

Julien-Pooya Weihs

September 2019





Norwegian University of  
Science and Technology

## Julien-Pooya Weihs

MSc in Physics

Submission date: September 2019

Supervisor: Prof. Patrick Espy (NTNU) & Prof. Aleksey Marchenko (UNIS)

Norwegian University of Science and Technology  
Department of Physics



Julien-Pooya Weihs

# **Interaction of surface waves with sea-ice in the marginal ice zone**

Investigation of floe-floe collisions and floe buoyancy oscillations

Supervisors : Prof. Patrick Espy (NTNU) & Prof. Aleksey Marchenko (UNIS)

Master thesis  
for the degree of MSc in Physics

Trondheim, Summer 2019

Norwegian University of Science and Technology  
Faculty of Natural Sciences  
Department of Physics



Norwegian University of  
Science and Technology

NTNU

Norwegian University of Science and Technology

Master thesis  
for the degree of MSc in Physics

Faculty of Natural Sciences  
Department of Physics

© 2019 Julien-Pooya Weihs. All rights reserved

ISBN N/A (printed version)  
ISBN N/A (electronic version)  
ISSN 2019:N/A

Master theses at NTNU, 2019:N/A

Printed by NTNU-trykk

## **Abstract**

This thesis investigates the energy dispersion and damping mechanisms in ice floe-floe collisions and flow buoyancy oscillation in order to gain better understanding of the diffusion of surface waves entering the marginal sea ice zone. We derive some mathematical models motivating preliminary laboratory work in which we perform added mass and collision characteristic time investigations. These experiments lead to an in-situ field work on the sea ice during which characteristic ice-ice collision times will be measured in the air and in the water, as well as buoyancy oscillation characteristic damping times. We find characterizing values of 0.003 [s] and 0.015 [s] respectively in the air and in the water for ice-ice interactions and of 13.63 [s] for buoyancy oscillations.

# Contents

<b>Contents</b>	<b>ii</b>
<b>List of Tables</b>	<b>iii</b>
<b>List of Figures</b>	<b>iv</b>
<b>Acknowledgements</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Context . . . . .	2
1.3 Investigations . . . . .	3
<b>2 Theory</b>	<b>4</b>
2.1 Conservation of mass . . . . .	4
2.1.1 The continuity equation . . . . .	4
2.1.2 The mass balance equation . . . . .	6
2.2 Conservation of momentum . . . . .	6
2.2.1 The momentum balance equation . . . . .	6



---

2.2.2	The Euler equation . . . . .	7
2.2.3	The Navier-Stokes equation . . . . .	7
2.3	Potential flow theory . . . . .	9
2.3.1	The Laplace equation . . . . .	9
2.3.2	The unsteady Bernoulli equation . . . . .	9
<b>3</b>	<b>Modeling</b>	<b>11</b>
3.1	Physical pendulum and added mass . . . . .	11
3.1.1	Simple pendulum . . . . .	11
3.1.2	Physical pendulum in air . . . . .	14
3.1.3	Physical pendulum in water . . . . .	15
3.2	Collision pressure force . . . . .	17
3.3	Interactions between block and wall in water . . . . .	18
3.4	Wave energy dissipation . . . . .	21
3.4.1	Periodic motion of a viscous fluid between two plates . . . . .	21
3.4.2	Buoyancy oscillations of ice cubes . . . . .	24
<b>4</b>	<b>Experimental investigations</b>	<b>28</b>
4.1	Accelerometer : Dytran 4400A1 Vibration Recorder . . . . .	28
4.1.1	Hardware specifications . . . . .	28
4.1.2	Software operations . . . . .	29
4.1.3	Data processing . . . . .	30
4.2	Labwork : added mass investigation . . . . .	34
4.2.1	Added mass investigation : experimental setup . . . . .	34
4.2.2	Added mass investigation : protocol . . . . .	34
4.2.3	Theoretical moment of inertia and angular frequency . . . . .	35
4.2.4	Theoretical added mass of a sphere and a cylinder . . . . .	38

4.3	Labwork : collision pressure force . . . . .	42
4.3.1	Collision pressure force : experimental setup . . . . .	42
4.3.2	Collision pressure force : protocol . . . . .	43
4.3.3	Thales theorem and extrapolation . . . . .	43
4.3.4	Tracker and video analysis . . . . .	44
4.4	Fieldwork : experiments on Valluden facing Sveagruva . . . . .	46
4.5	Fieldwork : Ice block interactions . . . . .	48
4.5.1	Ice block interactions : experimental setup . . . . .	48
4.5.2	Ice block interactions : protocol . . . . .	49
4.5.3	3D modeling and ice block dimensions . . . . .	49
4.6	Fieldwork : Ice cube buoyancy oscillations . . . . .	50
4.6.1	Ice cube buoyancy oscillations : experimental setup . . . . .	50
4.6.2	Ice cube buoyancy oscillations : protocol . . . . .	51
4.6.3	Freeboard and total mass of the ice cube . . . . .	51
<b>5</b>	<b>Data analysis and discussion</b>	<b>55</b>
5.1	Labwork : added mass investigation . . . . .	55
5.1.1	Oscillations in the air . . . . .	55
5.1.2	Oscillations in the water . . . . .	56
5.1.3	Results . . . . .	57
5.2	Labwork : collision pressure force . . . . .	58
5.2.1	Dytran and accelerations . . . . .	59
5.2.2	Tracker and velocities . . . . .	59
5.2.3	Results . . . . .	60
5.3	Fieldwork : Ice block interactions . . . . .	61
5.3.1	Acceleration files . . . . .	62
5.3.2	Video files . . . . .	62

---

5.3.3	Results . . . . .	62
5.4	Fieldwork : Ice cube buoyancy oscillations . . . . .	64
5.4.1	Short time behaviour . . . . .	64
5.4.2	Long time behaviour . . . . .	65
5.4.3	Results . . . . .	66
<b>6</b>	<b>Conclusions</b>	<b>71</b>
6.1	Damping and wave attenuation in the marginal ice zone . . . . .	71
6.2	Future prospects . . . . .	72
6.3	Possible improvements . . . . .	73
<b>7</b>	<b>Appendix</b>	<b>74</b>
7.1	Mathematics . . . . .	74
7.1.1	Curl of a gradient . . . . .	74
7.1.2	Gradient of a product . . . . .	74
7.1.3	Integrating a partial derivative, the Leibniz rule . . . . .	76
7.1.4	Average function value . . . . .	76
7.1.5	Huygens-Steiner or Parallel Axis Theorem . . . . .	77
7.1.6	Gauss or Fundamental Theorem for Divergences . . . . .	78
7.1.7	Material derivative . . . . .	78
7.2	Python . . . . .	79
7.2.1	File loading and reading . . . . .	79
7.2.2	Calibration, filtering and optimization . . . . .	81
7.2.3	Butterworth filter . . . . .	82
7.2.4	Sinus regression . . . . .	83
7.2.5	Graphical display . . . . .	86
7.2.6	Miscellaneous . . . . .	93

7.2.7 Libraries and main code . . . . .	93
<b>Bibliography</b>	<b>99</b>

# List of Tables

2.1	Summary of the presented laws of fluid motion with their hypotheses	5
3.1	Summary of the developed models supporting the various experiments . . . . .	12
4.1	Summary of the items used in the pendulum experiments . . . . .	36
4.2	Measurements for the calibration of the dimensions of the ice block used in the ice block experiments . . . . .	50
5.1	Summary of the results obtained from the added mass investigations	57
5.2	Summary of the results obtained from the collision pressure force experiment . . . . .	62
5.3	Summary of the results obtained in the ice block interactions experiments . . . . .	64
5.4	Summary of the results obtained in the ice cube buoyancy oscillations experiments . . . . .	67

# List of Figures

3.1	Representation of a simple pendulum . . . . .	13
3.2	Representation of a physical pendulum in air . . . . .	14
3.3	Representation of a physical pendulum in water . . . . .	16
3.4	Representation of a body colliding with a displaceable plane . . . . .	17
3.5	Representation of a cylinder approaching a vertical plane . . . . .	18
3.6	Representation of a buoyant oscillating ice cube . . . . .	22
4.1	Illustration of the Dytran 4000A1 Vibration Recorder . . . . .	29
4.2	Graph of an arbitrary pendular motion recorded by a Dytran 4000A1 . . . . .	30
4.3	Illustration of an arbitrary .txt file generated by a Dytran 4000A1 recording . . . . .	31
4.4	Diagram of the general approach for data processing with the Python functions from 7.2 . . . . .	32
4.5	Photograph and representation of the added mass investigation setup . . . . .	35
4.6	Photograph of the support of the Dytran 4000A1 . . . . .	37
4.7	Representation of the pendulum used for the added mass calculation . . . . .	38
4.8	Representation of an accelerating sphere in a fluid . . . . .	39
4.9	Representation of an accelerating cylinder in a fluid . . . . .	40

---

4.10	Photograph and representation of the collision pressure investigation setup . . . . .	42
4.11	Representation of the pendulums used for the collision pressure calculations . . . . .	44
4.12	Photographs of the camera angles when recording from the bottom (left) and from the top (right) in the collision pressure force experiments . . . . .	45
4.13	Coordinates and position of the fieldwork camp on Vallunden facing Sveagruva (Svalbard, Norway) . . . . .	47
4.14	Photograph of the expedition camp on Vallunden . . . . .	47
4.15	Photograph and representation of the pendulum used for the ice block experiments . . . . .	49
4.16	Photograph and representation of the ice cube used in the buoyancy oscillations experiments . . . . .	51
4.17	Representation of the buoyant ice cube in its basin and freeboard . . . . .	52
4.18	Illustration of the Tracker software video file manipulation . . . . .	53
4.19	Illustration of the 3D model of the ice blocks and measurements . . . . .	54
5.1	Time series of the accelerations in the added mass investigation with the ball in the air . . . . .	56
5.2	Time series of the regression in the added mass investigation with the ball in the air . . . . .	57
5.3	Time series of the accelerations in the added mass investigation with the disc in the water . . . . .	58
5.4	Time series of the regression in the added mass investigation with the disc in the water . . . . .	59
5.5	Time series of the accelerations in the collision pressure force experiment with the disc in the air . . . . .	60
5.6	Time series of the zoom on a collision event in collision pressure force experiment with the disc in the air . . . . .	61

5.7	Time series of the acceleration during a collision event between an ice cube and a small block in the air from the ice block interactions experiments . . . . .	63
5.8	Time series of the approach of an ice block (rectangular on the left, pointy on the right) towards a wall in the water in the ice block interactions experiments . . . . .	63
5.9	Time series of the acceleration data recovered from the buoyancy oscillations experiments . . . . .	65
5.10	Time series of the local regression at the start of the buoyancy oscillation data set . . . . .	66
5.11	Time series of the total regression of the buoyancy oscillation data set . . . . .	67
5.12	Illustration of the trajectory of a disc in the air from the collision pressure force experiment . . . . .	69
5.13	Illustration of the trajectory of an ice block against a wall in the ice block interactions experiment . . . . .	70



## Acknowledgements

I would like to thank Prof. Aleksey Marchenko (UNIS) for leading the project and introducing me to exciting and inspiring physics while remaining very available during the summer to answer my questions and uncover doubts I might have about my scientific investigations ; Prof. Vladimir Markov (MSU) for partnering up during lab- and fieldwork and revealing a creative modular approach to experiments that shed an honest and experienced light on the reality of science in arctic conditions ; Dr. Nataly Marchenko (UNIS) for the warm and welcoming hospitality and assistance at the department of Arctic Technologies as well as the provided funding during my stay through the AOCEC project.

I warmly salute Benjamin and Méline Ruffieux and Antonin Pons Braley who, while the first ones introduced me to Svalbard during an unforgettable sailing trip in 2016 and the second one cemented the bonds I was forging between my inner North and these earthly latitudes, became diagonal elements of my identity matrix these years. Aleksandra Visich and Renate Fenzlein for the group working sessions either at the library or at the university that introduced a healthy and productive routine into my days on Svalbard ; Markus Ritschel and David Hokken for the fruitful discussions about coding and the unsuspected introduction to Python that they induced in me during our year of studies at UNIS ; all my friends and fellow students, skiers, ice-climbers, artists and arctic explorers that contributed to embellishing my human adventures on Svalbard.

I am deeply thankful to Valeriya Klementyeva for all the love, support and understanding shown during my two years of master degree and through which I immediately felt like finding a new home when laying foot on Norwegian soil ; Prof. Patrick Espy (NTNU) for his openness and kindness when having me enter his office in the first week of the first semester and making me feel listened to as an independent student despite the originality of my study plan ; all my friends and fellow students, rock-climbers, cyclists, musicians and researchers that rounded up a life experience in Trondheim the memories of which I will cherish all my life.

My highest gratitude goes to my brother Adrien-Kiyan Weihs for the complicity in our mirroring academic achievements, the mathematical counsel and brotherly strength that maintained a precious momentum in me when different drag forces caught up in my trajectory ; my parents Ladan Pooyan-Weihs and Bruno Weihs for the patronage, trust and unconditional love that have been the roots on which all my seeds have been able to grow.

To all these people and to you who reads this thesis I dedicate my work.

# Chapter 1

## Introduction

"Mad Hatter: *Why is a raven like a writing-desk?*

...

- *Have you guessed the riddle yet?*  
the Hatter said, turning to Alice again.

- *No, I give it up,* Alice replied:  
*What's the answer?*

- *I haven't the slightest idea,* said the Hatter"

---

Lewis Carroll

### 1.1 Motivation

The objective of this thesis is the study of ice floe-floe interactions and floe buoyancy oscillations. Through a set of developments, we will display the very wide range of tools from a physicist' skillset this project motivated and stimulated : physical theory, mathematical modeling, coding and programming, systematic laboratory work and arctic fieldwork. We would be very honoured if this thesis reveals to be comprehensive for both a motivated student with a background in physics, mathematics or engineering and a long-time scientist from the field. It is very exciting to realize from the young age of the publication used that this project allowed to grasp the evolutions and discoveries that happen at the level of contemporary research.

## 1.2 Context

The physics of wave action work to reduce the ice in the Arctic through a coincidence of phenomena : the Arctic being mostly ice-covered during the winter season, wave action primarily develops in the warmer summer months. When fracturing the ice, the waves increase the lateral surface area, which in turn accelerates the melting rate of the ice zone due to the resulting smaller floe sizes. But an increase in the fraction of open water in those areas also leads to an increase of the albedo feedback effects and an increase of fetch. The latter induces a further potential for the development of bigger and more damaging waves, amplifying this loop from the start over. This transition zone between this sea-ice and the ice-free waters is called the Marginal Ice Zone.

As the waves create a permanent action on the marginal ice zones, they set the floes in motions, which in turn will cause the ice-ice interactions that we will study here. We would therefore like to remember how sea ice moves on the sea surface. Although the motion of drift ice is two-dimensional, we derive its equation of motion in all three dimensions. The starting point shall be Newton's second law :

$$m \cdot \frac{d\vec{u}}{dt} = \vec{F} \quad (1.1)$$

where  $\vec{u} = (u, v, w)$  is the velocity vector with  $w$  the vertical velocity, and  $\vec{F}$  represents the forcing on the ice.

Following the assumptions of [1], we consider normal continuum mechanics :

- the mass field of continuum particles is given by density in kg/m<sup>3</sup>.
- the thermal stress field forcing is  $\vec{\nabla} \cdot \vec{\Sigma}$ , using the internal stress tensor  $\vec{\Sigma} = \vec{\sigma} - [p_a + \rho_w g D] \cdot \vec{I}$  with  $\vec{\sigma}$  the internal ice stress,  $p_a$  the hydrostatic pressure from air,  $\rho_w g D$  the hydrostatic pressure from sea water ( $D$  being the depth of the water) and  $\vec{I}$  the unit tensor. Hydrostatic pressure compresses the particle normal to the surface and is equal in all directions.

The equation of motion for a continuum therefore is the Cauchy equation:

$$\rho \cdot \left( \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla} u \right) = \vec{\nabla} \cdot \vec{\Sigma} + \vec{F}_{ext} \quad (1.2)$$

where  $\vec{F}_{ext}$  represents the external forcing on the ice in terms of force per unit volume.

The author [1] also introduces geophysical effects in the equation of motion, which add up to:

$$\rho \cdot \left( \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \underbrace{2\vec{\Omega} \times \vec{u}}_{\text{Coriolis}} \right) = \vec{\nabla} \cdot \left( \vec{\sigma} - [p_a + \rho_w g D] \cdot \vec{I} \right) - \underbrace{\rho \vec{\nabla} \Phi}_{\text{gravity}} \quad (1.3)$$

where  $\vec{\Omega}$  is the Coriolis vector and  $\Phi$  is the geopotential height of the sea surface.

Integrating the equation of motion through the thickness of ice, we assume the surface boundary condition to be that the ice shear stress must match the shear stresses of air  $\tau_a$  and water  $\tau_w$  on ice, but also that the horizontal Coriolis acceleration arising from vertical motion is very small compared with the part arising from horizontal motion. This reduces the Coriolis acceleration to  $f\vec{k} \times \vec{u}$  where  $f = 2\Omega \sin \phi$  is the Coriolis parameter (with  $\phi$  the latitude). Our general form of the equation of motion of sea ice on the sea surface plane then becomes:

$$\rho h \cdot \left( \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + f\vec{k} \times \vec{u} \right) = \vec{\nabla} \cdot \vec{\sigma} + \vec{\tau}_a + \vec{\tau}_w - \rho h g \vec{\beta} - h \vec{\nabla} p_a \quad (1.4)$$

where  $\rho h g \vec{\beta}$  is the sea surface tilt and comprises the slope of the sea surface  $\vec{\beta}$ .

### 1.3 Investigations

This thesis presents a work built upon a logical progression of thoughts, skills and motivations. The chapter on Theory will introduce all necessary physical and mathematical tools both for the understanding of the intellectual context and for the preparation of the following personal work. The main fluid dynamics equations relevant to our research will be derived. The chapter on Modeling will produce and present a set of theoretical models on which the applied work will be based. With those we will at the same time support and motivate the laboratory and field investigations. The chapter on the Experimental setup presents the software, hardware as well as the protocols to work with them in our different experiments. It also computes most theoretical values that we will compare our experimental results with. The chapter about the Data analysis and discussion brings the work into this thesis and explains the processing of the data from our investigations. It communicates our findings and results and discusses them with comments and comparisons with publications of peers.

# Chapter 2

## Theory

"The only thing that you absolutely have to know, is the location of the library."

---

Albert Einstein

This chapter aims at presenting the underlying physics to our investigations and will summarize the most important theoretical elements that would be required to follow the upcoming development of mathematical models for the studied sea-ice and ice-ice interactions.

As our experiments bring together solid bodies and fluids, the study of the former can't be consistent without an understanding of the latter. We will therefore present the quantities describing the flow of a fluid ; the laws of physics governing its evolution are derived from the general principles of the conservation of mass and momentum. A summary of the main derived equations is presented in the Table [2.1](#).

### 2.1 Conservation of mass

#### 2.1.1 The continuity equation

We consider a fixed volume  $V$  of fluid with mass :

$$m = \int_V \rho dV \tag{2.1}$$

Continuum	Hypothesis	Equation	Ref.
Fluid	Compressible $\frac{D\rho}{Dt} \neq 0$	Continuity $\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0$	(2.3)
	Incompressible $\frac{D\rho}{Dt} = 0$	Mass balance $\nabla \cdot \vec{u} = 0$	(2.15)
	Viscous $\mu \neq 0$	Momentum balance $\rho \frac{D\vec{u}}{Dt} = \nabla \cdot \sigma + \rho \vec{b}$	
	Inviscid $\mu = 0$	Euler $\rho \frac{D\vec{u}}{Dt} = -\nabla p + \rho \vec{b}$	(2.12)
	Irrotationality $\nabla \times \vec{u} = \vec{0}$	Unsteady Bernoulli $\rho \frac{\partial \phi}{\partial t} + \frac{1}{2} \rho  \nabla \phi ^2 + p = f(t)$	(2.18)

**Table 2.1:** Summary of the presented laws of fluid motion with their hypotheses

where  $\rho$  is the density of the fluid.

To study its variation in time, we follow the intuition of [2] and look at the mass flux density  $\rho \vec{u}$  flowing through the surface  $S$  enclosing the volume  $V$ . By taking the surface element  $d\vec{S}$  to be aligned with the  $\vec{n}$  normal to the surface, which translates into  $d\vec{S} = \vec{n}dS$ , we can write :

$$\begin{aligned}
 \frac{dm}{dt} &= - \oint_S \rho \vec{u} \cdot d\vec{S} \\
 \stackrel{(2.1) \& 7.1.6}{\Leftrightarrow} \int_V \frac{\partial \rho}{\partial t} dV &= - \int_V \nabla \cdot \rho \vec{u} dV \\
 \Leftrightarrow \int_V \left( \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{u} \right) dV &= 0
 \end{aligned} \tag{2.2}$$

Since (2.2) is true for every volume, we can neglect the integral-self and obtain the continuity equation :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \tag{2.3}$$

or with the material derivative presented in 7.1.7 :

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \vec{u} \tag{2.4}$$

### 2.1.2 The mass balance equation

If we assume that the variations of the densities are small enough because the fluid is incompressible, we would have :

$$\frac{D\rho}{Dt} = 0$$

which transforms (2.4) into the mass balance equation :

$$\nabla \cdot \vec{u} = 0 \quad (2.5)$$

## 2.2 Conservation of momentum

### 2.2.1 The momentum balance equation

The development of the balance of momentum equation is motivated by the consideration of the acceleration of a fluid particle. Since the fluid particle is moving along a vector field  $\vec{u}(t, x(t), y(t), z(t))$  in cartesian coordinates, we require the use of the material derivative here as well and express the acceleration as :

$$\vec{a}(t) = \frac{D\vec{u}}{Dt} \quad (2.6)$$

The forces acting on an fluid element can be of two kinds : they can either be body forces affecting the continuum per unit volume (such as gravity or electromagnetic fields) or they can be stress forces acting on the surface of element due to the other elements around it (forces represented through stress tensors of the form  $\sigma_{ij}$  such as hydrostatic pressure or viscosity).

With the theorem 7.1.6, we can write the orthogonal stress forces exerted on the fluid inside a volume  $V$  through the stresses on the boundary  $S$  of  $V$  as :

$$\vec{F}_{\text{stress, viscous}} = \oint_S \sigma_{ij} n_j dS = \int_V (\nabla \cdot \sigma) dV \quad (2.7)$$

where the Einstein summation is used for the repeated index  $j$  and  $n_i$  are the vectors normal to each of the 3 surfaces.

For generic body forces per unit mass  $\vec{b}(t, x, y, z)$ , the total body force on the fluid in the volume  $V$  would be :

$$\vec{F}_{\text{body}} = \int_V \rho \vec{b} dV \quad (2.8)$$

The total force per unit volume is therefore the sum of (2.7) and (2.8) :

$$\vec{F}_{\text{total, unit volume}} = \nabla \cdot \sigma + \rho \vec{b} \quad (2.9)$$

Applying Newton's second law (1.1) brings together (2.6) and (2.9), and yields the momentum balance equation :

$$\rho \frac{D\vec{u}}{Dt} = \nabla \cdot \sigma + \rho \vec{b} \quad (2.10)$$

### 2.2.2 The Euler equation

Considering an ideal fluid where the stress forces are always orthogonal to the surface so that  $\sigma_{ij} = -p\delta_{ij}$ , with  $\delta_{ij}$  the Dirac delta function, we reduce our study to a situation in which the absence of tangential forces means that flow along a surface is non-viscous and irrotational [3]. The difference with (2.7) show in :

$$\vec{F}_{\text{stress, non-viscous}} = - \oint_S p \vec{n} dS = - \int_V (\nabla p) dV \quad (2.11)$$

where  $p = p_0 + \rho g(z_0 + z)$  is the water pressure with the shallow water approximation that the density is considered to be constant,  $z$  the depth,  $z_0$  a reference point (for instance the sea level) and  $p_0 = p(z = z_0)$  the pressure at the reference point (for instance the atmospheric pressure at sea level).

