\infty = 2\ m/s$
- $\rho = 1050\ kg/m^3$



Flow velocities [m/s]:

(Starting at top, CW)

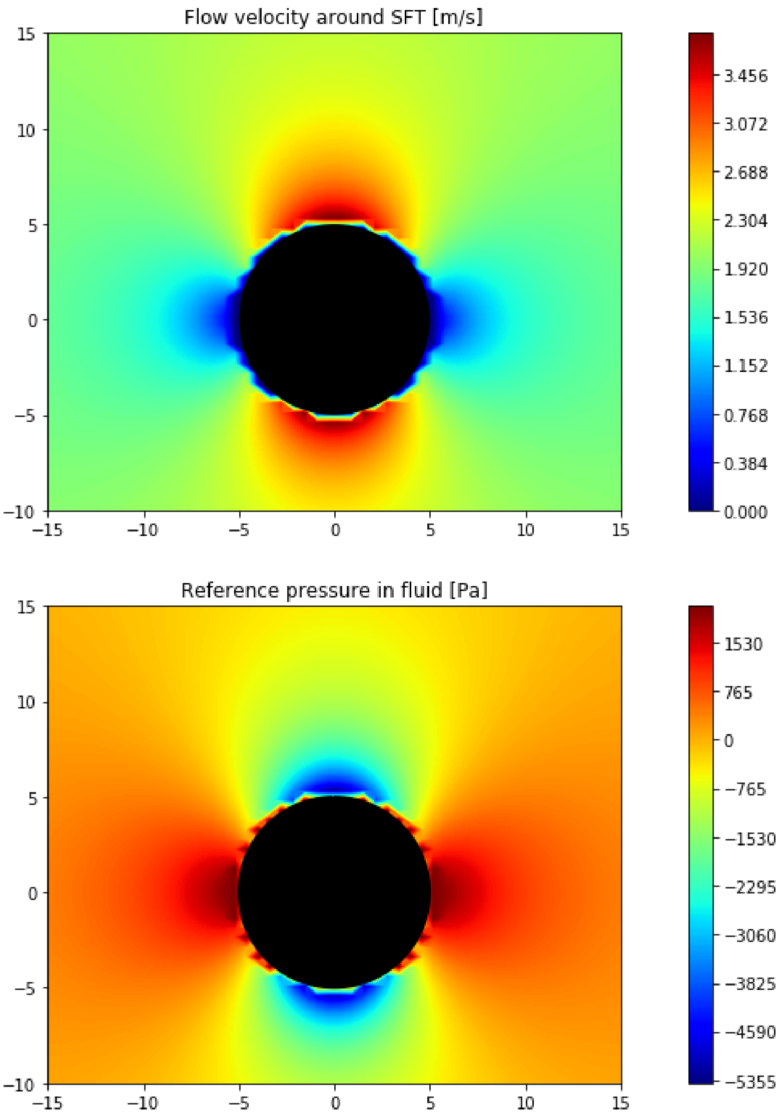
```
[ 3.99143458  3.92313934  3.78772051  3.58749181  3.32587818  3.00735812
  2.63738125  2.22228359  1.76915509  1.28575889  0.78035914  0.26161484
 -0.26161424 -0.78036196 -1.28575701 -1.76915653 -2.22228193 -2.63738308
 -3.00735865 -3.32587616 -3.58749078 -3.78772109 -3.92314038 -3.99143417
 -3.99143415 -3.92313958 -3.78772022 -3.58749094 -3.32587786 -3.00735944
 -2.63738081 -2.22228314 -1.76915452 -1.28575817 -0.7803584  -0.261614
  0.26161488  0.78036076  1.28575754  1.76915532  2.22228121  2.63738343
  3.00735834  3.32587505  3.58749094  3.78772067  3.92314076  3.99143469]
```

Figure E.5: The flow profile along a cylinder for $V_\infty = 2\ m/s$

Right, the tangential velocity is shown along the structures' surface. It is visible that it is approximately zero at the leading and the trailing edge. When the normal velocity would be plotted, it is indeed zero. At the left, the total velocity profile is plotted. Note that some numerical errors exist due to the inaccurate grid. However, if the grid would become too dense, the plot would not be readable anymore. What is visible is that the flow starts to curve around the cylinder already before it reaches the structure. This is in line with expectations. Besides, the boundary condition is achieved as well, which stated that the flow can not go through the submerged body. It nicely streams along the cylinder.

Furthermore, when looking at the array at the bottom of figure E.5 the flow velocities can be seen. It starts at the top of the circle and goes all the way around in clockwise direction. The flow velocity at the top and the bottom is 3.99 m/s, which is basically twice the uniform flow velocity. This is in line with the general theory presented at the start of this Appendix.

In figure E.6, the velocity and pressure contour plot is shown for the velocity profile shown above. The pressure is calculated using Bernoulli's principle. See equation 4.1. Again, note that some numerical errors are visible due to the 'inaccuracy' of the grid.



Flow velocity around SFT [m/s]

Reference pressure in fluid [Pa]

Figure E.6: The velocity and pressure profile

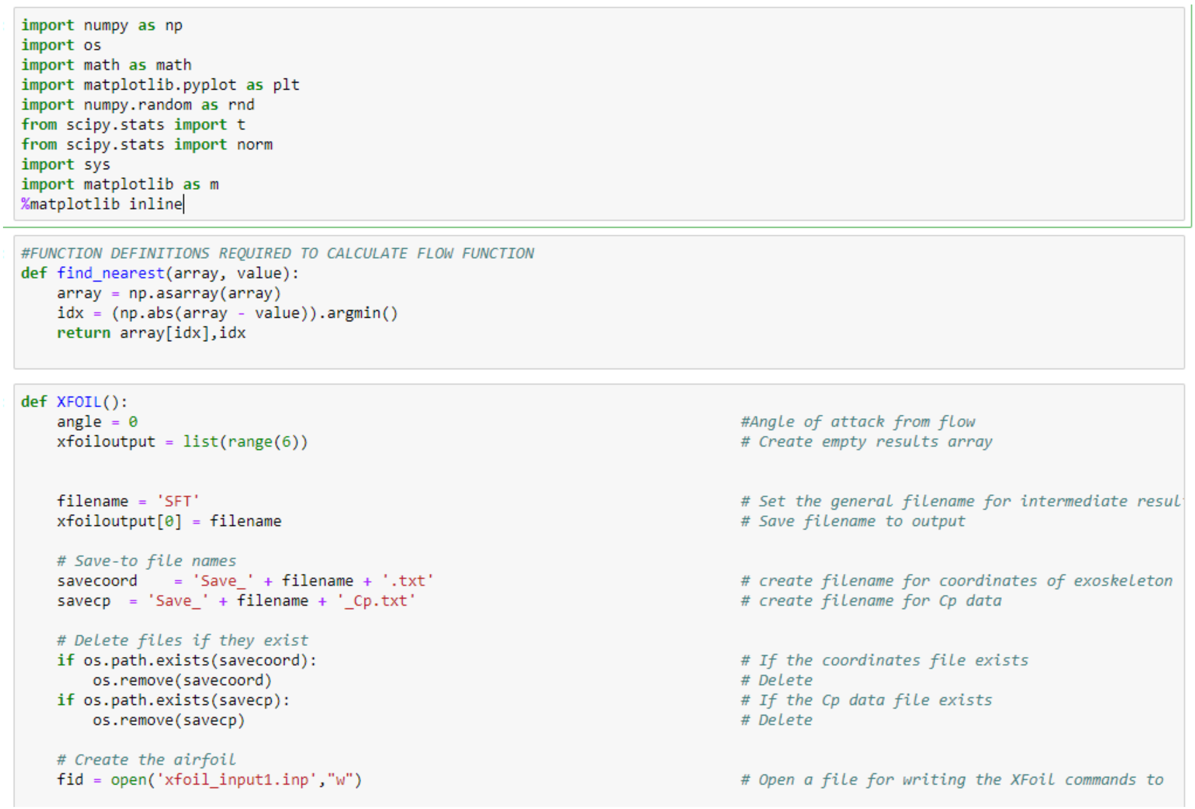
The script coded in Python can execute this Panel Method for any random geometry. The geometry is generated and exported from Grasshopper. The output data from GH for the geometry to be analysed by the code is:

- Control coordinates (x_i, y_i)
- Boundary coordinates (X_j, Y_j)
- Panel length s_j
- Angle β between freestream and outward normal
- Angle phi_i

F

Panel Method in Python

In this Chapter, the code in Python is shown. This code is linked with the results found in Grasshopper/Rhinoceros via external files. One must press the 'write to file' button in Grasshopper. Make sure that this Python Notebook is in the same folder as the data which is written to the external file. Otherwise, Python can not import them and work out the Panel Method.
Moreover, when the 'exoskeleton' is switched on in Grasshopper, Python will evaluate the panel method with xFoil. This programme 'xFoil.exe' must also be within the same folder as the Python Notebook. xFoil is an external software program which also executes the Panel Method. In addition, it is capable of analyzing viscous flow by making an approximation for the boundary layer equations. These approximations only hold for 'airfoil like' structures, with a sharp tip at the leading edge and trailing edge. For more information, see the main report. Behind most parts of the code, some explanation is given.

