




University of
Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

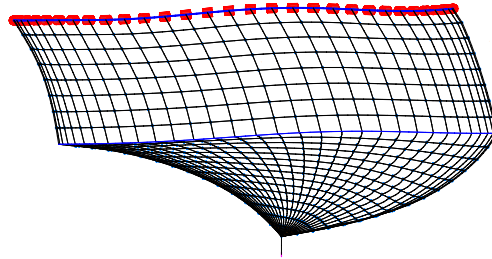
MASTER'S THESIS

Study programme/specialisation: Marine and Offshore Technology	Spring semester, 2019 Open
Author: Mathias Bruset	 (signature of author)
Programme coordinator: Prof. Muk Chen Ong Supervisor(s): Prof. Muk Chen Ong Dr. Lin Li	
Title of master's thesis: Dynamic Analysis of a Floating Fish Cage with Feeding Systems	
Credits: 30	
Keywords: - Aquaculture - Dynamic analysis - Numerical modeling - Environmental loading - Hydrodynamics - OrcaFlex API - Python	Number of pages: 108 + supplemental material/other: 23 Stavanger, 29.06.2019



University of
Stavanger

Dynamic Analysis of a Floating Fish Cage with Feeding Systems



Author: Mathias Fredrik Bruset

Supervisor: Prof. Muk Chen Ong

Co-supervisor: Dr. Lin Li

Master's Thesis

Department of Mechanical and Structural Engineering and Materials Science

University of Stavanger

Abstract

Aquaculture is one of the fastest growing industries in Norway, and one of the fastest growing food production industries in the world. Limited access to available locations along the coast is gradually forcing aquacultural operations into exposed locations offshore. It is of interest to explore loading effects and the behaviors of fish farms exposed to considerable hydrodynamic loads. Furthermore, there has been limited analysis of feeding systems and load characteristics. Understanding these characteristics will be beneficial for the industry in learning how to establish and maintain sustainable long-term operations in harsh environments.

A numerical model of a fish farm is developed using a combination of software programs. The numerical model is comprised of a feed barge, a mooring system, and a feeding tube. The feed barge model is established using GeniE, while response amplitude operators and damping coefficients are determined using Wadam. The fish cage, feeding tubes, and mooring lines are created using the programming language Python in conjuncture with the OrcaFlex Application Programming Interface (API). The fish cage net equivalence is determined, and dynamic analyses of the fish farm are subsequently presented. Three fish cage models of varying mesh density are developed, and a convergence study is conducted for different current velocities to compare volume reduction and mooring line tension. Time domain simulations for the numerical model of the fish farm have been conducted for different environmental parameters, solidity ratios, and feeding tube lengths. The focus of the thesis is on the response of mooring lines and feeding tubes, as they represent the loading and behavior of the system for various environmental conditions.

The findings suggest that for operational environmental conditions, oscillations in the feeding tube may induce snap loads. For extreme conditions, the mooring system tensions increased considerably. Increasing the fish cage solidity ratio contributes to increased drag force, hence also increasing mooring line tension, and fish cage volume deformation. Furthermore, increasing feeding tube length resulted in a decrease in tension for the feeding tube in extreme conditions due to improved elastic characteristics. Similar trends are seen for the bending moment, but for extreme conditions, bend stiffeners can be used to reduce bending moment at connection points. In this study, the findings suggest that operational and extreme environmental conditions cause significant tension in the feeding tube that may cause snap loads and ruptures.

Acknowledgments

This thesis was completed in the spring of 2019 as the final part of the Master of Science degree in Marine and Offshore Technology at the University of Stavanger (UiS).

I am grateful for the support from my supervisor, Professor Muk Chen Ong, for his encouragement, enthusiasm, and advice throughout my time as a graduate student at the University.

To Dr. Lin Li, I would like to extend my sincere gratitude for her words of encouragement, mentoring, and valuable discussions in the past months.

Furthermore, I would like to thank Dr. Xiaopeng Wu for all his invaluable support in the numerical modeling process of the thesis.

I would also like to thank my fellow co-students and friends for a collaborative atmosphere, their words of encouragement and motivation these past two years. Lastly, I would like to express my gratitude to the University of Stavanger and the Department of Mechanical and Structural Engineering and Materials Science for providing the necessary resources to complete the work. Working on this project has been an excellent opportunity to get acquainted with the aquaculture industry.

Stavanger, June 2019

Mathias Bruset



Contents

1	Introduction	1
1.1	Background and Motivation	1
1.2	Technological Development and Current Trends	3
1.3	Scope and Thesis Outline	6
2	Theory	8
2.1	Potential Flow Theory	8
2.1.1	Linear Wave Theory	9
2.2	Response of a Vessel in Irregular Waves	10
2.2.1	Irregular Waves	10
2.2.2	Response Amplitude Operators	13
2.2.3	Damping Effects and Added Mass	14
2.3	Loads on Fish Farm System	15
2.3.1	Morison’s Equation	16
2.3.2	Hydrodynamics of a Fish Cage	19
2.3.3	Hydrodynamic Loading of the Feeding Tube	21
3	Numerical Methods and Preliminary Analysis	25
3.1	Numerical Tools - Wadam and OrcaFlex	25
3.2	Hydrodynamic Analysis of Feed Barge	25
3.2.1	Frequency Domain Analysis of Feed Barge	25
3.2.2	Sensitivity Study of Viscous Damping Effects	30
3.3	Preliminary Analysis of the Fish Farm System	34
4	Methodology and Modeling Technique	41
4.1	Numerical Modeling of the Fish Farm System	41
4.1.1	Numerical Modeling Using OrcaFlex API and Python	42
4.1.2	Net Equivalence	50
4.1.3	Mooring System Configuration for Numerical Fish Cage Model	56
4.1.4	Modeling of the Feeding Tube	58
4.2	Convergence Study of Fish Cage Models	60
5	Results and Discussions	66
5.1	Time Domain Simulation Setup	66
5.1.1	Environmental Parameters	66
5.1.2	Time Domain Simulations Setup	68
5.1.3	Simulation Matrix	69

5.2	Time Domain Simulations and Results	71
5.2.1	Responses Under Operational and Extreme Conditions	71
5.2.2	Responses Under Different Wave and Current Direction	78
5.2.3	Sensitivity Study on Solidity Ratio	91
5.2.4	Sensitivity Study on Feeding Tube Configurations	95
6	Conclusions and Future Work	103
6.1	Conclusions	103
6.2	Future Work	105
A	Appendix	109
A.1	OrcaFlex API Codes - Numerical Model of 64 Section Cage	109
A.2	Simulation Script	131

List of Figures

1.1	Production levels of Atlantic salmon from 1980 until 2015 [2].	1
1.2	Locations of aquaculture facilities along the Norwegian coastline according to the Directorate of Fisheries from March 2019 [4].	2
1.3	Conventional fish farm with support systems [5].	3
1.4	Project Havfarm is a vessel shaped fish farm developed for exposed environmental conditions [6].	4
1.5	Ocean Farm 1 operated by SalMar outside the island of Frøya [7].	5
1.6	Submerged fish cage concept developed by Atlantis Subsea Farming AS with water based feeding from nearby feed barge [8].	6
2.1	Linear waves (top four subplots) superimposed to form an irregular wave (bottom subplot) [9].	11
2.2	Comparison between the Pierson-Moskowitz and JONSWAP spectra [11].	12
2.3	Response amplitude operators for floating vessels described in six degrees of freedom as shown by Calqlata [12].	13
2.4	Viscous roll damping effects for a feed barge occurring as a result of vortices forming because of roll motion [14].	15
2.5	Conventional fish farm system including feed barge, feeding tube, fish cage and mooring lines.	16
2.6	Similar to a twine in a fish cage, an inclined slender cylinder indicating normal and tangential force (left) and lift force for the cylinder (right) based on Morison theory.	20
2.7	The net screen model as introduced by Kristiansen and Faltinsen [10].	21
2.8	Feeding tubes from a feeding barge going to multiple fish cages at a Scottish aquaculture facility [18].	22
2.9	High-frequency limit of vertical added mass coefficient as a function of water depth [20].	24
3.1	The WaveMaster AC 850 feed barge, side-view(a) and top view (b), which is used for fish farm operations [14].	27
3.2	Panel model of the feed barge, Wavemaster AC 850, created in GeniE with 0.5 m mesh visible on the exterior panels.	27

3.3	Response amplitude operators of the feed barge without viscous damping effects in (a) heave, (b) pitch, and (c) roll for a range of directions varying from 0° to 180° , with 15° degree intervals.	29
3.4	Roll response amplitude operators of the feed barge for six different viscous damping coefficients.	31
3.5	Roll response comparison of feed barge, (a) without any damping effects and (b) with viscous damping coefficients, $B_{44} = B_{55} = 5.0 \cdot 10^5 \frac{kgm^2}{s}$	33
3.6	Displacement of fish farm system with mooring lines, barge and feeding tube for current velocity $U_c = 1.50 \frac{m}{s}$, with a wave and current direction of 0° . . .	35
3.7	Effective tensions in anchor lines 9, 12, 15, 6 and feeding tube. Current velocity $U_c = 0.50 \frac{m}{s}$ and a wave and current direction of 0° for condition 1. .	37
3.8	Effective tensions in anchor lines 9, 12, 15, 6 and feeding tube. Current velocity $U_c = 1.00 \frac{m}{s}$ and a wave and current direction of 0° for condition 2. .	38
3.9	Effective tensions in anchor lines 9, 12, 15, 6 and feeding tube. Current velocity $U_c = 1.50 \frac{m}{s}$ and a wave and current direction of 0° for condition 3. .	39
4.1	A cylindrical fish cage with a conical bottom section, manufactured by Egersund Net AS, here seen undergoing steady current tests [24].	41
4.2	Elements added using the OrcaFlex API and their connections for a fish cage.	42
4.3	The angle between two buoys (red) on the top, connecting the floating collars, from the center point of the net is used by Python to iteratively connect on section (a) to multiple sections (b), constructing a circular fish cage by generating multiple sections in a circular fashion.	46
4.4	One section for a fish cage model intended to be iterated in a circle using the reiterative code to establish a cylindrical numerical fish cage model.	47
4.5	Net panel sections built iteratively and connected to form a net mesh, is used to build a numerical cylindrical fish cage model with a conical bottom section.	48
4.6	Side view of the numerical fish cage model with a coarse net mesh quality, with 32 sections and 4 meters spacing between knots.	49
4.7	Side view of the numerical fish cage model with a medium net mesh quality, with 48 sections and 3 meters spacing between knots.	49
4.8	Side view of the numerical fish cage model with a fine net mesh quality, with 64 sections and 2 meters spacing between knots.	49

4.9	An overview of the numerical fish cage models with 32 section (a), 48 sections (b), and 64 sections (c).	50
4.10	Net equivalence (red) of net twines (black) representing the net of the numerical model compared to a full-scale fish cage. A few red twines is laid to intersect many black lines.	52
4.11	A conventional mooring system shown for a fish farm with multiple feed lines and fish cages showing feed vessel and mooring lines at the facility [25]. . . .	56
4.12	Schematic overview of fish cage model with complete mooring system consisting of anchor lines, frame lines, bridles, and fish cage.	57
4.13	Fish cage mooring line and anchor system seen from OrcaFlex. The fish cage model can be seen with anchor lines connected to the seabed from top view (a) and side view (b)	58
4.14	Side view of fish cage models showing the cage deformation with current flowing from left to right. The models are a coarse mesh, 32 section fish cage with 4 meters spacing (left), a medium mesh, 48 section fish cage with 3 meters spacing (middle), and a fine mesh, 64 section fish cage with 2 meters spacing (right).	61
4.15	Top view of fish cage models showing the cage deformation with current floating from left to right. The models are a coarse mesh, 32 section fish cage (left), a medium mesh 48 section fish cage (middle), and a fine mesh, 64 section fish cage (right).	62
4.16	Bar plot of the relative volume deformation of the coarse (dark grey), medium (light gray) and fine (white) fish cage models in steady current conditions.	65
5.1	The location of the Mowi AS fish farm facility at Grøttinsøy (arrow) along the coast of mid-Norway, which has been selected as a basis for the environmental parameters in this study [4].	67
5.2	Top view of fish farm assembly with feed barge, feeding tube, fish cage, and mooring lines and buoys.	68
5.3	Anchor line tensions for various steady current velocities for a coarse, 32 section fish cage, with net solidity $S_n = 0.20$	69
5.4	Fish cage system in the operational environmental conditions with feed barge, feeding tube, fish cage and mooring lines, with wave and current direction of 0°	72
5.5	Maximum effective tensions in anchor lines for fish cage during the operational and extreme environmental conditions with a wave and current direction of 0°	73

5.6	Fish farm system in operational environmental conditions, showing effective tension in anchor lines 6, 8 and feeding tube.	74
5.7	Fish farm system in extreme environmental conditions, showing effective tension in anchor lines 6, 8 and feeding tube.	75
5.8	Deformation of the fish cage in extreme conditions, floater deformation is non-circular, indicating tension in the feeding tube.	78
5.9	Fish cage system with varying wave and current direction set to 90°.	79
5.10	Maximum effective tensions in anchor lines during operational (1-year return period) and extreme (50-year return period) conditions for wave and current direction at 90°.	80
5.11	Effective tensions in anchor lines 7, 3 and feeding tube in the operational environmental conditions, with perpendicular wave and current direction.	80
5.12	Effective tensions in anchor lines 7, 3 and feeding tube in the extreme environmental conditions, with perpendicular wave and current direction.	81
5.13	Numerical model during time domain simulation with wave direction set to 90° and current direction set to 0°.	83
5.14	Maximum effective tensions in anchor lines and feeding tube for fish cage during the operational and extreme environmental conditions for wave direction at 90° and current direction set to 0°.	84
5.15	Effective tensions for anchor lines and feeding tube in operational environmental conditions, with wave direction 90° and current direction 0°.	86
5.16	Effective tensions for anchor lines and feeding tube in extreme environmental conditions, with wave direction 90° and current direction 0°.	87
5.17	Maximum effective tensions in anchor lines and feeding tube for fish farm during the operational environmental conditions for a current and wave direction of 0° (white), 90° (light gray), and for alternative current and wave direction of 0° and 90° (dark gray) respectively.	89
5.18	Maximum effective tensions in anchor lines and feeding tube for fish farm during the extreme environmental conditions for a current and wave direction of 0° (white), 90° (light gray), and for alternative current and wave direction of 0° and 90° (dark gray) respectively.	90

5.19	Mean tensions for anchor lines and feeding tube in the operational and extreme environmental conditions for fish cages of net solidity 0.15 (a), 0.20 (b) and 0.25 (c).	93
5.20	Relative increase in mean tensions of line elements between solidity ratios when $S_n = 0.15$ and $S_n = 0.20$	94
5.21	Relative increase in mean tension of line elements between solidity ratios when $S_n = 0.20$ and $S_n = 0.25$	95
5.22	One hour (3600 seconds) time domain simulations for various feeding tube lengths in the operational environmental conditions.	97
5.23	One hour (3600 seconds) time domain simulations for feeding tube length in the extreme environmental conditions.	98
5.24	Distribution of effective tension (left) and bending moment (right) along the length of the feeding tube, for lengths of $L_{1A,B,C} = 100\ m$ (a), $L_{2A,B,C} = 300\ m$ (b), and $L_{3A,B,C} = 600\ m$ (c) for the operational and extreme conditions. . .	100

List of Tables

2.1	Equations to describe response amplitude operators for floating vessels [9].	14
3.1	Properties of the feed barge used for input in Wadam [14].	28
3.2	Natural periods for the feed barge in heave, pitch, and roll.	30
3.3	Six roll and pitch damping coefficients used in simulations selected to compare roll response values of the feed barge from Wadam to experimental values (the same coefficient is used for both roll and pitch).	31
3.4	Dimensions of a feed barge used for the operation of fish farms and a barge studied by reference [14, 22].	32
3.5	Environmental parameters used in the preliminary study of coarse fish farm system in OrcaFlex.	35
3.6	Effective tension for fish cage anchor lines of the combined fish cage and barge simulations at a direction of 0° relative to the x-axis.	40
4.1	Full-scale cylindrical fish cage developed by Aqualine AS [26].	51
4.2	Twine discretization for the coarse, medium and fine numerical fish cage models.	53
4.3	Fish cage equivalence properties for a full-scale fish cage and for numerical fish cage models with net mesh density that is coarse, medium, and fine.	55
4.4	Properties and dimensions of the feeding tube used for fish farming [30].	59
4.5	Relative volume of varying mesh density for fish cage model of coarse, medium, and fine mesh density compared to empirical studies for similar fish cages and conditions [32].	64
5.1	Environmental parameters as defined by Norsk Allmenstandardisering [31].	66
5.2	Operational and extreme environmental conditions for time domain simulations.	67
5.3	Simulation matrix of environmental parameters and variables for time domain simulations in OrcaFlex.	70
5.5	Maximum, minimum, mean tension and standard deviation for anchor lines and feeding tube for simulation of operational condition (1 year return period) and extreme conditions (50 year return period) conditions.	76
5.6	Minimum, maximum, mean tensions and standard deviations for line element for 1 year return period, in operational and extreme conditions for wave and current direction of 90°	82
5.7	Minimum, maximum, mean tension and standard deviation for anchor lines and feeding tube for simulations for operational and extreme environmental conditions with wave direction set to 90° and current direction set to 0°	88

5.8	Standard deviations for anchor lines 3, 7, 8 6 and feeding tube in operational and extreme conditions	91
5.9	Maximum bending moments in operational conditions without (L_A) and with (L_C) bend stiffener for feed barge, and fish cage connection points.	101

1 Introduction

In this chapter, the primary motivations for the thesis and the scope are presented, in addition to an introduction to the current status of the Norwegian aquaculture industry, developments and challenges.

Over the past decades, aquaculture has grown to be one of the industrial pillars in Norway, with massive fish farming operations for seafood such as Atlantic salmon, rainbow trout, cod, and halibut[1]. Furthermore, the Norwegian government has established a goal of increasing the production levels of 2010 fivefold to meet the demands of a growing global population, increased wealth, and demand for seafood [1]. In historical terms, the growth of the industry has been tremendous since aquaculture was commercialized in Norway around 1970, with Atlantic salmon accounting for over 80% of the total aquaculture production [2]. As can be seen in Fig. 1.1, production levels have grown significantly since the 1990s with considerable growth in the last decade. The industry has to make giant strides in improving the value chain to maximize the production of existing facilities. There is also little doubt that new technology will have a significant impact on the next generation of fish farms.

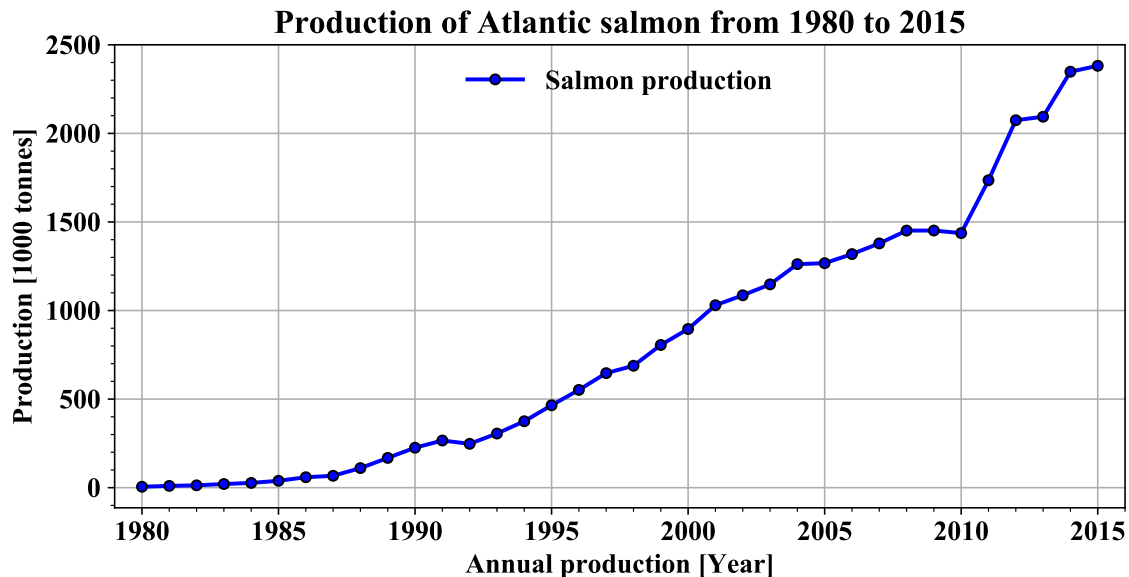


Figure 1.1: Production levels of Atlantic salmon from 1980 until 2015 [2].

1.1 Background and Motivation

There is a growing demand for fish, such as salmon and trout in the international seafood market. As shown in Fig. 1.2, the coast of Norway is filled by a coastline crowded with fish farms, which has

in addition to being among the weakest structural components. It provides the fish with feed at a high rate, in addition to being exposed to current and wave loads. Furthermore, static electricity can build up when using air-based feeding systems. For a fish farm operating in remote locations, ruptures in the feeding tube, maintenance and repair can cause problems due to the challenging logistics and environmental parameters. Therefore, research on the environmental parameters and loading characteristics of feeding tubes is of interest to the industry.

1.2 Technological Development and Current Trends

The primary technological concerns of the industry are the lack of available locations to establish fish farming operations along the Norwegian coastline and licenses to establish a fish farm are costly. Modern-day fish farming has come a long way from the simpler operations set up by the fish farmers that established commercial aquaculture in Norway around 1970. Conventional fish farms that are typically used in the industry consist of a mooring system, a floating collar, feeding tubes, a sinker tube, and a net. They are typically moored to the seabed, or the shore and floaters are built to keep the facility buoyant, but not for harsh environmental conditions. As shown in Fig. 1.3, a conventional fish farm consists of a feed barge, feeding tubes, a fish cage, and other operational support systems. However, a modern fish farm also incorporates sensors to optimize feeding, monitor behavior of the livestock, and monitor oxygen levels[5].



Figure 1.3: Conventional fish farm with support systems [5].

Offshore fish farming is not limited by technological development, but rather by the fact that there are no apparent alternatives for operating a fish farm in exposed climates for an extended period of time. There are also limitations in the structural capacity of conventional fish farms, which limit the scalability of some of the smaller facilities that operate along the coast. Several concepts have been developed recently, such as the vessel-shaped fish farm called Project Havfarm shown in Fig. 1.4. It is established as an alternative for the sustainable production of aquaculture in challenging environmental conditions. It is tailored for the final growth phase of the fish in conditions that provide optimal environmental conditions [6].

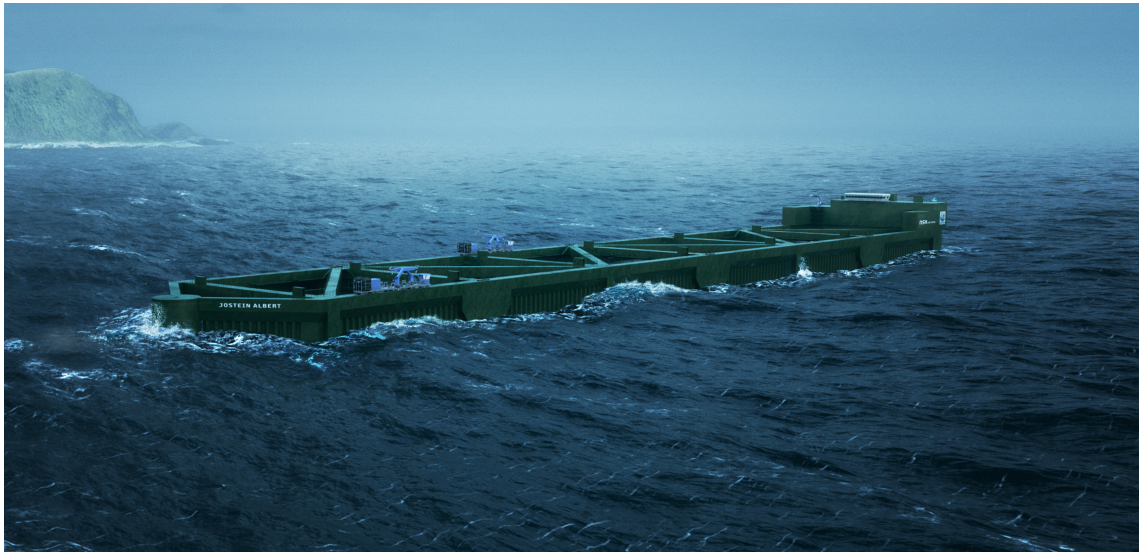


Figure 1.4: Project Havfarm is a vessel shaped fish farm developed for exposed environmental conditions [6].

One of the largest structures built for aquaculture and currently undergoing testing in the North Sea outside of the island of Frøya is the Ocean Farm 1 operated by SalMar as shown in Fig. 1.5. It is the world's first offshore fish farm and is designed to test biological and technological aspects in offshore fish farming. The fish cage in the Ocean Farm 1 has a volume of 250 000 m³ and can contain up to 6 249 tonnes of salmon [7]. The facility has a variety of sensors to monitor the salmon population such, as surveillance, echolocation, and automated feeding. There are also sensors for navigation, communication, decision support, and energy control that aim to simplify and optimize operations. The facility is submersible and is, therefore, better suited to deal with the challenging climate in the North Sea. It is, however, vital to maintain sufficient production levels to cover increased costs of logistics, operations, and maintenance costs for an offshore facility. Although the installations are large, the scalability of such a facility is limited by large costs and the necessary magnitude of the operations. An example of the limitations is the cost for the Ocean Farm 1, which

was almost 700 million NOK. The cost of the facility is an indicator that offshore fish farming requires considerable volumes of produce to maintain profitability.



Figure 1.5: Ocean Farm 1 operated by SalMar outside the island of Frøya [7].

These massive fish farms are constructed for the offshore environment and by far dwarf the coastal fish farms. A typical fish farm typically consists of a feed barge, several fish cages, feeding tubes, and mooring lines. Also, there are multiple support systems to monitor the fish, perform maintenance, and sustain the operation. However, a company called Atlantis Subsea Farming AS is developing a submerged fish farm, which can be lowered into the ocean to reduce exposure to waves, wind, and lice [8]. Their approach to offshore fish farming is better suited to the needs of the smaller coastal operators and has the potential to solve the challenges of small scale offshore fish farming. The primary reason why a submerged fish cage is better suited for the offshore environment is the ability to avoid a significant portion of the wave motion at the surface of the ocean, and thereby also avoid the wave load. As can be seen in Fig. 1.6, the cage is built with an air dome at the top of the cage to provide the fish with the necessary oxygen levels, in addition to feeding the fish while the cage is submerged [8]. A submersible fish farm requires a mooring system able to adjust the vertical displacement of the cage, oxygen in the top of the cage, and net covering the top of the fish cage. Moreover, the submerged fish farm is dependent on providing the fish with water-based feed, and oxygen must be pumped into the submerged fish cage. The overall goal is to achieve adequate care for the livestock and reduce stress triggering factors for facilities located in exposed locations. An essential factor in the testing of the facility is to maintain profitability, in order for the submerged fish cage to be a realistic alternative in order to ensure commercial feasibility, which is a major concern for coastal fish farms when considering offshore fish farming.

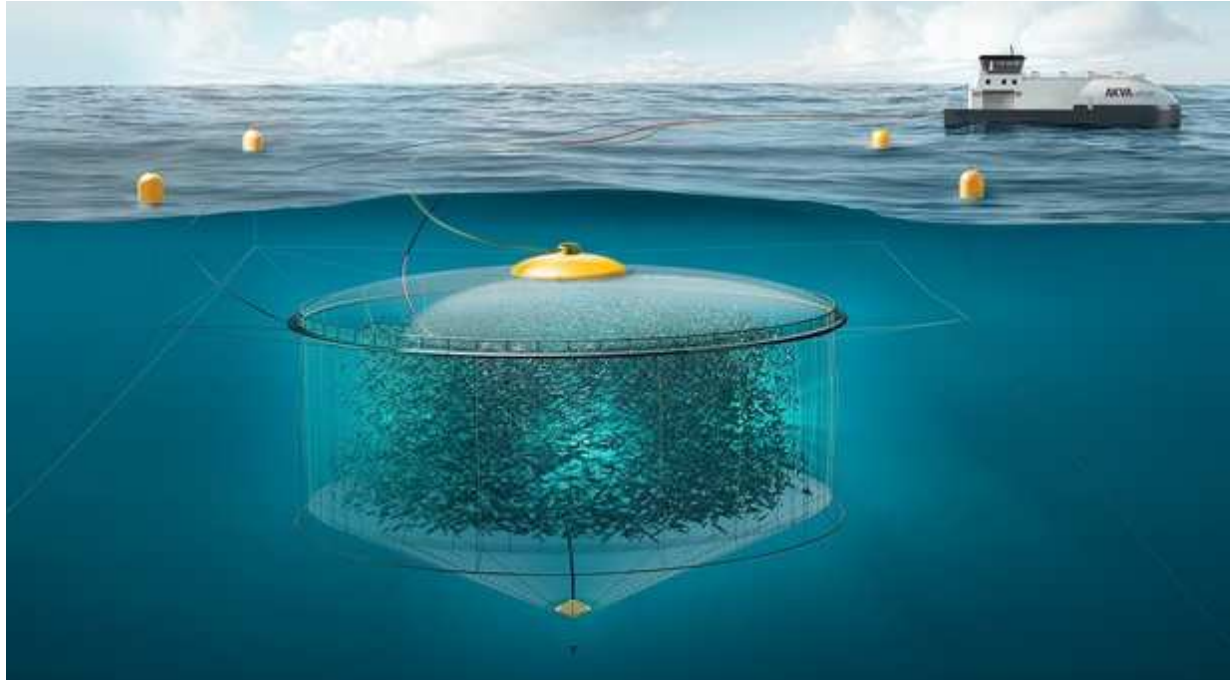


Figure 1.6: Submerged fish cage concept developed by Atlantis Subsea Farming AS with water based feeding from nearby feed barge [8].

1.3 Scope and Thesis Outline

As the Norwegian aquaculture industry is increasingly dependent on placing fish farm facilities in exposed environmental locations along the coastline, there is an increased need for facilities able to withstand harsh environmental conditions in the North Sea. This thesis seeks to explore the loading mechanisms and effects that are critical in order to establish sustainable aquaculture facilities, which will be necessary for the industry to meet its ambitious goals of increased production in the coming decades. A numerical fish farm model and feeding system are established using the object-oriented programming software called Python in conjuncture with the dynamic analysis software OrcaFlex and the application programming interface provided by Orcina Ltd. The numerical model is then used to investigate the behavior and response of the system to different environmental loading, under current and irregular waves. The different loading mechanisms are analyzed by studying net deformation and mooring line tension, and the findings are subsequently presented.

Chapter 2 introduces fundamental theories applied in the thesis, such as potential flow theory, vessel response in irregular waves, and damping effects. Also, the hydrodynamic loading on a fish cage and slender cylinders are presented.

Chapter 3 contains a description of the software applied in the thesis work and a preliminary study of a fish farm numerical model. The hydrodynamic modeling in the frequency domain for the feed barge is also introduced.

Chapter 4 presents the method for calculating equivalent net properties and also the procedure of establishing a fish cage numerical model using Python and the OrcaFlex API (Application programming interface). A short overview of a conventional mooring system and feeding tubes are also provided, in addition to a convergence study for three numerical fish cage models.

Chapter 5 introduces relevant results and discussions from dynamic time domain simulations performed in OrcaFlex. The numerical model has been analyzed for the impact of different parameters such as environmental loading, wave and current direction, solidity ratio, and feeding tube length.

Chapter 6 presents a summary of the findings in the report, and possible sources of errors encountered in the thesis are also discussed. A recommendation for future work is also introduced.

2 Theory

In this chapter, relevant theories related to the hydrodynamic response and loading effects on a feed barge, fish cage, mooring system, and feeding tube are presented. These lay the foundation for understanding the loading mechanisms and environmental effects on a fish farm. Understanding the behavior of the fish farm system will be important for the analysis of the numerical models established in subsequent chapters.

2.1 Potential Flow Theory

Potential flow theory is used to describe the motion of water particles assuming non-rotational flow[9]. This is useful when applied to the motion of fluids for the feed barge, fish cage, mooring lines and feeding tubes for a fish farm. In order to establish a fundamental understanding of important hydrodynamic elements, the relations of the potential function are established. The potential function describes a velocity field as a gradient of a scalar function. In hydrodynamics, this is done by combining conservation of mass, momentum and establishing boundary conditions, which can then be used to establish a velocity field. Eq. (2.1), gives the relation of the potential function in three-dimensional directions and time [9].

$$\varphi = \varphi(x, y, z, t) \quad (2.1)$$

Partial derivatives of the potential function with respect to direction results in the velocities in that direction is given by Eq. (2.2).

$$\nabla\varphi = \frac{\partial\varphi}{\partial x}\vec{i} + \frac{\partial\varphi}{\partial y}\vec{j} + \frac{\partial\varphi}{\partial z}\vec{k} = \vec{U} \quad (2.2)$$

Based on the conditions that the rotation of the fluid is zero ($\nabla \times \vec{U} = \vec{0}$) and that the fluid is incompressible ($\nabla \cdot \vec{U} = 0$), the Laplace differential equation of second order is then given by Eq. (2.3).

$$\nabla^2\varphi = 0 \quad (2.3)$$

Also, Eq. (2.4) gives the relation for potential flow as.

$$\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} + \frac{\partial^2\varphi}{\partial z^2} = 0 \quad (2.4)$$

When solving for the velocity potential, four boundary conditions are applied, which express the constraints of the flow of a fluid.

1. Kinematic bed boundary conditions, or the bottom boundary condition as there can be no flow through the seabed, expressed as, $\frac{\partial \varphi}{\partial z} \Big|_{z=-d} = 0$.
2. No water can flow through the surface of a body and there can be no disturbance of the incident wave.
3. There are two free surface boundary conditions:
 - Kinematic free surface boundary condition, states that water particles located at the surface will remain at the surface, $\frac{\partial \varphi}{\partial z} \Big|_{z=0} = \frac{\partial \xi}{\partial t}$.
 - Dynamic free surface boundary condition, states that the pressure at free surface is constant and equal to atmospheric pressure, $\xi = -\frac{1}{g} \frac{\partial \varphi}{\partial t} \Big|_{z=0}$.

Hence, for a linear velocity profile, the velocity potential can be derived as shown in Eq. (2.5).

$$\varphi(x, z, t) = \frac{\xi_0 g \cosh k(z + d)}{\omega \cosh(kd)} \cos(\omega t - kx) \quad (2.5)$$

Where:

- ξ_0 = Wave amplitude
- g = Gravitational constant,
- z = Mean surface elevation
- d = Wave height
- k = Wave number
- ω = Angular frequency

2.1.1 Linear Wave Theory

To understand the behavior of waves, linear wave theory is established to describe the propagation of waves at the surface layer. By linearizing the boundary conditions mentioned in Section 2.1,

regular waves are obtained [9]. Regular waves have sinusoidal shapes, which higher order waves do not (they have higher crests than troughs). A sine or cosine function is called regular waves, and a surface profile equation can be derived from the potential function by using the dynamic boundary condition. Assumptions for linear wave theory states that:

- The wave amplitude is small compared to the wavelength.
- Body stays in its mean position.
- Motion of the body is of the same order as the amplitude of the wave.

The equation for a deep water linear wave profile is given by Eq. (2.6). In Fig.2.1, four linear wave profiles are shown in the top four subplots.

$$\zeta = \zeta_0 \sin(\omega t - kx) \quad (2.6)$$

Where the wave number, k is given by the equation, $k = \frac{2\pi}{L}$, which can be used to find the relationship between wave period and wavelength shown by Eq. (2.7) [9].

$$T = \sqrt{\frac{2\pi}{g} L} \quad (2.7)$$

The water depth for linear wave theory is an important factor and is divided into shallow, intermediate and deep water [9]. Deep water is used for water depths of more than 500 m. However, the water depth relation, d , is usually in reference to the wavelength, L . The relation states that water is deep for $d > \frac{L}{2}$, intermediate for $\frac{1}{20} < \frac{d}{L} < \frac{L}{2}$, shallow for $\frac{d}{L} < \frac{1}{20}$. The linear velocity profile is shown in Eq. (2.5). However, in this thesis, all wave conditions are based on deep water waves, as the focus of this thesis is on environmental conditions in the offshore environment of the North Atlantic.

2.2 Response of a Vessel in Irregular Waves

2.2.1 Irregular Waves

Regular waves are established for ideal conditions, but for studies on the surface profile of the sea, the surface profile is too chaotic to keep track of all the waves. Furthermore, irregular waves

are based on the need for a description of the chaotic sea surface in a more manageable way. By combining or superimposing an infinite number of regular waves with different amplitudes and wave numbers, through a Fourier analysis, waves can be used to form an approximation of the sea surface [9]. As can be seen in Fig. 2.1, the superpositioning of multiple linear waves results in a chaotic surface that forms the surface layer of the water column. In the figure, four linear waves are superimposed to give an irregular wave.

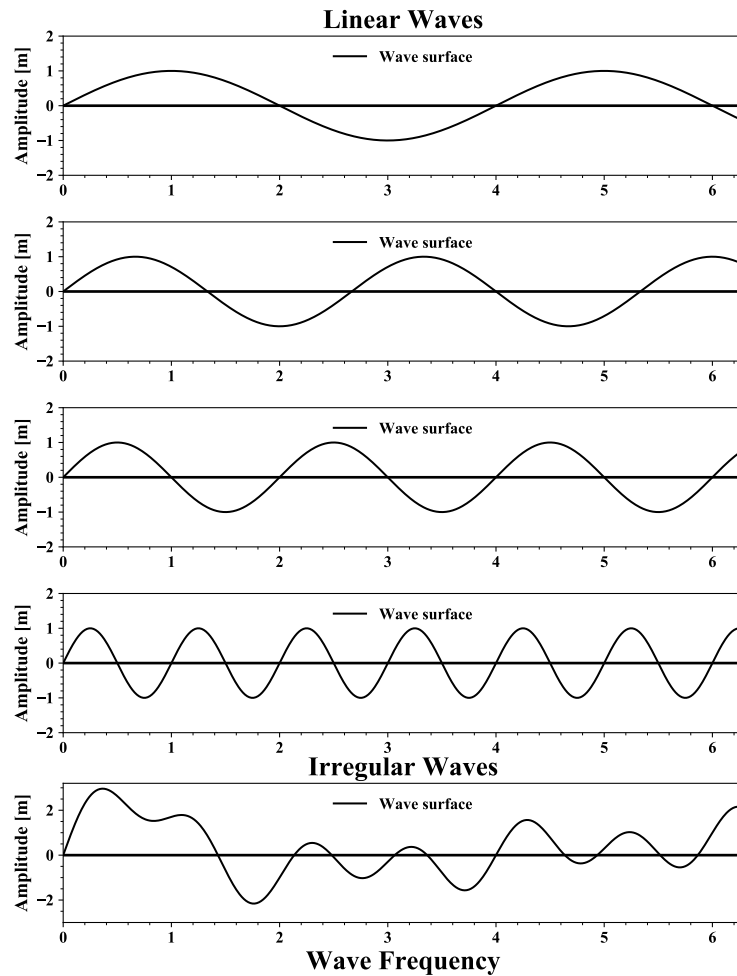


Figure 2.1: Linear waves (top four subplots) superimposed to form an irregular wave (bottom subplot) [9].

Repeating this process infinitely will result in a chaotic wave surface profile, approximate to that of the sea. Conducting numerical simulations in the time domain, it is important to study the impact of irregular waves as it is a chaotic process that changes over time and has no repetitive cycle. An alternate way of describing the sea surface is in terms of ocean wave surface energy per area.

Wave energy spectra such as the JONSWAP (Joint North Sea Wave Project) spectrum is used to describe irregular wave conditions. The spectrum is an extension of the Pierson-Moskowitz spectrum, which was based on measurements in the North Atlantic. It was initially proposed for fully-developed sea, meaning that wind and waves reach equilibrium as winds blow steadily for an extended period of time. The JONSWAP spectrum includes fetch limited seas and describes developing sea states, taking into account the effect that waves grow with distance and describes developing sea [11]. A comparison of the Pierson-Moskowitz and the JONSWAP spectrum is shown in Fig. 2.2.

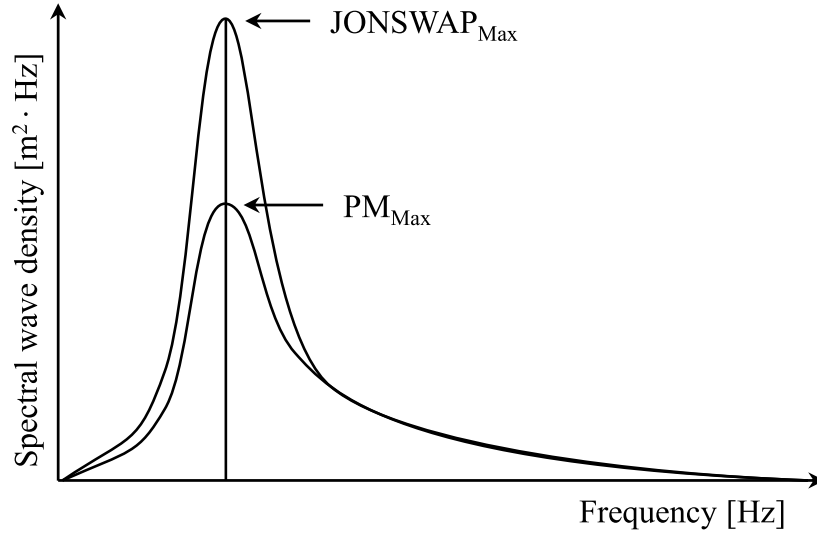


Figure 2.2: Comparison between the Pierson-Moskowitz and JONSWAP spectra [11].

The Pierson-Moskowitz spectrum is given by Eq. (2.8).

$$S_{PM}(\omega) = \frac{5}{16} \cdot H_S^2 \cdot \omega_P^4 \cdot \omega^{-5} \exp\left(-\frac{5}{4} \left(\frac{\omega}{\omega_P}\right)^{-4}\right) \quad (2.8)$$

The JONSWAP spectrum is given by Eq. (2.9).

$$S_J(\omega) = A_\gamma S_{PM}(\omega) \gamma^{\exp(-0.5(\frac{\omega - \omega_P}{\sigma \omega_P})^2)} \quad (2.9)$$

Where:

- H_S = Significant wave height
- ω = Angular wave frequency
- ω_P = Angular spectral peak frequency
- S_{PM} = Pierson-Moskowitz spectrum
- γ = Non-dimensional peak shape parameter
- σ = Spectral width parameter
- A_γ = Normalizing factor

2.2.2 Response Amplitude Operators

The motion of marine structures is described by response amplitude operators (RAO), which refers to the movement of a rigid body in three-dimensional space. Movement can occur both as translational, and rotational movement. The translational movements surge, sway and heave refer to displacement from one point to another, while rotational motion such as roll, pitch, and yaw refer to rotation about a fixed point.

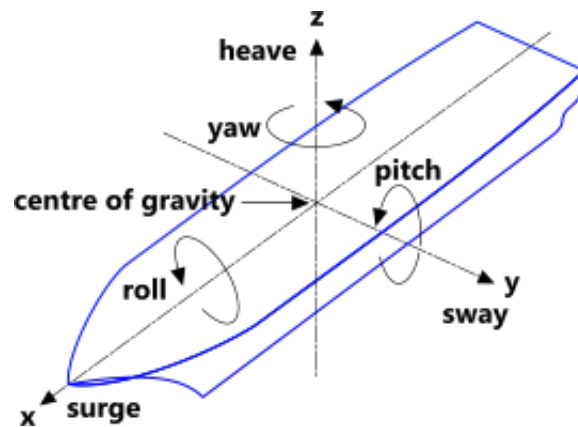


Figure 2.3: Response amplitude operators for floating vessels described in six degrees of freedom as shown by Calqlata [12].