The equation momentum balance equation (2.10) then transforms accordingly into the Euler equation :

$$\rho \frac{D\vec{u}}{Dt} = -\nabla p + \rho \vec{b} \quad (2.12)$$

### 2.2.3 The Navier-Stokes equation

We will derive the Navier-Stokes equation by combining the mass balance (2.5) and the momentum balance (2.10) equations. In fact, the Navier-Stokes equation describing the motion of every viscous fluid, we introduce a strain rate of the form  $\epsilon_{ij}$  (such as viscosity or shear) in function of the velocity field  $\vec{u}$  :



$$\epsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.13)$$

Viscous fluids are characterized by a direct relationship between the stress and this strain rate, this simplest of which is linear [4] :

$$\sigma_{ij} = 2\mu\epsilon_{ij} - \delta_{ij}p \quad (2.14)$$

where we split the shear and bulk components such that the viscosity  $\mu$  only expresses the resistance to shear motion.

The Navier-Stokes equation being computed for an incompressible fluid, we make the assumption  $\nabla \cdot \vec{u} = 0$ . Furthermore, by inserting (2.14) into (2.10) using the strain rate definition of (2.13), we can rewrite the divergence of the stress tensor in coordinates as :

$$\frac{\partial \sigma_{ij}}{\partial x_j} = \frac{\partial(2\mu\epsilon_{ij})}{\partial x_j} - \frac{\partial(\delta_{ij}p)}{\partial x_j} = \mu \frac{\partial}{\partial x_j} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{\partial p}{\partial x_i}$$

Formulating the equation in this way allow us to swap the partial derivatives in  $x_i$  and  $x_j$ , which leads us to the elimination of one of the derivatives in the velocity term :

$$\frac{\partial}{\partial x_j} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) = \frac{\partial}{\partial x_j} \frac{\partial u_i}{\partial x_j} + \frac{\partial}{\partial x_i} \frac{\partial u_j}{\partial x_j} \stackrel{\nabla \cdot \vec{u}=0}{=} \frac{\partial}{\partial x_j} \frac{\partial u_i}{\partial x_j} = \nabla^2 u_i$$

which now allows us to come back to (2.10) and express the Navier-Stokes equation for every  $u_i$  component :

$$\rho \frac{Du_i}{Dt} = \mu \frac{\partial^2 u_i}{\partial x_j^2} - \frac{\partial p}{\partial x_i} + \rho b_i$$

or in an equivalent, vectorial form :

$$\rho \frac{D\vec{u}}{Dt} = \mu \nabla^2 \vec{u} - \nabla p + \rho \vec{b} \quad (2.15)$$

This equation is a reformulation of Newton's second law (1.1) for fluids : the density of the fluid times its acceleration composed of time-dependent and convective

elements is equal to the sum of hydrostatic effects, the diffusion of momentum through viscosity and body forces on the fluid elements.

## 2.3 Potential flow theory

A potential flow is an irrotational flow in which the fluid particles do not rotate, their angular velocity therefore being null.

The angular velocity of a flow is defined in terms of the flow velocity  $\vec{u}$  as :

$$\omega = \frac{1}{2} \cdot \nabla \times \vec{u}$$

and hence, if the angular velocity is zero, then  $\nabla \times \vec{u} = \vec{0}$ , which induces that in such irrotational flows, we can define a velocity potential function  $\phi(x, z, t)$  verifying :

$$\vec{u} = \nabla \phi \tag{2.16}$$

and making sure that  $\omega = 0$  by virtue of the vector identity 7.1.1.

### 2.3.1 The Laplace equation

As explained by [5], the flow velocity  $\vec{u}$  must still satisfy the mass balance equation (2.5). If we substitute our potential function from (2.16) into it, we arrive to the Laplace equation :

$$\nabla \cdot \vec{u} = \nabla \cdot (\nabla \phi) = \nabla \cdot \nabla \phi \equiv \nabla^2 \phi = 0 \tag{2.17}$$

### 2.3.2 The unsteady Bernoulli equation

The unsteady Bernoulli equation is derived from Newton's second law (1.1) for fluids in inviscid conditions, which corresponds to the Euler equation (2.12), but within the frame work of potential flow theory [6]. In fact, decomposing the material derivative in the Euler equation yields :

$$\rho \left( \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} \right) = -\nabla p + \rho \vec{b}$$

Two specifications arise for irrotational flows however :

- $\nabla \times \vec{u} = \vec{0}$ , which means that  $(\vec{u} \cdot \nabla)\vec{u} = \frac{1}{2}\nabla(\vec{u} \cdot \vec{u})$  with respect to the vector identity 7.1.2
- we can use the velocity potential (2.16) instead of the flow velocity

and thus we can write, assuming no body forces  $\vec{b}$  :

$$\begin{aligned}\rho \left( \frac{\partial \nabla \phi}{\partial t} + \frac{1}{2} \nabla (\nabla \phi \cdot \nabla \phi) \right) &= -\nabla p \\ \nabla \left[ \rho \frac{\partial \phi}{\partial t} + \frac{1}{2} \rho (\nabla \phi \cdot \nabla \phi) + p \right] &= 0\end{aligned}$$

An integration through all the spatial derivatives gives us the unsteady Bernoulli equation :

$$\rho \frac{\partial \phi}{\partial t} + \frac{1}{2} \rho |\nabla \phi|^2 + p = f(t) \quad (2.18)$$

where  $f(t)$  is a time-dependent function working as the constant of integration.

# Chapter 3

## Modeling

"All models are wrong but some are useful."

---

George Box

The series of performed experiments motivates us to develop a few models to describe the phenomena at play during the interactions of bodies. We establish a description of pendulum interactions ; first alone with the fluids they oscillate in and then with one another, motivated by the work carried out during the laboratory work. After describing how to compute the collision pressure force between two bodies, we investigate the interactions of a submerged ice block moving towards a wall as it recreates an experiment carried out during the field work. At last, we approach the oscillations of a buoyant ice cube with energy dissipation through its lateral faces in water ; first by describing the dynamics of a viscous fluid between two plates and then through the damping equations affecting the velocity amplitude of the oscillations, a phenomenon observed during another experiment carried out during field work.

The Table 3.1 summarizes the models presented in this chapter for the reader to gain a better overview and access into the mathematical developments hereunder.

### 3.1 Physical pendulum and added mass

#### 3.1.1 Simple pendulum

To work with a physical model for a pendulum, we need to derivate the adequate equations of motion first. For that, we will work in a rotational system with circular

Experiment	Model Name	Inputs	Outputs	Ref.
Pendulum interactions	Physical pendulum in air	$\omega_a$	$I_S = \frac{mgl_{cm}}{\omega_a^2}$	(3.8)
	Physical pendulum in water	$I_S, \omega_w$	$m_{\text{add}} = \frac{1}{l^2} \left( \frac{mgl_{cm}}{\omega_w^2} - I_S \right)$	(3.10)
	Collision pressure force	$r, v_{be}, v_{af}$	$F_{\text{coll}} = -\frac{m}{2r} (v_{af}^2 - v_{be}^2)$	(3.11)
Ice block interactions	Ice block and wall in water	$h_0, h'_0$	$h(t) = h_0 \left( 1 - \frac{h'_0 t}{2h_0} \right)^{-2}$	(3.19)
Buoyant ice cube	Viscous fluid between two plates	$\omega, l$	$u(x, t) = u_0 F(\Lambda, x, l) e^{i\omega t}$	(3.20)
	Buoyancy oscillations of an ice cube	$\omega, l, h_i$	$u_0(t_s) = \eta'_0 e^{-\frac{t_s}{\tau}}$	(3.35)

**Table 3.1:** Summary of the developed models supporting the various experiments

coordinates as introduced in Figure 3.1.

Using the definition of the torque about the pivot point  $P$ , we have :

$$\begin{aligned}
 \vec{\tau}_P &= \vec{r}_{P,m} \times m\vec{g} \\
 &= l\hat{r} \times mg(\cos\theta\hat{r} - \sin\theta\hat{\theta}) \\
 &= -mgl \sin\theta\hat{k}
 \end{aligned} \tag{3.1}$$

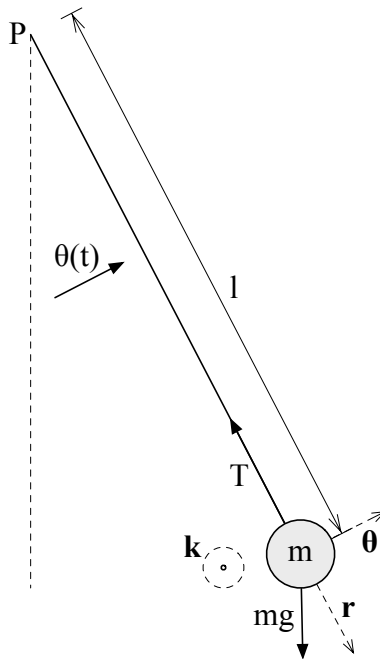
where  $m$  is the mass of the simple pendulum,  $g$  is the gravitational acceleration,  $l$  is the length of the pendulum and  $\theta$  is the angular displacement.

The  $\hat{k}$ -component of the torque about  $P$  is :

$$(\tau_P)_k = -mgl \sin\theta$$

But using another definition for the torque that introduces the angular momentum,  $\vec{\tau} = \frac{d\vec{L}}{dt}$ , and knowing that for a point particle  $\vec{L} = I\vec{\omega}$  where  $I$  is the moment of inertia and  $\omega = \frac{d\theta}{dt}$  the orbital angular velocity, we can write :

$$(\tau_P)_k = I_P \alpha_k = I_P \frac{d^2\theta}{dt^2} \tag{3.2}$$



**Figure 3.1:** Representation of a simple pendulum

Since the moment of inertia for a point mass about a pivot point  $P$  is  $I_P = ml^2$ , we obtain the following differential equation of motion :

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l} \sin \theta \quad (3.3)$$

### Small angle approximation

Assuming that the angle  $\theta$  is very small, we can use the following small angle approximation :

$$\sin \theta \stackrel{\theta \ll 1}{\approx} \theta$$

which transforms (3.3) into :

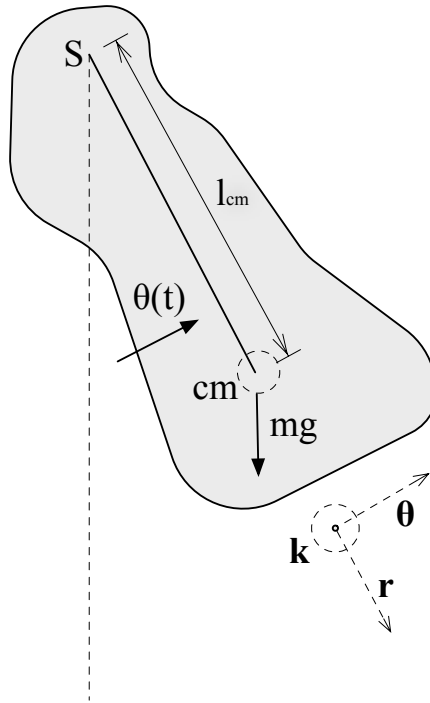
$$\frac{d^2\theta}{dt^2} + \frac{g}{l}\theta = 0 \quad (3.4)$$

Using the ansatz  $\theta(t) = \theta_0 \sin \omega_0 t$ , it follows that the solution to (3.4) is :

$$\theta(t) = \theta_0 \sin \sqrt{\frac{g}{l}}t \quad (3.5)$$

This solution describes a simple harmonic motion with amplitude  $\theta_0 \ll 1$  and an angular frequency of motion  $\omega_0 = \sqrt{\frac{g}{l}}$ .

### 3.1.2 Physical pendulum in air



**Figure 3.2:** Representation of a physical pendulum in air

Contrarily to the simple pendulum, a physical pendulum consists of a rigid body swinging about its fixed point  $S$ , as illustrated in Figure 3.2. This time, the gravitational force acts at its center of mass but the torque analysis remains very similar to that of the simple pendulum. The equation (3.1) rewrites as :

$$\begin{aligned}
 \vec{\tau}_S &= \vec{r}_{S,cm} \times m\vec{g} \\
 &= l_{cm} \hat{r} \times mg(\cos \theta \hat{r} - \sin \theta \hat{\theta}) \\
 &= -mgl_{cm} \sin \theta \hat{k}
 \end{aligned} \tag{3.6}$$

We can then follow the same steps that led to (3.3) and obtain :

$$I_S \frac{d^2\theta}{dt^2} = -mgl_{cm} \sin \theta \tag{3.7}$$

with  $I_S$  the moment of inertia about the fixed point  $S$ .

### Moment of inertia

With the same small angle approximation as before and the ansatz  $\theta(t) = \theta_0 \sin \omega_a t$ , we now derive a expression for the moment of inertia  $I_S$  :

$$I_S = \frac{mgl_{cm}}{\omega_a^2} \tag{3.8}$$

where  $\omega_a$  is the angular frequency of the pendulum in the air.

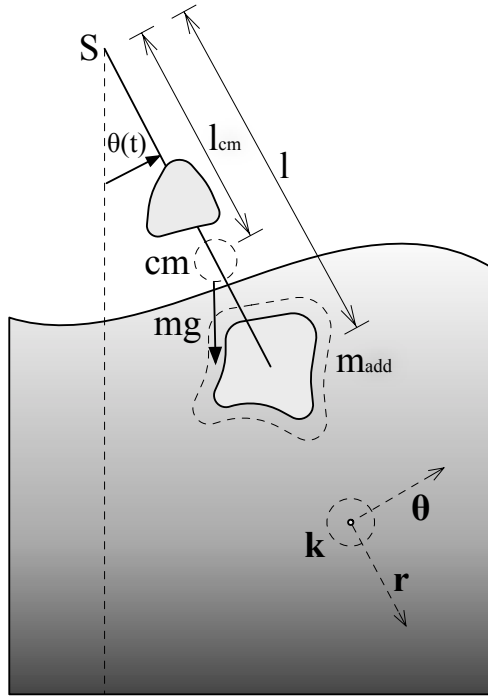
### 3.1.3 Physical pendulum in water

When submerged in water, the motion of the pendulum will be submitted to 2 additional effects : the inertia from the volume of fluid that the accelerating pendulum will displace, formulated in terms of a mass  $m_{\text{add}}$  multiplied by the length of the pendulum's rod  $l$  and added to the moment of inertia [7], and a drag acting against the motion of the pendulum with respect to the fluid, coming from the Morison formula presented in [8] :

$$(I_S + m_{\text{add}}l^2) \frac{d^2\theta}{dt^2} = -mgl_{cm} \sin \theta - \underbrace{\frac{1}{2}\rho_w C_D S l^2 \left| \frac{d\theta}{dt} \right| \frac{d\theta}{dt}}_{\text{drag}} \tag{3.9}$$

where  $\rho_w$  is the density of the water,  $C_D$  the drag coefficient depending on the shape of the pendulum and  $S$  the cross section of the displaced area in the fluid.





**Figure 3.3:** Representation of a physical pendulum in water

### Added mass

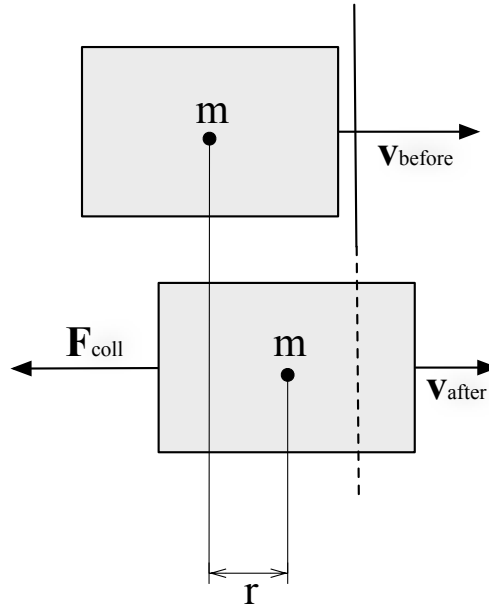
We here proceed to 2 approximations, the small angle and the low speed one. The first one reduces  $\sin \theta$  to  $\theta$  as before and the second one makes us neglect the drag term containing  $(\frac{d\theta}{dt})^2$ . With the ansatz  $\theta(t) = \theta_0 \sin \omega_w t$ , (3.9) reduces to :

$$I_S + m_{\text{add}} l^2 = \frac{m g l_{\text{cm}}}{\omega_w^2}$$

This allows for a convenient way to express the added mass  $m_{\text{add}}$  as a function of our other parameters, using the form of  $I_S$  from (3.8) :

$$\begin{aligned} m_{\text{add}} &= \frac{1}{l^2} \left( \frac{m g l_{\text{cm}}}{\omega_w^2} - I_S \right) \\ &= \frac{m g l_{\text{cm}}}{l^2} \left( \frac{1}{\omega_w^2} - \frac{1}{\omega_a^2} \right) \end{aligned} \quad (3.10)$$

## 3.2 Collision pressure force

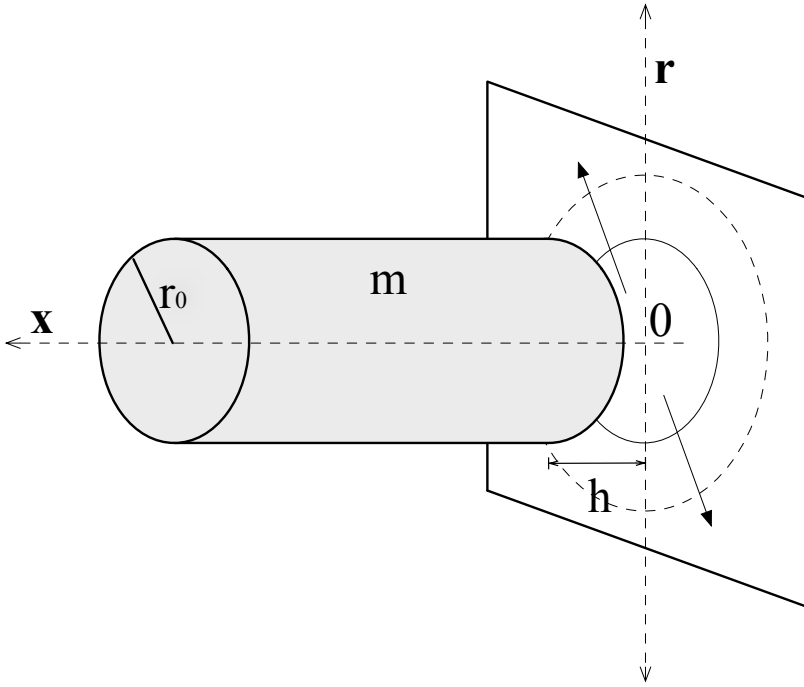


**Figure 3.4:** Representation of a body colliding with a displaceable plane

We build a simple model in order to be able to measure the collision pressure forces from the interactions of physical pendulums of different shapes in the air and in the water.

At the moment of a horizontal collision of a body with a "displaceable" (such that the velocity after the collision is reduced by not 0) wall, we neglect the gravitational force for the objects we will consider are in equilibrium with respect to the vertical direction. We will compute the average collision pressure force over the collision time using the work-energy principle : the work done on the body  $W = \vec{F} \cdot \Delta\vec{s}$  is equal to its change in kinetic energy  $\Delta K = \frac{1}{2}m\Delta v^2$ , expressed as :

$$\begin{aligned}
 W = \Delta K &\Leftrightarrow -F_{\text{coll}}r = \frac{1}{2}m(v_{\text{after}}^2 - v_{\text{before}}^2) \\
 &\Leftrightarrow F_{\text{coll}} = -\frac{m}{2r}(v_{\text{after}}^2 - v_{\text{before}}^2)
 \end{aligned}
 \tag{3.11}$$



**Figure 3.5:** Representation of a cylinder approaching a vertical plane

### 3.3 Interactions between block and wall in water

Our model for the interaction between the submerged block and the wall presents an axially symmetric interaction between a cylinder and a vertical plane as illustrated in Figure 3.5.

We will approach the equation of motion for the cylinder through the mass (2.5) and momentum balance (2.12) equations, neglecting body forces such as gravity, for the water layer between the cylinder and the wall :

$$\begin{aligned} r \frac{dh}{dt} + h \frac{\partial(rv_r)}{\partial r} &= 0 \\ \frac{\partial v_r}{\partial t} + v_r \frac{\partial v_r}{\partial r} &= -\frac{1}{\rho} \frac{\partial p}{\partial r} \end{aligned} \quad (3.12)$$

where  $r$  is the radial coordinate,  $v_r$  is the radial velocity of the water,  $h(t)$  is the water layer thickness,  $t$  is the time,  $p$  is the water pressure and  $\rho$  the water density.

Isolating the radial velocity in the first equation of (3.12) yields ( $h > 0$ ) :

$$\begin{aligned}
 & r \frac{dh}{dt} + h \frac{\partial(rv_r)}{\partial r} = 0 \\
 \Leftrightarrow & \frac{\partial(rv_r)}{\partial r} = -\frac{r}{h} \frac{dh}{dt} \\
 \int \frac{dr}{dr} & \Leftrightarrow rv_r = -\frac{r^2}{2h} \frac{dh}{dt} \\
 \Leftrightarrow & v_r = -\frac{r}{2h} \frac{dh}{dt}
 \end{aligned} \tag{3.13}$$

The two integration steps in (3.13) are bound to the condition 7.1.3.

We can now substitute the result of (3.13) into the second equation of (3.12) and find an expression for the water pressure  $p$  :

$$\begin{aligned}
 & \frac{\partial}{\partial t} \left( -\frac{r}{2h} \frac{dh}{dt} \right) - \frac{r}{2h} \frac{dh}{dt} \cdot \frac{\partial}{\partial r} \left( -\frac{r}{2h} \frac{dh}{dt} \right) = -\frac{1}{\rho} \frac{\partial p}{\partial r} \\
 \Leftrightarrow & \frac{r}{2h} h'^2 - \frac{r}{2h} h'' + \frac{r}{2h} h' \cdot \frac{1}{2h} h' = -\frac{1}{\rho} \frac{\partial p}{\partial r} \\
 \Leftrightarrow & \frac{r}{2h} \left( h'' - \frac{3}{2} \frac{h'^2}{h} \right) = \frac{1}{\rho} \frac{\partial p}{\partial r} \\
 \stackrel{7.1.3}{\Leftrightarrow} & \int_{r_0}^r \frac{r}{2h} \left( h'' - \frac{3}{2} \frac{h'^2}{h} \right) dr = \int_{p_0}^p \frac{1}{\rho} dp \\
 \Leftrightarrow & \frac{r^2 - r_0^2}{4h} \left( h'' - \frac{3}{2} \frac{h'^2}{h} \right) = \frac{p - p_0}{\rho} \\
 \Leftrightarrow & p = \frac{\rho \cdot (r^2 - r_0^2)}{4h} \left( h'' - \frac{3}{2} \frac{h'^2}{h} \right) + p_0
 \end{aligned} \tag{3.14}$$

where  $p_0$  denotes the pressure on the cylinder's surface.

