Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

Capucine Calonnec

Hydrodynamics of a flexible containment boom

Master's thesis in Marine Technology
Supervisor: David Kristiansen
June 2023

NTNU
Norwegian University of
Science and Technology

Capucine Calonnec

Hydrodynamics of a flexible containment boom

Master's thesis in Marine Technology
Supervisor: David Kristiansen
June 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

NTNU
Norwegian University of
Science and Technology

ABSTRACT

This master thesis studies the hydrodynamics of a flexible containment boom. A global model using cable structures theory was built to estimate the deformations of a boom structure facing a current. This global model only takes into account the horizontal plane. It uses the catenary equation, the drag force, and the strip theory to find the shape the structure should take. To validate this model, experiments were carried out on a $5m$ long flexible boom model with a $30cm$ draft. The structure was towed at velocities varying between $0.08m/s$ and $0.53m/s$ corresponding to a Reynolds number varying between 24000 and 159000. The boom was attached at both ends to the carriage and tests were carried out with different attachment widths of $2.5m$, $4m$, and $5m$. In addition, a numerical study was conducted, using the computational fluid dynamics (CFD) solver OpenFOAM. Simulations were performed using the experimental setup as input values and incorporated various representations of the boom's skirt. A flat plate and three different curves facing a current ($0.13m/s$) with various inclination angles were simulated. The numerical results demonstrate that the shape of the submerged membrane significantly influences the applied drag force. Therefore, the vertical dimension of the structure must be considered within the global model. By incorporating varying drag coefficients along the length of the boom in the global model, dependent on the curve and inclination angle, the vertical aspect was taken into account to some extent. A comparison between this model and the experimental study proved that the model was accurate but could need some development and improvement.

SAMMENDRAG

Denne masteroppgaven studerer hydrodynamikken til en fleksibel inneslutningsbom. En global modell basert på teori om kabelstrukturer ble utviklet for å estimere deformasjonene til en bomstruktur som møter strøm. Denne globale modellen tar bare hensyn til det horisontale planet. Den bruker kontaktledningsligningen, luftmotstandskraften og stripeteorien for å finne formen strukturen bør ha. For å validere denne modellen ble det utført eksperimenter på en $5m$ lang fleksibel bom med $30cm$ dypgang. Strukturen ble slept med hastigheter som varierte mellom $0,08m/s$ og $0,53m/s$, noe som tilsvarer et Reynoldstall som varierte mellom 24000 og 159000. Bommen ble festet til vognen i begge ender, og det ble utført tester med ulike festebredder på $2,5m$, $4m$ og $5m$. I tillegg ble det gjennomført en numerisk studie ved hjelp av CFD-beregningsprogrammet OpenFOAM. Simuleringene ble utført med det eksperimentelle oppsettet som inngangsverdier og inkluderte ulike representasjoner av bommens skjørt. En flat plate og tre ulike kurver som vender mot en strøm ($0,13m/s$) med ulike helningsvinkler ble simulert. De numeriske resultatene viser at formen på den nedsenkede membranen har stor innvirkning på den påførte motstandskraften. Derfor må strukturens vertikale dimensjon tas i betraktning i den globale modellen. Ved å inkorporere varierende motstandskoeffisienter langs bommens lengde i den globale modellen, avhengig av kurven og helningsvinkelen, ble det vertikale aspektet til en viss grad tatt hensyn til. En sammenligning mellom denne modellen og den eksperimentelle studien viste at modellen var nøyaktig, men at den kunne trenge en viss utvikling og forbedring.

PREFACE

This report is a presentation of the research work conducted during the spring of 2023 about the hydrodynamics of a flexible containment boom. It follows the project written in the autumn of 2022.

I would like to thank my supervisor David Kristiansen, for the help he gave me throughout this project. I would also want to thank Robert Opland, Terje Rosten, and Emil Bratlier for helping build my experimental model and perform the experiments in the laboratory. In addition, I want to thank Chittiappa Muthanna for the MATLAB script, catman read dt.m, given to the students in the course *TMR4 - Experimental Methods in Marine Hydrodynamics*. Last, I want to thank Marilena Greco for getting me access to the supercomputer Fram, developed by Sigma 2 within the project number NN9592K.

On a more personal note, I would like to thank my roommates at Le Moulin, especially Bastien, who went with me to the laboratory during my experiments, and Mathis, who read this thesis.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS

- L_{line} Length of the floating line
- $L_{membrane}$ Length of the membrane
- $W_{membrane}$ Width of the membrane
- D_{float} Diameter of a float
- L_{float} Length of a float
- d Length of the draft
- λ Scaling factor
- F Towing force
- S_X Standard deviation of X
- P_X Precision limit of X
- e_X Uncertainty
- ρ Water density
- μ Water dynamic viscosity
- ν Water kinematic viscosity
- D Distance between the horizontal axis and the center of the membrane
- θ Inclination angle
- Δ_T Numerical time step
- Re Reynolds number
- C_D Drag coefficient
- F_D Drag force
- U Velocity magnitude
- Ψ Angle between the horizontal tangent of the boom and the x-axis (global model)
- s Longitudinal coordinate of the boom (global model)
- T Tension (global model)

- t Time
- $\hat{t}$ Dimensionless time
- u_i Cartesian component of the velocity vector
- σ_{ij} Stress tensor
- b_i Body force per unit mass
- p Pressure
- δ_{ij} Kroenecker symbol
- y_n State of the system at step n (numerical study)
- Δx Celle size (numerical study)
- f_n Equation system at step n (numerical study)
- Φ_e Value at the face enter
- Φ_E Value at the center of the face backward
- Φ_P Value at the center of the face forward
- λ_E Linear interpolation factor
- C Courant number

ACRONYMS

- **NRC** Canadian National Research Council
- **CFD** Computational Fluid Dynamics
- **CAD** Computer-aided design
- **CAE** Computer-aided engineering
- **CAM** Computer-aided manufacturing
- **CV** Control Volume
- **CFL** Courant-Friedrichs-Lewy
- **FIR** Finite Impulse Response
- **LBM** Lattice Boltzmann Method
- **MC lab** Marine Cybernatics laboratory
- **NCCU** Nordic Coastal Clean Up
- **NTNU** Norwegian university of science and technology
- **PVC** Polyvinyl chloride
- **PISO** Pressure Implicit with Splitting of Operators
- **TOC** The Ocean Clean up

CHAPTER

ONE

INTRODUCTION

This chapter introduces the thesis by giving relevant background to the study, its objectives, and a literature review of the subject. As the background leading to such a study is the same as the one presented in the project made during the autumn semester of 2022, Section 1.1 is inspired by the one written in the project (Calonnec 2022). Moreover, a literature review was made during the project and the one presented here, apart from a few additions, remains similar.

1.1 Background

1.1.1 Plastic pollution

The impacts of plastic waste on marine life are diverse and significant. Plastic debris present on the surface of the ocean can act as carriers of harmful bacteria, posing a threat to coastal areas. Additionally, plastics can inadvertently transport species to new environments where they are not naturally meant to reside. Animals often become entangled in fishing gear and plastic packaging, leading to distressing consequences. Ingestion of plastic by marine organisms can result in stress and toxicity. Coral reefs and the benthic environment are particularly vulnerable to the effects of marine litter (United Nations Environment Programme 2021). However, the consequences of plastic pollution are not limited to marine life alone; humans are also impacted. Plastic contamination is found in food, water, and even the air we breathe. Studies have revealed that approximately 67% of the 391 examined commercial fish species contained microplastics within their bodies (Markic et al. 2020).

Between 1950 and 2017, a total of 9.2 billion tons of plastic have been produced, with half of that amount being generated since 2014 (United Nations Environment Programme 2021). In 2015, it was estimated that 60-99 million tonnes of plastic were wasted during production, transportation, or manufacturing processes. Furthermore, if the current trend of plastic production continues, this number could reach up to 265 million tonnes per year by 2050 (United Nations Environment Programme 2021).

In its initial form, plastic is manufactured as small pieces called plastic pellets, which are approximately the size of a lentil. According to As You Sow organization (As you sow n.d.), around ten trillion plastic pellets find their way into water bodies annually. These pellets enter the water in small quantities at various stages of the production, transportation, and manufacturing processes. However, accidents during transport can result in a significant release of pellets into the ocean. For instance, the X-Press Pearl incident in May 2021 near Sri Lanka led to the loss of 1680 tons of nurdles into the ocean (United Nations 2021).

Given that the 14th sustainable development goal established by the United Nations is to "Conserve and sustainably use the oceans, seas and marine resources" (United Nations n.d.), it is crucial to address the issue of plastic pollution in the ocean, starting with the removal of plastic nurdles. This action will not only help clean the marine environment but also prevent plastic from entering the food chain. Removing plastic from the ocean poses significant challenges due to its widespread distribution. Nonetheless, various approaches have been developed to tackle the problem of plastic pollution.

1.1.2 Solve the plastic pollution problem

The initial approach to address the problem of plastic pollution involves cleaning the marine environment. Nordic Coastal Clean Up (NCCU) (Nordic Coastal Clean Up 2022), for instance, organizes gatherings of approximately 200,000 people across Scandinavia to help protect coastal areas from litter. Each year, they designate a specific day, the first Saturday of May, as the Nordic Coastal Clean Up Day, where volunteers come together to remove rubbish from beaches, rivers, and lakes. In 2021, it was found that 71% of the litter collected on beaches consisted of plastic. Although these beach clean-ups require significant time and manpower, they have limitations in terms of efficiency and the prevention of plastic debris from impacting marine life before it reaches the shore.

Various techniques have been developed by organizations and companies to remove plastic from the ocean. One such company is 4Ocean, established in 2017 by two brothers (4Ocean 2022). With 182 members working across 12 locations, they employ vessels to retrieve plastic debris from oceans, rivers, and coastlines. Since its inception, 4Ocean has successfully collected and documented 11,942 tons of trash.

Another notable organization is The Ocean Clean Up, founded in 2013, which focuses on collecting plastic from the oceans (The Ocean Cleanup 2022). They utilize containment booms, similar to those used for oil spills, to gather floating plastic debris. The technology behind containment booms will be further discussed in Section 1.3. In their efforts to prevent plastic from entering the oceans, they also employ this technology to remove plastic waste from rivers. The organization emphasizes that rivers are the primary source of ocean plastic pollution, with 80% of river pollution originating from 1,000 rivers that could potentially be cleaned.

The final approach to combat plastic pollution involves better management of plastic waste. Out of over 1 billion tonnes of plastic waste generated in the last 70 years, only 10% has been recycled or reused, 14% has been incinerated, and the remaining portion has ended up in landfills, dumps, or the natural environment (Geyer 2020). Consequently, developing plastic waste recycling practices would significantly reduce the amount of plastic entering the environment, including the oceans. However, the primary solution lies in reducing the usage of plastics in our daily lives, particularly single-use plastics. Each year, individuals generate tons of unnecessary plastic debris. As an example, the European Union has implemented regulations since 2021 prohibiting the sale of the ten most commonly found single-use plastic items in marine litter, excluding fishing equipment, such as cotton bud sticks and plastic cups (European commission 2022). Although this idea has gained traction in recent years, changing people's habits is challenging and may take considerable time. Several institutions have created regulations to shift plastic consumption patterns.

1.2 Study

The Ocean Clean Up, along with other entities, is actively developing containment booms as a method to mitigate the impact of plastic pollution on marine life. This project focuses on conducting research to investigate the hydrodynamics of a containment boom specifically designed to collect plastic nurdles.

1.2.1 Objectives

The main objective of the master's thesis is to examine the hydrodynamic forces exerted on a floating boom structure and assess its effectiveness in collecting microplastics. This investigation is carried out in three distinct phases:

Experimental study

The primary purpose of the experiments is to investigate the behavior of a scaled boom structure. The initial objective is to construct a scaled model and subsequently subject it to testing in the laboratory at NTNU. The ultimate goal is to analyze the results of these tests and draw appropriate conclusions based on the findings.

Numerical study

The main objective of the numerical study is to use a 2D computational fluid dynamic (CFD) solver to investigate the behavior of a section of the boom facing the current. More specifically, the aim is to study the influence of the shape of the object representing the draft on the flow patterns and the hydrodynamic forces applied to it. To achieve this objective, multiple designs for the draft are constructed, and a numerical model able to run simulations reflecting the real-life situation, as well as possible, is created.

Global model

The final primary goal of this thesis is to create a global model that represents the boom. Consequently, the objective is to identify a theory applicable to the given situation and utilize it to construct a global model that accurately portrays the behavior of the boom when facing the current. Additionally, this model is intended to be compared with experimental results.

1.2.2 Structure of the thesis

This thesis focuses on investigating a plastic containment boom structure. The first chapter provides an overview of containment boom technology (Section 1.3), followed by a review of relevant literature covering both numerical and experimental studies on boom structures (Section 1.4). Chapter 2 presents the experimental model and setup, while Chapter 3 introduces the numerical study conducted using the CFD software OpenFOAM. Additionally, Chapter 4 explains the theory behind the global model. The findings from these sections are presented in Chapter 5, followed by a comprehensive discussion in Chapter 6, and finally concluding with Chapter 7.

1.3 Containment boom technology

Over 50 years ago, the company SlickBar in Seymour, Connecticut pioneered the development of containment booms. Initially designed for the purpose of oil spill response, these booms were created to effectively collect oil floating on the sea surface. Their primary objectives were to

prevent the spread of oil into specific areas, redirect it to designated locations for treatment, or concentrate it for controlled burning.

However, over time, the applications of containment booms have expanded beyond oil spill response. They have been utilized for various other purposes, such as collecting plastics, seagrass, and seaweed, with the aim of protecting coastal areas and marine life. This broader usage demonstrates the versatility of containment booms in addressing different types of environmental challenges.

1.3.1 Design

Due to its lower density compared to water, oil naturally floats on the sea surface. To address this issue, a containment boom is specifically designed to float on the water and act as a barrier to prevent or redirect the spread of oil into the surrounding environment. It typically consists of four main components: a floating system, a freeboard, a skirt, and tension members (Fingas and Charles 2001).

The choice of floats for the boom can either be solid or inflatable, and this selection determines the overall buoyancy of the boom. The freeboard refers to the portion of the boom that remains above the water level, while the skirt is the part that extends below the water surface. These components are commonly constructed using PVC (polyvinyl chloride) materials. Tension members, usually in the form of steel cables, are incorporated at the top or bottom of the boom to provide structural strength and withstand external forces.

Containment booms are typically assembled by connecting individual sections, which are usually around 15 to 30 meters in length. In some cases, ballast or weights may be added to the boom to ensure that the skirt maintains a vertical position, enhancing its effectiveness in containment.

There are two main technologies employed in boom design: fence boom and curtain boom (Ultratech 2022). Fence booms consist of a flat flotation system with a freeboard above the float, as depicted in Figure 1.3.1. They are relatively easy to deploy and cost-effective, but they are not suitable for rough waters. On the other hand, curtain booms are comprised of a circular flotation device, made of foam or inflated, without any freeboard above, accompanied by a continuous skirt, as shown in Figure 1.3.2. These booms can withstand more challenging conditions as they float along with the waves. However, they can be more cumbersome to store and deploy.

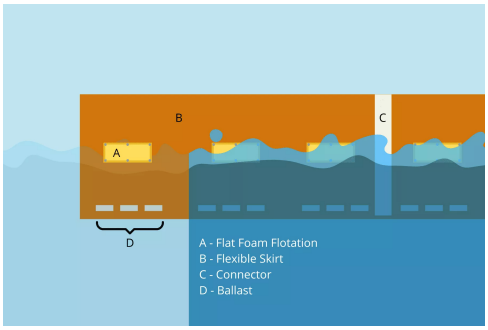


Figure 1.3.1: Fence boom. Source: Ultratech (Ultratech 2022)

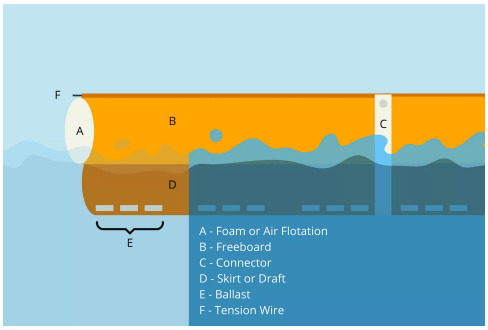


Figure 1.3.2: Curtain Boom. Source: Ultratech (Ultratech 2022)

Another type of boom that is less commonly known is the shore-sealing boom. It is specifically designed to create a barrier between the water and the shoreline. Instead of a flexible skirt, the boom is made of a water chamber acting like a ballast for it to touch the ground. This innovative design allows the boom to be positioned along the exposed shoreline during low tide, effectively preventing oil from reaching the beach.

1.3.2 Boom failure

The effectiveness of a containment boom in collecting oil is influenced by various factors such as water currents, waves, and wind conditions. However, there are situations where the boom may fail to fulfill its intended purpose. Figure 1.3.3, based on research presented in Fingas and Charles 2001, illustrates six major failure modes that can occur.

These failure modes represent different scenarios where the containment boom may not effectively contain or collect the oil. They serve as valuable insights for understanding the limitations and challenges associated with boom deployment and operation. By identifying these failure modes, researchers and engineers can work towards improving boom designs and deployment strategies to mitigate these shortcomings and enhance the overall performance of the containment boom systems.

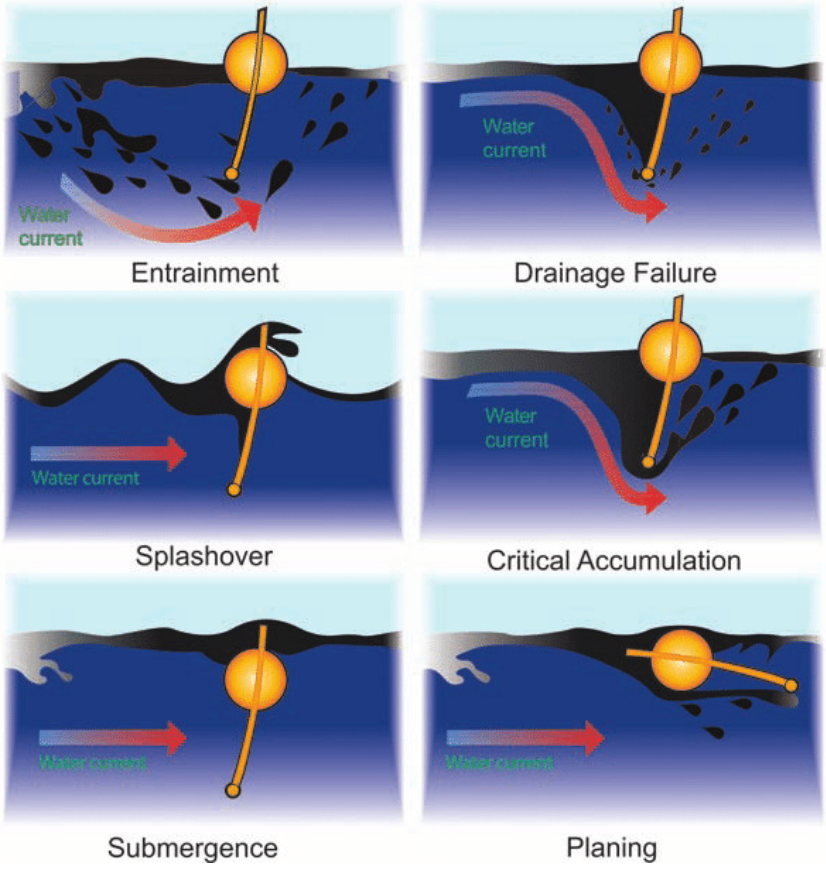


Figure 1.3.3: Boom failure modes. Source: The basics of oil spill cleanup (Fingas and Charles 2001)

Entrainment failure occurs when the water current is too fast and the oil being contained is lighter. In this case, the boom acts as a barrier, causing turbulence and allowing small droplets of oil to be drawn beneath the boom and emerge on the other side.

Drainage failure is also influenced by the speed of the water current, but it is more directly affected by the proximity of the oil to the boom. If the current is too strong, it can sweep the accumulated oil underneath the boom.

Splashover occurs in rough water conditions when the waves are higher than the freeboard of the boom. This leads to waves breaking over the boom and potentially causing oil to escape.

Critical accumulation is a failure mode associated with excessive oil accumulation when the oil being contained is heavier than what is observed in entrainment and drainage failure.

This accumulation can lead to overtopping or breaching of the boom.

Submergence happens when the water rises above the freeboard of the boom, typically when the floating system lacks sufficient buoyancy. Submergence is relatively less common compared to entrainment.

Planing occurs when the boom deviates from its vertical position and instead adopts a horizontal orientation. This is more likely to happen in extreme current and wave conditions. Under normal circumstances, planing is not a significant concern thanks to the bottom weight.

1.4 Litterature review

In this section, we will examine both numerical and experimental investigations conducted on containment booms, focusing on their application for plastic containment. However, since research specifically dedicated to plastic containment booms is limited, this literature review will also encompass studies related to oil containment booms. By incorporating relevant research in both areas, we aim to provide a comprehensive overview of the current knowledge and advancements in boom technology for containment purposes.

1.4.1 Experimental studies

In June 2016, The Ocean Cleanup conducted scale model tests of a portion of their boom at a ratio of 1:5 (Brambini et al. 2017a). Two models were utilized for the testing process. The first model was a rigid boom with a simplified geometry, while the second model was a flexible boom called the RO-BOOM 3200, manufactured by DESMI, which was one of the initial booms employed by The Ocean Cleanup. Both models were capable of simulating surge, heave, and pitch motions. Throughout the tests, various measurements were taken, including the 3 degrees of freedom movements of the boom, the loads on the mooring lines, the dynamic pressure on the screen, and the plastic capture efficiency.

The model was mounted on the carriage of the MARIN Concept Basin, which has a length of 220 meters. Care was taken in the design of the model to minimize three-dimensional effects caused by the model ends. The mooring lines were represented by four springs connected to the booms. Since the boom is intended for use in oceanic conditions, the tests were conducted under harsh conditions. Two different currents were tested, representing the daily and monthly return period values in the North Pacific Ocean, where The Ocean Cleanup operates. Regular wave tests were also performed, maximizing wave heights within the constraints of the basin's size. These tests served to validate the numerical studies and the methodology used in previous studies for analyzing the dynamics of the boom.

Within the framework of "The Reacting Atmosphere" series, an interdisciplinary project focused on climate research, a specific investigation was conducted on Oil Pollution Problems (Muttin 2015). This study involved a full-scale experiment with a floating boom deployed in a coastal environment. The tests were carried out during the winter season, which is known for having a higher incidence of oil spill incidents. Measurements were taken at mid-flooding tide to capture relevant environmental conditions.

The chosen test site offered a combination of factors, including significant maritime traffic and protection from ocean waves and swell. The floating boom used in the experiment had a length of 210 meters and was secured by four mooring lines. Various sensors were strategically placed on the moored buoy, a small vessel, and the boom itself. These sensors collected important data during the tests, such as current velocity at the bottom of the buoy, wind vector at the buoy's head, water column height beneath the buoy, buoy position, mooring tension, vertical skirt angulation, sea current velocity components, and boom geometry.

The primary objective of this experiment was to analyze stress concentration on the boom under different tide conditions in order to assess its effectiveness. By studying the collected data, researchers aimed to gain insights into the boom's performance and its ability to withstand and mitigate oil pollution in coastal environments.

In his doctoral thesis, Mohsen Bardestani conducted a comprehensive study focusing on a two-dimensional floater equipped with a net and sinker tube, analyzing its behavior under the combined influence of waves and current (Bardestani 2017). Although the specific structure investigated was not a containment boom, the findings from his five sets of experiments can be potentially applied to a containment boom model. The various sets of experiments differed in their configurations, and measurements were taken to evaluate the system's performance under different conditions. The specific differences between the sets and the corresponding measurements conducted are outlined below.

Set up	**Measurements**
Floater in waves	Acceleration of the floater in horizontal and vertical directions Forces in the mooring lines Water surface elevations
Net panel with sinker tube in current	Deformation of the net structure Forces in the uppermost net twines at certain connecting points to the bar
Floater with net panel and sinker tube in waves	Deformation of the net panel Forces in the uppermost net twines connected to the floater Floater motion, net deformation, and sinker tube position (recorded on camera)
Floater in combined waves and current	Forces in the mooring lines at both sides of the floater
Floater with net panel and sinker tube in combined waves and current	Forces in the mooring lines at both sides of the floater

The Canadian National Research Council (NRC) undertook a study focused on exploring different designs of Oil Containment Booms (Baker et al. 2019). The objective of this research was to compare three distinct boom designs: a classic boom, a ramped-boom system, and a screen boom system. To achieve this, scaled laboratory tests were conducted, where a geometric scale of 1:8 was employed, and Froude scaling was applied to other relevant parameters. The experiments encompassed both 2D and 3D setups.

In the 2D tests, the boom was affixed to plates, allowing flow to pass through it. On the other hand, in the 3D experiments, the boom was towed through a tank at varying speeds. Furthermore, the tests incorporated wave conditions to simulate real-world scenarios. To represent the oil, a colored liquid was introduced into the water, enabling researchers to visually observe its behavior during the experiments. Additionally, sensors were utilized to measure the current and wave elevation, providing further data for analysis and evaluation.

In her master thesis in 2021, Emma Løyland Vassanyi focused on investigating the impact of current on a plastic containment boom (Emma Løyland Vassanyi 2021). The experiments were conducted using a rigid containment boom model. Two sets of tests were performed to analyze the effects of the current. In the first set of tests, the boom model was subjected

to the application of current. In the second set of tests, a net panel and a sinker tube were incorporated into the boom model. To simulate the current, the models were towed at varying speeds ranging from $0.1m/s$ to $0.5m/s$. Throughout the experiments, different sensors were employed to measure horizontal and vertical forces, pressure on the model, and wave elevation.

To further enhance the visualization of the flow, potassium permanganate was introduced into the water. This allowed for the recording of the flow patterns using a camera. These combined measurements and visualizations provided valuable insights into the behavior of the containment boom under the influence of the current.

1.4.2 Numerical studies

In the second part of their study (Brambini et al. 2017b), The Ocean Cleanup conducted a numerical investigation of their containment boom. The primary objective of this study was to validate a CFD model for predicting the capture efficiency of the boom by comparing it to the experimental model tests.

For the numerical study, the Ocean Cleanup utilized the software XFlow, which is based on the Lattice Boltzmann Method (LBM). Instead of directly solving the Navier-Stokes equations, the LBM simulates fluid density on a lattice using a propagate-collide scheme. In this study, the Discrete Phase Model was combined with the LBM to capture the Lagrangian behavior of particles. To model turbulence, a Large Eddy Simulation model, specifically the Wall-Adapting Local Eddy viscosity was employed. Additionally, an adaptive mesh refinement technique was implemented around the boom to ensure that the mesh followed the boom's motion.

In addition to the XFlow software, another numerical approach was conducted using Orcaflex (Brambini et al. 2017a). Orcaflex is a fully 3D non-linear time domain finite element program. The boom was divided into three line elements in Orcaflex: the floater, the screen, and the ballast. Both rigid and flexible models were tested using Orcaflex and compared against the experimental results. The results obtained from Orcaflex were satisfactory for the rigid model. However, for the flexible model, the predicted forces were found to be overestimated. The author suggested that this discrepancy might be attributed to the inability to accurately model the flexibility of the screen using drag lines in Orcaflex.

The NRC also conducted numerical simulations using CFD (Baker et al. 2019). For their study, they employed the CFD software OpenFOAM, which is a versatile finite-volume solver. The simulations focused on two immiscible fluids, water, and oil, using OpenFOAM's interFoam solver. This solver utilizes a volume-of-fluid method based on phase-fraction to capture the interface between the fluids. It involves solving the Navier-Stokes equations coupled with an interface advection equation. The density at each location is determined based on the densities of water and oil, as well as the volume fraction of water. To account for turbulence effects, a $k - \epsilon$ model was employed in the simulations.

Various model shapes were tested in both 2D and 3D configurations, with speeds ranging from 1.54 to 2.57 m/s. It was expected that some differences would arise between the 2D and 3D models due to variations in oil concentration and flow conditions across the main flow direction, as well as disparities in the approach and mesh characteristics. However, the results obtained from the 2D and 3D simulations exhibited consistency. These simulation outcomes played a crucial role in informing the implementation of the experimental setup for subsequent studies.