In Fig. 2.3, for a conventional vessel with no forward speed, surge motion is defined as forward or backward motion. Directional steering in yaw is usually controlled by a rudder, while sway motion is generally considered to be an unnatural motion of a vessel. However, for modern ships, thrusters may be mounted underneath the ship for dynamic positioning systems, which allow for

sideways sway motion. The remaining motions such as roll, pitch, and heave are usually controlled by the external environment and the chaotic motion of the sea. A feed barge is usually moored to the seabed and has no thrust, forward motion or rudders. Therefore, it has no surge, sway, or yaw motion. However, it is important to understand the motion of the vessel, and the response amplitude operators for heave, pitch and roll are relevant when trying to understand the vessel motion of the feed barge for different environmental parameters. Equations for response amplitude operators are established to describe the movement of a vessel in six degrees of freedom and can be seen in Table 2.1 [9]. The equations are used to indicate the response of the vessel in the different degrees of freedom and are important parameters for studying hydrodynamic response.

Table 2.1: Equations to describe response amplitude operators for floating vessels [9].

1.	Surge	$x(t) = x_a \cdot \cos(\omega t + \varepsilon_x \zeta)$
2.	Sway	$y(t) = y_a \cdot \cos(\omega t + \varepsilon_y \zeta)$
3.	Heave	$z(t) = z_a \cdot \cos(\omega t + \varepsilon_z \zeta)$
4.	Roll	$\phi(t) = \phi_a \cdot \cos(\omega t + \varepsilon_k \zeta)$
5.	Pitch	$\theta(t) = \theta_a \cdot \cos(\omega t + \varepsilon_k \zeta)$
6.	Yaw	$\Psi(t) = \Psi_a \cdot \cos(\omega t + \varepsilon_k \zeta)$

2.2.3 Damping Effects and Added Mass

A vessel floating in the ocean will experience translational and rotational motion such as sway, heave, yaw, pitch and roll motion, as discussed in Section 2.2.2. These motions generate waves that propagate radially, meaning that the vessel is losing energy to create waves, which dampen the motion of the vessel. Hydrodynamic damping effects for a vessel are primarily caused by the two components potential damping and viscous damping [13].

Potential damping is the predominant damping component for vessel motion. Potential damping is primarily caused by the waves that are formed, which dissipate energy from a moving vessel, and it is proportional to the velocity of the vessel in a linear system. Viscous damping is caused by both skin friction and the formation of vortices because of the vessel motion. The forming vortices lead to fluid separation, causing a pressure drop along the hull of the vessel, which dissipates energy. Viscous damping effects are relatively small compared to potential damping effects and are often ignored for calculations of the motion of offshore structures and vessels.

The damping components have very different contributions depending on the motion considered. As an example, surge damping is primarily dominated by potential damping, as the wave damping increases proportionally with the velocity of the vessel, and viscous damping is small by comparison. However, for roll damping, the roll motion of a vessel usually does not generate a lot of wave motion; hence, the potential damping in roll is low in comparison to the viscous damping effects, especially for bilge keels. Viscous damping effects due to roll motion is shown in Fig. 2.4.

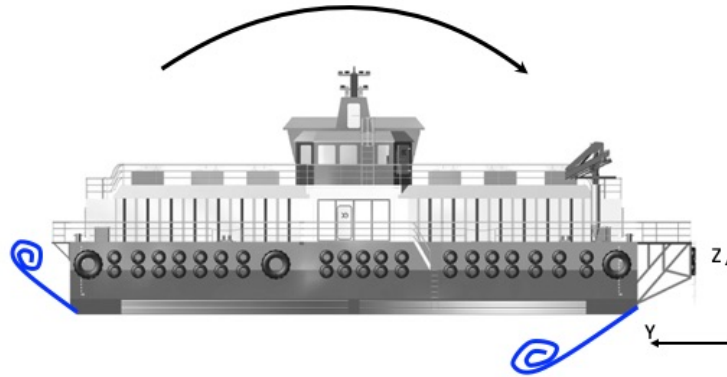


Figure 2.4: Viscous roll damping effects for a feed barge occurring as a result of vortices forming because of roll motion [14].

For a vessel floating unsteadily on a liquid surface, the submerged body of the vessel will be exposed to added mass effects. Added mass is an effect as a result of the fluid acting on the submerged body of a vessel and can be described as the inertia added to a system because an accelerating body displaces the volume of a fluid as it moves through it. For slender elements, the expression for the added mass coefficient is given by Eq. (2.10).

$$C_A = \frac{m_a}{\rho A} \quad (2.10)$$

The added mass is usually included in numerical simulations by including an added mass coefficient, where m_a is the added mass per unit length, and A is the geometric cross-sectional area of a slender element [9]. Potential damping and added mass can be obtained from potential flow theory, but viscous damping is obtained from tests or empirical formulas.

2.3 Loads on Fish Farm System

As the focus of the thesis is to analyze the environmental loads on a fish farm system, it is important to establish the load mechanisms that affect the feed barge, the fish cage, the feeding tube and

the mooring lines connecting the system. These variables will provide insight into the loading mechanisms of an offshore environment for a fish farm system and are important factors to consider when designing a system for challenging environmental conditions. The hydrodynamic loading on a fish cage lays the foundation for the analysis of this thesis and considers the tension in mooring and net lines, in addition to the volume deformation, drag and mass forces on the net of the fish cage. A complete fish farm system can be seen in Fig. 2.5. In the figure, the feeding tube can be seen as connecting the feed barge to the fish cage and the subsequent mooring lines anchoring the feed barge and fish cage to the seafloor.

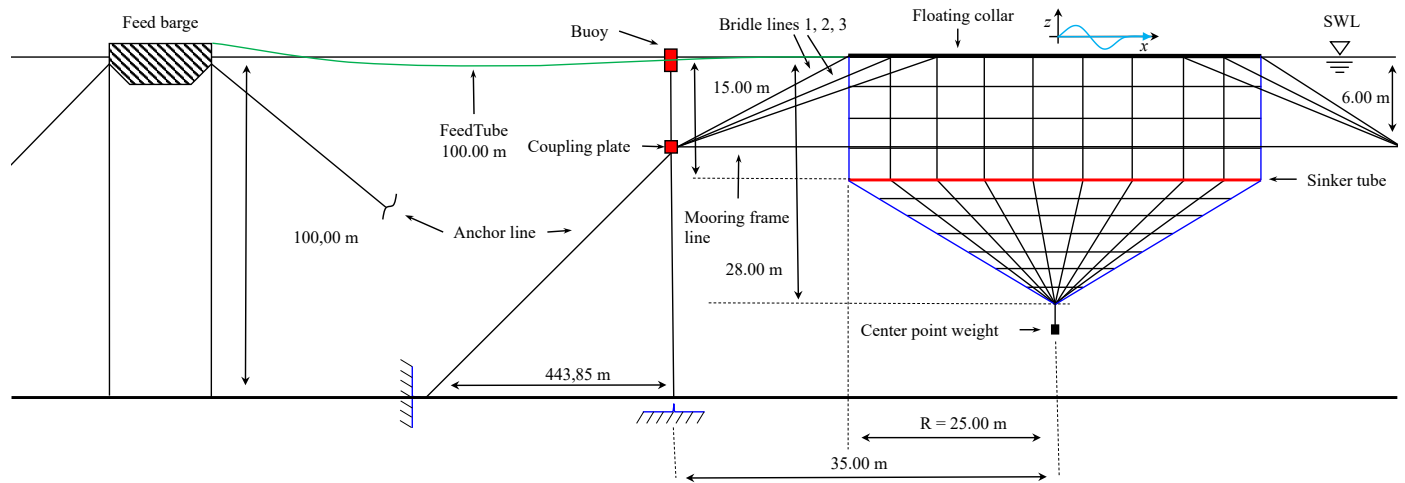


Figure 2.5: Conventional fish farm system including feed barge, feeding tube, fish cage and mooring lines.

In this section, the loading mechanisms on submerged cylinders such as the net twines, or mooring lines, will be established. In addition, the loading effects on a floating feeding tube will be introduced. This is done by considering the cylinders as long, and thin cylinders, such that the acceleration is constant over the length of the cylinder and $D/L < 1/5$ [9].

2.3.1 Morison's Equation

Morison's equation gives a relation for the force on a slender body oscillating in flow. The formula is based on the assumption that the relationship between horizontal dimensions and wavelength is small. The equation is therefore adequate for calculating the drag force of slender elements, such as the twines of a fish cage. The formula is the sum of the mass force and the drag force. The equation is based on the mass force f_M , in phase with the local flow of acceleration and a drag force f_D , which is proportional to the square of instantaneous flow velocity, or relative velocity.

For a submerged cylinder in a wave with a $\frac{D}{L} < \frac{1}{5}$ ratio, Morison's equation based on experiments is given by Eq.(2.11) [9].

$$f(z, t) = f_M + f_D = \frac{\pi D^2}{4} \rho C_m \cdot \dot{u}(t) + \frac{1}{2} \rho C_D D \cdot u(t) |u(t)| \quad (2.11)$$

Where:

C_m = Inertia coefficient

C_D = Drag coefficient

ρ = Mass density of the fluid

D = Diameter of twine

$\dot{u}(t)$ = Acceleration of flow at center of cylinder

$u(t)$ = Horizontal water particle velocity

There are vortex shedding effects, which in addition to the oscillatory in-line force, there are lift forces perpendicular to the flow direction. These are not covered by Morison's equation, which only considers the horizontal forces (also known as in-line forces) exerted on a body. However, Eq. (2.11) gives the relation of the Morison equation for a cylindrical body in a stationary position. When exposed to wave motion and loads, it will no longer be stationary. It is, therefore, necessary to develop an expression for the Morison equation for a vertical cylinder moving with a velocity $v(t)$, in a fluid with velocity $u(t)$. The Morison equation is then given by Eq. (2.12) [15].

$$f(z, t) = \underbrace{\frac{\pi D^2}{4} \rho C_m \cdot \dot{u}(t)}_I - \underbrace{\frac{\pi D^2}{4} \rho C_A \cdot \dot{v}(t)}_{II} + \underbrace{\frac{1}{2} \rho C_D D \cdot (u(t) - v(t)) |u(t) - v(t)|}_{III} \quad (2.12)$$

Where:

I = Froude - Krylov and diffraction force

II = Hydrodynamic mass force

III = Mass density of the fluid

C_A = Added mass coefficient

$\dot{v}(t)$ = Acceleration of the body

The expression to calculate the force on the numerical model of the fish cage for simulations for various wave and current values. For steady current conditions, the Froude - Krylov and diffraction forces are zero, and it is only the drag term that contributes to hydrodynamic forces. When exposed to wave-current conditions, both terms will contribute to the hydrodynamic response as the model experiences acceleration.

The drag coefficient of Morison's equation is a function of the flow, described by Reynolds number Re , and the roughness of the cylinder surface k . If the roughness of a structure increases, so will the drag coefficient, and so will the drag force. For a fish cage, the Reynolds number is used to describe the characteristics of the water flowing around the twines of the fish cage and is given by the following Eq. (2.13) [9].

$$Re = \frac{V \cdot D}{\nu} \quad (2.13)$$

Where:

V = Normal component of fluid velocity relative to mesh bar

D = Diameter of the twine

ν = Kinematic viscosity of water

From the Reynolds number, the drag coefficient of a cylinder for a Morison model, can be calculated by applying the following Eqs. (2.14) and (2.15) [9].

$$C_n = \begin{cases} \frac{8\pi}{Re_n s} (1 - 0.87s^{-2}) & (0 < Re_n \leq 1) \\ 1.45 + 8.55Re_n^{-0.9} & (0 < Re_n \leq 30) \\ 1.1 + 4Re_n^{-0.5} & (0 < Re_n \leq 10^5) \end{cases} \quad (2.14)$$

$$C_\tau = \pi\mu(0.55Re_n^{0.5} + 0.084Re_n^{\frac{2}{3}}) \quad (2.15)$$

Here, the coefficient C_n is the normal drag coefficient, and C_τ is tangential drag coefficients for the mesh bars. An illustration of tangential and normal drag force can be seen in Fig. 2.6. For combined wave-current flows, it is indicated that the drag force coefficient is smaller than that of simulations with waves only. Furthermore, the mass force coefficient, or inertia coefficient, C_m can be found as an expression based on the added mass coefficient shown in Eq. (2.16).

$$C_m = 1 + C_A \quad (2.16)$$

In Eq. (2.16), the mass force coefficient is based on the added mass coefficient and the relation for the non-dimensional added mass coefficient, C_A coefficient is given by Eq. (2.10) [11].

2.3.2 Hydrodynamics of a Fish Cage

Two models are widely used to calculate drag forces on net structures with slender cylinder elements. These are used to express the loading mechanisms and are applicable to fish cages. The two models are known as the Morison model and the Screen model. The Morison model determines drag coefficients based on the Reynolds number and the diameter of the twine. The Screen model determines drag based on the ratio of the solid area in the screen, the angle of attack, and Reynolds number at the twines. For dynamic analysis in OrcaFlex, hydrodynamic loads on lines and buoys are calculated by utilizing Morison theory.

Morison Model

The fish cage modeled in this thesis utilizes Morison's equation, where net twines are considered to be slender cylinder elements, as the diameter of each twine in relation to its length is small. This means that during wave and current loads, expressions for the loading can be found by applying Morison theory. For fish cage deformation, the force on inclined twines can be decomposed into a normal and tangential force as illustrated by Fig. 2.6 and are shown in Eqs. (2.17) and (2.18) [11]. The total drag force of a net structure is computed by summing the drag and lift forces of all knot and twine elements. However, the interaction between twine elements is not accounted for.

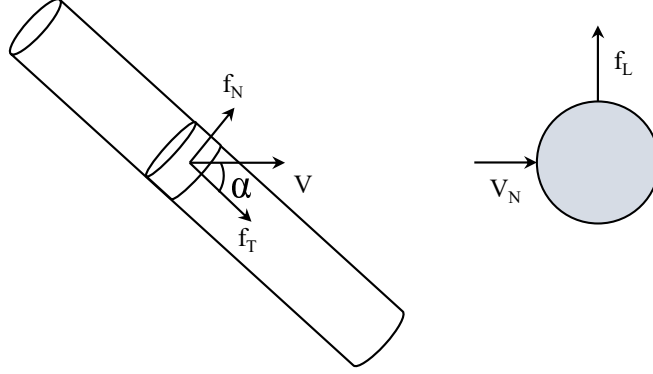


Figure 2.6: Similar to a twine in a fish cage, an inclined slender cylinder indicating normal and tangential force (left) and lift force for the cylinder (right) based on Morison theory.

$$f_N = \frac{1}{2} \rho C_N D v_n |v_n| \quad (2.17)$$

$$f_T = \frac{1}{2} \rho C_T D v_n |v_n| \quad (2.18)$$

Where the drag coefficient normal to the pipe c_n , depends on the Reynolds number and incident angle of the flow. The tangential drag coefficient can be treated as a constant and mainly depends on the skin friction of the material $c_f = 0.02$ for smooth circular cylinder [11].

Screen Model

The total drag force of a net structure can also be computed by summing drag and lift forces on individual net panels. The drag coefficients of the net panel are determined from the solidity ratio (Sn), Reynolds number (Re), and the angle of attack θ , as shown by Eqs. (2.19) and (2.20). The model was developed by Kristiansen and Faltinsen [10].

$$f_L = \frac{1}{2} \rho C_L(Re, Sn, \theta) A \cdot U_{rel}^2 \quad (2.19)$$

$$f_D = \frac{1}{2} \rho C_D(Re, Sn, \theta) A \cdot U_{rel}^2 \quad (2.20)$$

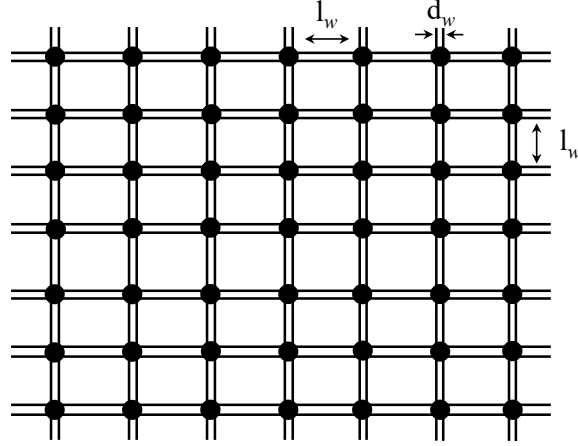


Figure 2.7: The net screen model as introduced by Kristiansen and Faltinsen [10].

As can be seen from Fig.2.7, a screen model of a fish cage can be seen, where l_w is the twine length and d_w is the diameter of the twine. Comparing the calculations with those of empirical studies has rewarded similar results; however, there are some limitations to this method. For large deformation in the structure of the net, this method is non-applicable as responses in wave conditions are too complicated for the method when the angle of attack continually changes.

Solidity Ratio

For a fish cage, the solidity ratio is the relationship between the solid area of a net and the total area enclosed by the net. For a fish cage, the relationship between the area of the twines of a net and the total area is an essential property for the drag that the net will experience when submerged. The expression for the solidity ratio is given in Eq. (2.21) and for a normal fish cage, the solidity ratio usually varies from 0.20 to 0.35.

$$S_n = 2 \cdot \frac{d_w}{l_w} - \left(\frac{d_w}{l_w}\right)^2 \quad (2.21)$$

Effects such as biofouling can, over time, lead to an increase in the solidity ratio of the net and significantly increase the drag related loading on the fish cage. For a square-woven net, the solidity ratio is the relationship between the projected area of the twines to the total area of the net, where d_w is the twine diameter and l_w is the twine length [16].

2.3.3 Hydrodynamic Loading of the Feeding Tube

A feeding tube is usually a long homogeneous pipe built in high-density polyethylene (HDPE) and is used to transport fish feed from a feed barge to livestock in a fish cage. High-density polyethylene

is the standard material used because it is affordable, durable, flexible, and floats in saltwater, allowing for easy maintenance and installation of the pipes [17]. As shown in Fig. 2.8, feeding tubes are partly submerged and move up and down in the water surface. It is, therefore, necessary to consider the slamming loads on the tubes.



Figure 2.8: Feeding tubes from a feeding barge going to multiple fish cages at a Scottish aquaculture facility [18].

As the material choice of constructing feeding tubes using HDPE, there is little flexibility with regards to choosing a suitable material. Therefore, the thesis will focus on studying the loading characteristics of the pipe for different environmental conditions. Traditionally, the feed is transported through the feeding tube using air based feeding systems, but water-based systems are being developed. For this thesis, the focus will primarily be on feed systems based on transporting feed using air. A feeding tube floating on the surface can at times be experiencing large wave response, and parts of the floater may be fully submerged, or dry at times, which leads to water entry/exit mechanisms, such as slamming loads. In addition to slamming loads and Morison's theory, it is important to consider the wave loads and motion for the feeding tube [19].

For linear potential theory, it is reasonable to assume that the feeding tube is a floating, slender, elastic structure and that the response for wave heights are small compared to the cylinder diameter. The loading on the feeding tube is related to the bending stiffness and material properties of the feeding tube. The water plane stiffness for the cross-sectional area of a cylinder, assuming horizontal cylinder and free surface, is important in quantifying and understanding the behavior of the floating cylinder. Combined with the water plane stiffness, the distributed buoyancy is expressed by Eqs. (2.22) and (2.23) [19].

$$k_w = \rho_w g A_w \quad (2.22)$$

$$B_{distr} = \rho_w g l_w A_{sub} \quad (2.23)$$

A feeding tube floating on the water surface will be bobbing up and down vertically in the sea surface because of its positive buoyancy. However, based on linear wave potential, the tube will follow the surface motion of small amplitude waves. It is also important to take into account and establish free surface effects acting on the feeding tube. Slamming loads is an impulse load that can be caused by the impact of a body hitting the water. The slamming of an object that is lowered through the free surface is defined as the rate of change of fluid momentum, and is described by Eq. (2.24) [11, 20]. However, it should be noted that the buoyancy and drag force of the cylinder should be adjusted for the cylinder because of the oscillating motion of the positively buoyant feeding tube.

$$f_s(t) = \frac{d}{dt}(a_{33}^{\infty} v_s) = a_{33}^{\infty} \dot{v}_s + v_s^2 \frac{d}{dh}(a_{33}^{\infty}) \quad (2.24)$$

For a horizontal cylinder, the slamming force per unit length F_S , of a horizontal cylinder is given by the relation in Eq. (2.25) [11].

$$f_s(t) = \frac{1}{2} \rho C_s D v_s^2 \quad (2.25)$$

Where:

ρ = Mass density of water

D = Diameter of the cylinder

C_s = Slamming coefficient

The slamming coefficient is given by the relation, Eq. (2.26) [20].

$$C_s = \frac{2}{\rho D} \frac{d}{dh}(a_{33}^{\infty}) \quad (2.26)$$

Where the rate of change of sectional added mass is given by $\frac{d}{dh}(a_{33}^{\infty})$ for submergence h . The added mass coefficient in the vertical direction for high frequency as a function of submergence is given by Fig. 2.9 [20]. The solid line is used for non-dimensional added mass, while the dotted line is used for the derivative of the added mass with respect to the submergence (given by $\frac{d}{dh}(a_{33}^{\infty})$).

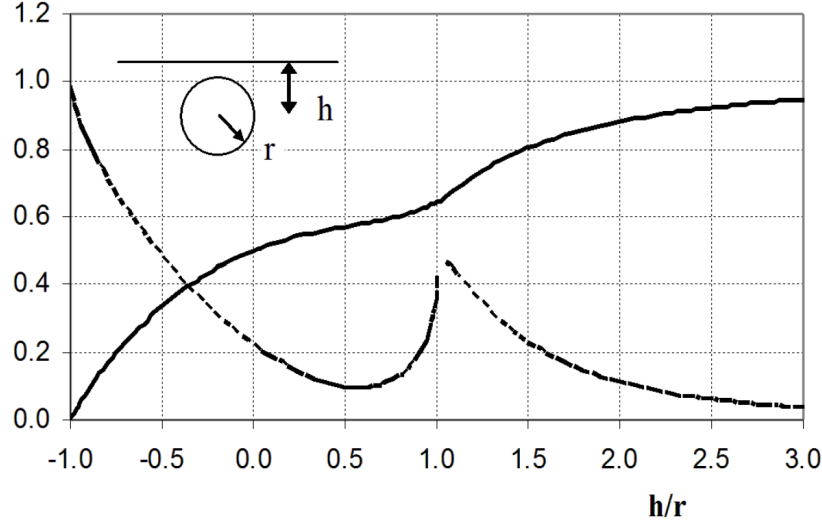


Figure 2.9: High-frequency limit of vertical added mass coefficient as a function of water depth [20].

The non-dimensional added mass is given by Eq. (2.10), but can also be expressed in a similar manner by Eq. (2.27).

$$C_s = \frac{a_{33}}{\rho\pi r^2} \quad (2.27)$$

The vertical added mass for a cylinder crossing the free surface at high speed can be expressed as the high frequency limit. The sectional slamming force shown in Eq. (2.24) can therefore be expressed as shown in Eq. (2.28) [11].

$$f_s(t) = \frac{d}{dt}(m_a v_s) = m_a \dot{v}_s + v_s^2 \frac{d}{dh}(m_a) \quad (2.28)$$

Where:

m_a = Vertical added mass

\dot{v}_s = Slamming acceleration

3 Numerical Methods and Preliminary Analysis

In this chapter, frequency domain analysis of the feed barge is conducted to find the response amplitude operators and damping coefficients using software such as Wadam and GeniE. A preliminary study of the responses and environmental loads on a simple numerical fish farm model is also presented.

3.1 Numerical Tools - Wadam and OrcaFlex

Wadam is a software program developed by DNV GL used to perform hydrodynamic analysis for calculating wave-structure interaction in the frequency domain. The software is executed through Sesam HydroD, which is used for ballasting, hydrostatic, and hydrodynamic analysis [21]. The modeling of the environmental parameters for Wadam is done in HydroD, and the hydrodynamic loads are computed by Wadam using potential flow theory. In this thesis, Wadam is utilized to obtain hydrodynamic loads and responses of the feed barge in the frequency domain. The aim is to obtain response amplitude operators (RAOs), which can be used in time domain simulations for a fish farm system in OrcaFlex.

OrcaFlex is a software program developed by Orcina Ltd. and is used to perform static and dynamic analysis for risers, mooring systems, installation analysis for the oil and gas industry, renewable energy, defense, and several other industries. The software can simulate non-linear time domain finite element models and has many useful applications for offshore technology and numerical modeling. For this thesis, a complex fish cage model will be built using the OrcaFlex Application Programming Interface (OrcaFlex API) utilizing the Python programming language. The fish cage model developed using the OrcaFlex API will model the properties of a fish cage for properties such as drag coefficients, mass, solidity ratio, and other structural properties. In Section 3.2.1, the response amplitude operators and damping coefficients for a feed barge are established. Furthermore, in Section 3.3, a time domain analysis using OrcaFlex of a coarse fish cage model with the feed barge and the feeding tube is presented.

3.2 Hydrodynamic Analysis of Feed Barge

3.2.1 Frequency Domain Analysis of Feed Barge

To study systems in irregular sea states with linear characteristics, the frequency domain method is used to study the properties under different wave frequencies. The frequency domain is based

on linear solutions to the equations of motion. The previously mentioned irregular waves and superposition of regular, linear waves, is the fundamental basis for the frequency domain and linear dependency [9]. Using computer software such as Wadam, frequency domain analysis can be completed by calculating the vessel response in the frequency domain in six degrees of freedom to find response amplitude operators. A stationary sea state is used consisting of regular waves, where non-linear forces such as current or wind are assumed to be negligible. One of the reasons why this method is preferred is because it does not require significant amounts of computer power to process calculations to find the hydrodynamic response. The degree of accuracy for frequency domain analysis can be adjusted by optimizing the frequency set and the finite element model mesh of the vessel used. As discussed previously, the vessel is moored to the seabed. Hence, it is primarily heave, pitch, and roll directions that are of interest concerning the response of the feed barge.

In order to establish a numerical model for the feed barge to be used in time domain simulations of a fish farm, hydrodynamic analysis in the frequency domain must be performed. This can be completed by utilizing software packages such as GeniE and Wadam. After defining structural dimensions of a feed barge in GeniE, the hydrodynamic analysis can be performed in Wadam. The model that is established using GeniE is based on the finite element method (FEM) and generates a mesh of the submerged section of a feed barge, which can be imported to Wadam.

In order to move a fish farm facility offshore, it is clear that the barge is required to be of adequate size in order to operate in the environmental conditions of the North Sea. The numerical model will be based on the physical properties of the WaveMaster AC 850 feed vessel, which is manufactured and distributed by AKVA Group AS. The barge is among the larger feed vessels provided by AKVA Group and has a loading capacity of 850 tons, with a total of 16 silos and eight feeding lines to distribute feed from the barge to the fish cages. A figure of the feed barge is shown in Fig. 3.1.

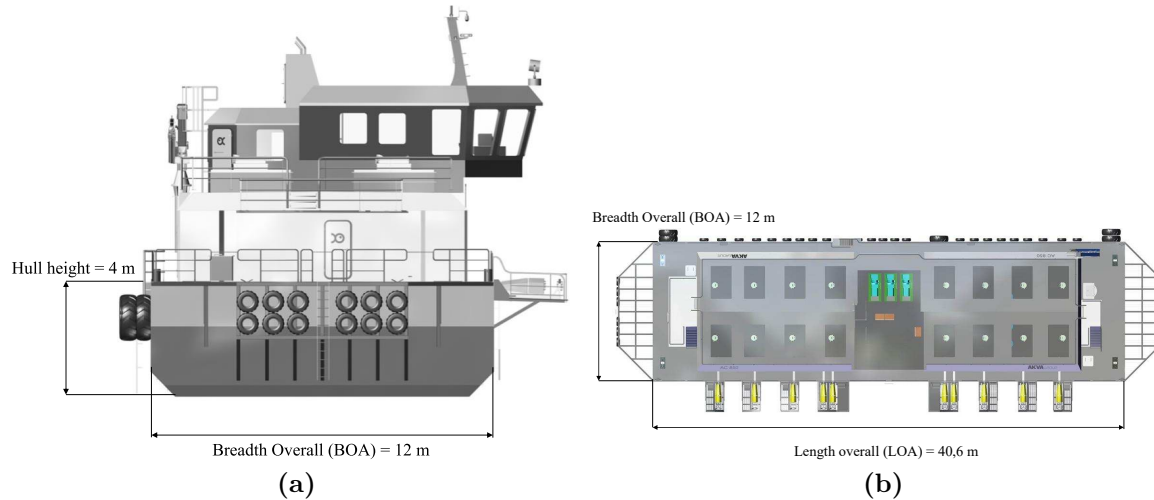


Figure 3.1: The WaveMaster AC 850 feed barge, side-view(a) and top view (b), which is used for fish farm operations [14].

Based on the dimensions and physical properties of the Wavemaster AC850 feed barge, a panel model is established in Genie, which is shown in Fig. 3.2. The panel most has an element mesh of 0.5 m, which is sufficient for frequency domain analysis for this vessel.

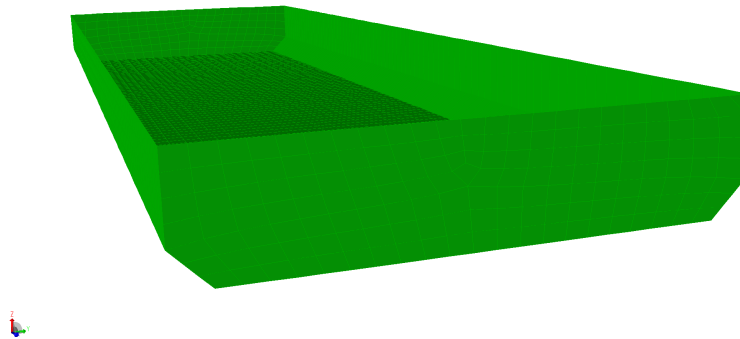


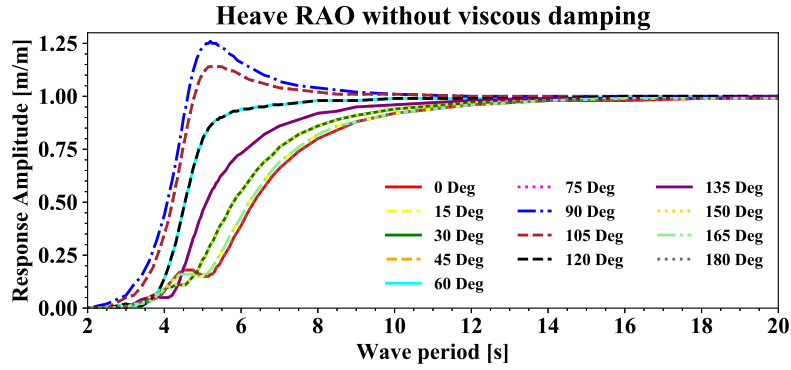
Figure 3.2: Panel model of the feed barge, Wavemaster AC 850, created in GeniE with 0.5 m mesh visible on the exterior panels.

As seen in Fig. 3.2, a panel model of the feed barge is shown for the submerged section of the feed barge. The mesh of the barge is set to 0.5 m elements, and simulations indicate that the mesh density is sufficient for the project. The panel model can be exported as a finite element model (FEM) to Wadam for hydrodynamic analysis for various parameters. By analyzing the response of the panel model for different periods and directions in Wadam, the hydrodynamic response can be found. The mass model for the barge is obtained in Wadam and listed in table 3.1.

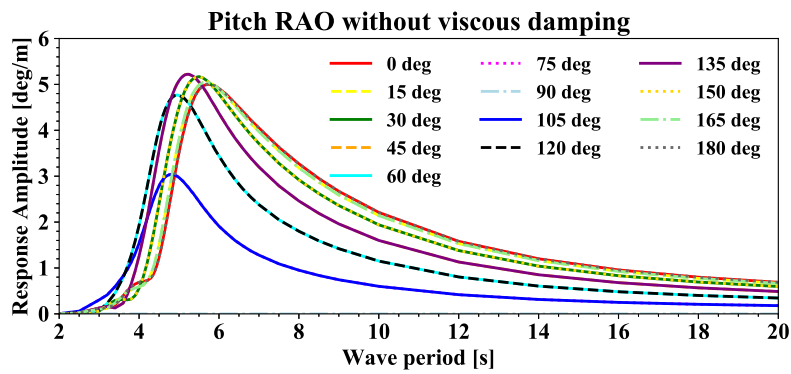
Table 3.1: Properties of the feed barge used for input in Wadam [14].

Buoyancy volume [m^3]	1245.61
Total mass [kg]	1276748.08
Center of buoyancy [m]	(9.86E-13, -3.04E-16, -1.28)
Center of gravity [m]	(9.86E-13, -3.04E-16, -0.58)
Radius of gyration, RX [m]	3.4813
Radius of gyration, RY [m]	11.7439
Radius of gyration, RZ [m]	12.2036
Draft [m]	2.61

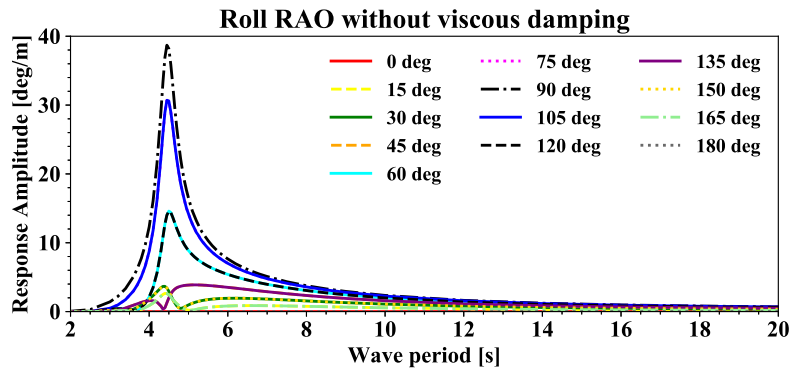
The simulations are run with wave directions between 0° and 180° degrees for 15° intervals, which means that there are 12 directions for each simulation of the feed barge. The simulations in Wadam are limited to 60 frequencies, and the simulations are run with wave periods from 0 to 20 seconds, with a larger concentration on data points between 4 and 6 seconds. The simulations are not expected to change significantly for wave periods of more than 20 seconds, as the barge would follow the heave motion of waves of that magnitude. Therefore, there are several data points from 0 to 4 seconds and from 6 to 8 seconds, but fewer points from 8 to 20 seconds. This is because the natural frequency of the barge is expected to be between 4 and 6 seconds. The heave, pitch, and roll response of the numerical model of the barge are shown in Fig. 3.3, where the peak of the figures indicates at what point in time the vessel has the largest response and the magnitude of the response.



(a)



(b)



(c)

Figure 3.3: Response amplitude operators of the feed barge without viscous damping effects in (a) heave, (b) pitch, and (c) roll for a range of directions varying from 0° to 180° , with 15° degree intervals.

From Figs. 3.3 (a), (b) and (c), the heave response of the feed barge peaks at 5.29 s and has a

maximum response of $1.20 \frac{m}{m}$ at a wave direction of 90° . The maximum response is largest at wave direction of 90° and the response is symmetric about this direction. Pitch response has a maximum response of 5.3° at $5.26 s$, for wave directions of 0° and 180° . The maximum response occurs at a direction of 0° and wave direction 180° , and the smallest response occurs at a direction of 90° . The roll response has a peak value at $4.55 s$, with a response of 40° , for waves at the direction of 90° . The maximum response for pitch has a similar distribution for the direction sets as for heave, with symmetric distribution about wave directions of 90° . A peak response in the area of $5.29 s$ is reasonable for a barge of this size, but the roll response magnitude of 40° seems to be large compared to other barges of similar dimensions or larger. A reason for this could be because viscous damping effects are not taken into account in this part of the analysis.

Table 3.2: Natural periods for the feed barge in heave, pitch, and roll.

Degree of freedom	Time [s]
Heave	5.29
Pitch	5.26
Roll	4.55

For a barge of this size, natural frequencies in heave, pitch, and roll are expected to be between 4 and 6 seconds. As shown by Magnuson [22], for larger barges compared to the feed barge in this analysis, a nominal roll period of 6 seconds are recorded. Some of the larger barges studied by Magnuson [22] had a nominal roll period of $10 s$. Furthermore, the maximum roll response of 40° seems to be large for a wave direction of 90° . The roll angles shown presented by Magnuson [22] was around 25° for a wave height $h_s = 3 m$. The following chapters include damping effects, to predict the response more accurately.

3.2.2 Sensitivity Study of Viscous Damping Effects

For a large barge, damping effects should be considered in order to predict the response of the vessel accurately. Potential damping and added mass is calculated in Wadam and included in the model. However, the viscous damping coefficient for the response of the barge in roll and pitch cannot be calculated but is determined through frequency domain analysis simulations and compared to empirical values [23]. In order to find the damping coefficients for the feed barge, several

simulations are run in Wadam. This is done in order to compare the responses of the feed barge to a barge of similar hydrodynamic properties. The damping coefficients used in Wadam are listed in Table 3.3.

Table 3.3: Six roll and pitch damping coefficients used in simulations selected to compare roll response values of the feed barge from Wadam to experimental values (the same coefficient is used for both roll and pitch).

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
$B_{44} \left[\frac{kg \cdot m^2}{s} \right]$	0	$1.0 \cdot 10^5$	$5.0 \cdot 10^5$	$1.0 \cdot 10^6$	$5.0 \cdot 10^6$	$1.0 \cdot 10^7$
$B_{55} \left[\frac{kg \cdot m^2}{s} \right]$	0	$1.0 \cdot 10^5$	$5.0 \cdot 10^5$	$1.0 \cdot 10^6$	$5.0 \cdot 10^6$	$1.0 \cdot 10^7$

Therefore, several simulations are run to compare the response of the barge to different roll and pitch damping coefficients. A series of simulations by adding different viscous roll damping coefficients for a wave direction of 90° are run in Wadam and can be seen in Fig. 3.4 .

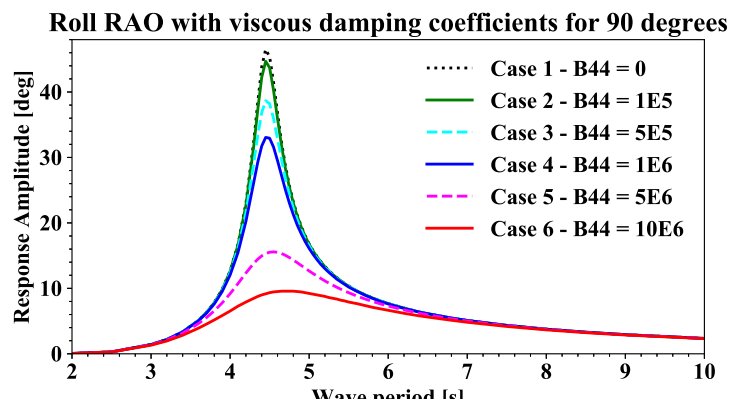


Figure 3.4: Roll response amplitude operators of the feed barge for six different viscous damping coefficients.

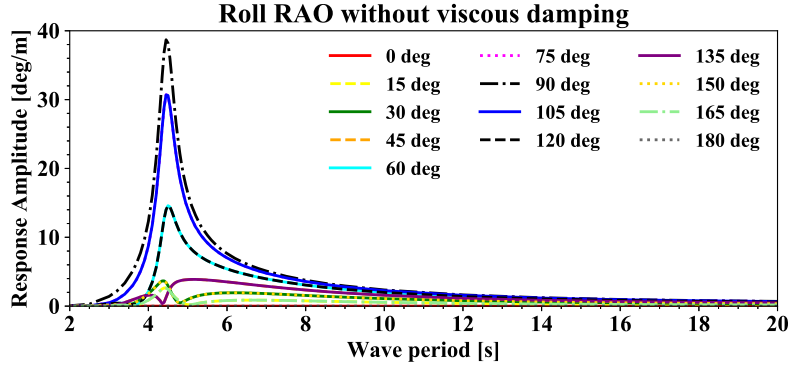
From the figure, it is an apparent reduction in the roll response when including viscous damping effects in the frequency domain analysis in Wadam. By comparing the response to the properties of empirical studies, a damping coefficient that is representative of the feed barge can be established. In Table 3.4, the properties of the feed barge, Wavemaster AC 850, is compared to that of a barge with similar properties from the reference [22]. From the table, it is clear that the length to breadth ratio $\frac{L}{B}$, draft, and roll period, is relatively similar. It should be mentioned that the keel to the center of gravity KG , the radius of gyration r_{xx} , and displacement, seem to be slightly different from the

feed barge.

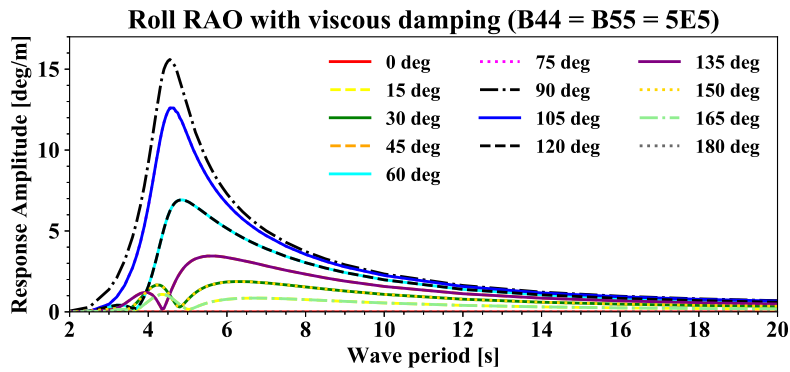
Table 3.4: Dimensions of a feed barge used for the operation of fish farms and a barge studied by reference [14, 22].

	Wavemaster AC 850	Barge 2
$\frac{L}{B}$	3.38	3.33
Draft [m]	2.61	2.74
KG [m]	2.03	9.76
r_{xx} [m]	3.48	15.79
Displacement [ton]	1271	6581
Roll period [s]	4.55	6

By comparing the roll damping coefficients of the two barges, empirical trials for barge 2 indicated a roll damping coefficient B_{44} , of $3.0 \cdot 10^5$ for $h_s = 3 m$ wave height [22]. By running a series of simulations of the feed barge and comparing the empirical results, it seems that the roll damping coefficient is most accurate for the damping coefficient in case 3, where the damping coefficient $B_{44} = B_{55} = 5.0 \cdot 10^5 \frac{kgm^2}{s}$. To find a more accurate roll damping coefficients, experiments of a barge with similar properties would have to be conducted.



(a)



(b)

Figure 3.5: Roll response comparison of feed barge, (a) without any damping effects and (b) with viscous damping coefficients, $B_{44} = B_{55} = 5.0 \cdot 10^5 \frac{kgm^2}{s}$.

The natural period of the roll response of the vessel does not change with the inclusion of viscous damping effects, but the peak response is reduced significantly. As can be seen in Fig. 3.5 (b), the damped roll response of the vessel is 16° at a direction of 90° . From the empirical studies of barge 2 mentioned in Table 3.4, the response of the barge had a natural roll period of around 6 seconds, with a peak roll response around 25° for an unknown direction, for a wave height $h_s = 3.00 m$ [22]. The roll response of the two barges is very different. However, barge 2 has a larger displacement, radius of gyration (r_{xx}) and a larger KG (distance from keel to center of gravity). Hence, it seems reasonable that the barge studied by Magnuson [22] has a smaller response than that of the feed barge.