The pressure force, the force created by a pressure applied over a surface, would be formulated for the base of the cylinder as :

$$F_p = 2\pi \int_0^{r_0} pr dr \tag{3.15}$$

By using the expression found in (3.14) in (3.15), we get a force :

$$\begin{aligned}
F_p &= 2\pi \int_0^{r_0} \left( \frac{\rho \cdot (r^2 - r_0^2)}{4h} \left( h'' - \frac{3}{2} \frac{h'^2}{h} \right) + p_0 \right) \cdot r dr \\
&= \frac{2\pi\rho}{4h} \cdot \left( h'' - \frac{3}{2} \frac{h'^2}{h} \right) \int_0^{r_0} (r^2 - r_0^2) \cdot r dr + \underbrace{2\pi \int_0^{r_0} p_0 r dr}_{F_{p0} = \pi r_0^2 p_0} \quad (3.16) \\
&= -\frac{\pi\rho r_0^4}{8h} \left( h'' - \frac{3}{2} \frac{h'^2}{h} \right) + F_{p0}
\end{aligned}$$

From the second law of Newton (1.1) and a dimensional analysis of the result in (3.16), it appears that  $\frac{\rho r_0^4}{h}$  has the dimensions of mass. Adding the mass of the cylinder  $m$  to the mass created by the pressure of the water, we get an equation describing the motion of the cylinder :

$$\left( m + \frac{\pi\rho r_0^4}{8h} \right) \cdot h'' = \frac{3}{16} \frac{\pi\rho r_0^4}{h^2} \cdot h'^2 + F_{p0} \quad (3.17)$$

When approaching the vertical wall, (3.17) can be simplified as  $h \rightarrow 0$  into :

$$h'' = \frac{3}{2} \frac{h'^2}{h} \quad \Leftrightarrow \quad 2hh'' = 3h'^2 \quad (3.18)$$

The equation (3.18) is a second-order nonlinear ordinary differential equation in  $h(t)$  that has to be solved in different steps :

1. we re-write (3.18) in differential form :

$$2h \frac{d^2h}{dt^2} = 3 \left( \frac{dh}{dt} \right)^2$$

2. we proceed to the change of variable  $v = \frac{dh}{dt}$ , which yields :

$$\frac{d^2h}{dt^2} = \frac{d}{dt} \left( \frac{dh}{dt} \right) = \frac{dv}{dt} = \frac{dv}{dh} \frac{dh}{dt} = \frac{dv}{dh} v$$

3. replacing  $v$  into (3.18), we now obtain a first order separable differential equation :

$$2h \frac{dv}{dh} v = 3v^2$$

4. solving it gives 2 solutions :

$$2h \frac{dv}{dh} v - 3v^2 = 0 \quad \Rightarrow \quad \begin{cases} v = 0 \\ 2h \frac{dv}{dh} - 3v = 0 \end{cases}$$

5. the first of the two induces  $h$  being a constant, which physically corresponds to an absence of motion. We will therefore focus on the second one and solve this new differential equation :

$$\begin{aligned} & \int_{v_0}^v 2 \frac{dv}{v} = \int_{h_0}^h 3 \frac{dh}{h} \\ \Leftrightarrow & 2 \log \frac{v}{v_0} = 3 \log \frac{h}{h_0} \\ \Leftrightarrow & \frac{v}{v_0} = \frac{h'}{h'_0} = \left( \frac{h}{h_0} \right)^{\frac{3}{2}} \\ \Leftrightarrow & h' = \frac{dh}{dt} = h'_0 \cdot \left( \frac{h}{h_0} \right)^{\frac{3}{2}} \tag{3.19} \\ \Leftrightarrow & \int_{h_0}^h h^{-\frac{3}{2}} dh = h'_0 h_0^{-\frac{3}{2}} \int_{t_0}^t dt \\ \Leftrightarrow & -2 \left( h^{-\frac{1}{2}} - h_0^{-\frac{1}{2}} \right) = h'_0 h_0^{-\frac{3}{2}} (t - t_0) \\ \Leftrightarrow & h(t) = 4h_0 \left( 2 - \frac{h'_0}{h_0} (t - t_0) \right)^{-2} \stackrel{t_0=0}{=} h_0 \left( 1 - \frac{h'_0 t}{2h_0} \right)^{-2} \end{aligned}$$

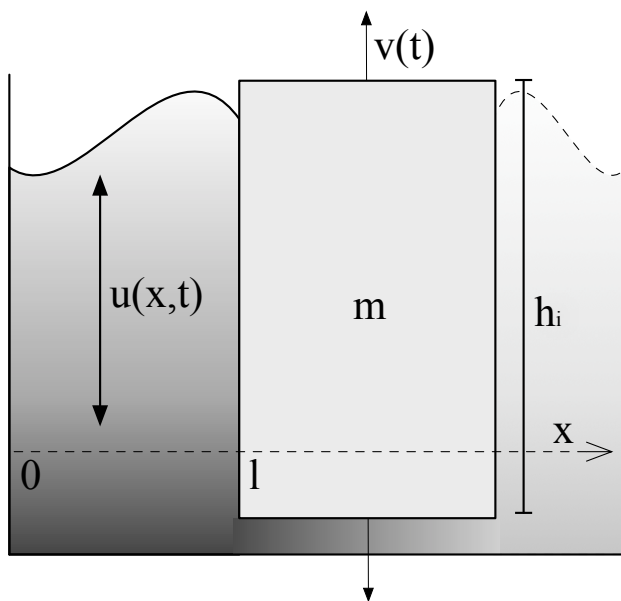
As we know that the initial cylinder velocity  $h'_0 < 0$  and the initial water layer thickness  $h_0 > 0$  by construction in Figure 3.5, it is quite easy to check that the solution of (3.19),  $h = h_0 \left( 1 - \frac{h'_0 t}{2h_0} \right)^{-2}$  exists, verifies (3.18) properly and doesn't diverge for  $t > 0$ .

It is important to realize that although  $\lim_{t \rightarrow \infty} h(t) = 0$ , there is no existing time  $t_{\text{col}}$  at which  $h(t_{\text{col}}) = 0$  ; the interactions between a submerged block and a vertical wall don't lead to a physical contact with our model. This no-contact behaviour has been theoretically proven in other similar situations like that of a rigid ball moving into a viscous incompressible fluid over a fixed plane by [9].

## 3.4 Wave energy dissipation

### 3.4.1 Periodic motion of a viscous fluid between two plates

When a buoyant cube oscillates vertically in a hole of water, the damping of the oscillations results from a dissipation of the energy in the gaps between the cube and



**Figure 3.6:** Representation of a buoyant oscillating ice cube

the walls of the water hole. We can estimate it analytically computing the solution describing the periodic motion of a viscous fluid between two plates separated by a distance  $l$ .

The Navier-Stokes equations describing the motion of the fluid (2.15) can here be reduced to a one-dimensional equation [10] :

$$\rho_w \frac{\partial u}{\partial t} = \mu_v \frac{\partial^2 u}{\partial x^2} \quad (3.20)$$

for  $x \in (0, l)$  and boundary conditions :

$$\begin{aligned} u(0, t) &= 0 \\ u(l, t) &= u_0 e^{i\omega t} \end{aligned} \quad (3.21)$$

Here  $u(x, t)$  is the water velocity in the direction parallel to the plates located in  $x = 0$  and  $x = l$  as drawn onto the Figure 3.6 ; one of the plates (the wall of the hole) is fixed and the other plate (the wall of the cube) oscillates with the frequency  $\omega$  along its plane. Furthermore,  $u_0, \omega \in \mathbb{R}$ .

By defining  $\kappa = \frac{\mu v}{\rho_1}$ , (3.20) becomes an analog of the one-dimensional heat equation and will be resolved as such :

$$u_t = \kappa u_{xx} \quad (3.22)$$

To make the solution simpler, we first implement a non-dimensionalization of the terms by defining a characteristic length, time and velocity :  $L_*$ ,  $T_*$  and  $U_*$ . We can then introduce our dimensionless variables :

$$\begin{aligned} \hat{x} &= \frac{x}{L_*} \\ \hat{t} &= \frac{t}{T_*} \\ \hat{u}(\hat{x}, \hat{t}) &= \frac{u(x, t)}{U_*} \end{aligned}$$

We then have to choose the right characteristic quantities. When it comes to the length, a sensible choice seems to be  $L_* = l$  such that while  $0 < x < l$ , we have  $0 < \hat{x} < 1$ . To find the right  $T_*$ , we use the chain rule and write :

$$\begin{aligned} u_t &= \frac{\partial u}{\partial t} = U_* \frac{\partial \hat{u}}{\partial \hat{t}} \frac{\partial \hat{t}}{\partial t} = \frac{U_*}{T_*} \frac{\partial \hat{u}}{\partial \hat{t}} \\ u_x &= \frac{\partial u}{\partial x} = U_* \frac{\partial \hat{u}}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial x} = \frac{U_*}{L_*} \frac{\partial \hat{u}}{\partial \hat{x}} \\ u_{xx} &= \frac{U_*}{L_*^2} \frac{\partial^2 \hat{u}}{\partial \hat{x}^2} \end{aligned}$$

By substituting these into the heat equation (3.22), we find :

$$u_t = \kappa u_{xx} \quad \Rightarrow \quad \frac{\partial \hat{u}}{\partial \hat{t}} = \frac{T_* \kappa}{L_*^2} \frac{\partial^2 \hat{u}}{\partial \hat{x}^2}$$

which motivates us to choose the characteristic time scale of diffusion as  $T_* = \frac{L_*^2}{\kappa} = \frac{l}{\kappa}$ , that all together and dropping the hats for the sake of convenience gives us the dimensionless equation :

$$u_t = u_{xx} \quad (3.23)$$



with the boundary conditions (3.21) now reformulated as :

$$\begin{aligned} u(0, t) &= 0 \\ u(1, t) &= u_0 e^{i\omega t} \end{aligned} \quad (3.24)$$

The resolution of this system of equation and boundary conditions follows developed in the Section 7.4 of [11] and the author shows that the full solution bears the form :

$$u(x, t) = A \frac{e^{\sqrt{\frac{\omega}{2}}(1+i)x} - e^{-\sqrt{\frac{\omega}{2}}(1+i)x}}{e^{\sqrt{\frac{\omega}{2}}(1+i)} - e^{-\sqrt{\frac{\omega}{2}}(1+i)}} e^{i\omega t}$$

In fact, in our case it is [10] :

$$u(x, t) = u_0 \frac{e^{(1+i)\lambda x} - e^{-(1+i)\lambda x}}{e^{(1+i)\lambda l} - e^{-(1+i)\lambda l}} e^{i\omega t} \quad (3.25)$$

where  $\lambda = \sqrt{\frac{\rho_w \omega}{2\mu_v}}$ .

### 3.4.2 Buoyancy oscillations of ice cubes

The oscillations of a buoyant ice cube in a water hole can be approached by the energy conservation law. In fact, the energy of the buoyancy oscillations of a cube are the sum of its kinetic and potential energies :

$$E = m \cdot \left( \frac{\eta'^2}{2} + \frac{\rho_w g}{\rho_i h_i} \frac{\eta^2}{2} \right) \quad (3.26)$$

where  $m$  is the mass of the cube,  $\eta'$  is the vertical velocity of the cube,  $\eta$  is the vertical displacement of the cube,  $g$  is the gravitational acceleration,  $\rho_w$  is the density of the water,  $\rho_i$  is the density of the ice and  $h_i$  is the ice thickness. The Figure 3.6 schematizes such a setup.

From the energy conservation law  $\frac{dE}{dt} = 0$ , we have :

$$\frac{d}{dt} \left[ m \cdot \left( \frac{\eta'^2}{2} + \frac{\rho_w g}{\rho_i h_i} \frac{\eta^2}{2} \right) \right] = m\eta' \cdot \left( \eta'' + \frac{\rho_w g}{\rho_i h_i} \eta \right) = 0 \quad (3.27)$$

from which we discard the static case  $\eta' = 0$ . With the ansatz  $\eta(t) = \eta_0 \sin \omega t$ , we are able to derive a formula for the natural frequency of the oscillations of the cube  $\omega$  :

$$-\omega^2 \sin \omega t + \frac{\rho_w g}{\rho_i h_i} \sin \omega t = 0 \quad \Rightarrow \quad \omega = \pm \sqrt{\frac{\rho_w g}{\rho_i h_i}} \quad (3.28)$$

where we only keep the positive solution  $\omega = \sqrt{\frac{\rho_w g}{\rho_i h_i}}$  for physical consistency.

Inserting this frequency (3.28) into our ansatz, we express the displacement of the buoyant ice cube by :

$$\eta(t) = \eta_0 \sin \sqrt{\frac{\rho_w g}{\rho_i h_i}} t \quad (3.29)$$

with  $\eta_0 = \eta(0)$  the initial arbitrary value of the position of the cube.

In the configuration illustrated by the Figure 3.6, the energy dissipated over the unit area of the plates is given by [10] :

$$D = \frac{\mu_v}{2} \int_0^l \left| \frac{\partial u}{\partial x} \right|^2 dx \quad (3.30)$$

As we have computed the velocity of a viscous fluid between two plates in (3.25), we can substitute the solution into (3.30) and obtain :

$$\begin{aligned} D &= \frac{\mu_v}{2} \int_0^l \left| \frac{\partial}{\partial x} \left( u_0 \frac{e^{(1+i)\lambda x} - e^{-(1+i)\lambda x}}{e^{(1+i)\lambda l} - e^{-(1+i)\lambda l}} e^{i\omega t} \right) \right|^2 dx \\ &= u_0^2 \frac{\mu_v}{2} |e^{i\omega t}|^2 \int_0^l \left| \frac{\partial}{\partial x} \left( \frac{e^{(1+i)\lambda x} - e^{-(1+i)\lambda x}}{e^{(1+i)\lambda l} - e^{-(1+i)\lambda l}} \right) \right|^2 dx \end{aligned}$$

As  $\omega \in \mathbb{R}$ ,  $|e^{i\omega t}| = 1$  and therefore :

$$\begin{aligned} D &= u_0^2 \frac{\mu_v}{2} \int_0^l \left| \frac{\partial}{\partial x} \left( \frac{e^{(1+i)\lambda x} - e^{-(1+i)\lambda x}}{e^{(1+i)\lambda l} - e^{-(1+i)\lambda l}} \right) \right|^2 dx \\ &= \mu_v \lambda \gamma u_0^2 \end{aligned} \quad (3.31)$$

with  $\gamma = \frac{1}{4} \frac{\sinh 2\lambda + \sin 2\lambda}{\cosh 2\lambda - \cos 2\lambda}$ .

The oscillating plates here are the lateral surfaces of the buoyant ice cube, the width of the gap between the walls is  $l$  and the oscillation velocity corresponds to the boundary condition (3.21), which means that  $\eta' = u_0 \cos \omega t$ . We can integrate this expression to find the displacement  $\eta$  :

$$\eta' = \frac{d\eta}{dt} = u_0 \cos \omega t \quad \Rightarrow \quad \eta = \frac{u_0}{\omega} \sin \omega t \quad (3.32)$$

A quick comparison between this expression for  $\eta$  and the one in (3.29) allows us to identify that  $\eta_0 = \frac{u_0}{\omega}$ .

We can now properly compute the mean buoyant ice cube energy (3.26) averaged over a period of oscillation, with respect to 7.1.4 :

$$\begin{aligned} \langle E \rangle &= m \cdot \left( \frac{\langle \eta'^2 \rangle}{2} + \frac{\rho_w g}{\rho_i h_i} \frac{\langle \eta^2 \rangle}{2} \right) \\ &= m \cdot \left( \frac{\omega}{2\pi} \frac{\int_0^{2\pi} (u_0 \cos \omega t)^2 dt}{2} + \frac{\rho_w g}{\rho_i h_i} \frac{\omega}{2\pi} \frac{\int_0^{2\pi} \left(\frac{u_0}{\omega} \sin \omega t\right)^2 dt}{2} \right) \\ &\stackrel{\omega = \sqrt{\frac{\rho_w g}{\rho_i h_i}}}{=} \frac{m u_0^2}{2} \cdot \left( \frac{\omega}{2\pi} \int_0^{2\pi} (\cos \omega t)^2 dt + \frac{\omega}{2\pi} \int_0^{2\pi} (\sin \omega t)^2 dt \right) \quad (3.33) \\ &= \frac{m u_0^2}{2} \cdot \frac{\omega}{2\pi} \left( \left[ \frac{2\omega t + \sin 2\omega t}{4\omega} \right]_0^{2\pi} + \left[ \frac{2\omega t - \sin 2\omega t}{4\omega} \right]_0^{2\pi} \right) \\ &= \frac{m u_0^2}{2} \end{aligned}$$

This energy dissipation over the plates will lead to a damping mechanism affecting the amplitude of the oscillations  $u_0$ . By assuming that the damping time is much greater than the period of oscillations, in other words that the damping is overall small, we can introduce a new "slow" time  $t_s = \epsilon t$ ,  $\epsilon \ll 1$  such that  $u_0 = u_0(t_s)$ . We also set  $\eta'_0 \equiv \eta'|_{t_s=0} = u_0(t_s = 0) \cos \omega t$  and by expressing the area of submerged lateral surface of the ice cube as  $S_l = 4 \left( \frac{\rho_i h_i}{\rho_w} \right)^2$  and remembering the energy dissipation over the plates by unit area (3.31), we use (3.33) to write the energy balance as :

$$\frac{d\langle E \rangle}{dt_s} = -S_l D \quad (3.34)$$

We now have everything at hand to find the exponential decrease of the velocity amplitude, following up on (3.34) :

$$\begin{aligned}
 \frac{d\langle E \rangle}{dt_s} &= -S_l D \\
 \Leftrightarrow m u_0 \frac{du_0}{dt_s} &= -S_l \mu_v \lambda \gamma u_0^2 \\
 \Leftrightarrow \int_{\eta'_0}^{u_0(t_s)} \frac{du_0}{u_0} &= \int_0^{t_s} \frac{-S_l \mu_v \lambda \gamma}{m} dt_s \quad (3.35) \\
 \Leftrightarrow \log \frac{u_0(t_s)}{\eta'_0} &= \frac{-S_l \mu_v \lambda \gamma}{m} t_s \\
 \Leftrightarrow u_0(t_s) &= \eta'_0 \cdot e^{-\frac{t_s}{\tau}}
 \end{aligned}$$

where  $\tau = \frac{m}{S_l \mu_v \lambda \gamma}$  is the characteristic time of damping and has the same significance as a decay mean lifetime, here being the time at which the oscillation amplitude is reduced to  $\frac{1}{e} \approx 37\%$  of its initial value.

# Chapter 4

## Experimental investigations

"The difficult is what takes a little time. The impossible is what takes a little longer."

---

Fridtjof Nansen

This chapter aims at introducing and preparing all the necessary devices, setups, settings, and protocols to perform the different investigations and run the several experiments motivated by this thesis, as well as developing the necessary programming, numerical and practical tools for a successful data acquisition and analysis of the results. As such, the chapter presents both of the software and hardware used throughout our research and decomposes each section into parts on the experimental preliminaries, the working proceedings and the applied theory we could extract from our models in each situation.

### 4.1 Accelerometer : Dytran 4400A1 Vibration Recorder

Since all our experiments feature accelerometers, we will in this section develop a description of the devices we manipulated, how we collected, read and analyzed the recorded data.

#### 4.1.1 Hardware specifications

The accelerometers used in this series of measurements are Dytran 4000A1 triaxial acceleration recorder operating on a range of  $\pm 200 g_{\text{peak}}$  and  $0 - 1000$  [Hz] with a noise floor of  $0.06 g_{\text{rms}}$ , produced by Dytran (Dynamic Transducers and Systems) Instruments Incorporation and illustrated in the Figure 4.1. Further technical in-

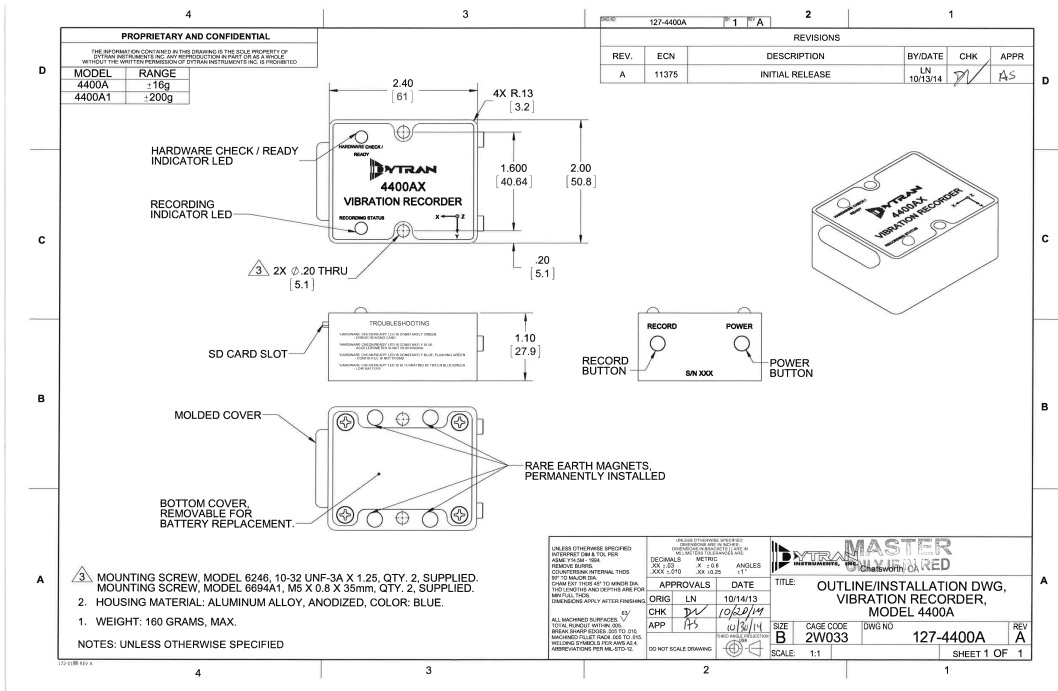


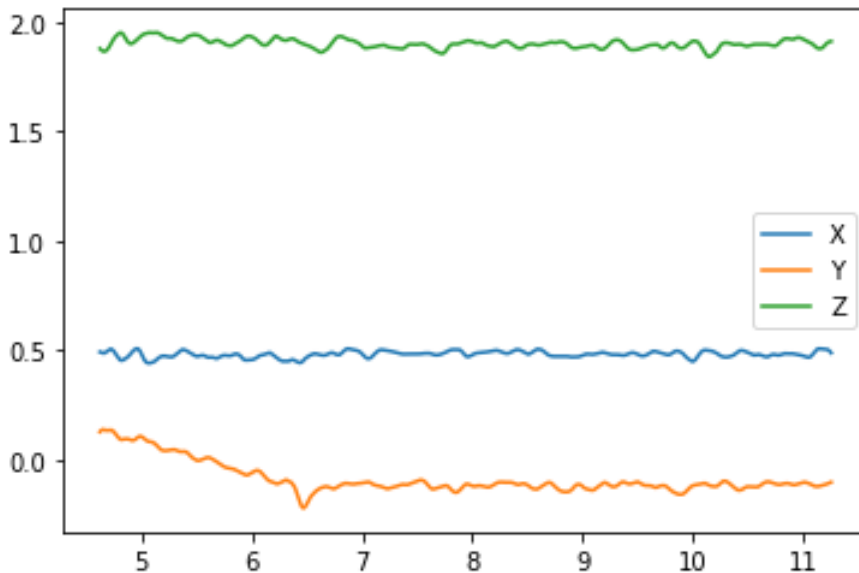
Figure 4.1: Illustration of the Dytran 4000A1 Vibration Recorder

formations and performance specifications being given in [12] but the devices being DC accelerometers, they would have an offset of  $1g$  if placed on a table in the  $z$ -direction, as described by [13]. As such, an arbitrary time series of a measurement along a pendular motion with the  $z$ -axis pointing upwards would produce something like the Figure 4.2.

### 4.1.2 Software operations

To effectively use the Dytran 4000A1, we have to download its appropriate configuration file onto its SD card, reformatting it in the process, for it to operate according to the specification that we will have chosen. This procedure is recommended before every use. The configuration file is to be created through the Vibracorder software and different options are available depending on how the data acquisition is supposed to happen in the experiments : the sampling rate, the triggered level and, in case of an automatic launch, the recording time.

Once the SD card is inserted into the accelerometer and the device hence set up, we activate it through the pressure of its power button and let its configuration run its data collection. The data retrieved is saved into a file in the Vibracorder native



**Figure 4.2:** Graph of an arbitrary pendular motion recorded by a Dytran 4000A1

format. We have to manually create a .txt file through the software in order to process the data with regards to our investigation. An example of .txt file that is created from a Dytran 4000A1 measurement is presented in the Figure 4.3.