In the study conducted by Feng, Zhang, and Wu 2018, CFD is utilized to simulate the oil collection capability of a flexible containment boom. The simulation focuses on tracking the two-phase flow of oil and water using the Volume-of-Fluid (VOF) method, which enables the visualization of the oil slick shape and its variation over time.

To account for turbulent effects in the flow, a $k - \epsilon$ model is employed in the simulations. This model provides an estimation of turbulence parameters such as kinetic energy (k) and the

dissipation rate of kinetic energy (ϵ). By incorporating the $k - \epsilon$ model, the researchers can better capture the complex flow characteristics associated with oil containment and evaluate the effectiveness of the flexible boom in collecting the oil.

Amini et al. 2017 realized a research project studying the behavior of flexible booms under various waves and current conditions, as most existing research has been conducted on rigid barriers. The objective is to assess the efficiency limits of an anti-pollution concept has been designed by Ugo Cavalli and evaluate the performance of the flexible skirt in containing the oil slick.

The methodology includes conducting numerical simulations and experimental studies to investigate the effectiveness of the Cavalli system under open sea conditions. Finite Element (FE) analysis is used for both fluid and structural analysis, with a focus on fluid-structure interaction (FSI) modeling. The fluid is modeled using the Navier-Stokes equations and a multi-phase fluid approach to account for the oil and water mixture. The structural response of the boom is analyzed using solid beam elements, considering large deformations and rotations.

The numerical simulations aim to capture the interactive behavior between the fluid flow and the boom's deformation. A coupling algorithm is used to solve the FSI problem, and a time integration approach is employed for transient analysis. The research also discusses the sensitivity of the boom's deformation to the ballast weight, as well as the influence of different parameters such as oil properties, oil layer thickness, towing velocity, sea state, and boom characteristics.

Preliminary results show that a flexible skirt can significantly affect the pressure field around the boom compared to a rigid barrier. The pressure distribution and hydrodynamic forces are different for rigid and flexible barriers, highlighting the importance of using flexible structures for realistic simulations and obtaining accurate results.

In her master's thesis, Emma Løyland Vassanyi 2021 conducted a numerical investigation of a boom structure using CFD. The open-source CFD solver OpenFOAM, which is based on the Finite Volume method, was employed for the simulations.

Various properties were assigned to the flow within the CFD model, including a Newtonian fluid behavior, isotropic fluid properties, single-phase flow, incompressible fluid assumption, isothermal fluid conditions, and laminar flow characteristics. To solve the incompressible and unsteady flow equations, the PISO algorithm was utilized in the OpenFOAM solver.

The primary objective of this study was to compare the numerical results obtained with OpenFOAM to the experimental setup. Consequently, the mesh and domain sizes used in the numerical simulations were set to match the values employed in the experimental tests. The results of the comparison revealed that the numerical study underestimated the drag coefficient value when compared to the experimental observations.

CHAPTER
TWO

EXPERIMENTAL STUDY

This chapter will present the experiments performed on a flexible containment boom. These experiments aim to have insight into the comportment of the structure facing a current of different speeds. To do so, a model was built and tested in the *Marine cybernetics laboratory* of NTNU between the 2^{nd} January and the $15th$ January. The model was designed and created during the fall of 2022 as part of the project (Calonnec 2022) and will therefore have similarities with this one.

To simulate the current, the model was towed at a speed varying between $0.13m/s$ and $0.53m/s$. The forces at the extremities were measured as well as the current behind the model. Moreover, the floating part of the model was equipped with markers to follow its shape during the simulations. A total of 167 runs were conducted.

2.1 Set up and model

2.1.1 Model

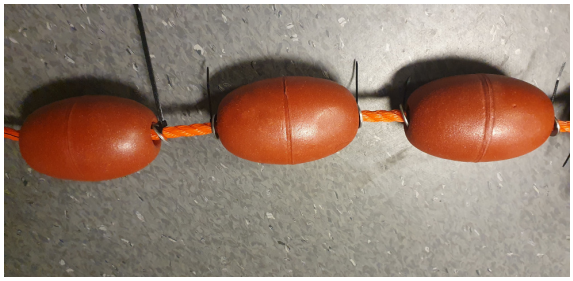
The model created is based on the structure used by The Ocean Cleanup to remove plastic from the oceans. This boom is 800 m long and the skirt is 4m deep as shown in Figure 2.1.1.

Figure 2.1.1: Sketch of the model used by The Ocean Cleanup. Source: The Ocean Cleanup 2022

The model is made of three parts:

- a floating line 2.1.2a: a rope and 35 fishing net floats of 150g buoyancy each. Each of the float has been fixed with a clamp on each side 2.1.2b.
- a flexible membrane made with sailing fabric in PVC 2.1.2c.
- a chain to add some weight at the bottom 2.1.2d.

The membrane has been sewn in such a way that the float line can be slipped into the upper part and the chain into the lower part.



(a) Floating line

(b) Float mounting

(c) Membrane fabric

(d) Chain

Figure 2.1.2: Model composition

The dimensions of the model are presented in Table 2.1.1. Even though the model is based on the Ocean Cleanup boom, it was not possible to geometrically scale the length, the draft, and the diameter of the floater with the same scaling factor. The draft would have been too small and impossible to study. Therefore, the dimensions were chosen independently of each other, while maintaining consistency. The scaling factor of the draft will then be the reference one.

Quantity	**Symbol**	**Value**	**Unit**
Length of the floating line	L_{line}	5.38	[m]
Length of the membrane	$L_{membrane}$	499	[m]
Width of the membrane	$W_{membrane}$	0.43	[m]
Diameter of a float	D_{float}	66.5	[mm]
Length of a float	L_{float}	102.5	[mm]
Length of the draft	d	0.27	[m]

Table 2.1.1: Dimensions of the boom model

The chain used in the model was chosen among the chains available at NTNU, therefore, the weight was not determined before. When fully built 2.1.3, the model weights 2kg.

Figure 2.1.3: Experimental model

2.1.2 Scaling

In the module *TMR04 - Experimental Methods in Marine Hydrodynamics*, different scaling laws were introduced to the students and tools to choose the correct one. In this study, during the experiments, no wave will be generated by the wave maker and only small ones will be generated by the model. Therefore, the resistance will mainly come from viscous effects. As Reynolds number deals with the relation between inertial and frictional force, Reynolds similarity should be applied. Nevertheless, the Froude similarity will be applied to the model to find the towing speed and the mass. This choice was made from many arguments:

- Froude number deals with the relation between inertial forces and gravitational forces. The model will be towed at the water surface, so the surface wave effects will influence the model. These surface effects are accounted for by Froude.
- In the Ocean Cleanup tests, they used the Froude similarity (Brambini et al. 2017a).
- To do Reynolds similarity, as the fluid can't be modified, the towing speed will have to increase as the length will decrease. It won't be possible to run the test with this range of speed.
- The drag force of the model can be considered independent of the Reynolds number so Froude similarity can be applied without correction.

Froude similarity considers the Froude number to be similar for the model scale boom and the full-scale boom. This similarity provides the scaling factors in Table 2.1.2.

Parameter	**Dimension**	**Scaling factor**
Length	L	λ
Mass	M	λ^3
Linear mass	ML^{-1}	λ^2
Velocity	LT^{-1}	$\sqrt{\lambda}$
Force	MLT^{-2}	λ^3

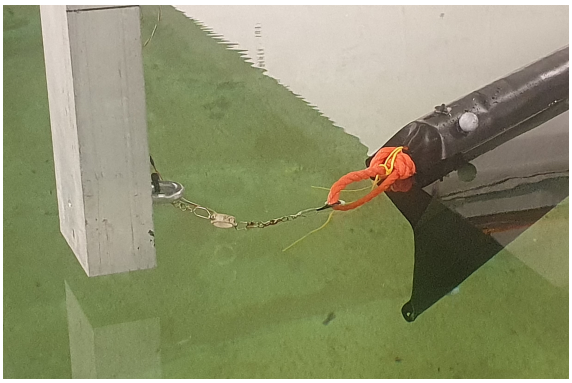
Table 2.1.2: Scaling factors with Froude similarity

As explained before, the scaling factor (λ) of the skirt will be the reference and used to scale the other properties. The full-scale boom is $4m$ deep and the model boom is $0.27m$ deep so the scaling factor is $\lambda = 15$.

2.1.3 Experimental set up

The model was mounted on the towing carriage in the tank. Two metal beams were fixed on the carriage with clamps as shown in Figure 2.1.4a. The model was connected to the beams with two force transducers on each end of the floating line. Figure 2.1.4b shows how the model is attached to the beams. The model could then move freely when being towed and the force applied on each end could be measured.

(a) Beams connected to the carriage



(b) Force transducers connecting the model to the beams

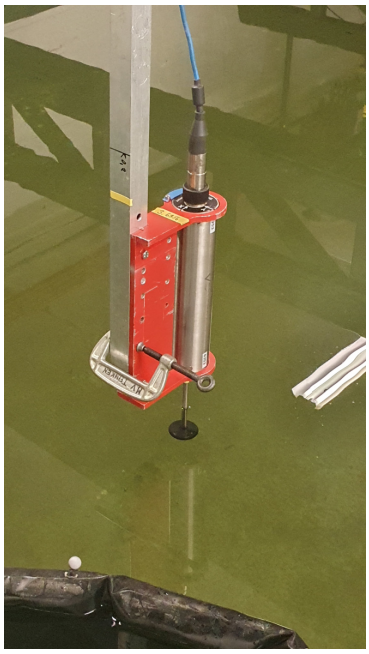
Figure 2.1.4: Connection between model and carriage

A current probe was also mounted on the carriage, see Figure 2.1.5 to measure the fluid velocity in x and y direction.

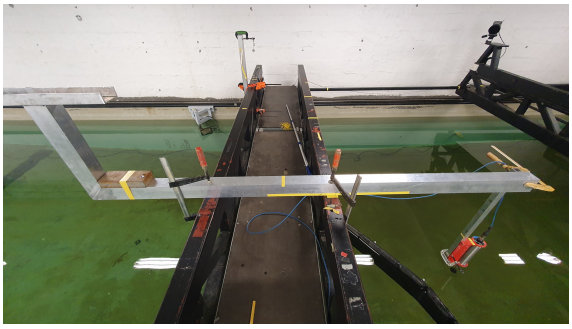
2.1.3.1 Positioning system

To follow the deformation of the boom during a run, the Qualysis positioning system was used. This system is composed of OQUS cameras (Qualysis 2021) and software, the Qualisys Track Manager. Qualisys system is a motion capture system that uses infrared cameras to track the position and movement of markers attached to a floating structure in three-dimensional space. The system can capture the movement of multiple markers simultaneously and with high accuracy, enabling the analysis of the structure's motion and deformation under different conditions. The captured data can be exported in different formats, processed, and visualized using software to provide insights into the behavior of the structure.

In this case, 4 cameras were mounted on the carriage, see Figure 2.1.6a and 6 cameras are installed underwater in the tank. The underwater cameras can be rotated and moved up and down a little bit to adjust their view but they can not be moved freely in the tank. Their position is described in Section 2.1.3.2.



(a) Current probe



(b) Current probe mounting

Figure 2.1.5: Current probe and mounting

The system required calibration before being used. To do so, an L-shaped structure equipped with markers is installed at the surface of the water and a calibration stick with markers on it is moved around the area while the camera is running. This enables the system to be correctly set up where the model will be placed. To do the underwater cameras' calibration, the same process has to be executed underwater.

Markers were mounted on the model as shown, indicated by a red arrow, in Figure 2.1.6b on the floating part and at the bottom of the skirt. 10 markers were installed on the floating part and 11 markers were at the bottom of the skirt.

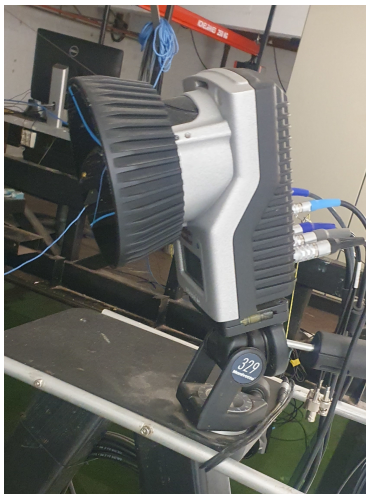
To follow the markers, a first run has to be performed to then associate each marker to its correct name, from M1 to M10 as in Figure 2.1.7. The first run was also necessary to confirm that during the whole simulation, each marker was captured by two cameras, as required by the tracking system.

2.1.3.2 Laboratory

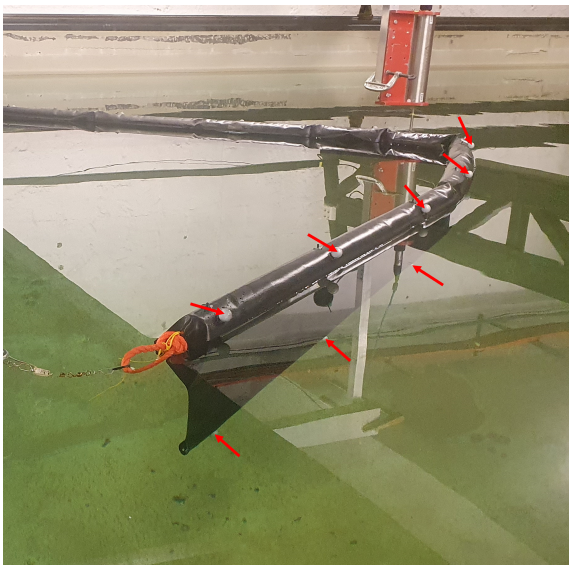
The experiments were performed in the *Marine Cybernetics laboratory* (MC lab) of NTNU 2.1.8. The tank is $40m$ long, $6.45m$ wide, and $1.5m$ deep. It is equipped with a wave maker, a towing carriage, and a positioning system. Figure 2.1.9 represents the tank and the position of the six underwater cameras.

2.1.4 Implementation

The carriage was towed $18m$ along the tank. The width of the membrane and the towing speed were modified to evaluate their influence on the structure. Three different widths were tested: $2.5m$, $4m$, and $5m$, and 10 different towing velocities varying between $0.08m/s$ and $0.53m/s$. The number of tests for each combination is presented in Table 2.1.3.



(a) OQUS camera



(b) Markers mounted on the model

Figure 2.1.6: Positioning system

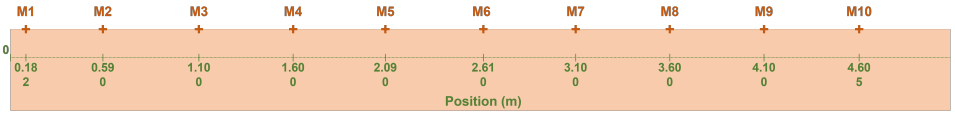


Figure 2.1.7: Position of the markers on the membrane

	Towing velocity [m/s]									
Width [m]	**0.08**	**0.13**	**0.18**	**0.23**	**0.28**	**0.33**	**0.38**	**0.43**	**0.48**	**0.53**
2.5	0	2	3	5	6	3	2	2	2	2
4	5	8	27	8	19	8	8	8	8	8
5	0	4	4	4	4	4	4	3	3	3

Table 2.1.3: Number of tests carried out for each setting

Some tests were performed after adding ropes on each side of the membrane. The rope was attached between the bottom of the skirt and the attached point to the force transducers. The goal of these ropes was to prevent the membrane from flipping and make some visual observations.

2.2 Data analysis

The software Catman was used to acquire the data measured during the experiments: forces, velocity, current and markers' position. This software provides for each simulation a binary file. To read this file, the script *catman_ read.m*, provided in the module *TMR04-Experimental methods in Marine Hydrodynamics* was run using MATLAB (Mathworks 2022). The data acquisition was made with a frequency of $200Hz$.

MATLAB was used to analyze the time series of the experiments. The main code *main.m*, given in the appendix, reads the binary file and applies the different functions presented in this section.

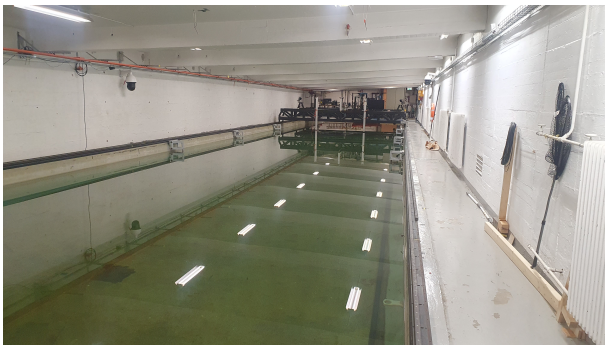


Figure 2.1.8: MC lab

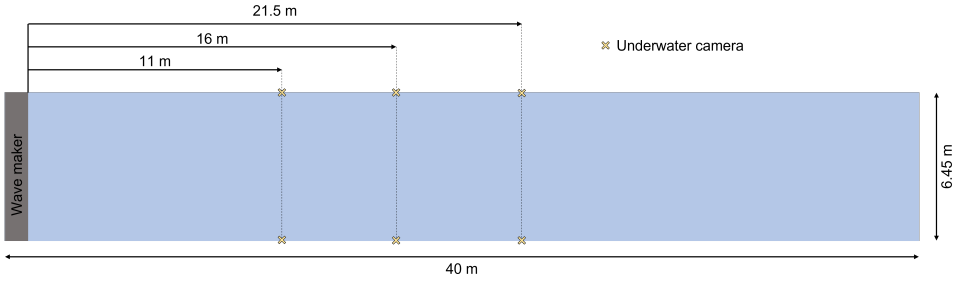


Figure 2.1.9: schematic representation of the pool

2.2.1 Shape

The shape of the model is provided through the coordinates of the ten markers throughout time. The shape can therefore be plotted at any time thanks to the script *shape.m*. However, The markers were not always correctly associated with their name or they could disappear for some seconds during the acquisition. Consequently, it was always necessary to pay attention to each result individually.

The second script created to follow the shape of the structure is *markers.m*. This function plots the position in x and y of each marker as a function of time. It can be used to check if some markers were lost by the cameras during the simulation or if the names were not correctly attributed to the markers.

2.2.2 Forces

The forces from each end were measured thanks to force transducers. A lot of noise was captured in the force acquisition. Thus a numerical filter was applied to the data. To design the filter, the function *designfilt* from MATLAB was used. This function allows to create any type of filter. In this case, the high-frequency noises were the ones making the analysis of the results difficult. Therefore, the filter used was a low pass filter FIR (Finite Impulse Response). This filter allows low-frequency signals to pass through while attenuating or reducing higher-frequency signals. The filter order was 200 and the cut-off frequency was $0.05Hz$. To choose these two characteristics, different combinations were applied before picking up the best one. To apply the filter, the function *filtfilt* available in MATLAB was used.

2.2.3 Velocity

The towing velocity was plotted as a function of time with the script *speed.m*. This towing velocity corresponds to the carriage speed measured during the different simulations.

2.2.4 Current

The current probe provided the velocity of the fluid behind the membrane in x and y directions. The script *current.m* was written to plot the current as a function of time. As the probe captured a lot of noise, a low pass filter was applied, similar to the one used on the forces, with an order of 100 and a cut-off frequency of $0.5Hz$.

2.2.5 Repeatability test

To verify that the experiments carried out are repeatable multiple tests were run with the same conditions in speed and width. These test are called repeatability tests.

2.2.6 Uncertainty

An uncertainty analysis was performed using Excel (Microsoft 2023) to assess the variability of the tests. The analysis involved determining the standard deviation, the precision error, and the uncertainty of a single test. The forces measured at the ends of the membrane during the tests were used to conduct this analysis. The standard deviation of N samples is calculated as

$$S_F = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(F_i - \bar{F}\right)^2} \tag{2.1}$$

In Excel, the function *STDEV.S(number 1, number 2,..., number N)* was used. The standard deviation of the mean value of N samples is then defined by

$$S_{\bar{F}} = \frac{S_F}{\sqrt{N}} \tag{2.2}$$

The precision limit of a test can therefore be calculated as

$$P_F = tS_F \tag{2.3}$$

And the precision limit of the mean can be calculated as

$$P_{\bar{F}} = t\frac{S_F}{\sqrt{N}} \tag{2.4}$$

The weight t needed to determine the precision limit can be found using Student's distribution or using the function *TINV(probability,deg_freedom)* in Excel. The *probability* is $1 - 0.95$ as we want 95% chance that the value is within $\pm t.S_F$ from the true value. The *deg_freedom* required is the degrees of freedom, defined as $N - 1$ with N the number of tests. As the number of tests varies depending on the chosen setup, this value will differ.

The uncertainty of a single test can then be calculated, see equation (2.5) as well as the uncertainty of the mean, equation (2.6).

$$e_F = \frac{P_F}{\bar{F}} \tag{2.5}$$

$$e_{\bar{F}} = \frac{P_{\bar{F}}}{\bar{F}} \tag{2.6}$$

CHAPTER

THREE

NUMERICAL STUDY

This chapter presents the numerical study conducted using the open-source software OpenFOAM, based on the Finite Volume Method. OpenFOAM was used with Windows Subsystem for Linux (WSL) as well as on the supercomputer Fram provided by Sigma 2 (Sigma2 2023). To look at the results, the application Paraview was used.

The goal of this study is to analyze the influence of physical parameters on the forces applied to a 2D section of the boom. These results will then be used to compare with the experimental setup see Chapter 2 and as an input in the global model presented in Chapter 4. Therefore, the dimensions that will be used in OpenFOAM come from the model size and the basins in which it was tested.

This chapter will present the methods used and, in some respects, draw inspiration from the project carried out in the first semester (Calonnec 2022), as the two studies have many points in common.

3.1 Governing equations

3.1.1 Flow properties

The properties of the numerical flow simulation are tailored to closely resemble the actual flow observed in physical laboratory experiments. Thats is why the value of the density was assumed to be $\rho = 1000 kg/m^3$ and the kinematic viscosity assumed to be $\nu = 10^{-6} m^2/s$. However, in order to simplify the flow problem and minimize computational time, certain assumptions regarding the flow characteristics were made:

- **Newtonian fluid:** stresses are proportional to the rate of change of the fluid's velocity vector.
- **Isotropic fluid:** the fluid properties do not depend on the direction.
- **One-phase flow:** the only fluid considered is water.
- **Incompressible flow:** the density ρ is constant.
- **Isothermal flow:** the temperature is constant, therefore the kinematic viscosity ν is constant.
- **Laminar flow:** the fluid travels smoothly or in a regular path, chosen to simplify the simulation.

3.1.2 Equations

The control volume approach is a straightforward method used to analyze and derive conservation laws. It involves the examination of a spatial region known as the Control Volume (CV) through which the fluid flow passes. The key properties of the fluid considered in this approach are mass and momentum, which are expressed through conservation equations. The continuity equation, representing the conservation of mass, can be written as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \tag{3.1}$$

where t denotes time and u_i represents the Cartesian component of the velocity vector. The momentum equation, on the other hand, is expressed as:

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial u_j}\right) = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho b_i \tag{3.2}$$

Here, u_j represents a Cartesian component of the velocity vector, σ_{ij} denotes the stress tensor, and b_i signifies the body force per unit mass. The stress tensor can be defined as:

$$\sigma_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right) \tag{3.3}$$

In this expression, μ corresponds to the dynamic viscosity, p represents the pressure, and δ_{ij} denotes the Kronecker symbol defined as:

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

By assuming incompressibility, the continuity equation simplifies to:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{3.5}$$

Combining equations 3.2 and 3.3 yields the complete momentum equation. Then, by applying the assumption of incompressibility and the continuity equation 3.5 to this equation, the three-dimensional Navier-Stokes equation for laminar flow can be derived:

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j}\right) = -\frac{\partial}{\partial x_j}(p\delta_{ij}) + \mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \rho g_i. \tag{3.6}$$

Here, g_i represents the gravitational acceleration. The Navier-Stokes equation consists of the inertia forces on the left-hand side, pressure forces as the first term on the right-hand side, and viscous forces as the remaining terms on the right-hand side.

3.2 Solver

- **PISO algorithm**

The PISO algorithm, which stands for Pressure Implicit with Splitting of Operators, is employed to solve the previously mentioned Navier-Stokes equation. Originally developed by Issa for non-iterative computation of unsteady, compressible flow, it has been adapted for steady state flow problems. As an implicit solver, it offers enhanced stability in obtaining results (OpenFOAM wiki 2022).

This algorithm addresses the coupling between pressure and velocity in the discretized fluid flow equation. It starts by predicting the pressure and then performs two subsequent steps to correct the pressure and velocity, ensuring mass conservation. While it is possible to utilize

additional correctors in the PISO algorithm to enhance accuracy, this study focuses on the version of PISO that employs two correctors for incompressible and unsteady flow.

- **Schemes**

To discretize the Navier-Stokes equations, two numerical schemes are employed: time discretization and interpolation. The specific schemes utilized are described below.

Time discretization: The Euler implicit scheme is employed, given by the equation:

$$y_{n+1} = y_n + \Delta x f_{n+1} \tag{3.7}$$

Here, y_{n+1} represents the state of the system at the later time, y_n represents the state at the current time, Δx denotes the step size, and f_{n+1} represents the equation system at the later time.

Interpolation: A linear interpolation scheme is employed for the interpolation process. This method involves determining the value at the center of a grid cell edge by considering the centers of the two neighboring cells. The interpolation equation is given as:

$$\Phi_e = \Phi_E \lambda_e + \Phi_P (1 - \lambda_e) \tag{3.8}$$

Here, Φ_e represents the value at the face center, Φ_E represents the value at the center of the cell forward, Φ_P represents the value at the center of the cell backward, and λ_e represents the linear interpolation factor. The interpolation factor λ_e is calculated using the following formula:

$$\lambda_e = \frac{x_e - x_P}{x_E - x_P} \tag{3.9}$$

where x_e represents the position of the face center, x_E represents the position of the cell center forward, and x_P represents the position of the cell center backward.

- **Stability**

In order to obtain accurate results, it is crucial for a simulation to be stable. The stability of the simulation in this study is assessed using the Courant-Friedrichs-Lewy (CFL) condition, which is a commonly employed stability criterion. The CFL condition is given by the equation:

$$C = \frac{\Delta T |u|}{\Delta x} \tag{3.10}$$

Here, ΔT represents the time step, Δx denotes the cell size, and u corresponds to the magnitude of velocity within the cell. To ensure stability, the CFL condition dictates that the value of C must be less than 1, i.e., $C < 1$. By satisfying this condition, the simulation is deemed stable, thereby enabling reliable and accurate results to be obtained.

3.3 Implementation

The implementation in OpenFoam was based on a tutorial case folder found on the OpenFoam website. Modifications of this folder were made to match the desired situation and solver. The function *snappyHexMesh* was used, which requires defining a mesh using *blockMesh* for the domain and an stl geometry for the solid studied. These are presented in this section as well as the folders and files needed to define the numerical solver. The text files are given in the appendix.

The folder used to run OpenFOAM contains three subfolders *0*, *constant*, and *system*. The organization of this folder is presented in Figure 3.3.1.