3.3 Preliminary Analysis of the Fish Farm System

A fish cage will deform over time as a fluid with a given velocity passes through the net of the system. It is important to consider the deformation of the structure in the time domain. For time domain analysis, the hydrodynamic response of a system from a wave is calculated as it passes, for a given time step. The response is calculated for each time step and requires more computing power than frequency domain analysis. In this thesis, frequency domain analysis will be performed for the hydrodynamic response of the barge in Wadam, while the response of the fish cage will be a time domain analysis in OrcaFlex.

In the early stages of the thesis work, an analysis of a very coarse fish cage model was completed with an example model to assess the numerical modeling process and to get acquainted with Wadam and OrcaFlex. A fish cage with a large number of elements requires both more time, and an increasing amount of computer power, to conduct time domain simulations. It is, therefore, useful to study the behaviors of a simple numerical model, before running simulations for a comprehensive numerical fish farm model. Also, it was used as a way of considering the behavior of a simple fish cage model in the time domain for different environmental conditions. In the study, the volume deformation of the fish cage, and the effective tension for the mooring system was considered.

However, it is important to stress the fact that this coarse fish cage model has several erroneous properties with regards to drag coefficients, net equivalence, net solidity, mass, and mooring system. The results can therefore not be validated, as a new fish cage model will have to be developed using the OrcaFlex API based on the programming language Python, where these properties are adjusted to that of a real fish cage. The development of this model is discussed in detail in Section 4.1.

In the study, the feed barge model that was established in Wadam is imported into OrcaFlex with the hydrodynamic properties defined previously. This means that the response of the feed barge will be similar to that of a real feed barge for time domain simulations under various environmental conditions. The feed barge, feeding tube, and anchor line system can be seen in Fig. 3.6.

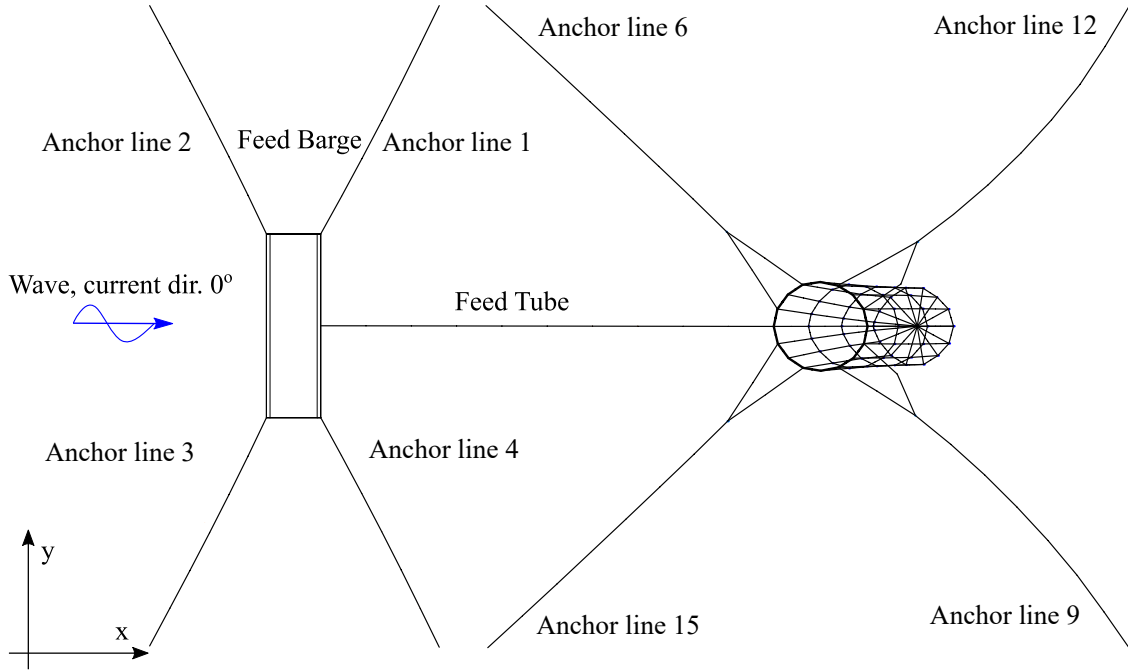


Figure 3.6: Displacement of fish farm system with mooring lines, barge and feeding tube for current velocity $U_c = 1.50 \frac{m}{s}$, with a wave and current direction of 0° .

After importing hydrodynamic coefficients obtained in Wadam to the vessel model in OrcaFlex, simulations based on irregular wave theory have been conducted. The environmental parameters used to analyze the responses in the preliminary study is shown in Table 3.5.

Table 3.5: Environmental parameters used in the preliminary study of coarse fish farm system in OrcaFlex.

Environmental conditions	$h_s [m]$	$t_p [s]$	$U_C [\frac{m}{s}]$
Condition 1	2.00	5.00	0.50
Condition 2	2.00	5.00	1.00
Condition 3	2.00	5.00	1.50

Hence, the simulation is performed under a significant wave height $h_s = 2 m$ and zero up-crossing period $t_z = 5 s$. Similar to the simulations for the fish cage, three current velocities $U_c = 0.50 \frac{m}{s}$, $U_c = 1.00 \frac{m}{s}$ and $U_c = 1.50 \frac{m}{s}$ are used for the simulations. The wave and current directions are set to 0° relative to the x-axis of the coupled system. To connect the feed barge to the fish cage,

a feeding tube has been added between the two, with an outer diameter of $D_o = 110 \text{ mm}$, inner radius of $r_i = 97.44 \text{ mm}$ and a length of $L = 100 \text{ m}$ and filled with seawater. The drag coefficient of the feeding tube has been set to $C_d = 1.00$ and the added mass coefficient $C_a = 1.00$. The duration of the simulation consisted of a ramp up period of 300 s , before a 1000 s simulation period with a time step of 0.05 s . In the preliminary study, slamming loads on the feeding tube has not been considered. The tension of the anchor lines and feeding tube has been plotted for the duration of the simulation and gives an indication of the axial force in the fish farm system. The effective tensions are shown in the Figs. 3.7, 3.8 and 3.9. As the mooring lines for the feed barge is utilizing a conventional mooring line setup, the focus is on the anchor lines for the fish cage, as they are good indicators for the environmental loading on the fish cage. Anchor lines are mooring lines that are anchored to the seabed.

The Figs. 3.7, 3.8, and 3.9, show the tensions in the anchor lines of the coarse numerical model. It is clearly seen that the tensions in the anchor lines oscillate significantly and seem to vary a considerable amount from the beginning compared to the end of the simulations. Furthermore, there appears to be a reduction in the magnitude of the oscillations for larger current velocities. Also, it is clear that with the orientation of the fish cage, anchor lines 15 and 6 are exposed to the largest tension for the fish cage. The tension is distributed more evenly compared to the simulations of the fish cage, as the direction of the waves distributes the wave load on the two anchor lines more evenly. In addition, the tensions in anchor lines 15 and 6 decrease slightly over time and are small compared to the tensions of mooring lines 9 and 12. The tension on the feeding tube is larger than the anchor lines, and it is uncertain if a feeding tube would be able to sustain such loading over a prolonged period. A reason for this could be that the feeding tube is being stretched between the feed barge and fish cage, in which case the feeding tube should be extended to let the anchor lines absorb the displacement of the fish cage. The maximum effective tensions of the mooring line and feeding line are listed in Table 3.6.

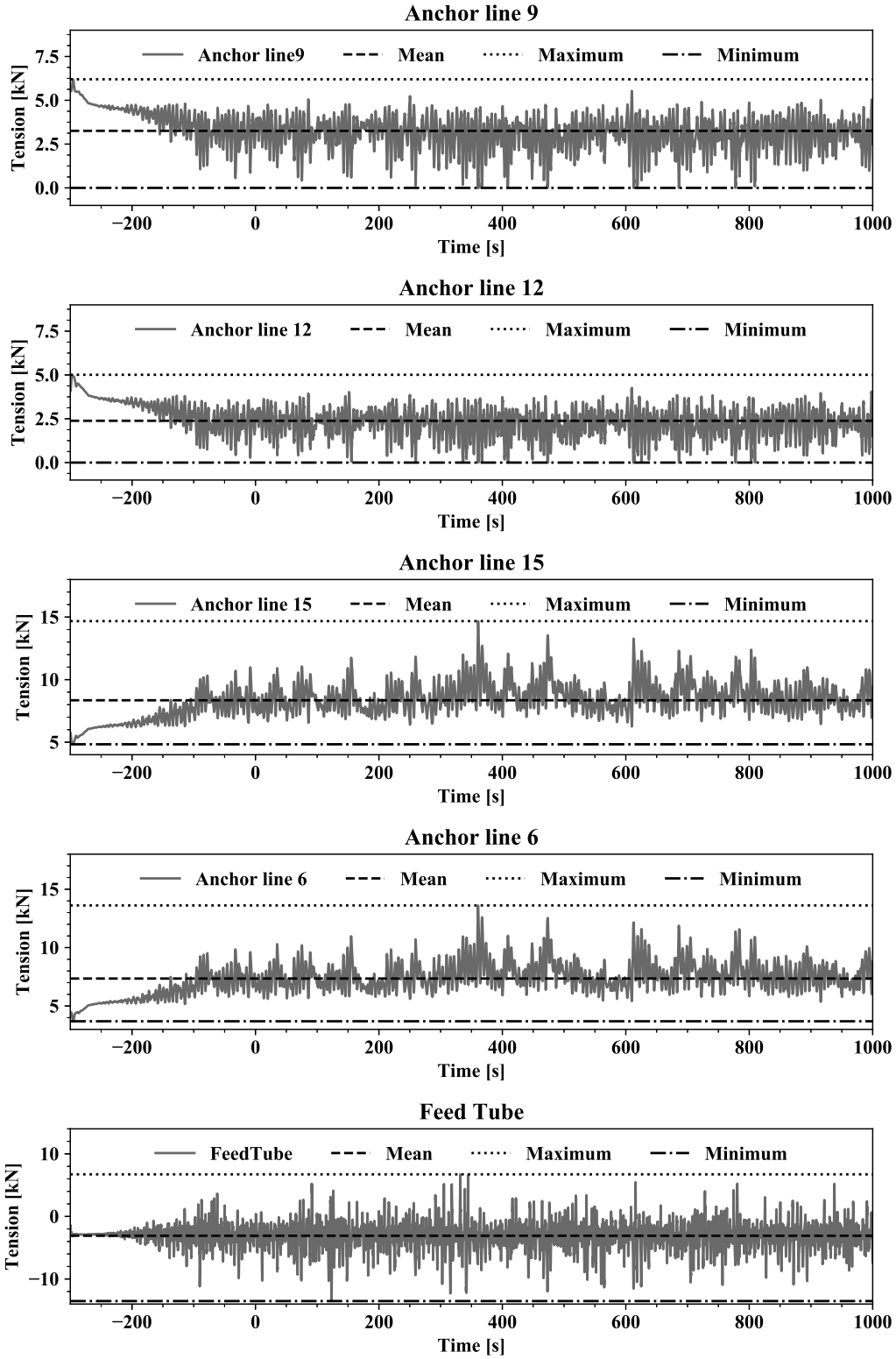


Figure 3.7: Effective tensions in anchor lines 9, 12, 15, 6 and feeding tube. Current velocity $U_c = 0.50 \frac{m}{s}$ and a wave and current direction of 0° for condition 1.

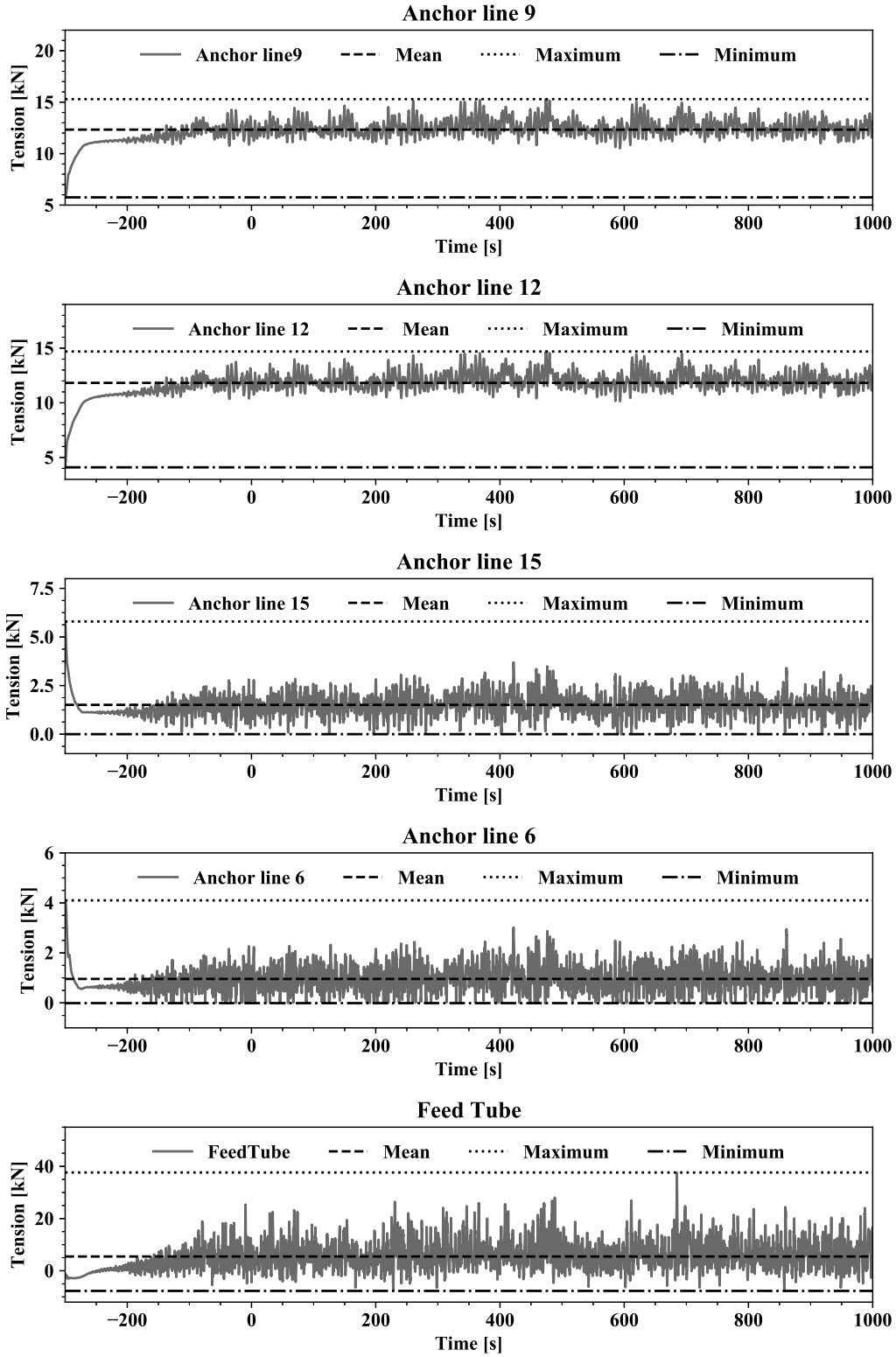


Figure 3.8: Effective tensions in anchor lines 9, 12, 15, 6 and feeding tube. Current velocity $U_c = 1.00 \frac{m}{s}$ and a wave and current direction of 0° for condition 2.

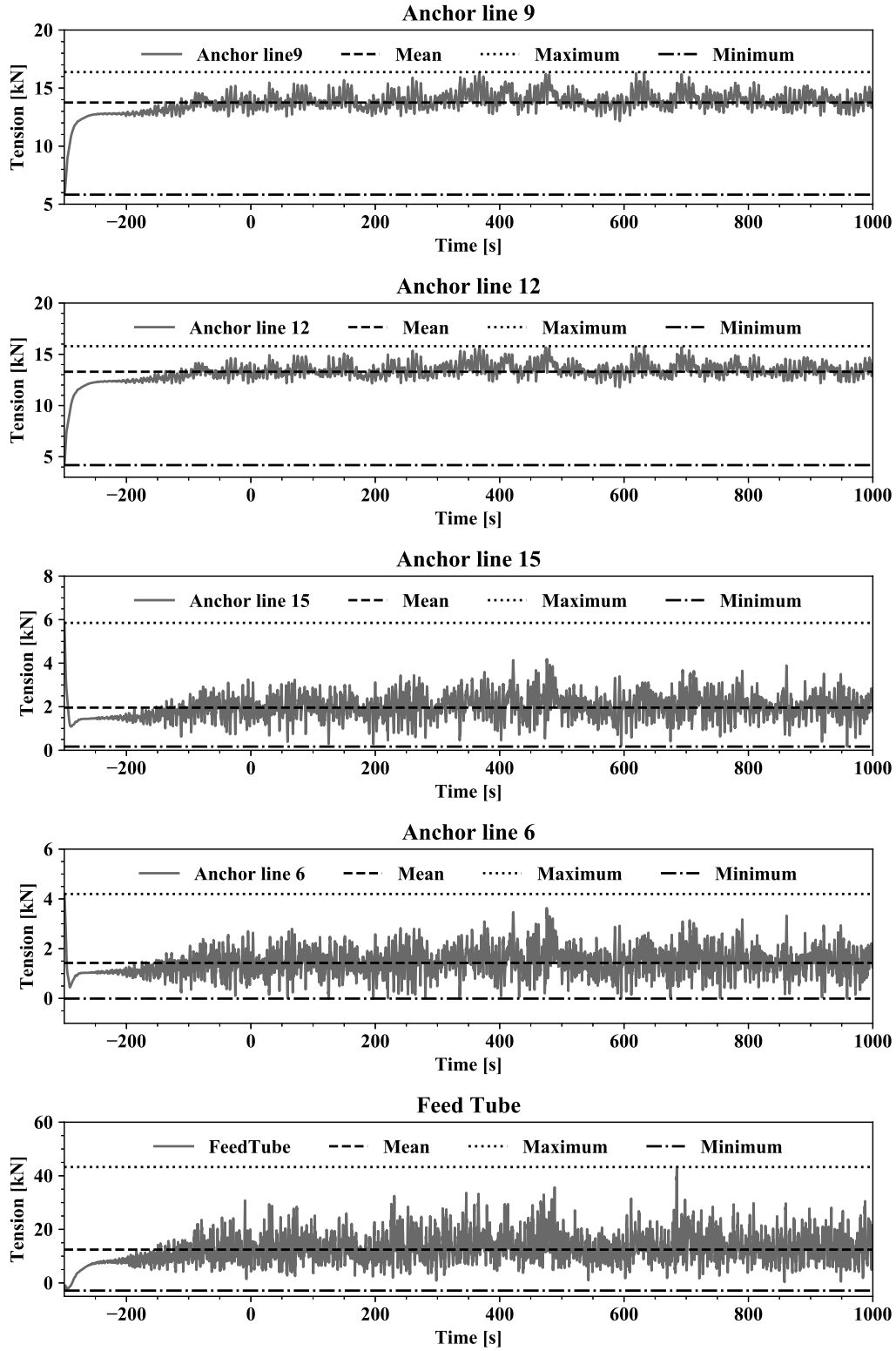


Figure 3.9: Effective tensions in anchor lines 9, 12, 15, 6 and feeding tube. Current velocity $U_c = 1.50 \frac{m}{s}$ and a wave and current direction of 0° for condition 3.

Table 3.6: Effective tension for fish cage anchor lines of the combined fish cage and barge simulations at a direction of 0° relative to the x-axis.

	$U_c = 0.50 \left[\frac{m}{s} \right]$	$U_c = 1.00 \left[\frac{m}{s} \right]$	$U_c = 1.50 \left[\frac{m}{s} \right]$
Anchor line 9 [kN]	12.30	15.20	16.50
Anchor line 12 [kN]	13.60	14.80	15.80
Anchor line 15 [kN]	3.50	3.10	3.60
Anchor line 6 [kN]	4.60	3.60	4.30
Feeding Tube [kN]	30.10	40.10	46.00

The tensions in the anchor lines for the barge is found to be minor compared to those of the fish cage. For irregular waves at 0° relative to the x-axis of the combined system, the largest effective tensions of anchor lines 9, 12, 15 and 6 are in the region $12.2 kN$ to $13.6 kN$, which is similar to that of the fish cage. The difference in the maximum effective tensions could be because the differences are only slight differences in peak values. One reason for this could be that the feeding tube is absorbing some of the tensions or that the tensions are spread more evenly on all the anchor lines of the combined barge and cage installation. Another reason for this could be that the feed barge is blocking some of the waves, effectively reducing the anchor lines on the fish cage. The maximum effective tensions in the feeding tube seems to be excessive at $30 kN$ for a current velocity $U_c = 0.50 \frac{m}{s}$ and $46 kN$ for a current velocity of $U_c = 1.50 \frac{m}{s}$, nearly three times higher than the maximum effective tensions in the anchor lines.

The fish cage model used for the preliminary study has several limitations, and the essential properties that are used to model the behavior of a real fish cage are not accurate. The results from the preliminary study are only used for a preliminary assessment of the behavior and responses of a numerical fish farm model in OrcaFlex. To establish a model with properties that replicate the responses of a full-scale fish cage, it is necessary to take into account the loading mechanisms and behaviors discussed in the Chapter 2. Important properties of a fish cage such as drag coefficient, net solidity, mass, axial stiffness, in addition to structural elements such as feeding tube and anchor lines will be expanded upon in Section 4.1. For the analysis in Chapter 4, slamming loads will be included for the feeding tube, and the mooring system will be configured according to a conventional mooring system. A more comprehensive analysis of a numerical fish cage model will be presented, and a numerical fish farm model established.

4 Methodology and Modeling Technique

In this chapter, the process of developing a numerical model of a fish cage, net, mooring system, and feeding tubes are introduced. The equivalent properties between the physical nets and the net of the numerical model are established. Furthermore, an overview of the OrcaFlex Application Programming Interface (API), used in conjunction with the Python programming language, is also presented. The numerical model is considered as a higher fidelity model compared to the preliminary model presented in Section 3.3.

4.1 Numerical Modeling of the Fish Farm System

In order to efficiently establish a numerical model of a fish cage to be used in time domain simulations, Orcina has provided an Application Programming Interface (API) extension to the OrcaFlex software. Although the design process is made simpler by the API, the design process is relatively laborious and requires a fundamental understanding of the open source programming language Python. Python is an object-oriented programming language that offers several building blocks that can be used as tools in developing a numerical model of a fish cage. The first step in the design process is to establish fundamental structural properties of the fish cage and subsequently find equivalent net properties for the numerical model. The fish cage model that was used as an outline for the fish cage model is a cylindrical cage with a floating collar, cylindrically shaped top section with a sinker tube attached, a conically shaped bottom section with a mass attached to the bottom. A cylindrical fish cage, similar to the numerical model, is shown in Fig. 4.1. The sinker tube and mass attached to the bottom stretch the cage vertically and prevent the deformation of the cage when exposed to current and waves.



Figure 4.1: A cylindrical fish cage with a conical bottom section, manufactured by Egersund Net AS, here seen undergoing steady current tests [24].

4.1.1 Numerical Modeling Using OrcaFlex API and Python

Three numerical models of varying net mesh density will be established in this section, and the modeling process will be explained. The three models will be used in a convergence study, which can be compared to empirical model tests to identify discrepancies in the numerical models compared to the model tests. In the OrcaFlex API, several elements will be defined to generate the necessary elements for building the numerical fish cage models. These elements are defined in OrcaFlex as buoys, knots, and lines, which all have defined mass, volume, geometry, stiffness, length, and position relevant to its purpose in the model (hence buoys do not have a defined stiffness and lines does not have a defined position). When calling upon the function of each element in the Python code, the element is generated with properties according to specifications in the given command. In the following sections, a few of the basic element functions used to create the numerical model are presented, and their purpose explained. The elements of the fish cage and their connections are shown in Fig. 4.2, which comprise one section of a fish cage. A fish cage section can be combined to make a full circular fish cage.

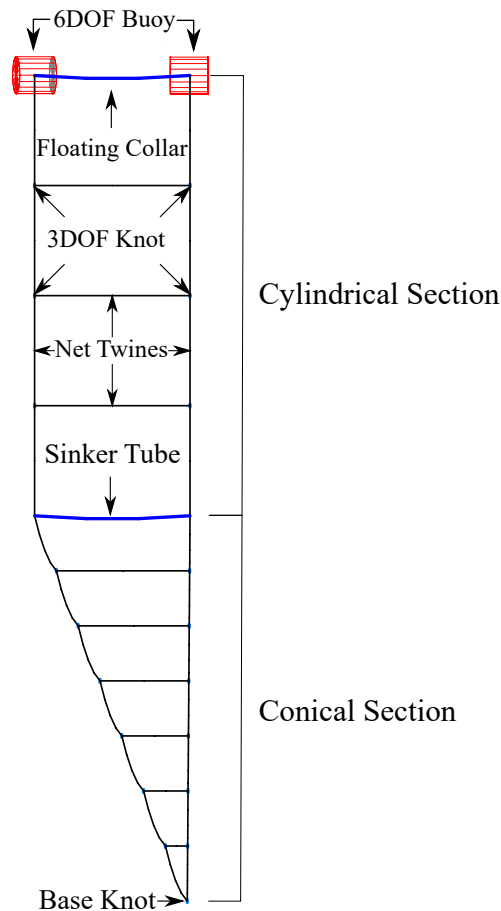


Figure 4.2: Elements added using the OrcaFlex API and their connections for a fish cage.

The six degrees of freedom buoy (6DOF buoy) is used as an element to connect the lines of the floating collar, sinker tube, and mooring lines, in addition to providing buoyancy as buoys for the mooring line around the cage. The motion of the buoys consider translational and rotational motion and are well suited to model complex behavior in wave and current conditions. When the “createbuoy” function is called upon, the elements are given a variable name and a position in three dimensions, as can be seen in the code.

```
def createbuoy(pos, name, x, y, z):
    float = model.CreateObject(OrcFxAPI.ot6DBuoy, name+str(pos))
    float.BuoyType = "Spar Buoy"
    float.DegreesOfFreedomInStatics = 'All'
    newX, newY = rotateposition(x,y)
    float.InitialX, float.InitialY, float.InitialZ = newX, newY, z
    float.InitialRotation1 = 90
    float.InitialRotation2 = -180 + pos*math.degrees(angle)
    float.InitialRotation3 = 0
    float.mass = 0.1
    float.MassMomentOfInertiaX = 0.058
    float.MassMomentOfInertiaY = 0.058
    float.MassMomentOfInertiaZ = 0.025
    float.CylinderOuterDiameter[0] = 1.0
    float.CylinderInnerDiameter[0] = 0.0
    float.CylinderLength[0] = 1.0
    float.StackBaseCentreZ = -0.5
    float.CenterOfMassX = 0
    float.CenterOfMassY = 0
    float.CenterOfMassZ = 0
```

The function “createbuoy” generates a buoy element with the following properties:

- Variable name, dimensions, orientation, and position in three dimensions.
- Buoy type (e.g. 6DOF Spar Buoy)
- Mass, moments of inertia, and center of mass are specified.

To establish a net panel, knots are used to connect every line element as the knots have similar properties to the six degrees of freedom buoy, but are only used to interpret translational motion in three degrees of freedom (3DOF). The knots have small mass, volume, and height, which is ideal for representing the net knots of a fish cage.


```

def createknot(pos, name, x, y, z): #creates net knots
    knot = model.CreateObject(OrcFxAPI.ot3DBuoy, name+str(pos))
    newX, newY = rotateposition(x,y)
    knot.InitialX, knot.InitialY, knot.InitialZ = newX, newY, z
    knot.Mass, knot.Volume, knot.Height = 0.0001, 0.0001, 0.01

```

The function “createknot” generates a knot element with the following properties:

- Variable name, and position in three dimensions.
- Mass, volume and height are specified.

The lines used in the floating collar connects multiple buoys and has a high axial and bending stiffness. The floating collar has rigid connections, which are defined by the expression “OrcFxAPI.OrcinaInfinity()”. The lengths of the lines are determined in the function, a slight bend radius and the orientation of the end connections are predetermined. When calling the function, the variable is given a name, connection points, length, segment length, and the orientation of the ends are given. The segment length is the number of nodes, or segments, that are used to complete one line element.

```

def createtoprings(sections, name, connA, connB, OAX, OAY, OAZ, OBX, OBY, OBZ):
    line = model.CreateObject(OrcFxAPI.otLine, name)
    line.LineType[0], line.Length[0] = "Ring line type", Ring1
    line.TargetSegmentLength[0], line.PreBendY[0] = 1.6, 0.04
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
    line.EndAAzimuth, line.EndADeclination, line.EndAGamma = OAX, OAY, OAZ
    line.EndBAzimuth, line.EndBDeclination, line.EndBGamma = OBX, OBY, OBZ
    line.IncludeTorsion = 'Yes'
    line.EndATwistingStiffness = OrcFxAPI.OrcinaInfinity()
    line.EndBTwistingStiffness = OrcFxAPI.OrcinaInfinity()
    line.EndAxBendingStiffness = OrcFxAPI.OrcinaInfinity()
    line.EndBxBendingStiffness = OrcFxAPI.OrcinaInfinity()
    line.EndAyBendingStiffness = OrcFxAPI.OrcinaInfinity()
    line.EndByBendingStiffness = OrcFxAPI.OrcinaInfinity()

```

The function “createtopring” generates a line element with the following properties:

- Variable name, line type, length, number of elements in the line, and pre-bend radius.
- End connections, offset and orientation for the line is set as unknown variables and are specified when the function is called.
- Connection torsion and bending stiffness is set as rigid using the “OrcFxAPI.OrcinaInfinity()” command. The line element should not bend at the end connections, but rather along the line length because of material properties.

Every knot is connected by line elements, which are essentially the twines for the net. The twines have a defined length, a given number of line elements, and specified end connections with orientation, as is shown in the following code.

```
def createneta(pos, name, connA, connB, offset):  
    line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))  
    line.LineType[0] = "Net line type"  
    line.Length[0] = - Spacing2  
    line.TargetSegmentLength[0] = (- Spacing2)  
    line.IncludeSeabedFrictionInStatics = 'No'  
    line.EndAConnection = connA  
    line.EndAX, line.EndAY, line.EndAZ = 0, offset, 0  
    line.EndBConnection = connB  
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
```

The function “createneta” generates a line element with the following properties:

- Variable name, line type, length, number of elements in the line.
- End connections, offset, and orientation for the line is set as unknown variables and are specified when the function is called.

In order to build the net, several functions are executed to generate a mesh section, which consisting of multiple variations of buoys, lines, and knots. By iterating this process multiple times, one section of the fish cage is generated. The way the code builds one section of the fish cage is by first generating two buoys, which is connected by the stiff and slightly curved top collar. Several knots are subsequently generated from left to right, top to bottom, generating the grid for which the lines

will be connected. The lines are then added to connect the knots, and a sinker tube is connected to the knots at the bottom of the cylindrical section of the fish cage, completing one section of the fish cage model. A section of the fish cage can be seen in Fig. 4.4. By iterating the sections multiple times, a complete circular fish cage is generated. An example of simple net panels that are combined to form multiple sections is shown in Fig. 4.3, where one small net section for the cylindrical portion of the fish cage can be expanded into multiple sections.

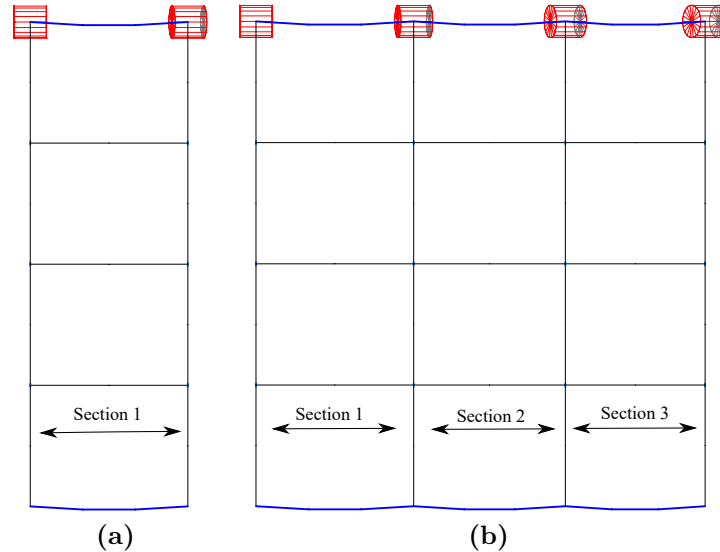


Figure 4.3: The angle between two buoys (red) on the top, connecting the floating collars, from the center point of the net is used by Python to iteratively connect on section (a) to multiple sections (b), constructing a circular fish cage by generating multiple sections in a circular fashion.

In order to construct the full model in an iterative process, a mathematical expression had to be established to retrieve the coordinates of each element and to make a fully circular fish cage. The mathematical expression is developed to return each position to the function, and then calculate the next position based on the section angle and position. The angle is defined as the angle between the first and the second buoy element (indicated as “6DOF Buoy” in Fig.4.2) and the center of the fish cage and can be said to form one section of the fish cage.

```
def rotateposition(oldX, oldY):
    newX = oldX * math.cos(angle*pos) - oldY * math.sin(angle*pos)
    newY = oldX * math.sin(angle*pos) + oldY * math.cos(angle*pos)
    return newX, newY
```

The function “rotateposition” executes the following process:

- Retrieves the old x- and y-coordinate positions of the element in relation to the global coordinate system
- A new coordinate position for the element is then calculated in x- and y-directions, based on the angle of a section in the fish cage shown in Fig. 4.4.

In Fig. 4.4, one section of the fish cage is shown, which will be rotated according to the number of sections designated to that specific numerical model. The iterative code retrieves the coordinates with respect to the angle, which is determined by the number of sections in the numerical model.

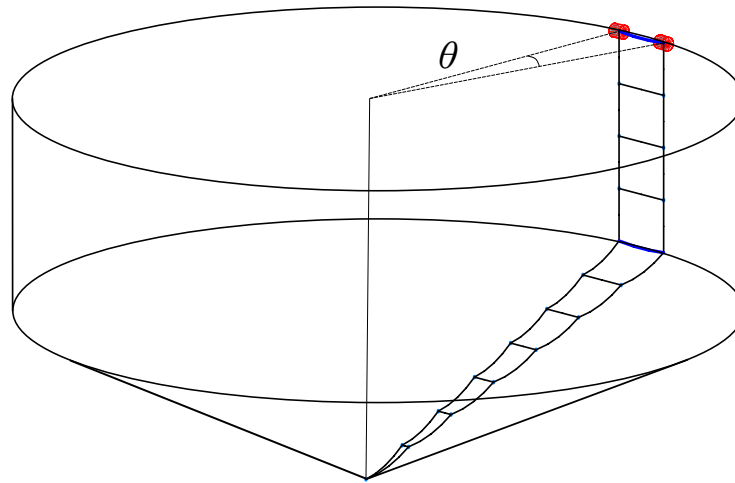


Figure 4.4: One section for a fish cage model intended to be iterated in a circle using the reiterative code to establish a cylindrical numerical fish cage model.

The expression is useful for generating a numerical model with different mesh density, as it is merely an expression of the angle of the fish cage and multiples of 360 degrees. Hence, the numerical models can be built based on sections of 8, 16, 24, 32, 48, 64 net panel sections, depending on the desired mesh density. The numerical models are designed to have evenly spaced net panels of 4 meters, 3 meters, and 2 meters between knots. Furthermore, it is decided to make numerical fish cage models with 32 sections for the coarse net fish cage model. The fish cage model with medium net mesh resolution has been built using 48 sections, and the fine net mesh model has been established using 64 sections for the fine numerical models of the fish cage. The angle for each section is found using the expression where the angle, $\theta = \frac{360}{No. Sections}$. The angle between sections for the coarse 32 section model is found to be 11.25° . The angle of the medium 48 section model is found to be 7.5° , and that of the fine 64 section model is then 5.625° .

The entire code used to build the 64 section fish cage model is shown in Appendix A.1. However, when calling the codes to build the fish cage models, initially the knots of the section are established

for the various coordinates and positions in relation to the center of the cage. After establishing a variable name for the floating collar, buoys and knot elements, the sinker tube is implemented to connect the bottom of the cylindrical portion of the fish cage model. The lines modeling the twines are then added vertically to tie the floating collar, buoys, and knots together by executing a for-loop. After the vertical lines are established from the top buoys to the base knot at the bottom, horizontal lines are added, connecting the knots of each level, from the floating collar to the base knot. The iterating process of building the net is shown in Fig. 4.5, where several sections are connected and form the shape of a 16 section numerical fish cage model with 11 levels, with a total height of 28 meters.

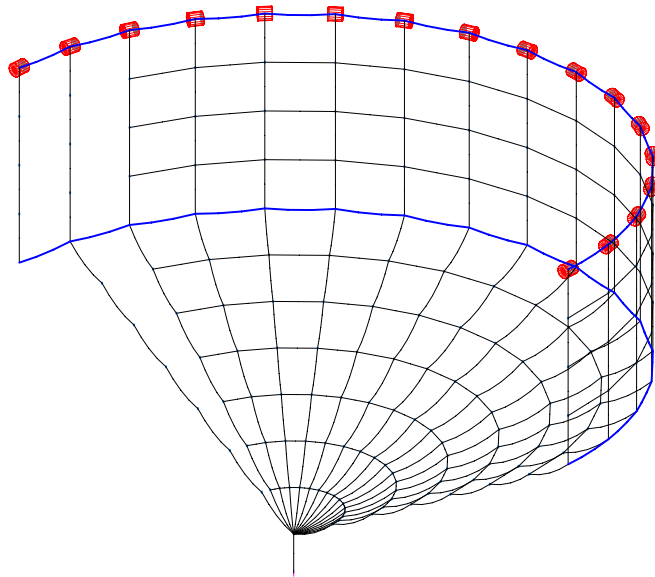


Figure 4.5: Net panel sections built iteratively and connected to form a net mesh, is used to build a numerical cylindrical fish cage model with a conical bottom section.

After executing the python script and establishing the entire fish cage model, it should include a floating collar line, sinker tube, buoys, knots, and a bottom mass element. These are the essential elements to form the fish cage structure, which can then be connected to a feed barge, feeding tube, and mooring lines. The numerical models are presented in the following figures, showing the 32 section coarse fish cage model in Fig. 4.6, the 48 section medium mesh numerical fish cage model in Fig. 4.7, and the 64 section fine mesh fish cage model is shown in Fig. 4.8.

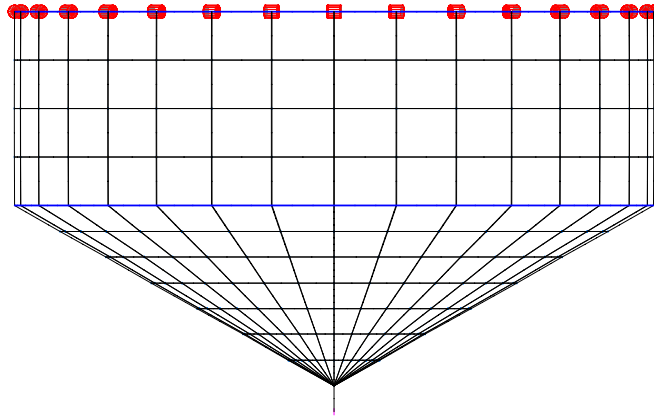


Figure 4.6: Side view of the numerical fish cage model with a coarse net mesh quality, with 32 sections and 4 meters spacing between knots.

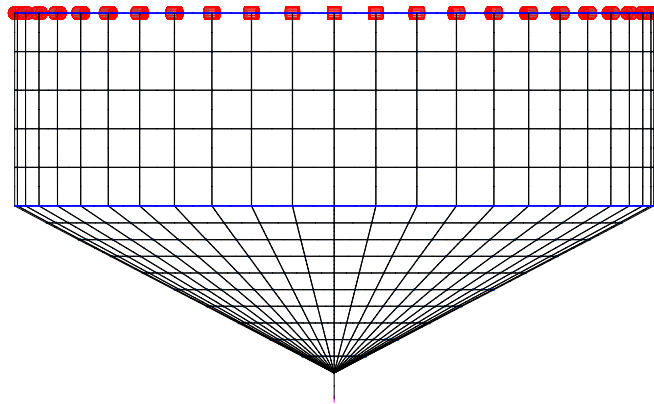


Figure 4.7: Side view of the numerical fish cage model with a medium net mesh quality, with 48 sections and 3 meters spacing between knots.

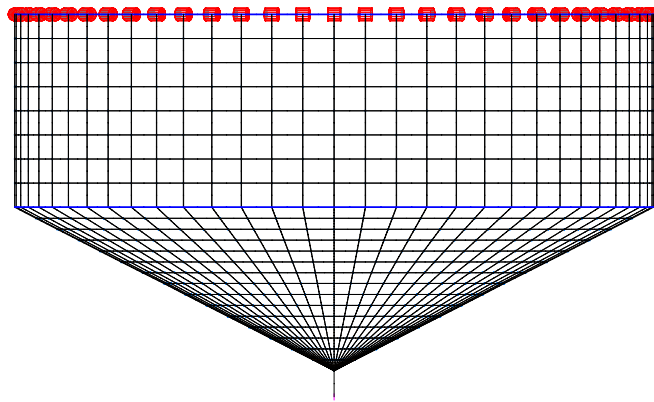


Figure 4.8: Side view of the numerical fish cage model with a fine net mesh quality, with 64 sections and 2 meters spacing between knots.

In the figures of the numerical models, the reduction in spacing for the square mesh in the net models is apparent. As the mesh density goes from large spacing to small spacing, the mesh is designated as having coarse, medium, and fine mesh density for the 32, 48, and 64 section models respectively. Furthermore, there is an increase in the number of levels, or an increase in the number of horizontal twines along with the height of the fish cage, for each numerical model. The increase in the number of levels is to compensate for a reduction in the width of the square panels as the number of sections increases to maintain a symmetric square net panel. An overview of the three fish cage models is presented in Fig. 4.9, to illustrate the increased number of net twines for the three numerical models.

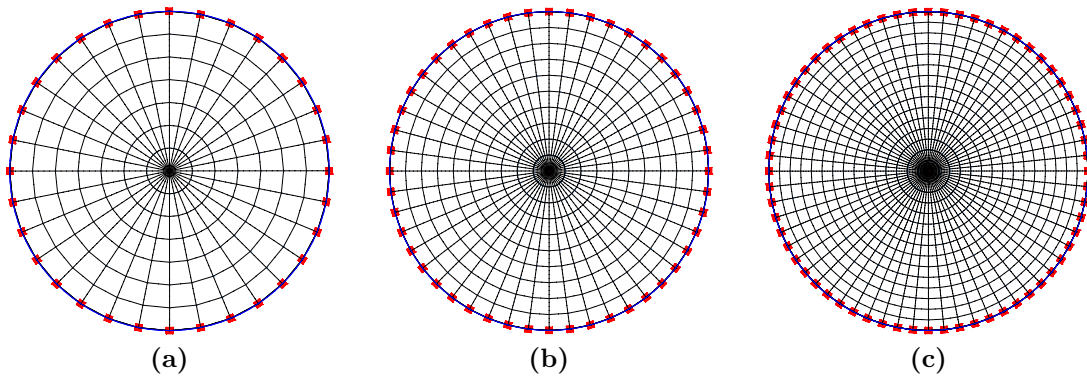


Figure 4.9: An overview of the numerical fish cage models with 32 section (a), 48 sections (b), and 64 sections (c).