### 4.1.3 Data processing

Despite the data processing being in essence different for each experiment, we can already make some general remarks about our proceeding with the data collected from the accelerometers.

#### Python

Our appendix section 7.2 lists the major functions that are used for the data processing and while some have been coded for a specific experiment, the global logic behind the process could be summarized as follows :

1. loading and reading the data
  2. calibrating, filter and cleaning the data
  3. making operations on the data and then creating a graphical plot of it
- or

1	Frequency: 1600		
2	X Axis - Acceleration (g)	Y Axis - Acceleration (g)	Z Axis - Acceleration (g)
3	0,488281	0	1,953125
4	0,390625	0,097656	1,855469
5	0,390625	0,195312	1,5625
6	0,292969	0,097656	1,855469
7	0,488281	0,097656	1,855469
8	0,683594	0	1,855469
9	0,292969	0	1,757812
10	0,585937	0	1,757812
11	0,488281	0,097656	1,757812
12	0,488281	0,097656	2,050781
13	0,585937	0	2,050781
14	0,390625	0,097656	1,757812
15	0,488281	0	1,660156
16	0,390625	0,097656	1,757812
17	0,488281	0,097656	1,855469
18	0,292969	0,097656	2,148437
19	0,585937	0,097656	1,660156
20	0,390625	0	1,855469
21	0,390625	0,292969	2,148437
22	0,390625	0	1,953125
23	0,390625	0,097656	2,050781
24	0,390625	0,195312	1,855469
25	0,683594	0	1,757812
26	0,292969	0	1,855469
27	0,292969	0,097656	2,246094
28	0,585937	0	2,050781
29	0,488281	0,097656	1,953125

**Figure 4.3:** Illustration of an arbitrary .txt file generated by a Dytran 4000A1 recording

creating a graphical plot of the data and then taking measurements from it

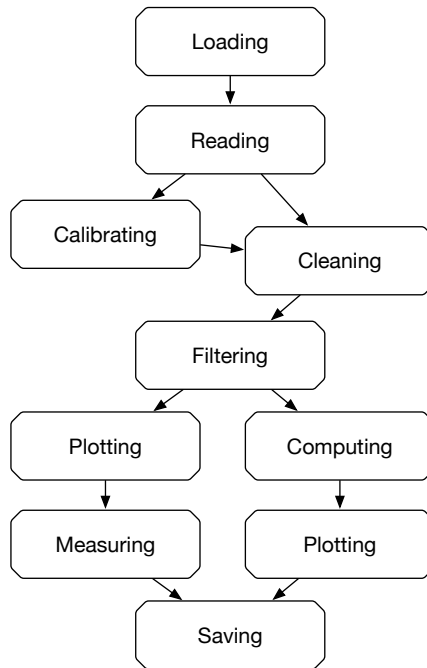
4. saving the graphical plot and the summary informations on the data

The Figure 4.4 summarizes those steps.

### Peak detection

Whenever we used two accelerometers at once, we were intrinsically inducing the technical issue of data synchronisation, seeing that the Dytran 4000A1 doesn't feature an absolute clock and only collects data that from the moment the Record button has been pressed, separating two measurements by a period inverse to the sampling rate and not even explicitly displaying a record of time in the resulting .txt file, as noticeable in the Figure 4.3. For this reason, we introduced a physical method of synchronisation consisting of applying a simultaneous excitation of the sensors through the means of a shock dispensed to both devices at the same time. The result on the data would be a high peak virtually initializing the data collection as an algorithm could then allow us to identify and synchronize both peaks once we decided on how to characterize them. With the peak finding criteria discussed in [14], we manually vary the parameters of our `scipy.signal.find_peaks` Python function in each dataset in order to compute the best fit for the algorithm to detect the peaks from both accelerometers. Once this step completed, we align the data





**Figure 4.4:** Diagram of the general approach for data processing with the Python functions from 7.2

according to the peaks and cut the latter out, keeping only the data measured after them and focus our processing on that.

### Butterworth filter

As we can see from the Figure 4.4, a necessary step concerns the data filtering. In fact, as we can choose the sampling rate of our data collection to be either 1600 or 3200 [Hz], we would respectively be considering phenomena of period  $6.25 \times 10^{-4}$  or  $3.125 \times 10^{-4}$  [s]. Anticipating a bit the upcoming investigations, we know from sources such as [10] that ice block collisions usually happen during a characteristic time scale of around 0.01 [s], which is 16 times longer than the time scales offered by the lowest sampling range. This motivates us to use a low-pass filter to clean the higher noise frequencies. Our choices ended on the Butterworth filter, defined within the *scipy.signal.butter* Python function.

For a transfer function  $H(s)$ , the gain  $G(\omega)$  of such an  $n$ -order Butterworth filter is given by :

$$G^2(\omega) = |H(j\omega)|^2 = \frac{G_0^2}{1 + \left(\frac{j\omega}{j\omega_c}\right)^{2n}}$$

with  $\omega_c$  the cutoff frequency one gives the function as an input and  $G_0$  the gain at zero frequency.

A more detailed mathematical approach to this calculation as well as a detailed derivation of the transfer function can be found in [15].

One can observe that when  $n$  approaches infinity,  $\lim_{\omega \rightarrow \infty} G^2$  becomes a rectangular function where frequencies below  $\omega_c$  will be passed with a gain  $G_0$  while frequencies above  $\omega_c$  will be suppressed. For lower values of  $n$ , the rolloff remains quick but is less sharp and presents no ripples, contrarily to the Chebyshev or Elliptic filters [16]. In general, the Butterworth filter is designed to have a frequency response as flat as possible in the range of frequencies that it processes.

### Optimization

Depending on the experimental situations we investigate, we will sometimes need to perform numerical regressions in order to fit our models for different quantities. One approach is the least squares method that minimizes the sum of the squares of the residuals coming from the results of the different equations of our system, in mathematical situations where there are more equations than unknowns.

The least squares problem are usually solved in the following steps :

1. we let  $\{(t_j, y_j)\}_{j=1}^n$  be the  $n$  pairs of observed covariate and response data.
2. we make the modeling assumption :

$$y_j \approx f(t_j, \beta)$$

where  $\beta \in \mathbb{R}^k$  is the function parameter we will optimize on

3. we finally seek to find the  $\hat{\beta}$  that minimizes the least square distance between  $y = \{y_j\}_{j=1}^n$  and  $\{f(t_j, \beta)\}_{j=1}^n$  :

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^k} \left( \sum_{j=1}^n |y_j - f(t_j, \beta)|^2 \right)$$

Numerically this optimization is performed with the Python function `scipy.optimize.leastsq` and we coded a function that solves the least squares problem based on first guesses that we give in as inputs. A more thorough mathematical treatment of the method, a fixed point iteration on the derivative of  $f$  with respect to  $\beta$ , can be found in [17].

## 4.2 Labwork : added mass investigation

The aim of the experimental investigations of the added mass force is to study the effects that appear from the motions of bodies in the water compared to motions in the air. Letting a pendulum oscillate in the air, measuring the oscillation period and computing the oscillation frequency will provide us with the moment of inertia following our model 3.1.2. A similar procedure in the water will allow us to derivate the added mass effect with respect to our model 3.1.3.

### 4.2.1 Added mass investigation : experimental setup

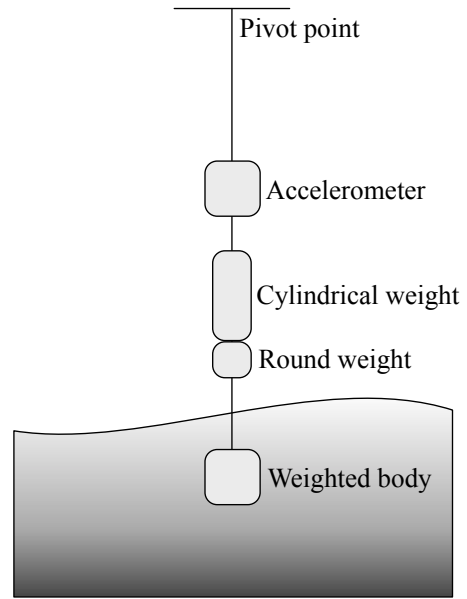
This series of experiments was performed during the week 40 of 2018, with the assistance of Prof. Aleksey Marchenko (UNIS) and Prof. Vladimir Markov (MSU) in the workshop of the University Centre in Svalbard, Longyearbyen (NO). As a matter of fact, I was supervising the students from the master level course *AT-332 Physical Environmental Loads on Arctic Coastal and Offshore Structures* given by Prof. Aleksey Marchenko, guiding them through the necessary theoretical and practical steps to perform the added mass investigations, and the measurements were performed in a joint manner.

A pendulum following the representation of the Figure 4.5 was built and each piece got measured in mass. The Table 4.1 summarizes all items that have been used in the pendulum experiments. The accelerometer had to be attached to the setup with the help of a custom-made support as pictured in Figure 4.6. The right assembly of parts allowed to create a distribution of masses along the rod as displayed in the Figure 4.7.

### 4.2.2 Added mass investigation : protocol

As we are looking to measure the oscillation periods of our pendulum first in the air and then in the water, we mounted the setup onto a support frame that could be placed on top of a water tank if needed. A ruler was aligned with the horizontal axis of the support frame and a camera placed in front of the setup to record the displacement of the pendulum over time. This data, taken and processed by the students, would support the data collected from the accelerometer at the same time.

The pendulum would be dropped from a small angle and left to oscillate for around 5-7 periods before being manually stopped again. Measurements were taken for



**Figure 4.5:** Photograph and representation of the added mass investigation setup

different shapes of bodies (ball, cylinder, disc) in two different continuums (air, water)

### 4.2.3 Theoretical moment of inertia and angular frequency

From the construction of our pendulum, it is possible to compute the theoretical moment of inertia  $I_{S, \text{theo}}$  and with that, the induced theoretical angular frequency  $\omega_{a, \text{theo}}$  for oscillations in the air.

In fact, the setup is represented in the Figure 4.7 with the approximation that all bodies are cylindrical. Remembering the calculation reasoning from (3.6) we can write the torque about the pivot point  $S$  as :

$$\begin{aligned} \vec{\tau}_S &= \vec{r}_{S,cm} \times m_r \vec{g} + \vec{r}_{S,A} \times m_A \vec{g} + \vec{r}_{S,B} \times m_B \vec{g} + \vec{r}_{S,C} \times m_C \vec{g} + \vec{r}_{S,D} \times m_D \vec{g} \\ &= - \left( m_r \frac{l}{2} + m_A |SA| + m_B |SB| + m_C |SC| + m_D |SD| \right) g \sin \theta \hat{k} \end{aligned}$$

where  $|SX|$  is a notation for the distance between the point  $S$  and the point  $X = \{A, B, C, D\}$ .

Category	Item	Mass [g]
Accelerometer	Dytran 4000A1	160
	Dytran with support, bolts and screws	220
Weights	Iron grip	58
	3x Iron grips with screws and bolts	530
	Column	316
	Big ball	358
Bodies	Small ball	258
	Small ball with bolt	260
	Disc	26
	3x Discs = Cylinder	78
Rod	(linear mass [ $g m^{-1}$ ])	168

**Table 4.1:** Summary of the items used in the pendulum experiments

With the argument made for (3.2), we obtain :

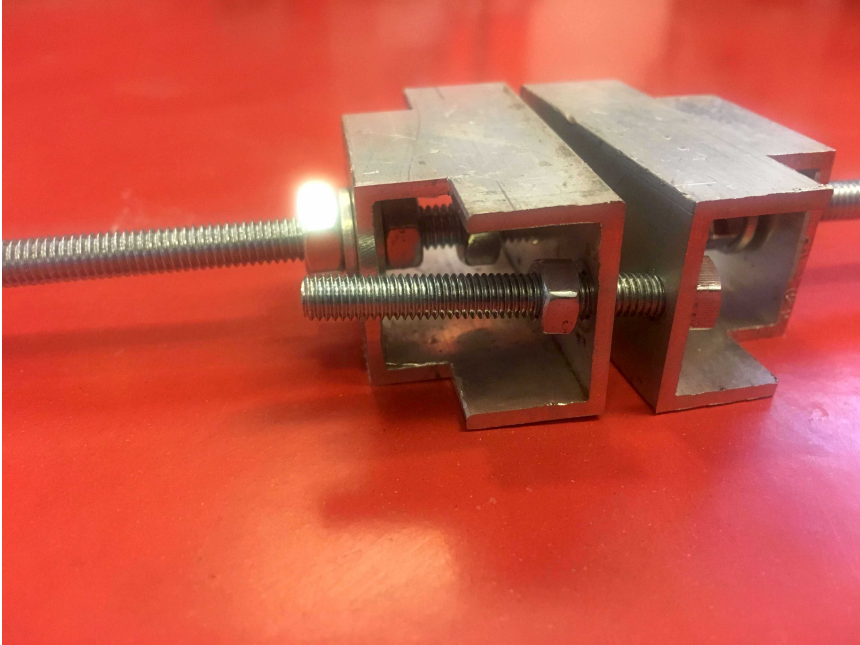
$$\begin{aligned}
 I_{S, \text{theo}} \frac{d^2\theta}{dt^2} &= - \left( m_r \frac{l}{2} + m_A |SA| + m_B |SB| + m_C |SC| + m_D |SD| \right) g \sin \theta \\
 &= - \left( m_r \frac{l}{2} + \sum_{X=\{A,B,C,D\}} m_X |SX| \right) g \sin \theta
 \end{aligned}$$

All we need here is to compute the moment of inertia  $I_{S, \text{theo}}$  and [18] suggests how to sum up the moments using the Huygens-Steiner Theorem 7.1.5 :

$$\begin{aligned}
 I_{S, \text{theo}} &= I_{S,r} + I_{S,A} + I_{S,B} + I_{S,C} + I_{S,D} \\
 &= \frac{1}{3} m_r l^2 + \sum_{X=\{A,B,C,D\}} I_{cm,X} + m_X |SX|^2 \\
 &= \frac{1}{3} m_r l^2 + \sum_{X=\{A,B,C,D\}} \frac{1}{2} m_X R_X^2 + m_X |SX|^2
 \end{aligned}$$

Following the small angle approximation, our rotational equation of motion becomes :

$$\frac{d^2\theta}{dt^2} = - \frac{\left( m_r \frac{l}{2} + \sum_{X=\{A,B,C,D\}} m_X |SX| \right) g}{\frac{1}{3} m_r l^2 + \sum_{X=\{A,B,C,D\}} \frac{1}{2} m_X R_X^2 + m_X |SX|^2} \theta$$



**Figure 4.6:** Photograph of the support of the Dytran 4000A1

This equation describes a simple harmonic motion with an angular frequency  $\omega_{a, \text{theo}}$  that is to be compared to (3.8), which we repeat here for convenience :

$$\omega_{a, \text{theo}} = \sqrt{\frac{\left(m_r \frac{l}{2} + \sum_{X=\{A,B,C,D\}} m_X |SX|\right) g}{\frac{1}{3} m_r l^2 + \sum_{X=\{A,B,C,D\}} \frac{1}{2} m_X R_X^2 + m_X |SX|^2}} \quad (4.1)$$

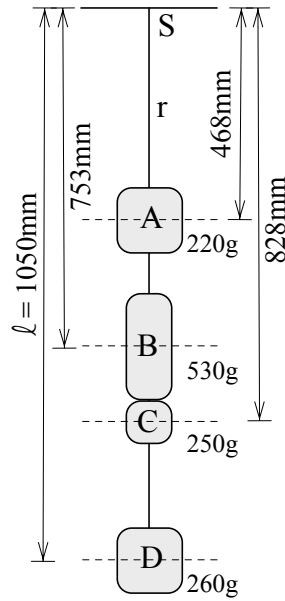
$$\omega_a = \sqrt{\frac{mgl_{cm}}{I_S}}$$

With values collected from 4.7,  $R_A = 0.03$  [m] from 4.1,  $R_B = 0.015$  [m],  $R_C = 0.021$  [m] and  $R_D = 0.021 \pm 0.002$  [m] for all three body shapes, we get :

$$\omega_{a, \text{theo}} = 3.477 \text{ [rad s}^{-1}\text{]}$$

$$T_{a, \text{theo}} = \frac{2\pi}{\omega_{a, \text{theo}}} = 1.807 \text{ [s]} \quad (4.2)$$

and hence, going slightly backwards by reversing (3.8) to find the theoretical mo-



**Figure 4.7:** Representation of the pendulum used for the added mass calculation

ment of inertia considering one has already found the angular velocity like we did in (4.2) :

$$\begin{aligned}
 I_{S, \text{theo}} &= \frac{mgl_{cm}}{\omega_{a, \text{theo}}^2} \\
 &= 0.872 \text{ [kg m}^2\text{]}
 \end{aligned}
 \tag{4.3}$$

With the Figure 4.7 and the linear mass of the rod from 4.1, we can also calculate the position of the centre of mass  $l_{cm}$  :

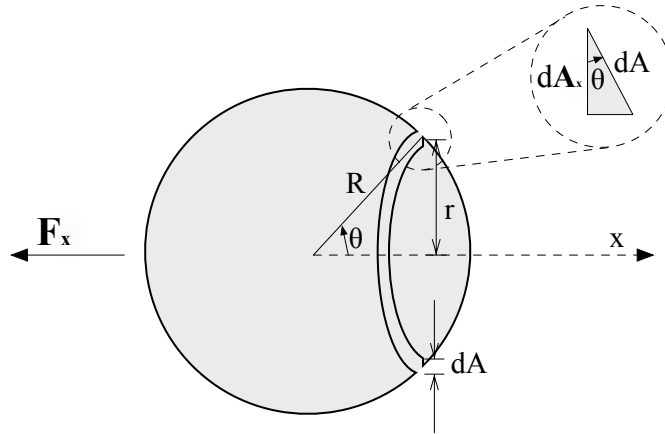
$$\begin{aligned}
 l_{cm} &= \frac{\sum_{X=\{A,B,C,D,\text{rod}\}} m_X |SX|}{\sum_{X=\{A,B,C,D,\text{rod}\}} m_X} \\
 &= 0.748 \text{ [m]}
 \end{aligned}
 \tag{4.4}$$

#### 4.2.4 Theoretical added mass of a sphere and a cylinder

While it was possible to calculate the theoretical moment of inertia and angular frequency of our pendulum, we can also derive theoretical values of added masses

around a sphere and a cylinder. We will approach these calculations by considering the hydrodynamic force acting on it as it accelerates, in a framework where the fluid is inviscid and irrotational, allowing for the use of the unsteady Bernoulli equation (2.18).

### The sphere



**Figure 4.8:** Representation of an accelerating sphere in a fluid

We consider a sphere of radius  $R$  accelerating at a rate  $\frac{\partial u}{\partial t}$  in the  $x$ -direction. The Figure 4.8 illustrates the configuration with  $dA_x = \cos \theta dA$ ,  $dA = 2\pi r R d\theta$  and  $r = R \sin \theta$ .

The hydrodynamic force is to be found by integrating the pressure of the area projected in the  $x$ -direction :

$$\vec{F}_x = \int p d\vec{A}_x \quad (4.5)$$

with  $p = -\rho \left[ \frac{\partial \phi}{\partial t} + \frac{1}{2} |\nabla \phi|^2 \right]$  and  $\phi = \frac{\partial u}{\partial t} \cos \theta \frac{R}{2}$  for an axisymmetric flow around a sphere [19], which leads to :



$$\begin{aligned}
 F_x &= \int_0^\pi \left[ -\rho \left[ \frac{\partial \phi}{\partial t} + \frac{1}{2} |\nabla \phi|^2 \right] \right] 2\pi R^2 \cos \theta \sin \theta d\theta \\
 &= -\rho 2\pi R^2 \frac{\partial u}{\partial t} \frac{R}{2} \underbrace{\int_0^\pi \sin \theta \cos^2 \theta d\theta}_{=2/3} - \rho 2\pi R^2 \frac{u^2}{2} \underbrace{\int_0^\pi (\sin \theta \cos^3 \theta + \frac{1}{4} \sin^3 \theta \cos \theta) d\theta}_{=0} \\
 &= -\frac{2}{3} \rho \pi R^3 \frac{\partial u}{\partial t}
 \end{aligned}$$

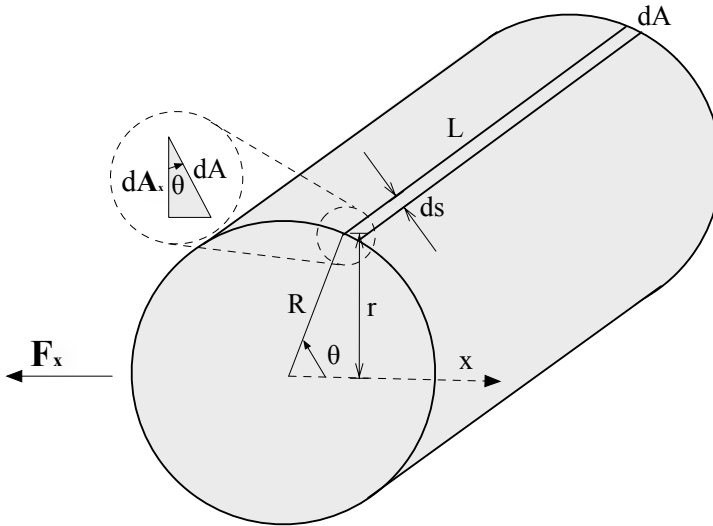
The negative sign here indicates that the force is opposing the acceleration and thus, the body would have to exert this extra force when moving in the  $x$ -direction, which makes the added mass for a sphere of radius  $R$  be :

$$m_{\text{add, sphere}} = \frac{2}{3} \rho \pi R^3 \tag{4.6}$$

Our case, with  $\rho_w = 1000 \text{ [kg m}^{-3}\text{]}$  and  $R_{\text{sphere}} = 0.021 \text{ [m]}$ , yields :

$$m_{\text{add, sphere}} = 19.4 \text{ [g]} \tag{4.7}$$

**The cylinder**



**Figure 4.9:** Representation of an accelerating cylinder in a fluid

Following a similar thought process, we consider a cylinder of radius  $R$  and length  $L$ , accelerating at a rate  $\frac{\partial u}{\partial t}$  in the  $x$ -direction. An illustration of the configuration can be found in the Figure 4.9 with  $dA_x = \cos \theta dA$ ,  $dA = L ds$  and  $ds = R d\theta$ .

We still invoke the the hydrodynamic force (4.5) with the same expression for the pressure coming from the unsteady Bernoulli equation, but this time with  $\phi = u \frac{R^2}{r} \cos \theta$  for an axisymmetric flow around a cylinder [19], and thus :

$$\begin{aligned} F_x &= \int_0^{2\pi} \left[ -\rho \left[ \frac{\partial \phi}{\partial t} + \frac{1}{2} |\nabla \phi|^2 \right] \right] RL \cos \theta d\theta \\ &= -\rho RL \frac{\partial u}{\partial t} R \underbrace{\int_0^{2\pi} \cos^2 \theta d\theta}_{=\pi} - \rho RL \frac{u^2}{2} \underbrace{\int_0^{2\pi} \cos \theta d\theta}_{=0} \\ &= -\rho \pi R^2 L \frac{\partial u}{\partial t} \end{aligned}$$

With the same comment on the sign of the force, we can extract the added mass for a cylinder of radius  $R$  and length  $L$  :

$$m_{\text{add, cylinder}} = \rho \pi R^2 L \quad (4.8)$$

Our case, with  $\rho_w = 1000 \text{ [kg m}^{-3}\text{]}$ ,  $R_{\text{cylinder}} = 0.025 \text{ [m]}$  and  $L_{\text{cylinder}} = 0.047 \text{ [m]}$ , yields :

$$m_{\text{add, cylinder}} = 92.3 \text{ [g]} \quad (4.9)$$

For a disc, the only difference is  $L_{\text{disc}} = 0.016 \text{ [m]}$  and we therefore obtain :

$$m_{\text{add, disc}} = 31.4 \text{ [g]} \quad (4.10)$$

Both the computed values (4.6) and (4.8) are in accordance with the theoretical added mass tables presented by [20].