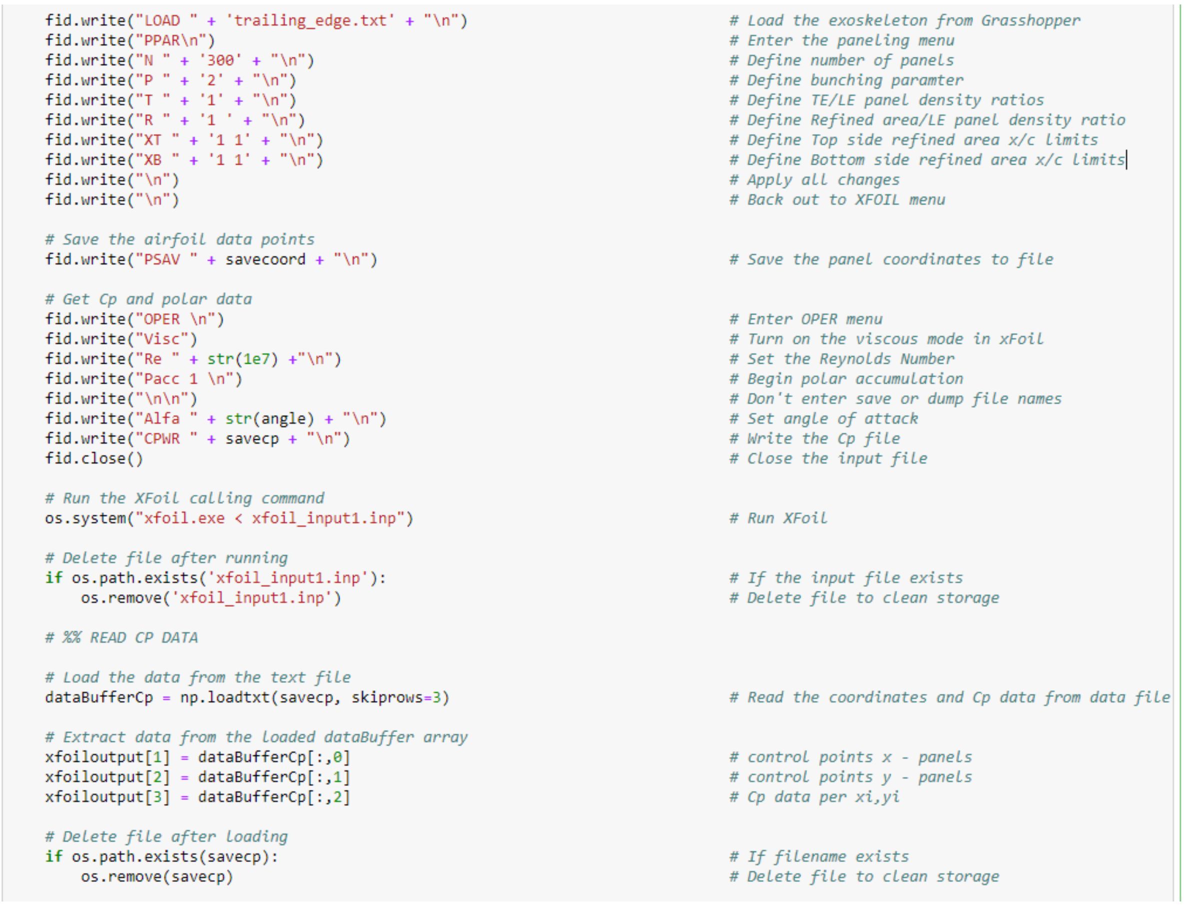


```
import numpy as np
import os
import math as math
import matplotlib.pyplot as plt
import numpy.random as rnd
from scipy.stats import t
from scipy.stats import norm
import sys
import matplotlib as m
%matplotlib inline
```

```
#FUNCTION DEFINITIONS REQUIRED TO CALCULATE FLOW FUNCTION
def find_nearest(array, value):
    array = np.asarray(array)
    idx = (np.abs(array - value)).argmin()
    return array[idx],idx
```

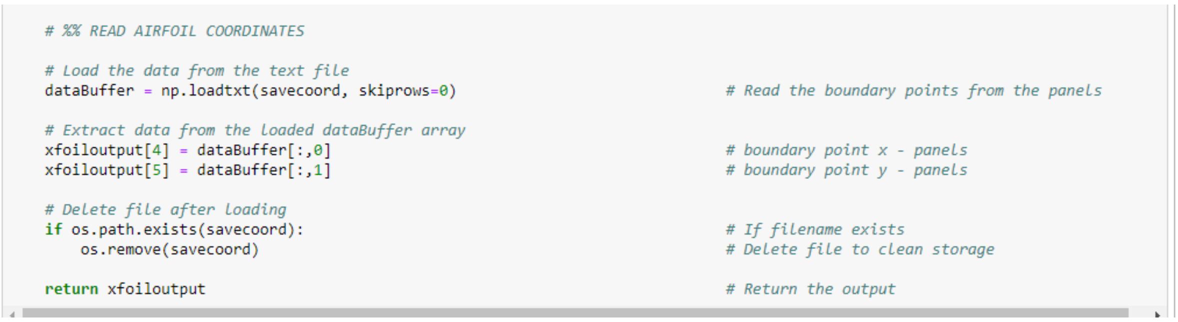
```
def XFOIL():
    angle = 0                                                                   #Angle of attack from flow
    xfoiloutput = list(range(6))                                                # Create empty results array


    filename = 'SFT'                                                            # Set the general filename for intermediate resul
    xfoiloutput[0] = filename                                                   # Save filename to output

    # Save-to file names
    savecoord    = 'Save_' + filename + '.txt'                                  # create filename for coordinates of exoskeleton
    savecp  = 'Save_' + filename + '_Cp.txt'                                    # create filename for Cp data

    # Delete files if they exist
    if os.path.exists(savecoord):                                               # If the coordinates file exists
        os.remove(savecoord)                                                    # Delete
    if os.path.exists(savecp):                                                  # If the Cp data file exists
        os.remove(savecp)                                                       # Delete

    # Create the airfoil
    fid = open('xfoil_input1.inp',"w")                                          # Open a file for writing the XFoil commands to
```

Figure F.1: The python script - 1



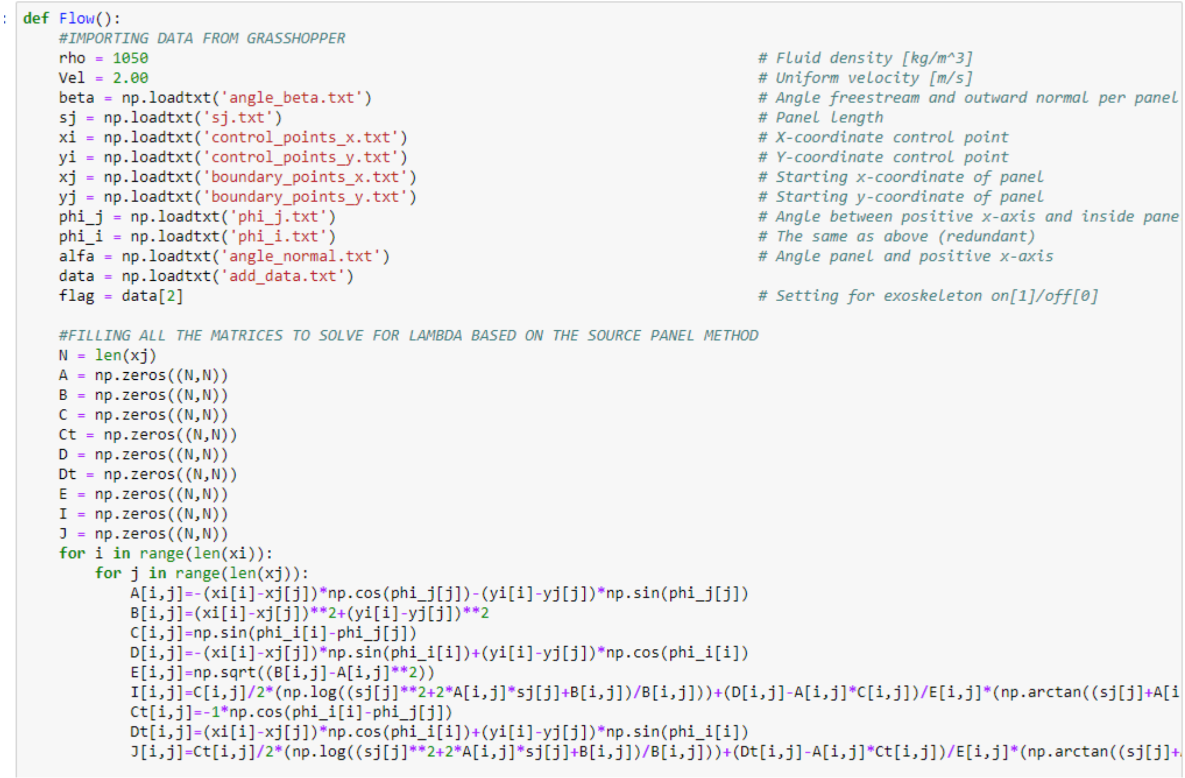
```
    fid.write("LOAD " + 'trailing_edge.txt' + "\n")                            # Load the exoskeleton from Grasshopper
    fid.write("PPAR\n")                                                          # Enter the paneling menu
    fid.write("N " + '300' + "\n")                                               # Define number of panels
    fid.write("P " + '2' + "\n")                                                 # Define bunching paramter
    fid.write("T " + '1' + "\n")                                                 # Define TE/LE panel density ratios
    fid.write("R " + '1 ' + "\n")                                                # Define Refined area/LE panel density ratio
    fid.write("XT " + '1 1' + "\n")                                              # Define Top side refined area x/c limits
    fid.write("XB " + '1 1' + "\n")                                              # Define Bottom side refined area x/c limits
    fid.write("\n")                                                              # Apply all changes
    fid.write("\n")                                                              # Back out to XFOIL menu

    # Save the airfoil data points
    fid.write("PSAV " + savecoord + "\n")                                        # Save the panel coordinates to file

    # Get Cp and polar data
    fid.write("OPER \n")                                                         # Enter OPER menu
    fid.write("Visc")                                                            # Turn on the viscous mode in xFoil
    fid.write("Re " + str(1e7) +"\n")                                            # Set the Reynolds Number
    fid.write("Pacc 1 \n")                                                       # Begin polar accumulation
    fid.write("\n\n")                                                            # Don't enter save or dump file names
    fid.write("Alfa " + str(angle) + "\n")                                       # Set angle of attack
    fid.write("CPWR " + savecp + "\n")                                           # Write the Cp file
    fid.close()                                                                  # Close the input file

    # Run the XFoil calling command
    os.system("xfoil.exe < xfoil_input1.inp")                                    # Run XFoil

    # Delete file after running
    if os.path.exists('xfoil_input1.inp'):                                       # If the input file exists
        os.remove('xfoil_input1.inp')                                            # Delete file to clean storage

    # %% READ CP DATA

    # Load the data from the text file
    dataBufferCp = np.loadtxt(savecp, skiprows=3)                                # Read the coordinates and Cp data from data file

    # Extract data from the loaded dataBuffer array
    xfoiloutput[1] = dataBufferCp[:,0]                                           # control points x - panels
    xfoiloutput[2] = dataBufferCp[:,1]                                           # control points y - panels
    xfoiloutput[3] = dataBufferCp[:,2]                                           # Cp data per xi,yi

    # Delete file after loading
    if os.path.exists(savecp):                                                   # If filename exists
        os.remove(savecp)                                                        # Delete file to clean storage
```