4.1.2 Net Equivalence

In the development of the numerical fish cage model, it is clear that the number of twines for the net cannot be as numerous as for the full-scale fish cage model used for fish farming. Simulations for a full-scale net would demand tremendous computer power; hence, a simplified model is established by utilizing the equivalent values of a full-scale fish cage, which is shown in this section. The numerical model properties have to replicate the response and behavior of a full-scale fish cage. The numerical model of the fish cage in this thesis is based on a cylindrical fish cage concept developed by Aqualine AS [26]. The dimensions of a full-scale fish cage are presented in Table 4.1.

Table 4.1: Full-scale cylindrical fish cage developed by Aqualine AS [26].

	Fish Cage	Units
Cage Height	28.00	m
Diameter	50.00	m
Floater		
Outer Diameter	0.50	m
Inner Diameter	0.44	m
Mass / Length	80.00	$\frac{kg}{m}$
Net		
Diameter	0.0025	$\frac{kg}{m}$
Mass / Length	0.006	$\frac{kg}{m}$
Spacing	0.025	m
Solidity	0.20	
Drag coefficient, C_D	1.20	
Axial stiffness	4.42	kN
Sinker Tube		
Outer Diameter	0.50	m
Mass / Length	196.00	$\frac{kg}{m}$
Bottom Mass	1000	kg

For the numerical model, all the properties for the floating collar, bottom collar, and bottom mass is modeled in the same way as the full-scale fish cage, but the net properties are more challenging to replicate. Hence, by applying Morison theory and hydrodynamic properties, discussed in Section 2.3.2, the equivalence properties such as diameter, twine mass, drag coefficient, and axial stiffness can be established by equating buoyancy and hydrodynamic forces. These properties are replicated as the fish cage net mesh can be represented by fewer twines, but with a corresponding change in drag coefficient, drag, and mass [27]. The net can be defined as being constructed by a large combination of knots and twines where a knot and connecting twines represent many smaller twines and knots, as is shown in Fig. 4.10.

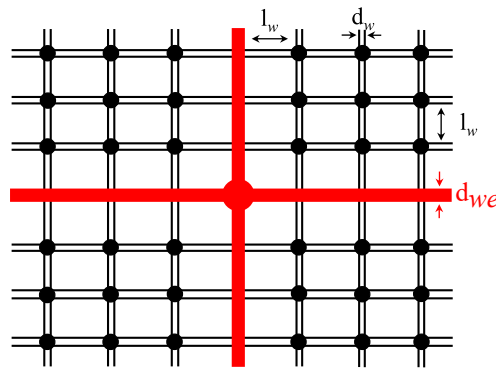


Figure 4.10: Net equivalence (red) of net twines (black) representing the net of the numerical model compared to a full-scale fish cage. A few red twines is laid to intersect many black lines.

The numerical model in OrcaFlex is based on a few fundamental mesh properties of the net. These properties are the diameter, mass per unit length, drag coefficient, and axial stiffness of the twine. In order to develop a model that ensures reliable results, three numerical models have been established. These three models consist of 4 meters, 3 meters, and 2 meters spacing between knots in the net mesh. These properties are established in the following paragraphs, but first, it is necessary to quantify the number of twines per net panel, similar to what is shown in Fig.4.10 for the coarse, medium and fine net models. To find the number of twines per full-scale net panel, the relation where the number of twines in the numerical model is given by the number of twines in the net panel for the fish cage compared to the numerical model, $n_{Twines} = \frac{Model\ spacing}{Twine\ spacing}$. The net properties such as the twine spacing are given in Table 4.1, which is used to find the spacing of the net twines in the different numerical models, shown in Table 4.2. The table also shows the number of levels, sections, twines, and knots, which indicates the increasing number of elements in the models, which also contribute to significantly increasing computational time for time domain simulations of the fish farm model. The net spacing and number of twines are for every section, while the number of

levels, sections, and knots are for the fish cage models of coarse, medium, and fine mesh densities.

Table 4.2: Twine discretization for the coarse, medium and fine numerical fish cage models.

Model	Net spacing [m]	No. of twines	No. Levels	No. of sections	No. of knots
Coarse	4.00	$\frac{4.00}{0.025} = 160$	11	32	353
Medium	3.00	$\frac{3.00}{0.025} = 120$	15	48	721
Fine	2.00	$\frac{2.00}{0.025} = 80$	23	64	1471

Diameter Equivalence

The equivalent diameter is found by equating the buoyancy of n twines to that of the equivalent net screen, which is given by the relation in Eq. (4.1)[27].

$$\sum_{i=1}^n F_b = F_{be} \quad (4.1)$$

Furthermore, the expression states that the buoyancy F_b , of the twines in the numerical models, are equal to that of the equivalent buoyancy force F_{be} . The buoyancy force is known to be expressed as $F = \rho g V$, where the equivalent volume for the twines V_e , can be found by the expression $\sum_{i=1}^n V = V_e$. The volume of a twine has the geometry of a cylinder, hence $V = \frac{\pi D^2}{4}$, resulting in the following relation, $n(\frac{\pi D^2}{4}) = (\frac{\pi D_e^2}{4})$. The equivalent diameter can, therefore, be expressed by Eq. (4.2).

$$D_e = \sqrt{n \cdot D^2} \quad (4.2)$$

Twine Mass Equivalence

The mass of the net twines for a full-scale fish cage is expressed as mass per unit length in OrcaFlex. Thus, for n twines per net panel with density $\rho = 1125 \frac{kg}{m^3}$, the equivalent mass per unit length m_e can be found by Eq. (4.3) where D_e is the equivalent diameter of the twine.

$$m_e = \rho \cdot \frac{\pi}{4} \cdot D_e^2 \quad (4.3)$$

Drag and Mass Force Equivalence

To find the drag and mass force of equivalent twine, given n twines per net panel, Morison theory is applied, where drag and mass forces are considered. The relations are based on the assumption that the average water particle velocities and accelerations for the net panels are assumed to be the same for the models. From Morison theory, the drag and mass forces are given by Eq. (4.4) for n twines in a net panel[27].

$$\sum_{i=1}^n F = \sum_{i=1}^n \left(\frac{\pi D^2}{4} \rho C_M \dot{u} + \frac{\rho}{2} C_D D u |u| \right) \quad (4.4)$$

Hence, the expression can be written as in Eq. (4.5).

$$n \cdot F = n \cdot \left(\frac{\pi D^2}{4} \rho C_M \dot{u} + \frac{\rho}{2} C_D D u |u| \right) \quad (4.5)$$

Furthermore, the expression for the drag and mass force is then given by Eq. (4.6)

$$F = \frac{\pi D^2}{4} \rho C_M \dot{u} + \frac{\rho}{2} C_D D u |u| \quad (4.6)$$

Here, the hydrodynamic force for n twines is given by F where the twine diameter is expressed by D , drag, and mass coefficients are given by C_D and C_M . The equivalent counterparts are given as the equivalent diameter D_e , the drag coefficient C_{De} , and the mass coefficient C_{Me} . Conclusively the relation for the mass and drag coefficients are satisfied by Eqs. (4.7) and (4.8).

$$nC_M D^2 = C_{Me} D_e^2 \quad (4.7)$$

$$nC_D D = C_{De} D_e \quad (4.8)$$

Axial Stiffness Equivalence

The axial stiffness is given by the modulus of elasticity E , and cross-sectional area A , resulting in the relation $k = EA$. The modulus of elasticity is a given constant material property and does not change. The axial stiffness is an expression based on the equivalent cross-sectional area, which is

dependent on the equivalent diameter D_e , which means that the axial stiffness for the equivalent net results in Eq. (4.9).

$$k_e = \sum_{i=1}^n AE = \sum_{i=1}^n \frac{\pi}{4} \cdot D^2 \cdot E \quad (4.9)$$

In conclusion, based on the equation for finding the equivalent properties, the numerical model properties for the coarse, medium, and fine numerical models are summarized in Table 4.3 and briefly mentioned previously in Table 4.2. The equivalent properties are established based on the full-scale fish cage model. The number of sections in the fish cage is the number of knots iterated in a circle to complete the fish cage. The number of levels refers to the number of layers of knots vertically that are iterated in the code from the top of the cage to the bottom. By adjusting the number of sections and levels, the mesh density of the fish cage is altered. The number of net knots is, therefore, a combination of the number of sections and levels, but gives an impression of the significant increase in the number of knots as there is an increase in sections and levels.

Table 4.3: Fish cage equivalence properties for a full-scale fish cage and for numerical fish cage models with net mesh density that is coarse, medium, and fine.

Net	Coarse	Medium	Fine	Units
Diameter	0.0035	0.033	0.022	<i>m</i>
Mass / Length	0.96	0.66	0.44	$\frac{kg}{m}$
Spacing	4.00	3.00	2.00	<i>m</i>
Axial stiffness	394.08	589.05	773.13	<i>kN</i>
Drag coefficient, C_D	15.91	13.14	10.71	
No. of cage sections	32	48	64	
No. of cage levels	11	15	23	
No. of net knots	353	721	1473	

4.1.3 Mooring System Configuration for Numerical Fish Cage Model

The floating fish farm consists of four essential parts, which is the fish cage itself, the feed vessel, feeding tubes, and mooring lines. A mooring system must be configured to absorb the environmental loading from current and waves in the ocean. A typical mooring grid with anchors, buoys, and barges allow for optimal positioning to the farm site. The mooring system of a single fish cage using a conventional mooring system would contain eight anchor lines that are attached to steel connector plates, with four frame cables that encapsulate the fish cage in one “cell” that is connected to buoys floating at the water surface. Twelve bridles are used to attach the fish cage and the connector plates [28]. An example of a conventional mooring system for a fish farm with multiple fish cages, a feed barge, and feed lines are shown in Fig. 4.11.

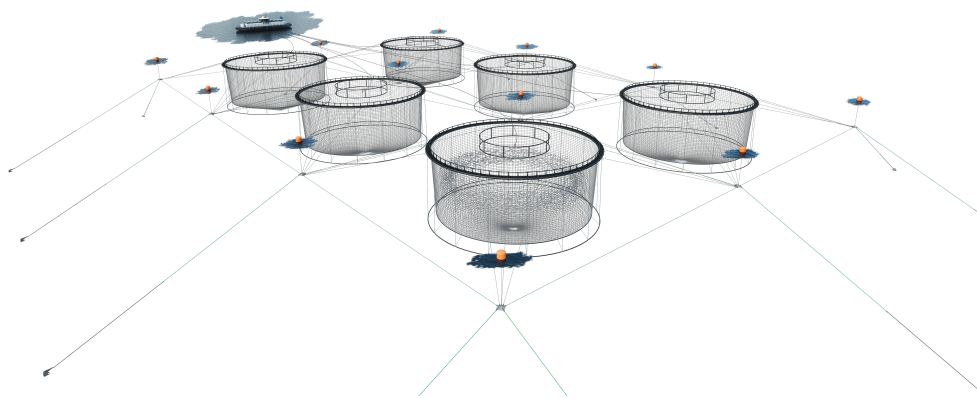


Figure 4.11: A conventional mooring system shown for a fish farm with multiple feed lines and fish cages showing feed vessel and mooring lines at the facility [25].

For a mooring system, it is important to maintain horizontal tension on the anchor lines to prevent displacement of the moored structure. Furthermore, it is important to maintain tension in the mooring lines to prevent translational movement of the fish cage and to avoid snap loads. Therefore, it is necessary to pre-tension the mooring lines. Using the OrcaFlex Line Wizard tool, it is easy to calculate the required length from the position of the anchor lines in order to pre-tension the lines to the desired level. The line wizard calculates a required length for the desired tension in an iterative process, based on the structural properties such as axial stiffness, line length, segment length, and endpoints [29]. The pre-tension for the mooring lines should be around 60.00 kN , as this would provide sufficient stability, while still not exceeding any structural thresholds for the mooring system. It should be noted that it is important to allow for some motion in the system, as a rigid system would not be sustainable in an offshore environment. A schematic overview of the fish cage model with a mooring system is shown in Fig. 4.12.

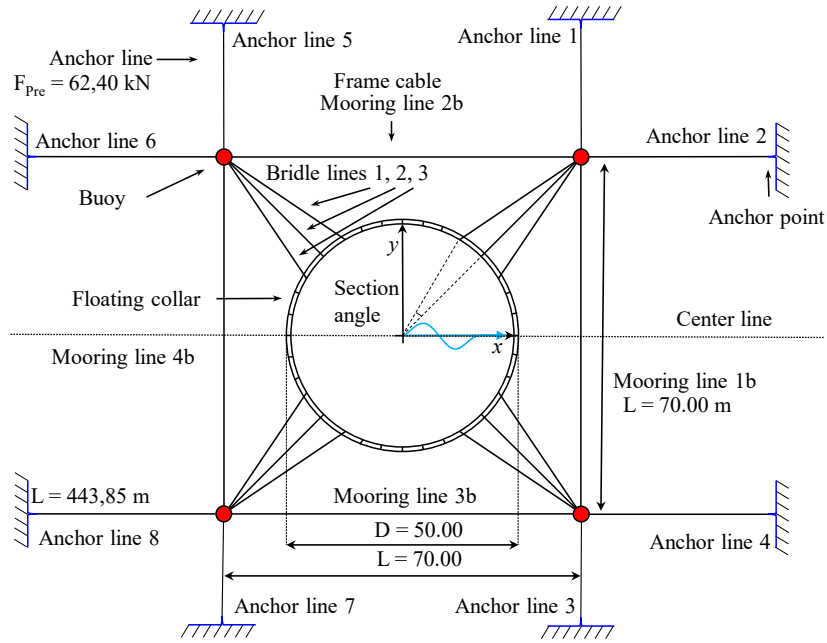


Figure 4.12: Schematic overview of fish cage model with complete mooring system consisting of anchor lines, frame lines, bridles, and fish cage.

The numerical model of the mooring system is generated using the OrcaFlex API (Application Programming Interface) and Python, but the code is not significantly different compared to the code used to generate the fish cage model. The primary difference is that the line type is defined with different structural properties, and the mooring lines are substantially longer. After establishing the mooring system for the fish cage, a pre-tension of 62.40 kN was obtained, which is within a reasonable limit of the desired tension in the mooring line, without taking into account environmental loading effects. In Fig. 4.13, the mooring system for the fish cage can be seen from the side and top, showing anchor lines, frame cables, bridles, and buoys.

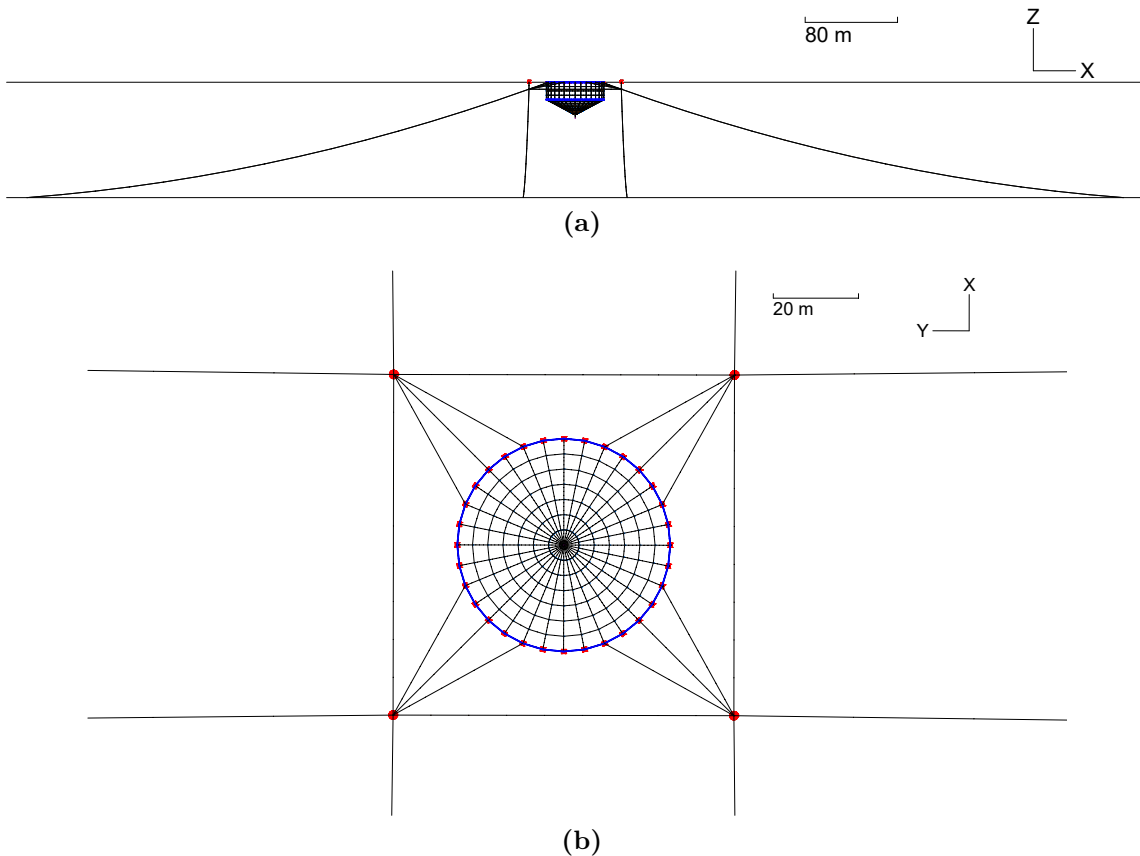


Figure 4.13: Fish cage mooring line and anchor system seen from OrcaFlex. The fish cage model can be seen with anchor lines connected to the seabed from top view (a) and side view (b)

The mooring system for the feed vessel is established in a similar fashion to the fish cage, with a pre-tensioned mooring system. However, the mooring system for the feed barge is based on a conventional mooring system for a vessel and is not displayed in this section. The mooring system for the feed barge restricts the displacement of the vessel while maintaining stability for the fish farm at the location.

4.1.4 Modeling of the Feeding Tube

At a fish farm facility located in the ocean, feeding tubes play a critical part in supplying the fish with feed either transported by air- or water-filled pipes [17]. A fish farm facility with multiple pens typically has one feeding tube per fish cage, transporting feed from the feed barge to nearby cages. In the industry, it is common to use air-based feeding systems; hence, the feeding tubes are usually floating in the surface. They are therefore exposed to slamming loads and environmental loading

due to current and waves. When modeling the feeding tube in OrcaFlex, the tube is connected from the feed vessel to the fish cage top collar and is based on HDPE (high-density polyethylene) material. The properties of the feeding tube are presented in Table 4.4.

Table 4.4: Properties and dimensions of the feeding tube used for fish farming [30].

Parameter	Dimension	Units
Inner Diameter	0.110	<i>m</i>
Outer Diameter	0.0974	<i>m</i>
Wall Thickness	0.0065	<i>m</i>
Material Density	960	$\frac{kg}{m^3}$
Young's Modulus	1.10	<i>GPa</i>

As the feeding tube is a long hollow cylinder, Morison theory applies to the environmental loading on the tube. By implementing a variable data set for the drag coefficient, the normal relative velocity over the element is calculated considering the impact of effects such as waves, current, and the speed of the line element [11]. Furthermore, the Reynolds number can be calculated to compute the drag coefficient for the element by considering the relative velocity of the feeding tube.

Feeding Tube with Bend Stiffener

In order to reduce the dynamic loading on Subsea flowlines or pipelines, bend stiffeners are sometimes used to increase joint stiffness in order to avoid breakage. In environmental conditions that may force large movements or oscillations, bend stiffeners prevent loading that may lead to breaking, rupture, or fatigue of the pipe connections. In some of the numerical simulations, bend stiffeners are included to simulate the beneficial effect of utilizing such equipment for aquaculture operations. The bend stiffeners are modeled as a conical connection to the feeding tubes that cover the feeding tube line and protects about the first 3 meters of the feeding tube element in OrcaFlex [30]. This is done by adding a variable diameter on the feeding tube line, which makes the connections conical. There is no hang-off for the feeding tube from the barge connection, as it is positioned at the water surface, and it is also positively buoyant. The dimensioning of the bending stiffener in the thesis is not considered in detail but is attached in OrcaFlex with a protected region of 3 meters

at both ends, and the diameter of the bend stiffeners is set to 0.12 meters at the connection to the feed barge and fish cage, and 0.0974 meters at the connection to the feeding tube[33].

4.2 Convergence Study of Fish Cage Models

Three numerical models of varying panel sizes are constructed to identify a satisfactory mesh density for the fish cage. As discussed, the equivalence for the respective cages is established found, hydrodynamic properties are calculated. Furthermore, steady current simulations are subsequently run to interpret the response and deformation of the fish cages for current velocities $U_C = 0.00 \frac{m}{s}$, $U_C = 0.20 \frac{m}{s}$, $U_C = 0.40 \frac{m}{s}$, $U_C = 0.60 \frac{m}{s}$, $U_C = 0.80 \frac{m}{s}$, and $U_C = 1.00 \frac{m}{s}$. The deformations of the cages are shown in Fig. 4.14 showing the deformation for varying current levels for coarse, medium and fine mesh fish cages with 32, 48, and 64 sections, respectively. The cages have 4.00, 3.00, and 2.00 meters spacing between knots in the fish cages. The deformation of the fish cages are similar to the equivalence of the fish cage ensures that the drag coefficients, diameters, mass, and axial stiffness are the same as for a full-sized fish cage. By calculating the position of each knot in three dimensions during steady current conditions, the volume deformation can be calculated and compared with the deformation of a fish cage, which serves to validate the behavior of the models. For the model, it is interesting to note that for increased current deformation, the volume deformation naturally increases. As the deformation increases, the coarse 32 section model seems to be slightly uneven in resolving the horizontal displacement for the twines for the conically shaped bottom section, but the fine 64 section models seem to deform this issue better.

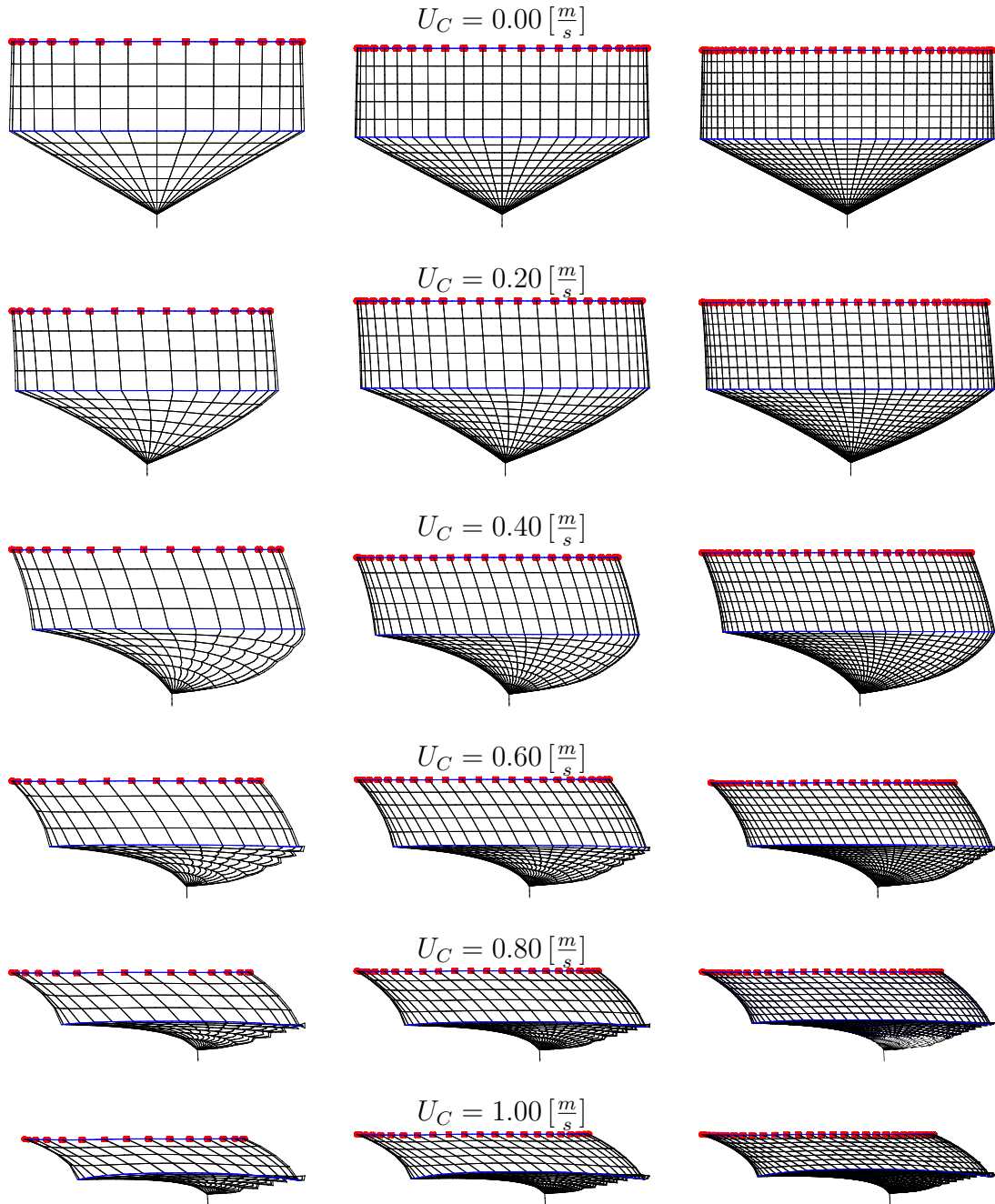


Figure 4.14: Side view of fish cage models showing the cage deformation with current flowing from left to right. The models are a coarse mesh, 32 section fish cage with 4 meters spacing (left), a medium mesh, 48 section fish cage with 3 meters spacing (middle), and a fine mesh, 64 section fish cage with 2 meters spacing (right).

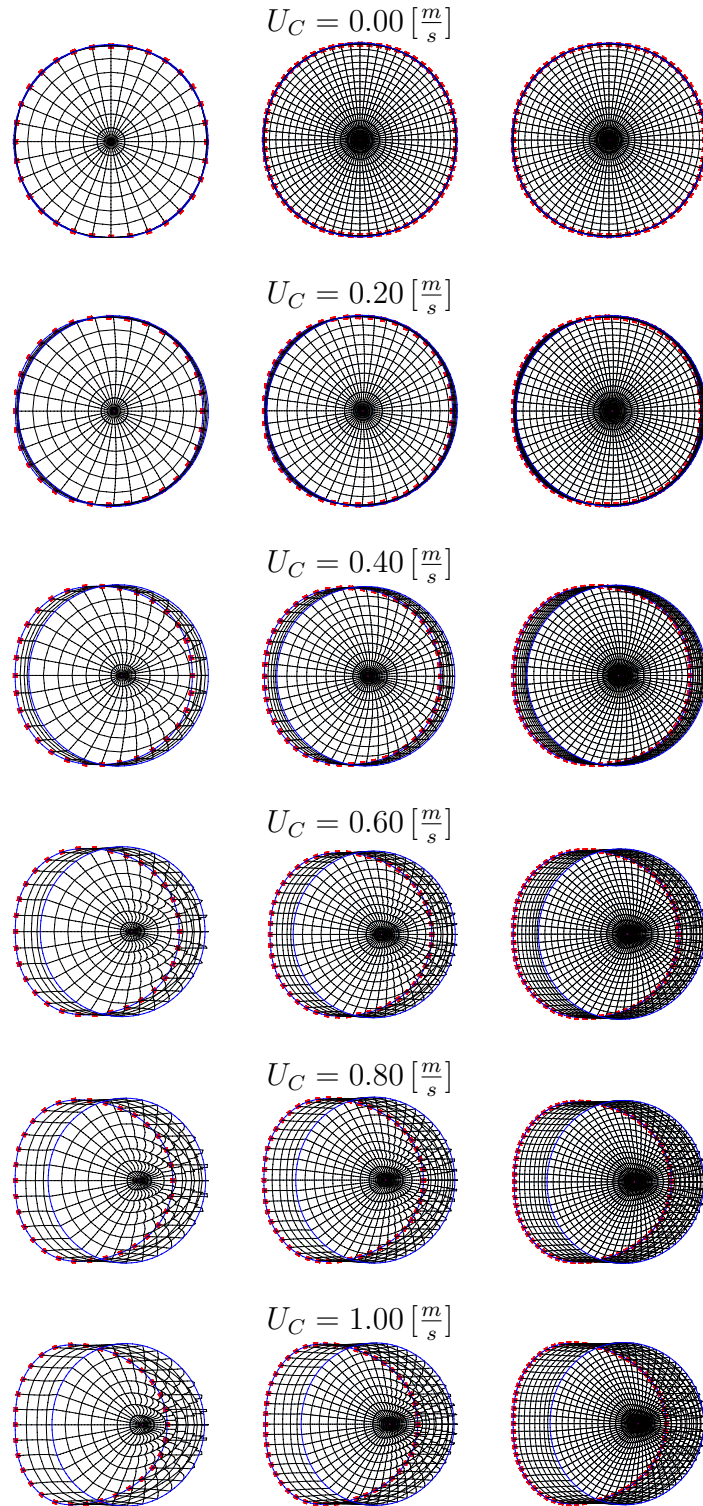


Figure 4.15: Top view of fish cage models showing the cage deformation with current floating from left to right. The models are a coarse mesh, 32 section fish cage (left), a medium mesh 48 section fish cage (middle), and a fine mesh, 64 section fish cage (right).

As can be seen for the various models and the cage deformation, the deformation due to current is significant for the conical section from $U_C = 0.40 \frac{m}{s}$ to $U_C = 1.00 \frac{m}{s}$, which coincides well with the previously discussed environmental parameters, where current conditions above $U_C = 0.40 \frac{m}{s}$ is classified as being moderate to large in NS 9415 [31]. Furthermore, as the conical section is lifted into the cage for current velocities above $U_C = 0.40 \frac{m}{s}$, this will significantly reduce the volume of the cage and harm the fish and can lead to a negative impact on the well being of the livestock in the fish cage. The cylindrical section above the sinker tube seems to retain shape relatively well, because of the mass of the sinker tube, which helps to maintain a stretched cylindrical section. However, the top collar does seem to be slightly displaced by the current. It is interesting to note that in Fig. 4.14, the twines in the coarse fish cage appear to be folding crookedly, which is probably due to a lack of sections elements in the twines. Furthermore, the medium and fine section models seem to resolve this issue as a result of increased mesh density.

The general deformation mechanism seems to be that the conical section is pushed up into the cylindrical section. The cylindrical section is subsequently compressed for higher current velocities. For the different net resolutions, it is clear that the coarse 32 section model has few twines, which is apparent when looking at the deformation for larger currents in Fig. 4.14. For the fine 64 section fish cage, the nets seem to deform better, as the twines appear to bend more like one would expect a real fish cage to deform. It is observed that when the current velocity is strong enough to displace the conical and cylindrical sections, a significant deformation of the fish cage occurs. As seen in Fig. 4.15, this has a significant increase in the displacement of the cylindrical section of the cage around $U_C = 0.40 \frac{m}{s}$.

The relative volume of the cage is shown in Table 4.5 and is calculated relative to the deformation of the various current velocities listed. Empirical studies provided by Shen et al. [32] are based on a cylindrical fish cage in steady current conditions with a sinker tube mass of $80 \frac{kg}{m}$ and a sinker mass of $1000 kg$. Their study analyzed a fish cage comprised of a floating collar with two concentric tubes, a flexible net cage including a cylindrical and conical part with center point mass and sinker tube attached with a conventional mooring system similar to the one studied in this thesis. Similar to the model, the inner tube diameter is 50 meters and with a total height of 25 meters for the full-scale model and 3.125 meters diameter and height 1.00 meters for the model [32]. The deformation of the models is found by calculating the displaced volume of the simulations in comparison to the original calculated volume. The relative volume ($\frac{V}{V_0}$) is used to express the deformation, as the difference in the number of knots and mesh density for each model has an impact on the accuracy of the volume calculations for the model, but will be sufficient for comparing the volume deformation of the models.

Table 4.5: Relative volume of varying mesh density for fish cage model of coarse, medium, and fine mesh density compared to empirical studies for similar fish cages and conditions [32].

$U_C [\frac{m}{s}]$	Coarse [%]	Medium [%]	Fine [%]	Empirical studies [%]
0.00	100	100	100	100
0.20	98.15	98.55	98.79	95
0.40	87.79	88.71	89.74	90
0.60	71.45	71.15	71.53	75
0.80	54.53	54.24	53.43	50
1.00	40.52	40.09	38.98	40s

The simulation for the different cages shows that the volume deformations for the different models are very similar, with a difference of less than one percent in the volume deformation for most current velocities. The higher density net models are likely to provide more accurate estimates of volume deformation, as models of higher net density have more knots in the models. This naturally means that fish cage models of higher density nets have a more accurate estimate for the volume and the volume deformation, as there are more knots for calculating knot position and cage volume. Because of the difference in net mesh and the number of nodes used for the measurements, it is useful to represent the results as relative percentage deformation for each model. By comparing the percentage deformation, the coarse model with 32 sections, with 4.00 meter spacing between the knots seems to provide similar results as the higher density mesh fish cage models. An overview of the relative percentage difference in the volume deformation of the models can be seen in Fig. 4.16.

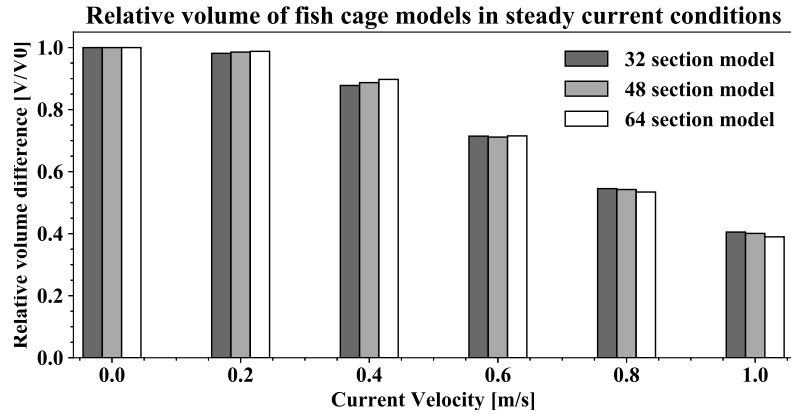


Figure 4.16: Bar plot of the relative volume deformation of the coarse (dark grey), medium (light gray) and fine (white) fish cage models in steady current conditions.

From the Fig. 4.16, it is clear that the overall reduction in volume for increasing current velocity is significant, but the differences in the volume deformation between the models seem to be small. In general, higher current velocity deformation is difficult to model accurately, because of mechanisms such as shielding effects and turbulent flow, which will have a more significant effect for larger current velocities. However, the difference in the relative volume of the model used for the empirical study in steady current conditions seems to differ by about less than 5 % between the numerical models and the empirical models. The difference between the relative volume of the models seen in Fig. 4.16 appears to be slightly lower for the lower current velocities, but slightly higher for the large current velocities, for the coarse 32 section fish cage compared to the fine 64 section fish cage model. The reasons could be that the coarse model does not accurately represent the deformation as precise as the fine model. Another reason may be that the higher number of nodes used to calculate the relative volume (as there are more positions to calculate the volume for) in the fine 64 section model, small differences in volume deformation become more apparent. However, in general, all three models appear to provide satisfactory values to represent the relative volume, also when compared to empirical values from the studies of Shen et al. [32]. Based on the deformation of the cages, in addition to the estimated relative difference in volume, it seems to be reasonable to complete simulations using the coarse 32 sections cage as there are multiple advantages to using this cage. One of the primary reasons is the significant increase in the simulation time required for the fine sections model, but for the present study, the coarse sections model is likely to produce results of satisfactory accuracy. Hence, the coarse 32 section fish cage model will be used for time domain analysis of the fish farm in OrcaFlex. For more detailed studies, the fine 64 section fish cage model can be utilized if a higher accuracy is desired for simulations.

5 Results and Discussions

5.1 Time Domain Simulation Setup

5.1.1 Environmental Parameters

As more fish farms are built in exposed locations, and more facilities are established in offshore environments, it is necessary to establish a clear guideline for the environmental loading expected and how to safely operate in such conditions. It is, therefore, important to clarify what is considered to be an operational average, in addition to what kind of extreme weather conditions can be expected. In order to create a comprehensive guideline for the operational criteria of fish farming in Norway, Norsk Allmenstandardisering established the Norsk Standard, NS 9415 [31]. The standard specifies acceptable environmental conditions and quantifies the parameters and operational criteria. The standard describes requirements for floating facilities and how to establish the correct documentation for floating facilities. The overall goal of the standard is to prevent fish from escaping from the fish farm and causing unwanted damage to the local ecosystem. In Table 5.1, the different classifications for environmental conditions are listed.

Table 5.1: Environmental parameters as defined by Norsk Allmenstandardisering [31].

Wave Classification	h_s [m]	T_p [s]	Current Class.	U_C [$\frac{m}{s}$]	Exposure
A	0.0 - 0.5	0.0 - 2.0	a	0.0 - 0.3	Low
B	0.5 - 1.0	1.6 - 3.2	b	0.3 - 0.5	Moderate
C	1.0 - 2.0	2.5 - 5.1	c	0.5 - 1.0	Large
D	2.0 - 3.0	4.0 - 6.7	d	1.0 - 1.5	High
E	> 3.0	5.3 - 18.0	e	> 1.5	Extreme

In the present study, an operational condition (1-year return period) and an extreme condition (50-year return period) are established with specified wave height, spectral peak period, and current velocity. These conditions are based on the conditions of Grøttingsøy where Mowi AS (previously Marine Harvest) has a fish farm facility, shown in the map in Fig. 5.1.

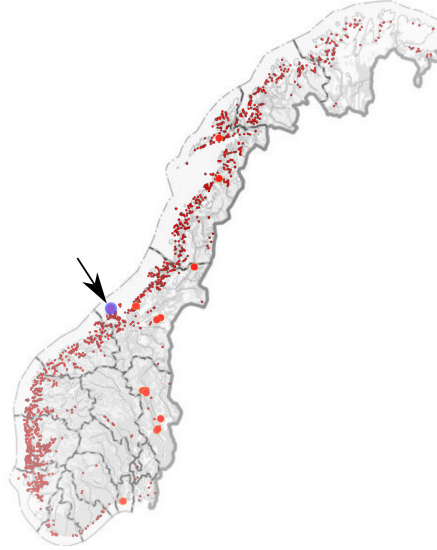


Figure 5.1: The location of the Mowi AS fish farm facility at Grøttingsøy (arrow) along the coast of mid-Norway, which has been selected as a basis for the environmental parameters in this study [4].

As shown in Table 5.2, the 1-year return period at Grøttingsøy is defined as having significant wave height $h_s = 2.71\text{ m}$, a spectral peak period $t_p = 6.00\text{ s}$, and a current velocity $U_C = 0.40\frac{\text{m}}{\text{s}}$ [31]. Furthermore, for a 50-year return period the significant wave height, $h_s = 3.98\text{ m}$, a spectral peak period, $t_p = 7.00\text{ s}$ and a current velocity, $U_C = 0.80\frac{\text{m}}{\text{s}}$. Following Table 5.1, as provided by Norsk Allmenstandardisering, the operational condition with 1-year return period for wave height and spectral peak period within classification “D” and the extreme condition with 50-year return period within classification “E”. For the current classifications, The operational 1-year return period condition has classification “b”, moderate, while the extreme 50-year return period is within classification C, large. Hence, they are all largely considered to be rather severe conditions and should be sufficient to represent reasonable environmental conditions for design purposes. The environmental conditions from Grøttingsøy that are considered in the time domain simulations of this thesis are presented in Table 5.2.

Table 5.2: Operational and extreme environmental conditions for time domain simulations.

	Environmental conditions	$h_s\text{ [m]}$	$t_p\text{ [s]}$	$U_C\text{ [}\frac{\text{m}}{\text{s}}\text{]}$
1-year return period	Operational conditions	2.71	6.00	0.40
50-year return period	Extreme conditions	3.98	7.00	0.80

5.1.2 Time Domain Simulations Setup

For the time domain analysis in OrcaFlex, the simulations are run with a 2 stage interval, where a 200 second build-up period where sea conditions are slowly ramped up from zero. The time domain simulations utilize the JONSWAP spectrum and static wave position (interpolated wave) to calculate wave kinematics. Furthermore, the first stage is preceded by a 3600 second simulation period, where both stages uses a variable time step, but with a maximum time step of 0.10 seconds and a tolerance of $25 \cdot 10^{-6}$. The computational period of one full simulation requires about 18 hours to complete for a two-stage 3800-second simulation, for a computer processor running four cores[29].

An overview of the fish farm assembly is shown in Fig. 5.2, where the mooring lines, the feed barge, and the feeding tube are visible. As shown in Fig. 5.2, the positive x-direction is defined as having a wave and current direction of 0° , and the positive y-direction is defined as a wave and current direction of 90° . The majority of the simulations focus on tensions for the anchor lines and the feeding tube. For the feeding tube, the effective tension that is measured is for the connection between the feeding tube and the feed vessel. The reason why the mooring line and feeding tubes are considered is that they are identified as the critical elements for operations and will sustain the largest loading. Furthermore, depending on the direction of waves and currents, the loads will be distributed on the mooring system relatively symmetrically. Hence, the anchor line tension restraining the fish cage from drifting is representative for what the mooring system will be exposed to various environmental conditions.

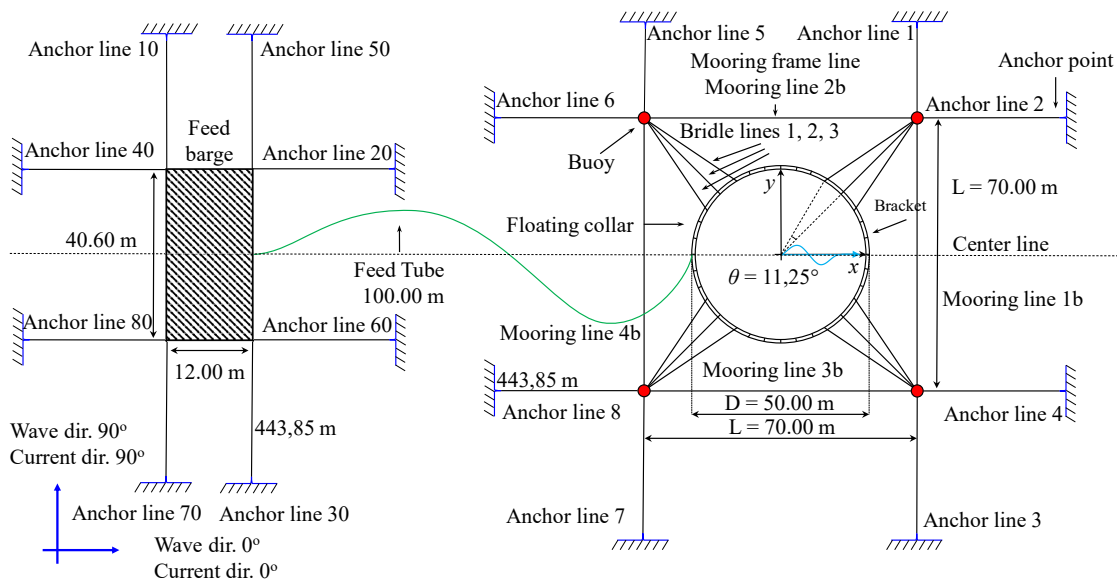


Figure 5.2: Top view of fish farm assembly with feed barge, feeding tube, fish cage, and mooring lines and buoys.