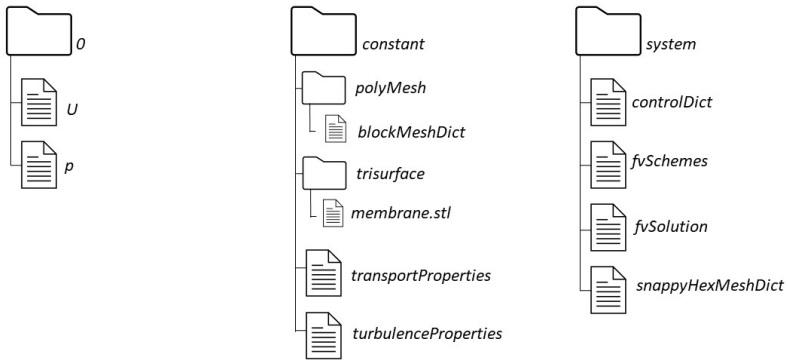


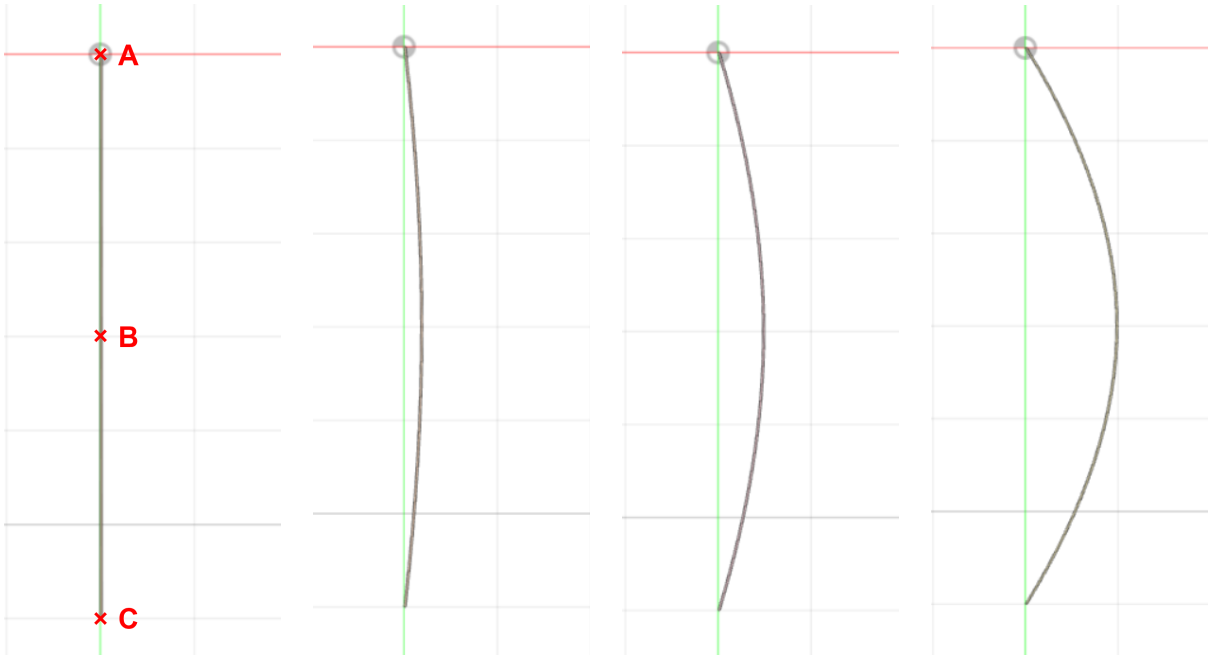
Figure 3.3.1: Organisation of the files and folders to use OpenFOAM. Source: OpenFOAM 2022b

3.3.1 Geometry

The geometry of the membrane was created using Fusion 360. Fusion 360 is a computer-aided design (CAD), computer-aided engineering (CAE), and computer-aided manufacturing (CAM) software developed by Autodesk (Autodesk 2023). It provides a comprehensive suite of tools for designing, simulating, and manufacturing various products and prototypes. Fusion 360 enables users to create 3D models, perform simulations and analyses, and generate toolpaths for machining. In this study, it was used to generate 4 geometries: flat plate, curve 1, curve 2, and curve 3, presented Figure 3.3.2. For the four geometries, the distance between the highest and lowest points, represented by points A and C on the flat plate model, is always the same. What varies is the distance between the horizontal axis and the center of the membrane, represented by point B on the flat plate model. In Table 3.3.1, this distance, called D, is given for the four geometries.

Geometry	Distance D [mm]
flat plate	0
curve 1	10
curve 2	25
curve 3	50

Table 3.3.1: Geometries definition



(a) flat plate **(b)** curve 1 **(c)** curve 2 **(d)** curve 3

Figure 3.3.2: Geometries

Moreover, for some of these geometries, files have been created with an inclination angle θ around the point of attachment. An example of such a geometry is shown in Figure 3.3.3, for curve 2, rotated by an angle of $\theta = 30°$. A summary of the geometries created is shown in table 3.3.2

After creating the geometry the geometry on Fusion360, each of them was exported into an *.stl* format as required by the mesh generator that is used.

Figure 3.3.3: Geometry of curve 2 rotated by 30°

	Geometry			
Rotation angle theta [°]	**flat plate**	**curve 1**	**curve 2**	**curve 3**
0	1	1	1	1
15	1	1	1	1
30	0	1	1	1
45	0	1	1	0
60	0	1	1	0

Table 3.3.2: Summary of the created geometries

3.3.2 Mesh generation

3.3.2.1 Domain

The mesh of the domain was generated to fit as much as possible the experimental setup used in the laboratory in January 2023, see Chapter 2. The membrane is represented as a solid of $d = 0.3m$ long and $t = 0.001m$ thick. As OpenFOAM generates 3D simulation, it is necessary to have a 3D object, therefore the last dimension $0.1m$. However, this will not affect the 2D results.

The domain used in the study has a depth of $1.5m$, which corresponds to the depth of the MC lab where the experiments are conducted. To maintain consistency with a previous study by Emma Løyland Vassanyi 2021, the sizes of the inlet and outlet of the domain were chosen to be the same as in her study. This decision was made because the current study is closely related to hers and it was confirmed that her implementation was accurate. Figure 3.3.4 illustrates a representation of the domain, although the figure does not depict the accurate scales.

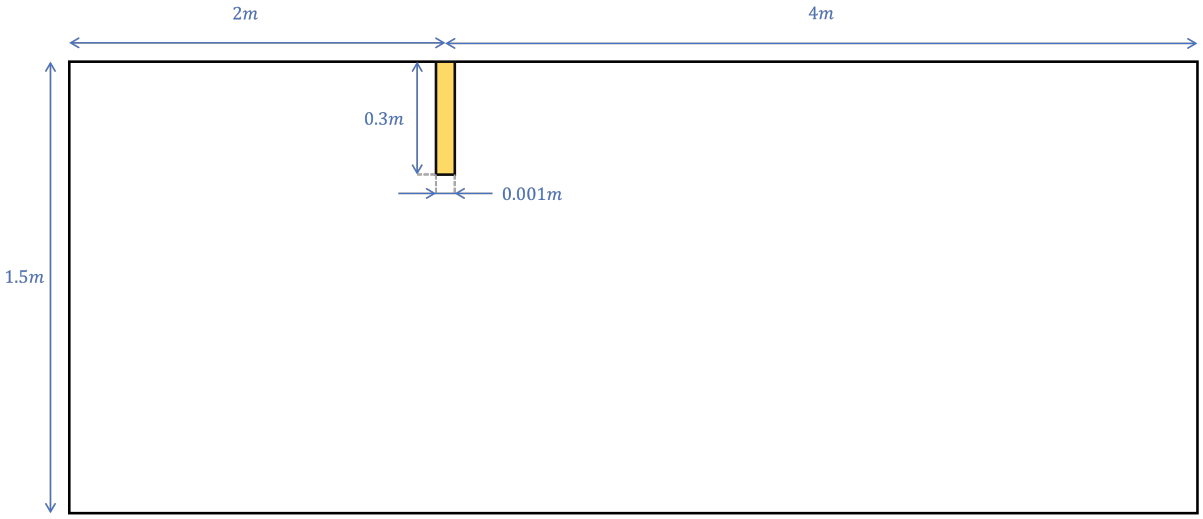


Figure 3.3.4: Domain size

The mesh utilized in this study was generated using the *blockMesh* generator, a tool provided by OpenFOAM (OpenFOAM 2022d). As the name suggests, *blockMesh* allows users to construct a mesh using blocks. The domain is divided into at least one 3D hexahedral block, where each block is defined by 8 vertices representing the corners of a hexahedron. Figure 3.3.4 illustrates the block created for the domain. While it is possible to have the edges of the blocks as straight lines, arcs, or splines, this mesh exclusively employs straight lines as it defines the domain and not the membrane.

The functionality of *blockMesh* is dependent on a single file called *blockMeshDict*, which is situated in the *system* folder. Within the *blockMeshDict* file, six sections are defined: *scale,*

vertices, blocks, edges, boundary, and *mergePatchPairs*. These sections encompass various parameters and specifications necessary for generating the mesh using *blockMesh*.

- *convertToMeters*: This parameter represents the scaling factor applied to the vertex coordinates. In this study, the coordinates are provided in meters, so convertToMeters is set as 1.

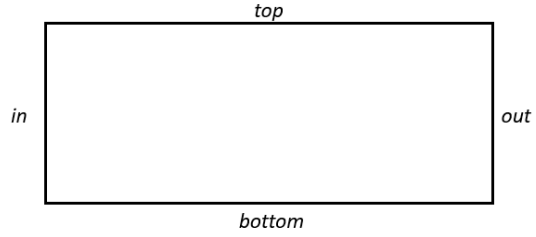
- *vertices*: This section contains a list of vertices that define the corners of the individual blocks. In the present study, there are 8 points designated as *vertex 0* to *vertex 7*. The first vertex is labeled as 0 following the $C++$ convention used in OpenFOAM. Among these points, the first 4 represent the corners of the 2D domain divisions, while the remaining 4 are duplicates with varying z coordinates, enabling the transition from a 2D to a 3D domain.

- *blocks*: Each block is characterized by three parameters. Firstly, the block is identified as a *hex* (hexahedron), and the vertex numbers define the 8 corners of the block. Care must be taken when choosing the order of the vertices, as it determines the orientation of the block. Secondly, the number of cells in each direction (x, y, and z) is specified. Lastly, the cell expansion ratio is entered for each direction of the block. In this study, the *simpleGrading* method is employed, which ensures a uniform expansion in all three directions. The assigned value represents the ratio between the size of the end cell (δ_e) and the size of the start cell (δ_s) along one edge of the block. However, the grid will be regular and the expansion ratios are the (1 1 1).

- *edges*: In this section, each edge connecting two points defined in the *vertices* section is considered to be a straight line by default. If there is a need to define an arc or a spline for an edge, it should be specified in this section. However, in the current study, the blocks within the domain are exclusively composed of straight lines. Therefore, this section is left empty as there are no arcs or splines involved.

- *boundary*: The mesh boundary is divided into regions known as patches, where each patch is defined by a name, a *type*, and a list of *faces*. In the current study, the *type* of patches can be classified as either *patch* (a generic patch), *symmetryPlane* (representing a plane of symmetry), *wall* (for wall conditions), or *empty* (used for the *FrontAndBack* faces to solve the problem in 2D instead of 3D). The boundaries and their corresponding types are illustrated in Figure 3.3.5. The *faces* parameter refers to the list of faces that comprise the boundary.

- *mergePatchPairs*: This section is used to combine blocks, it is not needed in this work so it is left empty.



Boundary name	Boundary type
in	patch
out	patch
top	symmetryPlane
bottom	patch
frontAndBack	empty

Figure 3.3.5: Boundaries types

3.3.2.2 Membrane

As explained previously, to mesh the membrane, the function *snappyHexMesh* was used. *snappyHexMesh* is a powerful meshing utility in OpenFOAM, used for generating high-quality, boundary-conforming meshes for complex geometries. It combines the advantages of structured (hexahedral) and unstructured (tetrahedral) meshing techniques (OpenFOAM 2022e).

The primary purpose of *snappyHexMesh* is to create meshes with precise boundary representation and a efficient resolution near the boundaries, commonly used for simulations involving fluid flow, heat transfer, and other physical phenomena.

The workflow of *snappyHexMesh* involves several steps:

1. **Geometry setup:** The geometry of the problem domain is defined using CAD software or by providing surface definitions in OpenFOAM format. This geometry serves as the input for *snappyHexMesh*.

2. **Snapping:** *snappyHexMesh* snaps the mesh vertices onto the boundary surfaces, ensuring that the mesh conforms accurately to the geometry. This snapping process helps create a high-quality boundary representation.

3. **Mesh generation:** Based on the snapped boundary, SnappyHexMesh generates a mesh consisting of hexahedral cells near the boundary and tetrahedral cells in the interior. Hexahedral cells provide better numerical accuracy and computational efficiency, while tetrahedral cells handle complex geometries and regions where hexahedral cells are not suitable.

4. **Layer addition:** *snappyHexMesh* allows for the addition of prism layers to the boundary regions. These layers capture near-wall effects and improve the resolution of the boundary layer. This feature is particularly useful for capturing boundary layer phenomena in fluid flow simulations.

5. **Mesh refinement:** *snappyHexMesh* provides options for refining the mesh locally in specific regions of interest. This refinement can be based on geometric features, proximity to boundaries, or specific field variables.

In this study, the geometry is included using an .stl file as presented in Section 3.3.1, named membrane. Most of the parameters are left as they are in the base file supplied by OpenFOAM. Modifications were made in the min and max refinement levels and in the umber of buffer layers of cells between different levels of refinement (*nCellsBetweenLevels*) to increase or decrease the size of the refined mesh. The first one was set at *level (2 3)* after some tests and the second one is the one that will be modified in the mesh convergence study 3.4.1

3.3.3 Setup

This section presents the numerical setup used in OpenFOAM. It is largely based on the work presented in the project carried out in the first semester (Calonnec 2022).

Boundary conditions:

Boundary conditions matching the ones implemented in the mesh generation have to be added for the velocity and the pressure in the files U and p, inside the folder *0* (OpenFOAM 2022a). The wanted conditions are, at the inlet (*in*), a uniform incoming velocity and a zero gradient condition on the pressure, and, at the outlet (*out*), a zero gradient condition on the velocity and a uniform pressure set to zero. The boundary conditions are presented in Table 3.3.3. The *symmetryPlane* condition means that the boundary is treated as a plane of symmetry.

Boundary name	U	p
in	fixedValue	zeroGradient
out	zeroGradient	fixedValue
top	symmetryPlane	symmetryPlane
bottom	slip	slip
membrane_ASCII	fixedValue	zeroGradient
frontAndBack	empty	empty

Table 3.3.3: Boundary conditions of U and p

Transport and turbulence properties:

The file named *transportProperties* contains data pertaining to the characteristics of the fluid being used, specifically focusing on its properties. The fluid in question is classified as Newtonian, and its kinematic viscosity is explicitly set at $\nu = 10^{-6} m^2/s$. On the other hand, the file labeled *turbulenceProperties* outlines the details of the turbulence model being employed. For the purposes of this study, the *simulationType* is specifically defined as *laminar*. This choice was made to streamline the workflow and simplify the analysis.

Time and data input/output:

The file called *controlDict* (OpenFOAM 2022g) contains specific parameters related to time and input/output data. It begins by specifying the solver utilized, which in this case is *pisoFoam*. Following that, the Time parameters are implemented, including the *startTime*, *endTime*, *deltaT* (time step), *writeInterval* (timing for output file writing), *writeFormat* (data file format), and the precision. It is crucial to choose an appropriate time step ΔT that satisfies the Courant-Friedrichs-Lewy condition, as explained in Section 3.2, to ensure simulation stability. The time step must be small enough to prevent instability.

Additionally, the *controlDict* file concludes by incorporating functions. One such function is *forceCoeffs*, which calculates the drag and lift coefficients on the model. Another function, *forces*, produces the forces field, in the time directories, alongside the velocity and pressure fields.

Numerical schemes:

The file named *fvSchemes* (OpenFOAM 2022f) serves the purpose of specifying the numerical schemes employed in the simulation. It assigns numerical schemes to various terms, including derivatives and interpolations. A total of 6 terms are associated with specific schemes, as outlined in Table 3.3.4. Here, the *Euler* scheme is employed for the time derivative, which utilizes a first-order, bounded, and implicit Euler scheme. The *Gauss* scheme is utilized for implementing a standard finite volume discretization of Gaussian integration. The *linear* scheme signifies a linear interpolation with central differencing. Regarding the *laplacianSchemes*, the commonly used *Gauss linear corrected* scheme is predominantly employed in OpenFOAM, requiring no modifications.

Name	Definition	Type
ddtScheme	first and second time derivatives	Euler
gradSchemes	gradient ∇	Gauss linear
divSchemes	divergence $\nabla\cdot$	none (Gauss linear)
laplacianSchemes	Laplacian ∇^2	Gauss linear corrected
interpolationSchemes	cell to face interpolations of values	linear
snGradSchemes	component of gradient normal to a cell face	corrected

Table 3.3.4: Numerical schemes in OpenFOAM

Solution and algorithm control:

The file called *fvSolution* plays a crucial role in controlling the equation solvers, tolerances, and algorithms in the simulation. For this study, the default parameters are maintained, without any modifications. The specified algorithm is the PISO algorithm, with a number of correctors set to 2. These settings dictate the computational procedures utilized to solve the equations and ensure convergence within acceptable tolerances.

3.4 Verification of the numerical model

3.4.1 Mesh convergence

A convergence study was performed to assess the parameter values of four different meshes. The objective was to ensure that the selected meshes would yield satisfactory results while minimizing computational time. The varying parameter between the mesh was the number of cells between different levels of refinement, *nCellsBetweenLevels*. Increasing this number would lead to increasing the area with a refined mesh. The four meshes can be found in the appendix and Table 3.4.1 summarises the parameters for each of the meshes.

The convergence analysis involved examining the drag coefficient values obtained from the simulations. By plotting the dimensionless time against the drag coefficient for each of the four meshes, it was possible to determine if they were converging toward a common value. MATLAB was employed to read the relevant files from OpenFOAM, extract the time step and drag coefficient data, and generate the plots. The script used for this purpose can be found in the appendix.

To investigate the impact of grid size, a numerical convergence test was conducted to examine the grid refinement of the four meshes. The converged drag coefficient values, obtained from the plot of C_D against dimensionless time, were plotted against 1/N, where N represents the total number of grid cells in the mesh. For this analysis, the drag coefficient value used was calculated as the mean value of the drag coefficients obtained between 30 and 75 seconds ($\bar{C_D}$). This approach accounted for the expected fluctuations in the coefficient due to the turbulent nature of the flow. Each mesh was evaluated using this procedure. By plotting the drag coefficient mean against the inverse of the number of grid cells, the grid convergence plot revealed which mesh provided the most accurate results with the least computational time. This enabled the selection of the optimal mesh size for achieving accurate and efficient simulations.

	mesh 1	**mesh 2**	**mesh 3**	**mesh 4**
$U[m/s]$	0.13	0.13	0.13	0.13
nCellsBetweenLevels	1	2	4	8
total number of cells	5425	6454	8484	14112

Table 3.4.1: Meshes parameters

3.4.2 Simulation at low Reynolds

To check that the model is working properly a simulation was conducted with a lower Reynolds number. Indeed, when the kinematic viscosity is chosen to be the water viscosity ($\nu = 10^{-6} m^2/s$), the Reynolds number is then calculated as:

$$Re = \frac{Ul}{\nu} = 39000 \tag{3.11}$$

with $U = 0.13 m/s$ and $l = 0.30 m$

This value is higher than the value used in most of the studies that can be found in the literature. Therefore, one simulation using the flat plate was conducted with a viscosity $\nu = 0.00039$ to find a Reynolds number $Re = 100$. This modification was made in the *transportPreoperties* file.

The dimensionless time against the drag coefficient was plotted for the flat plate simulations with low Reynolds and high Reynolds and the mean drag coefficient ($\bar{C_D}$) was calculated. The results of the simulations are presented in Section 5.2.1.2.

CHAPTER

FOUR

GLOBAL MODEL

In this chapter, we will present an analytical investigation of the global model. This study was conducted with valuable insights and guidance from the master thesis supervisor, David Kristiansen, and drew upon various research studies on cables in water, as referenced by Faltinsen and Irvine (O. Faltinsen 1990; Irvine 1990).

4.1 Catenary equation

A catenary refers to a curved shape that depicts the idealized form of a hanging chain or cable. This shape is assumed to be influenced solely by its weight and suspended only at its ends. Although resembling a parabola, the catenary curve is distinct from it. It is determined by establishing a state of equilibrium for the forces acting on a specific chain section.

In the study of a towed boom structure, a similar methodology can be employed. The boom is treated as either a chain or a cable, and the external forces applied include tension and drag force resulting from the towing velocity, denoted as U, see Figure 4.1.1.

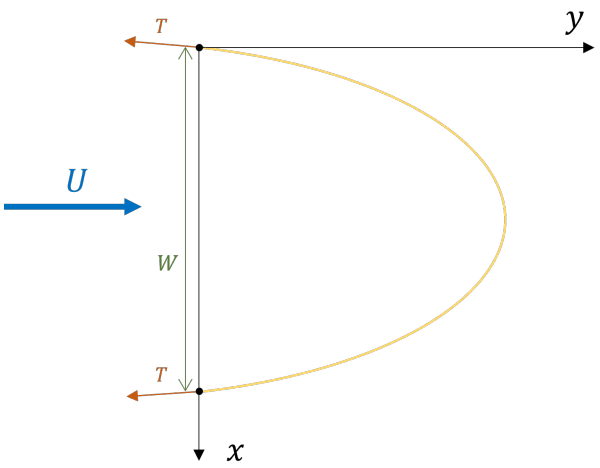


Figure 4.1.1: Representation of the boom as a catenary

The longitudinal position of the boom is denoted as s and $\psi(s)$ represents the angle between the horizontal tangent of the boom and the x-axis. In order to characterize the curve, an equilibrium of forces is established on a small segment Δs of the boom. This equilibrium is illustrated in Figure 4.1.2, where f_n and f_t correspond to the normal and tangential components of the drag force, respectively.

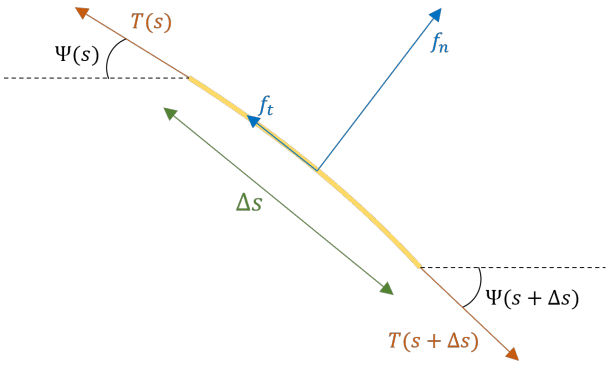


Figure 4.1.2: Forces applied on a boom element

This analysis results in the differential equation:

$$T\frac{d\Psi}{ds} = -f_n \tag{4.1}$$

The drag force is defined by Faltinsen (O. Faltinsen 1990) as:

$$F_D = -\frac{1}{2}\rho C_D d U^2 \tag{4.2}$$

with ρ the density of water, C_D is the drag coefficient, d is the draft of the boom and U the current velocity. The normal drag force is then the drag force with the normal speed:

$$f_n = -\frac{1}{2}\rho C_D d U_n^2 \tag{4.3}$$

The normal speed U_n is defined as $U_n = U sin(\Psi)$. Therefore, putting it in (4.3), the normal drag force is:

$$f_n = -\frac{1}{2}\rho C_D d U_n^2 sin^2(\Psi) \tag{4.4}$$

The differential equation (4.1) defining the shape of the boom becomes:

$$\frac{d\Psi}{ds} = -\frac{1}{2}\frac{\rho C_D d U^2}{T} sin^2(\Psi) \tag{4.5}$$

We introduce the constant $\alpha = \frac{1}{2}\frac{\rho C_D d U^2}{T}$ so (4.5) is:

$$\frac{d\Psi}{sin^2(\Psi)} = -\alpha ds \tag{4.6}$$

The solution for this differential equation is:

$$-cotan(\Psi) = C - \alpha s \tag{4.7}$$

with C a constant. To find C, we know that $\Psi(\frac{L}{2}) = \frac{\pi}{2}$ so we have:

$$C = \alpha\frac{L}{2} \tag{4.8}$$

Finally, using (4.8) and (4.7), the shape of the boom is defined by:

$$\Psi = arcotan\left(\alpha\left(s - \frac{L}{2}\right)\right) \tag{4.9}$$

4.2 Global model implementation

In order to analyze the overall shape of the boom, a Matlab code was developed, as outlined in the appendix. Due to the assumed symmetry, only half of the boom was considered for the study. The half boom was divided into 2690 elements, each represented by Δs. These elements possess coordinates in both the x and y directions. The initial point of the boom is defined as (0,0). To determine the coordinates of subsequent points, the relationships $\Delta x = \Delta s \cdot \sin(\Psi)$ and $\Delta y = \Delta s \cdot \cos(\Psi)$ were utilized, as depicted in Figure 4.2.1.

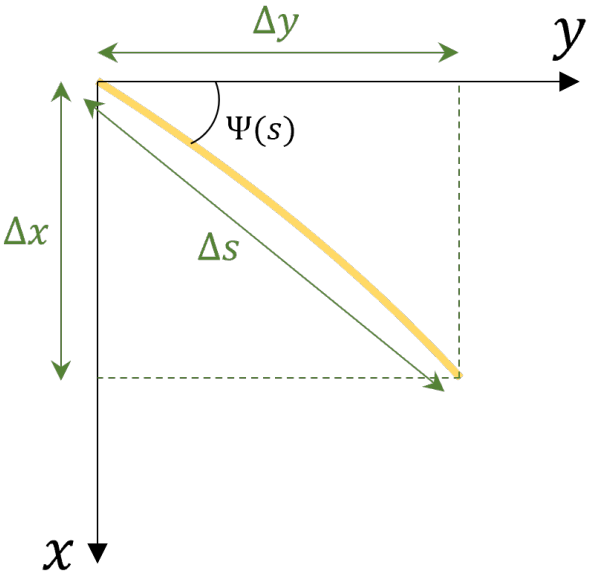


Figure 4.2.1: Coordinates of an element

Therefore, a list of coordinates x and y can be built using

$$\begin{cases} x_{i+1} = x_i + \Delta x \\ y_{i+1} = y_i + \Delta y \end{cases} \tag{4.10}$$

By assigning values to L (length in meters), ρ (density in kilograms per cubic meter), C_d (drag coefficient), d (draft in meters), and T (tension in Newtons), the shape of the boom can be graphically depicted. Furthermore, it is possible to determine the width (W in Figure 4.1.1) of the boom when it is being towed.

The different values used as inputs in the code come from the model. The length is the length of the floating line, the draft is the membrane's draft, the tension is the one measured during the experiments, and the density is chosen to be $1000 kg/m^3$. The drag coefficient comes from the numerical simulations (see Section 3) or from the previous study made in Emma Løyland Vassanyi 2021. In the first case, the drag coefficients are those obtained with curve 2 for 4 inclination angles 0°, 15°, 30°, and 45°, visible in Figure 5.2.13. In the second case, these are presented in Table 4.2.1.

Inclination angle (°)	0	10	20	30	45
Drag coefficient C_D	3.3	3.2	2.8	2.4	1.8

Table 4.2.1: Drag coefficient for a towing velocity of $0.13 m/s$ from Emma Løyland Vassanyi 2021

4.2.1 Strip theory

Strip theory is a simplified method used to analyze the hydrodynamic behavior of marine vessels, particularly in the context of ship resistance and seakeeping characteristics. It is based on the concept of dividing the hull into a series of long, slender strips or panels along the length of the ship.

In strip theory, the hull is divided into a number of sections or strips that are typically parallel to the waterline. Each strip is assumed to behave independently and is analyzed separately. This simplification allows for a more tractable analysis compared to considering the entire hull as a three-dimensional object.

The key assumption in strip theory is that the flow around each strip is two-dimensional, meaning that it is independent of the transverse direction. This assumption is valid for slender bodies such as ships and allows for the calculation of hydrodynamic forces and moments acting on each strip.

As the drag coefficient applied to the membrane varies with its shape, to make the results more accurate, the boom was decomposed into sections, using strip theory. Each section was assimilated with a drag coefficient.

4.2.2 End effect

The drag coefficients used to solve as an input to the code is the one found through the numerical study. However, this study is focused on a 2D section of the boom and therefore does not take into account the end effects. The three-dimensional effects are presented by Faltinsen in Sea loads on ships and offshore structures (O. Faltinsen 1990). The example of a ship is presented. To take into account the three-dimensional effects, the solution is to use a reduced effective incident flow at the ship ends. This corresponds to reducing the two-dimensional drag coefficient. A reduction factor is thus introduced, see Figure 4.2.2. This reduction factor has to be multiplied by the two-dimensional results.