Once the added masses have been obtained, a reversal of (3.10) allows us to find an expression for the theoretical angular velocity in the water :

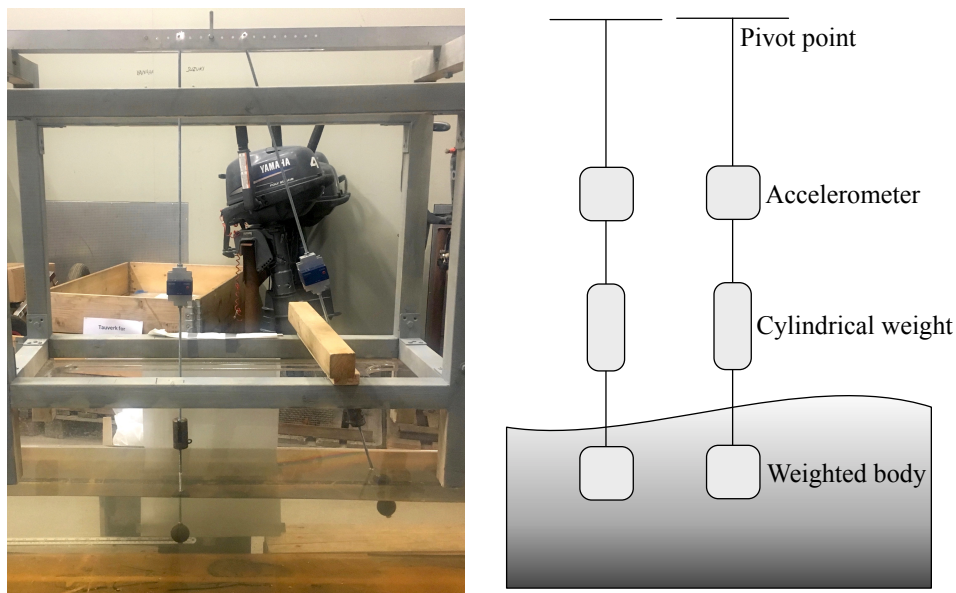
$$\omega_{w, \text{theo}} = \sqrt{\frac{mgl_{cm}}{I_{S, \text{theo}} + m_{\text{add, theo}} l^2}} \quad (4.11)$$

### 4.3 Labwork : collision pressure force

The aim of the experimental investigations of the collision pressure force is to extract the characteristic times of collision for interaction in different mediums (air, water and water surface) as well as the respective accelerations that characterize them. Integrating this acceleration over the collision times will provide us with data on the velocities before and after the interactions. Using the equation developed in the model 3.2, we will be able to present values for the collision pressure forces at play.

#### 4.3.1 Collision pressure force : experimental setup

This series of experiments was performed during the week 40 of 2018, with the assistance of Prof. Aleksey Marchenko (UNIS) and Prof. Vladimir Markov (MSU) in the workshop of the University Centre in Svalbard, Longyearbyen (NO). The work consisted of the right assembly of the two pendulums and the building of a correct setup allowing for the best collisions between the two devices. Many tests, corrections of the setup, modifications of the hardware and thoughts were had about the optimal parameters to achieve the best performances in the experiments.



**Figure 4.10:** Photograph and representation of the collision pressure investigation setup

Two pendulums following the representation of the Figure 4.10 were mounted after each piece got measured in mass, with the references of the Table 4.1. The

distances between each piece respect the indications provided by the Figure 4.11.

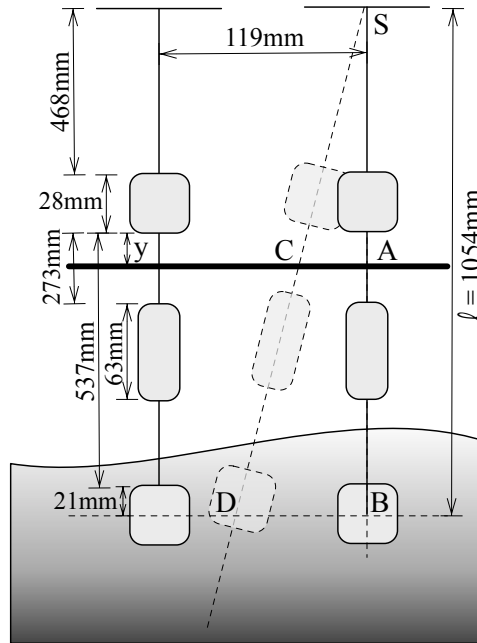
### 4.3.2 Collision pressure force : protocol

This investigation will process data from two different sources recording the collisions between the two pendulums ; on one hand, we will record the interactions between the two bodies with a camera. For that sake, a ruler is placed between the accelerometer and the weight on the frame so that we get a length reference when treating the video files. This data gives us information on the instantaneous velocities of the bodies before and after the impact, as well as a computed instantaneous acceleration of the bodies during the impact, which we will be using for an estimation of the collision pressure force. On the other hand, we measure the accelerations from each pendulum through the Dytran 4000A1 installed on them and will be using this data to characterize the collision times in the different continuums they occur. As we can also obtain the time between motions, when the right pendulum hits the left one, the time it takes for the second one to be set in motion, we can try to comment on the different values we get from the different setups.

In most iterations, only the pendulum on the right would be dropped into the pendulum on the left, initiated through a drop from a small angle and manually stopped right after the collision. Measurements were taken for different shapes of bodies (ball, cylinder, disc) in three different continuums (air, water and water surface).

### 4.3.3 Thales theorem and extrapolation

In this series of experiments, the camera came to be placed in two different positions : in some instances, it was placed at the same height as the bodies, allowing us to follow their displacement frame by frame directly. In other cases, it was placed above the setup, making us see down upon the markings of the ruler while the pendulums would be in motion, like one can notice on the Figure 4.12. As it is the effective motion of the bodies we're most interested in when recovering data from the video files, we have to find a design to extrapolate the variable distance between, using the notation from the Figure 4.11, a pre-measured rest-position  $B$  and an instantaneous position  $D$  from the readable variable distance between another rest-position  $A$  and the dynamic points  $C$ . By making the approximation that  $D$  remains the centre of the body initially at  $B$  and that  $C$  is the correct rotational projection of  $A$  on to the ruler as the pendulum moves, we create two similar triangles sharing the common edge  $S$  and where  $AC \parallel BD$ . By virtue of the Thales or Intercept theorem :



**Figure 4.11:** Representation of the pendulums used for the collision pressure calculations

$$\frac{|SA|}{|SB|} = \frac{|SC|}{|SD|} = \frac{|AC|}{|BD|} \implies |BD| = |AC| \cdot \frac{|SB|}{|SA|} \quad (4.12)$$

where  $|SA| = 456 + y$  [mm] can be extracted from the video analysis and  $|SB| = 1054$  [mm] can be measured, as reported on the Figure 4.11.

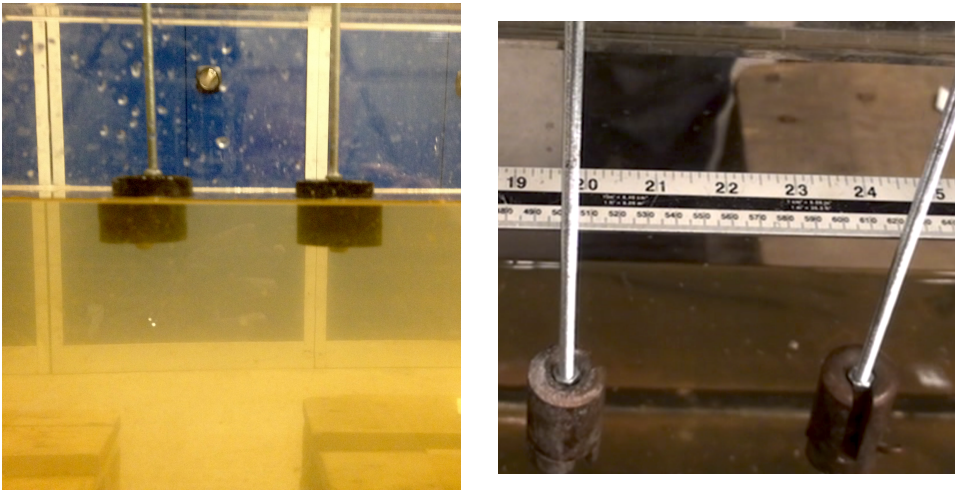
#### 4.3.4 Tracker and video analysis

The video analysis was done using the video analysis and modeling tool Tracker created by Douglas Brown and built on the Open Source Physics Java framework. While Tracker features the tracking of positions and can induce velocities and accelerations, it also offers calibration and perspective adjustments that reveal crucial to our different experiments.

##### Manipulating the software

In fact, before the extraction of data, a number of parameters have to be tuned in order to fit the physical configuration of the setup :

1. the video file has to be loaded and the perspective should be corrected right away. With the help of the perspective filter, an arbitrary quadrilateral, usu-



**Figure 4.12:** Photographs of the camera angles when recording from the bottom (left) and from the top (right) in the collision pressure force experiments

ally a right or obtuse trapezoidal, which edges have to be placed on two parallel lines from the physical setup on the frame, are being reshaped into a rectangle, correcting the optical distortion arisen from the camera angle. In other words, it maps a distorted plane shape in the input image to a straight-on rectangular plane shape in the output image

2. we introduce a calibration scale correlating a distance on the frame to a physical distance. For this, we usually either use markings on the ruler or the distance between the two pendulums, 119 [mm] as indicated in the Figure 4.11
3. we set a reference frame origin, as well as the angle of the axes for the measurements ; this part determines the correct reading of the positions and induced velocities and accelerations as they are given in this reference frame
4. the video clip is identified by two frame boundaries. This is done by browsing through all the frames and selecting two adequate limits, usually separated by 10-20 frames, in the case of a collision here. Tracker also automatically identifies the frame rate, usually 25 [Hz] and therefore knows the elapsed time between two frames, 0.04 [s] for two successive ones
5. we can start the tracking of the positions of our objects frame by frame. The desired resulting quantities have to be selected and will be displayed in a table next to the working frame.

A graphical summary of the steps is to be seen in the Figure 4.18. Such a procedure is repeated for each new video file.

### Finite difference algorithms

The velocities and accelerations in Tracker are calculated from the position-time data using the following finite difference algorithm [21] :

$$v_i = \frac{x_{i+1} - x_{i-1}}{2\Delta t}$$
$$a_i = \frac{2x_{i+2} - x_{i+1} - 2x_i - x_{i-1} + 2x_{i-2}}{7\Delta t^2}$$

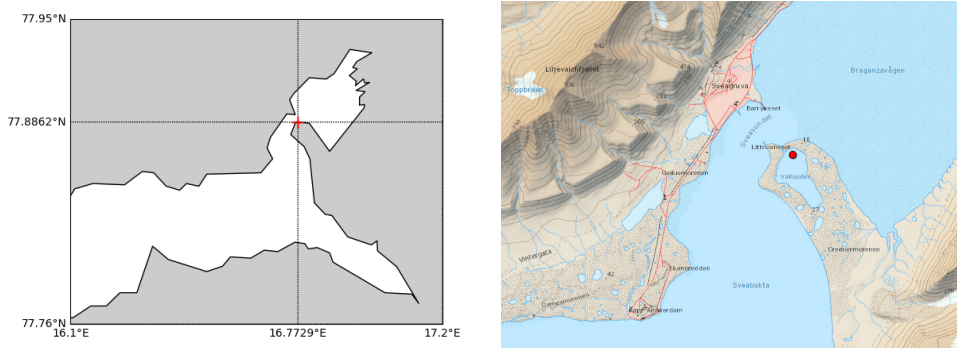
where the subscripts refer to the step numbers and  $\Delta t = t_{i+1} - t_i$  is independent from  $i$  and represents the time between two successive steps. The first discrete derivative is called a centered derivative and it has the odd property of missing the  $x_i$  term for the calculation of the  $v_i$  velocity while being an average over a 2-step interval. Although it pertains the advantage of being less sensitive to position uncertainties than other models, this characteristic presents two slight disadvantages for us : in our case of collision, it would be including the term after the collision already, which is irrelevant when we want to compute the instantaneous velocity before the contact. It also diffuses the information on the behaviour of the derivative function laterally, meaning that such a derivative would be slightly more flat than those offered by other finite difference algorithms [4]. The acceleration is a weighted average over a 4-step interval and represents the second derivative of a parabolic fit, with the step at the center.

We therefore prefer using a backward finite difference algorithm for both the velocity and the acceleration in order to avoid the effects of the collision in our results. The backward derivatives are defined as :

$$v_i = \frac{x_i - x_{i-1}}{\Delta t}$$
$$a_i = \frac{v_i - v_{i-1}}{\Delta t} \tag{4.13}$$

## 4.4 Fieldwork : experiments on Valluden facing Sveagruga

The fieldwork experiments were performed on a frozen water mass with distinctive characteristics. In fact, Valluden is a fresh water lake so low in altitude that it receives sea water from Sveabukta, and the Van Mijenfjord in which it remains,



**Figure 4.13:** Coordinates and position of the fieldwork camp on Vallunden facing Sveagruva (Svalbard, Norway)

during tidal events through a narrow channel located in the North-West. As such, the water mass also presents salt water and the sea ice that forms in Valluden displays the columnar structural characteristics of sea ice [22]. Furthermore, its sea ice remains sheltered from the wind and in the area where the work was performed, the channel's flow doesn't affect the current below the ice [23]. We therefore have an optimized setup for sea ice experiments, combining thick and sturdy ice as well as little environmental perturbations.



**Figure 4.14:** Photograph of the expedition camp on Vallunden

The fieldwork expedition lasted for two weeks during the weeks 10 and 11 of 2019. The logistics were taken care of by Prof. Aleksey Marchenko (UNIS) and Kaare Johansen (UNIS) and while the project was encompassing a large variety of experiments with professors and researchers from all partner institutions of the *Arctic Offshore and Coastal Engineering in changing Climate AOCEC* project, I participated in a greater number of tests and investigations — standard and extended cantilever beams, standard and fixed-ends compressive strength, tensile



and shear strength, indentation before and after a vibro-plate action, stress relaxation, horizontally oscillating floe turbulence, and of course, ice block interactions as well as ice cube buoyancy oscillations —, be it to prepare the location, cut the sea ice, install the gear and equipment, take measurements, collect data, discuss with the researchers, clean the site or learn first-hand from very experienced experts. The temperatures over the two weeks were comprised between daily averages of  $-26.6$  [ $^{\circ}\text{C}$ ] and  $-17.6$  [ $^{\circ}\text{C}$ ], with extrema of  $-33.8$  [ $^{\circ}\text{C}$ ] and  $-11.1$  [ $^{\circ}\text{C}$ ], whereas the winds blew between daily averages of  $1.3$  [ $\text{m s}^{-1}$ ] and  $9.37$  [ $\text{m s}^{-1}$ ], with extrema above  $11.0$  [ $\text{m s}^{-1}$ ] over several days [[https://www.yr.no/place/Norway/Svalbard/Sveagruva/detailed\\_statistics.html](https://www.yr.no/place/Norway/Svalbard/Sveagruva/detailed_statistics.html)]. These conditions made work either more strenuous, when it was still possible, or impossible because of cold wind storms. The Figure 4.14 displays the camp as it got set up on the 5th of March 2019.

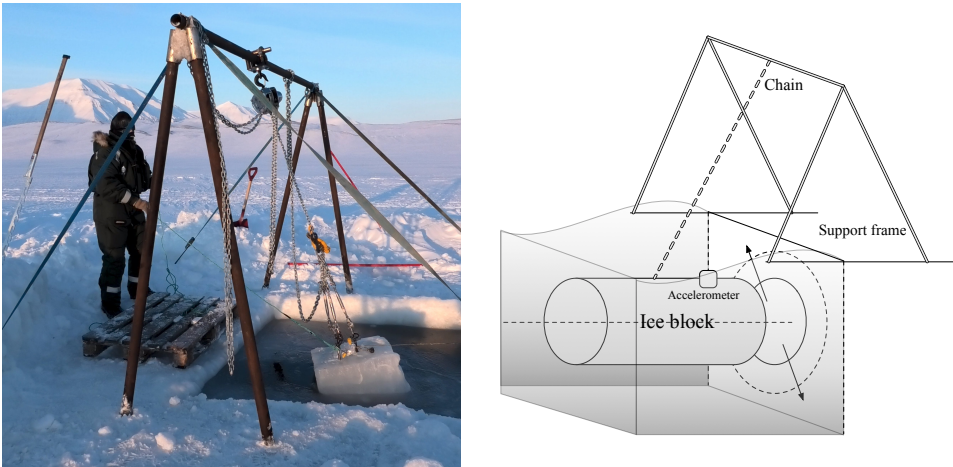
## 4.5 Fieldwork : Ice block interactions

A bit similarly to the laboratory pendulum experiment on the collision pressure force, the aim of the experimental work on the ice block interactions is to extract the characteristic times for ice-ice collisions in the water and in the air. We will also use data on the velocities from video files in order to verify the mathematical predictions of our model 3.3 about the horizontal velocity of the block when approaching the wall in fluid.

### 4.5.1 Ice block interactions : experimental setup

This series of experiments was performed during the week 11 of 2019, with the assistance of Prof. Aleksey Marchenko (UNIS) and Prof. Vladimir Markov (MSU). To realize this montage, we needed a basin of open water on top of which we would assemble a support frame to which we could then mount the lifting system and chain to create an ice block pendulum. A water basin of roughly  $2 \times 5$  [m] was cut out of the sea ice, in synergy with the site for the horizontal ice floe turbulence, large enough to avoid too quick refreezing of the water on the surface. A permanent maintenance was however needed to preserve the liquid water in the basin and ice formations needed to be constantly detached from the edges of the basin and fished out. The ice blocks for the experiment needing a clear cut, they required to be shaped with a chain saw and moved in and out of the water with care.

The ice block that was to be set in motion was fixed to the support frame chain through a triple anchor point fixed to the ice with ice screws. The accelerometer would be screwed straight into the ice as well and the setup is illustrated in the Figure 4.15.



**Figure 4.15:** Photograph and representation of the pendulum used for the ice block experiments

## 4.5.2 Ice block interactions : protocol

The data acquisition would both come from the accelerometer fixed to the ice block set to interact with an ice wall, in the water and outside of the water, and from video files taken with a handheld camera during the runs. The motion would be induced from a preliminary pull-back of the block in the direction opposite to the wall, and the release of that force would have the ice block pendulum oscillate back towards its resting position, colliding with the wall past its vertical point.

While the characterization of the collision time repeats the protocol of the laboratory pendulum and its collision pressure force, the analysis of the horizontal velocity of the ice block will motivate a comparison between the experimental values obtained through the video files and a theoretical computation of the model value with the right parameters.

## 4.5.3 3D modeling and ice block dimensions

While visiting the field work site during the week 11, Dr. Nataly Marchenko (UNIS) took a series of high-resolution pictures of the ice blocks and created a 3D model for the shapes using the Agisoft Metashape software. The result was compiled into a 3D-file that is presented in the Figure 4.19. From the file, using a reference calibration square of side 20 [cm] and the built-in 3D-distances measuring tool provided by Adobe Acrobat Reader DC, we could measure and infer the dimensions of the ice block used in the collisions. This data can in turn be used in

Tracker in order to calibrate the video frame during the interactions of the blocks with the wall.

Dimension	Measurement
Calibration : 20 [cm]	2.427
Length	3.292
Width	2.913
Height	2.702

**Table 4.2:** Measurements for the calibration of the dimensions of the ice block used in the ice block experiments

The results from the distance measurements feature in the Table 4.2. From there, it is straight forward to extract the values :

$$\begin{aligned} \text{Length} &: 27.1 \text{ [cm]} \\ \text{Width} &: 24 \text{ [cm]} \\ \text{Height} &: 22.3 \text{ [cm]} \end{aligned} \tag{4.14}$$

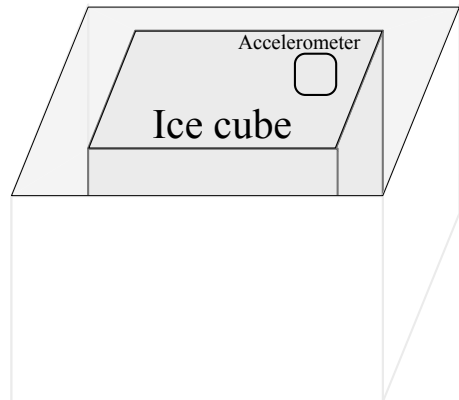
## 4.6 Fieldwork : Ice cube buoyancy oscillations

The objective of the ice cube buoyancy oscillations is to observe the exponential decrease of the velocity amplitude announced by our model 3.4.2 and to find a value for the characteristic time of damping according to our experimental parameters. We will thus compare our field value of the vertical velocities with the theoretical formula we derived.

### 4.6.1 Ice cube buoyancy oscillations : experimental setup

This series of experiments was performed during the week 11 of 2019, partially also introducing it to the visiting students from the bachelor level course *AT-221 Ice Mechanics, Loads on Structures and Instrumentation* given by Prof. Aleksey Marchenko, taking measurements with them both helping maintaining the basin and setting the ice cube in motion. The said basin of open water this time only needed to surround the floating ice cube with little open water width. I therefore cut open a 12 [cm] thick canal around what was to become a buoyant ice cube of side 111 [cm], as illustrated in the Figure 4.17.

The accelerometer was screwed onto the ice cube and at the cost of regular maintenance to free the surrounding canal from forming ice, the cube would remain



**Figure 4.16:** Photograph and representation of the ice cube used in the buoyancy oscillations experiments

buoyant and straight by itself. The setup is captured and illustrated in the Figure 4.17.

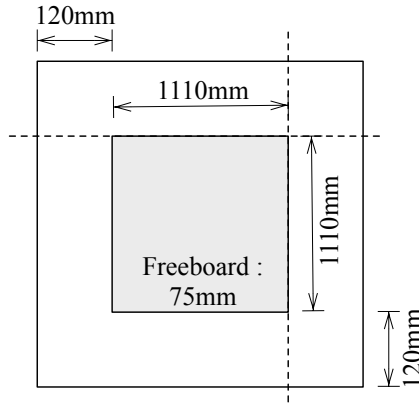
#### 4.6.2 Ice cube buoyancy oscillations : protocol

The record of data comes from the accelerometer fixed to the block and the latter had to be carefully packed and dealt with to avoid too intensive contact with the water, splashes and tilts of the block periodically drowning the surface of the ice cube during its oscillations. The accelerometer was left to run for a few oscillations to insure having enough data to observe the long-time behaviours as well.

The block was to be set in motion with an impulsion as vertical as possible and different attempts were made : either by two people, each with a stick in their hands and respectively pressing down on opposite sides of the cube simultaneously, or by one single person first jumping onto the ice block, forcing it down into the water, and releasing their weight by jumping off from it with care so that the cube rises back straight up. In general, this given attention only influences the first few oscillations of the block, the further ones seeing the block balance itself back to quasi-vertical oscillations.