5.1.3 Simulation Matrix

A series of simulations are completed for the various models with a variety of environmental conditions and configurations in order to understand the behavior of the system and critical parts. In Table 5.3, an overview of relevant simulations completed in OrcaFlex can be seen. Current speed U_C , wave amplitude h_s , and spectral peak period T_p is specified for each simulation. The environmental conditions selected for the simulations have been discussed in Section 5.1. The simulations focus on the fish farm assembly with varying environmental conditions, wave and current directions, solidity ratios for the cage, feeding tube length L_A for operational and L_A for extreme environmental conditions, and feeding tube with bend stiffener L_C .

When investigating the environmental loading on the fish farm system, there are a few elements that are of concern, such as the effective tensions in the mooring lines and feeding tube. Furthermore, as shown in Section 4.2 for the sensitivity study of the fish cage models, the volume deformation is useful in achieving an overview of the behavior and environmental loading on the system. Therefore, to understand the environmental loading on the system in different environmental conditions, it is necessary to consider the tensions in mooring lines and other elements. For various current velocities, the maximum tensions observed in anchor lines are extracted, in addition to the mean tensions across the anchor lines, which are shown in Fig. 5.3. The reason for this is to illustrate the difference between the maximum tensions in the anchor lines (in this case anchor line 6) absorbing the axial loading as the fish cage is displaced by the current, in addition to comparing it to the average loading on the system (across all the anchor lines). Hence, this is essentially a comparison of the mean tensions across all anchor lines to that of anchor line 6.

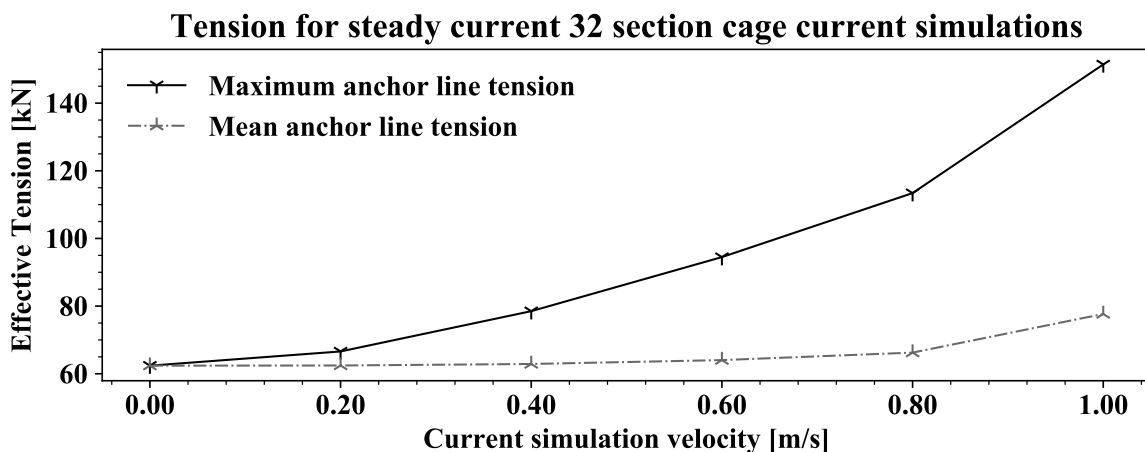


Figure 5.3: Anchor line tensions for various steady current velocities for a coarse, 32 section fish cage, with net solidity $S_n = 0.20$.

Table 5.3: Simulation matrix of environmental parameters and variables for time domain simulations in OrcaFlex.

Sim. no. [#]	Parameter	Env. cond.	Wave, current dir.
1	Fish farm assembly	Operational	0°
2	Fish farm assembly	Extreme	0°
3	Fish farm assembly	Operational	90°
4	Fish farm assembly	Extreme	90°
5	Fish farm assembly	Operational	90°, 0°
6	Fish farm assembly	Extreme	90°, 0°
7	$S_n = 0.15$	Operational	0°
8	$S_n = 0.20$	Operational	0°
9	$S_n = 0.25$	Operational	0°
10	$S_n = 0.15$	Extreme	0°
11	$S_n = 0.20$	Extreme	0°
12	$S_n = 0.25$	Extreme	0°
13	$L_{1A} = 100\ m$	Operational	0°
14	$L_{2A} = 300\ m$	Operational	0°
15	$L_{3A} = 600\ m$	Operational	0°
16	$L_{1B} = 100\ m$	Extreme	0°
17	$L_{1B} = 300\ m$	Extreme	0°
18	$L_{1B} = 600\ m$	Extreme	0°
19	$L_{1C} = 100\ m$	Operational	0°
20	$L_{2C} = 300\ m$	Operational	0°
21	$L_{3C} = 600\ m$	Operational	0°

The first thing to note is that all anchor lines in the model have been pre-tensioned to 62.38 kN , and there is no environmental loading added to the system for $U_c = 0.00 \frac{\text{m}}{\text{s}}$. As can be seen in the figure, it is clear that the maximum anchor line tensions has a significant increase from 62.38 kN to 155 kN for the largest current velocity $U_c = 1.00 \frac{\text{m}}{\text{s}}$. The mean anchor line tension has a smaller, but gradual increase from 62.38 kN to 70 kN . This increase in tension coincides well Morison's equation as the drag force is velocity to the second power. Furthermore, it is also important to consider that the maximum anchor line tensions increases because the maximum tensions in the anchor lines restraining the cage motion are the primary components in preventing the drifting of the structure. The opposite mooring lines, are not in tension as the distance to the fish cage structure is longer and the tension is reduced. This development in the mooring line tensions means that there is a significant increase in maximum tensions, but only a slight increase in the overall mean tensions, even though the environmental loading component increases. It appears that even though the mean loading on the mooring lines does not increase significantly for the cage for various currents, the maximum tensions for the mooring lines will experience an increased loading, due to the direction of the current. The tensions in the anchor lines seems to coincide with the increase in deformation of the fish cages for various current velocities as seen in the Figs. 4.14 and 4.15.

5.2 Time Domain Simulations and Results

5.2.1 Responses Under Operational and Extreme Conditions

In this subsection, the results of the simulations from the first two simulations shown in Table 5.3 will be presented. An overview of the fish farm system is provided, as shown in Fig.5.4. The fish farm is based on a conventional mooring line system, as discussed in Section 4.1.3, with a feeding tube connecting the feed barge with the fish cage. The results will be presented by a bar chart indicating the tensions for all anchor lines and the feeding tube, in order to gain an overview of the loading on the various line elements. The standard deviations of the effective tensions are presented in the tables and give an indication of the oscillations in the line elements. The oscillations for relevant line elements in the wave and current direction are subsequently plotted against time to study the cyclic loading pattern. The maximum, minimum, and standard deviations of the effective tensions will then be introduced in a table, and relevant observations will be presented, comparing the numerical values of the effective tensions in the line elements. The model is shown in Fig. 5.4, and is deformed due to current and wave loads, for current and wave directions set to 0° , and time domain simulations are run for 3600 seconds (1-hour simulations). The tension of the feeding tube is measured at the connection point to the feed barge.

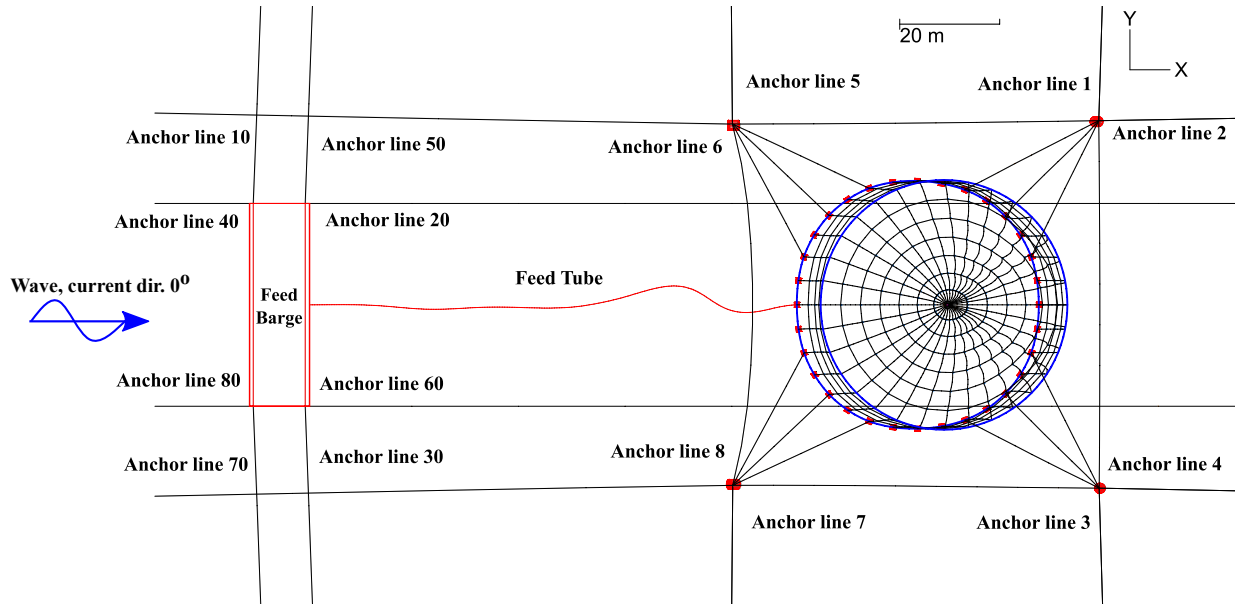


Figure 5.4: Fish cage system in the operational environmental conditions with feed barge, feeding tube, fish cage and mooring lines, with wave and current direction of 0° .

Simulations are carried out according to the environmental parameters as specified in the simulation matrix in Table 5.3. The operational environmental conditions are defined as having a significant wave height $h_s = 2.71 \text{ m}$, wave period $t_p = 6.00 \text{ s}$ and current velocity $U_c = 0.40 \frac{\text{m}}{\text{s}}$. Furthermore, the extreme environmental conditions are defined as having a significant wave height $h_s = 3.98 \text{ m}$, wave period $t_p = 7.00 \text{ s}$ and current velocity $U_c = 0.80 \frac{\text{m}}{\text{s}}$. In the first two simulations for operational and extreme conditions, the wave and current directions are both set to 0° . After carrying out the simulations, effective tensions for the time domain simulations have been extracted and plotted for the fish cage mooring lines and the feeding tube connecting the feed barge and the fish cage. As shown in Fig. 5.4, the fish farm in the operational conditions seems to deform in a relatively similar manner compared to the steady current conditions shown in Fig. 4.15. However, it is important to analyze the time-dependent tension in critical elements such as the feeding tube, and the anchor lines that restrain the cage, which will be experiencing the largest environmental loading.

In order to gain an overview of the loading on the anchor lines for the fish cage, the tensions are shown in Fig. 5.5. The anchor lines that are likely to be relevant for this study are lines 6 and 8, as they are restraining the displacement of the fish cage for the wave and current conditions. Furthermore, anchor lines 5 and 7 experience relatively large tensions, as they contribute to restrain the motion of the fish cage. However, not as much as the anchor lines in the direction of the wave and current motion. As expected, the load bearing anchor lines appear to be experiencing significantly larger tension for the extreme conditions compared to the operational conditions for

anchor lines 6 and 7. This increase is likely to be caused by the increased environmental loading in the direction, and there is a decrease in the loading for the anchor lines away from current and waves. The tensions and the standard deviations are shown in Table 5.5. The feeding tube tension is found to increase significantly for the extreme environmental conditions compared to the operational conditions. The standard deviation gives a measure of the oscillations of the tensions for the line elements.

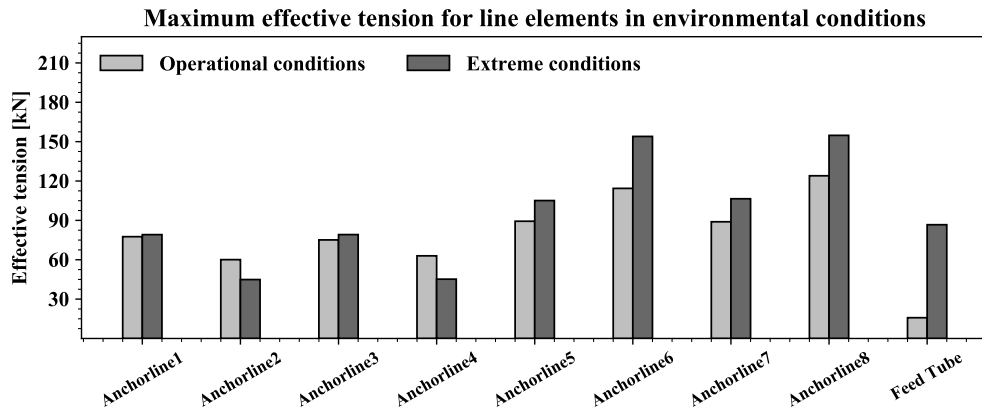


Figure 5.5: Maximum effective tensions in anchor lines for fish cage during the operational and extreme environmental conditions with a wave and current direction of 0° .

The tensions in anchor lines 6, 8, and the feeding tube have been extracted and plotted for the entire time domain simulation of 3600 seconds, to understand the cyclic behavior of the tension lines better. The tension for the feeding tube is measured at the connection to the feed barge, while for the anchor lines, the connections to the connector plates are measured. As can be seen in Fig. 5.6, the tensions in the mooring lines for operational conditions oscillate significantly, while the feeding tube seems to oscillate significantly less. However, there are a few spikes in the feeding tube tension during the time domain simulation. For the extreme environmental condition simulations are shown in Fig. 5.7, where it is obvious that the feeding tube tension and oscillation in the extreme conditions are much larger than for the operational conditions.

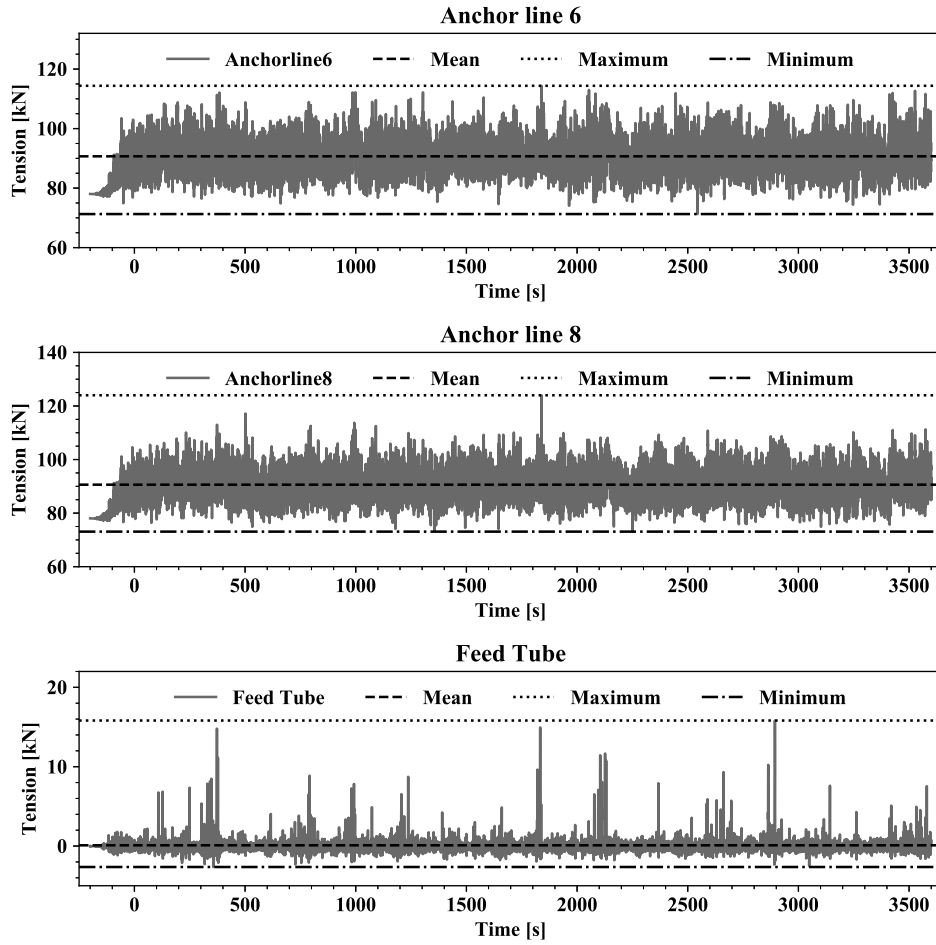


Figure 5.6: Fish farm system in operational environmental conditions, showing effective tension in anchor lines 6, 8 and feeding tube.

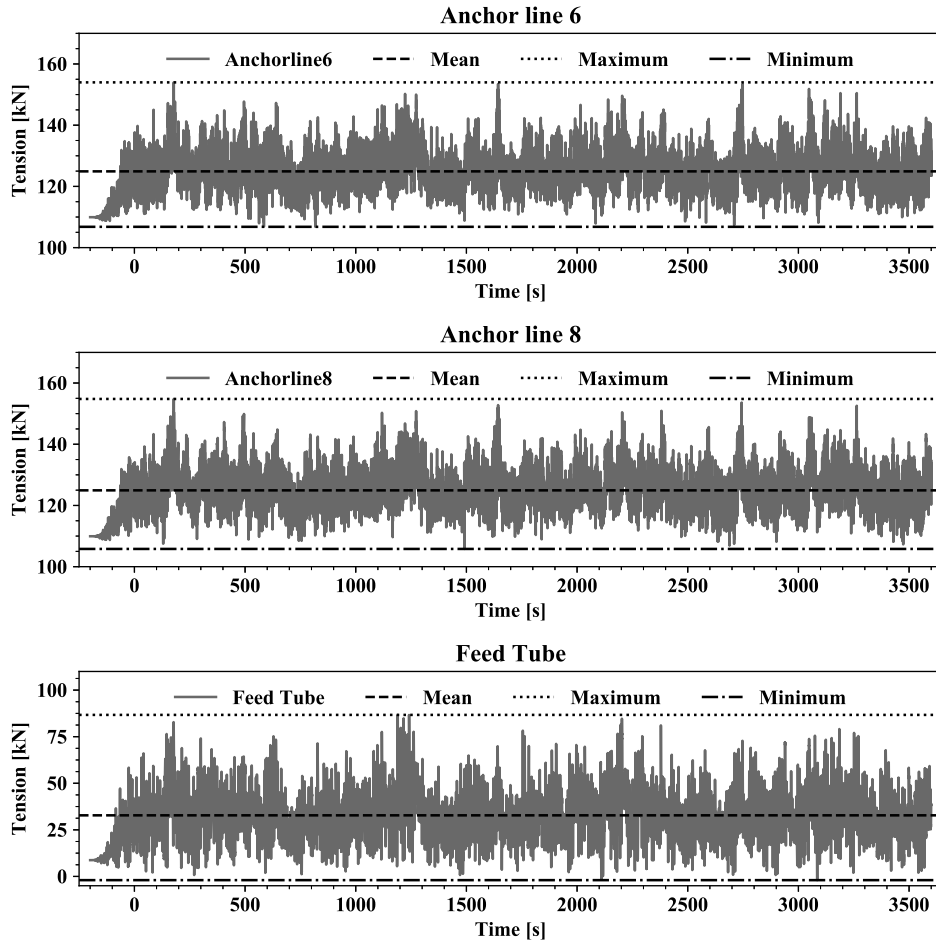


Figure 5.7: Fish farm system in extreme environmental conditions, showing effective tension in anchor lines 6, 8 and feeding tube.

It should be noted that the minimum tension of the feeding tube in Figs. 5.6, and 5.7 are negative, hence resulting in compression for the line element. Furthermore, the feeding tube is undergoing cyclic loading that is oscillating between being in compression, to being in tension. This cyclic loading pattern is likely to cause snap loads. Snap loads can cause significant damage, increases fatigue, and can lead to ruptures of the feeding tube. The oscillations in the tension of the anchor lines shown in Fig. 5.6 are much larger and show a different behavior compared to anchor lines in Fig. 5.7. For higher current velocities or higher solidity ratio, it is expected to find a relatively small increase in standard deviations, because the current loading is relatively constant over time, while the wave loads are cyclic. This trend can be seen in the figures showing the oscillations in anchor lines over time, but also for the difference in the value of the standard deviation for the simulations in Table 5.5. The standard deviations of the anchor lines do not seem to increase significantly for

the extreme conditions compared to the operational environmental condition. This indicates that there is an increased axial force on the feeding tube, which can cause severe damage to the feed pipe.

Table 5.5: Maximum, minimum, mean tension and standard deviation for anchor lines and feeding tube for simulation of operational condition (1 year return period) and extreme conditions (50 year return period) conditions.

Operational Conditions (1 year return period)				
	Min [kN]	Max [kN]	Mean [kN]	Std. Dev. [kN]
Anchor line 6	71.28	114.38	90.68	5.44
Anchor line 8	73.09	124.00	90.62	5.44
Feeding Tube	- 2.64	15.82	0.09	0.82
Extreme Conditions (50 year return period)				
Anchor line 6	106.80	153.99	124.91	6.32
Anchor line 8	105.80	154.80	124.94	6.34
Feeding Tube	- 2.00	86.70	32.79	14.09

For the operational and extreme environmental conditions, the tensions in the anchor lines 8 and anchor line 6, are very similar. The reason for this is that the wave and current directions are 0° , hence distributing loading evenly across the anchor lines. The increased mean tension for the operating conditions compared to the pre-tensioned anchor lines is $28.24 kN$. This increase can be interpreted as a percentage increase of 45 % compared to the pre-tension in the anchor line. However, the maximum added tension compared to the pre-tension in the mooring line is an additional $61.62 kN$, an increase of 100 %, which is a substantial increase for the operational environmental conditions. These maximum tensions seem to occur around 1800 seconds for a short period of time but do not appear to be a frequent occurrence for the simulation.

For the extreme environmental conditions, the tensions in the mooring lines increase significantly, but the increase in the standard deviation is relatively small. The difference in the tensions between the mean tensions in the anchor lines and the pre-tensioned value is $62.55 kN$, which can be interpreted as an increase of about 100 %. Furthermore, the maximum tension in the anchor lines

compared to the pre-tension is 92.42 kN , which is an increase of 147% relative to the pre-tension in the anchor lines. Also, the percentage increase in maximum tension between the operational and the extreme environmental conditions is almost 50% .

For the feeding tube, the difference between the operational and the extreme conditions is from 15.82 kN to 86.70 kN , which is a massive increase of 448% . The increased tension in the feeding tube is also observable in Figs. 5.5 and 5.7. Also, the mean tension and standard deviation for the feeding tube is significantly higher when comparing a value of 32.79 kN and 0.09 kN to 14.09 kN and 0.82 kN . The standard deviation gives an indication of the magnitude of the oscillation in the tension of the anchor line. Therefore, the extreme condition standard deviation is relatively small when compared to the magnitude of the tension for the operational conditions. The increase in standard deviation for the operational condition was 5.44 kN , which increased to 6.33 kN and 6.34 kN , an increase of 16% . This is a relatively small percentage increase compared to the increase in the magnitude of the tension. However, the increase in the standard deviation for the feeding tube is from 0.82 kN to 14.09 kN , an enormous increase of 1718% . It is noteworthy that the difference in tension may be because, during time domain simulation, instantaneous spikes in the feeding tube tension are caused by restricted motion between the feed barge and the fish cage. The reason for this may be that the length between the feed barge and the fish cage is too small and that the length of the feeding tube should be increased, in order to compensate for this tension. There is little doubt that the feeding tube has a large probability of breaking for a tension of 86.70 kN , especially due to fatigue over time. A possible reason for the excessive feeding tube loading is indicated by Fig. 5.8, where the deformation of the fish cage is shown for the extreme environmental conditions. The deformation of the fish cage floating collar appears to be non-circular, contrary to the deformations shown for operational conditions for Fig. 5.4 and Fig. 4.15 in the steady current simulations for the fish cage models. The deformation indicates that the feeding tube is restricted by the distance between feed barge and fish cage and deforms because of the tension on the feeding tube.

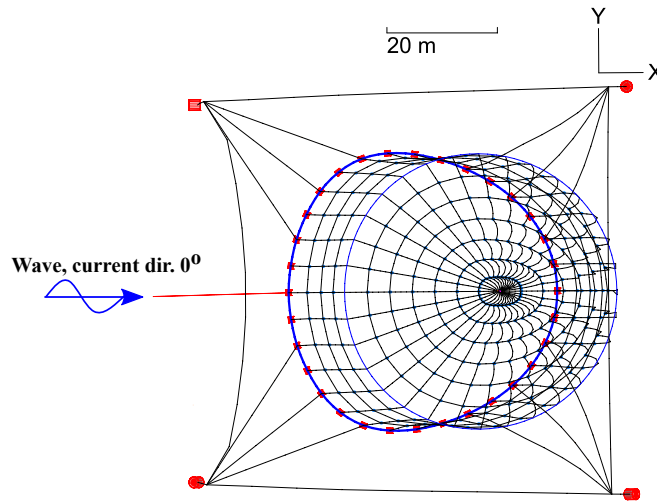


Figure 5.8: Deformation of the fish cage in extreme conditions, floater deformation is non-circular, indicating tension in the feeding tube.

5.2.2 Responses Under Different Wave and Current Direction

The results of the simulations from the third and fourth simulations shown in Table 5.3 will be presented. In order to better understand the effect of the wave and current loads on the feeding tube, it is important to consider alternate wave and current directions for the fish farm assembly. In this simulation, the wave and the current approach perpendicularly, along the positive y-axis, at a 90° angle relative to the feeding tube. Hence, wave and current directions are set to be 90° relative to the x-axis. Initially, an overview of the fish cage setup is provided, and the tensions of the fish cage anchor lines are plotted in a bar chart. The oscillations are plotted for the maximum anchor line tensions (expected to be anchor lines 7 and 3, which restrict the fish cage motion) before the minimum, maximum, mean and standard deviations are listed in a table, and relevant observations are presented.

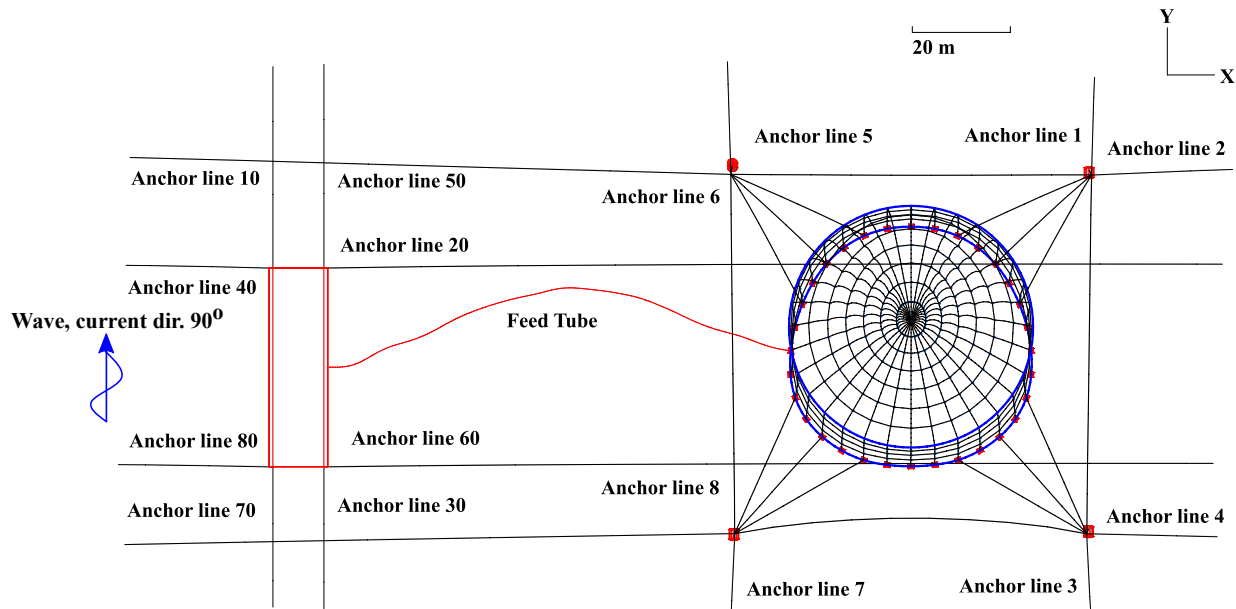


Figure 5.9: Fish cage system with varying wave and current direction set to 90° .

An overview of the fish farm setup for time domain simulations is shown in Fig.5.9, with a wave and current direction set to 90° . As can be seen, the feeding tube and fish cage deform in the 90° direction because of current and wave loads. For this orientation, there is a smaller probability for the fish cage or barge to tug on the feeding tube. Hence the loading on the feeding tube is primarily caused by environmental wave and current loads. Therefore, it is interesting to compare the tension of the feeding tube for various environmental directions, as it is likely to indicate the mechanisms that are contributing to increased tensions in anchor lines and feeding tubes. The tension of the feeding tube is measured at the connection point to the feed barge. The tensions in the anchor lines are plotted in a bar chart shown in Fig.5.10.

From the Fig.5.9, it is clear that the anchor lines restraining the fish cage will in this scenario be anchor line 7 and 3, based on observations from Fig. 5.10. Compared to the previous simulations, it is clear that the tension on the feeding tube varies significantly less from the two environmental conditions, but the behavior of the remaining anchor lines appear to be relatively similar. The maximum tension in the restraining anchor lines 3 and 7 appear to be dominating for these simulations and have large standard deviations because of significant oscillations. The anchor lines 7, 3, and the feeding tube tensions have been plotted over the simulation duration of 3600 seconds in Figs. 5.11 and 5.12. The tension in the feeding tube is measured at the connection to the feed barge, while anchor line tensions are measured at the connector plates close to the fish cage.

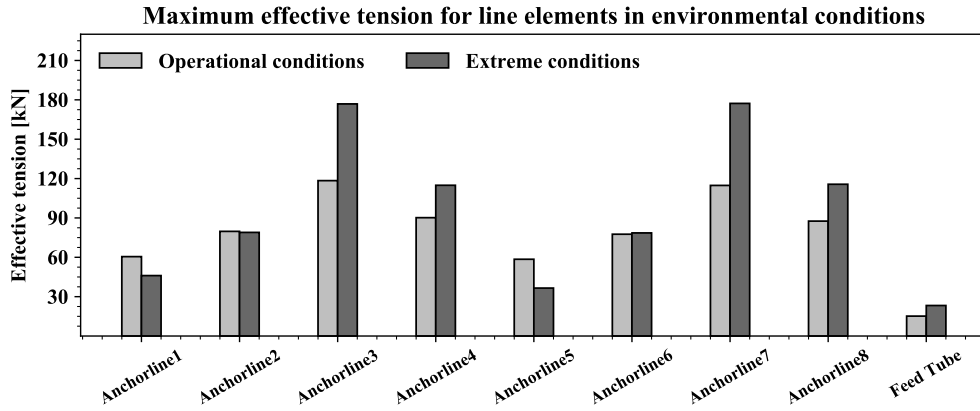


Figure 5.10: Maximum effective tensions in anchor lines during operational (1-year return period) and extreme (50-year return period) conditions for wave and current direction at 90° .

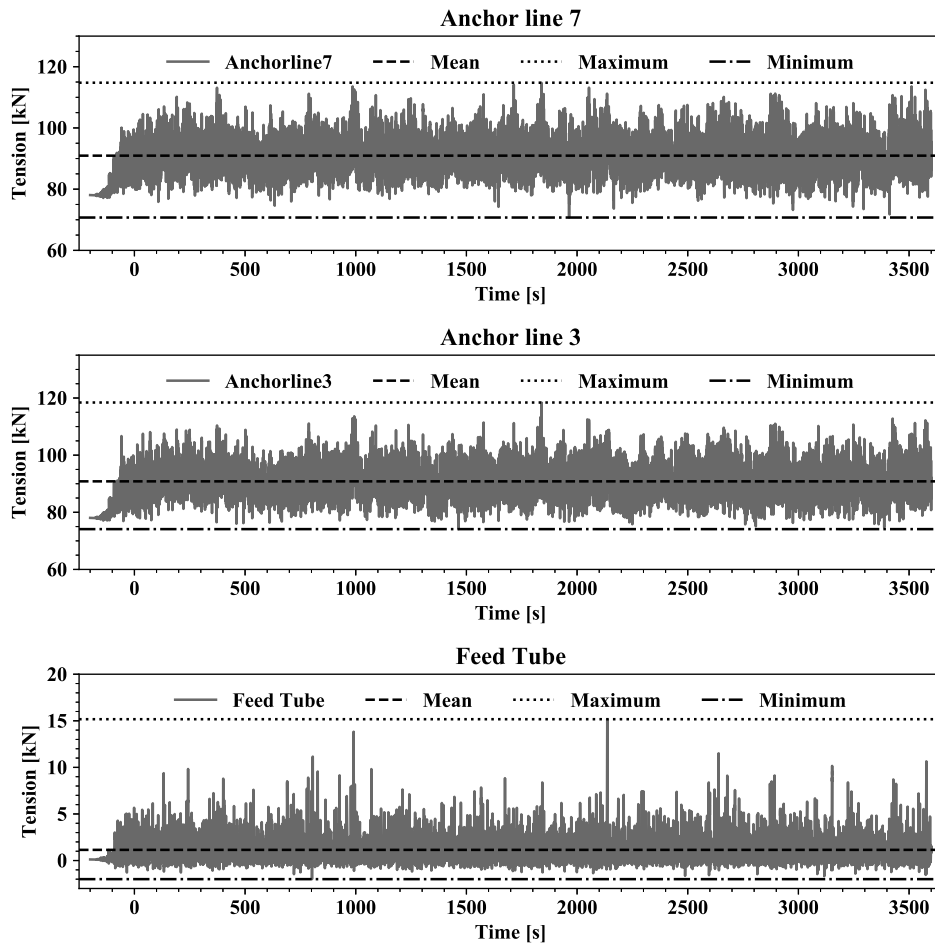


Figure 5.11: Effective tensions in anchor lines 7, 3 and feeding tube in the operational environmental conditions, with perpendicular wave and current direction.

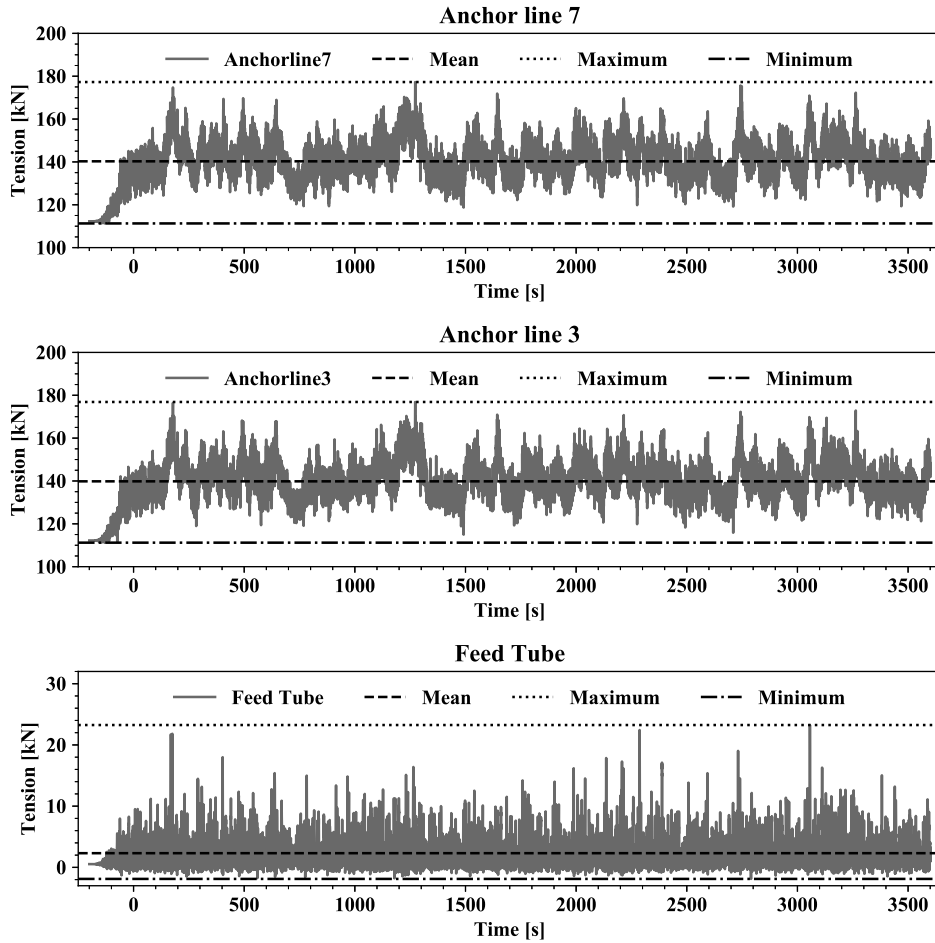


Figure 5.12: Effective tensions in anchor lines 7, 3 and feeding tube in the extreme environmental conditions, with perpendicular wave and current direction.

When comparing the operational and extreme environmental conditions, it is interesting to note the decrease in the magnitude of the oscillations for the extreme environmental conditions. The oscillations in the tensions appear to be smaller for the extreme environmental conditions compared to the operational environmental conditions. Furthermore, this may be because the increased loading results in significant tensions in the mooring system, restraining the lines and keeping a more consistent load on the system. For the feeding tube, however, there appears to be very little difference in the oscillations, indicating that there is little difference in the behavior of the system, other than the added environmental loading from waves and current. It should be noted that the feeding tube tension does oscillates significantly from being in compression to being in tension. These oscillations indicate that snap loads may occur. The minimum, maximum, mean tension, and standard deviations for the simulations for wave and current directions of 90° are shown in Table 5.6.

Table 5.6: Minimum, maximum, mean tensions and standard deviations for line element for 1 year return period, in operational and extreme conditions for wave and current direction of 90°.

Operational Conditions (1 year return period)				
	Min [kN]	Max [kN]	Mean [kN]	Std. Dev. [kN]
Anchor line 7	70.73	114.77	90.94	5.49
Anchor line 3	74.11	118.42	90.83	5.47
Feeding Tube	- 2.00	15.17	1.15	1.48
Extreme Conditions (50 year return period)				
Anchor line 7	111.30	177.28	140.28	9.30
Anchor line 3	111.21	176.87	139.83	9.26
Feeding Tube	- 1.86	23.25	2.32	2.73

From Table 5.6, it is apparent that the mean and maximum effective tensions increases for the extreme environmental conditions compared to the operational environmental conditions. The difference in maximum tension between the conditions is 15.17 kN compared to 23.25 kN , a difference of 53 %, which is comparatively small to the previous simulations (which had a difference of 448 %). Furthermore, the magnitude of the loading on the feeding tube is too large for it to only be caused by hydrodynamic loads. It seems more likely to be caused by the feed barge and fish cage tugging on the feeding tube in a cyclic manner. This loading mechanism is an important observation as it helps to locate the parameters that contribute to the tensions in the mooring lines and feeding tube.

Furthermore, when comparing the tensions in the anchor lines for 90° direction to the previous 0° direction in the operational conditions, it is clear that similar loading effects are acting on the system. For the extreme conditions, there appears to be a slightly larger minimum, maximum, and mean tensions than those for the operational environmental conditions. The increase in the maximum tensions in the anchor lines for the extreme conditions compared to operational conditions seems to be around 50 %, from 114.77 kN and 118.42 kN to 177.28 and 176.87 kN , respectively.

The increase in standard deviations seems to be around 60 %, increasing from 5.49 *kN* and 5.47 *kN* to 9.30 *kN* and 9.26 *kN*. This difference in the standard deviations for a wave and current direction of 90° to the previous 0° indicates that the increased tensions seems to reduce the oscillations of the anchor lines slightly, when comparing the values from Tables 5.6 and 5.5. This trend coincides well with previous observations that indicate an increase in the maximum tensions do not cause an increase in the oscillations of the anchor lines.

Here, Simulations no. 5 and 6, as shown in Table 5.3, will be introduced. As observed in the previous simulations, there is a difference in the loading for the feeding tube depending on the direction of the waves and current flow. Furthermore, there appears to be a smaller environmental loading for the anchor lines restraining the cage when the waves and current encounter the feed barge first. This issue will be studied further by alternating the wave and current directions. For these simulations, the wave direction is set to 90° and the current direction to 0° for both the operational and extreme environmental conditions. In Fig. 5.13, the model is shown during time domain simulations, showing that deformation occurs primarily due to current, as the current deforms the cage along the positive x-axis. Below, the tensions of the fish cage anchor lines are plotted in a bar chart, and the oscillations are subsequently plotted for the largest tensions of the anchor lines and the feeding tube, which is expected to be anchor lines 8, 6, 7 and 3, which restrict the fish cage motion in current and wave direction. The minimum, maximum, mean, and standard deviations will then be listed in a table, and relevant observations are presented and discussed accordingly.

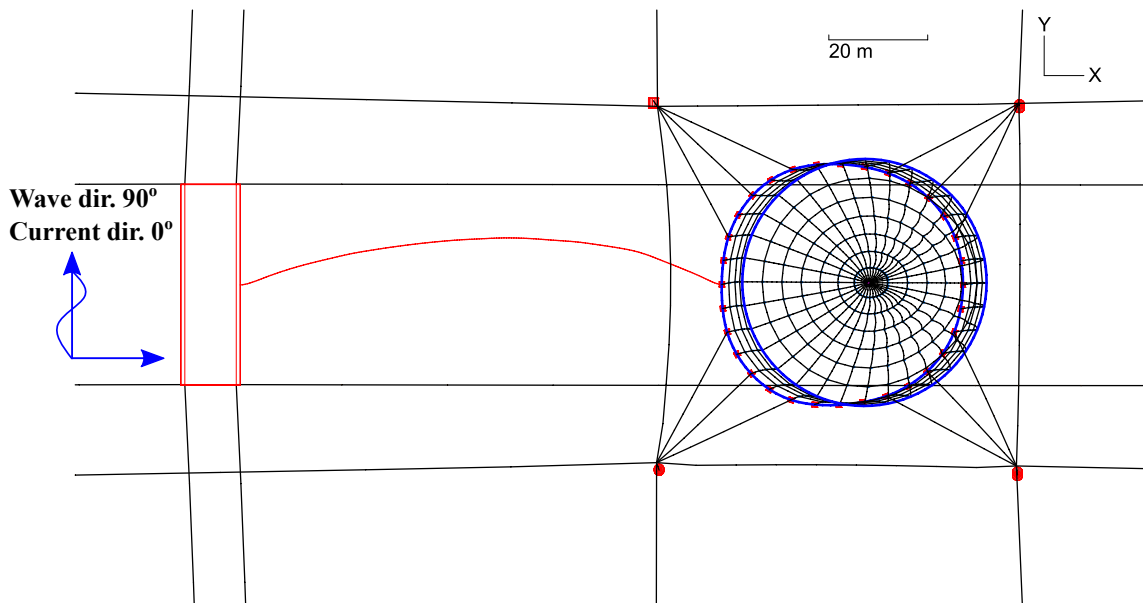


Figure 5.13: Numerical model during time domain simulation with wave direction set to 90° and current direction set to 0°.

In the simulation, the wave direction is set to 90° and the current direction to 0° for the time domain simulations of alternate wave and current direction. An overview of the tensions in the anchor lines and feeding tube is shown in Fig.5.14.