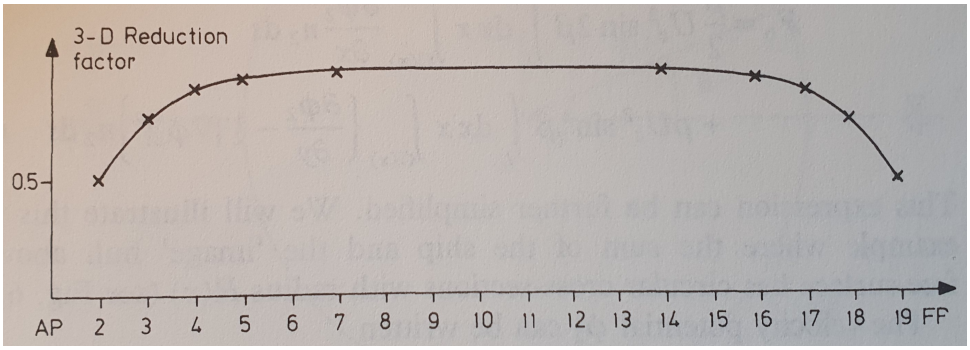


Figure 4.2.2: Reduction factor considering end effects. Source: O. Faltinsen 1990

CHAPTER

FIVE

RESULTS

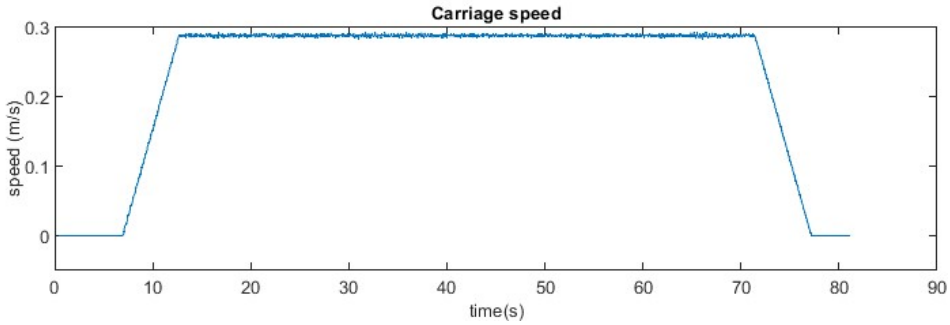
This chapter presents the results obtained during the experiments performed in the laboratories, the 2D CFD simulations run on OpenFOAM, and the analytical study.

5.1 Experimental results

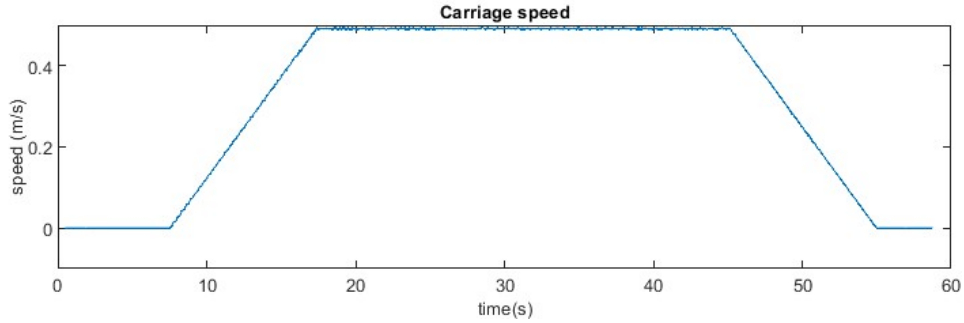
This section will present the time series of the data captured during the experiments (subsections 5.1.1, 5.1.2,5.1.3, 5.1.5) as well as the evolution of the shape of the membrane (subsection 5.1.4) and the calculation of the uncertainty (subsection 5.1.6).

5.1.1 Towing velocity

Figures 5.1.1 display the towing velocities measured during the experiments. The measurements were taken for two distinct input velocities applied in the carriage setup: $0.28m/s$ and $0.48m/s$.



(a) velocity: $0.28m/s$

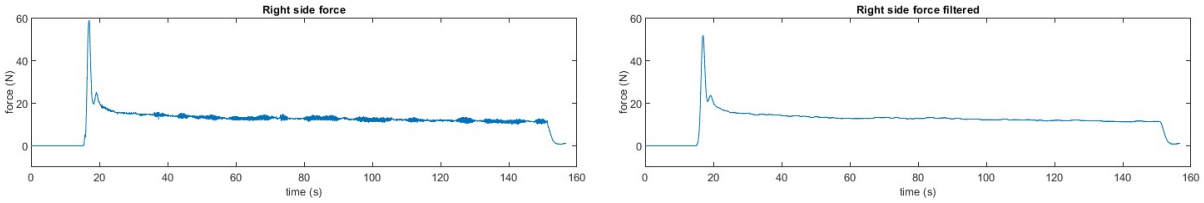


(b) velocity: $0.48m/s$

Figure 5.1.1: Time series of the measured towing velocity for different velocities

5.1.2 Forces

Figure 5.1.2 presents the same time series of the force measure on the right end of the membrane, the first one before applying the filter and the second one after applying the filter presented before, see subsection 2.2.2. As we can see in the figures, the use of the filter makes it easier to analyze the time series. Therefore, the next figures will represent the results after the filter was applied without mentioning it anymore.



(a) Not filtered **(b)** Filtered

Figure 5.1.2: Time series of the measured right side force for a width of $4m$ and a velocity of $0.13m/s$

Figure 5.1.3 represents the right and left forces measured during a simulation. As the value between the right and left sides does not differ a lot for most of the tests, the next plots will only figure the right force.

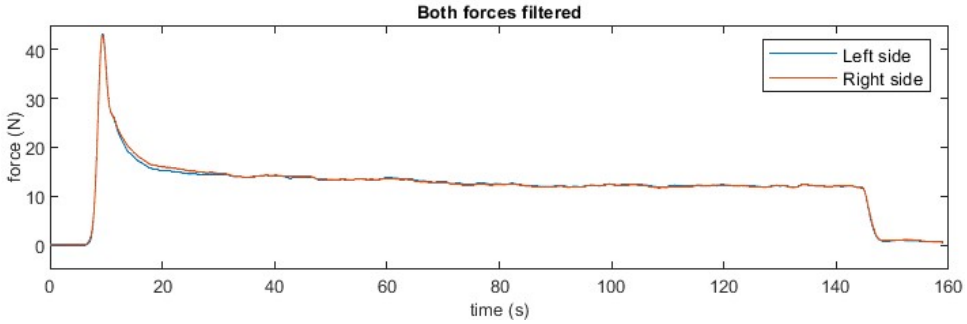
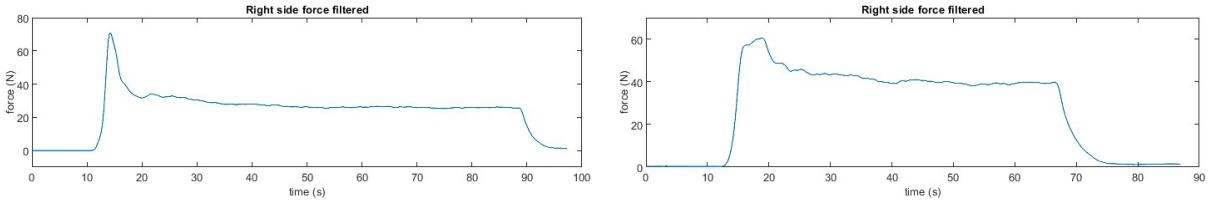


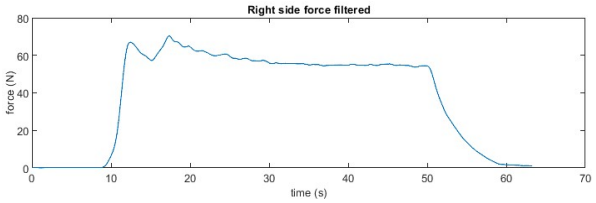
Figure 5.1.3: Time series of the measured right and left forces for a width of $4m$ and a velocity of $0.13m/s$

Figure 5.1.4 presents the force measured on the right side of the membrane when it was fixed with a width of $4m$. These 4 plots present the results for 4 different towing velocities: $0.23m/s$, $0.33m/s$, $0.43m/s$, and $0.53m/s$.

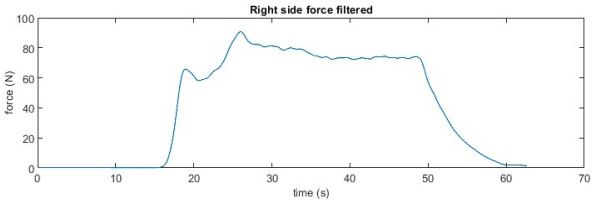


(a) Towing velocity: $0.23m/s$

(b) Towing velocity: $0.33m/s$



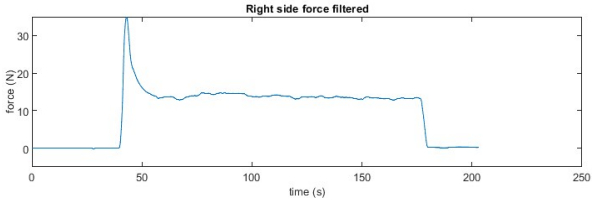
(c) Towing velocity: $0.43m/s$



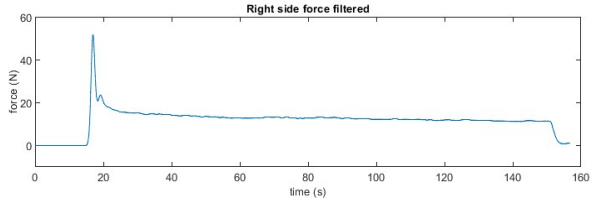
(d) Towing velocity: $0.43m/s$

Figure 5.1.4: Time series of the measured right force for a width of $4m$ and a different velocities

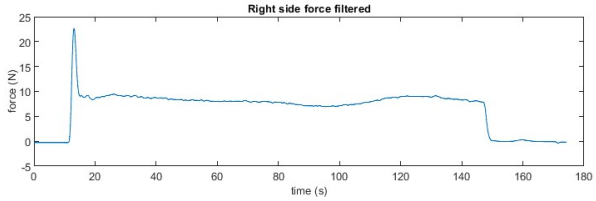
Figure 5.1.5 represents the value of the force on the right end of the membrane for three different widths: $2.5m$, $4m$, and $5m$ towed at the same velocity: $0.13m/s$.



(a) Width: $2.5m$



(b) Width: $4m$

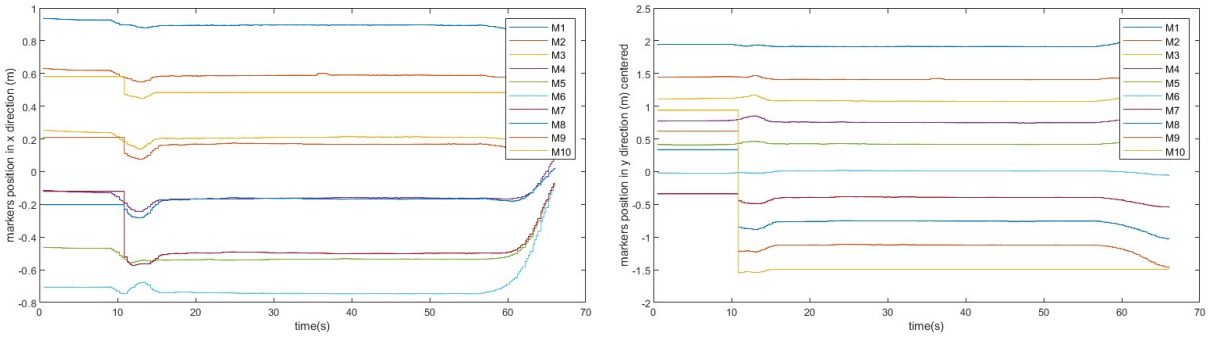


(c) Width: $5m$

Figure 5.1.5: Time series of the measured right force for a towing velocity of $0.13m/s$ and different widths

5.1.3 Time series markers

In Figure 5.1.6, the position of the markers in the x direction and the y direction is shown for a width of $4m$ and a towing velocity of $0.38m/s$. In the $y - dreiction$, the values were shifted to be centered around 0.



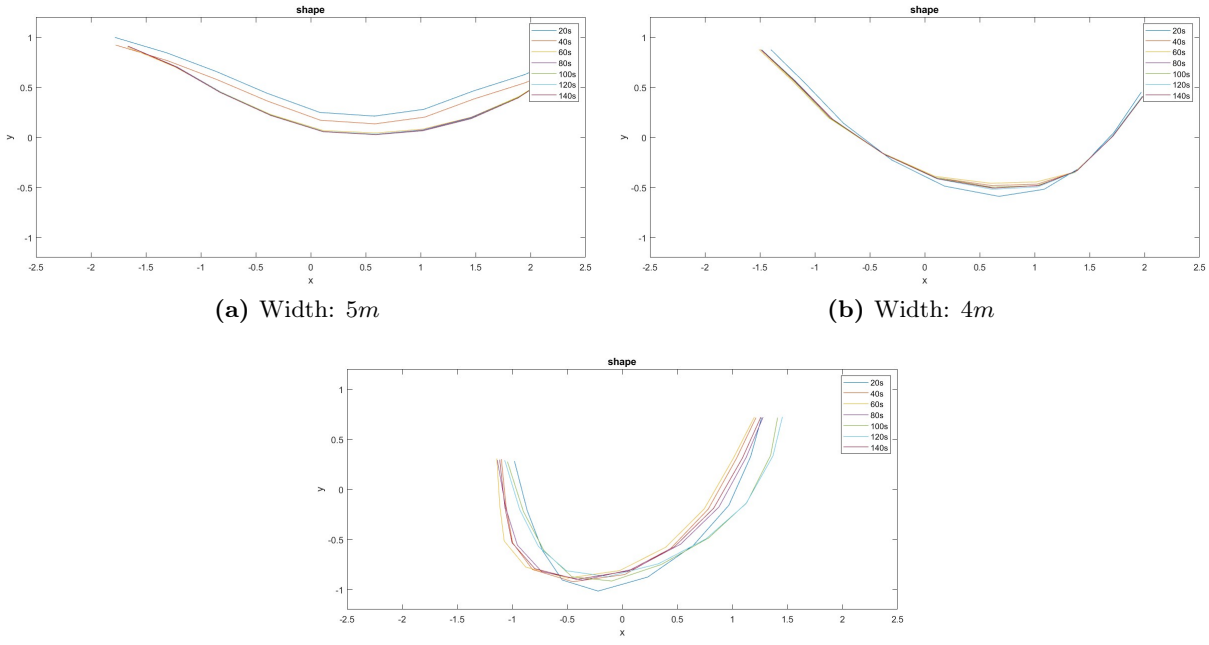
(a) x-direction **(b)** y-direction

Figure 5.1.6: Time series of the position of the markers for a width of $4m$ and a velocity of $0.38m/s$

5.1.4 Shape

Figure 5.1.7 shows the shape of the boom taken for three different widths: $5m$, $4m$, and $2.5m$ when the structure was towed at the same velocity of $0.13m/s$. In each figure, six curves are displayed, corresponding to 6 different times: $20s$, $40s$, $60s$, $80s$, $100s$, $120s$, and $140s$.

In Figure 5.1.8, the shape of the boom is displayed for a width of $4m$ for different towing velocities and $100s$ after starting the run.



(a) Width: $5m$ **(b)** Width: $4m$

(c) Width: $2.5m$

Figure 5.1.7: Shape of the boom towed at a speed of $0.13m/s$ for different widths

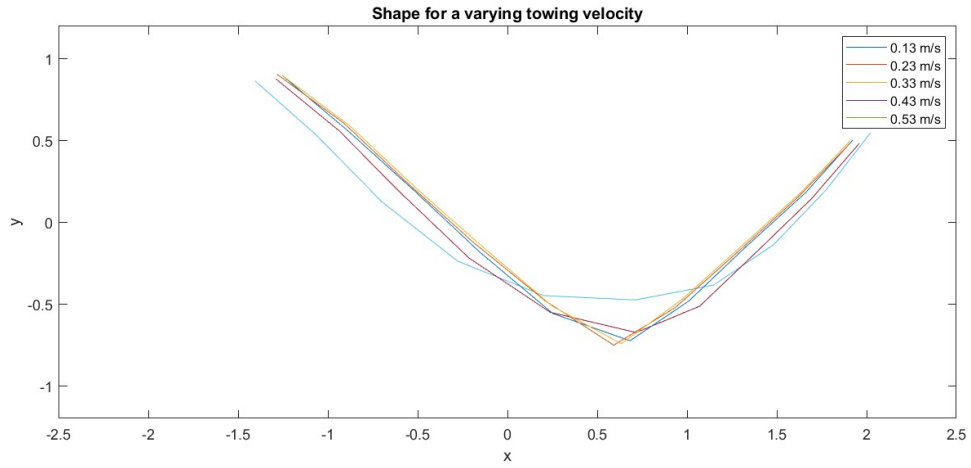
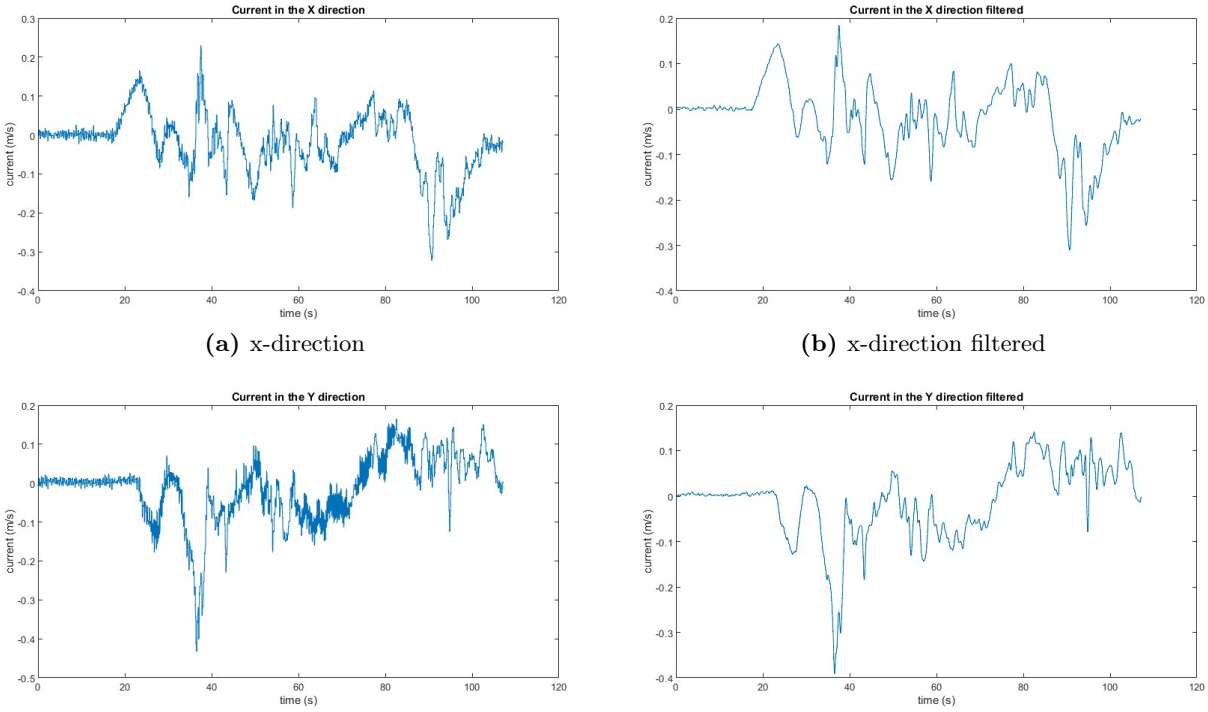


Figure 5.1.8: Shape of the boom towed at different velocities for three widths

5.1.5 Current

This subsection presents some results from the measurements of current. Figure 5.1.9 shows the use of a filter on the current measured during the experiments. As one can see the filter removes the noise that was captured but kept the main variations. Therefore, the study will focus on the filtered values. Figure 5.1.10 and Figure 5.1.11 shows the measured current in x and y directions respectively with different position of the current probe. The value in *cm* given in the caption is the distance measured between the boom and the current probe.



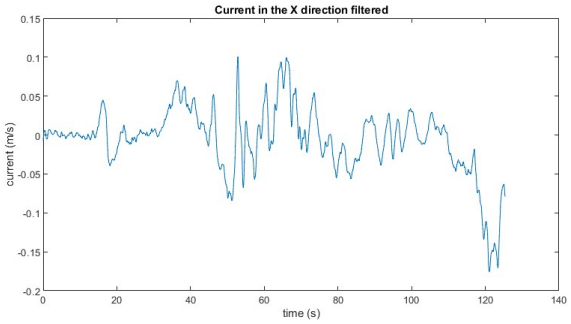
(a) x-direction

(b) x-direction filtered

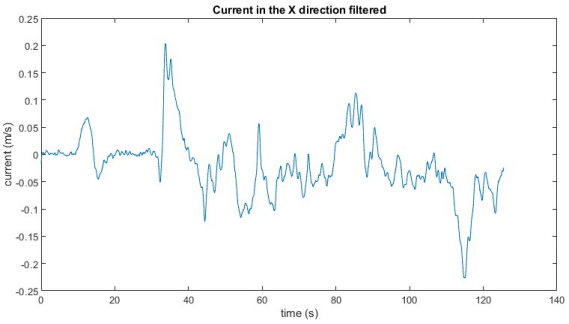
(c) y-direction

(d) y-direction filtered

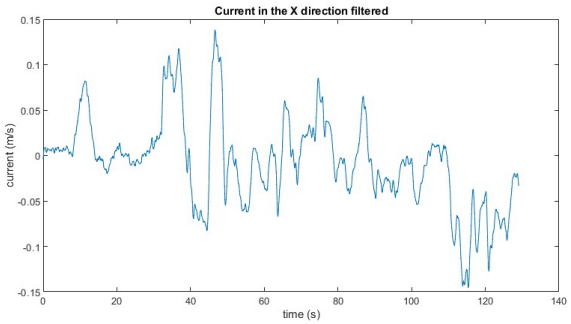
Figure 5.1.9: Current measured in x and y directions not filtered and filtered for a towing velocity of $0.28m/s$ and a width of $4m$



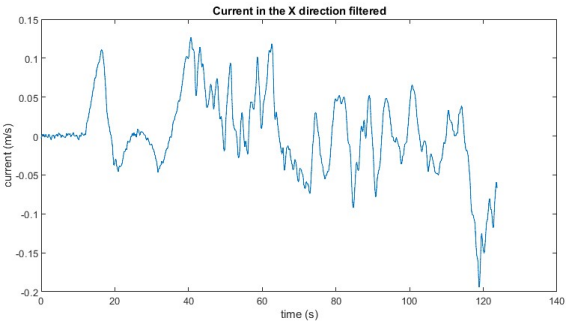
(a) 12cm



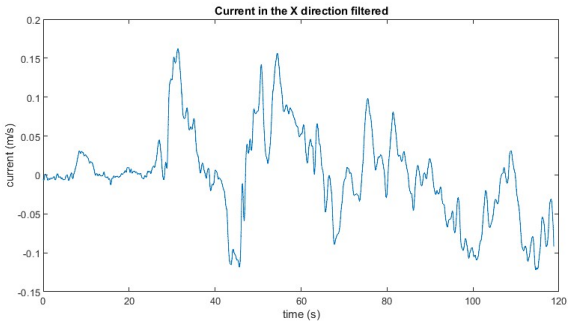
(b) 12cm



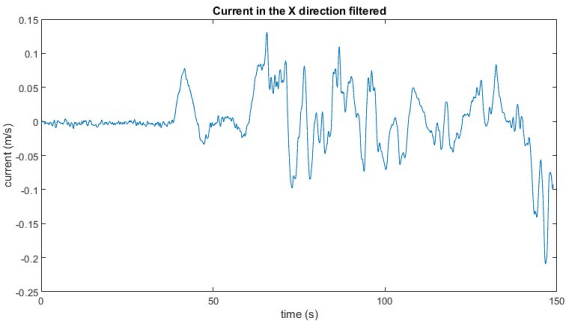
(c) 22cm



(d) 28cm

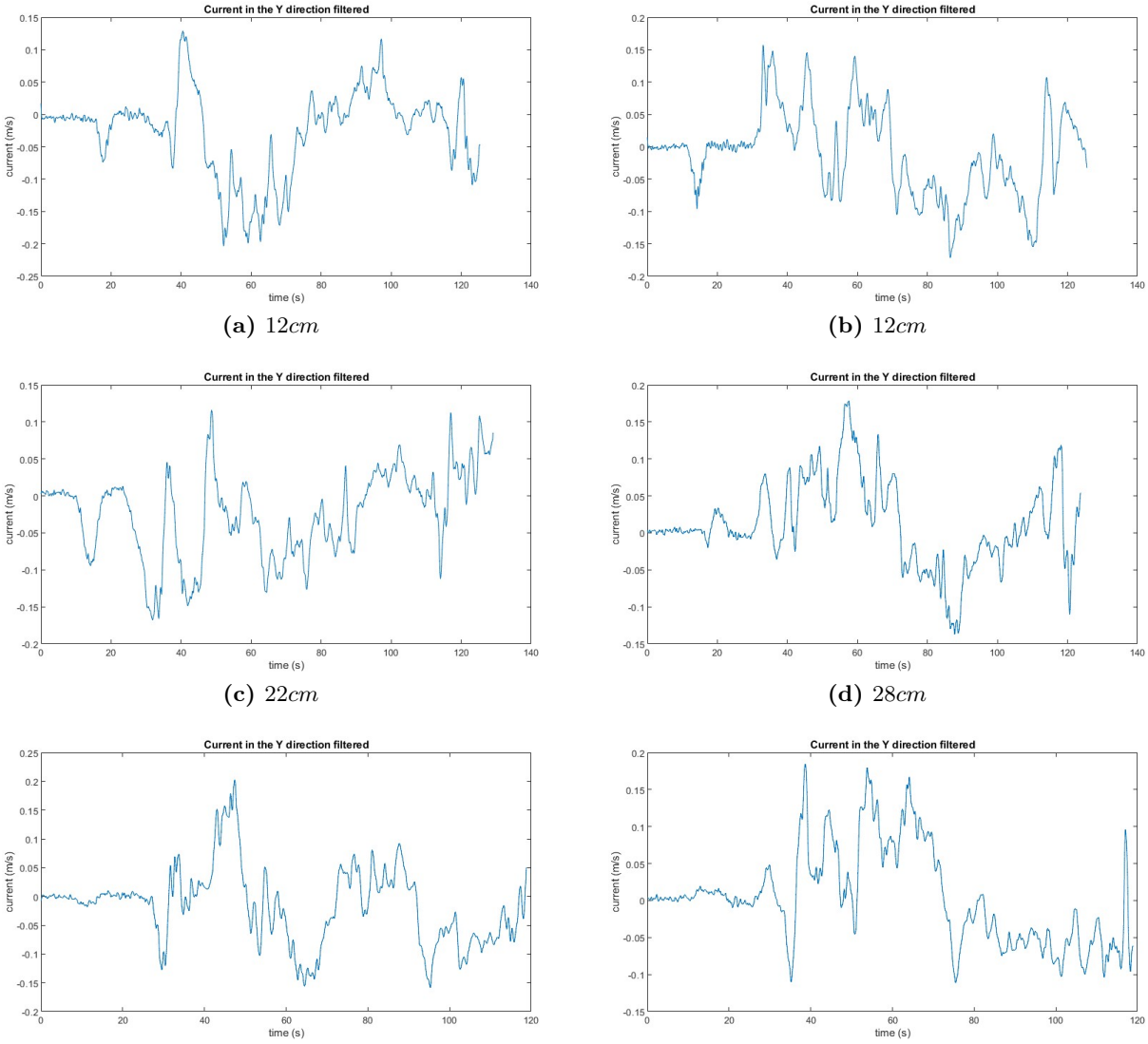


(e) 33cm



(f) 36cm

Figure 5.1.10: Current measured in the x-direction at different distances from the boom for a towing velocity of 0.18m/s and a width of 4m



(a) 12cm **(b)** 12cm

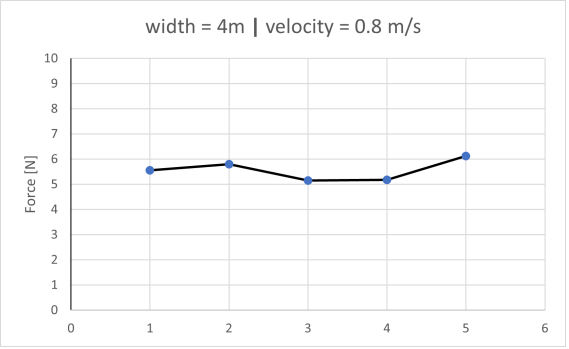
(c) 22cm **(d)** 28cm

(e) 33cm **(f)** 36cm

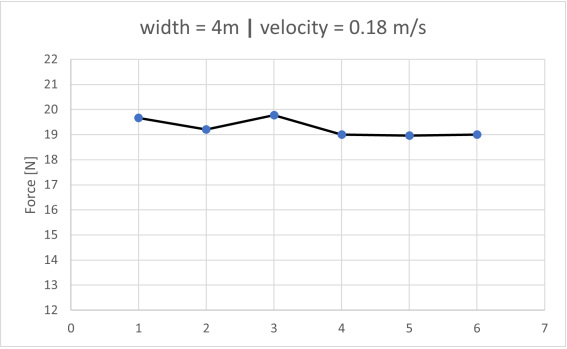
Figure 5.1.11: Current measured in the y-direction at different distances from the boom for a towing velocity of 0.18m/s and a width of 4m

5.1.6 Uncertainty

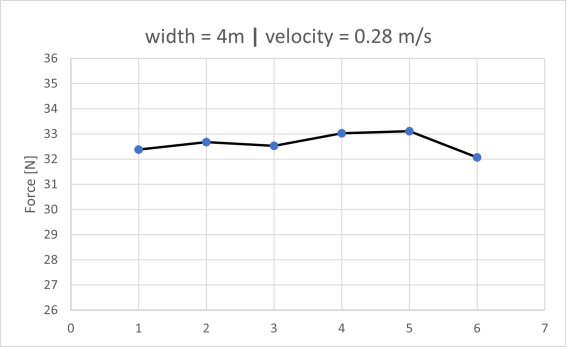
Figure 5.1.12 and 5.1.13 present the repeatability tests performed for different towing velocities and different widths.