#### 4.6.3 Freeboard and total mass of the ice cube

In this experiment, even though we had cut the ice ourselves, discovering the ice thickness through a direct physical measurement revealed technically hard. Instead, a measurement of the freeboard would help us retrieve the information indirectly. Borrowing the formula [24] uses to compute the ice density  $\rho_i$  as a function



**Figure 4.17:** Representation of the buoyant ice cube in its basin and freeboard

of the water density  $\rho_w$ , the freeboard  $F_i$  and the ice thickness  $h_i$ , we can isolate the ice thickness and write :

$$\rho_i = \rho_w - \frac{\rho_w F_i}{h_i} \implies h_i = \frac{\rho_w F_i}{\rho_w - \rho_i} \quad (4.15)$$

For a freeboard of  $F_i = 0.075$  [m] as shown on the Figure 4.17 and with  $\rho_w = 1025$  [kg m<sup>-3</sup>] and  $\rho_i = 920$  [kg m<sup>-3</sup>] , we get an ice thickness of :

$$h_i = 0.73 \text{ [m]} \quad (4.16)$$

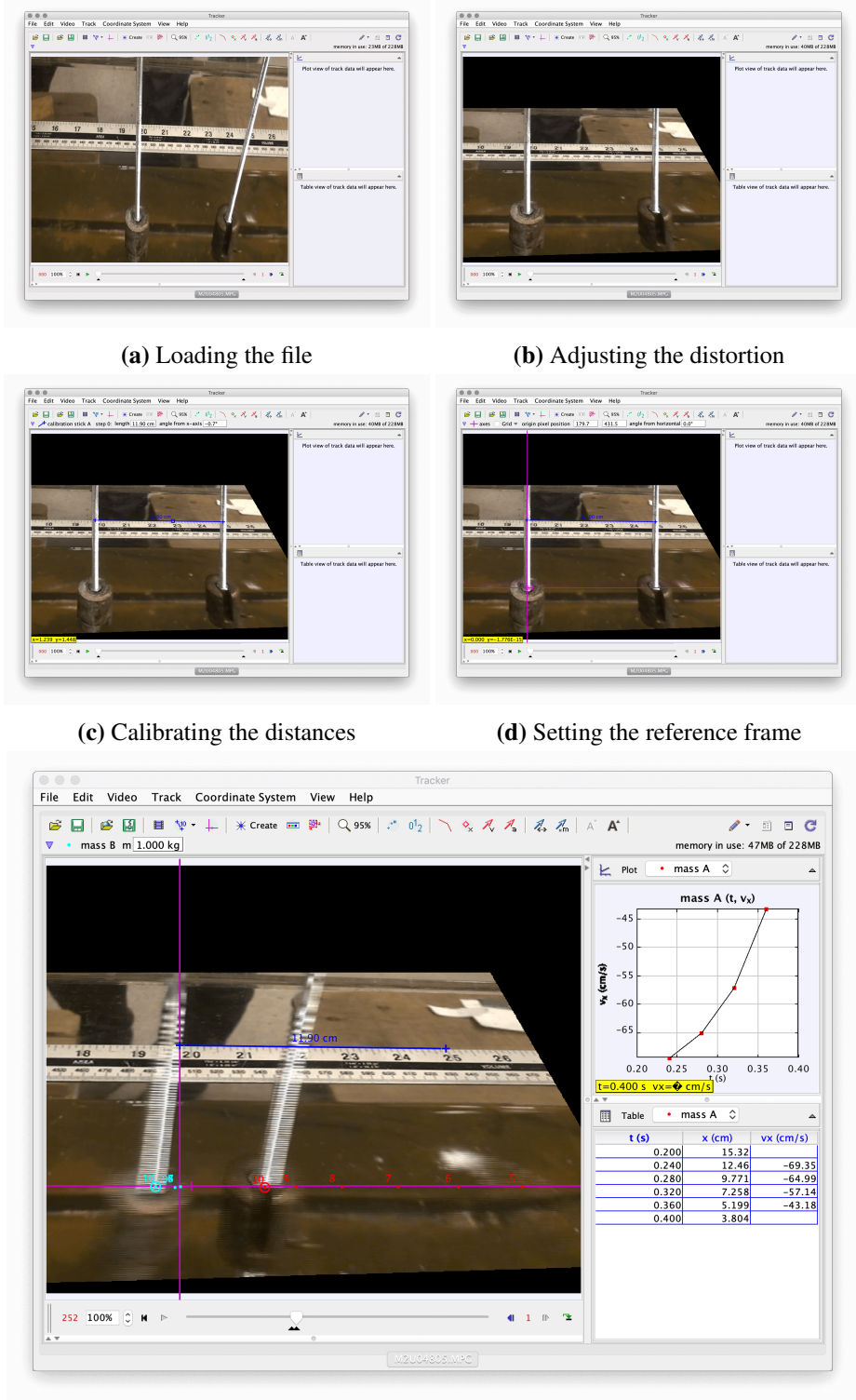
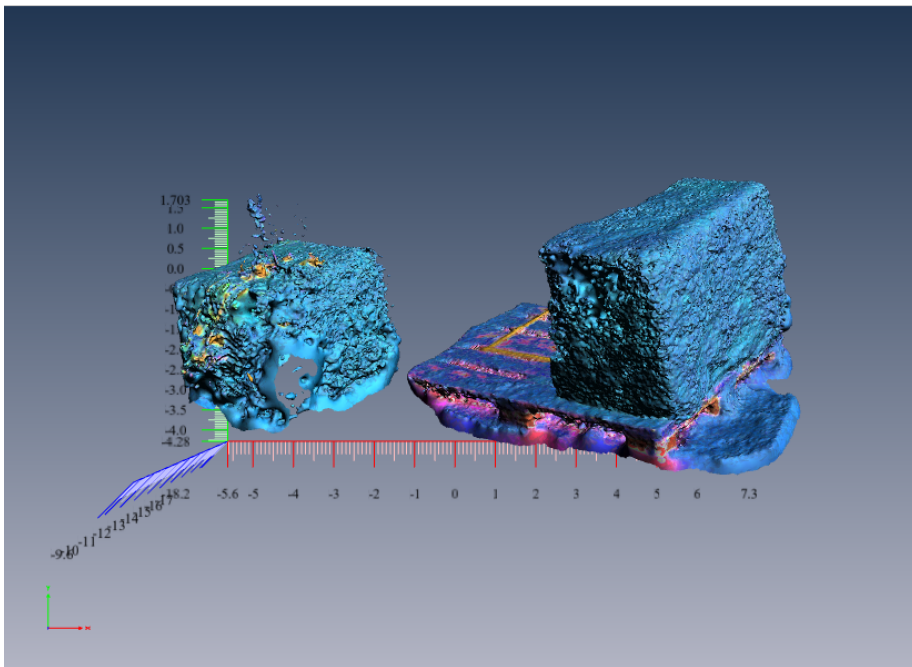
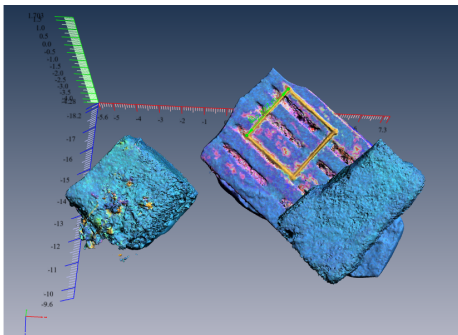


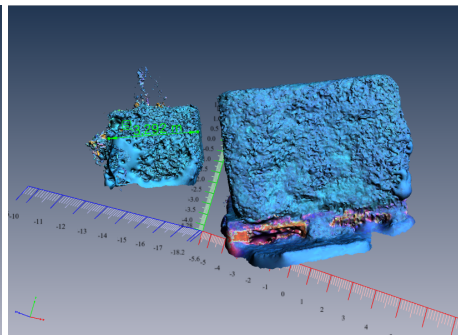
Figure 4.18: Illustration of the Tracker software video file manipulation



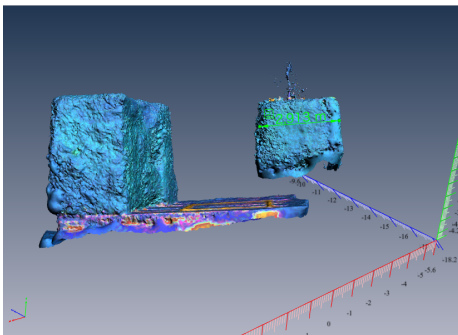
(a) Global view of the ice blocks



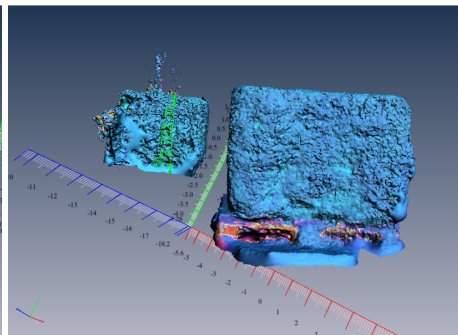
(b) Calibration



(c) Dimension 1



(d) Dimension 2



(e) Dimension 3

**Figure 4.19:** Illustration of the 3D model of the ice blocks and measurements

# Chapter 5

## Data analysis and discussion

"The good news about computers is that they do what you tell them to do. The bad news is that they do what you tell them to do."

---

Ted Nelson

### 5.1 Labwork : added mass investigation

We obtained datasets from the accelerometers for the three shapes of the bodies (ball, cylinder, disc) and the two continuums (air, water). The graphs of each dataset were obtained and, after identifying the axis of oscillation on the accelerometer, we ran a Butterworth filter on the data of that specific axis, with the approximation that the oscillations were happening within a small angle of oscillation and in an uniaxial motion.

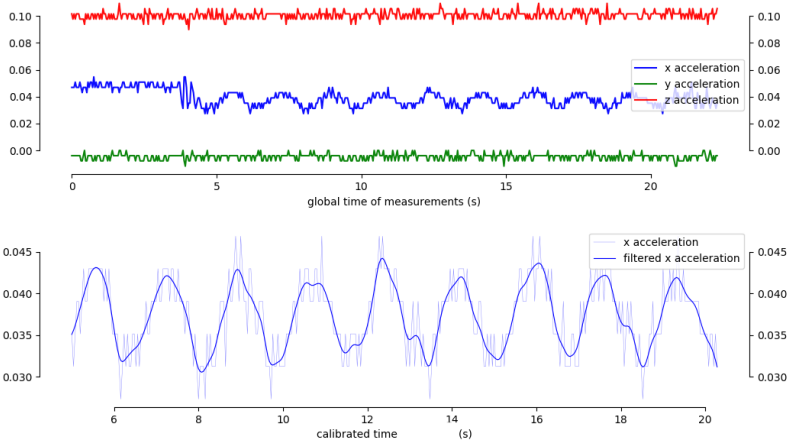
#### 5.1.1 Oscillations in the air

The time series of the bodies in the air all have in common a very slow, if not unnoticeable, decrease of their oscillations amplitudes. An example with the ball is given in the Figure [5.1](#).

In order to fit the filtered acceleration in the  $x$ -axis, we will use the following model  $\Xi(t)$  where the parameters are described by the different  $\xi_i$  :

$$\Xi(t) = \xi_1 \sin(\xi_2 t + \xi_3) + \xi_4 \tag{5.1}$$





**Figure 5.1:** Time series of the accelerations in the added mass investigation with the ball in the air

in which  $\xi_1$  represents the amplitude of the oscillations,  $\xi_2$  represents the angular frequency,  $\xi_3$  represents the phase and  $\xi_4$  represents the offset of the sinusoidal function.

An optimization algorithm using the model (5.1) allows us to create the Figure 5.2

### 5.1.2 Oscillations in the water

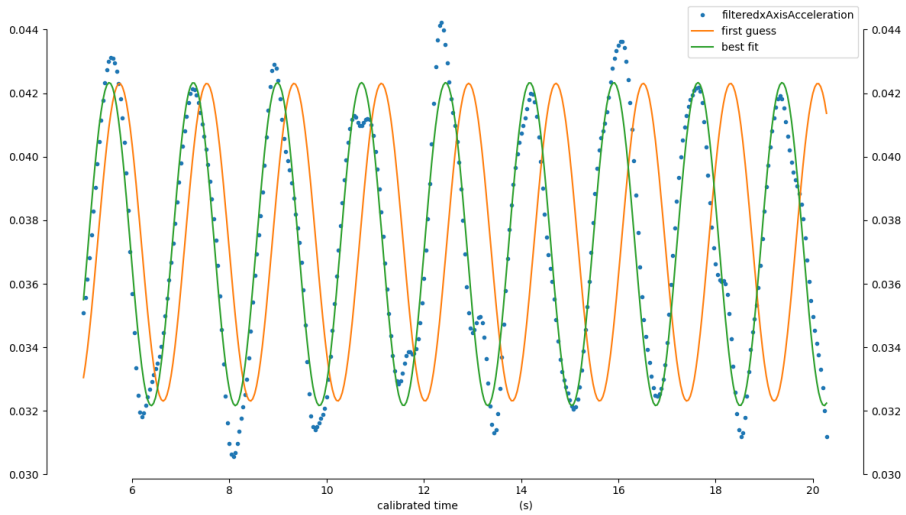
The time series of the bodies in the water in return all feature a noticeable decrease of their oscillations amplitudes denoting damping by and through the fluid. An example with the disc is given in the Figure 5.3.

In order to fit the filtered acceleration in the  $x$ -axis, we will use the following model  $\Xi(t)$  where the parameters are described by the different  $\xi_i$  :

$$\Xi(t) = \frac{\xi_1}{t^{\xi_2}} \sin(\xi_3 t + \xi_4) + \xi_5 \quad (5.2)$$

in which  $\xi_1$  and  $\xi_2$  influence the amplitude of the oscillations,  $\xi_3$  represents the angular frequency,  $\xi_4$  represents the phase and  $\xi_5$  represents the offset of the sinusoidal function.

An optimization algorithm using the model (5.2) allows us to create the Figure 5.4



**Figure 5.2:** Time series of the regression in the added mass investigation with the ball in the air

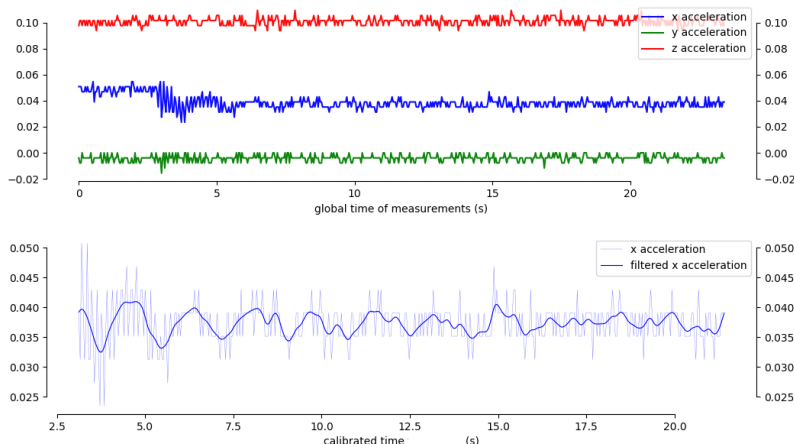
### 5.1.3 Results

A reading from the angular frequency outputs of the optimization functions leading to the Figures 5.2 and 5.4 as well as the use of the formulas (4.3), (4.4), (4.7), (4.9), (4.10) and (4.11) lead to the creation of the Table 5.1.

	$\omega_{a, \text{theo}}$	$\omega_a$	$\omega_{w, \text{theo}}$	$\omega_w$	$I_S$	$m_{\text{add, theo}}$	$m_{\text{add}}$
Units		$[\text{rad s}^{-1}]$			$[\text{kg m}^2]$	$[\text{g}]$	
Ball	3.477	3.633	3.435	3.558	0.799	19.4	30.4
Cylinder	3.477	3.745	3.290	3.456	0.751	92.3	119
Disc	3.477	3.776	3.410	3.571	0.740	31.4	78.7

**Table 5.1:** Summary of the results obtained from the added mass investigations

From comparison with [25] running a similar experiment with a cylinder only and identifying the frequencies visually from a video recording of the oscillations of the pendulum, we note similar values for the angular frequencies, both theoretical and experimental. We notice from 5.1 that the values of the angular frequencies are systematically lower in the water than in the air, respectively in theory and in practice, a direct consequence of the added mass effect which intervenes in a force opposing the direction of motion of the pendulum in water. Between the theory and the experiments however, the laboratory values for the angular velocities

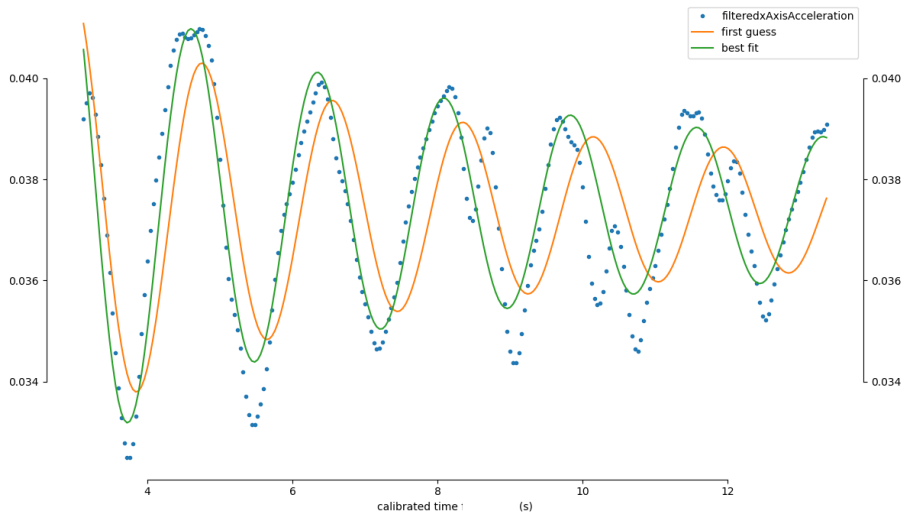


**Figure 5.3:** Time series of the accelerations in the added mass investigation with the disc in the water

are slightly larger than those we computed beforehand and this could be due to a number of parameters acting beyond our model like the drag forces we neglected, dissipation through friction in the dynamics of the pendulum or more surfaces to be taken into consideration for the added mass effect. This latter hypothesis is supported by larger experimental values for the added masses of each of the three bodies compared to their theoretical calculations. While the respective ratios between the two values differ largely for a body to another, we would motivate further iterations of the experiments to lower the propagated error from more measurements.

## 5.2 Labwork : collision pressure force

We obtained multiple datasets from the accelerometers for the three shapes of the bodies (ball, cylinder, disc) and the three mediums (air, water and water surface). The graphs of each dataset were obtained and, after identifying the calibration peaks, the axis of oscillation on each accelerometers, we ran a Butterworth filter on the data of those specific axes, with the approximation that the oscillations were happening within a small angle of oscillation and in an uniaxial motion as before. In a last graph within each series, we superposed the filtered data of both accelerometers in order to identify the moments of collision at the same time as we could note the physical response time from the still-standing accelerometer to the collision induced by the dynamic accelerometer. An example of one of the test from the disc in the air can be found in the Figure 5.5.



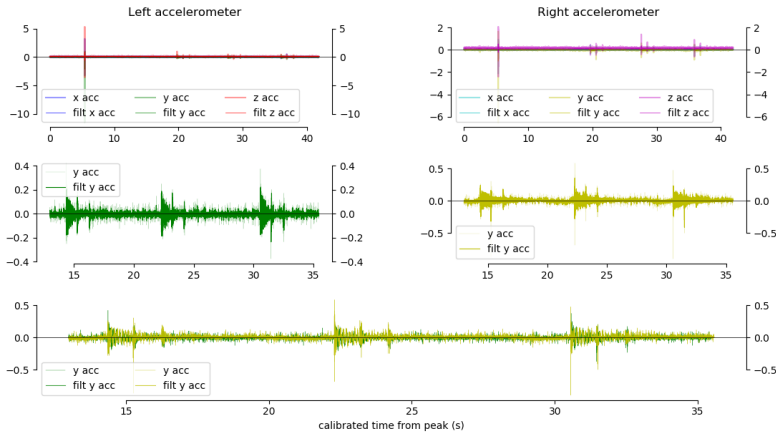
**Figure 5.4:** Time series of the regression in the added mass investigation with the disc in the water

### 5.2.1 Dytran and accelerations

From combined and total data plots such as the one presented in the Figure 5.5, we visually identify collision events characterized by a sudden peak in oscillation amplitude and zoom into them manually in order to access the time stamp at which they occur. We look for the first total sinusoidal oscillation of the data points and take the period of the oscillation as the characteristic time for the collision. An example of a zoom with visible collision oscillations features in the Figure 5.6. Although the color code from the plots above helps differentiating each devices (the yellow curves come from the right accelerometer whereas the green curves belong to the left accelerometer, which makes apparent from the chronological order of response of the accelerometers that it is the right device that was set in motion and collided with the static left device), experiments were run arbitrarily with either the left or the right accelerometer initiating the collision, which makes a distinction between right and left irrelevant here. We will therefore refer to the accelerometers as dynamic and static, when required.

### 5.2.2 Tracker and velocities

We observed 48 different collisions events from video files that we could link to the corresponding accelerometer files by the means of a systematically run archive in our note book at the time of the laboratory works. Similarly to the accelerometer data, these occurrences contain multiple files from interactions of a same pendu-



**Figure 5.5:** Time series of the accelerations in the collision pressure force experiment with the disc in the air

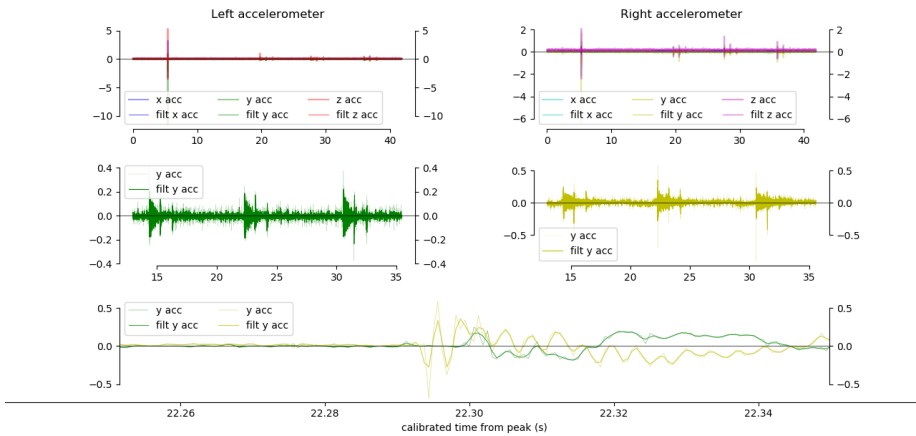
lum in the same setup, which allows us to gain in precision when analyzing and ultimately average certain data either over the shape of the bodies or the medium of their interaction. The collision that the Figure 5.6 focuses on is being illustrated frame by frame in the Figure 5.12.

### 5.2.3 Results

The results from both our accelerometer and video data collections, together with the computations performed based on (3.11) lead to the creation of the Table 5.2.

The quantity  $v_{\text{coll}} = a_{\text{inst, af}} \cdot t_{\text{coll}}$  represents the change in velocity that the dynamic pendulum underwent during the collision. As such,  $v_{\text{inst, af}} = v_{\text{inst, be}} + v_{\text{coll}}$  is the resulting velocity of the dynamic pendulum after the collision. The distance  $r = v_{\text{inst, be}} \cdot t_{\text{coll}}$  from the Figure 3.4 is the distance covered by the dynamic colliding body during the collision. It is used to calculate the collision force  $F_{\text{coll}}$  according to (3.11).

We characterized collisions between solid masses in the laboratory through their time scale and all 3 average values are comprised around 0.005 [s]. The times are fairly similar among the three mediums (air, water and water surface) and local variations should be explained on the account of experimental imprecisions. The velocities before the collision however display a noticeable behaviour as they decrease from being measured in the air, on the water surface or in the water. The same goes noticeably for the collision forces, looking at the global values from the



**Figure 5.6:** Time series of the zoom on a collision event in collision pressure force experiment with the disc in the air

interactions in the air, at the water surface and in the water. All this suggests dissipative mechanisms as the bodies enter the fluid or are being completely submerged in it and, from our results in 5.1, it is clear that the added mass dissipative force would play a substantial role. A deeper study into the drag effects of submerged pendulums could help establishing predictions between a data set taken in the air and data sets from identical experiments but at the water surface and in the water.

### 5.3 Fieldwork : Ice block interactions

In the field investigation, we collected data from two different sources : the accelerometers screwed into the colliding ice blocks, the data of which we used to characterize the ice-ice collision time scales, and the video files from recordings made alongside the collision events, the data of which we used to observe and analyse the displacements of the ice blocks in the water when moving onto and reaching the vicinities of the wall. The ice blocks we mounted to the support frame to create our pendulums had different shapes : an rectangular piece of ice with a flat contact area, but also a piece of ice with a point edge on which the contacts were to occur. The targets of the collisions were also varied : we collided with a wall in the water but with ice blocks in the air, a small and light one as well as a larger and heavier one. An illustration of the decomposed trajectory of an ice block interacting with a wall in the water is featured in the Figure 5.13.