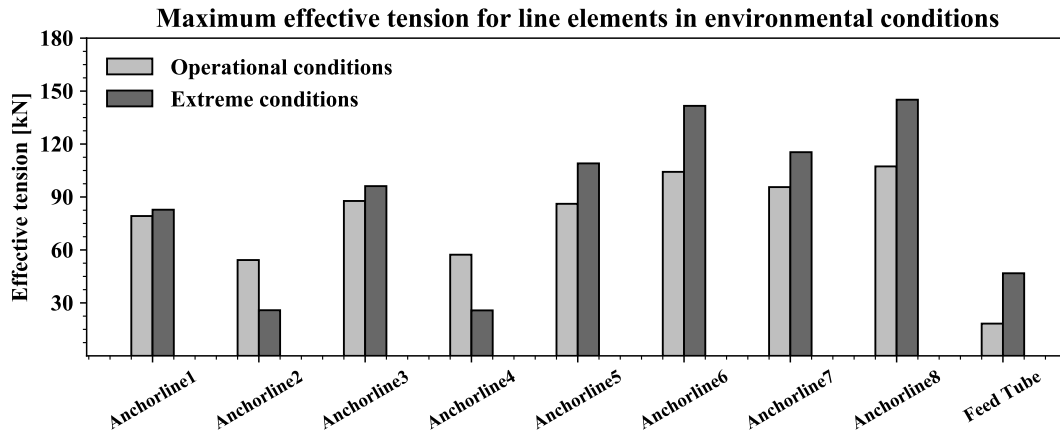


Figure 5.14: Maximum effective tensions in anchor lines and feeding tube for fish cage during the operational and extreme environmental conditions for wave direction at 90° and current direction set to 0° .

From Fig. 5.14, it is clear that the distribution of the tensions appear to be distributed more evenly across the mooring system. For the previous one-directional simulations, a few mooring lines carry the majority of the environmental loading. The distribution of the tensions seem to be as a result of the resultant force, which will be at an angle, because of a different wave and current directions. It appears that the current loading is the dominating force component. Hence, the maximum tensions will be for the anchor lines along the current direction, which is anchor lines 8 and 6. The third and fourth largest anchor line tensions are 7 and 3, which restrict the displacement of the fish cage in the wave direction. The forces in the anchor lines seem to behave similar to previous simulations. There is a substantial increase in the tension and standard deviation of the feeding tube, indicating that the oscillations in the feeding tube will be large. As this simulation has wave and current loads from two directions, anchor lines 3, 7, 6, and 8 will be further evaluated for the time domain, in addition to the feeding tube. See Fig. 5.9 for reference to the anchor lines and feeding tube schematic. The tensions for the anchor lines and feeding tubes for both operational and extreme environmental conditions are shown in Fig. 5.15, 5.16, where the minimum, maximum and the mean tensions are plotted for the duration of the simulation.

From the Figs. 5.15 and 5.16, it appears that the tensions in anchor lines 7 and 3 are relatively symmetric with regards to the oscillation in tension, but the tensions in anchor line 3 has a smaller magnitude than that in anchor line 7. Furthermore, it is apparent that the anchor lines 6 and 8

display similar tendencies. This trend indicates that it is the current load that is the dominant force contributor for the structure. Hence, it appears that the drag and inertia force components of Morison's equation are causing the majority of the load to the system. The overall tension in the mooring lines appears to be distributed more evenly than the previous simulations, which is not surprising as wave and current directions do not coincide. The feeding tube tension seems to be at about the same magnitude as for the first simulation in operational environmental conditions.

The behavior of the fish farm system in the extreme environmental conditions appears to be similar to the operational conditions, but the magnitude of the tensions are expected to be larger. Anchor line 7 appears to have a few large spikes and the oscillations for anchor lines 6 and 8 seem to increase as well. Because of the multi-directional flow of the environmental loading, anchor line 7 is one of the anchor lines exposed to the largest wave load, in addition to the current load. Furthermore, anchor line 8 is exposed to the largest maximum force. The oscillations for the feeding tube are very different for the two environmental conditions and indicates that the feed vessel and the fish cage is transferring some of the loadings onto the feeding tube, due to large motions in extreme environmental conditions. This phenomenon is indicated by the significant difference in maximum tension and standard deviation, indicating large oscillations in tension. These loads can cause ruptures of the feeding tube, and will over time cause fatigue damage to the feeding tube.

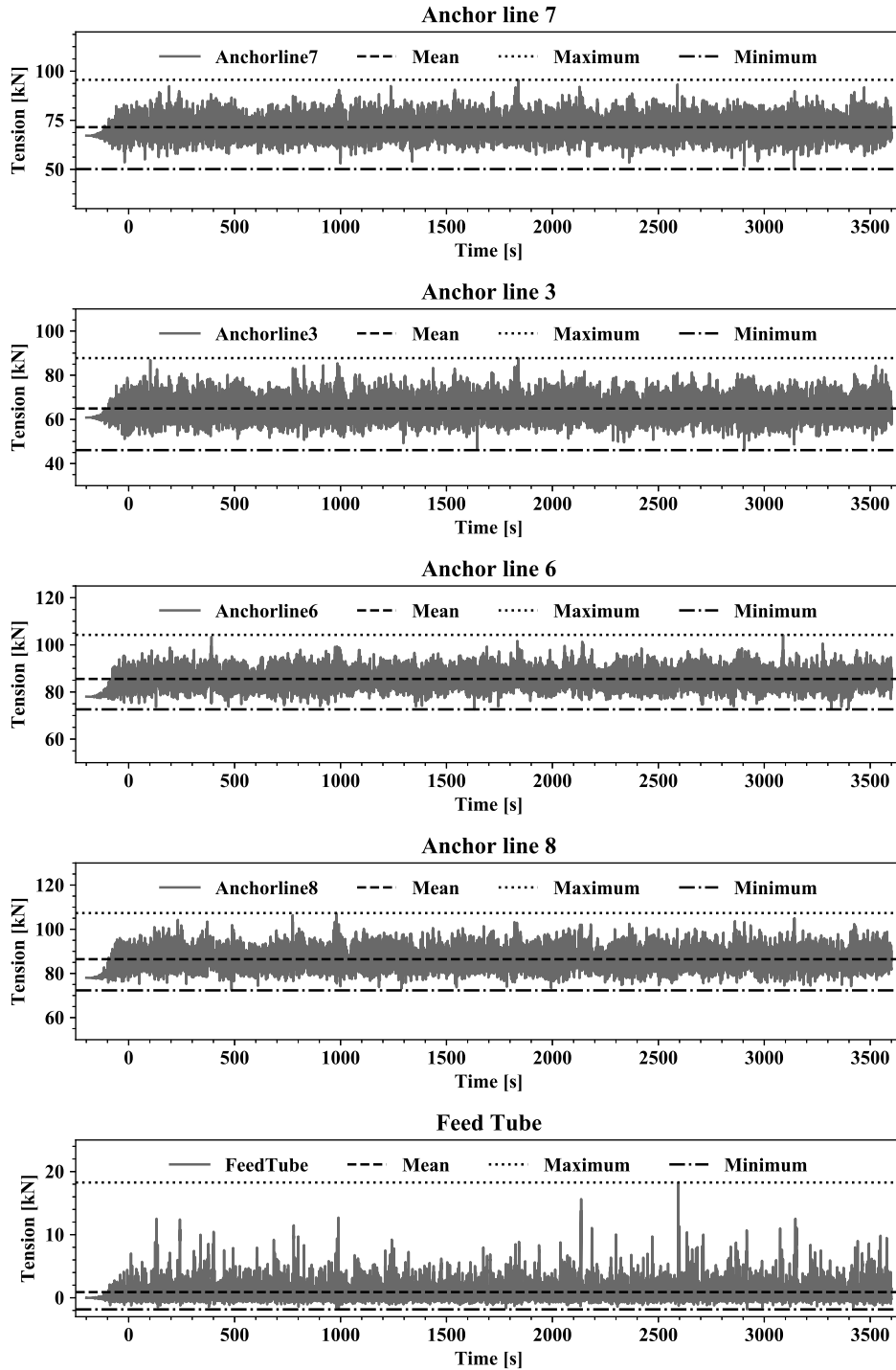


Figure 5.15: Effective tensions for anchor lines and feeding tube in operational environmental conditions, with wave direction 90° and current direction 0° .

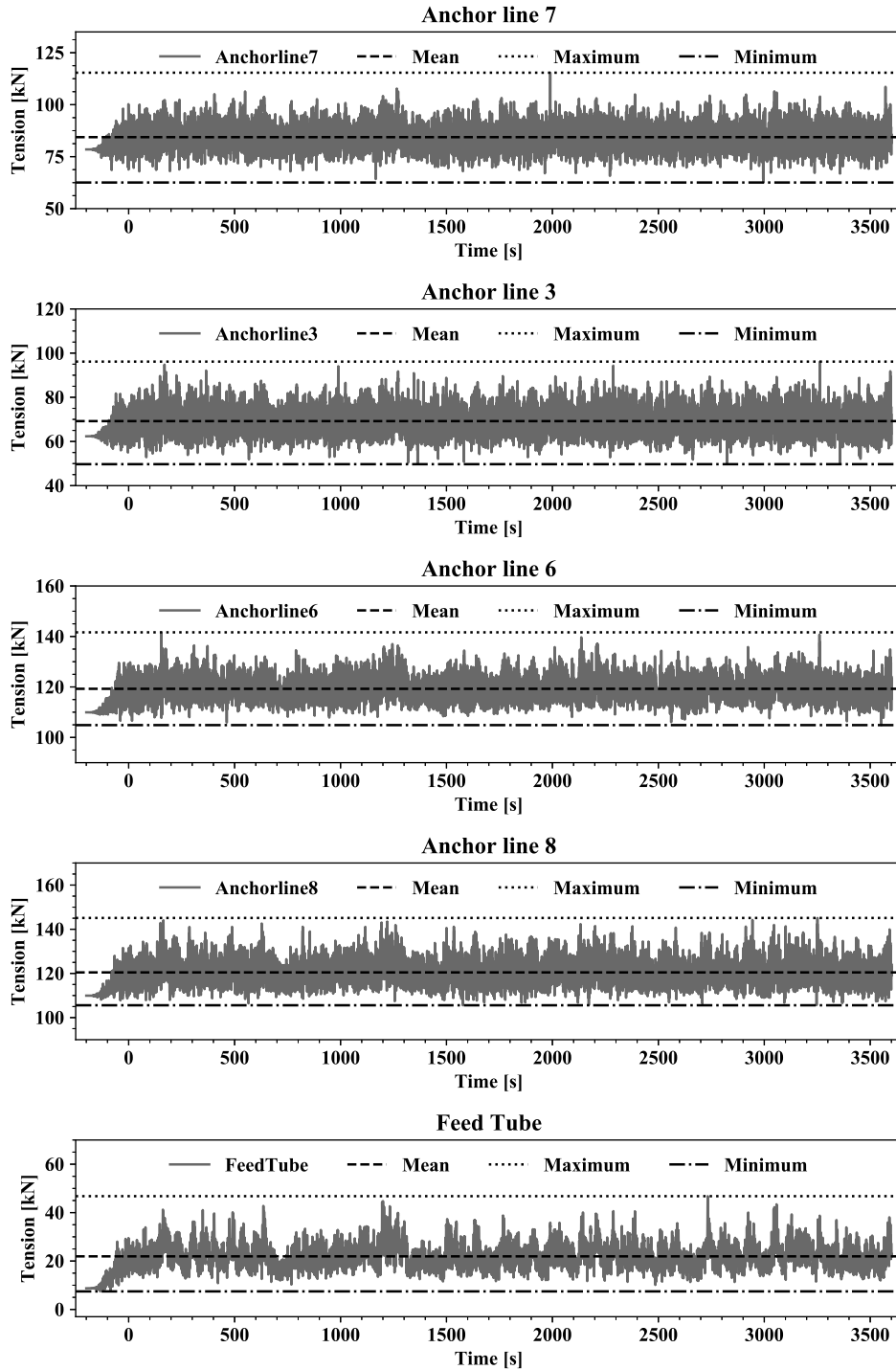


Figure 5.16: Effective tensions for anchor lines and feeding tube in extreme environmental conditions, with wave direction 90° and current direction 0° .

Table 5.7: Minimum, maximum, mean tension and standard deviation for anchor lines and feeding tube for simulations for operational and extreme environmental conditions with wave direction set to 90° and current direction set to 0°.

Operational Conditions (1 year return period)				
	Min [kN]	Max [kN]	Mean [kN]	Std. Dev. [kN]
Anchor line 7	50.15	96.60	71.47	4.64
Anchor line 3	46.08	87.75	64.93	4.43
Anchor line 6	72.62	104.24	85.54	3.47
Anchor line 8	72.33	107.33	86.44	4.13
Feeding Tube	- 1.84	18.27	0.89	1.56
Extreme Conditions (50 year return period)				
Anchor line 7	62.56	115.39	84.38	5.22
Anchor line 3	49.72	96.16	69.22	5.29
Anchor line 6	104.88	141.63	119.26	4.31
Anchor line 8	105.58	145.11	120.46	5.28
Feeding Tube	7.45	46.79	21.93	5.32

The magnitude of the tensions in the anchor lines for the fish cage and feed barge in wave direction at 90° along the y-axis and current direction at 0° along the x-axis, seem to be relatively similar to the previous study. However, the tensions are distributed more evenly among the remaining anchor lines as the current and waves displace the fish cage structure in the direction of flow. Hence, it is noticeable that anchor line 7 is experiencing a significant tension at 96.60 kN for the operational environmental conditions and 115.39 kN for extreme environmental conditions, an increase of 20 %. The symmetric distribution for the anchor lines in current and wave direction seen in the previous mono-directional simulations is no longer seen. This behavior is expected, as the resultant force on the fish farm system is no longer symmetric for the anchor and mooring lines. The maximum tension in the anchor lines of 107.33 kN for the operational conditions and 145.11 kN for the extreme conditions is lower than what is seen in the previous simulations with

the coincidental wave and current direction. In addition, the overall mean tension in the anchor lines appears to be lower with a wave and current flowing in separate directions. The difference in tension is an interesting observation, as it appears that when current and waves flow in the same direction, the tension in the anchor lines is magnified by the coincidental wave and current loading. The maximum tension for the feeding tube in the operational conditions is found to be 18.27 kN and 46.79 kN for the extreme conditions, a significant increase of 156 %. The standard deviations are found to be 1.56 kN for the the operational conditions and 5.32 kN for the extreme conditions, which is an increase of 241 %. Hence, the previous observations for the oscillations in the time domain that the standard deviation is significantly larger and that the magnitude of the tension in the feeding tube is increased substantially.

Direction Comparison

In this section, the tensions for the anchor lines and feeding tube will be compared for the various wave and current directions for the operational and extreme environmental conditions. The tensions in the line elements are presented in Figs. 5.17 and 5.18. Comparing the magnitudes of the environmental loading for the various wave and current directions give valuable insight into the different loading mechanisms for the fish farm. The anchor lines that will experience the most substantial tensions are as previously discussed anchor lines 6, 8, 7, and 3.

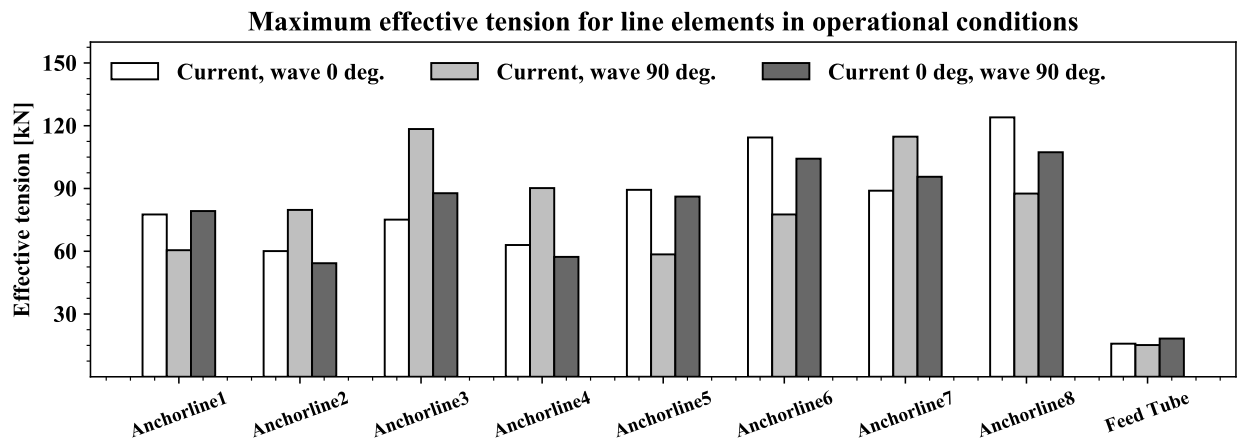


Figure 5.17: Maximum effective tensions in anchor lines and feeding tube for fish farm during the operational environmental conditions for a current and wave direction of 0° (white), 90° (light gray), and for alternative current and wave direction of 0° and 90° (dark gray) respectively.

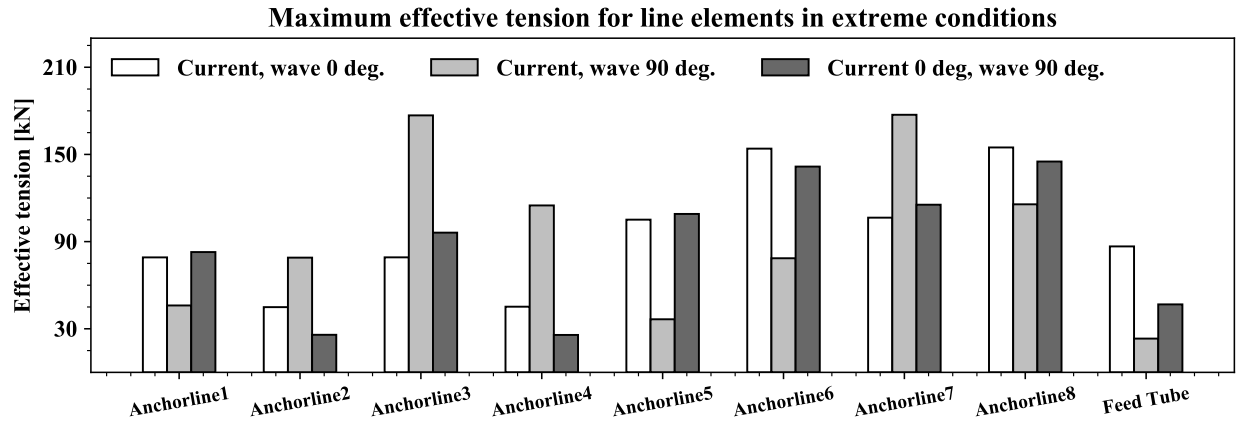


Figure 5.18: Maximum effective tensions in anchor lines and feeding tube for fish farm during the extreme environmental conditions for a current and wave direction of 0° (white), 90° (light gray), and for alternative current and wave direction of 0° and 90° (dark gray) respectively.

For the operational conditions for the three wave and current directions shown in Fig. 5.17, the feeding tube tension seems to differ little compared to the tensions in the anchor lines. Also, the tensions for anchor lines 3 and 7 are quite similar to the tensions of anchor lines 6 and 8. The reason for this similarity is because they are showing the current and wave loads for one-directional wave and current directions. However, the dark gray bar show simulations for two-directional wave and current directions. In other words, the tensions are the symmetric loading for the two anchor lines restricting the fish cage displacement. The anchor line tensions for different wave and current directions appear to be more evenly distributed than the two other simulations. The distribution of tension loads is similar to what is observed and commented on previously. Also, the maximum tensions in anchor lines 6 and 8 is because they are restricting motion in the direction of the current, as the current appears to be the dominating loading mechanism on the mooring system.

The standard deviations of the line elements for different current and wave directions are shown in Table 5.8. The standard deviations indicate the oscillation of the tension of the line elements. As expected, the anchor line tensions both have larger magnitude and standard deviations for extreme environmental conditions, but appear to be experiencing a similar load distribution for the different line elements. From the table, the feeding tube has the largest difference in oscillation between the operational and extreme environmental conditions. For anchor lines 7 and 3 in the extreme conditions for wave and current directions of 90° have the largest standard deviations of the anchor lines, by over 40 %, which is interesting to note. From Fig. 5.18, it is apparent that for wave and current directions of 90° , anchor lines 7 and 3 have the largest tension and standard deviation. The difference in anchor line tension between a wave and current direction of 0° and 90°

could be because the feeding tube may absorb some of the tension, hence reducing the loading on the anchor lines.

Table 5.8: Standard deviations for anchor lines 3, 7, 8 6 and feeding tube in operational and extreme conditions

	Std. dev. for 0°[kN]		Std. dev. for 90°[kN]		Std. dev. for 0°, 90° [kN]	
	Oper.	Ext.	Oper.	Ext.	Oper.	Ext.
Anch. Line 7	3.88	4.77	5.49	9.30	4.64	5.22
Anch. Line 3	2.84	3.24	5.47	9.26	4.43	5.29
Anch. Line 6	5.44	6.32	2.79	3.23	3.47	4.31
Anch. Line 8	5.44	6.34	3.90	5.69	4.13	5.28
Feeding Tube	0.82	14.09	1.48	2.73	1.55	5.32

As previously seen in Fig. 5.18, the feeding tube tension for wave and current directions of 0° is significantly larger than the tension for the other directions. The standard deviation of the feeding tube in the extreme conditions with wave and current direction of 0°, is also found to be substantial, and appears to be largest observed standard deviation for any of the directions. The magnitude of the tension for the feeding tube for this condition is also found to be large. A possible explanation for this behavior may be because the feeding tube is too short to compensate for the fish cage displacement in the extreme environmental conditions.

5.2.3 Sensitivity Study on Solidity Ratio

In aquaculture, there are important effects such as biofouling that increase the drag force on the fish cage and which reduces the flow of fresh water to the cage. Hence, it is vital to regularly clean the fish cage in order to maintain a steady flow of saltwater through the cage. Cleaning the cage will also help mitigate drag forces, which can result in large deformations for the cage. To describe the spacing for the net in a fish cage, it is common to use solidity ratio, which is a property that defines the ratio of the area of the solid part of the net screen to the total area of the screen. Net solidity is a critical parameter in deciding the drag force on a smooth, slender cylinder. It is, therefore, important to study the sensitivity of the fish cage with regards to the difference in the solidity of the net. The impact of an increase in the drag force of the fish cage can be studied by adjusting the

solidity ratio of the fish cage, this is discussed in the Section 4, and the theory behind solidity ratio is explained in Section 5.2.1. In OrcaFlex, numerical simulations to study these effects are carried out for a fish cage for various net solidity ratios, primarily by adjusting the diameter of the twines in the fish cage.

For the purpose of this analysis, three different solidity ratios are considered. Hence, three models are established with different solidity ratios. Two models are built with solidity ratios, $S_n = 0.15$, and $S_n = 0.25$. The model used to study responses for various environmental parameters in Section 5.2.1 utilizes a model with a solidity ratio $S_n = 0.20$. Furthermore, in order to quantify the increased environmental loading on the fish cage, it is relevant to study the difference in the tensions for the anchor lines in order to identify differences in the drag force and environmental loading on the fish cage. The results of the numerical simulations for a fish cage with a net solidity of 0.15, 0.20 and 0.25 are shown in Fig. 5.19 (a), (b) and (c). The figures show the tensions of the anchor lines for the fish cage for the operational and extreme environmental conditions, with error bars indicating the magnitude of the standard deviation. As stated previously, the standard deviation is a property that indicates the oscillations of the tension in the line elements, hence gives a relation for the change in the magnitude of the tensions over time. The cyclic loading on the system originates from the wave load, not from the current load, as the current has a constant velocity, while wave orbitals move in a circular (or oscillates) over time.

From Fig. 5.19 (a), (b), and (c), it is clearly seen that the tensions in the anchor lines for the three simulations vary significantly. From Morison's equation, one would expect that for larger solidity ratios, the tensions in the mooring lines would also increase, and this largely seems to be the case for the anchor lines that are restricting the displacement of the cage (anchor lines 5, 6, 7, and 8). However, the anchor lines opposite the load bearing anchor lines seem to have a reduction in the effective tensions (anchor lines 1, 2, 3, and 4). Considering the operational and extreme environmental conditions seen for the model with solidity ratio $S_n = 0.15$, it is observed that for both conditions, the anchor line tensions vary relatively little over time. However, the standard deviations for anchor lines 2, 4, 6, and 8 are comparatively large. By comparing the three figures for different solidity ratios, the mean tensions observed in the three figures seem to increase from around 90 kN for $S_n = 0.15$ to 120 kN for $S_n = 0.20$, and furthermore to around 140 kN for $S_n = 0.25$. This development seems to indicate a relatively proportional relationship between the increase in the solidity ratio and the increasing environmental loading for the mooring system.

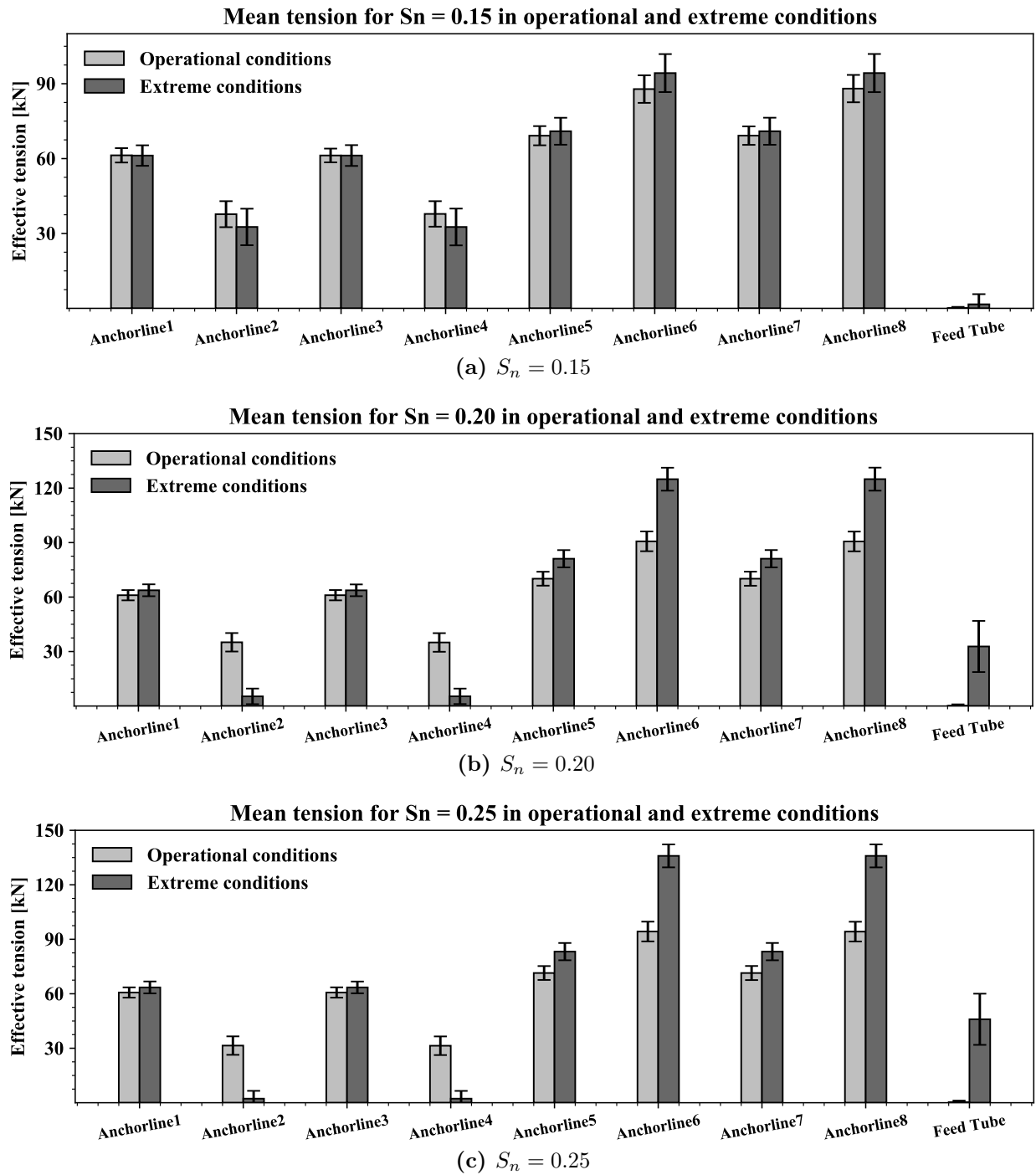


Figure 5.19: Mean tensions for anchor lines and feeding tube in the operational and extreme environmental conditions for fish cages of net solidity 0.15 (a), 0.20 (b) and 0.25 (c).

Furthermore, the standard deviation and maximum tension of the feeding tube in the extreme environmental conditions for all three models are significantly different. This has previously been observed and commented on in the study on responses for various environmental parameters in

Section 5.2.1. The maximum anchor line tensions for line elements 5, 6, 7, and 8 range from 100 kN to 170 kN . As mentioned previously, this behavior makes sense intuitively as these anchor lines restrict the motion of the fish cage, and carry the majority of the load for fish cage motion along the wave and current directions. Furthermore, as the magnitude of the tensions in the anchor lines increases, the standard deviation does not seem to change proportionately. The reason for the small difference in standard deviation appears to be because the oscillation of the loading does not increase. A reason for this may be that the current flow is exerting a more consistent load on the system.

A useful way of illustrating the differences in the loading for various solidity ratios can be done by expressing the relative increase in the tensions of the line elements for different solidity ratio. This is presented in Figs. 5.20 and 5.21, where the anchor lines and feeding tube tensions are compared for the three different solidity ratios. Comparing the relative difference in the mean tensions of the load bearing anchor lines is a useful way to understanding the mechanisms of the environmental loading when changing net mesh densities.

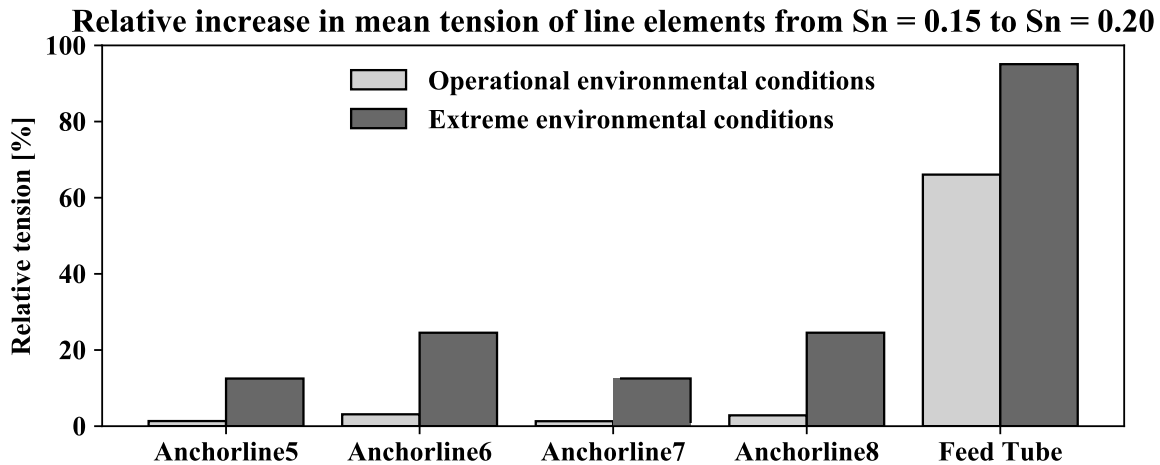


Figure 5.20: Relative increase in mean tensions of line elements between solidity ratios when $S_n = 0.15$ and $S_n = 0.20$.

From the figure, it is apparent that the relative difference in tensions for the environmental conditions is significant. The increase is rather apparent but indicates that for a higher solidity, there will be an increase in the environmental loading on a fish cage. The relative difference in the tensions on the feeding tube is very apparent at 70 % for the operational conditions and almost 100 % for the extreme conditions. However, the value increase is from 0.03 kN to 0.095 kN , which is not a significant increase in tension for the feeding tube. The increase in the anchor line tensions are not significant for the operational environmental conditions but appears to be considerably larger for

the extreme environmental conditions. The increase in tensions are symmetric for lines 5 and 7, and lines 6 and 8, as they are absorbing symmetric environmental loading for the same direction. In general, it seems that the behavior of the relative difference in tensions for the fish cage is similar to the operational and extreme environmental conditions, although substantially larger for the extreme environmental conditions.

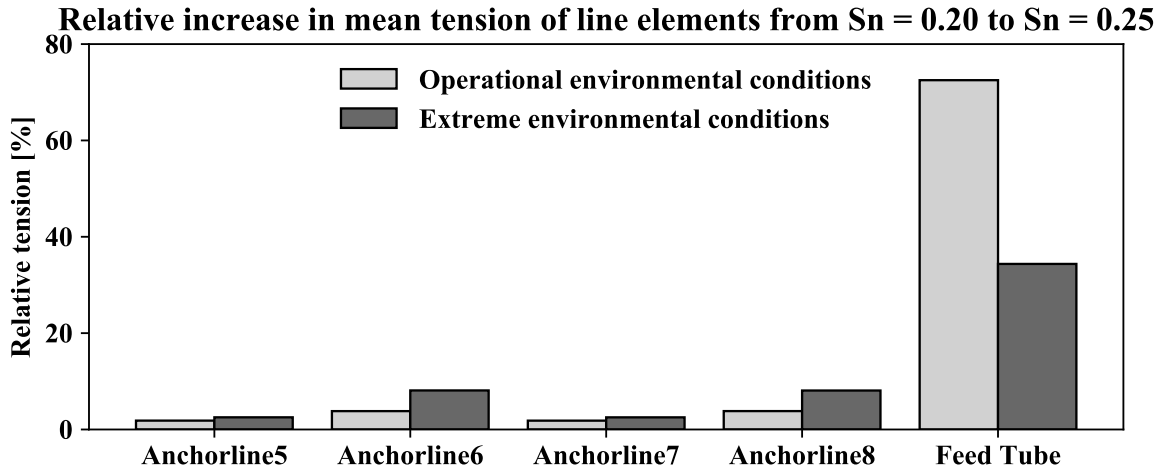


Figure 5.21: Relative increase in mean tension of line elements between solidity ratios when $S_n = 0.20$ and $S_n = 0.25$.

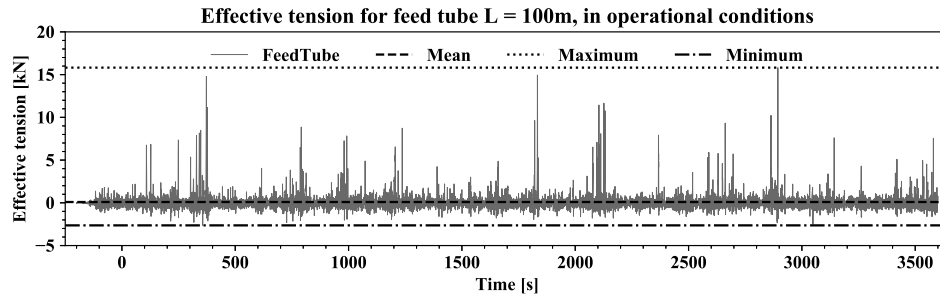
For the relative difference in tensions when comparing the net solidity ratio of $S_n = 0.20$ to $S_n = 0.25$ is similar to the difference from $S_n = 0.15$ to $S_n = 0.20$. The difference in solidity ratio indicates a significant increase in the loading on the feeding tube for the operational conditions. The increase in the feeding tube tension is measured to be increasing from 0.095 kN to 0.345 kN . This is a significant relative difference, but it is not a significant load on the feeding tube compared to the extreme environmental conditions. This trend is also visible for the bar diagrams showing effective tensions in the sub-figures of Fig. 5.19. As the environmental loading on the fish cage increases with a larger solidity ratio, the tensions in the mooring lines, restricting the displacement of the fish farm system will increase. Conversely, the mooring lines opposite will experience a reduction in tensions and the lines will become less taught. However, the increased tension on the feeding tube is noticeable, and the load absorption of the feeding tube seems to be a common denominator for the numerical simulations when the fish farm system is exposed to significant environmental loading.

5.2.4 Sensitivity Study on Feeding Tube Configurations

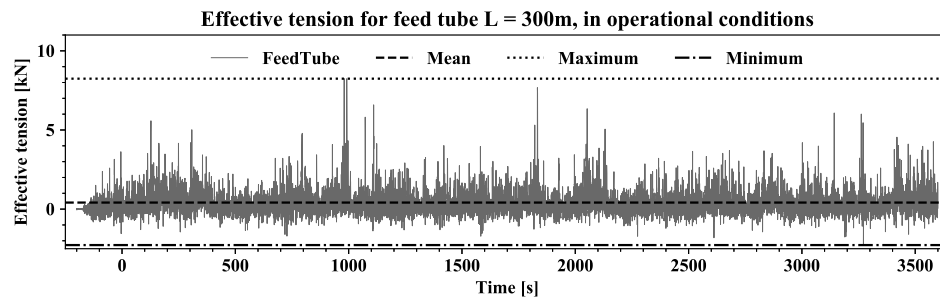
As seen in the numerical analysis in Section 5.2.3, there appears to be considerable loading effects for the feeding tube for various environmental conditions. In order to better understand the loading

mechanisms of the feeding tube, several simulations are run for different feeding tube lengths and environmental conditions. The effective tension in the feeding tube for the duration of the simulation is measured at the feed barge connection point, as this point is likely to be restraining the feeding tube motion and inducing tension in the tube.

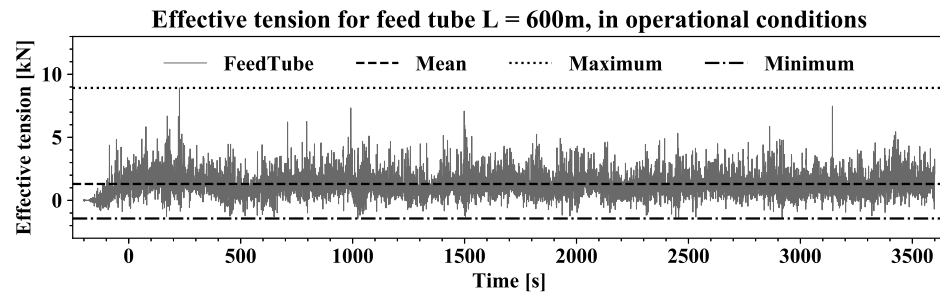
In the industry, long feeding tubes allow for access to a larger number of pens (fish cage installations) and allow for more spacing, but the feeding tube lengths can vary from a few hundred meters up to 1000 meters, depending on the needs of the facility [17]. For the purpose of the present study, it is interesting to see the characteristic behavior of the feeding tube for different lengths and environmental conditions. Hence, the different cases studied utilized feeding tubes of lengths $L_{1A,B} = 100\text{ m}$, $L_{2A,B} = 300\text{ m}$ and $L_{3A} = 600\text{ m}$ for the operational environmental conditions (denoted L_A) and the extreme environmental conditions (denoted L_B). The primary focus will be on the feeding tube loads in the time domain and the distribution of the axial forces along the arc length of the feeding tube. The effective tension on the feeding tube for various lengths in the time domain for the operational environmental conditions is shown in Fig. 5.22, and for the extreme environmental conditions is shown in Fig. 5.23. The effective tension is measured at the feeding tube connection point to the feed barge.



(a) $L_{1A} = 100\text{ m}$

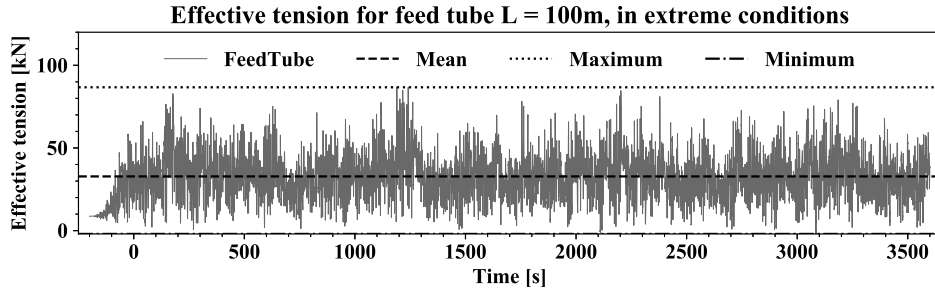


(b) $L_{2A} = 300\text{ m}$

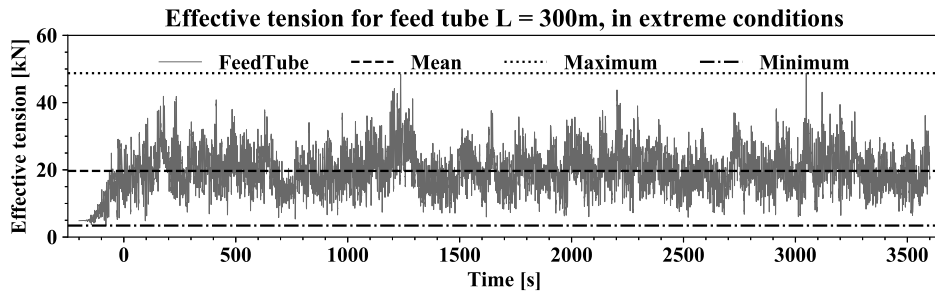


(c) $L_{3A} = 600\text{ m}$

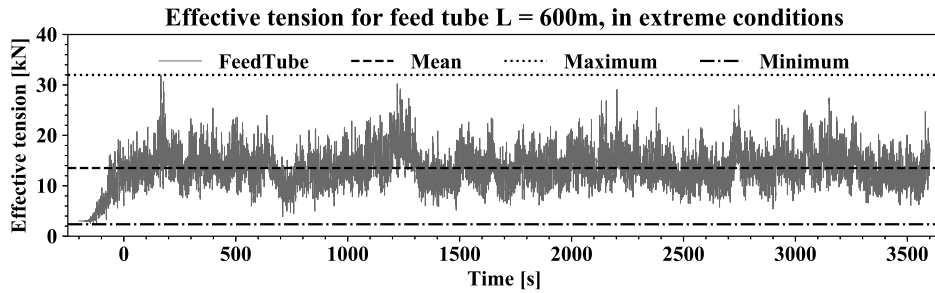
Figure 5.22: One hour (3600 seconds) time domain simulations for various feeding tube lengths in the operational environmental conditions.



(a) $L_{1B} = 100\text{ m}$



(b) $L_{1B} = 300\text{ m}$



(c) $L_{1B} = 600\text{ m}$

Figure 5.23: One hour (3600 seconds) time domain simulations for feeding tube length in the extreme environmental conditions.

As shown in Fig. 5.22, there is a clear trend that for the operational environmental conditions, the maximum tension in the feeding tube decreases in magnitude. However, the mean tension seems to increase slightly. A reason for this behavior could be that for longer feeding tubes, it is better suited to bend and deform, hence reducing the maximum tension in the feeding tube. Furthermore, even though the maximum tension decreases, the increase in feeding tube length contributes to a slightly larger mean tension in the tube due to current and wave loads. As the maximum effective tension is slightly lower, it may be an indication that longer feeding tubes utilize the elastic properties of the

feeding tube material better. Hence, for longer feeding tubes, the increased lengths allow for more flexibility between the feed vessel and the fish cage. As seen previously, the feeding tubes may still be experiencing snap loads as the feeding tube goes from being in slight compression to being in tension. Snap loads should be avoided as it can cause feeding tube ruptures and contributes to an increase in fatigue over time.