Figure F.2: The python script - 2



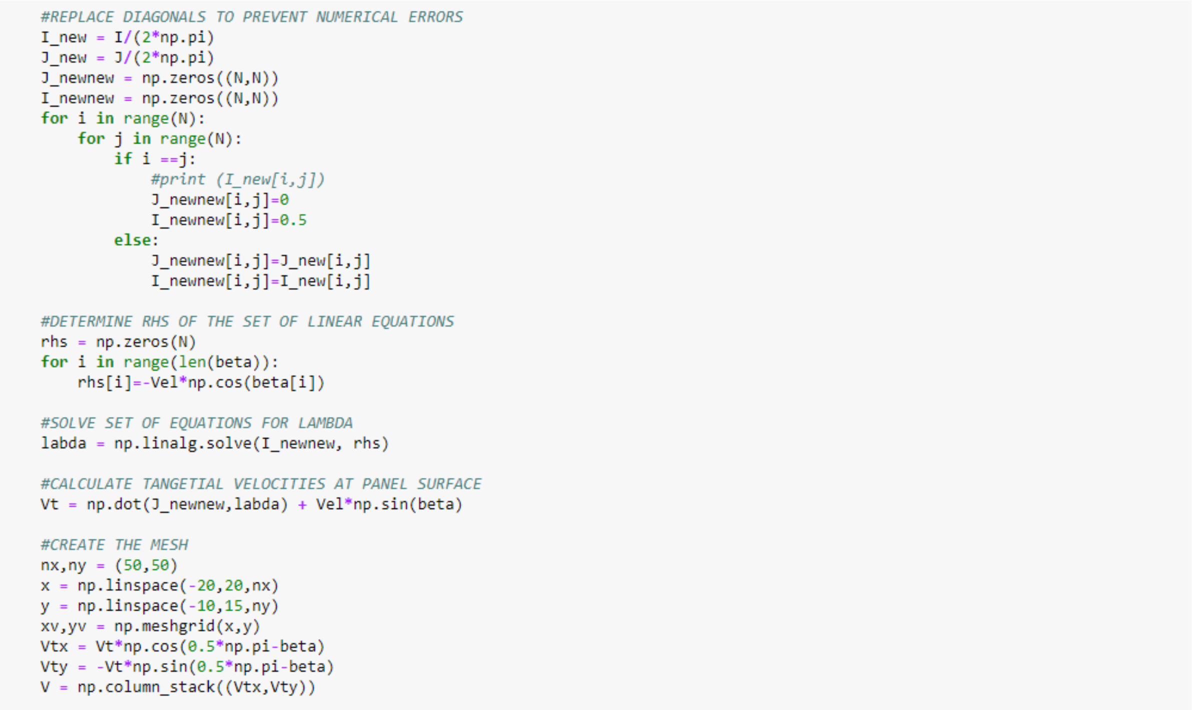
```
    # %% READ AIRFOIL COORDINATES

    # Load the data from the text file
    dataBuffer = np.loadtxt(savecoord, skiprows=0)                               # Read the boundary points from the panels

    # Extract data from the loaded dataBuffer array
    xfoiloutput[4] = dataBuffer[:,0]                                             # boundary point x - panels
    xfoiloutput[5] = dataBuffer[:,1]                                             # boundary point y - panels

    # Delete file after loading
    if os.path.exists(savecoord):                                                # If filename exists
        os.remove(savecoord)                                                     # Delete file to clean storage

    return xfoiloutput                                                           # Return the output
```

Figure F.3: The python script - 3



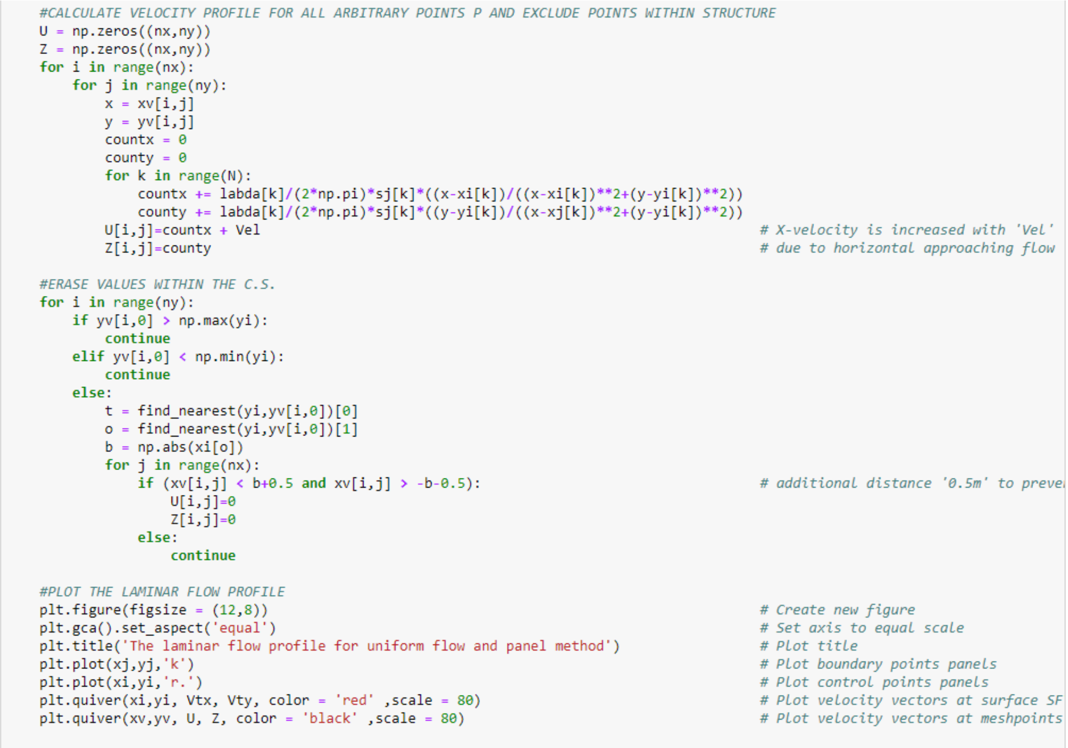
```
: def Flow():
    #IMPORTING DATA FROM GRASSHOPPER
    rho = 1050                                                                    # Fluid density [kg/m^3]
    Vel = 2.00                                                                    # Uniform velocity [m/s]
    beta = np.loadtxt('angle_beta.txt')                                           # Angle freestream and outward normal per panel
    sj = np.loadtxt('sj.txt')                                                     # Panel length
    xi = np.loadtxt('control_points_x.txt')                                       # X-coordinate control point
    yi = np.loadtxt('control_points_y.txt')                                       # Y-coordinate control point
    xj = np.loadtxt('boundary_points_x.txt')                                      # Starting x-coordinate of panel
    yj = np.loadtxt('boundary_points_y.txt')                                      # Starting y-coordinate of panel
    phi_j = np.loadtxt('phi_j.txt')                                               # Angle between positive x-axis and inside pane
    phi_i = np.loadtxt('phi_i.txt')                                               # The same as above (redundant)
    alfa = np.loadtxt('angle_normal.txt')                                         # Angle panel and positive x-axis
    data = np.loadtxt('add_data.txt')
    flag = data[2]                                                                # Setting for exoskeleton on[1]/off[0]

    #FILLING ALL THE MATRICES TO SOLVE FOR LAMBDA BASED ON THE SOURCE PANEL METHOD
    N = len(xj)
    A = np.zeros((N,N))
    B = np.zeros((N,N))
    C = np.zeros((N,N))
    Ct = np.zeros((N,N))
    D = np.zeros((N,N))
    Dt = np.zeros((N,N))
    E = np.zeros((N,N))
    I = np.zeros((N,N))
    J = np.zeros((N,N))
    for i in range(len(xi)):
        for j in range(len(xj)):
            A[i,j]=-(xi[i]-xj[j])*np.cos(phi_j[j])-(yi[i]-yj[j])*np.sin(phi_j[j])
            B[i,j]=(xi[i]-xj[j])**2+(yi[i]-yj[j])**2
            C[i,j]=np.sin(phi_i[i]-phi_j[j])
            D[i,j]=-(xi[i]-xj[j])*np.sin(phi_i[i])+(yi[i]-yj[j])*np.cos(phi_i[i])
            E[i,j]=np.sqrt((B[i,j]-A[i,j]**2))
            I[i,j]=C[i,j]/2*(np.log((sj[j]**2+2*A[i,j]*sj[j]+B[i,j])/B[i,j]))+(D[i,j]-A[i,j]*C[i,j])/E[i,j]*(np.arctan((sj[j]+A[i
            Ct[i,j]=-1*np.cos(phi_i[i]-phi_j[j])
            Dt[i,j]=(xi[i]-xj[j])*np.cos(phi_i[i])+(yi[i]-yj[j])*np.sin(phi_i[i])
            J[i,j]=Ct[i,j]/2*(np.log((sj[j]**2+2*A[i,j]*sj[j]+B[i,j])/B[i,j]))+(Dt[i,j]-A[i,j]*Ct[i,j])/E[i,j]*(np.arctan((sj[j]+
```