	$t_{\text{coll}}$	$a_{\text{inst, af}}$	$v_{\text{inst, be}}$	$v_{\text{coll}}$	$v_{\text{inst, af}}$	$r$	$F_{\text{coll}}$
Units	[s]	[m s <sup>-2</sup> ]		[m s <sup>-1</sup> ]		[m]	[N]
Air ball	0.006	-9.259	1.052	-0.053	0.999	0.006	8.794
Air cylinder	0.005	-11.15	1.028	-0.058	0.970	0.005	9.857
Air disc	0.005	-7.179	1.183	-0.043	1.140	0.006	5.852
Air all	0.005	-9.211	1.079	-0.052	1.027	0.006	7.917
Surface cylinder	0.005	-4.213	0.986	-0.020	0.966	0.005	3.234
Surface disc	0.004	-7.863	1.080	-0.035	1.046	0.005	5.896
Surface all	0.004	-6.649	1.050	-0.028	1.022	0.004	5.494
Water ball	0.010	-4.772	0.932	-0.014	0.914	0.009	1.793
Water cylinder	0.003	-5.431	0.709	-0.016	0.693	0.002	4.007
Water disc	0.005	-7.815	0.882	-0.044	0.838	0.004	7.094
Water all	0.006	-5.862	0.808	-0.023	0.785	0.005	3.299

**Table 5.2:** Summary of the results obtained from the collision pressure force experiment

### 5.3.1 Acceleration files

Our acceleration files provided time series of measurements from the interactions between different ice blocks. The reading and processing followed a protocol similar to that used in the laboratory pendulum investigations 5.2. After identifying the moments of collisions in the files, we collected the time stamps from the beginning and ends of the acceleration drop and deduced the collision time scale. A special case, a collision event between the rectangular block and the small and light ice block in the air is presented in the Figure 5.7. As the static block would be pushed or even topple from the impact, the event seems to be split in two parts, a first contact and an immediate second stress on the target. While we marked both peaks in our notes, we consider only the first one to be of relevance for it comprises the physics of collision we are looking for.

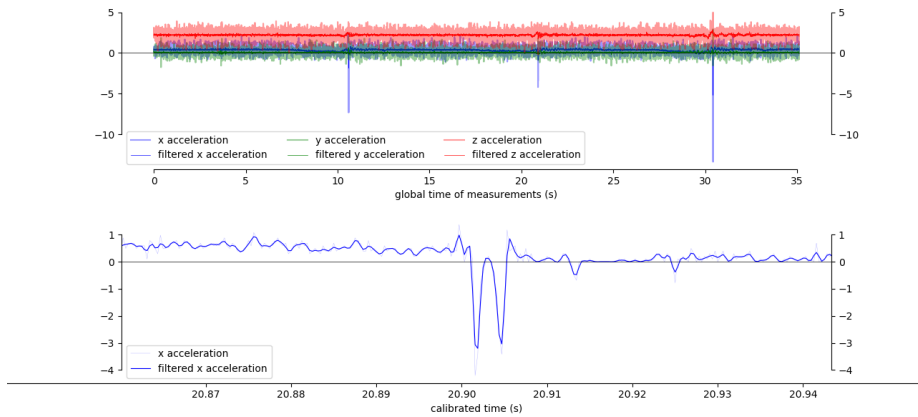
### 5.3.2 Video files

Several collision moments were observed and from them, we tracked the horizontal position of the ice block in the water moving towards its target. That data would then be loaded to Python where we plotted it together with the theoretical expectation from our model formula (3.19). All graphs presented a similar behaviour to the time series provided in the Figure 5.8.

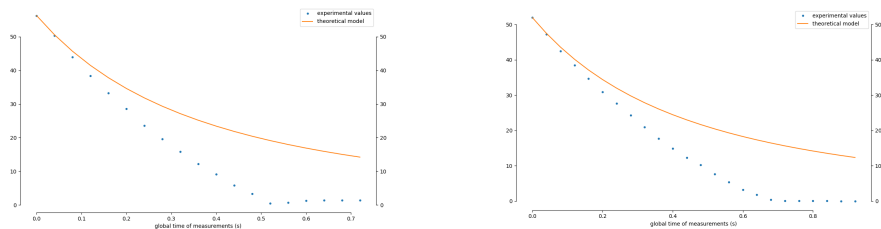
### 5.3.3 Results

The values from the collision times collection are presented in the Table 5.3.

Our results offer a clear contrast between ice block interactions in the air and in



**Figure 5.7:** Time series of the acceleration during a collision event between an ice cube and a small block in the air from the ice block interactions experiments



**Figure 5.8:** Time series of the approach of an ice block (rectangular on the left, pointy on the right) towards a wall in the water in the ice block interactions experiments

the water. As such, we find a much shorter time scale in the air, an averaged value from all experiments in the air being found around 0.003 [s]. In the water however, the times are 4 times as long and the characteristic time for such collisions would be around 0.012 [s]. These investigations follow up on the work done by [10] on ice-ice collision times, updating the values through measurements performed with devices with higher resolutions. The differences we measured in acceleration between the collisions in the air and the collisions in the water can be explained by the momentum transfer to the waves and water motion, as observable in the Figure 5.13.

The video file date extraction has proven to question our model on the water layer



Units	$t_{\text{coll}}$ [s]
Air — Rectangular block against small block	0.002
Air — Rectangular block against large block	0.004
Air — Pointy block against large block	0.005
Air — all	0.0033
Water — Rectangular block against wall	0.010
Water — Pointy block against wall	0.013
Water — all	0.015

**Table 5.3:** Summary of the results obtained in the ice block interactions experiments

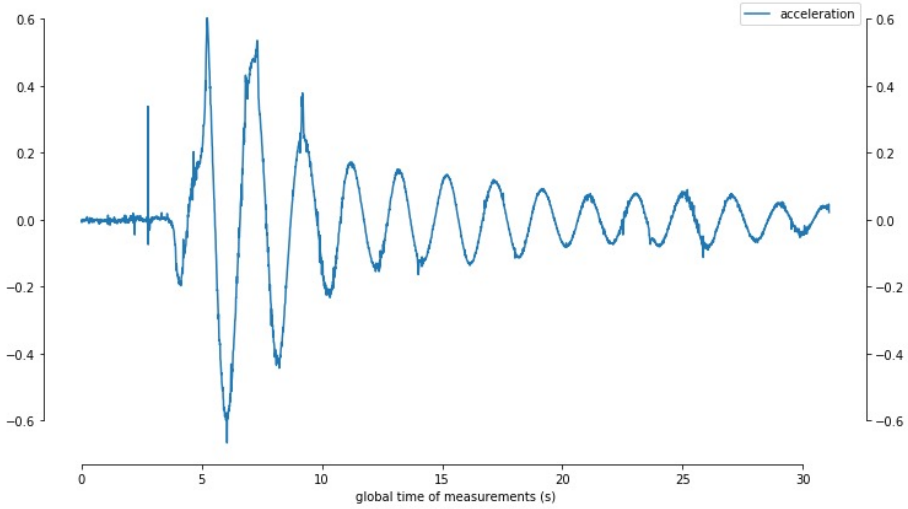
evolution between a solid body and a wall it moves towards as none of the experimental data could be satisfactorily fitted with our theoretical expectation. This could however be explained by different factors such as the error propagation from the video reading to the points tracking, the presence of ice in the bassin, which could have changed the physics of the interaction. But most importantly perhaps, the blocks we used where only half submerged, causing the physics in play to reach the limits of the applicability of our model. Further experiments in a more controlled environment and precise measurement devices to observe the presence or absence of contact during the interaction of a submerged ice block with an ice well would help lift these doubts. These questions are also a bigger topic in theoretical physics of fluid dynamics and inherently lead to more fundamental and mathematical aspects of this science, so [26].

## 5.4 Fieldwork : Ice cube buoyancy oscillations

Our results in this experiments were provided through the files produced by an accelerometer lying on the buoyant oscillating ice cube. The raw data can be observed in the Figure 5.9 and a peculiar behaviour can be taken from it : past the initialization and from the start of the effective collection of buoyancy measurements around  $t = 5$  [s], the very first oscillation looks unaffected by damping yet and features an almost perfect local sinusoidal function fit potential. This is supported by our model 3.4.2 an in particular (3.35) predicting two different time scales on which two different phenomena would occur : an undamped oscillation on very short times at first and an exponential damping over long times afterwards.

### 5.4.1 Short time behaviour

For each iteration of the buoyancy oscillation tests, the time series all had in common an unaltered first oscillation in amplitude. This situation presents a good



**Figure 5.9:** Time series of the acceleration data recovered from the buoyancy oscillations experiments

opportunity to extract the initial velocity  $u_0$  in (3.32) from this local data.

Since we have access to acceleration data through our measurements, in other words  $\eta''$ , we begin the analysis by numerically integrating our curves over time. Local integration of a sinusoidally shaped graph yields another periodic function of the same nature and we choose null for the value of the integration constant corresponding to a geometric offset of our result. In order to fit the integrated periodic velocity, we use the following model  $\Xi(t)$  where the parameters are described by the different  $\xi_i$  :

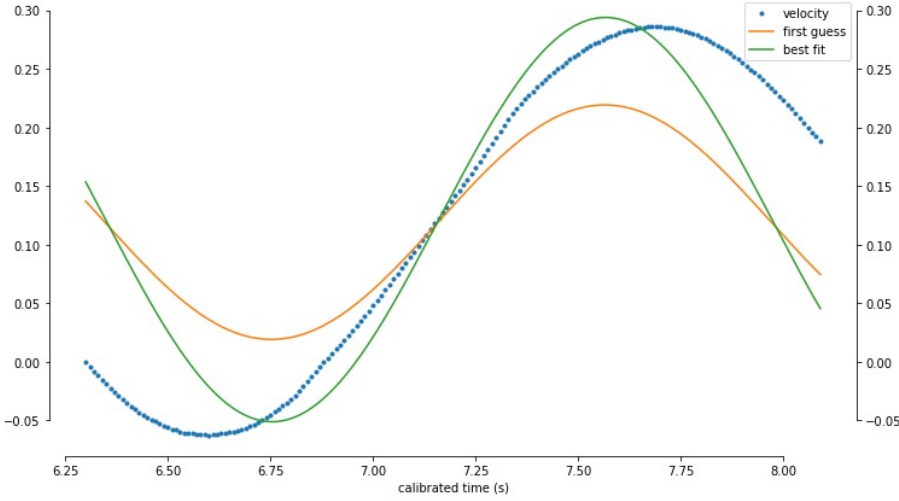
$$\Xi(t) = \xi_1 \sin(\omega t + \xi_2) + \xi_3 \quad (5.3)$$

in which  $\xi_1$  represents the amplitude of the oscillations,  $\omega = \sqrt{\frac{\rho_w g}{\rho_i h_i}}$  represents the angular frequency,  $\xi_2$  represents the phase and  $\xi_3$  represents the offset of the sinusoidal function.

An optimization algorithm using the model (5.4) allows us to create the Figure 5.10.

### 5.4.2 Long time behaviour

The previous regression allowed us to find a value for the initial velocity and hence, we can now establish a model for a regression on the global data, incorporating the



**Figure 5.10:** Time series of the local regression at the start of the buoyancy oscillation data set

damping terms responsible for the exponential decrease of the buoyancy oscillation amplitude. We use the formula (3.35) and aim at finding a value for the effective dynamic viscosity  $\mu_v$ , a variable that we will enter as a parameter into our next regression model. Among the concerned terms,  $\lambda = \sqrt{\frac{\rho_w \omega}{2\mu_v}} = \lambda(\mu_v)$ , which means that  $\gamma = \frac{1}{4} \frac{\sinh 2\lambda + \sin 2\lambda}{\cosh 2\lambda - \cos 2\lambda} = \gamma(\mu_v)$  whereas  $S_l = 4 \left( \frac{\rho_i h_i}{\rho_w} \right)^2$  doesn't depend on  $\mu_v$ . Our model  $\Xi(t)$  where the parameters are described by the different  $\xi_i$  then reads :

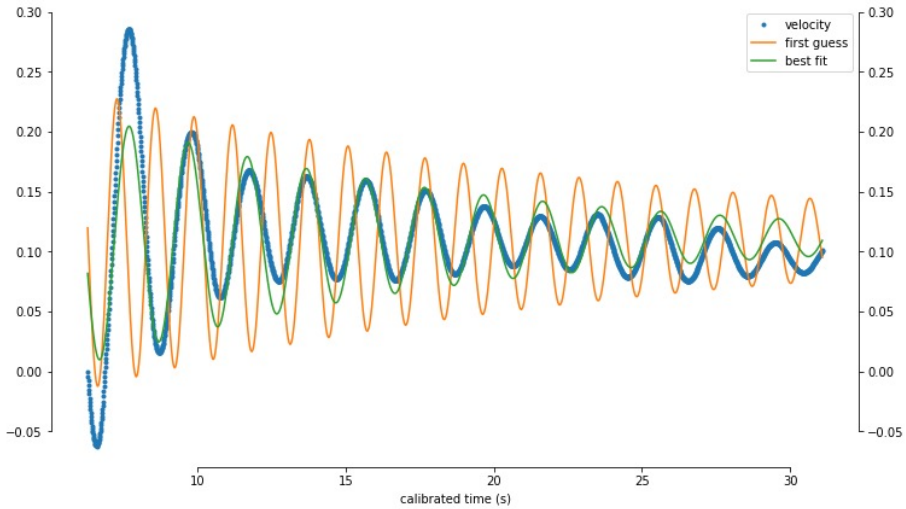
$$\Xi(t) = u_0 \sin \left( \omega \frac{t}{\xi_2} + \xi_1 \right) \cdot e^{-\frac{\xi_2 t m}{S_l \xi_3 \lambda(\xi_3) \gamma(\xi_3)}} + \xi_4 \quad (5.4)$$

in which  $\xi_1$  represents the phase,  $\xi_2$  represents the short-to-long time conversion parameter,  $\xi_3$  represents the dynamic viscosity and  $\xi_4$  represents the offset of the complete function.

A result of the optimization algorithm on the full data set is presented in the Figure 5.11.

### 5.4.3 Results

Using our previous numerical values sur as 4.16 but also data from the Figure 4.17 as well as the extracted parameters from our regression models, we can present our



**Figure 5.11:** Time series of the total regression of the buoyancy oscillation data set

results in the Table 5.4. Our values are gathered from two sets of experiments and we present a grand average at the bottom of the table.

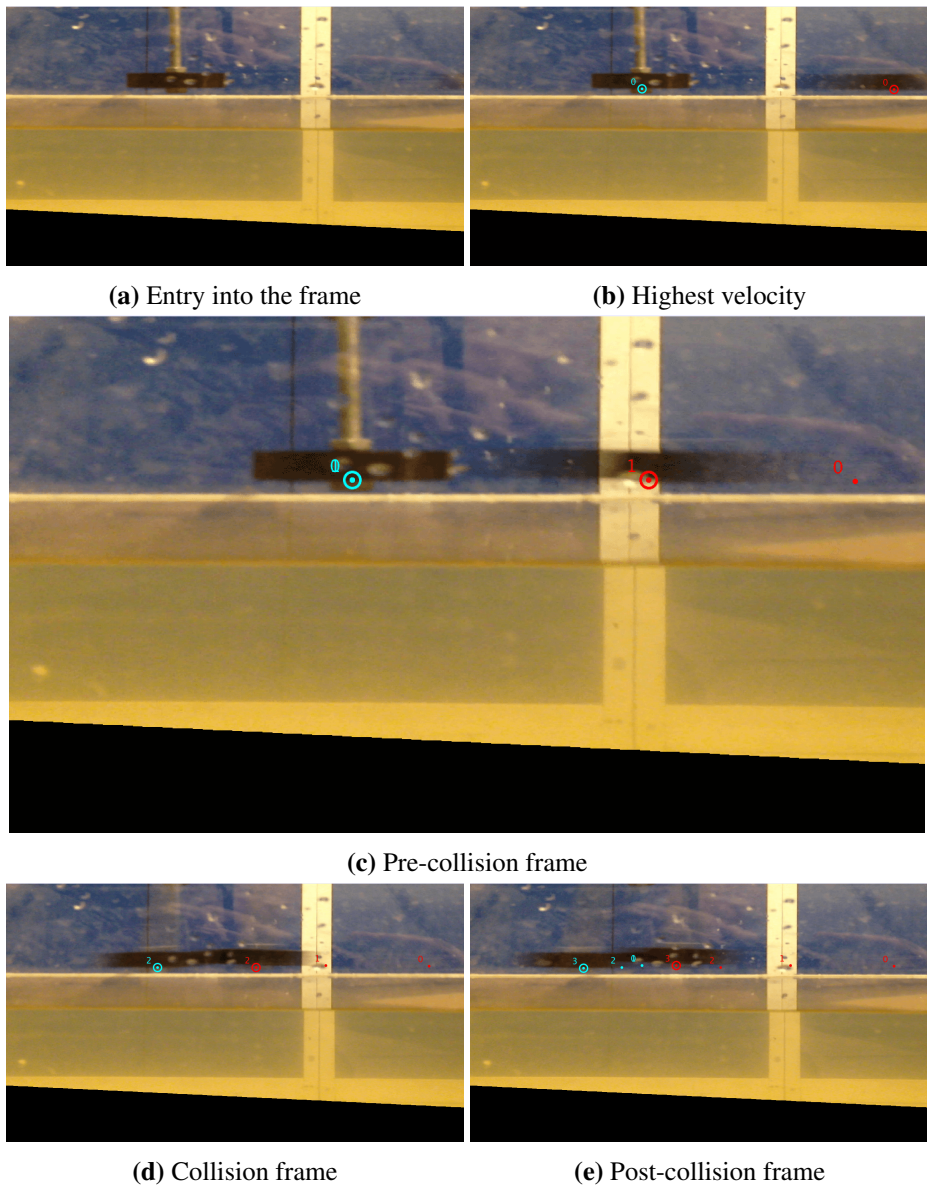
	$u_0$	$\mu_v$	$\tau$
Units	$[\text{m s}^{-1}]$	$[\text{kg m}^{-1} \text{s}^{-1}]$	$[\text{s}]$
Set 1	0.147	1.575	14.64
Set 2	0.172	1.830	12.61
Mean	0.160	1.703	13.63

**Table 5.4:** Summary of the results obtained in the ice cube buoyancy oscillations experiments

Both sets gave us very close values for the characteristic time of damping, which is consistent with a continuity in the experimental setup as we performed next to identical tests each time. As discussed previously, this time is the analog for buoyancy oscillation of the mean lifetime for nuclear decay reaction ; it is the time at which the oscillation amplitudes have reduce by 0.37 of their initial value.

While the velocity amplitudes coming from the initial oscillation respect the experimental order of magnitude, our dynamic viscosity is too large, independently of how our regression model is run, despite the test of several mathematical variations of it and despite how well our algorithm managed to fit the data set curve. In fact, our result outgrows the theoretical value by a factor  $10^3$ . The author [10] performed similar tests, found a value around 200 times larger than the theoret-

ical value of  $\mu_{v, \text{theo}} = 0.0015 \text{ [kg m}^{-1} \text{ s}^{-1}]$  and concluded that different measurements lead to similar numerical results than the one published. With this theoretical value, we can compute a theoretical characteristic damping time  $\tau_{\text{theo}} = 1117.56 \text{ [s]}$ , which is close to 19 minutes and wouldn't comply either with our experimental values close to 14 [s] nor with our observations in the field. In this light, we are tempted to proceed in a similar way as [10] and try to understand our result despite remaining critical of different aspects : in the field, the slush in the canals around the ice cube as well as the horizontal motions and turbulences induced during the buoyancy oscillations could have contributed to a virtually higher viscosity value. We are also ready to question the time-shifting idea introducing the  $\epsilon$  parameter into our model 3.4.2, also considering how complex it already is. But our intuition rather goes towards further investigations, both theoretically and experimentally to uncover more about the topic of energy dissipation from, in and around the ice and through the water.

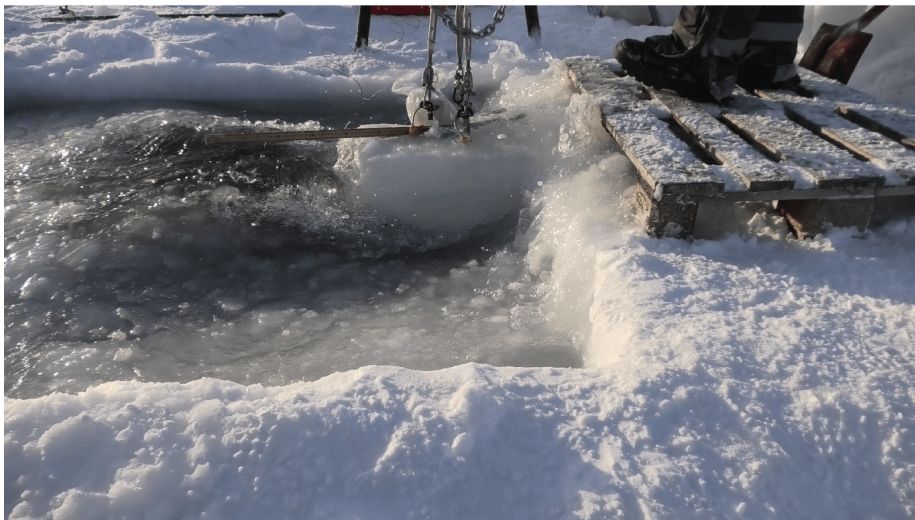


**Figure 5.12:** Illustration of the trajectory of a disc in the air from the collision pressure force experiment



(a) Entry into the water

(b) Momentum transfer to the water



(c) Collision frame



(d) Wave reflection on the wall

(e) Pull-back from the reflected wave

**Figure 5.13:** Illustration of the trajectory of an ice block against a wall in the ice block interactions experiment

# Chapter 6

## Conclusions

"What one man can think, another man can do."

---

Jules Vernes

### 6.1 Damping and wave attenuation in the marginal ice zone

In this work, we studied some fundamental phenomena reigning over certain damping mechanisms dissipating the surface wave energy when it enters the marginal ice zone. For that, theory was introduced and the main fluid dynamics equations relevant to our work were derived. We then established a series of theoretical and mathematical models in order to both support and motivate experimental work, both in the laboratory and in the field.

As we worked with physical pendulums and wanted to understand the effects of the added mass force on different shapes placed on the surface and then submerged into a fluid, we proceeded to find both the moments of inertia  $I_S$  for these shapes in the air and the added mass term  $m_{\text{add}}$  in the water. Our moments of inertia values, given in  $[\text{kg}^2 \text{m}]$ , are 0.799 for the ball, 0.751 for the cylinder and 0.741 for the disc. Our added mass values, given in  $[\text{g}]$ , are 30.4 for the ball, 119 for the cylinder and 78.7 for the disc.

We also performed a long series of collision and interactions between two physical pendulums placed in different continuums and wondered how each medium would affect both their characteristic collision time  $t_{\text{coll}}$  and their collision pressure force  $F_{\text{coll}}$ . Our collision time value, given in  $[\text{s}]$ , are 0.005 in the air, 0.004 on the water surface and 0.006 in the water. Our collision pressure forces values, given in  $[\text{N}]$



are 7.917 in the air, 5.494 on the water surface and 3.299 in the water.

Once in the field, we performed a succession of experiments from building pendulums with ice blocks and setting up collision events between them both in the air and in the water, characterizing them through their time scale  $t_{\text{coll}}$ . In the water in particular, we observed the dynamic interactions between an ice block colliding into a wall and the water layer between them and tried to verify our theoretical prediction about a no-contact collision between the block and the wall. Our collision time value, given in [s], are 0.003 in the air and 0.015 in the water. Our model couldn't be properly verified following all theoretical conditions and we observe a contact in each collision event.