In Fig. 5.23, the feeding tube lengths seem to have a slight reduction in both average and maximum tension. The mean tension is reduced from 35 kN to 20 kN and furthermore down to 13 kN , which is a stepwise reduction of 40 %. Moreover, there appears to be a reduction in the maximum effective tension from 90 kN to 50 kN and a further reduction to 32 kN , also reductions of around 40%. The oscillations shown in the figures also seem to reduce slightly, which may be because for the full length of the feeding tube, the sections that experience wave loading do not transfer the loading to other sections. They behave elastically and follow the motion of the wave. The behavior is relatively similar to what is observed for the operational conditions and seems to confirm that longer feeding tubes increases flexibility and gives the feeding tube a better ability to utilize the elastic properties in the feeding tube. The simulations seem to suggest that the overall loading of the feeding tube seems to be slightly lower for longer tubes as the elastic properties are better utilized. However, the tensions are measured at the feed barge connection point, and it does not give an overview of the distribution of the bending moment and tensions along the feeding tube.

To better understand the distribution of tension and moment along the length of the feeding tube, Fig. 5.24 shows tension (left) and bending moment (right) along the arc length of the feeding tube for the lengths $L_{1A,B,C} = 100\text{ m}$, $L_{2A,B,C} = 300\text{ m}$ and $L_{3A,B,C} = 600\text{ m}$. Here, A is used to represent the operational conditions, B is for the extreme conditions, and C is for the feeding tube with a bend stiffener attached for the operational conditions. In the figures, 0 meters is at the connection to the feed barge, and the full length is at the connection to the fish cage.

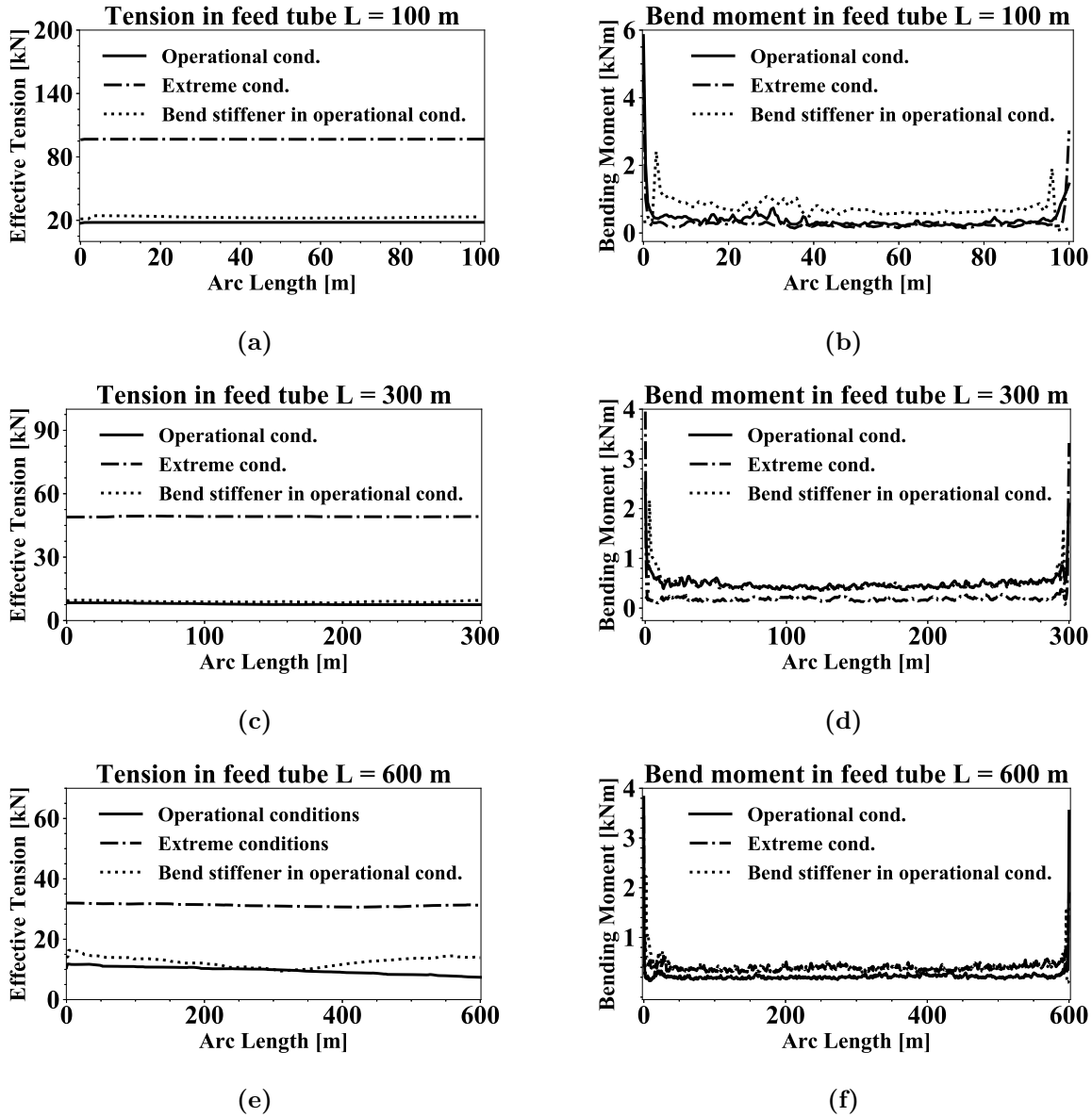


Figure 5.24: Distribution of effective tension (left) and bending moment (right) along the length of the feeding tube, for lengths of $L_{1A,B,C} = 100\text{ m}$ (a), $L_{2A,B,C} = 300\text{ m}$ (b), and $L_{3A,B,C} = 600\text{ m}$ (c) for the operational and extreme conditions.

As shown in Fig.5.24, the effective tensions in the feeding tube change little for the length of the feeding tube but seem to have a significant reduction in tension as the length of the feeding tube increases. Furthermore, for the extreme conditions and feeding tube length $L_{1B} = 100\text{ m}$ the feeding tube is experiencing a tension of 100 kN , which is reduced to 50 kN for $L_{2B} = 300\text{ m}$ and furthermore to 35 kN for a feeding tube length of $L_{3B} = 600\text{ m}$. This reduction in tension is quite substantial for the extreme conditions and could be caused by an improved ability for the feeding

tube to follow the motion of the waves without distributing tension to other segments of the tube in severe conditions. Different feeding tube lengths in the operational conditions have tension in the feeding tube ranging from 20 kN for $L_{1A} = 100\ m$, which is reduced to around 10 kN for feeding tube length of $L_{2A} = 300\ m$, but around 15 kN for a feeding tube length of $L_{3A} = 600\ m$ which is not a substantial difference in feeding tube loading for different lengths. However, from Fig. 5.24, it seems reasonable to conclude that the longer feeding tubes are better suited in absorbing the wave loads and has a better ability to roll in the motion of the waves, without resulting in tension for the connection points of the feeding tube. The bend stiffeners which are used to increase the stiffness of the feeding tube connection points do not seem to have a significant impact on the effective tension in the feeding tube. The bend stiffeners are designed to absorb the bending moment at the joints of the feeding tube, not axial force. An example of applications for bend stiffeners is in the oil and gas industry where bend stiffeners are used in flexible flowlines in the oil and gas industry, to increase bend radius and to avoid kinks in the line.

The bending moment in the feed tube for various lengths with and without bend stiffeners are shown in Table 5.9. From the bending moment in the right column of Fig.5.24, it appears that the bending moments differ little between the operational and extreme environmental conditions. The bend stiffener does seem to reduce the bending moments at the joints, hence fulfilling its purpose. However, the bending moment is slightly larger along the length of the feeding tube. The small peaks seen in the figures is at the tip of the bend stiffener, and the stiffener will be absorbing the bending moment, instead of the feeding tube. There appears to be a more even distribution of the bending moment along the feed tube, which could help reduce the maximum bending at the connection points for the feeding tube. The values of the bending moment from the right column of Fig.5.24, is given in Table 5.9 to provide a clear overview of the difference in bending moment with and without bend stiffener for different feeding tube lengths.

Table 5.9: Maximum bending moments in operational conditions without (L_A) and with (L_C) bend stiffener for feed barge, and fish cage connection points.

	Feed Barge Connection Point		Fish Cage Connection Point	
	L_A [kNm]	L_C [kNm]	L_A [kNm]	L_C [kNm]
$L_1 = 100\ m$	5.84	2.42	1.45	1.91
$L_2 = 300\ m$	2.51	2.17	2.09	1.60
$L_3 = 600\ m$	3.82	2.23	3.54	1.54

The right column of Fig.5.24 shows the bending moment along the full length of the feeding tube, where 0 meters on the axis is defined as the end connection the feed barge, and the opposite connection is to the fish cage. From Table 5.9, the bending moment values for the feed barge connection points are measured to be 5.84 kNm , 2.51 kNm , and 3.82 kNm for the three feeding tube lengths. These are reduced to 2.42 kNm , 2.17 kNm , and 2.23 kNm respectively, by attaching a bend stiffener the feeding tubes. This means that the bending moment for the feed barge is reduced by 58.56 %, 13.55 %, and 41.62 % of the original bending moment. The bending moment at the fish cage is found to be 1.45 kNm , 2.09 kNm , and 3.54 kNm without a bend stiffener, while with a bend stiffener attached the bending moments are found to be 1.91 kNm , 1.60 kNm , and 1.54 kNm , respectively. This corresponds to a relative difference in bending moment of an increase of 31.72 %, a decrease of 23.44 %, and a decrease of 56.49 %. In general, the trend appears that there is a decrease in the tension for the feeding tube when attaching a bend stiffener. However, for the fish cage connection, there is an increase of 31.72 %, and the reason for this is unclear. It is also observed that there is a significant difference in the bending moment for the feed barge compared to the fish cage. In conclusion, the general trend seems to be that the bending moment is reduced when a bend stiffener is applied. Hence it can be a useful application for harsh environmental conditions to reduce the bending at the end connection points for the feeding tube.

6 Conclusions and Future Work

6.1 Conclusions

The purpose of the present study is to investigate the loading effects on a fish farm with a feeding system. This is completed by establishing a numerical fish farm model using the OrcaFlex Application Programming Interface (API) and the programming language Python. Python in conjunction with the OrcaFlex API provides a versatile and flexible numerical modeling process. The purpose is to explore the environmental loading induced on a fish farm for the harsh environmental conditions of the North Sea. Understanding the loading mechanisms is important for the development of sustainable long term aquaculture operations in challenging environmental conditions. Hydrodynamic analysis to find the response amplitude operators and damping coefficients for a feed barge is performed. Three fish cage models are developed with varying mesh density. A convergence study is subsequently conducted for the fish cage models, and the volume deformation is calculated and compared to empirical values. The numerical fish farm model is used to study the response and loading for two different environmental conditions, varying wave and current directions, different solidity ratios, and feeding tube lengths. The following findings are observed in the time domain simulations:

1. A sensitivity study has been conducted for the volume reduction for the equivalent fish cage models with coarse (32 sections), medium (48 section), and fine (64 sections) mesh density, and a solidity ratio of 0.20. In the study, a difference of less than one percent in volume deformation for most current velocities, despite a significant difference in knot and twine elements. By comparing the steady current results to that of empirical studies, the fish cage models indicate satisfactory accuracy for all three models. Based on the findings, the coarse fish cage model is chosen because of the reduced computational requirement for the time domain simulations.
2. Time domain simulations for the operational environmental conditions found a 45 % increase in the mean anchor line tensions compared to the pre-tensioned values and some oscillations in the anchor line tensions. For the extreme environmental conditions, there is an increase of 100 % in the mean mooring line tension by comparison. Hence, there appears to be a significant difference between the operational and extreme environmental conditions. The feeding tube in the operational conditions has relatively large oscillations compared to the mean tension, which oscillates from negative to positive values. This loading mechanism

is likely to cause snap loads, which induce fatigue. Also, for the extreme environmental conditions, the mean tension is found to be considerably larger, with even larger oscillations compared to the operational conditions. This trend indicates significant axial forces and oscillations that will cause fatigue in the feeding tube over an extended period of time.

3. Time domain simulations for the fish farm with perpendicular (90°) wave and current directions relative to the system, has similar loading mechanisms for the anchor lines as for 0° wave and current direction. However, the feeding tube for the operational and extreme conditions have small tension and no significant oscillations. For alternate wave (90°) and current (0°) direction, it is found that current is the dominant load component. Anchor lines in the direction of the current had the largest anchor line tension. However, it is found that tension is distributed across the mooring system because of differing wave and current directions.
4. The sensitivity study on solidity ratio indicated that larger solidity ratios contribute to an increase in drag force. The increased drag force is reflected primarily in the increased tensions of the anchor lines and volume deformation of the fish cage. There will also be a reduction in the fresh seawater flowing through the cage. However, the findings for different solidity ratios are expressed in relative terms:
 - A comparison of the mooring line tensions between fish cage models of the net solidity ratio 0.15 and 0.20 indicate a difference of about 25 % for the anchor lines under extreme conditions. However, there is no significant increase for the operational conditions. There is also a significant relative increase in the feeding tube tension, but the numerical values are small.
 - A comparison of the mooring line tensions between numerical fish cage models of net solidity ratio 0.20 and 0.25 indicates a small relative difference in anchor line tensions of around 5 % and 10 % in the operational, and extreme conditions respectively. However, there is a significant relative increase in feeding tube tension of 75 % and 40 % in the operational and extreme conditions, respectively, but the numerical values are small.
5. In the feeding tube sensitivity study, it appears that, for longer feed tube lengths, the tube is more flexible and distributes the tension along the line better. This contributes to a reduction in axial tension along the arc length of the feeding tube, reducing tension by 50 % from 100 meters to 300 meters, and by 30 % from 300 meters to 600 meters in the operational and extreme conditions. Furthermore, there is a reduction in the bending moment at the end connections for longer feeding tubes. Attaching a bend stiffener reduces the bending moment

at the end connections. However, the mean bending moment is distributed along the length of the feeding tube and is slightly larger than without a bend stiffener.

6.2 Future Work

- In a more comprehensive study, a fish farm with multiple pens can be developed and analyzed using the OrcaFlex API. For a more complex fish farm system with higher mesh density for the fish cages, several feeding tubes, and mooring lines, it would be interesting to see the effect of the environmental conditions and the response of the system.
- A submersible fish farm model could be developed using the OrcaFlex API and analyzed for various environmental conditions. There are several advantages of a submerged system, such as reduced exposure to wave loads and lice. It would, therefore, be interesting to make a model of the fish farm developed by Atlantis Subsea Farming AS, to compare the difference in the response of the fish farm.
- A fish farm utilizing water-based feeding systems could have alternative configurations for the feeding tube, where the tube is submerged in the water. In such a configuration, the feeding tube would not be exposed to slamming loads and would not be exposed to wave motion. This could contribute to reducing environmental loading and is an interesting alternative to the air-based feeding systems used today.
- A study on the difference between environmental loading on a feeding tube utilizing water-based feeding compared to air based feeding methods. Alternate installation arrangements could potentially contribute to reducing loading effects and reduce fatigue on the feeding tube.
- Using the OrcaFlex API, variable drag coefficients for the twines could easily be implemented for different line types, depending on wave and current directions. This could be a method to compensate for shielding effects and turbulent flow around the fish cage. Computational fluid dynamics could potentially be applied to study these effects.
- OrcaFlex has a built-in function utilizing rainflow counting, which can be used to estimate fatigue limit states (FLS) for a fish farm. In a future study, it would be interesting to study the fish farm system and fatigue loading. This could also be an important analysis to find optimal configurations to achieve a sustainable fish farm for fish farms exposed to harsh environmental conditions.

References

- [1] Ministry of Trade, Industry and Fisheries (2019). Predictable and environmentally sustainable growth in salmon- and troutfarming, Meld. St. 16 (2014-2015).
- [2] Food and Agriculture Organization of the United Nations (2019), National Aquaculture Sector Overview, http://www.fao.org/fishery/countrysector/naso_norway/en. Accessed on: 24.05.2019.
- [3] Directorate of fisheries (2019), 13 000 tonn død laks i nord, <https://www.fiskeridir.no/Akvakultur/Nyheter/2019/0519/13-000-tonn-laks-doed-i-nord>. Accessed on: 24.05.2019.
- [4] Directorate of fisheries (2019), Akvakultur lokaliteter, kartverktøy, <https://kart.fiskeridir.no/>. Accessed on: 24.05.2019.
- [5] Akva Group, (2019). Cage Farming Aquaculture. <https://www.akvagroup.com/pen-based-aquaculture>. Accessed on: 25.05.2019.
- [6] Nordlaks, (2019). Havfarm. <https://www.nordlaks.no/havfarm>. Accessed on: 25.05.2019.
- [7] Salmar (2019). Offshore fish farming, A new era in fish farming is on its way. <https://www.salmar.no/en/offshore-fish-farming-a-new-era/>. Accessed on: 25.05.2019.
- [8] Akva Group (2019). Atlantis Subsea Farming has applied for six development licences. <https://www.akvagroup.com/nyheter/arkiv/nyheter-view/atlantis-subsea-farming-as-har-s%C3%B8kt-om-seks-utviklingskonsesjoner->. Accessed on: 25.05.2019.
- [9] O. T. Gudmestad (2015). Marine Technology and Operations, Theory & Practice. WIT Press.
- [10] T. Kristiansen, and O. M. Faltinsen (2012). Modeling of current loads on aquaculture net cages. *Journal of Fluids and Structures*, 34, 218-235.
- [11] DNV-GL (2014). Environmental conditions and environmental loads. DNV-RP-C205. DNV GL.
- [12] Calqlata, (2018). RAO Calculator (response amplitude operators). <http://www.calqlata.com/productpages/00081-help.html>. Accessed on: 29.11. 2018
- [13] W. W. Massie, and J. M. J. Journée, (2001). Offshore hydromechanics. Delft University of Technology: Delft, The Netherlands.

- [14] Akva Group. (2018). Wavemaster Feed Barge Specifications. <http://www.akvagroup.com/Products/Cage%20farming%20aquaculture%20products/Feed%20Barges/Wavemaster>. Accessed on: 29.11.2018.
- [15] A. H. Tectet, "Morrison's equation," Massachusetts Institute of Technology, Cambridge, MA, USA, 2004, Online Course.
- [16] T.-J. Xu et al. (2011). Analysis of hydrodynamic behaviors of gravity net cage in irregular waves. *Ocean Engineering*, 38(13), 1545-1554.
- [17] Akva Group, (2019). Pipes and Pipework. <https://www.akvagroup.com/pen-based-aquaculture/pipes-and-pipework>. Accessed on: 28.05.2019.
- [18] Kyst (2018). Mener å ha løst statistisk problem. <https://www.kyst.no/article/mener-arving-ha-loslash-st-statisk-problem/>. Accessed on: 28.05.2019.
- [19] P. E. Thomassen (2008). Methods for Dynamic Response Analysis and Fatigue Life Estimation of Floating Fish Cages. Phd thesis, NTNU.
- [20] DNV-GL (2017). Recommended practice: Modelling and analysis of marine operations, DNVGL-RPN103.
- [21] DNV-GL (2010a). Sesam user manual - Wadam (Wave Analysis by Diffraction and Morison Theory). Det Norske Veritas Software, Høvik.
- [22] A. H. Magnuson (2010). Nonlinear analysis of heavy-lift barge roll motion. *Oceans 2010 MTS/IEEE*, 0197-7385.
- [23] T. Haver (2016). Roll Damping Prediction for Vessels; Sheerlegs and Barges: A study on effects of viscosity and roll-sway coupling on the roll damping coefficient. Graduation thesis, TU Delft.
- [24] Egersund Net, (2019). FoU – Forskning og utvikling. https://www.egersundnet.no/fou?fbclid=IwAR0mqf3-qPaWL7_o6hP3qHV-DuU8toj5ZiOU8Y8-ludKHEBspkDWfn7m-hk. Accessed on 01.06.2019
- [25] Akva Group, (2019). Cage Farming Aquaculture. <https://www.akvagroup.com/pen-based-aquaculture/pens-nets/anchoring-and-mooring>. Accessed on: 25.05.2019.
- [26] Aqualine (2019). Available: <https://aqualine.no/en/products/cages>. Accessed on: 12.06.19

- [27] Y. Chang (2017). Dynamic analysis of gravity based fish cage. Graduate Thesis, University of Stavanger.
- [28] D. Kristiansen et al. (2015). “Numsim - numerical simulation of complex systems involving interaction between elements with large and varying stiffness properties,” SINTEF Ocean AS, Tech. Rep., 2015.
- [29] Orcina Ltd. (2018). Orcina: OrcaFlex Documentation.
- [30] A. Gjuka (2017). Dynamic Analysis of Feed Pipes for Fish Farming in Open Sea. Graduation thesis, University of Stavanger.
- [31] Marine fish farms - Requirements for design, dimensioning, production, installation and operation (Corrigendum AC:2004 incorporated) (2003). NS9415. Norsk Standardiseringsforbund.
- [32] Y. Shen et al. (2018). Numerical and experimental investigations on mooring loads of a marine fish farm in waves and current. *Journal of Fluids and Structures*, 79, 115-136.
- [33] Trelleborg (2019). Seismic Split Bend Stiffener. <https://www.trelleborg.com/en/offshore/products/bend-control-solutions/seismic-split-bend-stiffener>. Accessed on: 29.05.2019.

A Appendix

A.1 OrcaFlex API Codes - Numerical Model of 64 Section Cage

```
import OrcFxAPI
import math

#Floor
Spacing1 = - 2
Spacing2 = - 1
Spacing3 = - 1.875
ConeSection = - 13 #13m conical section under base section (15m)

Floor1 = 0 * Spacing3
Floor2 = 1 * Spacing3
Floor3 = 2 * Spacing3
Floor4 = 3 * Spacing3
Floor5 = 4 * Spacing3
Floor6 = 5 * Spacing3
Floor7 = 6 * Spacing3
Floor8 = 7 * Spacing3
Floor9 = 8 * Spacing3

model = OrcFxAPI.Model('Start file 32 sections.dat')
Seabed = 100
sections = 32
radius = 25.0
ycoord = 0
BotCoord = Floor9
BaseHeight = - 24
BaseDepth = BotCoord + BaseHeight
angle = math.radians(360.0 / sections)
angle2 = (360/sections)
ringlength = 0.80*(radius * math.sin(0.5*angle))
TopRingLength = math.sin(angle/2)*(radius)
BottomRingLength = TopRingLength * 0.85
LayerKnot1X = (1/15) * (radius)
LayerKnot2X = (2/15) * (radius)
LayerKnot3X = (3/15) * (radius)
LayerKnot4X = (4/15) * (radius)
LayerKnot5X = (5/15) * (radius)
LayerKnot6X = (6/15) * (radius)
LayerKnot7X = (7/15) * (radius)
LayerKnot8X = (8/15) * (radius)
LayerKnot9X = (9/15) * (radius)
LayerKnot10X = (10/15) * (radius)
LayerKnot11X = (11/15) * (radius)
LayerKnot12X = (12/15) * (radius)
LayerKnot13X = (13/15) * (radius)
LayerKnot14X = (14/15) * (radius)
#Layer 1 to 15 from center to find z coordinate
LayerKnot1Z = (BotCoord + 1/15 * ConeSection)
```

```

LayerKnot2Z = (BotCoord + 2/15 * ConeSection)
LayerKnot3Z = (BotCoord + 3/15 * ConeSection)
LayerKnot4Z = (BotCoord + 4/15 * ConeSection)
LayerKnot5Z = (BotCoord + 5/15 * ConeSection)
LayerKnot6Z = (BotCoord + 6/15 * ConeSection)
LayerKnot7Z = (BotCoord + 7/15 * ConeSection)
LayerKnot8Z = (BotCoord + 8/15 * ConeSection)
LayerKnot9Z = (BotCoord + 9/15 * ConeSection)
LayerKnot10Z = (BotCoord + 10/15 * ConeSection)
LayerKnot11Z = (BotCoord + 11/15 * ConeSection)
LayerKnot12Z = (BotCoord + 12/15 * ConeSection)
LayerKnot13Z = (BotCoord + 13/15 * ConeSection)
LayerKnot14Z = (BotCoord + 14/15 * ConeSection)
LayerKnot15Z = (BotCoord + 15/15 * ConeSection)
#Length of ring 1 to 10
Ring1 = (2.4529 - 0.0) * 1.00
Ring2 = (2.2897 - 0.0) * 1.00
Ring3 = (2.1266 - 0.0) * 1.00
Ring4 = (1.9623 - 0.0) * 1.00
Ring5 = (1.7992 - 0.0) * 1.00
Ring6 = (1.6360 - 0.0) * 1.00
Ring7 = (1.4718 - 0.0) * 1.00
Ring8 = (1.3070 - 0.0) * 1.00
Ring9 = (1.1460 - 0.0) * 1.00
Ring10 = (0.9812 - 0.0) * 1.00
Ring11 = (0.8180 - 0.0) * 1.00
Ring12 = (0.6538 - 0.0) * 1.00
Ring13 = (0.4906 - 0.0) * 1.00
Ring14 = (0.3274 - 0.0) * 1.00
Ring15 = (0.1632 - 0.0) * 1.00
LengthNetB = (1.88 - 0.0) * 1.00

BottomlineLength = (math.sqrt((radius*radius)+(BaseHeight*BaseHeight))*1/16)*0.75
AnchorlineLength = 443.851
MooringlineBLength = 73.128
AnchorCorners = radius + 450
CageMooringlineLength1 = 30.473 #Long Mooring
CageMooringlineLength2 = 27.091 #Short Mooring
VerticalMooringline = 5.50
prebend = 0.04

def rotateposition(oldX, oldY):
    newX = oldX * math.cos(angle*pos) - oldY * math.sin(angle*pos)
    newY = oldX * math.sin(angle*pos) + oldY * math.cos(angle*pos)
    return newX, newY

def createbuoyA(pos, name, x, y, z): #creates top floats
    float = model.CreateObject(OrcFxAPl.ot6DBuoy, name+str(pos))
    float.BuoyType = "Spar Buoy"
    float.DegreesOfFreedomInStatics = 'All'
    newX, newY = rotateposition(x,y)
    float.InitialX, float.InitialY, float.InitialZ = newX, newY, z
    float.InitialRotation1 = 90

```

```

float.InitialRotation2 = -180 + pos*math.degrees(angle)
float.InitialRotation3 = 0
float.mass
float.MassMomentOfInertiaX = 0.05
float.MassMomentOfInertiaY = 0.058
float.MassMomentOfInertiaZ = 0.025
float.CylinderOuterDiameter[0] = 1.0
float.CylinderInnerDiameter[0] = 0.0
float.CylinderLength[0] = 1.0
float.StackBaseCentreZ = -0.5
float.CenterOfMassX, float.CenterOfMassY, float.CenterOfMassZ = 0, 0, 0

def createbuoyB(pos, name, x, y, z): #creates midtopknots (is a float)
float = model.CreateObject(OrcFxAPI.ot6DBuoy, name+str(pos))
float.BuoyType = "Spar Buoy"
float.DegreesOfFreedomInStatics = 'All'
newX, newY = rotateposition(x,y)
realX = newX * math.cos(angle / 2) - newY * math.sin(angle / 2)
realY = newX * math.sin(angle / 2) + newY * math.cos(angle / 2)
float.InitialX, float.InitialY, float.InitialZ = realX, realY, z
float.InitialRotation1 = 90
float.InitialRotation2 = -180 + pos*math.degrees(angle) + (angle2/2)
float.InitialRotation3 = 0
float.mass = 0.05
float.MassMomentOfInertiaX = 0.058
float.MassMomentOfInertiaY = 0.058
float.MassMomentOfInertiaZ = 0.025
float.CylinderOuterDiameter[0] = 1.0
float.CylinderInnerDiameter[0] = 0.0
float.CylinderLength[0] = 1.0
float.StackBaseCentreZ = -0.5
float.CenterOfMassX, float.CenterOfMassY, float.CenterOfMassZ = 0, 0, 0

def createknotA(pos, name, x, y, z): #creates net knots
knot = model.CreateObject(OrcFxAPI.ot3DBuoy, name+str(pos))
newX, newY = rotateposition(x,y)
knot.InitialX, knot.InitialY, knot.InitialZ = newX, newY, z
knot.Mass, knot.Volume, knot.Height = 0.0001, 0.0001, 0.1

def createmidknot(pos, name, x, y, z):
knot = model.CreateObject(OrcFxAPI.ot3DBuoy, name+str(pos))
newX, newY = rotateposition(x,y)
realX = newX * math.cos(angle / 2) - newY * math.sin(angle / 2)
realY = newX * math.sin(angle / 2) + newY * math.cos(angle / 2)
knot.InitialX, knot.InitialY, knot.InitialZ = realX, realY, z
knot.Mass, knot.Volume, knot.Height = 0.0001, 0.0001, 0.1

def createBotBuoy(pos, name, x, y, z): #creates net knots
float = model.CreateObject(OrcFxAPI.ot6DBuoy, name+str(pos))
float.BuoyType = "Lumped Buoy"
newX, newY = rotateposition(x,y)
float.InitialX, float.InitialY, float.InitialZ = newX, newY, z
float.InitialRotation1 = 90
float.InitialRotation2 = -180 + pos*math.degrees(angle) + (angle2/2)
float.InitialRotation3 = 0
float.mass = 0.25

```

```

float.MassMomentOfInertiaX = 0.01
float.MassMomentOfInertiaY = 0.01
float.MassMomentOfInertiaZ = 0.01
float.CenterOfMassX, float.CenterOfMassY, float.CenterOfMassZ = 0, 0, -0.10
float.NumberOfVertices, float.volume, float.Height = 0, 0.0001, 0.1
float.DegreesOfFreedomInStatics = 'All'
def createMidbotBuoy(pos, name, x, y, z):
float = model.CreateObject(OrcFxAPI.ot6DBuoy, name+str(pos))
float.BuoyType = "Lumped Buoy"
newX, newY = rotateposition(x,y)
realX = newX * math.cos(angle / 2) - newY * math.sin(angle / 2)
realY = newX * math.sin(angle / 2) + newY * math.cos(angle / 2)
float.InitialX, float.InitialY, float.InitialZ = realX, realY, z
float.InitialRotation1 = 90
float.InitialRotation2 = -180 + pos*math.degrees(angle) + (angle/2)
float.InitialRotation3 = 0
float.mass = 0.25
float.MassMomentOfInertiaX = 0.01
float.MassMomentOfInertiaY = 0.01
float.MassMomentOfInertiaZ = 0.01
float.CenterOfMassX, float.CenterOfMassY, float.CenterOfMassZ = 0, 0, -0.10
float.NumberOfVertices, float.volume, float.Height = 0, 0.050, 0.3
float.DegreesOfFreedomInStatics = 'All'
def createnetA(pos, name, connA, connB, offset):
line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
line.LineType[0] = "Net line type"
line.Length[0] = - Spacing3
line.TargetSegmentLength[0] = (- Spacing3)
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = connA
line.EndAX, line.EndAY, line.EndAZ = 0, offset, 0
line.EndBConnection = connB
line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createnetB(pos, name, connA, connB, offset):
line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
line.LineType[0] = "Net line type"
line.Length[0] = - Spacing2 * 1.87
line.TargetSegmentLength[0] = (- Spacing2 * 1.87)/4
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = connA
line.EndAX, line.EndAY, line.EndAZ = 0, offset, 0
line.EndBConnection = connB
line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0

#Ring Elements
def createtopringa(sections, name, connA, connB, OAX, OAY, OAZ, OBX, OBY, OBZ):
line = model.CreateObject(OrcFxAPI.otLine, name)
line.LineType[0] = "Ring line type"
line.Length[0] = Ring1
line.TargetSegmentLength[0] = 1.6
line.PreBendY[0] = 0.04
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = connA

```

```

line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
line.EndBConnection = connB
line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
#Orientation of original line
line.EndAAzimuth,line.EndADeclination, line.EndAGamma = OAX, OAY, OAZ
line.EndBAzimuth,line.EndBDeclination, line.EndBGamma = OBX, OBY, OBZ
#Including Torsion and infinite twisting stiffness
line.IncludeTorsion = 'Yes'
line.EndATwistingStiffness = OrcFxAPI.OrcinaInfinity()
line.EndBTwistingStiffness = OrcFxAPI.OrcinaInfinity()
#Introducing infinite bending Stiffness in the top ring line
line.EndAxBendingStiffness = OrcFxAPI.OrcinaInfinity()
line.EndBxBendingStiffness = OrcFxAPI.OrcinaInfinity()
line.EndAyBendingStiffness = OrcFxAPI.OrcinaInfinity()
line.EndByBendingStiffness = OrcFxAPI.OrcinaInfinity()
def createtopringb(sections, name, connA, connB, OAX, OAY, OAZ, OBX, OBY, OBZ):
line = model.CreateObject(OrcFxAPI.otLine, name)
line.LineType[0] = "Ring line type"
line.Length[0] = Ring1
line.TargetSegmentLength[0] = 1.6
line.PreBendY[0] = 0.04
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = connA
line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
line.EndBConnection = connB
line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
line.EndAAzimuth,line.EndADeclination, line.EndAGamma = OAX, OAY, OAZ
line.EndBAzimuth,line.EndBDeclination, line.EndBGamma = OBX, OBY, OBZ
#Including Torsion and infinite twisting stiffness
line.IncludeTorsion = 'Yes'
line.EndATwistingStiffness = OrcFxAPI.OrcinaInfinity()
line.EndBTwistingStiffness = OrcFxAPI.OrcinaInfinity()
#Introducing infinite bending Stiffness in the top ring line
line.EndAxBendingStiffness = OrcFxAPI.OrcinaInfinity()
line.EndBxBendingStiffness = OrcFxAPI.OrcinaInfinity()
line.EndAyBendingStiffness = OrcFxAPI.OrcinaInfinity()
line.EndByBendingStiffness = OrcFxAPI.OrcinaInfinity()
def createbottomcollar(pos, name, connA, connB, OAX, OAY, OAZ, OBX, OBY, OBZ):
line = model.CreateObject(OrcFxAPI.otLine, name)
line.LineType[0] = "Bottom ring line type"
line.Length[0] = Ring1
line.TargetSegmentLength[0] = 1.6
line.PreBendY[0] = 0.04
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = connA
line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
line.EndBConnection = connB
line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
line.EndAAzimuth,line.EndADeclination, line.EndAGamma = OAX, OAY, OAZ
line.EndBAzimuth,line.EndBDeclination, line.EndBGamma = OBX, OBY, OBZ
#Including Torsion and infinite twisting stiffness
line.IncludeTorsion = 'Yes'
line.EndATwistingStiffness = OrcFxAPI.OrcinaInfinity()

```

```

line.EndBTwistingStiffness = OrcFxAPI.OrcinaInfinity()
#Introducing infinite bending Stiffness in the top ring line
line.EndAxBendingStiffness = OrcFxAPI.OrcinaInfinity()
line.EndBxBendingStiffness = OrcFxAPI.OrcinaInfinity()
line.EndAyBendingStiffness = OrcFxAPI.OrcinaInfinity()
line.EndByBendingStiffness = OrcFxAPI.OrcinaInfinity()
def createbotring(pos, name, connA, connB): #creates bottom ring sections
line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
line.LineType[0] = "Net line type"
line.Length[0] = Ring1
line.TargetSegmentLength[0] = Ring1
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = connA
line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
line.EndBConnection = connB
line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasering1(pos, name, connA, connB):
line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
line.LineType[0] = "Net line type"
line.Length[0] = Ring2
line.TargetSegmentLength[0] = Ring2
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = connA
line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
line.EndBConnection = connB
line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasering2(pos, name, connA, connB):
line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
line.LineType[0] = "Net line type"
line.Length[0] = Ring3
line.TargetSegmentLength[0] = Ring3
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = connA
line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
line.EndBConnection = connB
line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasering3(pos, name, connA, connB):
line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
line.LineType[0] = "Net line type"
line.Length[0] = Ring4
line.TargetSegmentLength[0] = Ring4
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = connA
line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
line.EndBConnection = connB
line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasering4(pos, name, connA, connB):
line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
line.LineType[0] = "Net line type"
line.Length[0] = Ring5
line.TargetSegmentLength[0] = Ring5
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = connA

```

```

    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasing5(pos, name, connA, connB):
    line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
    line.LineType[0] = "Net line type"
    line.Length[0] = Ring6
    line.TargetSegmentLength[0] = Ring6
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasing6(pos, name, connA, connB):
    line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
    line.LineType[0] = "Net line type"
    line.Length[0] = Ring7
    line.TargetSegmentLength[0] = Ring7
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasing7(pos, name, connA, connB):
    line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
    line.LineType[0] = "Net line type"
    line.Length[0] = Ring8
    line.TargetSegmentLength[0] = Ring8
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasing8(pos, name, connA, connB):
    line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
    line.LineType[0] = "Net line type"
    line.Length[0] = Ring9
    line.TargetSegmentLength[0] = Ring9
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasing9(pos, name, connA, connB):
    line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
    line.LineType[0] = "Net line type"
    line.Length[0] = Ring10
    line.TargetSegmentLength[0] = Ring10
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0

```



```

def createbasering10(pos, name, connA, connB):
    line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
    line.LineType[0] = "Net line type"
    line.Length[0] = Ring11
    line.TargetSegmentLength[0] = Ring11
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasering11(pos, name, connA, connB):
    line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
    line.LineType[0] = "Net line type"
    line.Length[0] = Ring12
    line.TargetSegmentLength[0] = Ring12
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasering12(pos, name, connA, connB):
    line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
    line.LineType[0] = "Net line type"
    line.Length[0] = Ring13
    line.TargetSegmentLength[0] = Ring13
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasering13(pos, name, connA, connB):
    line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
    line.LineType[0] = "Net line type"
    line.Length[0] = Ring14
    line.TargetSegmentLength[0] = Ring14
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createbasering14(pos, name, connA, connB): #creates bottom ring sections
    line = model.CreateObject(OrcFxAPI.otLine, name+str(pos))
    line.LineType[0] = "Net line type"
    line.Length[0] = Ring15
    line.TargetSegmentLength[0] = Ring15
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createweightlineA(name, EndA, x, y, z): #creates lower line with weight attachment
    line = model.CreateObject(OrcFxAPI.otLine, name)

```

```

line.LineType[0], line.Length[0], line.TargetSegmentLength[0] = "Net line type", 2, 2
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = EndA
line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
line.EndBConnection = 'Free'
line.EndBX, line.EndBY, line.EndBZ = x, y, z
line.NumberOfAttachments = 1
line.AttachmentType[0], line.Attachmentz[0] = "500kg Weight", 2
def createweightlineB(name, EndA, x, y, z): #creates lower line with weight attachment
line = model.CreateObject(OrcFxAPI.otLine, name)
line.LineType[0], line.Length[0], line.TargetSegmentLength[0] = "Net line type", 2, 2
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = EndA
line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
line.EndBConnection = 'Free'
line.EndBX, line.EndBY, line.EndBZ = x, y, z
line.NumberOfAttachments = 1
line.AttachmentType[0], line.Attachmentz[0] = "1000kg Weight", 2

#Mooring
def createmooring(pos, name, connA, x, y, z): #creates inner and outer moorings
line = model.CreateObject(OrcFxAPI.otLine, name)
line.LineType[0] = "Mooring line type"
line.Length[0] = AnchorlineLength
line.TargetSegmentLength[0] = 30
line.IncludeSeabedFrictionInStatics = 'No'
line.EndAConnection = connA
line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
line.EndBConnection = 'Anchored'
line.EndBX, line.EndBY, line.EndBZ = x, y, z
def createmooringknot(name, x, y, z): #creates net knots
knot = model.CreateObject(OrcFxAPI.ot3DBuoy, name)
knot.InitialX, knot.InitialY, knot.InitialZ = x, y, z
knot.Mass, knot.Volume, knot.Height = 0.0001, 0.0001, 0.1
def createmooringbuoy(name, x, y, z): #creates net knots
float = model.CreateObject(OrcFxAPI.ot6DBuoy, name)
float.BuoyType = "Lumped Buoy"
float.InitialX, float.InitialY, float.InitialZ = x, y, z
float.InitialRotation1 = 0
float.InitialRotation2 = 0
float.InitialRotation3 = 0
float.mass = 0.2
float.MassMomentOfInertiaX = 0.01
float.MassMomentOfInertiaY = 0.01
float.MassMomentOfInertiaZ = 0.01
float.CenterOfMassX, float.CenterOfMassY, float.CenterOfMassZ = 0, 0, -0.10
float.NumberOfVertices, float.volume, float.Height = 0, 0.0001, 0.1
float.Connection = 'Free'
float.DegreesOfFreedomInStatics = 'X,Y,Z'
def createfloatingbuoy(pos, name, x, y, z): #creates top floats
float = model.CreateObject(OrcFxAPI.ot6DBuoy, name)
float.BuoyType = "Spar Buoy"
float.DegreesOfFreedomInStatics = 'X,Y,Z'

```

```

float.InitialX, float.InitialY, float.InitialZ = x, y, z
float.InitialRotation1 = 0
float.InitialRotation2 = 0
float.InitialRotation3 = 0
float.mass = 0.5
float.MassMomentOfInertiaX = 0.292
float.MassMomentOfInertiaY = 0.292
float.MassMomentOfInertiaZ = 1.5
float.StackBaseCentreZ = -0.5
float.CylinderOuterDiameter[0] = 2.0
float.CylinderInnerDiameter[0] = 0.0
float.CylinderLength[0] = 2.0
float.Connection = 'Free'
def createmooringlinea(pos, name, connA, connB):
    line = model.CreateObject(OrcFxAPI.otLine, name)
    line.LineType[0] = "Mooring line type"
    line.Length[0] = VerticalMooringline
    line.TargetSegmentLength[0] = VerticalMooringline/2
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, 0, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createmooringlineb(pos, name, connA, connB, offset):
    line = model.CreateObject(OrcFxAPI.otLine, name)
    line.LineType[0] = "Mooring line type"
    line.Length[0] = MooringlineBLength
    line.TargetSegmentLength[0] = 8
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, offset, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createmooringlinecage1(pos, name, connA, connB, offset):
    line = model.CreateObject(OrcFxAPI.otLine, name + str(pos))
    line.LineType[0] = "Mooring line type"
    line.Length[0] = CageMooringlineLength1
    line.TargetSegmentLength[0] = 4
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, offset, 0
    line.EndBConnection = connB
    line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0
def createmooringlinecage2(pos, name, connA, connB, offset):
    line = model.CreateObject(OrcFxAPI.otLine, name + str(pos))
    line.LineType[0] = "Mooring line type"
    line.Length[0] = CageMooringlineLength2
    line.TargetSegmentLength[0] = 4
    line.IncludeSeabedFrictionInStatics = 'No'
    line.EndAConnection = connA
    line.EndAX, line.EndAY, line.EndAZ = 0, offset, 0