Figure F.4: The python script - 4



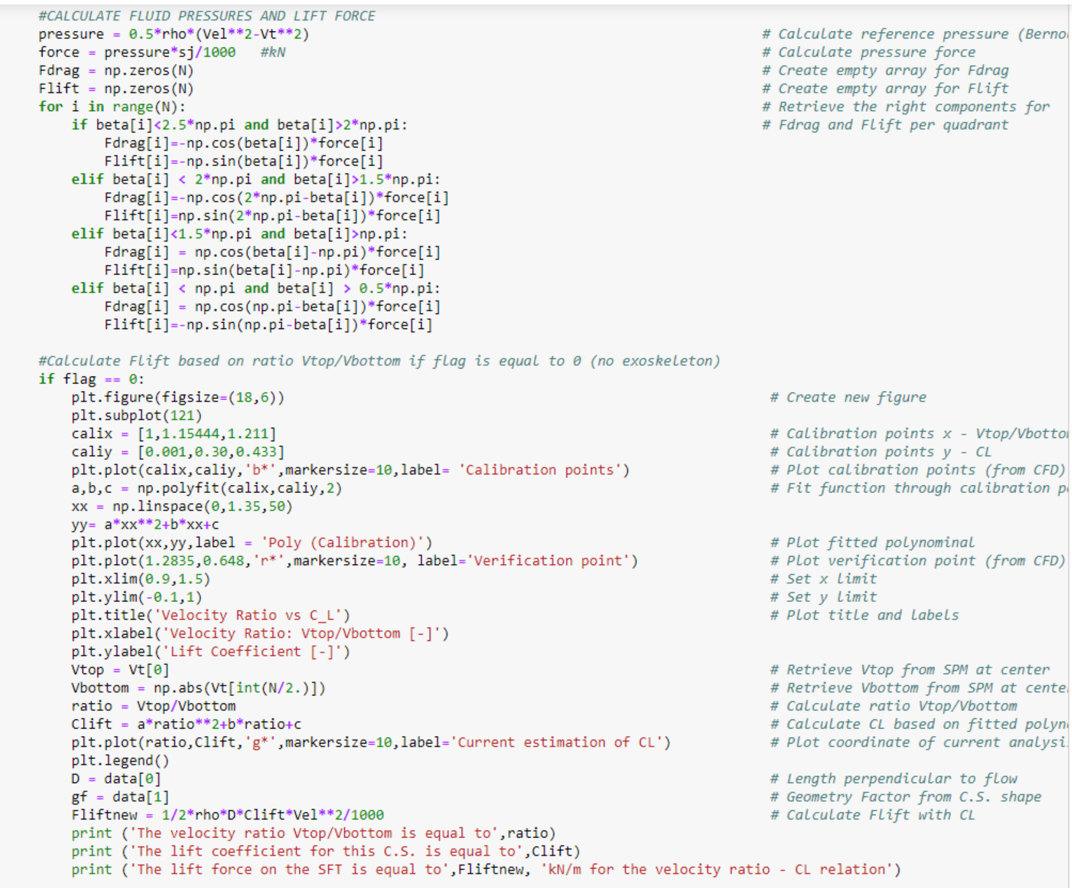
```
    #REPLACE DIAGONALS TO PREVENT NUMERICAL ERRORS
    I_new = I/(2*np.pi)
    J_new = J/(2*np.pi)
    J_newnew = np.zeros((N,N))
    I_newnew = np.zeros((N,N))
    for i in range(N):
        for j in range(N):
            if i ==j:
                #print (I_new[i,j])
                J_newnew[i,j]=0
                I_newnew[i,j]=0.5
            else:
                J_newnew[i,j]=J_new[i,j]
                I_newnew[i,j]=I_new[i,j]

    #DETERMINE RHS OF THE SET OF LINEAR EQUATIONS
    rhs = np.zeros(N)
    for i in range(len(beta)):
        rhs[i]=-Vel*np.cos(beta[i])

    #SOLVE SET OF EQUATIONS FOR LAMBDA
    labda = np.linalg.solve(I_newnew, rhs)

    #CALCULATE TANGETIAL VELOCITIES AT PANEL SURFACE
    Vt = np.dot(J_newnew,labda) + Vel*np.sin(beta)

    #CREATE THE MESH
    nx,ny = (50,50)
    x = np.linspace(-20,20,nx)
    y = np.linspace(-10,15,ny)
    xv,yv = np.meshgrid(x,y)
    Vtx = Vt*np.cos(0.5*np.pi-beta)
    Vty = -Vt*np.sin(0.5*np.pi-beta)
    V = np.column_stack((Vtx,Vty))
```

Figure F.5: The python script - 5



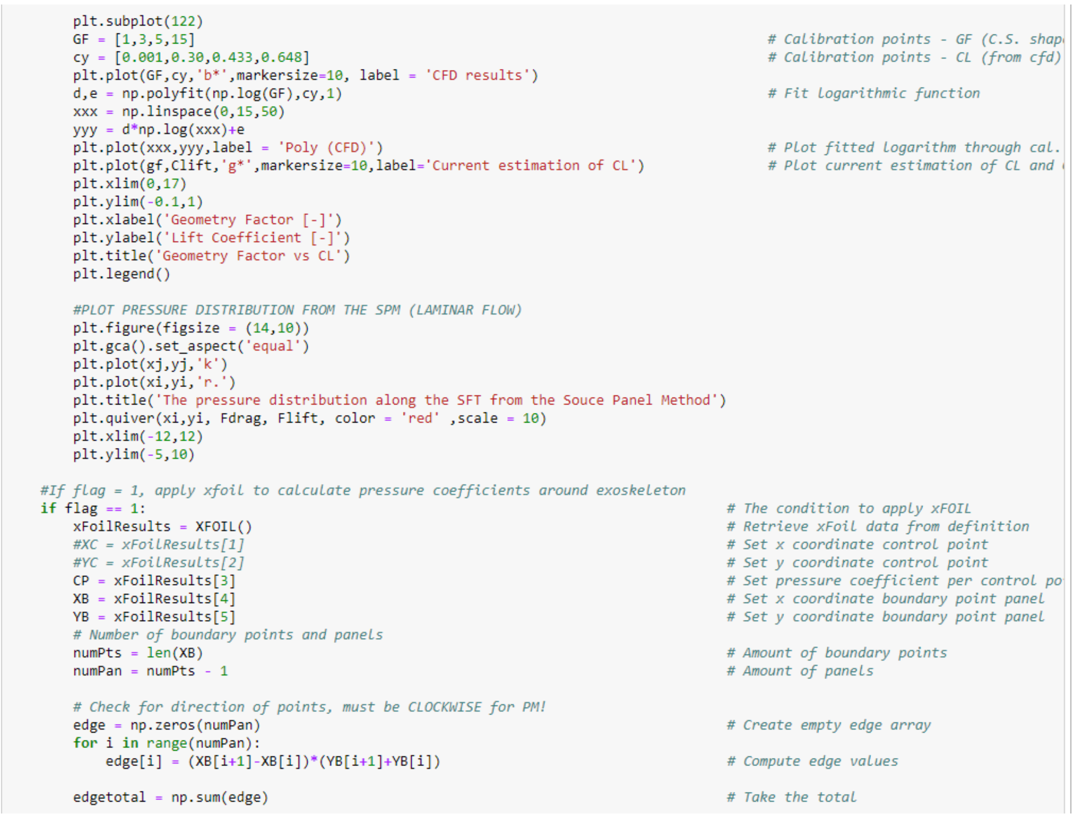
```python
#CALCULATE VELOCITY PROFILE FOR ALL ARBITRARY POINTS P AND EXCLUDE POINTS WITHIN STRUCTURE
U = np.zeros((nx,ny))
Z = np.zeros((nx,ny))
for i in range(nx):
    for j in range(ny):
        x = xv[i,j]
        y = yv[i,j]
        countx = 0
        county = 0
        for k in range(N):
            countx += labda[k]/(2*np.pi)*sj[k]*((x-xi[k])/((x-xi[k])**2+(y-yi[k])**2))
            county += labda[k]/(2*np.pi)*sj[k]*((y-yi[k])/((x-xj[k])**2+(y-yi[k])**2))
        U[i,j]=countx + Vel                                                             # X-velocity is increased with 'Vel'
        Z[i,j]=county                                                                   # due to horizontal approaching flow

#ERASE VALUES WITHIN THE C.S.
for i in range(ny):
    if yv[i,0] > np.max(yi):
        continue
    elif yv[i,0] < np.min(yi):
        continue
    else:
        t = find_nearest(yi,yv[i,0])[0]
        o = find_nearest(yi,yv[i,0])[1]
        b = np.abs(xi[o])
        for j in range(nx):
            if (xv[i,j] < b+0.5 and xv[i,j] > -b-0.5):                                  # additional distance '0.5m' to preve
                U[i,j]=0
                Z[i,j]=0
            else:
                continue

#PLOT THE LAMINAR FLOW PROFILE
plt.figure(figsize = (12,8))                                                            # Create new figure
plt.gca().set_aspect('equal')                                                           # Set axis to equal scale
plt.title('The laminar flow profile for uniform flow and panel method')                 # Plot title
plt.plot(xj,yj,'k')                                                                     # Plot boundary points panels
plt.plot(xi,yi,'r.')                                                                    # Plot control points panels
plt.quiver(xi,yi, Vtx, Vty, color = 'red' ,scale = 80)                                  # Plot velocity vectors at surface SF
plt.quiver(xv,yv, U, Z, color = 'black' ,scale = 80)                                    # Plot velocity vectors at meshpoints
```