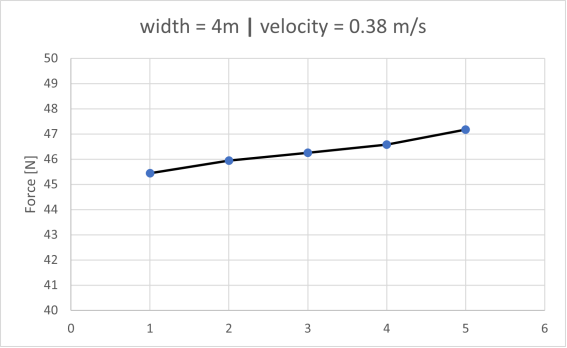
(a) Towing velocity $0.08m/s$



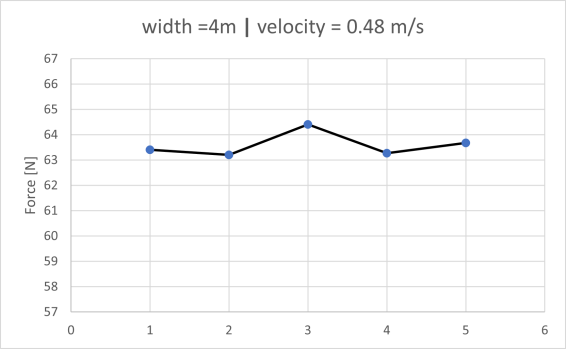
(b) Towing velocity $0.18m/s$



(c) Towing velocity $0.28m/s$

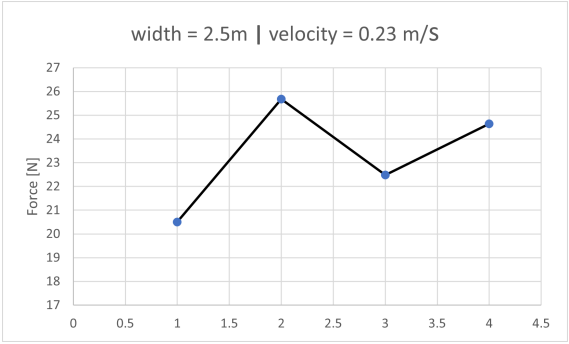


(d) Towing velocity $0.38m/s$

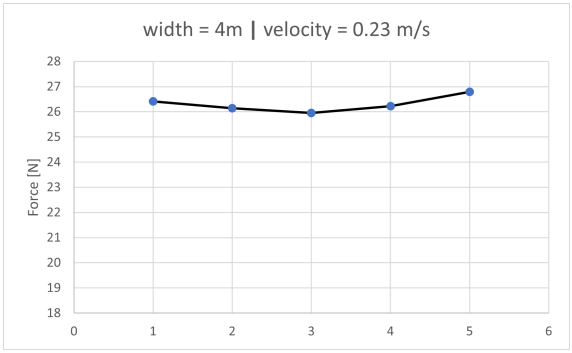


(e) Towing velocity $0.48m/s$

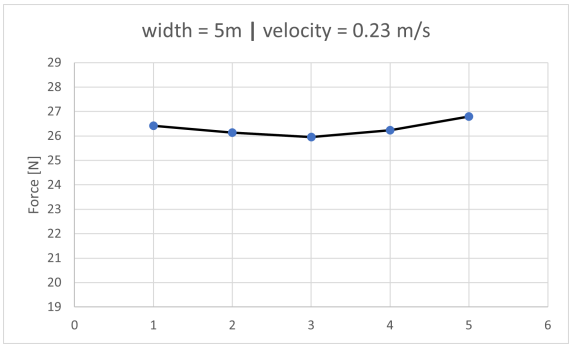
Figure 5.1.12: Repeatability test for a width of $4m$ and varying velocities



(a) Width $2.5m$



(b) Width $4m$



(c) Width $5m$

Figure 5.1.13: Repeatability test for towing velocity of $0.23m/s$ and varying widths

Table 5.1.1 and Table 5.1.2 present the calculation of the uncertainties and errors, with N being the number of tests performed under the conditions. The first table shows the uncertainty calculation for a fixed width of $4m$ and a varying towing velocity. In the second table, the towing velocity is fixed at $0.23m/s$ and the width varies.

U [m/s]	$\bar{F}[N]$	N	$S_F[N]$	$S_{\bar{F}}[N]$	t	P_F	$P_{\bar{F}}$	$e_F[\%]$	$e_{\bar{F}}[\%]$
0.08	5.56	5	0.41	0.19	2.78	1.15	0.51	20.66	9.24
0.13	11.79	7	0.46	0.18	2.45	1.13	0.43	9.61	3.63
0.18	19.21	25	0.31	0.06	2.06	0.64	0.13	3.31	0.66
0.23	26.11	8	0.50	0.18	2.36	1.19	0.42	4.56	1.61
0.28	32.66	19	0.59	0.14	2.10	1.24	0.28	3.79	0.87
0.33	39.19	7	0.60	0.23	2.45	1.47	0.55	3.75	1.42
0.38	46.87	7	1.15	0.43	2.45	2.80	1.06	5.98	2.26
0.43	55.08	6	0.83	0.34	2.57	2.13	0.87	3.87	1.58
0.48	63.60	5	0.49	0.22	2.78	1.35	0.61	2.13	0.95
0.53	74.00	6	0.64	0.26	2.57	1.64	0.67	2.22	0.91

Table 5.1.1: Uncertainty

Width [m]	$\bar{F}[N]$	N	$S_F[N]$	$S_{\bar{F}}[N]$	t	P_F	$P_{\bar{F}}$	$e_F[\%]$	$e_{\bar{F}}[\%]$
2.5	23.33	4	2.31	1.15	3.18	7.34	3.67	31.47	15.73
4	26.11	8	0.50	0.18	2.36	1.19	0.42	4.56	1.61
5	26.31	5	0.32	0.14	2.77	0.88	0.40	3.36	1.50

Table 5.1.2: Uncertainty

5.1.7 Visual observations

Nothing was added to the water to be able to visualize the flow properly. However, some visual observations could still be made. First, the end effects were clearly visible during the experiments. Water was escaping down the sides of the structure, creating significant turbulence. This phenomenon was all the more visible during high towing velocity tests.

In addition, one could see during the runs that the membrane was flipping to a horizontal position. Planning, as described in Section 1.3.2 was occurring. No precise measurements of this deformation were done during the tests but this still influenced the forces applied to the model.

5.2 Numerical results

This section presents the numerical results from the simulation computed with OpenFOAM, presented Chapter 3. Most of the simulations will be performed using the mesh 3.

5.2.1 Verification of the numerical model

In this subsection, the results from the simulation preformed to verify the quality of the numerical model will be presented.

5.2.1.1 Mesh convergence

Figure 5.2.1 shows the plot of the drag coefficient C_D as a function of the dimensionless time $\hat{t}$. In Figure 5.2.2, the mesh convergence plot is presented. It is done by plotting the mean drag coefficient as a function of $1/N$ with N being the number of cells in the mesh. Thus, $1/N$ is the highest for mesh 4 and the smallest for mesh 1.

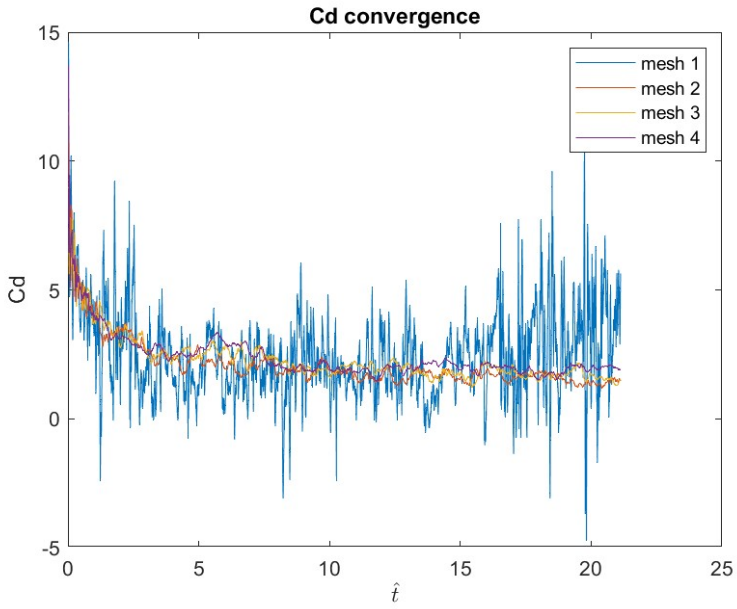


Figure 5.2.1: Convergence of the drag coefficient C_D

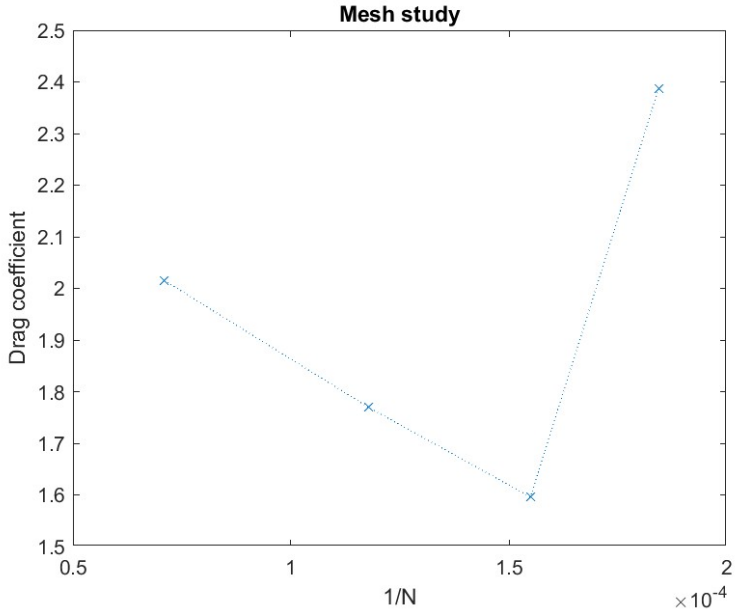


Figure 5.2.2: Mesh convergence plot of numerical C_D

5.2.1.2 Simulation with a flat plate

Figure 5.2.3 presents the drag coefficient of a flat plate normal to current. The flat plate is $0.3m$ long, $0.001m$ thick and the current is $0.13m/s$. For the low Reynolds number, the viscosity is set at $0.00039m^2/s$ so the $Re = 100$. For the high Reynolds, the viscosity is the water viscosity, $\nu = 10^{-6}m^2/s$

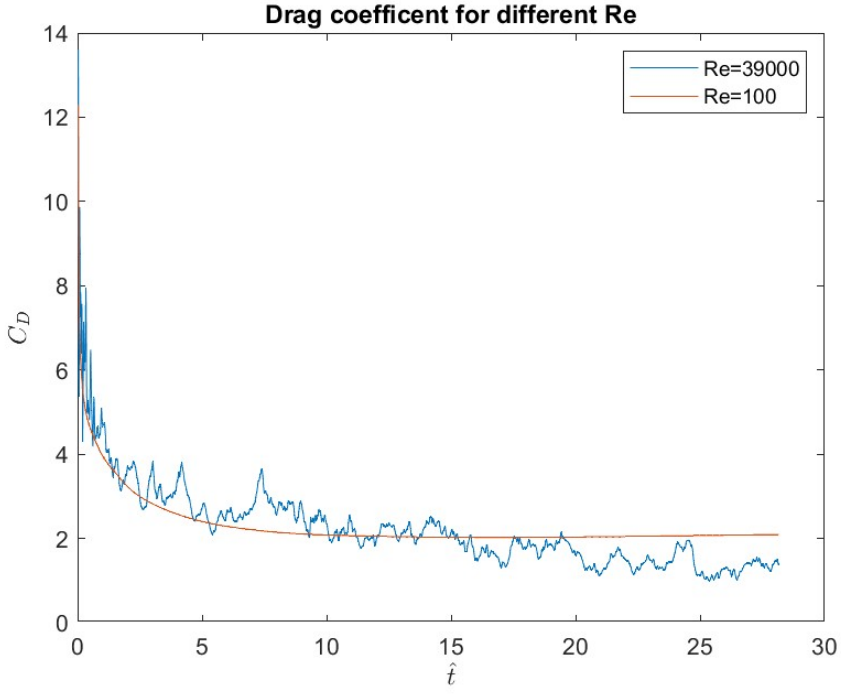


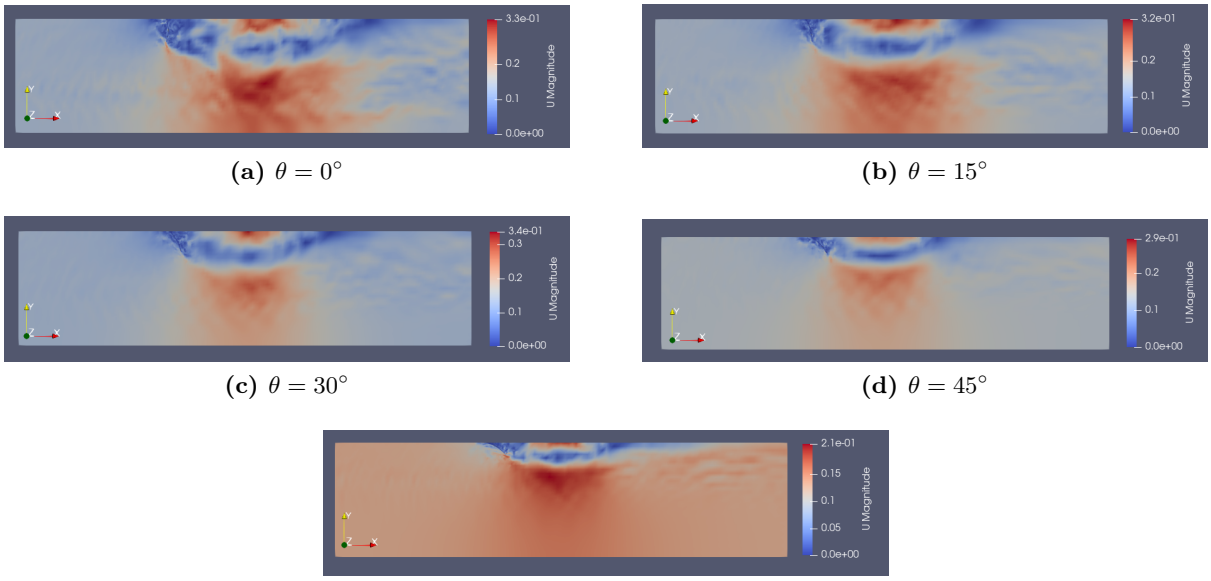
Figure 5.2.3: Drag coefficient C_D of a flat plate normal to current for different Reynolds numbers

5.2.2 Contour plots

In this subsection, plots from the simulations will be presented. The current velocity is always $0.13m/s$ and the shape of the membrane will change.

5.2.2.1 Velocity magnitude

Figure 5.2.4 shows the velocity contours of the flow for the second curve for different inclination angles: 0°, 15°, 30°, 45°, and 60°. Figure 5.2.5 presents the velocity contours of the flow for the first and third curves for an inclination angle of 15°.

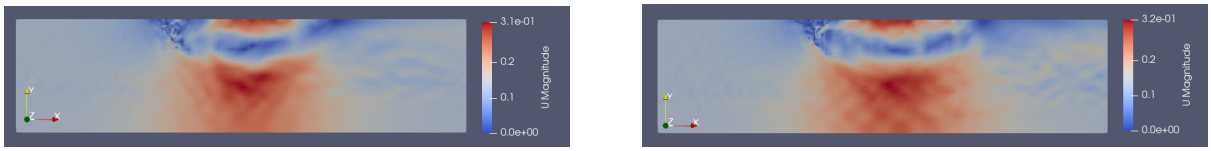


(a) $\theta = 0°$ **(b)** $\theta = 15°$

(c) $\theta = 30°$ **(d)** $\theta = 45°$

(e) $\theta = 60°$

Figure 5.2.4: Velocity contour for the second curve for 5 inclination angles



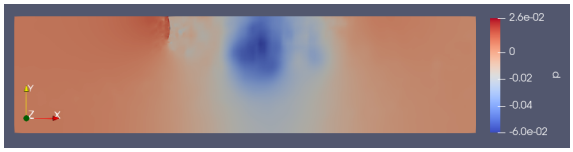
(a) Curve 1 **(b)** Curve 3

Figure 5.2.5: Velocity contour for the first and third curve with an inclination angle of 15°

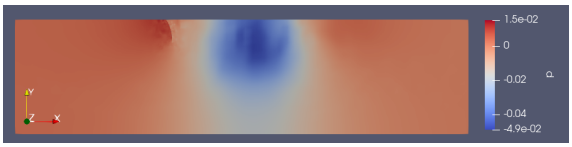
5.2.2.2 Pressure

In Figure 5.2.6, the pressure contours of the flow are depicted for the second curve, illustrating various inclination angles: 0°, 15°, 30°, 45°, and 60°.

In Figure 5.2.7, the pressure contours of the flow are shown for the first and third curves at an inclination angle of 15°.

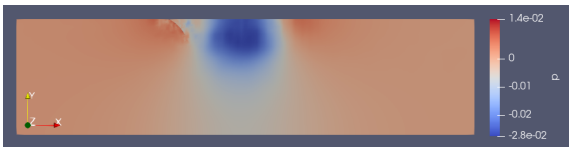


(a) $\theta = 0°$



(b) $\theta = 15°$

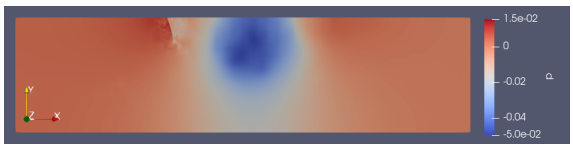
(c) $\theta = 30°$



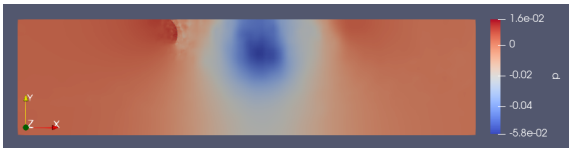
(d) $\theta = 45°$

(e) $\theta = 60°$

Figure 5.2.6: Pressure contour for the second curve for 5 inclination angles



(a) Curve 1



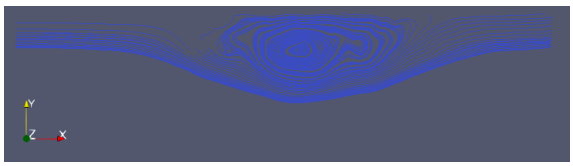
(b) Curve 3

Figure 5.2.7: Pressure contour for the first and third curve with an inclination angle of 15°

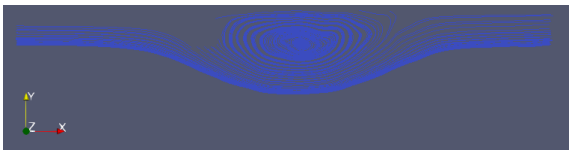
5.2.3 Streamlines

Figure 5.2.8 displays the streamlines of the flow across the domain, focusing on the second curve. The figure showcases different inclination angles, including 0°, 15°, 30°, 45°, and 60°.

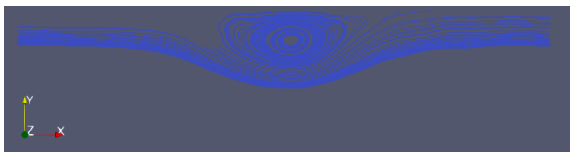
Figure 5.2.9 presents the streamlines of the flow near the membrane for the first, second, and third curves. The figure specifically emphasizes an inclination angle of 15°. As the membrane is too thin to be correctly visualized, a yellow line has been added on top.



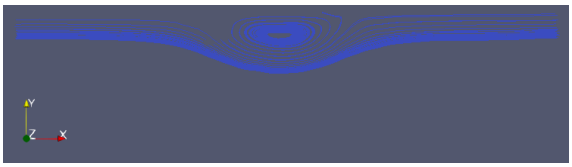
(a) $\theta = 0°$



(b) $\theta = 15°$



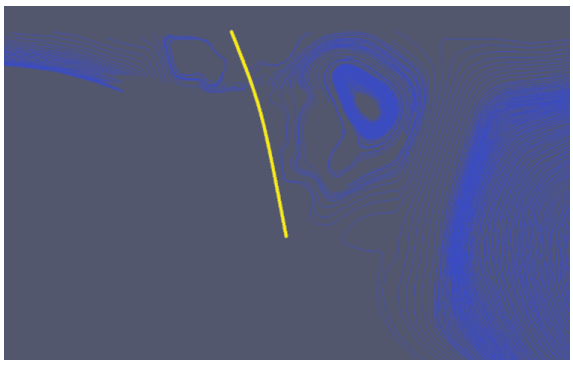
(c) $\theta = 30°$



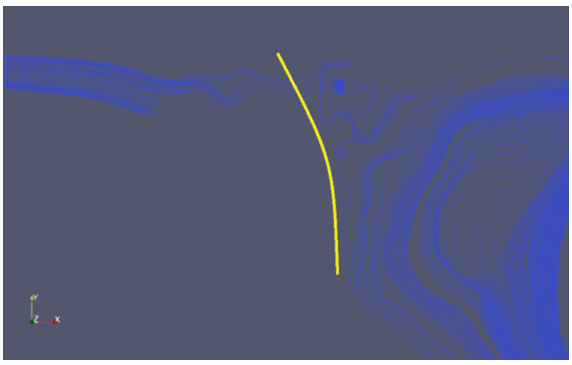
(d) $\theta = 45°$

(e) $\theta = 60°$

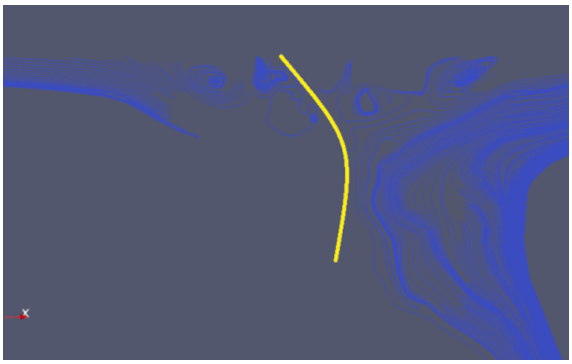
Figure 5.2.8: Streamlines in the doamain



(a) Curve 1



(b) Curve 2



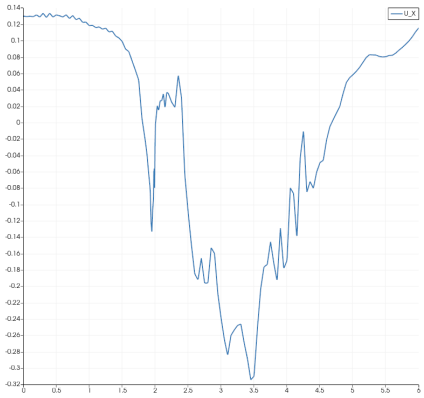
(c) Curve 3

Figure 5.2.9: Streamlines around the membrane

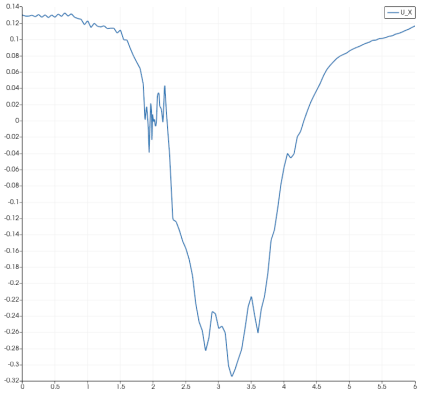
5.2.4 Velocity profiles

Figure 5.2.10 showcases the velocity profiles of the horizontal velocity along the surface for the second curve. The figure presents a range of inclination angles: 0°, 15°, 30°, 45°, and 60°.

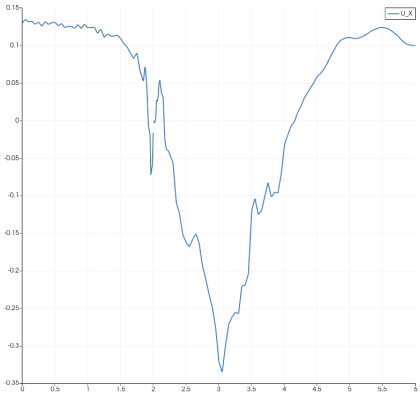
In Figure 5.2.11, the velocity profiles are depicted for the first and third curves, with an inclination angle of 15°.