```

```

line.EndBConnection = connB
line.EndBX, line.EndBY, line.EndBZ = 0, 0, 0

#Main code
#Top Ring
for pos in range(sections):
    #create all buoys - Buoy eller float A - Top Floor
    createbuoyB(pos, 'Midtopknot', radius, ycoord, Floor1)
    createbuoyA(pos, 'Float', radius, ycoord, Floor1)
    #First floor
    createmidknot(pos, 'Midbotknot1', radius, ycoord, Floor2)
    createknotA(pos, 'Botknot1', radius, ycoord, Floor2)
    #Second floor
    createmidknot(pos, 'Midbotknot2', radius, ycoord, Floor3)
    createknotA(pos, 'Botknot2', radius, ycoord, Floor3)
    #Third floor
    createmidknot(pos, 'Midbotknot3', radius, ycoord, Floor4)
    createknotA(pos, 'Botknot3', radius, ycoord, Floor4)
    #Fourth floor
    createmidknot(pos, 'Midbotknot4', radius, ycoord, Floor5)
    createknotA(pos, 'Botknot4', radius, ycoord, Floor5)
    #Fifth floor
    createmidknot(pos, 'Midbotknot5', radius, ycoord, Floor6)
    createknotA(pos, 'Botknot5', radius, ycoord, Floor6)
    #Sixth floor
    createmidknot(pos, 'Midbotknot6', radius, ycoord, Floor7)
    createknotA(pos, 'Botknot6', radius, ycoord, Floor7)
    #Seventh floor
    createmidknot(pos, 'Midbotknot7', radius, ycoord, Floor8)
    createknotA(pos, 'Botknot7', radius, ycoord, Floor8)
    #Eight floor
    createMidbotBuoy(pos, 'Midbotknot8', radius, ycoord, Floor9)
    createBotBuoy(pos, 'Botknot8', radius, ycoord, Floor9)
    # Layered bottom knots from layer 1 to 7
    createmidknot(pos, 'Layerknotb1', LayerKnot14X, ycoord, LayerKnot1Z)
    createknotA(pos, 'Layerknota1', LayerKnot14X, ycoord, LayerKnot1Z)
    createmidknot(pos, 'Layerknotb2', LayerKnot13X, ycoord, LayerKnot2Z)
    createknotA(pos, 'Layerknota2', LayerKnot13X, ycoord, LayerKnot2Z)
    createmidknot(pos, 'Layerknotb3', LayerKnot12X, ycoord, LayerKnot3Z)
    createknotA(pos, 'Layerknota3', LayerKnot12X, ycoord, LayerKnot3Z)
    createmidknot(pos, 'Layerknotb4', LayerKnot11X, ycoord, LayerKnot4Z)
    createknotA(pos, 'Layerknota4', LayerKnot11X, ycoord, LayerKnot4Z)
    createmidknot(pos, 'Layerknotb5', LayerKnot10X, ycoord, LayerKnot5Z)
    createknotA(pos, 'Layerknota5', LayerKnot10X, ycoord, LayerKnot5Z)
    createmidknot(pos, 'Layerknotb6', LayerKnot9X, ycoord, LayerKnot6Z)
    createknotA(pos, 'Layerknota6', LayerKnot9X, ycoord, LayerKnot6Z)
    createmidknot(pos, 'Layerknotb7', LayerKnot8X, ycoord, LayerKnot7Z)
    createknotA(pos, 'Layerknota7', LayerKnot8X, ycoord, LayerKnot7Z)
    createmidknot(pos, 'Layerknotb8', LayerKnot7X, ycoord, LayerKnot8Z)
    createknotA(pos, 'Layerknota8', LayerKnot7X, ycoord, LayerKnot8Z)
    createmidknot(pos, 'Layerknotb9', LayerKnot6X, ycoord, LayerKnot9Z)
    createknotA(pos, 'Layerknota9', LayerKnot6X, ycoord, LayerKnot9Z)
    createmidknot(pos, 'Layerknotb1000', LayerKnot5X, ycoord, LayerKnot10Z)

```

```

createknotA(pos, 'Layerknota1000', LayerKnot5X, ycoord, LayerKnot10Z)
createmidknot(pos, 'Layerknotb1100', LayerKnot4X, ycoord, LayerKnot11Z)
createknotA(pos, 'Layerknota1100', LayerKnot4X, ycoord, LayerKnot11Z)
createmidknot(pos, 'Layerknotb1200', LayerKnot3X, ycoord, LayerKnot12Z)
createknotA(pos, 'Layerknota1200', LayerKnot3X, ycoord, LayerKnot12Z)
createmidknot(pos, 'Layerknotb1300', LayerKnot2X, ycoord, LayerKnot13Z)
createknotA(pos, 'Layerknota1300', LayerKnot2X, ycoord, LayerKnot13Z)
createmidknot(pos, 'Layerknotb1400', LayerKnot1X, ycoord, LayerKnot14Z)
createknotA(pos, 'Layerknota1400', LayerKnot1X, ycoord, LayerKnot14Z)
#Base knot can not be a part of the for loop
createmoorinknot('Base1', 0, 0, LayerKnot15Z)
#Create base knot
createtopringa(sections, 'Topringa1',
'Float' + str(sections - 32), 'Midtopknot' + str(sections - 32))
createtopringb(sections, 'Topringb1',
'Midtopknot' + str(sections - 32), 'Float' + str(sections - 31))
createtopringa(sections, 'Topringa2',
'Float' + str(sections - 31), 'Midtopknot' + str(sections - 31))
createtopringb(sections, 'Topringb2',
'Midtopknot' + str(sections - 31), 'Float' + str(sections - 30))
createtopringa(sections, 'Topringa3',
'Float' + str(sections - 30), 'Midtopknot' + str(sections - 30))
createtopringb(sections, 'Topringb3',
'Midtopknot' + str(sections - 30), 'Float' + str(sections - 29))
createtopringa(sections, 'Topringa4',
'Float' + str(sections - 29), 'Midtopknot' + str(sections - 29))
createtopringb(sections, 'Topringb4',
'Midtopknot' + str(sections - 29), 'Float' + str(sections - 28))
createtopringa(sections, 'Topringa5',
'Float' + str(sections - 28), 'Midtopknot' + str(sections - 28))
createtopringb(sections, 'Topringb5',
'Midtopknot' + str(sections - 28), 'Float' + str(sections - 27))
createtopringa(sections, 'Topringa6',
'Float' + str(sections - 27), 'Midtopknot' + str(sections - 27))
createtopringb(sections, 'Topringb6',
'Midtopknot' + str(sections - 27), 'Float' + str(sections - 26))
createtopringa(sections, 'Topringa7',
'Float' + str(sections - 26), 'Midtopknot' + str(sections - 26))
createtopringb(sections, 'Topringb7',
'Midtopknot' + str(sections - 26), 'Float' + str(sections - 25))
createtopringa(sections, 'Topringa8',
'Float' + str(sections - 25), 'Midtopknot' + str(sections - 25))
createtopringb(sections, 'Topringb8',
'Midtopknot' + str(sections - 25), 'Float' + str(sections - 24))
createtopringa(sections, 'Topringa9',
'Float' + str(sections - 24), 'Midtopknot' + str(sections - 24))
createtopringb(sections, 'Topringb9',
'Midtopknot' + str(sections - 24), 'Float' + str(sections - 23))
createtopringa(sections, 'Topringa10',
'Float' + str(sections - 23), 'Midtopknot' + str(sections - 23))
createtopringb(sections, 'Topringb10',
'Midtopknot' + str(sections - 23), 'Float' + str(sections - 22))
createtopringa(sections, 'Topringa11',

```

```

'Float' + str(sections - 22), 'Midtopknot' + str(sections - 22))
createtopringb(sections, 'Topringb11',
'Midtopknot' + str(sections - 22), 'Float' + str(sections - 21))
createtopringa(sections, 'Topringa12',
'Float' + str(sections - 21), 'Midtopknot' + str(sections - 21))
createtopringb(sections, 'Topringb12',
'Midtopknot' + str(sections - 21), 'Float' + str(sections - 20))
createtopringa(sections, 'Topringa13',
'Float' + str(sections - 20), 'Midtopknot' + str(sections - 20))
createtopringb(sections, 'Topringb13',
'Midtopknot' + str(sections - 20), 'Float' + str(sections - 19))
createtopringa(sections, 'Topringa14',
'Float' + str(sections - 19), 'Midtopknot' + str(sections - 19))
createtopringb(sections, 'Topringb14',
'Midtopknot' + str(sections - 19), 'Float' + str(sections - 18))
createtopringa(sections, 'Topringa15',
'Float' + str(sections - 18), 'Midtopknot' + str(sections - 18))
createtopringb(sections, 'Topringb15',
'Midtopknot' + str(sections - 18), 'Float' + str(sections - 17))
createtopringa(sections, 'Topringa16',
'Float' + str(sections - 17), 'Midtopknot' + str(sections - 17))
createtopringb(sections, 'Topringb16',
'Midtopknot' + str(sections - 17), 'Float' + str(sections - 16))
createtopringa(sections, 'Topringa17',
'Float' + str(sections - 16), 'Midtopknot' + str(sections - 16))
createtopringb(sections, 'Topringb17',
'Midtopknot' + str(sections - 16), 'Float' + str(sections - 15))
createtopringa(sections, 'Topringa18',
'Float' + str(sections - 15), 'Midtopknot' + str(sections - 15))
createtopringb(sections, 'Topringb18',
'Midtopknot' + str(sections - 15), 'Float' + str(sections - 14))
createtopringa(sections, 'Topringa19',
'Float' + str(sections - 14), 'Midtopknot' + str(sections - 14))
createtopringb(sections, 'Topringb19',
'Midtopknot' + str(sections - 14), 'Float' + str(sections - 13))
createtopringa(sections, 'Topringa20',
'Float' + str(sections - 13), 'Midtopknot' + str(sections - 13))
createtopringb(sections, 'Topringb20',
'Midtopknot' + str(sections - 13), 'Float' + str(sections - 12))
createtopringa(sections, 'Topringa21',
'Float' + str(sections - 12), 'Midtopknot' + str(sections - 12))
createtopringb(sections, 'Topringb21',
'Midtopknot' + str(sections - 12), 'Float' + str(sections - 11))
createtopringa(sections, 'Topringa22',
'Float' + str(sections - 11), 'Midtopknot' + str(sections - 11))
createtopringb(sections, 'Topringb22',
'Midtopknot' + str(sections - 11), 'Float' + str(sections - 10))
createtopringa(sections, 'Topringa23',
'Float' + str(sections - 10), 'Midtopknot' + str(sections - 10))
createtopringb(sections, 'Topringb23',
'Midtopknot' + str(sections - 10), 'Float' + str(sections - 9))
createtopringa(sections, 'Topringa24',
'Float' + str(sections - 9), 'Midtopknot' + str(sections - 9))

```

```

createtopringb(sections, 'Topringb24',
'Midtopknot' + str(sections - 9), 'Float' + str(sections - 8))
createtopringa(sections, 'Topringa25',
'Float' + str(sections - 8), 'Midtopknot' + str(sections - 8))
createtopringb(sections, 'Topringb25',
'Midtopknot' + str(sections - 8), 'Float' + str(sections - 7))
createtopringa(sections, 'Topringa26',
'Float' + str(sections - 7), 'Midtopknot' + str(sections - 7))
createtopringb(sections, 'Topringb26',
'Midtopknot' + str(sections - 7), 'Float' + str(sections - 6))
createtopringa(sections, 'Topringa27',
'Float' + str(sections - 6), 'Midtopknot' + str(sections - 6))
createtopringb(sections, 'Topringb27',
'Midtopknot' + str(sections - 6), 'Float' + str(sections - 5))
createtopringa(sections, 'Topringa28',
'Float' + str(sections - 5), 'Midtopknot' + str(sections - 5))
createtopringb(sections, 'Topringb28',
'Midtopknot' + str(sections - 5), 'Float' + str(sections - 4))
createtopringa(sections, 'Topringa29',
'Float' + str(sections - 4), 'Midtopknot' + str(sections - 4))
createtopringb(sections, 'Topringb29',
'Midtopknot' + str(sections - 4), 'Float' + str(sections - 3))
createtopringa(sections, 'Topringa30',
'Float' + str(sections - 3), 'Midtopknot' + str(sections - 3))
createtopringb(sections, 'Topringb30',
'Midtopknot' + str(sections - 3), 'Float' + str(sections - 2))
createtopringa(sections, 'Topringa31',
'Float' + str(sections - 2), 'Midtopknot' + str(sections - 2))
createtopringb(sections, 'Topringb31',
'Midtopknot' + str(sections - 2), 'Float' + str(sections - 1))
createtopringa(sections, 'Topringa32',
'Float' + str(sections - 1), 'Midtopknot' + str(sections - 1))
createtopringb(sections, 'Topringb32',
'Midtopknot' + str(sections - 1), 'Float0')

#New for loop
for pos in range(sections):
    #then create the lines that join them together
    #First floor
    createnetA(pos, 'Midline1', 'Float'+str(pos), 'Botknot1'+str(pos), 0)
    createnetA(pos, 'Centremidline1', 'Midtopknot'+str(pos), 'Midbotknot1'+str(pos), 0)
    #Second floor
    createnetA(pos, 'Midline2', 'Botknot1' + str(pos), 'Botknot2' + str(pos), 0)
    createnetA(pos, 'Centremidline2', 'Midbotknot1'+str(pos), 'Midbotknot2'+str(pos), 0)
    #Third floor
    createnetA(pos, 'Midline3', 'Botknot2' + str(pos), 'Botknot3' + str(pos), 0)
    createnetA(pos, 'Centremidline3', 'Midbotknot2' + str(pos), 'Midbotknot3' + str(pos), 0)
    #Fourth floor
    createnetA(pos, 'Midline4', 'Botknot3' + str(pos), 'Botknot4' + str(pos), 0)
    createnetA(pos, 'Centremidline4', 'Midbotknot3' + str(pos), 'Midbotknot4' + str(pos), 0)
    #Fifth floor
    createnetA(pos, 'Midline5', 'Botknot4' + str(pos), 'Botknot5' + str(pos), 0)
    createnetA(pos, 'Centremidline5', 'Midbotknot4' + str(pos), 'Midbotknot5' + str(pos), 0)

```

```

#Sixth floor
createnetA(pos, 'Midline6', 'Botknot5' + str(pos), 'Botknot6' + str(pos), 0)
createnetA(pos, 'Centremidline6', 'Midbotknot5' + str(pos), 'Midbotknot6' + str(pos), 0)
#Seventh floor
createnetA(pos, 'Midline7', 'Botknot6' + str(pos), 'Botknot7' + str(pos), 0)
createnetA(pos, 'Centremidline7', 'Midbotknot6' + str(pos), 'Midbotknot7' + str(pos), 0)
#Eight floor
createnetA(pos, 'Midline8', 'Botknot7' + str(pos), 'Botknot8' + str(pos), 0)
createnetA(pos, 'Centremidline8', 'Midbotknot7' + str(pos), 'Midbotknot8' + str(pos), 0)
# Creating the base ring
createnetB(pos, 'Bottomline1A', 'Botknot8' + str(pos), 'Layerknota1' + str(pos), 0)
createnetB(pos, 'Bottomline1B', 'Midbotknot8' + str(pos), 'Layerknotb1' + str(pos), 0)
createnetB(pos, 'Bottomline2A', 'Layerknota1' + str(pos), 'Layerknota2' + str(pos), 0)
createnetB(pos, 'Bottomline2B', 'Layerknotb1' + str(pos), 'Layerknotb2' + str(pos), 0)
createnetB(pos, 'Bottomline3A', 'Layerknota2' + str(pos), 'Layerknota3' + str(pos), 0)
createnetB(pos, 'Bottomline3B', 'Layerknotb2' + str(pos), 'Layerknotb3' + str(pos), 0)
createnetB(pos, 'Bottomline4A', 'Layerknota3' + str(pos), 'Layerknota4' + str(pos), 0)
createnetB(pos, 'Bottomline4B', 'Layerknotb3' + str(pos), 'Layerknotb4' + str(pos), 0)
createnetB(pos, 'Bottomline5A', 'Layerknota4' + str(pos), 'Layerknota5' + str(pos), 0)
createnetB(pos, 'Bottomline5B', 'Layerknotb4' + str(pos), 'Layerknotb5' + str(pos), 0)
createnetB(pos, 'Bottomline6A', 'Layerknota5' + str(pos), 'Layerknota6' + str(pos), 0)
createnetB(pos, 'Bottomline6B', 'Layerknotb5' + str(pos), 'Layerknotb6' + str(pos), 0)
createnetB(pos, 'Bottomline7A', 'Layerknota6' + str(pos), 'Layerknota7' + str(pos), 0)
createnetB(pos, 'Bottomline7B', 'Layerknotb6' + str(pos), 'Layerknotb7' + str(pos), 0)
createnetB(pos, 'Bottomline8A', 'Layerknota7' + str(pos), 'Layerknota8' + str(pos), 0)
createnetB(pos, 'Bottomline8B', 'Layerknotb7' + str(pos), 'Layerknotb8' + str(pos), 0)
createnetB(pos, 'Bottomline9A', 'Layerknota8' + str(pos), 'Layerknota9' + str(pos), 0)
createnetB(pos, 'Bottomline9B', 'Layerknotb8' + str(pos), 'Layerknotb9' + str(pos), 0)
createnetB(pos, 'Bottomline11A', 'Layerknota9' + str(pos), 'Layerknota1000' + str(pos), 0)
createnetB(pos, 'Bottomline11B', 'Layerknotb9' + str(pos), 'Layerknotb1000' + str(pos), 0)
createnetB(pos, 'Bottomline12A', 'Layerknota1000' + str(pos), 'Layerknota1100' + str(pos), 0)
createnetB(pos, 'Bottomline12B', 'Layerknotb1000' + str(pos), 'Layerknotb1100' + str(pos), 0)
createnetB(pos, 'Bottomline13A', 'Layerknota1100' + str(pos), 'Layerknota1200' + str(pos), 0)
createnetB(pos, 'Bottomline13B', 'Layerknotb1100' + str(pos), 'Layerknotb1200' + str(pos), 0)
createnetB(pos, 'Bottomline14A', 'Layerknota1200' + str(pos), 'Layerknota1300' + str(pos), 0)
createnetB(pos, 'Bottomline14B', 'Layerknotb1200' + str(pos), 'Layerknotb1300' + str(pos), 0)
createnetB(pos, 'Bottomline15A', 'Layerknota1300' + str(pos), 'Layerknota1400' + str(pos), 0)
createnetB(pos, 'Bottomline15B', 'Layerknotb1300' + str(pos), 'Layerknotb1400' + str(pos), 0)
createnetB(pos, 'Bottomline16A', 'Layerknota1400' + str(pos), 'Base1', 0)
createnetB(pos, 'Bottomline16B', 'Layerknotb1400' + str(pos), 'Base1', 0)

if pos > 0:
    #First floor
    createbotring(pos, 'Botringa1', 'Botknot1'+str(pos-1), 'Midbotknot1'+str(pos-1))
    createbotring(pos, 'Botringb1', 'Midbotknot1'+str(pos-1), 'Botknot1'+str(pos))
    #Second floor
    createbotring(pos, 'Botringa2', 'Botknot2' + str(pos - 1), 'Midbotknot2' + str(pos - 1))
    createbotring(pos, 'Botringb2', 'Midbotknot2' + str(pos - 1), 'Botknot2' + str(pos))
    #Third floor
    createbotring(pos, 'Botringa3', 'Botknot3' + str(pos - 1), 'Midbotknot3' + str(pos - 1))
    createbotring(pos, 'Botringb3', 'Midbotknot3' + str(pos - 1), 'Botknot3' + str(pos))
    #Fourth floor
    createbotring(pos, 'Botringa4', 'Botknot4' + str(pos - 1), 'Midbotknot4' + str(pos - 1))

```



```

createbotring(pos, 'Botringb4', 'Midbotknot4' + str(pos - 1), 'Botknot4' + str(pos))
#Fifth floor
createbotring(pos, 'Botringa5', 'Botknot5' + str(pos - 1), 'Midbotknot5' + str(pos - 1))
createbotring(pos, 'Botringb5', 'Midbotknot5' + str(pos - 1), 'Botknot5' + str(pos))
#Sixth floor
createbotring(pos, 'Botringa6', 'Botknot6' + str(pos - 1), 'Midbotknot6' + str(pos - 1))
createbotring(pos, 'Botringb6', 'Midbotknot6' + str(pos - 1), 'Botknot6' + str(pos))
#Seventh floor
createbotring(pos, 'Botringa7', 'Botknot7' + str(pos - 1), 'Midbotknot7' + str(pos - 1))
createbotring(pos, 'Botringb7', 'Midbotknot7' + str(pos - 1), 'Botknot7' + str(pos))
#Base ring or bottom floor
createbasing1(pos, 'Basinga1',
'Layerknota1' + str(pos - 1), 'Layerknotb1' + str(pos - 1))
createbasing1(pos, 'Basingb1',
'Layerknota1' + str(pos), 'Layerknotb1' + str(pos - 1))
createbasing2(pos, 'Basinga2',
'Layerknota2' + str(pos - 1), 'Layerknotb2' + str(pos - 1))
createbasing2(pos, 'Basingb2',
'Layerknota2' + str(pos), 'Layerknotb2' + str(pos - 1))
createbasing3(pos, 'Basinga3',
'Layerknota3' + str(pos - 1), 'Layerknotb3' + str(pos - 1))
createbasing3(pos, 'Basingb3',
'Layerknota3' + str(pos), 'Layerknotb3' + str(pos - 1))
createbasing4(pos, 'Basinga4',
'Layerknota4' + str(pos - 1), 'Layerknotb4' + str(pos - 1))
createbasing4(pos, 'Basingb4',
'Layerknota4' + str(pos), 'Layerknotb4' + str(pos - 1))
createbasing5(pos, 'Basinga5',
'Layerknota5' + str(pos - 1), 'Layerknotb5' + str(pos - 1))
createbasing5(pos, 'Basingb5',
'Layerknota5' + str(pos), 'Layerknotb5' + str(pos - 1))
createbasing6(pos, 'Basinga6',
'Layerknota6' + str(pos - 1), 'Layerknotb6' + str(pos - 1))
createbasing6(pos, 'Basingb6',
'Layerknota6' + str(pos), 'Layerknotb6' + str(pos - 1))
createbasing7(pos, 'Basinga7',
'Layerknota7' + str(pos - 1), 'Layerknotb7' + str(pos - 1))
createbasing7(pos, 'Basingb7',
'Layerknota7' + str(pos), 'Layerknotb7' + str(pos - 1))
createbasing8(pos, 'Basinga8',
'Layerknota8' + str(pos - 1), 'Layerknotb8' + str(pos - 1))
createbasing8(pos, 'Basingb8',
'Layerknota8' + str(pos), 'Layerknotb8' + str(pos - 1))
createbasing9(pos, 'Basinga9',
'Layerknota9' + str(pos - 1), 'Layerknotb9' + str(pos - 1))
createbasing9(pos, 'Basingb9',
'Layerknota9' + str(pos), 'Layerknotb9' + str(pos - 1))
createbasing10(pos, 'Basinga1000', 'Layerknota1000' + str(pos - 1),
'Layerknotb1000' + str(pos - 1))
createbasing10(pos, 'Basingb1000', 'Layerknota1000' + str(pos),
'Layerknotb1000' + str(pos - 1))
createbasing11(pos, 'Basinga1100', 'Layerknota1100' + str(pos - 1),
'Layerknotb1100' + str(pos - 1))

```

```

        createbasing11(pos, 'Basingb1100', 'Layerknota1100' + str(pos),
'Layerknotb1100' + str(pos - 1))
        createbasing12(pos, 'Basinga1200', 'Layerknota1200' + str(pos - 1),
'Layerknotb1200' + str(pos - 1))
        createbasing12(pos, 'Basingb1200', 'Layerknota1200' + str(pos),
'Layerknotb1200' + str(pos - 1))
        createbasing13(pos, 'Basinga1300', 'Layerknota1300' + str(pos - 1),
'Layerknotb1300' + str(pos - 1))
        createbasing13(pos, 'Basingb1300', 'Layerknota1300' + str(pos),
'Layerknotb1300' + str(pos - 1))
        createbasing14(pos, 'Basinga1400', 'Layerknota1400' + str(pos - 1),
'Layerknotb1400' + str(pos - 1))
        createbasing14(pos, 'Basingb1400', 'Layerknota1400' + str(pos),
'Layerknotb1400' + str(pos - 1))

#First floor
createbotring(sections, 'Botringa1', 'Botknot1' + str(sections - 1),
'Midbotknot1' + str(sections - 1))
createbotring(sections, 'Botringb1', 'Midbotknot1' + str(sections - 1),
'Botknot10')
#Second floor
createbotring(sections, 'Botringa2', 'Botknot2' + str(sections - 1),
'Midbotknot2' + str(sections - 1))
createbotring(sections, 'Botringb2', 'Midbotknot2' + str(sections - 1),
'Botknot20')
#Third floor
createbotring(sections, 'Botringa3', 'Botknot3' + str(sections - 1),
'Midbotknot3' + str(sections - 1))
createbotring(sections, 'Botringb3', 'Midbotknot3' + str(sections - 1),
'Botknot30')
#Fourth floor
createbotring(sections, 'Botringa4', 'Botknot4' + str(sections - 1),
'Midbotknot4' + str(sections - 1))
createbotring(sections, 'Botringb4', 'Midbotknot4' + str(sections - 1),
'Botknot40')
#Fifth floor
createbotring(sections, 'Botringa5', 'Botknot5' + str(sections - 1),
'Midbotknot5' + str(sections - 1))
createbotring(sections, 'Botringb5', 'Midbotknot5' + str(sections - 1),
'Botknot50')
#Sixth floor
createbotring(sections, 'Botringa6', 'Botknot6' + str(sections - 1),
'Midbotknot6' + str(sections - 1))
createbotring(sections, 'Botringb6', 'Midbotknot6' + str(sections - 1),
'Botknot60')
#Seventh floor
createbotring(sections, 'Botringa7', 'Botknot7' + str(sections - 1),
'Midbotknot7' + str(sections - 1))
createbotring(sections, 'Botringb7', 'Midbotknot7' + str(sections - 1),
'Botknot70')
#Base ring
createbasing1(sections, 'Basinga1', 'Layerknota1' + str(sections - 1),
'Layerknotb1' + str(sections - 1))

```

```
createbasing1(sections, 'Basingb1', 'Layerknotb1' + str(sections - 1),
'Layerknota10')
createbasing2(sections, 'Basinga2', 'Layerknota2' + str(sections - 1),
'Layerknotb2' + str(sections - 1))
createbasing2(sections, 'Basingb2', 'Layerknotb2' + str(sections - 1),
'Layerknota20')
createbasing3(sections, 'Basinga3', 'Layerknota3' + str(sections - 1),
'Layerknotb3' + str(sections - 1))
createbasing3(sections, 'Basingb3', 'Layerknotb3' + str(sections - 1),
'Layerknota30')
createbasing4(sections, 'Basinga4', 'Layerknota4' + str(sections - 1),
'Layerknotb4' + str(sections - 1))
createbasing4(sections, 'Basingb4', 'Layerknotb4' + str(sections - 1),
'Layerknota40')
createbasing5(sections, 'Basinga5', 'Layerknota5' + str(sections - 1),
'Layerknotb5' + str(sections - 1))
createbasing5(sections, 'Basingb5', 'Layerknotb5' + str(sections - 1),
'Layerknota50')
createbasing6(sections, 'Basinga6', 'Layerknota6' + str(sections - 1),
'Layerknotb6' + str(sections - 1))
createbasing6(sections, 'Basingb6', 'Layerknotb6' + str(sections - 1),
'Layerknota60')
createbasing7(sections, 'Basinga7', 'Layerknota7' + str(sections - 1),
'Layerknotb7' + str(sections - 1))
createbasing7(sections, 'Basingb7', 'Layerknotb7' + str(sections - 1),
'Layerknota70')
createbasing8(sections, 'Basinga8', 'Layerknota8' + str(sections - 1),
'Layerknotb8' + str(sections - 1))
createbasing8(sections, 'Basingb8', 'Layerknotb8' + str(sections - 1),
'Layerknota80')
createbasing9(sections, 'Basinga9', 'Layerknota9' + str(sections - 1),
'Layerknotb9' + str(sections - 1))
createbasing9(sections, 'Basingb9', 'Layerknotb9' + str(sections - 1),
'Layerknota90')
createbasing10(sections, 'Basinga1000', 'Layerknota1000' + str(sections - 1),
'Layerknotb1000' + str(sections - 1))
createbasing10(sections, 'Basingb1000', 'Layerknotb1000' + str(sections - 1),
'Layerknota10000')
createbasing11(sections, 'Basinga1100', 'Layerknota1100' + str(sections - 1),
'Layerknotb1100' + str(sections - 1))
createbasing11(sections, 'Basingb1100', 'Layerknotb1100' + str(sections - 1),
'Layerknota11000')
createbasing12(sections, 'Basinga1200', 'Layerknota1200' + str(sections - 1),
'Layerknotb1200' + str(sections - 1))
createbasing12(sections, 'Basingb1200', 'Layerknotb1200' + str(sections - 1),
'Layerknota12000')
createbasing13(sections, 'Basinga1300', 'Layerknota1300' + str(sections - 1),
'Layerknotb1300' + str(sections - 1))
createbasing13(sections, 'Basingb1300', 'Layerknotb1300' + str(sections - 1),
'Layerknota13000')
createbasing14(sections, 'Basinga1400', 'Layerknota1400' + str(sections - 1),
'Layerknotb1400' + str(sections - 1))
createbasing14(sections, 'Basingb1400', 'Layerknotb1400' + str(sections - 1),
```

```

'Layerknota14000')

# Bottom Weighted Collar
createbottomcollar(sections, 'Bottomcollara1', 'Botknot8' + str(sections - 32),
'Midbotknot8' + str(sections - 32))
createbottomcollar(sections, 'Bottomcollarb1', 'Midbotknot8' + str(sections - 32),
'Botknot8' + str(sections - 31))
createbottomcollar(sections, 'Bottomcollara2', 'Botknot8' + str(sections - 31),
'Midbotknot8' + str(sections - 31))
createbottomcollar(sections, 'Bottomcollarb2', 'Midbotknot8' + str(sections - 31),
'Botknot8' + str(sections - 30))
createbottomcollar(sections, 'Bottomcollara3', 'Botknot8' + str(sections - 30),
'Midbotknot8' + str(sections - 30))
createbottomcollar(sections, 'Bottomcollarb3', 'Midbotknot8' + str(sections - 30),
'Botknot8' + str(sections - 29))
createbottomcollar(sections, 'Bottomcollara4', 'Botknot8' + str(sections - 29),
'Midbotknot8' + str(sections - 29))
createbottomcollar(sections, 'Bottomcollarb4', 'Midbotknot8' + str(sections - 29),
'Botknot8' + str(sections - 28))
createbottomcollar(sections, 'Bottomcollara5', 'Botknot8' + str(sections - 28),
'Midbotknot8' + str(sections - 28))
createbottomcollar(sections, 'Bottomcollarb5', 'Midbotknot8' + str(sections - 28),
'Botknot8' + str(sections - 27))
createbottomcollar(sections, 'Bottomcollara6', 'Botknot8' + str(sections - 27),
'Midbotknot8' + str(sections - 27))
createbottomcollar(sections, 'Bottomcollarb6', 'Midbotknot8' + str(sections - 27),
'Botknot8' + str(sections - 26))
createbottomcollar(sections, 'Bottomcollara7', 'Botknot8' + str(sections - 26),
'Midbotknot8' + str(sections - 26))
createbottomcollar(sections, 'Bottomcollarb7', 'Midbotknot8' + str(sections - 26),
'Botknot8' + str(sections - 25))
createbottomcollar(sections, 'Bottomcollara8', 'Botknot8' + str(sections - 25),
'Midbotknot8' + str(sections - 25))
createbottomcollar(sections, 'Bottomcollarb8', 'Midbotknot8' + str(sections - 25),
'Botknot8' + str(sections - 24))
createbottomcollar(sections, 'Bottomcollara9', 'Botknot8' + str(sections - 24),
'Midbotknot8' + str(sections - 24))
createbottomcollar(sections, 'Bottomcollarb9', 'Midbotknot8' + str(sections - 24),
'Botknot8' + str(sections - 23))
createbottomcollar(sections, 'Bottomcollara10', 'Botknot8' + str(sections - 23),
'Midbotknot8' + str(sections - 23))
createbottomcollar(sections, 'Bottomcollarb10', 'Midbotknot8' + str(sections - 23),
'Botknot8' + str(sections - 22))
createbottomcollar(sections, 'Bottomcollara11', 'Botknot8' + str(sections - 22),
'Midbotknot8' + str(sections - 22))
createbottomcollar(sections, 'Bottomcollarb11', 'Midbotknot8' + str(sections - 22),
'Botknot8' + str(sections - 21))
createbottomcollar(sections, 'Bottomcollara12', 'Botknot8' + str(sections - 21),
'Midbotknot8' + str(sections - 21))
createbottomcollar(sections, 'Bottomcollarb12', 'Midbotknot8' + str(sections - 21),
'Botknot8' + str(sections - 20))
createbottomcollar(sections, 'Bottomcollara13', 'Botknot8' + str(sections - 20),
'Midbotknot8' + str(sections - 20))

```

```

createbottomcollar(sections, 'Bottomcollarb13', 'Midbotknot8' + str(sections - 20),
'Botknot8' + str(sections - 19))
createbottomcollar(sections, 'Bottomcollara14', 'Botknot8' + str(sections - 19),
'Midbotknot8' + str(sections - 19))
createbottomcollar(sections, 'Bottomcollarb14', 'Midbotknot8' + str(sections - 19),
'Botknot8' + str(sections - 18))
createbottomcollar(sections, 'Bottomcollara15', 'Botknot8' + str(sections - 18),
'Midbotknot8' + str(sections - 18))
createbottomcollar(sections, 'Bottomcollarb15', 'Midbotknot8' + str(sections - 18),
'Botknot8' + str(sections - 17))
createbottomcollar(sections, 'Bottomcollara16', 'Botknot8' + str(sections - 17),
'Midbotknot8' + str(sections - 17))
createbottomcollar(sections, 'Bottomcollarb16', 'Midbotknot8' + str(sections - 17),
'Botknot8' + str(sections - 16))
createbottomcollar(sections, 'Bottomcollara17', 'Botknot8' + str(sections - 16),
'Midbotknot8' + str(sections - 16))
createbottomcollar(sections, 'Bottomcollarb17', 'Midbotknot8' + str(sections - 16),
'Botknot8' + str(sections - 15))
createbottomcollar(sections, 'Bottomcollara18', 'Botknot8' + str(sections - 15),
'Midbotknot8' + str(sections - 15))
createbottomcollar(sections, 'Bottomcollarb18', 'Midbotknot8' + str(sections - 15),
'Botknot8' + str(sections - 14))
createbottomcollar(sections, 'Bottomcollara19', 'Botknot8' + str(sections - 14),
'Midbotknot8' + str(sections - 14))
createbottomcollar(sections, 'Bottomcollarb19', 'Midbotknot8' + str(sections - 14),
'Botknot8' + str(sections - 13))
createbottomcollar(sections, 'Bottomcollara20', 'Botknot8' + str(sections - 13),
'Midbotknot8' + str(sections - 13))
createbottomcollar(sections, 'Bottomcollarb20', 'Midbotknot8' + str(sections - 13),
'Botknot8' + str(sections - 12))
createbottomcollar(sections, 'Bottomcollara21', 'Botknot8' + str(sections - 12),
'Midbotknot8' + str(sections - 12))
createbottomcollar(sections, 'Bottomcollarb21', 'Midbotknot8' + str(sections - 12),
'Botknot8' + str(sections - 11))
createbottomcollar(sections, 'Bottomcollara22', 'Botknot8' + str(sections - 11),
'Midbotknot8' + str(sections - 11))
createbottomcollar(sections, 'Bottomcollarb22', 'Midbotknot8' + str(sections - 11),
'Botknot8' + str(sections - 10))
createbottomcollar(sections, 'Bottomcollara23', 'Botknot8' + str(sections - 10),
'Midbotknot8' + str(sections - 10))
createbottomcollar(sections, 'Bottomcollarb23', 'Midbotknot8' + str(sections - 10),
'Botknot8' + str(sections - 9))
createbottomcollar(sections, 'Bottomcollara24', 'Botknot8' + str(sections - 9),
'Midbotknot8' + str(sections - 9))
createbottomcollar(sections, 'Bottomcollarb24', 'Midbotknot8' + str(sections - 9),
'Botknot8' + str(sections - 8))
createbottomcollar(sections, 'Bottomcollara25', 'Botknot8' + str(sections - 8),
'Midbotknot8' + str(sections - 8))
createbottomcollar(sections, 'Bottomcollarb25', 'Midbotknot8' + str(sections - 8),
'Botknot8' + str(sections - 7))
createbottomcollar(sections, 'Bottomcollara26', 'Botknot8' + str(sections - 7),
'Midbotknot8' + str(sections - 7))
createbottomcollar(sections, 'Bottomcollarb26', 'Midbotknot8' + str(sections - 7),

```

```

'Botknot8' + str(sections - 6))
createbottomcollar(sections, 'Bottomcollara27', 'Botknot8' + str(sections - 6),
'Midbotknot8' + str(sections - 6))
createbottomcollar(sections, 'Bottomcollarb27', 'Midbotknot8' + str(sections - 6),
'Botknot8' + str(sections - 5))
createbottomcollar(sections, 'Bottomcollara28', 'Botknot8' + str(sections - 5),
'Midbotknot8' + str(sections - 5))
createbottomcollar(sections, 'Bottomcollarb28', 'Midbotknot8' + str(sections - 5),
'Botknot8' + str(sections - 4))
createbottomcollar(sections, 'Bottomcollara29', 'Botknot8' + str(sections - 4),
'Midbotknot8' + str(sections - 4))
createbottomcollar(sections, 'Bottomcollarb29', 'Midbotknot8' + str(sections - 4),
'Botknot8' + str(sections - 3))
createbottomcollar(sections, 'Bottomcollara30', 'Botknot8' + str(sections - 3),
'Midbotknot8' + str(sections - 3))
createbottomcollar(sections, 'Bottomcollarb30', 'Midbotknot8' + str(sections - 3),
'Botknot8' + str(sections - 2))
createbottomcollar(sections, 'Bottomcollara31', 'Botknot8' + str(sections - 2),
'Midbotknot8' + str(sections - 2))
createbottomcollar(sections, 'Bottomcollarb31', 'Midbotknot8' + str(sections - 2),
'Botknot8' + str(sections - 1))
createbottomcollar(sections, 'Bottomcollara32', 'Botknot8' + str(sections - 1),
'Midbotknot8' + str(sections - 1))
createbottomcollar(sections, 'Bottomcollarb32', 'Midbotknot8' + str(sections - 1),
'Botknot8' + str(sections - 32))
createweightlineB('Baseweightline1','Base1', 0, 0, LayerKnot15Z - 3.0)

# Creating mooring line for cage
createfloatingbuoy(0, 'Mooringfloat1', (radius + 15), (radius + 15), Floor1 + 0.1)
createfloatingbuoy(0, 'Mooringfloat2', (radius + 15), -(radius + 15), Floor1 + 0.1)
createfloatingbuoy(0, 'Mooringfloat3', -(radius + 15), (radius + 15), Floor1 + 0.1)
createfloatingbuoy(0, 'Mooringfloat4', -(radius + 15), -(radius + 15), Floor1 + 0.1)

createmooringsbuoy('Mooringknot1', (radius + 15), (radius + 15), - 6.00)
createmooringsbuoy('Mooringknot2', (radius + 15), -(radius + 15), - 6.00)
createmooringsbuoy('Mooringknot3', -(radius + 15), (radius + 15), - 6.00)
createmooringsbuoy('Mooringknot4', -(radius + 15), -(radius + 15), - 6.00)

createmooringlinecage2(sections, 'Shortcageline1', 'Mooringknot1', 'Float' + str(sections - 28))
createmooringlinecage2(sections, 'Shortcageline2', 'Mooringknot3', 'Float' + str(sections - 20))
createmooringlinecage2(sections, 'Shortcageline3', 'Mooringknot4', 'Float' + str(sections - 12))
createmooringlinecage2(sections, 'Shortcageline4', 'Mooringknot2', 'Float' + str(sections - 4))
createmooringlinecage1(sections, 'Uppercageline1', 'Mooringknot1', 'Float' + str(sections - 30))
createmooringlinecage1(sections, 'Uppercageline2', 'Mooringknot1', 'Float' + str(sections - 26))
createmooringlinecage1(sections, 'Uppercageline3', 'Mooringknot3', 'Float' + str(sections - 22))
createmooringlinecage1(sections, 'Uppercageline4', 'Mooringknot3', 'Float' + str(sections - 18))
createmooringlinecage1(sections, 'Uppercageline5', 'Mooringknot4', 'Float' + str(sections - 14))
createmooringlinecage1(sections, 'Uppercageline6', 'Mooringknot4', 'Float' + str(sections - 10))
createmooringlinecage1(sections, 'Uppercageline7', 'Mooringknot2', 'Float' + str(sections - 6))
createmooringlinecage1(sections, 'Uppercageline8', 'Mooringknot2', 'Float' + str(sections - 2))

createmooringlinea(sections, 'Mooringline1A', 'Mooringfloat1', 'Mooringknot1')
createmooringlinea(sections, 'Mooringline2A', 'Mooringfloat2', 'Mooringknot2')

```

```
createmooringlea(sections, 'Mooringle3A', 'Mooringlefloat3', 'Mooringleknot3')
createmooringlea(sections, 'Mooringle4A', 'Mooringlefloat4', 'Mooringleknot4')

createmooringleb(sections, 'Mooringle1B', 'Mooringleknot1', 'Mooringleknot2', 0.0)
createmooringleb(sections, 'Mooringle2B', 'Mooringleknot1', 'Mooringleknot3', 0.0)
createmooringleb(sections, 'Mooringle3B', 'Mooringleknot2', 'Mooringleknot4', 0.0)
createmooringleb(sections, 'Mooringle4B', 'Mooringleknot3', 'Mooringleknot4', 0.0)

createmooringle(0, 'Anchorline1', 'Mooringleknot1', (radius + 20), AnchorCorners, 0)
createmooringle(0, 'Anchorline2', 'Mooringleknot1', AnchorCorners, (radius + 20), 0)
createmooringle(0, 'Anchorline3', 'Mooringleknot2', (radius + 20), - AnchorCorners, 0)
createmooringle(0, 'Anchorline4', 'Mooringleknot2', AnchorCorners, - (radius + 20), 0)
createmooringle(0, 'Anchorline5', 'Mooringleknot3', - (radius + 20), AnchorCorners, 0)
createmooringle(0, 'Anchorline6', 'Mooringleknot3', - AnchorCorners, (radius + 20), 0)
createmooringle(0, 'Anchorline7', 'Mooringleknot4', - (radius + 20), - AnchorCorners, 0)
createmooringle(0, 'Anchorline8', 'Mooringleknot4', - AnchorCorners, - (radius + 20), 0)

model.SaveData('32 Sections.dat')
```

A.2 Simulation Script

Example for running a steady current simulation at $1.00 \frac{m}{s}$.

```
import OrcFxAPI
import math

model = OrcFxAPI.Model('16 Sections.dat')
environment = model.environment
general = model.general

#Define time interval
general.ImplicitUseVariableTimeStep = 'Yes'
general.ImplicitVariableMaxTimeStep = 0.08
general.StageDuration[0] = 60
general.StageDuration[1] = 60
general.StaticsTolerance = 0.200

#Irregular wave theory
environment.WaveType = 'JONSWAP'
environment.WaveDirection = 180.0
environment.WaveHs = 0.00
environment.WaveTp = 10.00

#Regular wave theory
environment.WaveType = 'Airy'
environment.WaveDirection = 180.0
environment.WaveHeight = 2.00
environment.WavePeriod = 5.00

#Current conditions
#environment.RefCurrentSpeed = 0.0
#environment.RefCurrentSpeed = 0.2
#environment.RefCurrentSpeed = 0.4
#environment.RefCurrentSpeed = 0.6
#environment.RefCurrentSpeed = 0.8
environment.RefCurrentSpeed = 1.0

model.RunSimulation(True, 2, '16sUc100Sim.sim')
```