Figure F.6: The python script - 6



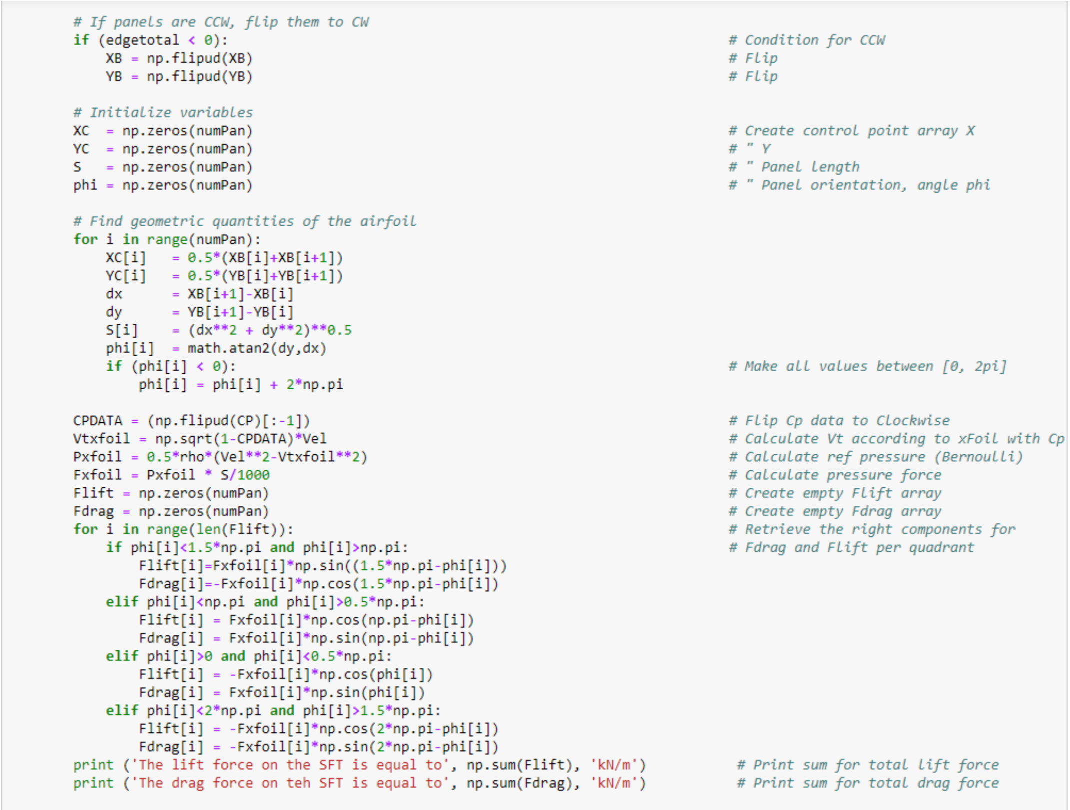
```python
#CALCULATE FLUID PRESSURES AND LIFT FORCE
pressure = 0.5*rho*(Vel**2-Vt**2)                                                       # Calculate reference pressure (Berno
force = pressure*sj/1000   #kN                                                          # Calculate pressure force
Fdrag = np.zeros(N)                                                                     # Create empty array for Fdrag
Flift = np.zeros(N)                                                                     # Create empty array for Flift
for i in range(N):                                                                      # Retrieve the right components for
    if beta[i]<2.5*np.pi and beta[i]>2*np.pi:                                           # Fdrag and Flift per quadrant
        Fdrag[i]=-np.cos(beta[i])*force[i]
        Flift[i]=-np.sin(beta[i])*force[i]
    elif beta[i] < 2*np.pi and beta[i]>1.5*np.pi:
        Fdrag[i]=-np.cos(2*np.pi-beta[i])*force[i]
        Flift[i]=np.sin(2*np.pi-beta[i])*force[i]
    elif beta[i]<1.5*np.pi and beta[i]>np.pi:
        Fdrag[i] = np.cos(beta[i]-np.pi)*force[i]
        Flift[i]=np.sin(beta[i]-np.pi)*force[i]
    elif beta[i] < np.pi and beta[i] > 0.5*np.pi:
        Fdrag[i] = np.cos(np.pi-beta[i])*force[i]
        Flift[i]=-np.sin(np.pi-beta[i])*force[i]

#Calculate Flift based on ratio Vtop/Vbottom if flag is equal to 0 (no exoskeleton)
if flag == 0:
    plt.figure(figsize=(18,6))                                                           # Create new figure
    plt.subplot(121)
    calix = [1,1.15444,1.211]                                                            # Calibration points x - Vtop/Vbotto
    caliy = [0.001,0.30,0.433]                                                           # Calibration points y - CL
    plt.plot(calix,caliy,'b*',markersize=10,label= 'Calibration points')                 # Plot calibration points (from CFD)
    a,b,c = np.polyfit(calix,caliy,2)                                                    # Fit function through calibration p
    xx = np.linspace(0,1.35,50)
    yy= a*xx**2+b*xx+c
    plt.plot(xx,yy,label = 'Poly (Calibration)')                                         # Plot fitted polynominal
    plt.plot(1.2835,0.648,'r*',markersize=10, label='Verification point')                # Plot verification point (from CFD)
    plt.xlim(0.9,1.5)                                                                    # Set x limit
    plt.ylim(-0.1,1)                                                                     # Set y limit
    plt.title('Velocity Ratio vs C_L')                                                   # Plot title and labels
    plt.xlabel('Velocity Ratio: Vtop/Vbottom [-]')
    plt.ylabel('Lift Coefficient [-]')
    Vtop = Vt[0]                                                                         # Retrieve Vtop from SPM at center
    Vbottom = np.abs(Vt[int(N/2.)])                                                      # Retrieve Vbottom from SPM at cente
    ratio = Vtop/Vbottom                                                                 # Calculate ratio Vtop/Vbottom
    Clift = a*ratio**2+b*ratio+c                                                         # Calculate CL based on fitted polyn
    plt.plot(ratio,Clift,'g*',markersize=10,label='Current estimation of CL')            # Plot coordinate of current analysi
    plt.legend()
    D = data[0]                                                                          # Length perpendicular to flow
    gf = data[1]                                                                         # Geometry Factor from C.S. shape
    Fliftnew = 1/2*rho*D*Clift*Vel**2/1000                                               # Calculate Flift with CL
    print ('The velocity ratio Vtop/Vbottom is equal to',ratio)
    print ('The lift coefficient for this C.S. is equal to',Clift)
    print ('The lift force on the SFT is equal to',Fliftnew, 'kN/m for the velocity ratio - CL relation')
```

Figure F.7: The python script - 7



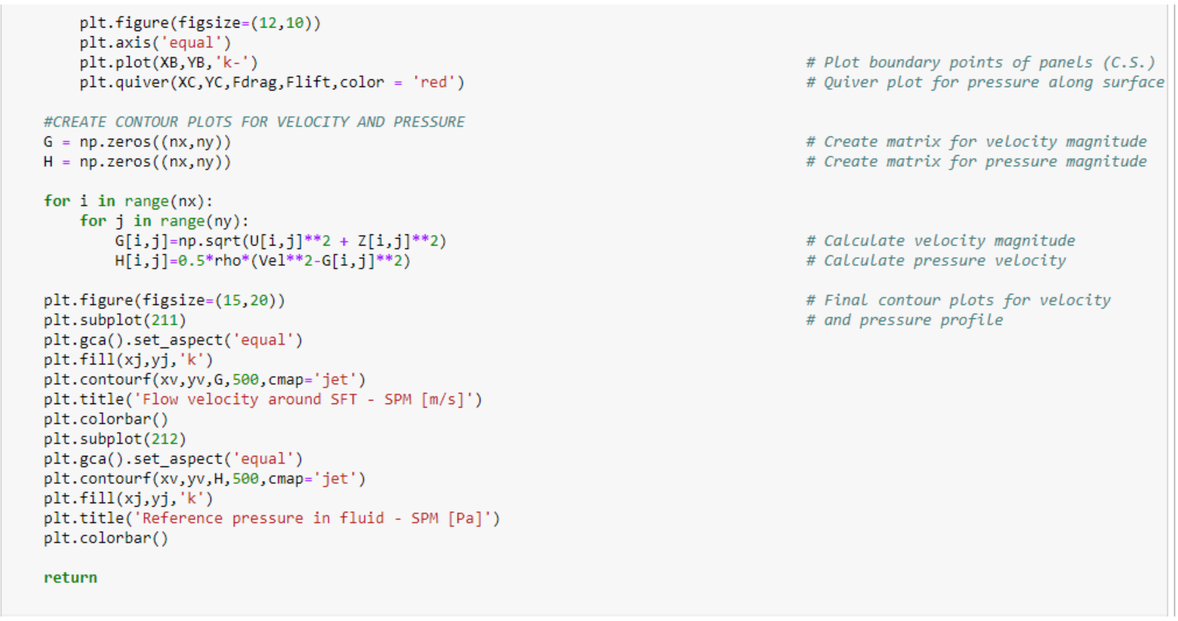
```
    plt.subplot(122)
    GF = [1,3,5,15]                                                                      # Calibration points - GF (C.S. shap
    cy = [0.001,0.30,0.433,0.648]                                                        # Calibration points - CL (from cfd)
    plt.plot(GF,cy,'b*',markersize=10, label = 'CFD results')
    d,e = np.polyfit(np.log(GF),cy,1)                                                    # Fit logarithmic function
    xxx = np.linspace(0,15,50)
    yyy = d*np.log(xxx)+e
    plt.plot(xxx,yyy,label = 'Poly (CFD)')                                               # Plot fitted logarithm through cal.
    plt.plot(gf,Clift,'g*',markersize=10,label='Current estimation of CL')               # Plot current estimation of CL and
    plt.xlim(0,17)
    plt.ylim(-0.1,1)
    plt.xlabel('Geometry Factor [-]')
    plt.ylabel('Lift Coefficient [-]')
    plt.title('Geometry Factor vs CL')
    plt.legend()

    #PLOT PRESSURE DISTRIBUTION FROM THE SPM (LAMINAR FLOW)
    plt.figure(figsize = (14,10))
    plt.gca().set_aspect('equal')
    plt.plot(xj,yj,'k')
    plt.plot(xi,yi,'r.')
    plt.title('The pressure distribution along the SFT from the Souce Panel Method')
    plt.quiver(xi,yi, Fdrag, Flift, color = 'red' ,scale = 10)
    plt.xlim(-12,12)
    plt.ylim(-5,10)

#If flag = 1, apply xfoil to calculate pressure coefficients around exoskeleton
if flag == 1:                                                                       # The condition to apply xFOIL
    xFoilResults = XFOIL()                                                          # Retrieve xFoil data from definition
    #XC = xFoilResults[1]                                                           # Set x coordinate control point
    #YC = xFoilResults[2]                                                           # Set y coordinate control point
    CP = xFoilResults[3]                                                            # Set pressure coefficient per control po
    XB = xFoilResults[4]                                                            # Set x coordinate boundary point panel
    YB = xFoilResults[5]                                                            # Set y coordinate boundary point panel
    # Number of boundary points and panels
    numPts = len(XB)                                                                # Amount of boundary points
    numPan = numPts - 1                                                             # Amount of panels

    # Check for direction of points, must be CLOCKWISE for PM!
    edge = np.zeros(numPan)                                                         # Create empty edge array
    for i in range(numPan):
        edge[i] = (XB[i+1]-XB[i])*(YB[i+1]+YB[i])                                   # Compute edge values

    edgetotal = np.sum(edge)                                                        # Take the total
```