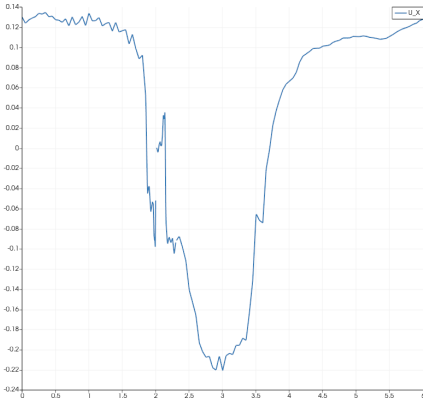
(a) $\theta = 0°$



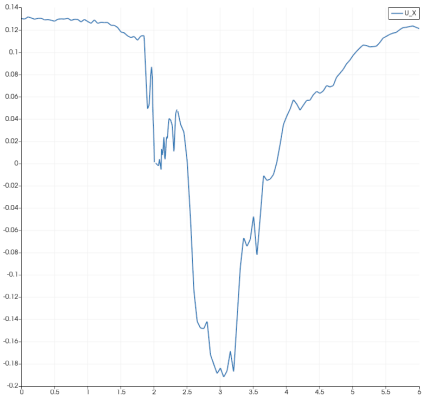
(b) $\theta = 15°$



(c) $\theta = 30°$

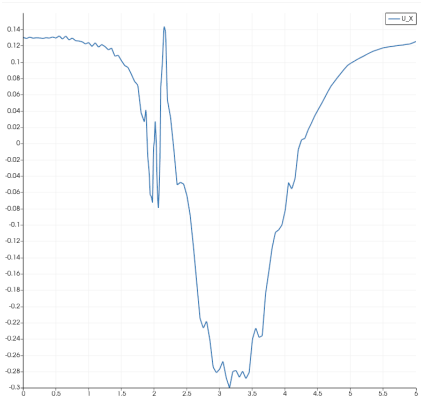


(d) $\theta = 45°$

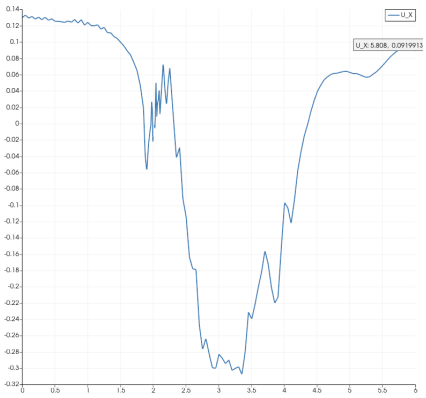


(e) $\theta = 60°$

Figure 5.2.10: Velocity profiles for the second curve for 5 inclination angles



(a) Curve 1

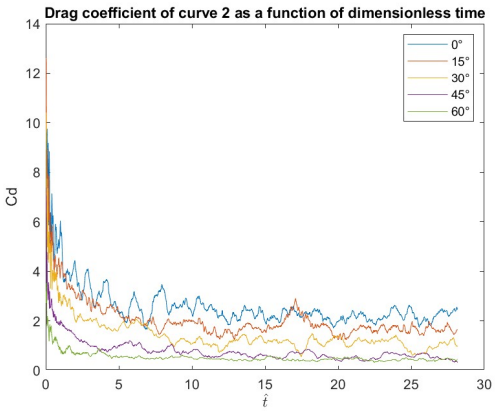


(b) Curve 3

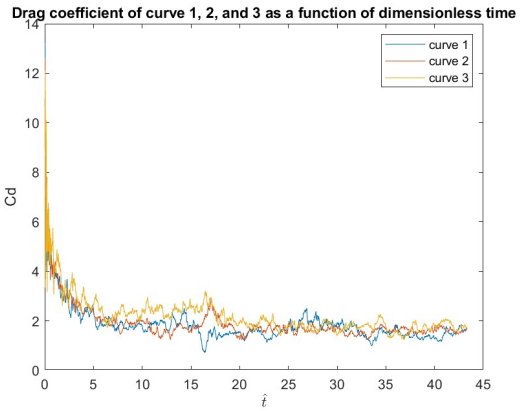
Figure 5.2.11: Velocity profiles for the first and third curve with an inclination angle of 15°

5.2.5 Drag coefficient and forces

Figure 5.2.12a and Figure 5.2.12b present the drag coefficients throughout time. In the first one, the results for curve 2 and different inclination angles are displayed. In the second one, the inclination angle is 15° and the curve varies.



(a) Mean drag coefficient C_D of the curve 2 for different inclination angles as a function of dimensionless time



(b) Mean drag coefficient C_D as a function of dimensionless time for different curves with an inclination angle of 15°

Figure 5.2.12: Drag coefficient as a function of dimensionless time

In Figure 5.2.13 the mean drag coefficient of each simulation is plotted. It was calculated in the steady state. In Figure 5.2.14, only four mean drag coefficients are represented as a function of the distance between the horizontal axis and the center of the membrane D and for an inclination angle of 15°.

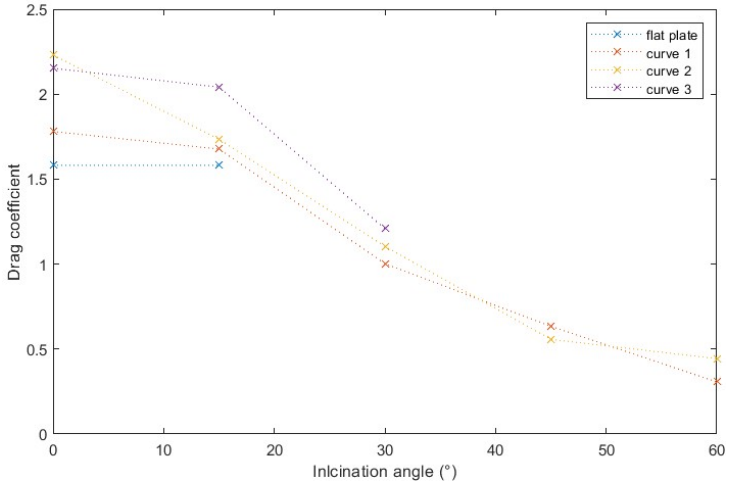


Figure 5.2.13: Mean drag coefficient C_D for different curves and different inclination angles

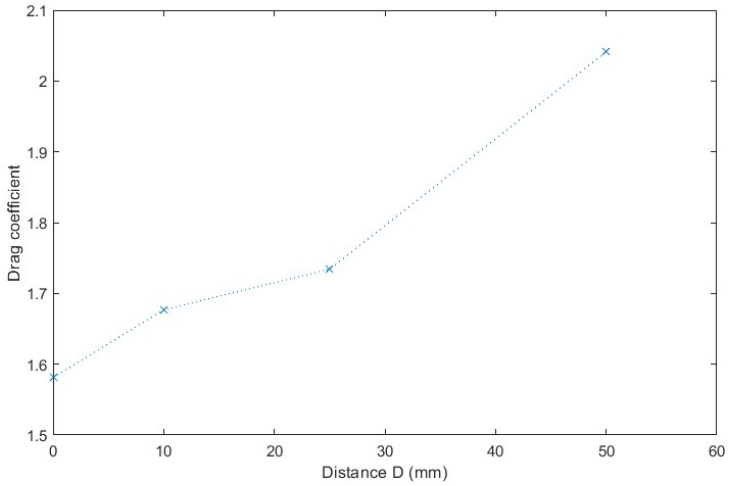
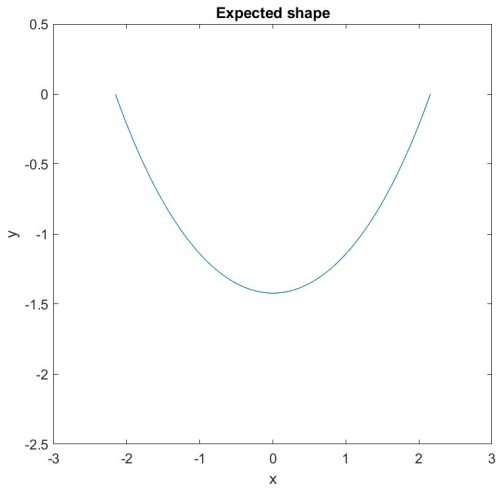


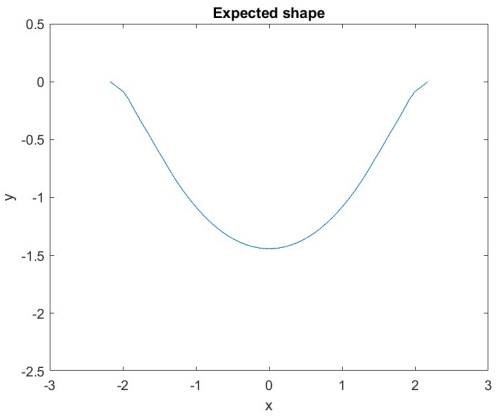
Figure 5.2.14: Mean drag coefficient C_D as a function of the distance between the horizontal axis and the center of the membrane D with an inclination angle of 15°

5.3 Global model

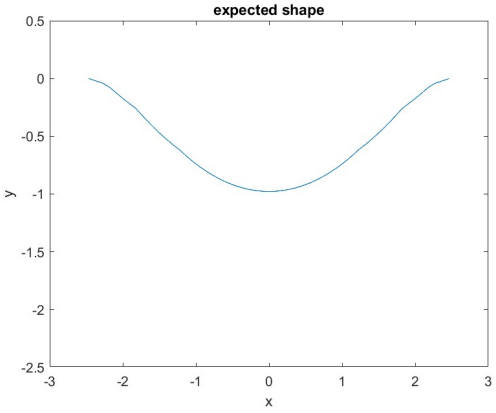
Figure 5.3.1 presents the expected shape of the boom based on the global model for three different choices of the drag coefficient. In the first plot, strip theory is not used and only one drag coefficient $C_D = 2.6$ is applied to the whole boom. In the second plot, the coefficients are taken from Emma Løyland Vassanyi'w master thesis Emma Løyland Vassanyi 2021. In the last plot, the drag coefficients are the ones calculated during the numerical study, for curve 2. The current velocity is $0.13m/s$ and the tension is $12.10N$ as it is the force measured during the experiments under these conditions.



(a) Shape of the boom without strip theory for a current velocity of $0.13m/s$



(b) Shape of the boom with experimental drag coefficient for a current velocity of $0.13m/s$



(c) Shape of the boom with numerical drag coefficient for a current velocity of $0.13m/s$

Figure 5.3.1: Expected shape of the boom for three different choices of drag coefficient

CHAPTER

SIX

DISCUSSION

6.1 Experimental results

6.1.1 Towing velocity

The plots of the towing velocities Figure 5.1.1 shows that the value of the towing velocity is close to the velocity input in the software used to control the carriage. However, the mean value of the measured towing velocity during the steady state in the first plot is $0.2868 m/s$ and in the second plot $0.4915 m/s$. Therefore, the real value is slightly higher than that entered in the software. This was the case for every test, and the difference between the desired velocity and the one measure was always the same for the same entered velocity. In Table 6.1.1, the value entered in the software and the one measured are compared. The difference between the two values increases when the towing velocity increases however the error stays approximately the same and is always low.

Desired velocity [m/s]	**Measured velocity [m/s]**	**error [%]**
0.08	0.082	2.50
0.13	0.1331	2.38
0.18	0.1843	2.39
0.23	0.2355	2.39
0.28	0.2868	2.43
0.33	0.3379	2.39
0.38	0.3891	2.39
0.43	0.4403	2.40
0.48	0.4915	2.40
0.53	0.5427	2.40

Table 6.1.1: Measured towing velocities

The plots in section 5.1.1 also show the noise generated by the carriage. One can clearly see the minor variations during the steady state. However, these variations can easily be ignored using a filter and will therefore not be a hindrance for the rest of the study.

The last information coming from the figures is the duration of the steady state. When the speed increases, the steady-state duration decreases and the structure will then be in a steady state for a shorter duration, which can complicate the study. For the lowest towing velocity, the steady state lasts $223 s$ while for the highest it lasts $23 s$.

6.1.2 Forces

As presented in Figure 5.1.2, a filter was applied to the time series of the forces. The details of this filter are presented in Section 2.2.2. The filter enables the removal of most of the noise measured without influencing the value of the forces. As explained in Section 5.1.2, the other plots present the results after the use of the filter.

Both right and left end forces were measured during the experiments, as shown in Figure 5.1.3. The two forces are quite similar for all the tests that have been performed thus, only the right force is then plotted. However, when the average force will be mentioned later, it will be the average force between the two sides, calculated during the steady state, in order to be as accurate as possible.

The time series of the force measured during the experiments (Figure 5.1.4 and 5.1.5) shows a significant increase at the start of each run before decreasing and stabilizing. This is due to the acceleration of the carriage before reaching the desired velocity. This large value of the force during acceleration is the reason why the sensors could not be more precise. Nevertheless, they were precise enough to obtain a correct value of the mean force during the steady state.

In Figure 6.1.1, the mean forces measured during the tests when the width was $4m$ are plotted as a function of the towing velocity (red crosses). The value of the mean force is the mean value of all the tests performed under the same conditions. One can see in the figure that the mean force increases as the velocity increases. The drag force is defined as $F_D = \frac{1}{2}\rho C_D d U^2$, therefore, the force should increase with U^2. Thus, a quadratic polynomial model is plotted over the data as well as the confidence bounds. The value of the force tends to follow this pattern as expected.

The force also has a tendency to decrease when the width increases as one can see in Figure 5.1.5.

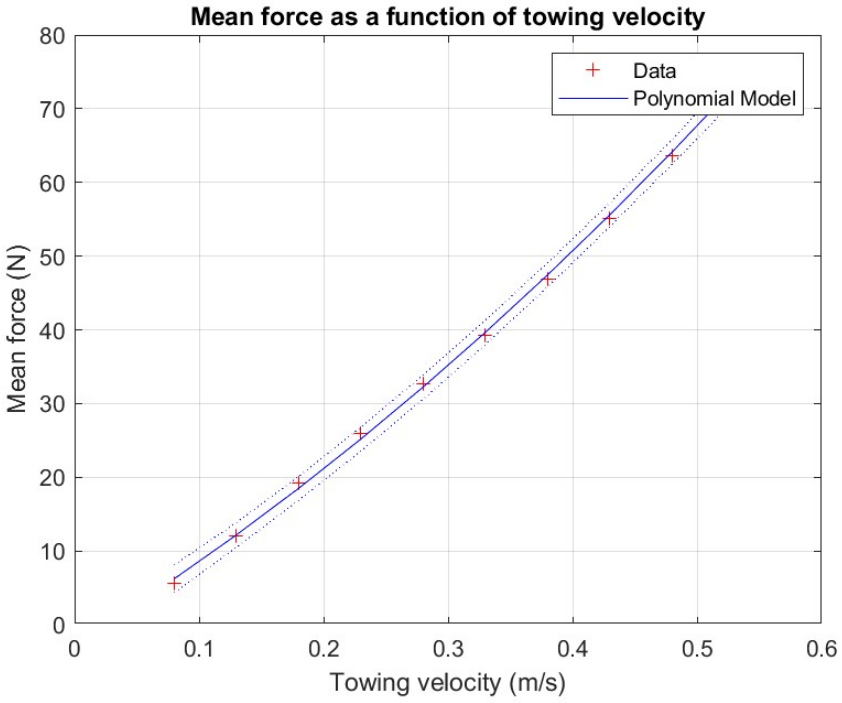


Figure 6.1.1: Mean force as a function of velocity for a width of $4m$

6.1.3 Time series markers

Figure 5.1.6 presents an example of the data obtained from the markers. These measurements show whether the cameras correctly tracked each marker throughout the test. During some

tests, the cameras may have lost certain markers or associated them incorrectly.

However, the data recorded is not directly relevant. They are used to obtain the shape of the boom at a given time by plotting the x and y coordinates of each marker, as done in Section 5.1.4.

6.1.4 Shape

In Figure 5.1.7, the shape of the boom is presented for the three different widths and at six times between $20s$ and $140s$. The first time was chosen to be $20s$ as before that, the membrane was not stable and as this study focuses on the steady state, it would not make sense to display it. For the same reason, the last time taken is $140s$ and not later, as the membrane kept moving after the carriage stops.

When the width is $5m$ or $4m$, the shape of the boom is quite stable. In the first case, it reaches its permanent shape between 40 and 60. In the second case, the structure also reaches its permanent shape around $40s$ and stays stable after. These observations can also be done for the other experiments with the same widths.

When testing with the smallest width, the results weren't as conclusive. As one can see in Figure 5.1.7c, the structure tends to oscillate during the run. For all the tests in this case, regardless of the towing velocity, the same observation was made. This might be due to manufacturing errors in the model. Indeed, the floating line is made of discontinuous floats and is therefore not regular. This can lead to uneven deformation of the membrane.
In Figure 5.1.8, one can observe the variation of the steady-state shape of the boom for different towing velocities. When the velocity increases, the shape of the structure becomes more triangular than parabolic. The distance between the middle of the boom and the attachment point in the y-direction increases with the velocity.

6.1.5 Current

As with force measurements, a filter was used for current measurements. Figure 5.1.9 displays the difference between the current measurements before and after applying the filter. Unlike forces, the current has smaller variations, which can make it difficult to apply a filter. Thus, several filters were tried before choosing the one presented in Section 2.2.4.

The current data is difficult to analyze due to the amount of variation. The figures displaying the x-direction current show first an increase of the current that follows the increase of the velocity. After a few seconds, the first turbulence appears, creating fluctuation. Different filters and transformations were used to try to obtain any information from these measurements but nothing significant came of it. The only thing one can conclude is that the flow is turbulent behind the structure.

6.1.6 Uncertainty

In Figure 5.1.12, the force is plot for different test conditions to test the repeatability of the tests. One can see that the spread of the force in Subfigure 5.1.12d is more significant than the three others, followed by the one in Subfigure 5.1.12e. That might be due to the increasing velocity. However, if the spread is compared to the value of the force, then, the lowest towing velocity has the most important fluctuation. Nevertheless, the spread is, most of the time, small compared to the value of the force. Therefore, it can be considered that for a $4m$, at any towing velocities, the tests are repeatable.

In Figure 5.1.13, the variations are way more important in plot 5.1.12a than in the others. Plot 5.1.12a displays the force (in N) measured during the test with a width of $2.5m$ and a towing velocity of $0.23m/s$. As explained in Section 6.1.4, the tests performed with the $2.5m$ width were not conclusive regarding the shape of the boom as it would move during the experiment,

without reaching any steady state. This oscillation affected the value of the forces measured by the sensor. The difference between the left and right sides is more important as well as the variation between the tests. This leads to conclude that the tests performed with the $2.5m$ width can not properly be used in the study.

The repeatability tests performed for the $4m$ and $5m$, Figure 5.1.13b and 5.1.13c, are, on the contrary, successful. The spreads of the figures are small compared to the measured values.

These tests also prove that there is a correlation between the value of the force and the value of the towing velocity as well as between the value of the force and the width, as has been mentioned before.

Table 5.1.1 presents the value of the uncertainties, presented in Section 2.2.6 for the different towing velocities when the width is $4m$. As expected, the runs with the lowest towing velocity have the lowest standard deviation but the highest uncertainty. The uncertainty of a single test, in this setup, reaches 20.66% which is a significantly large value. For a towing velocity of $0.13m/s$ the error is 9.61% which is again significant, but remains correct. The lowest towing velocities have the largest errors. However, for the other towing velocities, the uncertainty of a single test is always below 5%, except for $0.38m/s$ where it is 5.98%. To reduce the uncertainty, several tests should be completed for each case, before averaging the data. The average data would have fewer uncertainties as shown by the values of the $e_{\bar{F}}$, which are smaller than the values of e_F. Moreover, the tests with the lowest uncertainties are the ones repeated the most.

The uncertainties of the tests performed at a towing velocity of $0.23m/s$ for various widths are presented in Table 5.1.2. The results agree with what was observed previously. For a width of $2.5m$, the uncertainty of a single test reaches more than 30% while it is less than 5% for the others. Therefore, the tests with the narrowest width can not be used for further work as the uncertainties are too high.

The repeatability of the tests is adequate for most all tests except the ones measured for the lowest towing velocity and smallest width. For the other tests, the results are satisfactory.

6.1.7 Errors and limitation

The investigation of errors in physical experiments holds significant importance due to their abundant occurrence. Such errors can be attributed to a variety of factors, including human error, which is inherent in nearly all experiments. Additionally, inaccuracies stemming from measurement instruments present a more challenging task in terms of quantification. Within this subsection, we delve into the sources of error that emerge from the physical experiments conducted in the MC lab.

Measurement errors

Measurements and material errors can arise from various sources. Firstly, the instruments had to undergo calibration before their usage. The force transducers were calibrated by NTNU technicians in the laboratory. Regardless of their knowledge, errors can still occur during the calibration process, leading to incorrect results. To address this, the calibration was double-checked multiple times. On the other hand, calibrating the markers was a more intricate task, as outlined in Section 2.1.3. The procedure involved a minimum of two individuals: one to manipulate the calibration structure and another to ensure continuous capture of the markers. Involving more people could potentially introduce additional human errors. This process required multiple calibration attempts for completion.

Some measurements were made with little precision. The width was measured using a ruler accurate to the millimeter. However, the measurement was made on the carriage, and it was then necessary to try to fix the beams on which the membrane was hung as vertically as

possible. This method could create an offset of up to a few centimeters. The measurement of the distance between the structure and the current probe is likely the least accurate. As no specialized equipment was available for this purpose, it was measured using a ruler during the tests, resulting in a potential variation of $\pm 1cm$.

Instrument accuracy can also contribute to errors. The force transducers may exhibit an accuracy range of 1 to 5% of the measured value, and the current probe can provide values within $\pm 0.01m/s$. Regarding the markers, errors became apparent during data processing. The cameras occasionally jumbled the order of the markers, causing shifts or mirror-image naming. For instance, marker 1 might be labeled as marker 10, marker 2 as marker 9, and so on. This issue had to be identified individually for each test and resolved during data analysis.

Data extraction presents another potential source of error. MATLAB was used to process the data, and human error could occur when loading or analyzing the files, leading to potential misinterpretation of the results. Additionally, an error originating from the Catman files was observed during MATLAB code computation. In certain tests' initial milliseconds, the recorded marker data exhibited implausible values of -1000000, clearly deviating from accurate measurements. Rather than representing genuine data, this anomaly was identified as an error arising from the data measurement software.

Model errors

During the construction of the model, choices had to be made to simplify it, which can lead to errors during experiments.

First, the floating line was made of a fishing rope and 35 floats. The use of multiple floats resulted in a discontinuity in the floating line. As a result, the drag force cannot be evenly distributed over the structure, leading to non-uniform deformations. Moreover, some of the clamps securing the floats broke during the experiments, further exacerbating the unevenness of the floating line. This irregularity can affect the shape of the membrane and the measured forces at the boom's ends.

Next, the weight added to the lower part of the structure could not correspond precisely to the scaling. The choice was limited to the available chains in the NTNU reserves. Adding more weight would have hindered the membrane's planing ability and consequently affected the applied drag force.

Lastly, the fabric used to construct the membrane consists of sail remnants made of polyester. However, the flexibility of the fabric was not selected to match the scaling of the structure created by the Ocean Cleanup. Therefore, it lacked the necessary flexibility.

Additionally, only the floating line was fixed to the carriage, allowing the bottom of the membrane to move freely at its ends, leading to planing. Attempts were made to address this issue by adding a rope between the bottom and the top of the skirt, but the results were inconclusive. In a full-scale model, additional modification to the model would be required to prevent this phenomenon. Consequently, significant variations in the model's shape are expected between experimental observations and the global model.

Environment errors

During experiments, errors can occur due to environmental factors. In this particular case, the boom was fixed to a width of up to $5m$, while the width of the MC lab is $6.45m$. Hence, there was 75 cm between the attachment point of the structure and the wall. This proximity can result into creating wall effects, especially when the towing velocity is at its highest. The turbulence generated at the end of the boom can cause waves to reflect off the wall before reaching the structure. This phenomenon can impact the shape of the boom and the applied forces. Although the effect is less pronounced when the width is reduced, it remains present. To address this issue, it would have been necessary to test the model in a wider tank.

The results were influenced by the length of the tank as well. When the towing velocity was set to its maximum while the acceleration remained low, the structure did not have sufficient time to attain a steady state before reaching the end of the tank. Alternatively, if it did manage to reach a steady state, the duration was too short. Consequently, the collected data cannot be used to draw any meaningful conclusions regarding the behavior of the boom.

Limitation

As explained in Section 5.1.7, some results could not be measured and were therefore only observed visually. On one hand, the deformation of the membrane underwater was not captured. The first objective was to mount markers underwater and capture their position to study the membrane deformation. However, due to the membrane size and the available material in the MC lab, it was not possible to carry out this task. Indeed, the underwater cameras have a fixed position that did not allow them to see the membrane as it is wider than the models usually tested with these cameras.

On the other hand, it was not possible to add plastic pellets or something else to try the ability of the boom to capture plastic or only visualize the flow. This would have required more time and people to clean the tank before it is used by someone else. Therefore, the only comments that can be made on it come solely from visual observations.

6.2 Numerical results

6.2.1 Verification of the numerical model

6.2.1.1 Mesh convergence

Figure 5.2.1 shows how the four meshes are converging. The coarsest mesh, Mesh 1, has the highest fluctuations. Therefore, it would not be a good choice to pick Mesh 1 to run the simulations. Moreover, in Figure 5.2.2, one can see that the value of the mean drag coefficient is higher for Mesh 1 than for the others.

The three other meshes seem to converge toward the same value in Figure 5.2.1. All of them show fluctuation throughout the simulation. However, the mean drag coefficients are slightly different for the different meshes. The calculated value increases when the number of cells increases. Therefore, picking Mesh 4 to run the simulation would be the safest solution. Nevertheless, simulations with Mesh 4 take almost twice as long as with Mesh 3. Therefore, Mesh 3 was chosen to run the simulations.

6.2.1.2 Flat plate

To verify the model, a simulation was performed with a flat plate. Using the water viscosity, $\nu = 10^{-6} m^2/s$ leads to a high Reynolds number $Re = 39000$. However, studies of a normal flat plate with high Reynolds are not common. Therefore, a simulation with a low Reynolds number was performed. As explained in Section 5.2.1.2, the viscosity was chosen so the Reynolds number is 100. In Figure 5.2.3, the drag coefficient of the flat plate is plotted as a function of dimensionless time. With low Reynolds, the drag coefficient converges, to a value of 2.09. Comparisons can be made with other studies of a normal flat plate. In his article published in 2009, (Narasimhamurthy and Andersson 2009), Narasimhamurthy studies the turbulent wake behind a normal flat plate. The work includes a calculation of the drag coefficient for a Reynolds number of 750. The numerical study provides a drag coefficient $C_D = 2.31$. This article also makes reference to other studies of a normal flat plat, (Najjar and Vanka 1995) and (Saha 2007), where drag coefficients are calculated for other Reynolds numbers. In the first one, the drag coefficient is found to be around 2 for a Reynolds of 200 while in the second one, it is found to be 2.6 for a Reynolds number of 100. Therefore, finding a drag coefficient of 2.09 seems

correct. Moreover, in the studies cited, the plate is in the middle of the domain, whereas in the simulation presented here, it is at the top, so the boundary conditions are not the same.

The second test presented in Figure 5.2.3 is a test with the same flat plate but with the water viscosity. One can see on the plot that the drag coefficient is not as stable as with a low Reynolds number. This is due to higher turbulence. As explained previously, studies under these conditions are not common. Nevertheless, Narasimhamurthy (Narasimhamurthy and Andersson 2009) refers to another study for higher Reynolds numbers: (Fage and Johansen 1927). In this paper, the drag coefficient is found to be 2.13 for a Reynolds number of $1.5 \cdot 10^5$. The drag coefficient seems to decrease when the Reynolds number increases, which is also the case in the present work. However, the mean drag coefficient is found to be 1.35 which is significantly lower than in Fage study (Fage and Johansen 1927). Yet, in another numerical study realized by Koumoutsakos and Shields (Koumoutsakos and Shiels 1996), the drag coefficients calculated on a normal flat plate are lower than in the previously mentioned articles. Therefore, having a drag coefficient of 1.35 for a high Reynolds simulation seems reasonable. The numerical model can thus be used for this study.

6.2.2 Contour plots

Velocity contours displayed in Figure 5.2.4 show that the size of the swirl behind the membrane is affected by the inclination angle. Indeed, the length and the depth of the swirl decrease when the angle of rotation increases. For 0°, 15°, and 15°, the difference is not very significant and therefore not very visible but for 45° and 60° the size of the swirl is clearly smaller. A more in-depth study of the impact of the angle of inclination on the length of the vortex will be carried out in Section 6.2.4. Moreover, the maximum value of the velocity magnitude is approximately the same for the three smallest angles ($U \approx 0.33 m/s$) while it is reduced for the two largest angles: $U \approx 0.29 m/s$ for $\theta = 45°$ and $U \approx 0.21 m/s$ for $\theta = 45°$.

The velocity contours are also affected by the shape of the curve, see Figure 5.2.5. Even though the maximum value of the velocity magnitude is very close for all three shapes, the wake is affected. It seems that the one created by curve 1 is shorter than the one created by curve 2, which is itself shorter than the one created by curve 3.

A similar analysis can be made for the pressure contour plots. In Figure 5.2.6, one can see how the swirl is affected by the rotation angle. This time, the difference between a 15° rotation angle and a 30° rotation angle is more apparent. In addition, it is easier to see the difference between the values taken by the pressure in the various simulations, especially for the 0° rotation angle, where the maximum and minimum values are higher than for any other simulations. On the other hand, for the simulations with different curves, it is difficult to see any difference between the wakes of the three setups.