Figure F.8: The python script - 8



```
    # If panels are CCW, flip them to CW
    if (edgetotal < 0):                                                             # Condition for CCW
        XB = np.flipud(XB)                                                          # Flip
        YB = np.flipud(YB)                                                          # Flip

    # Initialize variables
    XC  = np.zeros(numPan)                                                          # Create control point array X
    YC  = np.zeros(numPan)                                                          # " Y
    S   = np.zeros(numPan)                                                          # " Panel length
    phi = np.zeros(numPan)                                                          # " Panel orientation, angle phi

    # Find geometric quantities of the airfoil
    for i in range(numPan):
        XC[i]   = 0.5*(XB[i]+XB[i+1])
        YC[i]   = 0.5*(YB[i]+YB[i+1])
        dx      = XB[i+1]-XB[i]
        dy      = YB[i+1]-YB[i]
        S[i]    = (dx**2 + dy**2)**0.5
        phi[i]  = math.atan2(dy,dx)
        if (phi[i] < 0):                                                            # Make all values between [0, 2pi]
            phi[i] = phi[i] + 2*np.pi

    CPDATA = (np.flipud(CP)[:-1])                                                   # Flip Cp data to Clockwise
    Vtxfoil = np.sqrt(1-CPDATA)*Vel                                                 # Calculate Vt according to xFoil with Cp
    Pxfoil = 0.5*rho*(Vel**2-Vtxfoil**2)                                            # Calculate ref pressure (Bernoulli)
    Fxfoil = Pxfoil * S/1000                                                        # Calculate pressure force
    Flift = np.zeros(numPan)                                                        # Create empty Flift array
    Fdrag = np.zeros(numPan)                                                        # Create empty Fdrag array
    for i in range(len(Flift)):                                                     # Retrieve the right components for
        if phi[i]<1.5*np.pi and phi[i]>np.pi:                                       # Fdrag and Flift per quadrant
            Flift[i]=Fxfoil[i]*np.sin((1.5*np.pi-phi[i]))
            Fdrag[i]=-Fxfoil[i]*np.cos(1.5*np.pi-phi[i])
        elif phi[i]<np.pi and phi[i]>0.5*np.pi:
            Flift[i] = Fxfoil[i]*np.cos(np.pi-phi[i])
            Fdrag[i] = Fxfoil[i]*np.sin(np.pi-phi[i])
        elif phi[i]>0 and phi[i]<0.5*np.pi:
            Flift[i] = -Fxfoil[i]*np.cos(phi[i])
            Fdrag[i] = Fxfoil[i]*np.sin(phi[i])
        elif phi[i]<2*np.pi and phi[i]>1.5*np.pi:
            Flift[i] = -Fxfoil[i]*np.cos(2*np.pi-phi[i])
            Fdrag[i] = -Fxfoil[i]*np.sin(2*np.pi-phi[i])
    print ('The lift force on the SFT is equal to', np.sum(Flift), 'kN/m')           # Print sum for total lift force
    print ('The drag force on teh SFT is equal to', np.sum(Fdrag), 'kN/m')           # Print sum for total drag force
```

Figure F.9: The python script - 9



```
        plt.figure(figsize=(12,10))
        plt.axis('equal')
        plt.plot(XB,YB,'k-')                                               # Plot boundary points of panels (C.S.)
        plt.quiver(XC,YC,Fdrag,Flift,color = 'red')                        # Quiver plot for pressure along surface

    #CREATE CONTOUR PLOTS FOR VELOCITY AND PRESSURE
    G = np.zeros((nx,ny))                                                  # Create matrix for velocity magnitude
    H = np.zeros((nx,ny))                                                  # Create matrix for pressure magnitude

    for i in range(nx):
        for j in range(ny):
            G[i,j]=np.sqrt(U[i,j]**2 + Z[i,j]**2)                          # Calculate velocity magnitude
            H[i,j]=0.5*rho*(Vel**2-G[i,j]**2)                              # Calculate pressure velocity

    plt.figure(figsize=(15,20))                                            # Final contour plots for velocity
    plt.subplot(211)                                                       # and pressure profile
    plt.gca().set_aspect('equal')
    plt.fill(xj,yj,'k')
    plt.contourf(xv,yv,G,500,cmap='jet')
    plt.title('Flow velocity around SFT - SPM [m/s]')
    plt.colorbar()
    plt.subplot(212)
    plt.gca().set_aspect('equal')
    plt.contourf(xv,yv,H,500,cmap='jet')
    plt.fill(xj,yj,'k')
    plt.title('Reference pressure in fluid - SPM [Pa]')
    plt.colorbar()

    return
```

Figure F.10: The python script - 10

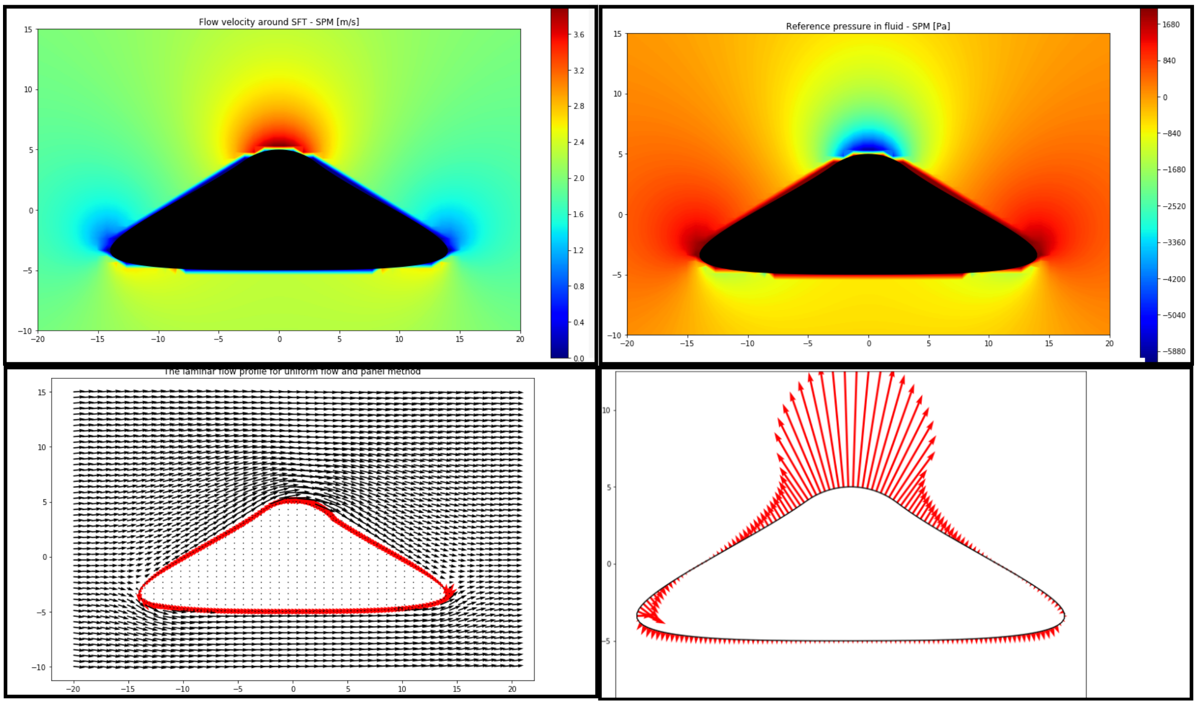


Figure F.11: Output of Python script - with exoskeleton and xFoil analysis

G

CFD-Results

In section 4.5, the calculation of the lift force is described. Two methods are presented there. The lift forces are evaluated for the different cross-sections as found and described by section 4.4.
The first option is to simulate a CFD-model, which is accurate but very time consuming (BRON). Often, at early stages within the design process, a quick estimation of F_{Lift} is desired instead of a detailed analysis with a complicated model. The second option is to apply the Source Panel Method (SPM). The Panel Method has been elaborated in Appendix E and coded in Appendix F. This method is computationally very efficient and gives accurate results **for the leading edge**. The trailing edge is where turbulent phenomena occur which cannot be modelled by the SPM. This method only handles laminar flow, which results in zero drag and lift for any shape. This is not realistic, so a compromise between the two options is searched for.

The velocity at the top and bottom of the SFT, V_{Top} and V_{Bottom} are found to be within an acceptable error margin from the CFD results and therefore assumed to be realistic. Therefore, these values are used to find a relation between the shape and the lift force/coefficient. The relation between the flow velocity, or the flow velocity ratio, and the lift coefficient is assumed to be quadratic according to:

$$C_L = \frac{F_{Lift}}{\frac{1}{2} \cdot \rho \cdot D \cdot V^2}$$

To find this relation, three 'equations' or points on the curve are required to fit a curve through. These three 'calibration points' are first shown, after which a verification point is shown as well. The outputs from these simulations is further elaborated in section 4.5.

G.1. Calibration point 1 - Circle

The first point follows from simple theory. A shape which is symmetric over the horizontal, like a circle, has zero lift and a ratio $\frac{V_{Top}}{V_{Bottom}}$ equal to 1.0. The flow velocities solved by the SPM for a simple circle in an uniform flow equal to 2 m/s is shown in figure G.1.

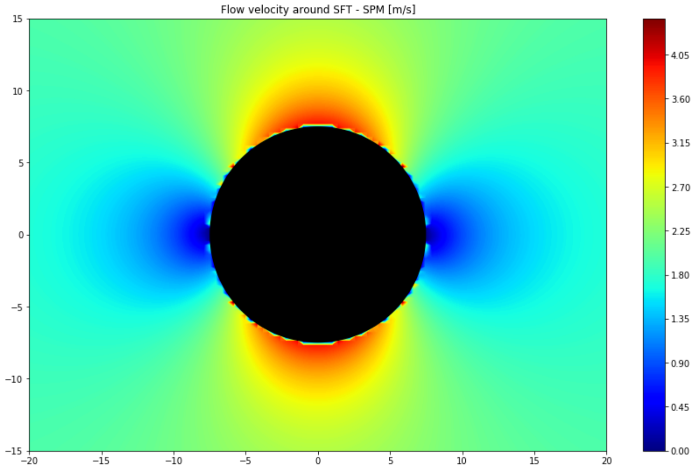


Figure G.1: The velocity profile for a circle in $V_\infty = 2\ m/s$ based on SPM

As turns out from this SPM, the flow velocities at the top and bottom are equal to 4 m/s. Note that this is in line with theory as presented in figure E.1 already. The first point is equal to:

- Ratio = $\frac{V_{Top}}{V_{Bottom}} = \frac{4.0}{4.0} = 1.0$
- $C_L = 0.0$ (Theory)

G.2. Calibration point 2 - GF = 3.0

Here, the cross-section with geometry factor 3.0 is evaluated with the CFD-model. For the definition of this GF, one is referred to section 4.4. The cross-section looks as below in figure G.2:

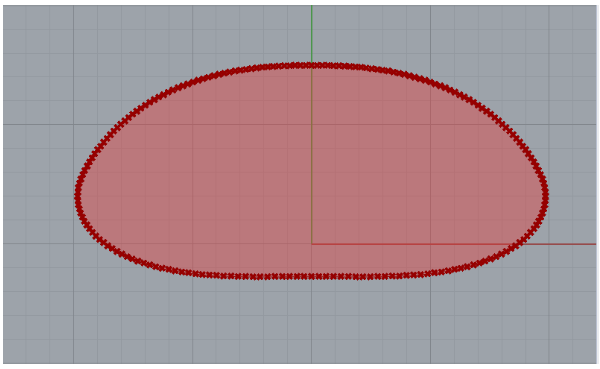


Figure G.2: The cross-section with GF = 3.0

The ratio $\frac{V_{Top}}{V_{Bottom}}$ is again determined based on the Panel Method. This ratio is the same as for the CFD-output or within an acceptable error margin.

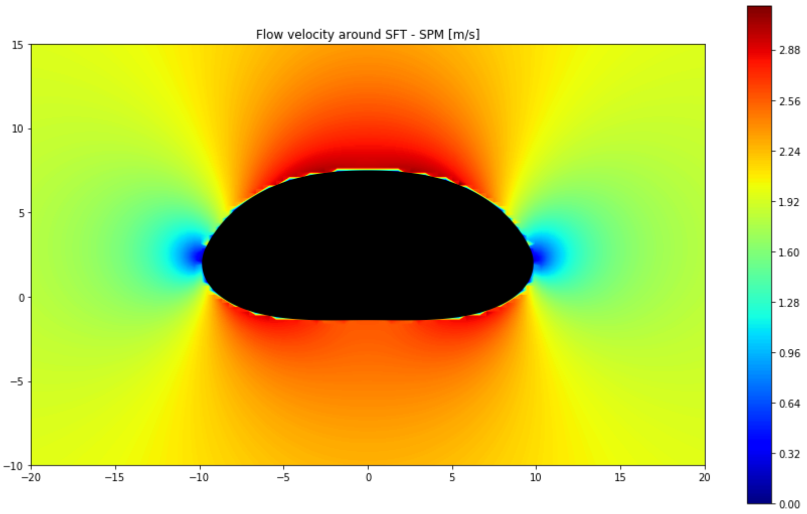


Figure G.3: The flow velocity profile for cross-section with GF = 3.0

The output from the SPM, more accurately retrieved from the Python script itself, is:

- $V_{Top} = 2.99\ m/s$
- $V_{Bottom} = 2.59\ m/s$

The ratio between the flow velocities is known. The lift force and lift coefficient can be determined based on the CFD. As mentioned before, the SPM is unable to calculated these forces. Therefore, the CFD-simulation is required. For the output of the CFD, see the figures below.

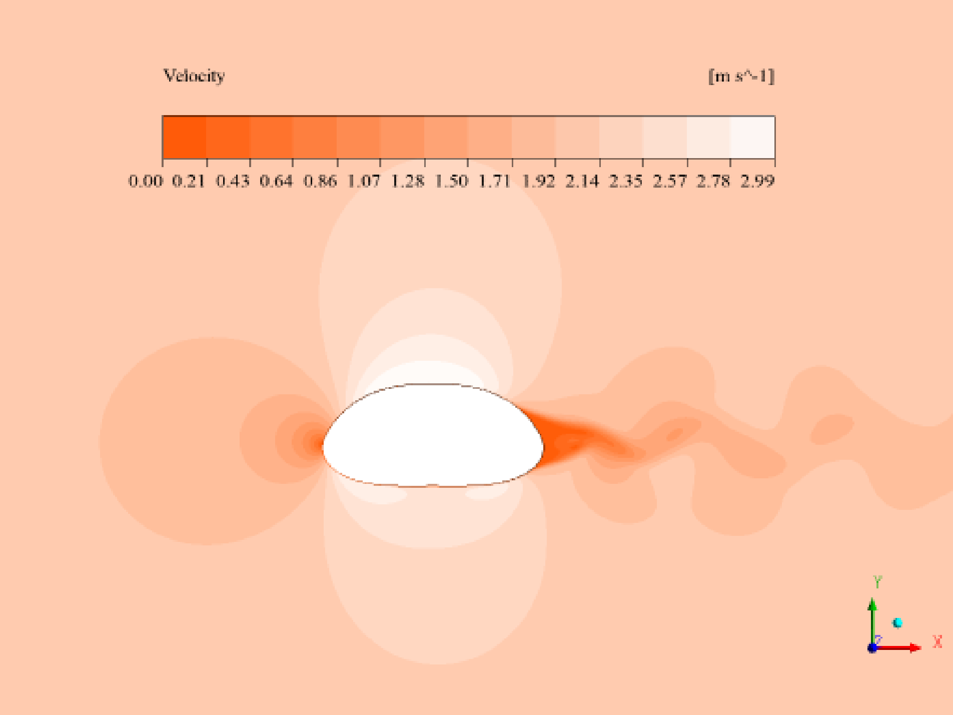


Figure G.4: The velocity field for cross-section with GF = 3.0 and $V_{\infty} = 2\ m/s$ [43]

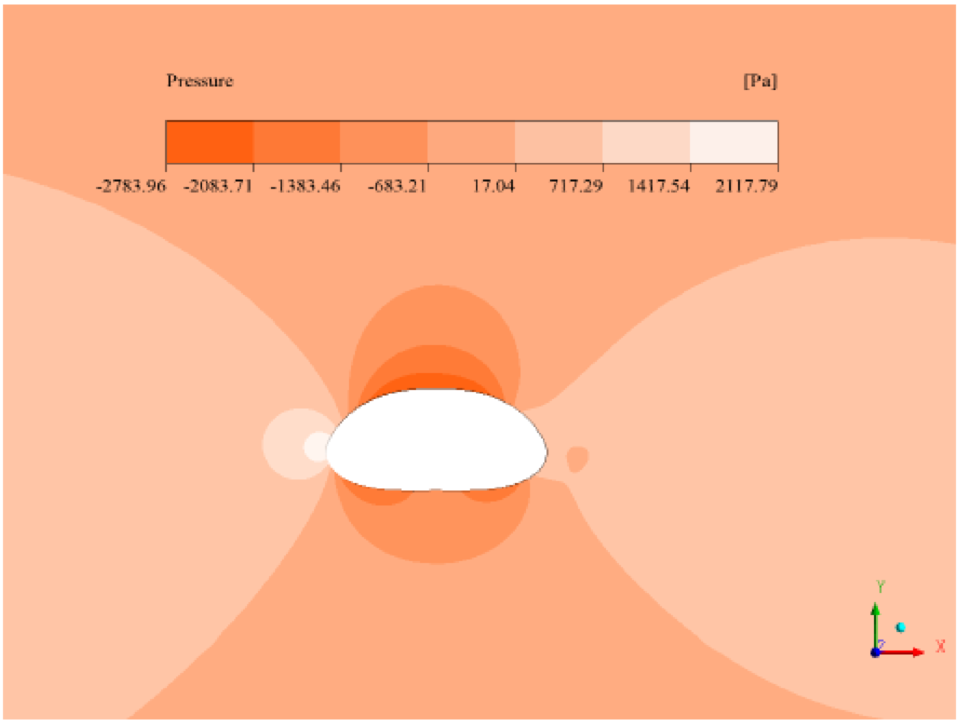


Figure G.5: The pressure field for cross-section with GF = 3.0 and $V_\infty = 2\ m/s$ [43]

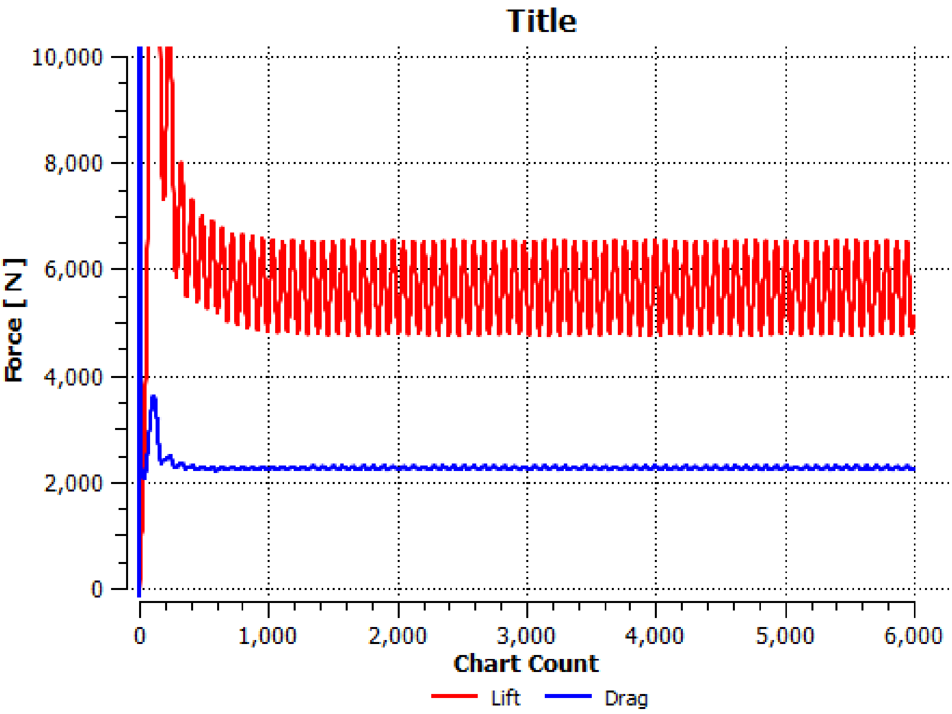


Figure G.6: The drag and lift force for cross-section with GF = 3.0 and $V_\infty = 2\ m/s$ [43]

The lift coefficient can now be calculated:

$$C_L = \frac{F_{Lift}}{\frac{1}{2}\cdot\rho\cdot D\cdot V_\infty^2} = \frac{5.600}{\frac{1}{2}\cdot 1050\cdot 8.9\cdot 2^2} = 0.30$$

Which leads to the second calibration point to find the relation:

- Ratio = $\frac{V_{Top}}{V_{Bottom}} = \frac{2.99}{2.59} = 1.15$
- $C_L = 0.30$

G.3. Calibration point 3 - GF = 5.0

This CFD output of this calibration point was already shown in the main report. For the shape of the cross-section, one is referred to figure 4.6b. In paragraph 4.5.1 the CFD output is shown in figures 4.8, 4.9 and 4.10. The velocity field as calculated by the SPM in Python is described in paragraph 4.5.2 and shown in figure 4.12. This all led to the following calibration point:

- Ratio = $\frac{V_{Top}}{V_{Bottom}} = \frac{3.10}{2.56} = 1.21$
- $C_L = 0.43$

G.4. Verification point - GF = 15.0

With the three points above, enough information is available to fit a quadratic curve which shows the relation between $\frac{V_{Top}}{V_{Bottom}}$ and C_L. To see if this relation is actually valid, an additional situation is required to verify. This is done for a Geometry Factor equal to 15. The shape looks as in figure G.7

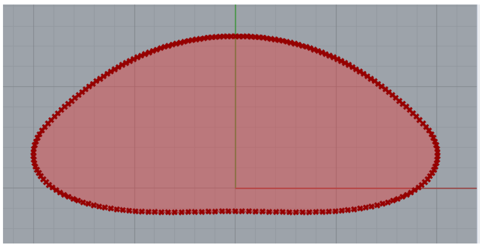


Figure G.7: The cross-section for Geometry Factor = 15.0

The ratio $\frac{V_{Top}}{V_{Bottom}}$ is again determined based on the Panel Method. This ratio is practically the same as for the CFD-output.