6.2.3 Streamlines

6.2.3.1 Large

As with the contour plots, the streamlines observed over the whole domain are intended to determine visually the influence of the shape of the curve and its inclination on the vortex produced behind it. One can now clearly see in Figure 5.2.8 that the inclination angle affects the swirl. Indeed, it becomes smaller in length and depth when the angle increases. Moreover, without any inclination, the streamlines seem less "regular" than for the other inclination angles. The shape of the lines moves away from the ellipses which can be seen in the other cases.

6.2.3.2 Zoom

Streamlines can be seen over a smaller area in Figure 5.2.9 for the three different curves. As explained in Section 5.2.3, as the membrane was too thin to be correctly observed on Paraview, a yellow line has been added above in order to be able to correctly visualize the results.

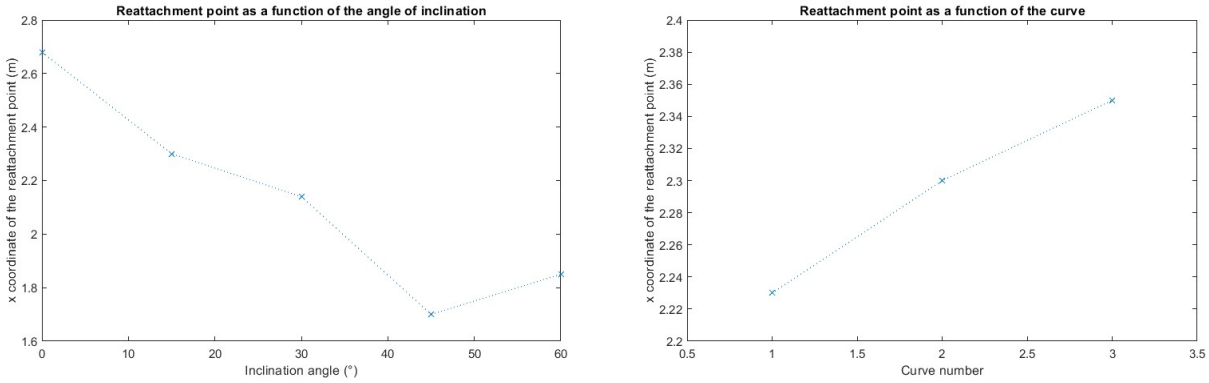
For all three situations, one or more vortices can be observed in the corner right after the membrane. This is due to a high Reynolds number. In their study about cavity flow at high Reynolds number (Nallasamy and Prasad 1977), Nallasamy and Krishna Prasad show that eddies are created in the cavity corner when the Reynolds number increases. The corner behind the membrane can be compared to the bottom corner of the cavity as there is a uniform flow in the $x - direction$ and a large eddy. Therefore, it was expected to obtain these smaller swirls. Furthermore, a small eddy is also created in front of the curve.

The eddies are quite different between the three curves. For curve 1, one can see one eddy in front and one behind but for the two other curves, multiple eddies appear. The shape of the curves seems to affect the turbulence close to the membrane. The more the membrane is curved, the less the turbulence is regular. Moreover, a small vortex was sometimes created for curve 3 at the bottom of the membrane, at the point of detachment.

6.2.4 Velocity profiles

The goal of studying the velocity profile along the top of the domain is to find the reattachment point. The velocity profiles are plotted after $70s$ of simulation. As one can see in Figure 5.2.10 and Figure 5.2.11, the velocity in the $x - direction$ drops after the membrane, where the eddy appears. It then goes up again. The reattachment point can be found where the velocity in the $x - direction$ crosses the 0 axis. Figure 6.2.1 displays the x coordinate of the reattachment point for different inclination angles and different curves. As observed on the velocity and pressure contours plots, the reattachment point decreases when the inclination angle increases, except for the 60° inclination angle. With only 4 different angles, it is difficult to conclude a trend that this value could follow.

In figure 6.2.1b, one can observe the length of the reattachment point being the highest for curve 3 and the smallest for curve 1, as expected. The variations are less significant for the curves than for the inclination angles. However, this still affects the shape of the swirl and proves that the shape of the curve influences the hydrodynamic forces applied to the structure.



(a) Varying inclination angle with curve 2

(b) Varying curve with inclination angle $\theta = 15°$

Figure 6.2.1: x coordinate of the reattachment point

6.2.5 Drag coefficient

Figure 5.2.12 presents the drag coefficient throughout time. One can see that no matter the setup conditions, curve and inclination angle, the drag coefficient converges toward a value.

However, using the time plots makes it difficult to compare the results between them due to the small variations present. Therefore, the mean drag coefficient is studied.

Figure 5.2.13 displays the value of the mean drag coefficient for all the simulations performed. One can see that, for every membrane shape, its value decreases when the inclination angle increases. This observation was expected as it is a conclusion drawn by Emma Løyland Vassanyi 2021.

In addition, in Figure 5.2.14, the drag coefficient for an inclination angle is plotted as a function of the distance between the horizontal axis and the center of the membrane. These distances, corresponding to the curves, are given in Table 3.3.1. The mean drag coefficients increase when the distance increases. To better observe the influence of the shape of the membrane on the coefficient of drag, the coefficient increase, compared to the flat plate, is presented in Table 6.2.1. These results, especially when $D = 50mm$, prove that the shape chosen has an important influence on the drag coefficient and thus on the forces applied to the structure. Therefore, representing the membrane as a plate when in reality it is curved can lead to false results and therefore false conclusions.

Distance D [mm]	**Drag coefficient increase [%]**
10	6.05
25	9.68
50	29.1

Table 6.2.1: Drag coefficient increase compared to the flat plate

6.2.6 Error

The field of numerical methods and CFD software constantly advances, leading to enhanced performance over time. Nevertheless, it is essential to acknowledge that numerical solutions inherently entail errors. Examining these potential errors becomes crucial as it helps determine the reliability and accuracy of the results and models obtained. This section focuses on elucidating the various sources of errors that arise during CFD simulations carried out using OpenFOAM.

First, assumptions were made in the numerical model. The flow is considered to be laminar, thus, no turbulence model was built. Turbulence models are useful and necessary when studying boundary-layer flows, especially when the separation is investigated. In this study, the separation point is known to be at the bottom of the membrane. Therefore, the flow was chosen to be studied as laminar. However, choosing a turbulence model would surely have affected the results on the shape of the vortices. The flow is turbulent at the edges of the large eddy and changing the numerical model would have affected the results. The attachment points could have differed from what has been found here. Nevertheless, using a laminar model seems to be sufficient for the conclusions drawn in this thesis.

The assumptions of incompressibility and isothermality in fluid dynamics introduce uncertainties since the density and kinematic viscosity of water are not truly constant in reality. These properties vary depending on the temperature of the water. Furthermore, the assigned values for the kinematic viscosity and density of water are approximations based on their values at a reference temperature of 20°C (Anton Paar 2022).

The last major assumption made in the numerical model is the top boundary conditions. In the model, it is set to *symmetryPlane* to represent the surface. The *symmetryPlane* boundary condition is designed to simulate a slip condition at the boundary, effectively reflecting the flow field pattern along the surface of the edge. However, to model the free surface correctly, a multiphase solver should have been used. During the experiment, one could observe a displacement of the water just in front of the membrane. The towing caused the water to rise. In the numerical model, the water level was assumed to be the same everywhere. Therefore, some simplifications

were made. This choice of modeling of the free surface may have affected the results in terms of force applied to the membrane and drag coefficient.

Errors and uncertainties can arise from the software itself and are beyond the control of the user. These errors can include round-off errors, which occur due to the limited number of digits that can be represented by the program. However, it is often unnecessary to have an infinite number of digits in a number when the error resulting from a finite number of digits is negligible. This is why the convergence of the simulation model is tested, to determine if the results are sufficiently close to the exact values for them to be practically useful. Therefore, there will always be some degree of error in the results obtained from numerical simulations. This error, however, is not solely attributed to the limited number of digits but also to the discretization of the exact equations.

Lastly, errors can be human errors. When writing down codes to read the results or results themselves, mistakes could have been made leading to incorrect results.

6.3 Global model

Figure 5.3.1 displays the expected shape of the boom when using the global model presented in Section 4. When a single coefficient is used 5.3.1a along the entire length of the membrane, the shape of the boom is similar to a parabola. When several coefficients are used 5.3.1b 5.3.1c, the shape moves away from a parabola. When using the numerical drag coefficients, the boom extends over a longer width than when using Emma Løyland Vassanyi's coefficients (Emma Løyland Vassanyi 2021). This is because the numerically calculated coefficients are lower than those found by Emma Løyland Vassanyi in her experiments. In Figure 6.3.1, the two models using strip theory are displayed along with the shape captured during one test. The test conditions were the ones used as input in the global models: velocity $U = 0.13m/s$, tension force $T = 12N$, draft $d = 0.3m$, water density $\rho = 1000kg/m^3$, and length $L = 5.38m$. One can see that when using the drag coefficients found experimentally by Emma Løyland Vassanyi 2021, the expected shape is closer to the experimental one than when using the numerical drag coefficient. One of Emma Løyland Vassanyi's study conclusions was that her numerical drag coefficient would underestimate the experimental one. Therefore, one can imagine that the same is happening here and that the drag coefficients are underestimated by the numerical model.

Other assumptions made to build the global model can increase the difference with the experimental result. These assumptions are listed below:

Bending stiffness neglected:

In the global model created, the bending stiffness effects were ignored. However, the bending stiffness of a cable hanging freely and subjected to gravity determines its shape and behavior, therefore, the same applies to this study. Bending stiffness determines the amount of sag and deflection in the boom. Higher bending stiffness results in less sag and deflection, while lower bending stiffness leads to more pronounced sag. The boom's shape stability is influenced by its bending stiffness. Higher bending stiffness helps maintain the boom's shape and resist deformation caused by external forces. Tension distribution along the boom is influenced by its bending stiffness. Higher bending stiffness promotes more uniform tension distribution, reducing localized stress concentrations.

Static modeling:

The global model only covers a static study of the structure. The inertia effects are not taken

into account whereas they can be important when a floating structure is towed. First, when the towing force is applied to the floating structure, it experiences acceleration or deceleration. Inertia causes the structure to resist these changes, leading to a lag in its response. As a result, the structure may take some time to adjust its position and align with the towing direction. Then, inertia can cause the floating structure to oscillate or swing while being towed. When the towing force changes direction or magnitude, the inertia of the structure may cause it to overshoot or undershoot its desired position, leading to oscillations or swinging motion. Lastly, inertia effects can introduce dynamic loads on the floating structure during towing. Rapid changes in the towing speed or direction can create additional forces and moments on the structure, potentially causing stress concentrations or structural vibrations.

Study in the horizontal plane:

The global model only studies the structure in the horizontal plane. The effects coming from the vertical dimension are neglected. However, as the numerical results showed it 6.2.5, the shape and inclination angle of the membrane affect the drag coefficient and the drag force. Therefore, it is important to take into account this last dimension in a global model. This has been done to a lesser extent using coefficients dependent on the curve and the angle of inclination. Nevertheless, the choice of coefficients was based solely on visual observation of the experiments. The actual shape of the membrane under water is unknown and remains to be studied.

Despite the many simplifications that have been made to build this global model, it remains fairly faithful to real-life behavior. Indeed, as the geometry of the structure is quite simple, the global model can be applied and provides correct results.

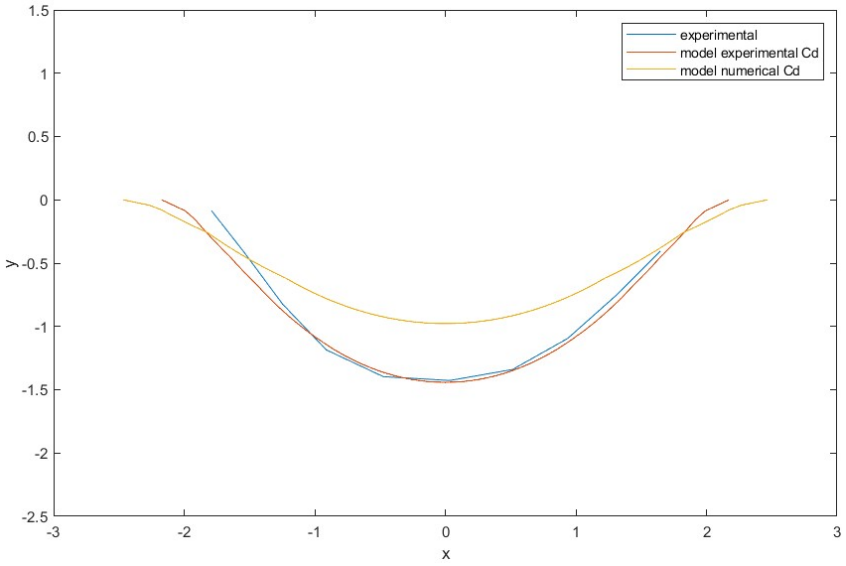


Figure 6.3.1: Comparison between the models and the experiment

6.4 Further work

Improvement on the numerical study:

The CFD simulations that were made in this study simplified the membrane into a solid of different shapes. This model can be improved by taking into account the flexibility of the membrane, the weight added at its bottom, and the fact that it is only fixed at the top. To do so, the coupling will have to be introduced. This can be done thanks to dynamic meshing or with the immersed boundary method. The immersed boundary method is a numerical technique used

to simulate fluid-structure interactions. It allows for the modeling of complex and deformable structures (such as flexible boundaries or immersed objects) within a fluid domain. By using a combination of interpolation and force coupling, the method allows for accurate simulation of the fluid flow around and interaction with the immersed objects, without the need for complex grid generation or boundary-fitting procedures. This method was first considered during this project, but faced with a number of difficulties, the idea was abandoned. However, it remains a method that allows a section of the membrane to be studied correctly without wasting too much calculation time. An example of the use of this method for a flexible structure is the study made by Huang and Sung (Huang and Sung 2009).

As explained in Section 6.2.6, the free surface is also simplified in the current numerical model. To properly take it into account, a multiphase solver has to be used. A basic multiphase solver used in open foam is *interFoam* (OpenFOAM 2022c). *interFoam* solves the Navier Stokes equations for two incompressible, isothermal immiscible fluids. Other free surface solvers have been created. For example, Paulin Ferro (Ferro et al. 2022) developed a free surface flow solver using the Ghost Fluid Method on OpenFOAM. It is thus possible to find an efficient way to solve the free surface issue in the current situation.

Improvement of the global model:

Section 6.3 presents the drawbacks of the global model. Even though the present model can be considered quite accurate, it can always be improved. One of the best ways of improving it would be to model the membrane under water, i.e. to take into account the vertical dimension of the boom. To do so, an accurate method can be to use a truss model. A truss model is a simplified representation of a structure that consists of interconnected bars or members. It is used to analyze and design load-bearing structures such as bridges and roofs. The truss model assumes the members are slender and capable of supporting axial forces only, with connections that allow rotation but no bending at the joints. This simplification enables efficient analysis and calculation of internal forces in the truss members based on the principles of static equilibrium. This kind of model has been used in the past to represent fishing nets, (Kristiansen and O. M. Faltinsen 2012; Shi 2019). As the membrane presented is flexible like a fishing net, it seems that modeling it using a truss model is a feasible solution.

CHAPTER
SEVEN

CONCLUSION

The hydrodynamics of a flexible containment boom to collect plastic pellets were investigated in this thesis. A global model was developed based on cable structures theory to assess the deformations experienced by a boom structure when subjected to a current. This model specifically focuses on the horizontal plane and employs the catenary equation, drag force, and strip theory to determine the optimal shape for the structure. To check the veracity of the model, experiments were conducted on a $5m$ long flexible boom with a $30cm$ draft. They were performed in the MC lab at NTNU. The model was towed in a $40m$ long, $6.5m$ wide, and $1.5m$ deep tank. The towing velocity varied between $0.08m/s$ and $0.53m/s$ and the attachment width was either $2.5m$, $4m$, or $5m$. A total of 167 tests were realized. The shape of the boom, the towing forces and the current were measured during the experiments. Furthermore, a numerical investigation was carried out utilizing the CFD solver OpenFOAM. The simulations were conducted based on the experimental setup's input values and involved the use of different geometries to represent the skirt of the boom. The flexibility of the membrane has not been simulated numerically. To fill this gap, four different geometries were tested: a plate and three parabolic curves. The object was oriented towards a current of $0.13m/s$, and various inclination angles were considered.

The numerical study proved that the shape of the membrane influences the forces applied to it and the flow. Indeed, the shape of the membrane influences the shape of the wake behind it. More specifically, the reattachment point was investigated. It was found that the reattachment point is closer to the membrane as the angle of inclination increases. For the different geometries, the more curved the curve, the further away the reattachment point is. The shape of the object also influences the drag coefficient. It decreases when the inclination angle decreases and it increases with the bending of the curve. This demonstrates that the forces applying to the boom are influenced by the shape taken by the membrane underwater and thus the fabric used to build it.

The experiments showed that the forces varied with the towing velocity and the attachment width, as expected. The current measured behind the structures proved that the system was creating turbulence. In addition, the shape of the boom was captured thanks to markers and cameras. The shape taken by the boom was a parabola as expected with some differences depending on the towing velocity. However, some issues occurred during the tests. For example, when the width was the narrowest, the boom started to oscillate.

Finally, the global model allowed us to observe the expected shape of the boom. To run this model, a drag coefficient was required as input. Three different setups were displayed. First, only one drag coefficient across the whole boom was used. Then, strip theory was introduced and different values of drag coefficients were attributed to the sections. The coefficients used

originated from (Emma Løyland Vassanyi 2021) or from the numerical study. These models were compared to each other and to the experimental study. The comparison showed that even though the model is a simplified one, it represents quite accurately the measured shape of the boom. It can however be improved by using, for example, a truss model to portray the flexible membrane.

REFERENCES

4Ocean (2022). URL: https://www.4ocean.com/ (visited on 09/15/2022).

Amini et al. (2017). "Fluid-structure interaction analysis applied to oil containment booms". In: *International oil spill conference*.

Anton Paar (2022). *Viscosity of Water*. URL: https://wiki.anton-paar.com/en/water/ (visited on 09/15/2022).

As you sow (n.d.). *Plastic pellet pollution*. URL: https://www.asyousow.org/our-work/circular-economy/plastic-pellets.

Autodesk (2023). *Fusion 360*. URL: https://www.autodesk.com/products/fusion-360/ (visited on 02/15/2023).

Baker, Scott et al. (2019). "Research and Development of Oil Containment Boom Designs". In.

Bardestani, Mohsen (2017). "A two dimensional numerical and experimental study of a floater with net and sinker tube in waves and current". PhD thesis. NTNU.

Brambini, Roberto et al. (2017a). "Hydrodynamics and Capture Efficiency of Plastic Cleanup Booms: Part I — Experiments and Dynamic Analysis". In.

— (2017b). "Hydrodynamics and Capture Efficiency of Plastic Cleanup Booms: Part II — 2D Vertical Capture Efficiency and CFD Validation". In.

Calonnec, Capucine (2022). *Hydrodynamics of flexible containment boom*. NTNU.

Emma Løyland Vassanyi (2021). *Experimental and Numerical study of current effects on a plastic containment boom*. NTNU.

European commission (2022). URL: https://environment.ec.europa.eu/topics/plastics/single-use-plastics_en (visited on 09/15/2022).

Fage and Johansen (1927). "On the flow of air behind an inclined flat plate of infinite span". In: *Proceedings of the Royal Society of London. Series A*, pp. 170–197.

Faltinsen, O.M. (1990). *Sea loads on ships and offshore structures*. Cambrige University Press.

Feng, Y. Zhang, and Z. Wu (2018). "Three Dimensional Numerical Simulation of Oil Containment Process by Flexible Oil Booms in Inland Waters". In: *Science and Engineering*, pp. 463–468.

Ferro, Paulin et al. (2022). "Development of a free surface flow solver using the Ghost Fluid Method on OpenFOAM". In: *Ocean Engineering* 253.

Fingas, Merv and Jennifer Charles (2001). *The Basics of Oil Spill Cleanup, second edition*. Lewis Publisher.

Geyer, R. (2020). *Production, Chapter 2 - Production, Use and Fat of Synthetic Polymers in Plastic Waste and Recycling*.

Huang, Wei-Xi and Hyung Jin Sung (2009). "An immersed boundary method for fluid–flexible structure interaction". In: *Computer Methods in Applied Mechanics and Engineering* 198, pp. 33–36.

Irvine, M. (1990). *Cable structures*. The MIT Press.

Koumoutsakos, P. and D. Shiels (1996). "Simulations of the viscous flow normal to an impulsively started and uniformly accelerated flat plate". In: *Journaml of fluid mechanics* 328, pp. 177–227.

Kristiansen, Trygve and Odd M. Faltinsen (2012). "Modelling of current loads on aquaculture net cages". In: *Journal of Fluids and Structures* 34, pp. 218–235.

Markic, Ana et al. (2020). "Plastic ingestion by marine fish in the wild". In: *Critical Reviews in Environmental Science and Technology* 50:7, pp. 657–697.

Mathworks (2022). *Matlab*. URL: https://se.mathworks.com/products/matlab.html (visited on 09/25/2022).

Microsoft (2023). *Microsoft Excel*. URL: https://www.microsoft.com/en-us/microsoft-365/excel (visited on 01/10/2023).

Muttin, Frederic (2015). "Mathematical Modelling and Numerical Simulation of Oil Pollution Problems - Chapter 7". In.

Najjar and Vanka (1995). "Effects of intrinsic three-dimensionality on the drag characteristics of a normal flat plate". In: *International Journal of Heat and Fluid Flow* 7, pp. 2516–2518.

Nallasamy, M. and K. Prasad (1977). "On cavity flow at high Reynolds numbers". In: *Journal of Fluid Mechanics* 79, pp. 391–414.

Narasimhamurthy and Andersson (2009). "Numerical simulation of the turbulent wake behind a normal flat plate". In: *International Journal of Heat and Fluid Flow* 30, pp. 1037–1043.

Nordic Coastal Clean Up (2022). URL: https://nordiccoastalcleanup.com/results (visited on 10/10/2022).

OpenFOAM (2022a). *Boundary Conditions*. URL: https://www.openfoam.com/documentation/user-guide/5-models-and-physical-properties/5.1-boundary-conditions (visited on 09/15/2022).

— (2022b). *File structure of OpenFOAM case*. URL: https://www.openfoam.com/documentation/user-guide/2-openfoam-cases/2.1-file-structure-of-openfoam-cases#x5-40002.1 (visited on 09/15/2022).

— (2022c). *interFoam*. URL: https://openfoamwiki.net/index.php/InterFoam (visited on 09/15/2022).

— (2022d). *Mesh generation with the blockMesh utility*. URL: https://www.openfoam.com/documentation/user-guide/4-mesh-generation-and-conversion/4.3-mesh-generation-with-the-blockmesh-utility (visited on 09/15/2022).

— (2022e). *Mesh generation with the snappyHexMesh utility*. URL: https://www.openfoam.com/documentation/user-guide/4-mesh-generation-and-conversion/4.4-mesh-generation-with-the-snappyhexmesh-utility (visited on 09/15/2022).

— (2022f). *Numerical schemes*. URL: https://www.openfoam.com/documentation/user-guide/6-solving/6.2-numerical-schemes (visited on 09/15/2022).

— (2022g). *Time and data input/output control*. URL: https://www.openfoam.com/documentation/user-guide/6-solving/6.1-time-and-data-inputoutput-control (visited on 09/15/2022).

OpenFOAM wiki (2022). *PISO*. URL: https://openfoamwiki.net/index.php/PISO (visited on 09/15/2022).

Qualysis (2021). *Qualysis Oqus*. URL: https://www.qualisys.com/product-tag/oqus/ (visited on 11/15/2021).

Saha (2007). "Far-wake characteristics of two-dimensional flow past a normal flat plate". In: *Physics of Fluids* 19.

Shi, Luhao (2019). *Dynamic Analysis of Semisubmersible Offshore Fish Farm Operated in China East Sea*. NTNU.

Sigma2 (2023). *Sigma 2*. URL: https://www.sigma2.no/ (visited on 03/15/2023).

The Ocean Cleanup (2022). URL: https://theoceancleanup.com/ (visited on 09/15/2022).

Ultratech (2022). *All about oil containment boom*. URL: https://spillcontainment.com/ultratech-university/spill-response-decon/all-about-oil-spill-containment-booms/ (visited on 09/15/2022).

United Nations (2021). "X-Press Pearl Maritime Disaster Sri Lanka". In.

— (n.d.). *UN sustainable development goals*. URL: https://www.un.org/sustainabledevelopment/.

United Nations Environment Programme (2021). "Drowning in Plastics – Marine Litter and Plastic Waste Vital Graphics". In: *Journal of Computational Physics* 48, pp. 387–411.