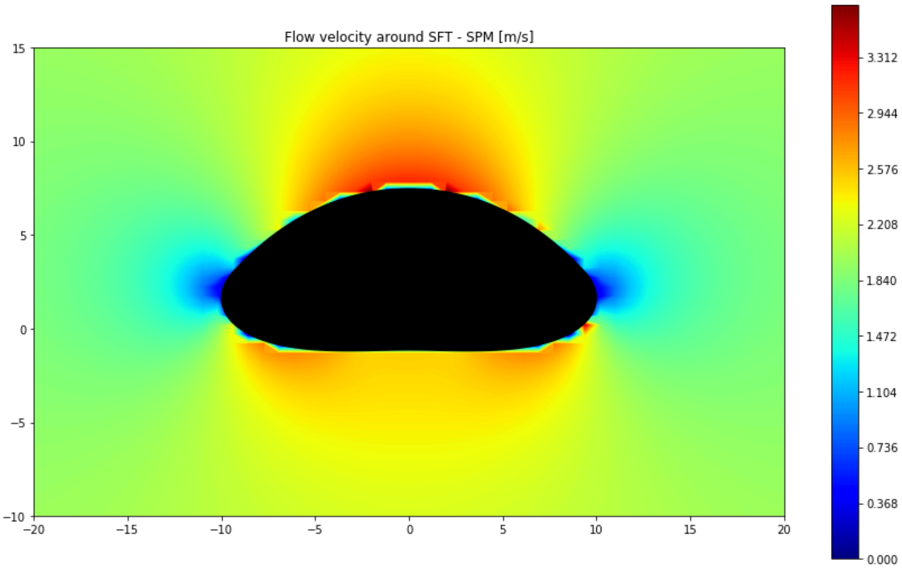


Figure G.8: The flow velocity profile for cross-section with GF = 15.0

The output from the SPM, more accurately retrieved from the Python script itself, is:

- $V_{Top} = 3.26\ m/s$

- $V_{Bottom} = 2.54\ m/s$

The ratio between the flow velocities is known. The lift force and lift coefficient can be determined based on the CFD results. As mentioned before, the SPM is unable to calculated these forces. Therefore, the CFD-simulation is required. For the output of the CFD, see the figures below.

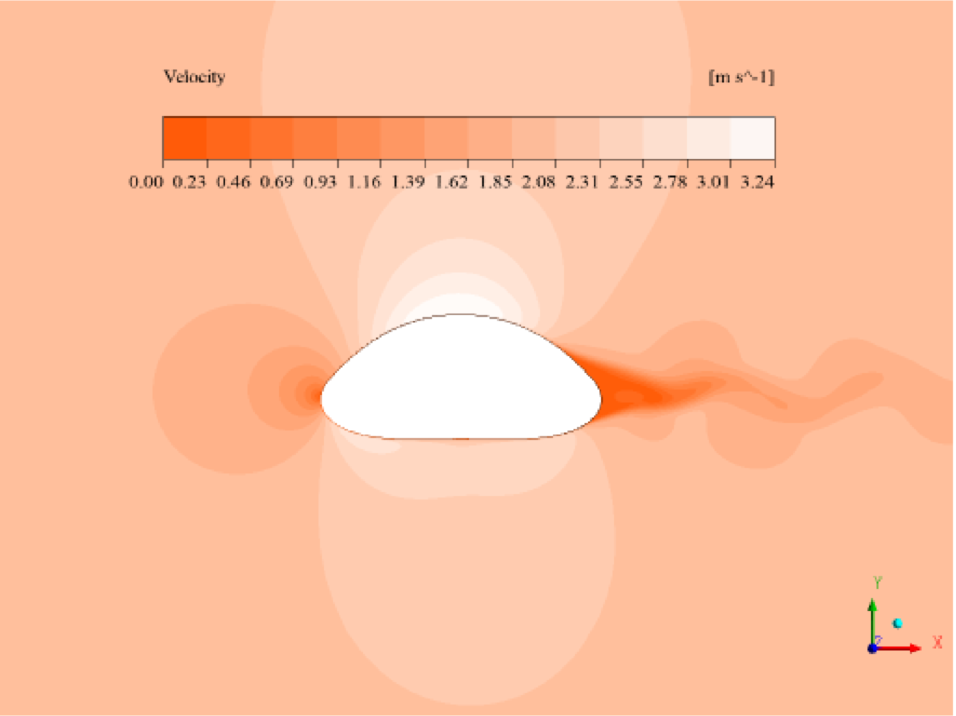


Figure G.9: The velocity field for cross-section with GF = 15.0 and $V_\infty = 2\ m/s$ [43]

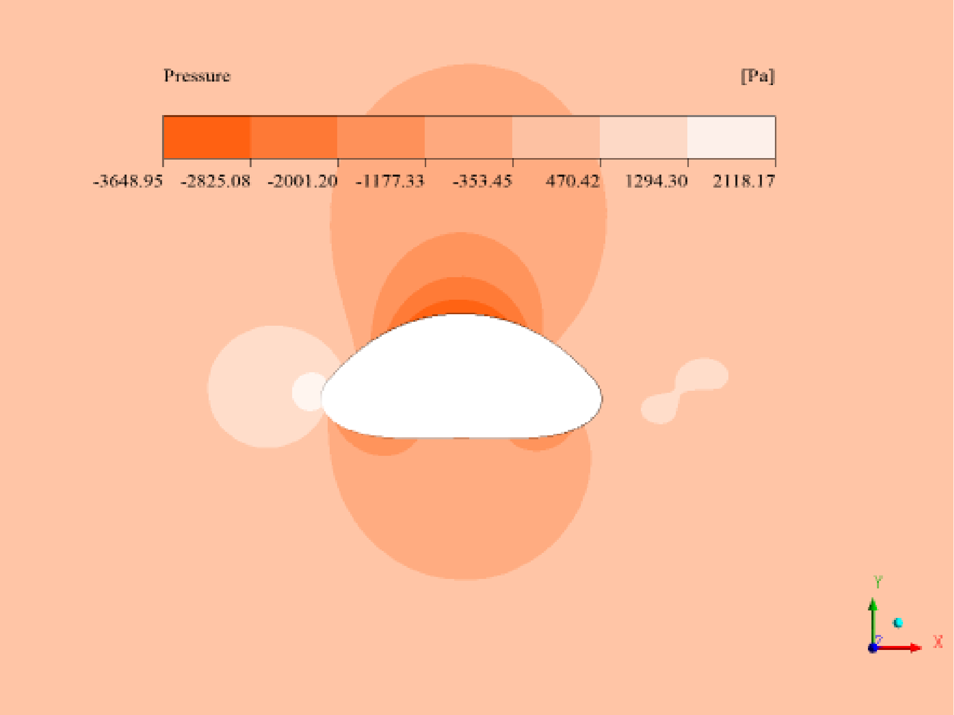


Figure G.10: The pressure field for cross-section with GF = 15.0 and $V_\infty = 2\ m/s$ [43]

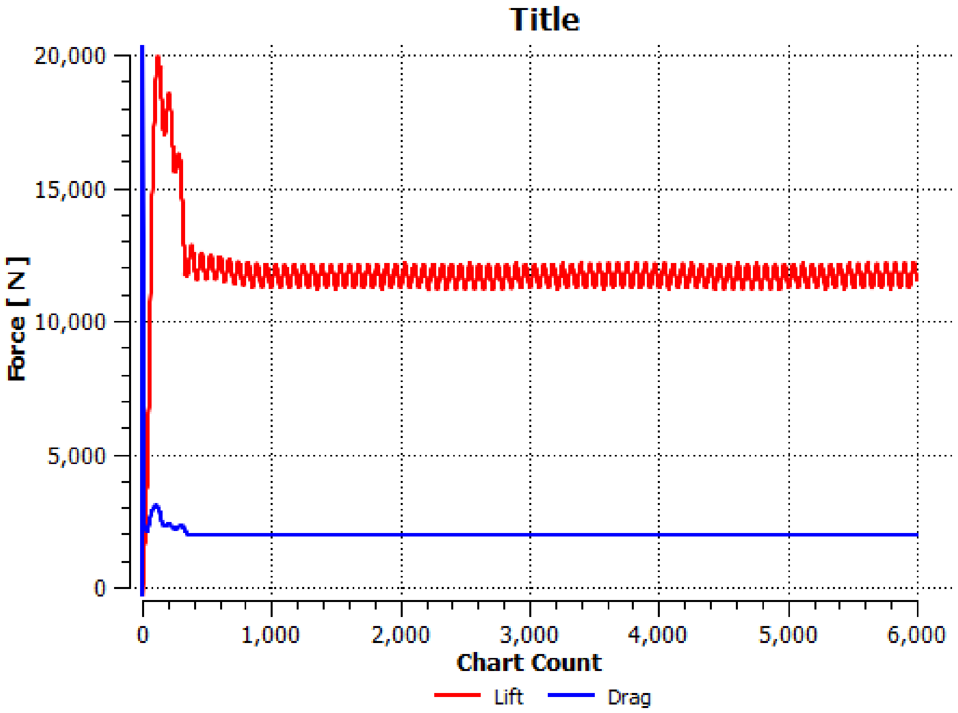


Figure G.11: The drag and lift force for cross-section with GF = 15.0 and $V_\infty = 2\ m/s$[43]

The lift coefficient can now be calculated:

$$C_L = \frac{F_{Lift}}{\frac{1}{2}\cdot\rho\cdot D\cdot V_\infty^2} = \frac{11.700}{\frac{1}{2}\cdot 1050\cdot 8.6\cdot 2^2} = 0.648$$

Which leads to the verification point to establish and check the relation:

- Ratio = $\frac{V_{Top}}{V_{Bottom}} = \frac{3.26}{2.54} = 1.28$
- $C_L = 0.648$

Again, the further processing of all this output to a relation between the ratio of the flow velocities and a lift coefficient is further executed in paragraph 4.5.3.