APPENDICES

CONTENTS

- **A** - OPENFOAM FILES
- **B** - GLOBAL MODEL MATLAB CODES
- **C** - EXPERIMENTAL RESULTS ANALYSIS MATLAB CODES
- **D** - DRAG COEFFICIENT PLOT MATLAB CODE
- **E** - MESHES

A1 - blockMeshDict.txt

```
    /*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

convertToMeters 1;

vertices
(
    (-2   -1.5 -0.04)
    ( 4   -1.5 -0.04)
    ( 4    0 -0.04)
    (-2    0 -0.04)
    (-2   -1.5  0.04)
    ( 4   -1.5  0.04)
    ( 4   0  0.04)
    (-2    0  0.04)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (120 30 1) simpleGrading (1 1 1)
);

edges
(
```

```
);

boundary
(
    in
        {        type patch;
                faces
                (        (0 4 7 3)   );
        }

    out
        {        type patch;
                faces
                (        (2 6 5 1)   );
        }

    top
        {        type symmetryPlane;
                faces
                (        (3 7 6 2)        );
        }

    bottom
        {        type patch;
                faces
                (        (1 5 4 0)        );
        }

    frontAndBack
        {        type empty;
                faces
                (        (0 3 2 1)
                        (4 5 6 7)        );
        }
);

mergePatchPairs
(
);

// ************************************************************************* //
```

A2 - snappyHexMeshDict.txt

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | foam-extend: Open Source CFD                    |
|  \\    /   O peration     | Version:     4.1                                |
|   \\  /    A nd           | Web:         http://www.foam-extend.org         |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      snappyHexMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// Which of the steps to run
castellatedMesh true;
snap            true;
addLayers       false;

// Geometry. Definition of all surfaces. All surfaces are of class
// searchableSurface.

geometry
{

    membrane.stl
    {
        type triSurfaceMesh;
                name membrane;
    }

};

// Settings for the castellatedMesh generation.

castellatedMeshControls
{

    // Refinement parameters
    // ~~~~~~~~~~~~~~~~~~~~~
    maxLocalCells 100000;
    maxGlobalCells 2000000;
    minRefinementCells 0;
    //maxLoadUnbalance 0.10;
    nCellsBetweenLevels 4;
```

```
    // Explicit feature edge refinement
    // ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

        features
    (
    );

    // Surface based refinement
    // ~~~~~~~~~~~~~~~~~~~~~~~~


    refinementSurfaces
    {
        membrane
    {
        // Surface-wise min and max refinement level
        level (2 3);
        // Grading
        distanceGrading 0.8;
    }
    }

    resolveFeatureAngle 30;

    // Region-wise refinement
    // ~~~~~~~~~~~~~~~~~~~~~~


    refinementRegions
    {
    }

    // Mesh selection
    // ~~~~~~~~~~~~~~


    locationInMesh (-1 0 0);

    allowFreeStandingZoneFaces true;
}

// Settings for the snapping.
snapControls
{
    nSmoothPatch 3;

    tolerance 4.0;

    nSolveIter 30;

    nRelaxIter 5;
```

```
}

// Settings for the layer addition.
addLayersControls
{
    relativeSizes true;

    layers
    {
        "membrane_.*"
        {
            nSurfaceLayers 1;

        }
    }

    expansionRatio 1.0;

    finalLayerThickness 0.3;

    minThickness 0.25;

    nGrow 1;

    // Advanced settings

    featureAngle 60;

    nRelaxIter 5;

    nSmoothSurfaceNormals 1;

    nSmoothNormals 3;

    nSmoothThickness 10;

    maxFaceThicknessRatio 0.5;

    maxThicknessToMedialRatio 0.3;

    minMedianAxisAngle 130;

    nBufferCellsNoExtrude 0;

    nLayerIter 50;

    nRelaxedIter 20;
}

// Generic mesh quality settings. At any undoable phase these determine
// where to undo.
```

```
meshQualityControls
{
    maxNonOrtho 65;

    maxBoundarySkewness 20;
    maxInternalSkewness 4;

    maxConcave 80;

    minFlatness 0.5;

    minVol 1e-13;

    minArea -1;

    minTwist 0.05;

    minDeterminant 0.001;

    minFaceWeight 0.05;

    minVolRatio 0.01;

    minTriangleTwist -1;

    // Advanced

    nSmoothScale 4;
    //- amount to scale back displacement at error points
    errorReduction 0.75;

    relaxed
    {
        maxNonOrtho 75;
    }
}

mergeTolerance 1E-6;


// ************************************************************************* //
```

A3 - U.txt

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0.13 0 0);

boundaryField
{
    in
    {   type            fixedValue;
        value           uniform (0.13 0 0);    }

    top
    {   type            symmetryPlane;    }

    out
    {   type            zeroGradient;    }

    bottom
    {   type                    slip;    }

    membrane
    {   type                    fixedValue;
        value           uniform (0 0 0);    }

    frontAndBack
    {   type            empty;    }
}

// ************************************************************************* //
```

A4 - p.txt

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    in
    {   type            zeroGradient;
        value           uniform 0;    }

    top
    {   type            symmetryPlane;    }

    out
    {   type            fixedValue;
        value           uniform 0;    }

    bottom
    {   type            slip;    }

    membrane
    {   type            zeroGradient;    }

    frontAndBack
    {   type            empty;    }
}

// ************************************************************************* //
```

A5 - transportProperties.txt

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      transportProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

transportModel         Newtonian;

nu              nu [0 2 -1 0 0 0 0] 1e-6; // viscosity of water


// ************************************************************************* //
```

A6 - turbulenceProperties.txt

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      turbulenceProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

simulationType laminar;

// ************************************************************************* //
```

A7 - controlDict.txt

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     pisoFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         100;

deltaT          0.0001;

writeControl    runTime;

writeInterval   0.1;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable false;
```

```
// function calculating forces and coefficients on the model
functions
{
        forceCoeffs
        {
                type                forceCoeffs; // folder
                libs                ("libforces.so");
                writeControl        runTime;
                writeInterval        0.1;
                patches         ( membrane_ASCII  );
                pName                P;
                UName                U;
                rho                rhoInf;
                rhoInf                1000;
                magUInf         0.13; // reference velocity
                log                true;
                liftDir         (0 1 0);
                dragDir          (1 0 0);
                CofR                (0 0 0); // centre of rotation
                pitchAxis         (0 0 1);
                lRef                0.3;
                Aref                0.024; // projected area
        }

        forces
        {
                type                forces;
                libs                ("libforces.so");
                patches         ( membrane_ASCII  );
                writeFields        yes;
                writeControl        runTime;
                writeInterval        0.1;
                rho                rhoInf;
                rhoInf                1000;
                U                 U;
                p                 p;
                log                true;
                CofR                (0 0 0);
                liftDir         (0 1 0);
                dragDir          (1 0 0);
                Aref                0.024;
        }


}


// ************************************************************************* //
```

A8 - fvSchemes.txt

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
ddtSchemes
{
    default         Euler;
}
gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
}
divSchemes
{
    default         none;
    div(phi,U)      Gauss linear;
    div((nuEff*dev2(T(grad(U))))) Gauss linear ;
}
laplacianSchemes
{
    default         Gauss linear orthogonal;
}
interpolationSchemes
{
    default         linear;
}
snGradSchemes
{
    default         corrected;
}

// ************************************************************************* //
```

A9 - fvSolution.txt

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2012                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver          GAMG;
        tolerance       1e-06;
        relTol          0.1;
        minIter 1;
        smoother        GaussSeidel;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }

    pFinal
    {
        $p;
        tolerance        1e-06;
        relTol          0;
    }

    "(U|k|epsilon|omega|R|nuTilda)"
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-05;
        relTol          0;
    }
}

PISO
```

```
{
    nCorrectors     2;
    nNonOrthogonalCorrectors 0;
    pRefCell        0;
    pRefValue       0;
}


// ************************************************************************* //
```

B - GLOBAL MODEL MATLAB CODES

B1 - expected_shape.m

```
rho=1000; %water dentsity [kg/m^3]
d=0.3; %draft length [m]
U=0.13; %towing velocity [m/s]
T0=12.10; %towing force [N]
L_demi = 2.69; %length of half the boom
N = 2690; %number of elements
delta_s = L_demi/N; %longitudinal element size
s_tot = delta_s*(0:N);
xe=0;
ye=0;


% coefficient found by Emma during her master thesis, which have to be
% modified to use thos found numerically
MCd = [3.3 3.2 2.8 2.4 1.8]


% the decomposition of the section is the one used for Emma's drag
% coefficients and is different for the numerical coefficients
for i=1:101 %section of the structure only made of the floating rope, without membrane
    s = s_tot(i);
    Cd =  MCd(5)*0.5;
    alpha=-0.5*rho*Cd*d*U^2/T0;
    Phi0 = atan(1/(alpha*(s_tot(102)-L_demi)));
    ang =s/T0 + Phi0;
    delta_x = delta_s*sin(ang);
    delta_y = delta_s*cos(ang);
    next_x = xe(i)+delta_x;
    next_y = ye(i)+delta_y;
    xe = [xe next_x];
    ye = [ye next_y];
end

for i=102:151
    s = s_tot(i);
    Cd =  MCd(5)*0.5; %drag coefficent with reduction factor
    alpha=-0.5*rho*Cd*d*U^2/T0;
```

```
    [next_x,next_y,ang] = next_element(xe(i),ye(i),s,delta_s,alpha,L_demi);
    xe = [xe next_x];
    ye = [ye next_y];
end

for i=152:201
    s = s_tot(i);
    Cd =  MCd(5)*0.8; %drag coefficent with reduction factor
    alpha=-0.5*rho*Cd*d*U^2/T0;
    [next_x,next_y,ang] = next_element(xe(i),ye(i),s,delta_s,alpha,L_demi);
    xe = [xe next_x];
    ye = [ye next_y];
end

for i=202:301
    s = s_tot(i);
    Cd =  MCd(5); %drag coefficient
    alpha=-0.5*rho*Cd*d*U^2/T0;
    [next_x,next_y,ang] = next_element(xe(i),ye(i),s,delta_s,alpha,L_demi);
    xe = [xe next_x];
    ye = [ye next_y];
end

for i=302:701
    s = s_tot(i);
    Cd =  MCd(4);
    alpha=-0.5*rho*Cd*d*U^2/T0;
    [next_x,next_y,ang] = next_element(xe(i),ye(i),s,delta_s,alpha,L_demi);
    xe = [xe next_x];
    ye = [ye next_y];
end

for i=702:1001
    s = s_tot(i);
    Cd =  MCd(3);
    alpha=-0.5*rho*Cd*d*U^2/T0;
    [next_x,next_y,ang] = next_element(xe(i),ye(i),s,delta_s,alpha,L_demi);
    xe = [xe next_x];
    ye = [ye next_y];
end

for i=1002:1601
    s = s_tot(i);
    Cd =  MCd(2);
    alpha=-0.5*rho*Cd*d*U^2/T0;
    [next_x,next_y,ang] = next_element(xe(i),ye(i),s,delta_s,alpha,L_demi);
    xe = [xe next_x];
    ye = [ye next_y];
end

for i=1602:N-1
```

```
    s = s_tot(i);
    Cd =  MCd(1);
    alpha=-0.5*rho*Cd*d*U^2/T0;
    [next_x,next_y,ang] = next_element(xe(i),ye(i),s,delta_s,alpha,L_demi);
    xe = [xe next_x];
    ye = [ye next_y];
end

for i=1:N
    xe(i) = xe(i)-xe(N);
    ye(i) = -ye(i);
end

for i =1:N-1
    xe = [xe -xe(N-i)];
    ye = [ye ye(N-i)];
end

%plot of the expected shape
figure(1);
plot(xe,ye);
axis([-3 3 -2.5 0.5]);
title('Expected shape');
xlabel('x');
ylabel('y');
```

B2 - next_element.m

```
function [next_x,next_y,ang] = next_element(x,y,s,delta_s,alpha,L_demi);

ang = atan(1/(alpha*(s-L_demi)));
delta_x = delta_s*sin(ang);
delta_y = delta_s*cos(ang);
next_x = x+delta_x;
next_y = y+delta_y;
```

C - EXPERIMENTAL RESULTS ANALYSIS MATLAB CODES

C1 - main.m

```
clear all;
close all;

file='2316'; %choice of test studied
[a1,a2] = catman_read('2316.bin'); %reading the .bin file

forces(file,a1,a2);
speed(file,a1,a2);
markers(file,a1,a2);
shape(file,a1,a2);
current(file,a1,a2);
```

C2 - forces.m

```
function forces(file,a1,a2);

% Plot force on the left side

Fl=a2(55).data;
time = a2(1).data;
figure(1);
plot(time,Fl);
xlabel('time (s)');
ylabel('force (N)');
title('Left side force');

% Plot force on the right side

Fr=a2(54).data;
time = a2(1).data;
figure(2);
plot(time,Fr);
xlabel('time (s)');
ylabel('force (N)');
title('Right side force');

% Plot both forces
figure(3);
plot(time,Fl,time,Fr);
xlabel('time (s)');
ylabel('force (N)');
title('Both');
legend('Left side','Right side');


%filter
T=0.005; %sampling period
Fs=1/T;
d = designfilt('lowpassfir','FilterOrder',200,'CutoffFrequency',0.05,'SampleRate',Fs);
Fl_filt = filtfilt(d,Fl);
Fr_filt = filtfilt(d,Fr);

%Plot force on the left side filtered
figure(11);
plot(time,Fl_filt);
xlabel('time (s)');
ylabel('force (N)');
title('Left side force filtered');

%Plot force on the rigth side filtered
figure(21);
plot(time,Fr_filt);
xlabel('time (s)');
```

```
ylabel('force (N)');
title('Right side force filtered');

%Plot both force filtered
figure(31);
plot(time,Fl_filt,time,Fr_filt);
xlabel('time (s)');
ylabel('force (N)');
title('Both forces filtered');
legend('Left side','Right side');

%calcultaion of the mean force
Flm=mean(Fl(11001:20001))
Frm=mean(Fr(11001:20001))
Fm=(Frm+Flm)/2
```

C3 - speed.m

```
function speed(file,a1,a2);

time = a2(1).data;
L = length (time);
time = time(100:L);
speed = a2(49).data(100:L);
figure(4);
plot(time,speed);
xlabel('time(s)');
ylabel('speed (m/s)');
title('Carriage speed');

%calculation of the mean towing velocity
Sm=mean(speed(6001:8001));
```

C4 - markers.m

```
function markers(file,a1,a2);

time = a2(1).data;
L = length(time);

%reading the markers data
M1x = a2(2).data(100:L);
M1y = a2(3).data(100:L);
M2x = a2(6).data(100:L);
M2y = a2(7).data(100:L);
M3x = a2(10).data(100:L);
M3y = a2(11).data(100:L);
M4x = a2(14).data(100:L);
M4y = a2(15).data(100:L);
M5x = a2(18).data(100:L);
M5y = a2(19).data(100:L);
M6x = a2(22).data(100:L);
M6y = a2(23).data(100:L);
M7x = a2(26).data(100:L);
M7y = a2(27).data(100:L);
M8x = a2(30).data(100:L);
M8y = a2(31).data(100:L);
M9x = a2(34).data(100:L);
M9y = a2(35).data(100:L);
M10x = a2(38).data(100:L);
M10y = a2(39).data(100:L);

time = a2(1).data(100:L);

%plot the markers position in the x-direction
figure(5);
plot(time,M1x,time,M2x,time,M3x,time,M4x,time,M5x,time,M6x,time,M7x,...
    time,M8x,time,M9x,time,M10x);
xlabel('time(s)');
ylabel('markers position in x direction (m)');
legend('M1','M2','M3','M4','M5','M6','M7','M8','M9','M10');

%plot the markers position in the x-direction
figure(6);
plot(time,M1y,time,M2y,time,M3y,time,M4y,time,M5y,time,M6y,time,M7y,...
    time,M8y,time,M9y,time,M10y);
xlabel('time(s)');
ylabel('markers position in y direction (m)');
legend('M1','M2','M3','M4','M5','M6','M7','M8','M9','M10');

%center the markers position in the y direction around 0
m = mean(M6y);
M1y = M1y-m;
M2y = M2y-m;
```

```
M3y = M3y-m;
M4y = M4y-m;
M5y = M5y-m;
M6y = M6y-m;
M7y = M7y-m;
M8y = M8y-m;
M9y = M9y-m;
M10y = M10y-m;

figure(7);
plot(time,M1y,time,M2y,time,M3y,time,M4y,time,M5y,time,M6y,time,M7y,...
    time,M8y,time,M9y,time,M10y);
xlabel('time(s)');
ylabel('markers position in y direction (m) centered');
legend('M1','M2','M3','M4','M5','M6','M7','M8','M9','M10');
```

C5 - shape.m

```
function [XX,YY,t]=shape(file,a1,a2);

time = a2(1).data;
L = length(time);
ind = [4001,8001,12001,16001,20001,24001,28001]; %indexes of the times for which
one wishes to display the boom shape
n = length(ind);

%plot the shape for each selected time
for i= 1:n
    t = time(ind(i));
    b = a2(22).data(ind(i));
    c = a2(19).data(ind(i));
    M1x = a2(2).data(ind(i));
    M1y = a2(3).data(ind(i));
    M2x = a2(6).data(ind(i));
    M2y = a2(7).data(ind(i));
    M3x = a2(10).data(ind(i));
    M3y = a2(11).data(ind(i));
    M4x = a2(14).data(ind(i));
    M4y = a2(15).data(ind(i));
    M5x = a2(18).data(ind(i));
    M5y = a2(19).data(ind(i));
    M6x = a2(22).data(ind(i));
    M6y = a2(23).data(ind(i));
    M7x = a2(26).data(ind(i));
    M7y = a2(27).data(ind(i));
    M8x = a2(30).data(ind(i));
    M8y = a2(31).data(ind(i));
    M9x = a2(34).data(ind(i));
    M9y = a2(35).data(ind(i));
    M10x = a2(38).data(ind(i));
    M10y = a2(39).data(ind(i));
    XX = [M1x M2x M3x M4x M5x M6x M7x M8x M9x M10x];
    YY = [M1y M2y M3y M4y M5y M6y M7y M8y M9y M10y];
    XX = fliplr(XX);
    figure(1);
    axis([-2.5,2.5,-1.2,1.2])
    plot(YY,XX);
    hold on;
end

xlabel('x');
ylabel('y');
title('shape');
legend('20s','40s','60s','80s','100s','120s','140s');
```

C6 - current.m

```
function current(file,a1,a2);

%Plot current in x direction
currentX = a2(51).data;
time = a2(1).data;
figure(8);
plot(time,currentX);
xlabel('time (s)');
ylabel('current (m/s)');
title('Current in the X direction');

%Plot current in the y direction
currentY = a2(52).data;
time = a2(1).data;
figure(9);
plot(time,currentY);
xlabel('time (s)');
ylabel('current (m/s)');
title('Current in the Y direction');

%filter

T = 0.005;
Fs= 1/T;
d = designfilt('lowpassfir','FilterOrder',100,'CutoffFrequency',0.05,'SampleRate',Fs);
a = filtfilt(d,currentX);
b = filtfilt(d,currentY);

%plot current in the x direction filtered
figure(81);
plot(time,a);
xlabel('time (s)');
ylabel('current (m/s)');
title('Current in the X direction filtered');

%plot current in the y direction filtered
figure(91);
plot(time,b);
xlabel('time (s)');
ylabel('current (m/s)');
title('Current in the Y direction filtered');
```

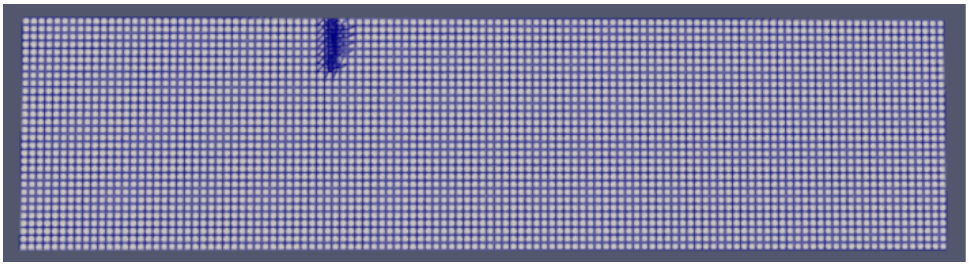
D - DRAG COEFFICIENT PLOT

D1 - Cdplot.m

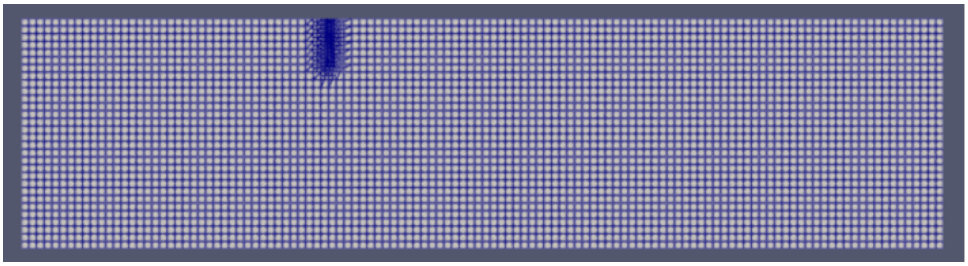
```
file0_0 = 'flatPlateHighRe.csv';
file0_15 = 'flatPlateTheta15.csv';
file1_0 = 'curve1Theta0.csv';
file1_15 = 'curve1Theta15.csv';
file1_30 = 'curve1Theta30.csv';
file1_45 = 'curve1Theta45.csv';
file1_60 = 'curve1Theta60.csv';
file2_0 = 'curve2Theta0.csv';
file2_15 = 'curve2Theta15.csv';
file2_30 = 'curve2Theta30.csv';
file2_45 = 'curve2Theta45.csv';
file2_60 = 'curve2Theta60.csv';
file3_0 = 'curve3Theta0.csv';
file3_15 = 'curve3Theta15.csv';
file3_30 = 'curve3Theta30.csv';

A0_0 = importdata(file0_0);
A0_15 = importdata(file0_15);
A1_0 = importdata(file1_0);
A1_15 = importdata(file1_15);
A1_30 = importdata(file1_30);
A1_45 = importdata(file1_45);
A1_60 = importdata(file1_60);
A2_0 = importdata(file2_0);
A2_15 = importdata(file2_15);
A2_30 = importdata(file2_30);
A2_45 = importdata(file2_45);
A2_60 = importdata(file2_60);
A3_0 = importdata(file3_0);
A3_15 = importdata(file3_15);
A3_30 = importdata(file3_30);

T = A1_0(:,1);
Cd0_0 = A0_0(:,3);
Cd0_15 = A0_15(:,3);
Cd1_0 = A1_0(:,3);
```

```
Cd1_15 = A1_15(:,3);
Cd1_30 = A1_30(:,3);
Cd1_45 = A1_45(:,3);
Cd1_60 = A1_60(:,3);
Cd2_0 = A2_0(:,3);
Cd2_15 = A2_15(:,3);
Cd2_30 = A2_30(:,3);
Cd2_45 = A2_45(:,3);
Cd2_60 = A2_60(:,3);
Cd3_0 = A3_0(:,3);
Cd3_15 = A3_15(:,3);
Cd3_30 = A3_30(:,3);

Cd0_0_5080=Cd0_0(300000:700000);
Cd0_0_mean = mean(Cd0_0_5080);
Cd0_15_5080=Cd0_15(300000:700000);
Cd0_15_mean = mean(Cd0_15_5080);
Cd1_0_5080=Cd1_0(300000:700000);
Cd1_0_mean = mean(Cd1_0_5080);
Cd1_15_5080=Cd1_15(300000:700000);
Cd1_15_mean = mean(Cd1_15_5080);
Cd1_30_5080=Cd1_30(300000:700000);
Cd1_30_mean = mean(Cd1_30_5080);
Cd1_45_5080=Cd1_45(300000:700000);
Cd1_45_mean = mean(Cd1_45_5080);
Cd1_60_5080=Cd1_60(300000:700000);
Cd1_60_mean = mean(Cd1_60_5080);
Cd2_0_5080=Cd2_0(300000:700000);
Cd2_0_mean = mean(Cd2_0_5080);
Cd2_15_5080=Cd2_15(300000:700000);
Cd2_15_mean = mean(Cd2_15_5080);
Cd2_30_5080=Cd2_30(300000:700000);
Cd2_30_mean = mean(Cd2_30_5080);
Cd2_45_5080=Cd2_45(300000:700000);
Cd2_45_mean = mean(Cd2_45_5080);
Cd2_60_5080=Cd2_60(300000:700000);
Cd2_60_mean = mean(Cd2_60_5080);
Cd3_0_5080=Cd3_0(300000:700000);
Cd3_0_mean = mean(Cd3_0_5080);
Cd3_15_5080=Cd3_15(300000:700000);
Cd3_15_mean = mean(Cd3_15_5080);
Cd3_30_5080=Cd3_30(300000:700000);
Cd3_30_mean = mean(Cd3_30_5080);

Theta = [0 15 30 45 60];
Theta3 = [0 15 30];
Theta0 = [0 15];
Cd0 = [Cd0_0_mean Cd0_15_mean];
Cd1 = [Cd1_0_mean Cd1_15_mean Cd1_30_mean Cd1_45_mean Cd1_60_mean];
Cd2 = [Cd2_0_mean Cd2_15_mean Cd2_30_mean Cd2_45_mean Cd2_60_mean];
Cd3 = [Cd3_0_mean Cd3_15_mean Cd3_30_mean];
```

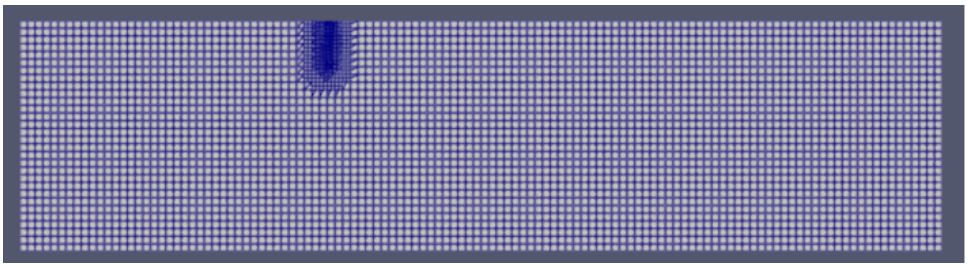
```
figure(1);
plot(Theta0,Cd0,':x',Theta,Cd1,':x',Theta,Cd2,':x',Theta3,Cd3,':x');
xlabel('Inlcination angle (°)');
ylabel('Drag coefficient');
legend('flat plate','curve 1','curve 2','curve 3');

D = [0 10 25 50];
CdD =[Cd0_15_mean Cd1_15_mean Cd2_15_mean Cd3_15_mean];
figure(2);
plot(D,CdD,':x');
xlabel('Distance D (mm)');
ylabel('Drag coefficient');
axis([0 60 1.5 2.1]);
```



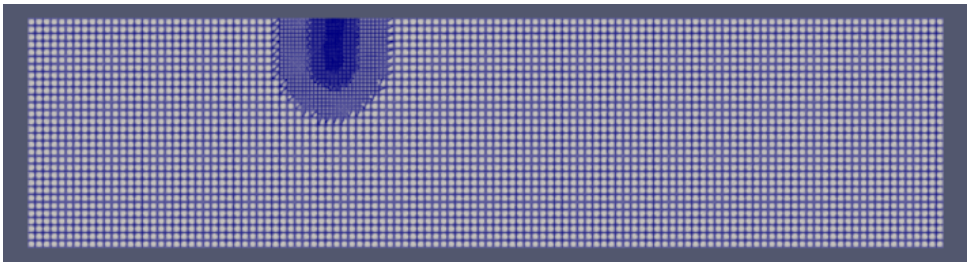
(a) mesh 1



(b) mesh 2

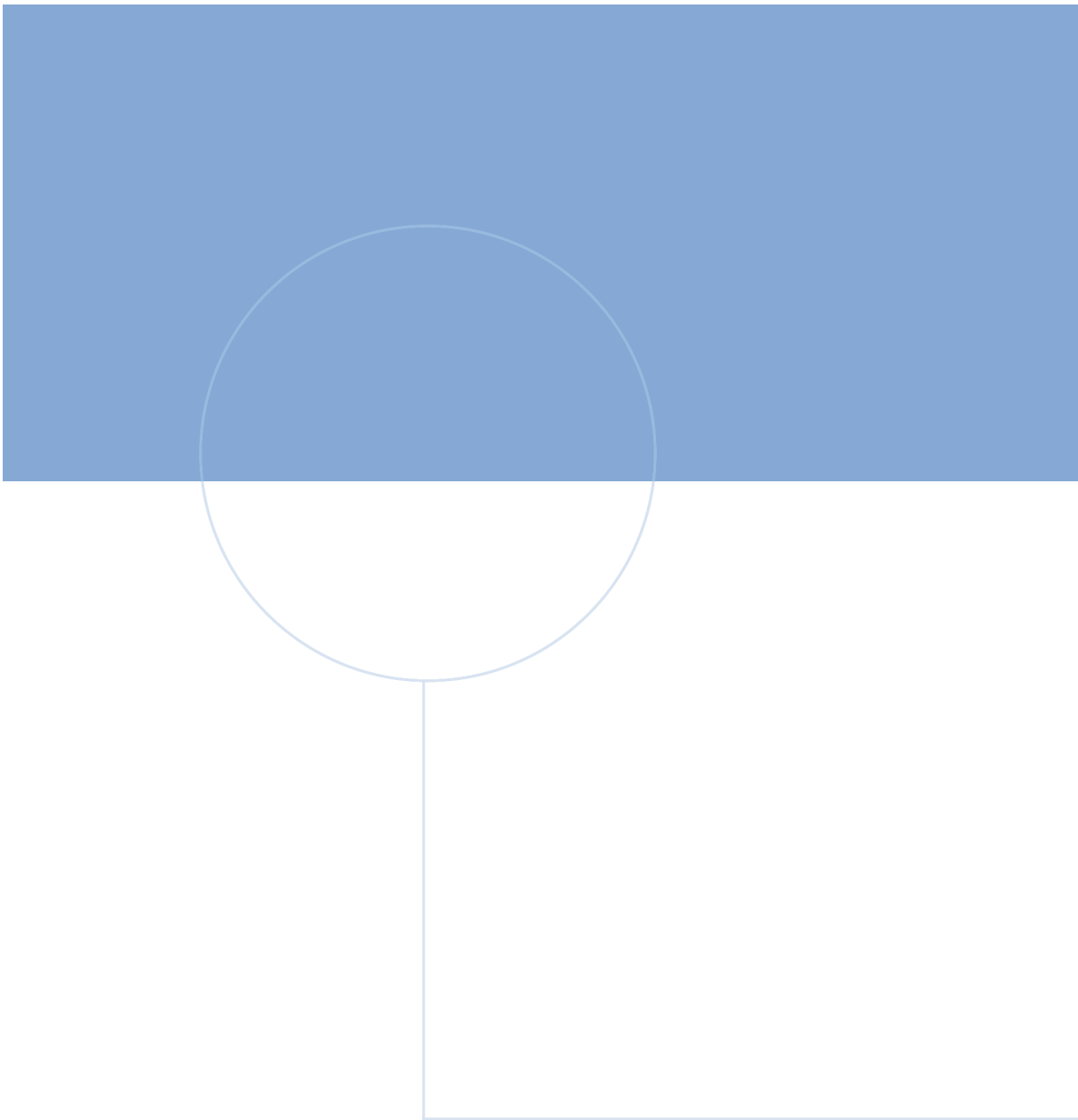


(c) mesh 3



(d) mesh 4

Meshes for the mesh convergence study



NTNU

Norwegian University of
Science and Technology