Our second main experiment in the field consisted of a large buoyant ice cube on which we would research the characteristic time of oscillation damping  $\tau$ . This would also motivate an experimental value for the dynamic viscosity of the water  $\mu_v$  in those conditions. Our oscillation damping time value, given in [s], is 13.36 and our dynamic viscosity value, given in [ $\text{kg m}^{-1} \text{s}^{-1}$ ] is 1.703.

## 6.2 Future prospects

Along with the many remarks that have been made to comment our results, we here suggest a few direct steps that could push such investigations of surface waves with sea ice in the marginal ice zone ahead.

One idea would be to expand the range of damping mechanisms from the ice when given energy from surface waves. Many effects add up to the two major ones we investigated here and only a understanding of the larger number of them would allow a thorough progress of this research. As [27] describes, one could study the turbulences in the surface boundary layer under the ice, a topic that itself comprised several phenomena such as effective eddy viscosities leading to diffusivity, a mixing in the pycnocline or the natural deep convection that sea ice triggers within the ice columns under it. We could also research on the Stokes boundary layer problem arising from the oscillations of a rigid plane, in this case the horizontally oscillating sea ice, on top of fluid.

Another theoretical approach would be the development of more complex mathematical models taking account for more dissipative forces like the drag terms we sometimes had to leave out of our equations for the sake of an analytical solution. But one could also include most understood air-ice-sea phenomena into a numerical simulation, in order to link the physical models developed here and elsewhere on sea ice dynamics to energy dispersion models of waves in sea ice like those presented by [28].

As we have been discussing the further variety but also the broader possible future prospects of such a work, we would like to mention what came to our everyday mind during this research and presented itself either like a logical continuation of our progress or like a good bridge to a new topic. One could for instance break down the scale of observations and focus on ice mechanics of and the mechanics of material in general to look at either the collisions or the drag effects from a surface roughness, contact area, shear and tensile stresses and loads perspective. When doing work in the laboratory, one could also try to establish the optimal parameters to replicate Reynolds numbers from the field in controlled and smaller environments. Arctic field work is in fact as wonderful and uplifting as it is demanding and constraining.

### 6.3 Possible improvements

Very naturally, we suggest herewith small adjustments and improvements that could be made to the existing protocols and experimental setup, for either smoother or more precise experiments.

Thinking about the pendulum experiments, more fluid bearing and more steady pieces could improve the collisions while a setup in which the accelerometers would themselves be swinging at the bottom of the pendulum, fixed to much stiffer rods to avoid the harmonic modes' vibrations would increase the precision of the measurements.

Furthermore, rudimentary laser interferometers like that used by [29] would give a sub-millimetric precision to measurements that sometimes rely on very small scales. On the hardware still, some thoughts could be given to a dynamic method of data collection, which would then allow more reactivity towards the initial conditions of some experiments. When it comes to the hardware and induced theoretical or inherent quantities, a gain in precision in the measurement of the setup could reduce the error to which we would compare our experimental values.

As a general thought, the design and commitment to a whole chapter on errors and error propagations in the future could help building up an archive of protocols and experiments that get profit from each other's improvements.

Finally, the next step we could take in the processing of the data from the accelerometer could be the abandonment of the small angles and uniaxial motion approximation to the profit of a tridimensional data processing.

# Chapter 7

## Appendix

"Not only cunning casts in clay :  
Let Science prove we are, and then  
What matters Science unto men,  
At least to me ? I would not stay."

---

Lord Alfred Tennyson

### 7.1 Mathematics

#### 7.1.1 Curl of a gradient

The curl of a gradient is computed in the following way :

$$\begin{aligned}\nabla \times (\nabla t) &= \begin{vmatrix} \hat{x} & \hat{y} & \hat{z} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ \frac{\partial t}{\partial x} & \frac{\partial t}{\partial y} & \frac{\partial t}{\partial z} \end{vmatrix} \\ &= \hat{x} \left( \frac{\partial^2 t}{\partial y \partial z} - \frac{\partial^2 t}{\partial z \partial y} \right) + \hat{y} \left( \frac{\partial^2 t}{\partial z \partial x} - \frac{\partial^2 t}{\partial x \partial z} \right) + \hat{z} \left( \frac{\partial^2 t}{\partial x \partial y} - \frac{\partial^2 t}{\partial y \partial x} \right) \\ &= 0\end{aligned}$$

by equality of the cross-derivatives.

#### 7.1.2 Gradient of a product

The gradient of a product is computed in the following way :

$$\begin{aligned}
\nabla(\vec{A} \cdot \vec{B}) &= \nabla(A_x B_x + A_y B_y + A_z B_z) \\
&= \sum_{i=x,y,z} \hat{i} \frac{\partial}{\partial i} (A_x B_x + A_y B_y + A_z B_z) \\
&= \sum_{i=x,y,z} \hat{i} \left( A_x \frac{\partial B_x}{\partial i} + B_x \frac{\partial A_x}{\partial i} + A_y \frac{\partial B_y}{\partial i} + B_y \frac{\partial A_y}{\partial i} + A_z \frac{\partial B_z}{\partial i} + B_z \frac{\partial A_z}{\partial i} \right)
\end{aligned}$$

Having in mind the length of the calculus, for graphical reasons, we prefer to compute the quantities only according to a single arbitrary axis from now on, for instance the  $x$ -axis :

$$\left[ \nabla(\vec{A} \cdot \vec{B}) \right]_x = A_x \frac{\partial B_x}{\partial x} + B_x \frac{\partial A_x}{\partial x} + A_y \frac{\partial B_y}{\partial x} + B_y \frac{\partial A_y}{\partial x} + A_z \frac{\partial B_z}{\partial x} + B_z \frac{\partial A_z}{\partial x}$$

Considering that we know this vector identity from [30], we will derive its elements part by part and verify that they fit the gradient of our product when added together :

$$\begin{aligned}
\left[ \vec{A} \times (\nabla \times \vec{B}) \right]_x &= A_y (\nabla \times \vec{B})_z - A_z (\nabla \times \vec{B})_y \\
&= A_y \left( \frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} \right) - A_z \left( \frac{\partial B_x}{\partial z} - \frac{\partial B_z}{\partial x} \right) \\
\left[ \vec{B} \times (\nabla \times \vec{A}) \right]_x &= B_y \left( \frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y} \right) - B_z \left( \frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x} \right) \\
\left[ (\vec{A} \cdot \nabla) \vec{B} \right]_x &= \left( A_x \frac{\partial}{\partial x} + A_y \frac{\partial}{\partial y} + A_z \frac{\partial}{\partial z} \right) B_x \\
&= A_x \frac{\partial B_x}{\partial x} + A_y \frac{\partial B_x}{\partial y} + A_z \frac{\partial B_x}{\partial z} \\
\left[ (\vec{B} \cdot \nabla) \vec{A} \right]_x &= B_x \frac{\partial A_x}{\partial x} + B_y \frac{\partial A_x}{\partial y} + B_z \frac{\partial A_x}{\partial z}
\end{aligned}$$

We can now combine the parts into :

$$\begin{aligned}
C &\equiv \left[ \vec{A} \times (\nabla \times \vec{B}) + \vec{B} \times (\nabla \times \vec{A}) + (\vec{B} \cdot \nabla) \vec{A} + (\vec{B} \cdot \nabla) \vec{A} \right]_x \\
&= A_y \frac{\partial B_y}{\partial x} - A_y \frac{\partial B_x}{\partial y} - A_z \frac{\partial B_x}{\partial z} - A_z \frac{\partial B_z}{\partial x} + A_x \frac{\partial B_x}{\partial x} + A_y \frac{\partial B_x}{\partial y} + A_z \frac{\partial B_x}{\partial z} \\
&\quad + B_y \frac{\partial A_y}{\partial x} - B_y \frac{\partial A_x}{\partial y} - B_z \frac{\partial A_x}{\partial z} - B_z \frac{\partial A_z}{\partial x} + B_x \frac{\partial A_x}{\partial x} + B_y \frac{\partial A_x}{\partial y} + B_z \frac{\partial A_x}{\partial z} \\
&= A_x \frac{\partial B_x}{\partial x} + B_x \frac{\partial A_x}{\partial x} + A_y \frac{\partial B_y}{\partial x} + B_y \frac{\partial A_y}{\partial x} + A_z \frac{\partial B_z}{\partial x} + B_z \frac{\partial A_z}{\partial x} \\
&= \left[ \nabla(\vec{A} \cdot \vec{B}) \right]_x
\end{aligned}$$

Doing the same for the other axes, we end up with the total vectorial identity for the gradient of a product :

$$\nabla(\vec{A} \cdot \vec{B}) = \vec{A} \times (\nabla \times \vec{B}) + \vec{B} \times (\nabla \times \vec{A}) + (\vec{B} \cdot \nabla) \vec{A} + (\vec{B} \cdot \nabla) \vec{A}$$

In the special case where  $\vec{A} = \vec{B} \equiv \vec{u}$ , we would have :

$$\frac{1}{2} \nabla(\vec{u} \cdot \vec{u}) = \vec{u} \times (\nabla \times \vec{u}) + (\vec{u} \cdot \nabla) \vec{u}$$

### 7.1.3 Integrating a partial derivative, the Leibniz rule

Let  $f(x, y)$  be continuous over a closed interval and hence, Riemann-integrable. If  $x$  and  $y$  are independent variables, which is equivalent to  $y$  being held constant during the integration, then :

$$\int \frac{\partial}{\partial x} f(x, y) dx = f(x, y) + C(y)$$

where  $C(y)$  is the integration constant.

This theorem is a corollary of the Lebesgue Dominated Convergence Theorem and its proof can be found in [31].

### 7.1.4 Average function value

The average value of a function  $f(x)$  over the interval  $[a, b]$  is given by :

$$f_{avg} = \frac{1}{b-a} \int_a^b f(x) dx$$

The simple proof starts by subdividing  $[a, b]$  into  $n$  subintervals of length  $\Delta x = \frac{b-a}{n}$  and choosing  $n$  points  $x_i^*$  respectively each in one of these intervals such that the average of the function values  $f(x_i^*)$  is :

$$\frac{f(x_1^*) + f(x_2^*) + \cdots + f(x_n^*)}{n} \stackrel{n = \frac{b-a}{\Delta x}}{=} \frac{1}{b-a} \sum_{i=1}^n f(x_i^*) \Delta x$$

As we increase  $n$  and goes to infinity, we get :

$$f_{avg} = \lim_{n \rightarrow \infty} \frac{1}{b-a} \sum_{i=1}^n f(x_i^*) \Delta x = \frac{1}{b-a} \int_a^b f(x) dx$$

### 7.1.5 Huygens-Steiner or Parallel Axis Theorem

If a body of mass  $m$  initially made to rotate about an axis  $z$  passing through its centre of mass is made to rotate about another axis  $z'$  which is parallel to and at a distance  $l$  from  $z$ , its moment of inertia with respect to  $z'$  is :

$$I = I_{cm} + ml^2$$

with  $I_{cm}$  is the moment of inertia with respect to the initial axis  $z$ .

The proof considers a Cartesian system of coordinates in which the distance between  $z$  and  $z'$  lies along the  $x$ -axis, with the centre of mass at the origin. Relatively to the  $z$ -axis, the moment of inertia would be :

$$I_{cm} = \int (x^2 + y^2) dm$$

Now computing the moment of inertia relative to  $z'$  becomes :

$$\begin{aligned} I &= \int ((x+l)^2 + y^2) dm \\ &= \int (x^2 + y^2) dm + l^2 \int dm + \underbrace{2l \int x dm}_{=0 \text{ since cm at origin}} \\ &= I_{cm} + ml^2 \end{aligned}$$

### 7.1.6 Gauss or Fundamental Theorem for Divergences

The fundamental theorem of integral calculus states that the integral of a derivative over a region is equal to the value of the function at the boundary. In this particular case, the integral of a divergence over a volume  $V$  is equal to the value of the function at the surface  $S$  that bounds the volume, and can be expressed as :

$$\int_V (\nabla \cdot \vec{v}) dV = \oint \vec{v}_S \cdot d\vec{S}$$

To get a physical interpretation of this theorem, we can consider  $\vec{v}$  to be the flow of an incompressible fluid such that the right side of the equation is the flux of  $\vec{v}$ , in other words the total amount of fluid flowing out through the closed surface  $S$  per unit time. The left side presents a divergence which describes the outflow of the vectors from a point, here "faucets" pouring out liquid. If we have a region with several faucets pouring out and remembering that the liquid is incompressible, an equal amount of liquid will be pushed through the boundaries of the region. To verify the theorem, one could either count all the faucets and record how much they respectively put out, or go around the boundary and measure the flow at each point, so that :

$$\int (\text{faucets within the volume}) = \oint (\text{flow out through the surface})$$

A mathematical proof of the theorem is provided by [32].

### 7.1.7 Material derivative

If a function  $T(t, x, y, z)$  depends on variables that themselves are a function of the variable of differentiation  $x = x(t), y = y(t), z = z(t)$ , the chain rule yields :

$$\begin{aligned} \frac{d}{dt} T(t, x(t), y(t), z(t)) &= \frac{\partial T}{\partial t} + \left( \frac{\partial T}{\partial x} \right) \left( \frac{\partial x}{\partial t} \right) + \left( \frac{\partial T}{\partial y} \right) \left( \frac{\partial y}{\partial t} \right) + \left( \frac{\partial T}{\partial z} \right) \left( \frac{\partial z}{\partial t} \right) \\ &= \frac{\partial T}{\partial t} + u_x \frac{\partial T}{\partial x} + u_y \frac{\partial T}{\partial y} + u_z \frac{\partial T}{\partial z} \\ &= \frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T \\ &= \frac{DT}{Dt} \end{aligned}$$

such that :

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \vec{u} \cdot \nabla$$

## 7.2 Python

Since we coded every single tool of data processing and analysis that has been used in these investigations, we hereunder provide all the codes we wrote with the aim of presenting a ready-to-use package of Python dynamic functions to the curious reader. Provided they don't forget to load the following functions in their main code, they should be able to reproduce most of the steps we achieved in this work.

### 7.2.1 File loading and reading

#### Labwork : file loading

```
def accelerometerLabworkFileLoading(filename , SIUnits):

    filename = filename

    # define how the extracted columns are going to be named
    columnsDytran = [ 'xAxisAcceleration' , 'yAxisAcceleration' , '
        zAxisAcceleration' ]

    # save the frequency of the data file
    frequency = int(open('../Data/Labwork/Data/' + filename).readline
        () [12:])

    # extract the data file and append it to the data object
    data = pd.read_csv('../Data/Labwork/Data/' + filename , delimiter
        = '\t' , skiprows=2 , names=columnsDytran , decimal=",")

    # change the accelerations to SI units
    if SIUnits == True:
        data = data * 9.81
    if SIUnits == False:
        data = data

    # create a time column with values 1 (added to the end of the data
        file)
    data['time'] = 1

    # reorder the data file to have the time column in the beginning
    data = data[['time' , 'xAxisAcceleration' , 'yAxisAcceleration' , '
        zAxisAcceleration']]

    # set the time in seconds according to the frequency
    for i in range(len(data)):
        data.iloc[i,0] = i/frequency

    # set the time column as the index for the data file
    data = data.set_index('time' , drop = True)

    # change the filename in order to remove the "/" and " "
```



```
filename = filename.replace("/", "_")
filename = filename.replace(" ", "_")

# remove the extensions from the filename
filename = filename.replace(".TXT", "")
filename = filename.replace(".txt", "")

return data, frequency, filename
```

## Fieldwork : file loading

```
def accelerometerFieldworkFileLoading(filename, SIUnits):

    filename = filename

    # define how the extracted columns are going to be named
    columnsDytran = ['xAxisAcceleration', 'yAxisAcceleration', 'zAxisAcceleration']

    # save the frequency of the data file
    frequency = int(open('../Data/Fieldwork/Data/Pendulum exp/' +
        filename).readline()[12:])

    # extract the data file and append it to the data object
    data = pd.read_csv('../Data/Fieldwork/Data/Pendulum exp/' +
        filename, delimiter='\t', skiprows=2, names=columnsDytran,
        decimal=",")

    # change the accelerations to SI units
    if SIUnits == True:
        data = data * 9.81
    if SIUnits == False:
        data = data

    # create a time column with values 1 (added to the end of the data
    # file)
    data['time'] = 1

    # reorder the data file to have the time column in the beginning
    data = data[['time', 'xAxisAcceleration', 'yAxisAcceleration',
        'zAxisAcceleration']]

    # set the time in seconds according to the frequency
    for i in range(len(data)):
        data.iloc[i,0] = i/frequency

    # set the time column as the index for the data file
    data = data.set_index('time', drop = True)

    # change the filename in order to remove the "/" and " "
    filename = filename.replace("/", "_")
    filename = filename.replace(" ", "_")

    # remove the extensions from the filename
    filename = filename.replace(".TXT", "")
    filename = filename.replace(".txt", "")
```

```
return data , frequency , filename
```

## 7.2.2 Calibration, filtering and optimization

### Labwork : calibration peak correction

```
def calibrationPeakCorrection(filename , data , calibrationAxis ,
    cutFromStart , cutFromFinish , findpeaksThreshold , findpeaksDistance ,
    findpeaksProminence , findpeaksWidth , findpeaksWlen ,
    findpeaksRelativeHeight , findpeaksPlateauSize):

    # set up the description of the calibration process
    calibrationUpdatesText = "Updates on the calibration:"
    print('\n',
          "Filename: ", filename ,
          '\n',
          calibrationUpdatesText ,
          '\n',
          "-" * len(calibrationUpdatesText) ,
          '\n',
          "Calibration axis: ", calibrationAxis)

    # update the filename by adding the calibration axis at the end
    filename = filename + "_" + calibrationAxis.upper()

    # create a list to get the axis' index
    coordinates = ['x','y','z']

    # select the axis to work with
    for i in coordinates:
        if calibrationAxis == i:
            selectedColumn = coordinates.index(i)

    # correct the calibration axis' acceleration by centering it
    around 0 and taking the absolute value
    data['correctedPositive{ }AxisAcceleration'.format(calibrationAxis)
        ] = abs(data.iloc[:,selectedColumn] - data.iloc[:,
        selectedColumn].mean())

    # define a criterion for the height of the peaks
    peakHeightCriterion = max(abs(data.iloc[:,selectedColumn].min()),
        data.iloc[:,selectedColumn].max())//2

    # find the peaks in the data set
    peaks, _ = scipy.signal.find_peaks(data['correctedPositive{ }
        AxisAcceleration'.format(calibrationAxis)], height=
        peakHeightCriterion , threshold=findpeaksThreshold , distance=
        findpeaksDistance , prominence=findpeaksProminence , width=
        findpeaksWidth , wlen=findpeaksWlen , rel_height=
        findpeaksRelativeHeight , plateau_size=findpeaksPlateauSize)

    # pick the first of the peaks in the data set
    firstPeak = peaks[0]

    # run the same height criterion test on the two points before the
    first peak and eventually selected them if they succeed
```

```
if abs(data.iloc[firstPeak - 1,selectedColumn]) >=
    peakHeightCriterion:
    firstPeak = firstPeak - 1
if abs(data.iloc[firstPeak - 1,selectedColumn]) >=
    peakHeightCriterion:
    firstPeak = firstPeak - 1
print('The first peak as been found at ',
      data.index[firstPeak],
      's.')
```

```
# drop the data before the first peak and restart the set from
there
data = data.iloc[firstPeak:]

# restart the index value from the first peak
data.index = data.index - data.index[0]
print('The index on the calibrated data now restarts from the
calibration peak.')
```

```
# only take data from cutFromStart to cutFrom Finish
data = data[data.index[0]+cutFromStart:data.index[-1]-
cutFromFinish]
print('The data has been cut to ',
      cutFromStart,
      's after the calibration peak and ',
      cutFromFinish,
      's before the end of the measurements.')
```

```
return data, calibrationAxis, filename, cutFromStart,
cutFromFinish
```

### Fieldwork : calibration and cut

```
def calibrationAndCut(filename, data, cutFromStart, cutFromFinish):

    # set up the description of the calibration process
    calibrationUpdatesText = "Updates on the calibration:"
    print('\n',
          "Filename: ", filename,
          '\n',
          calibrationUpdatesText,
          '\n',
          "-" * len(calibrationUpdatesText))

    # only take data from cutFromStart to cutFrom Finish
    data = data[data.index[0]+cutFromStart:data.index[-1]-
cutFromFinish]
    print('The data has been cut to ',
          cutFromStart,
          's after the calibration peak and ',
          cutFromFinish,
          's before the end of the measurements.')
```

```
return data, filename, cutFromStart, cutFromFinish
```

## 7.2.3 Butterworth filter

```

def butterworthFilter(data, samplingFrequency, cutoffFrequency,
    filterOrder):

    # normalize the frequency, the -3dB point
    w = cutoffFrequency / (samplingFrequency / 2)

    # create the Butterworth digital filter
    b, a = signal.butter(filterOrder, w, 'low')

    # apply the Butterworth filter to the data
    filteredxAxisAcceleration = signal.filtfilt(b, a, data.
        xAxisAcceleration)
    filteredyAxisAcceleration = signal.filtfilt(b, a, data.
        yAxisAcceleration)
    filteredzAxisAcceleration = signal.filtfilt(b, a, data.
        zAxisAcceleration)

    # add the filtered horizontal acceleration to the data file
    data = data.assign(filteredxAxisAcceleration=
        filteredxAxisAcceleration)
    data = data.assign(filteredyAxisAcceleration=
        filteredyAxisAcceleration)
    data = data.assign(filteredzAxisAcceleration=
        filteredzAxisAcceleration)

    return data, samplingFrequency, cutoffFrequency, filterOrder

```

## 7.2.4 Sinus regression

```

def sinusRegression(xAxis, yAxis, guessAmp, guessFreq, guessPhase):

    guess_amp = guessAmp #0.007
    guess_freq = guessFreq #3.5
    guess_phase = guessPhase #1.4
    guess_mean = np.mean(yAxis)

    # first estimate
    data_first_guess = guess_amp*np.sin(guess_freq*xAxis+guess_phase)
        + guess_mean

    # define the function to optimize, minimizing the difference
    # between the actual data and our guessed parameters
    optimize_func = lambda x: x[0]*np.sin(x[1]*xAxis+x[2]) + x[3] -
        yAxis
    est_amp, est_freq, est_phase, est_mean = leastsq(optimize_func, [
        guess_amp, guess_freq, guess_phase, guess_mean])[0]

    # recreate the fitted curve using the optimized parameters
    data_fit = est_amp*np.sin(est_freq*xAxis+est_phase) + est_mean

    # define the geometry in which to place the subplots
    grid = plt.GridSpec(1, 1)

    # create the figure environment for the plots
    fig = plt.figure(figsize=(12, 7))

```

```
# set the plot location plots
ax = fig.add_subplot(grid[0,0])

plt.plot(xAxis, yAxis, ".")
plt.plot(xAxis, data_first_guess, label='first guess')
plt.plot(xAxis, data_fit, label='best fit')

# set the label for the bottom subplot
ax.set_xlabel('calibrated time from peak (s)')

# create lines for the x-axes
#axhline(xmin=0, xmax=1, linewidth=0.5, c='k')

# turn off some spines
ax.spines['top'].set_color('none')

# create arrays with the automated locations of the ticks

xlocs = ax.get_xticks()
ylocs = ax.get_yticks()

# create arrays with the boundary values of the ticks
xTicksMinMax = (xlocs[1], xlocs[-2])
yTicksMinMax = (ylocs[1], ylocs[-2])

ax.spines['left'].set_bounds(yTicksMinMax[0],yTicksMinMax[1])
ax.spines['right'].set_bounds(yTicksMinMax[0],yTicksMinMax[1])
ax.spines['bottom'].set_bounds(xTicksMinMax[0],xTicksMinMax[1])

ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('both')

# set the tick labels
ax.tick_params(labeltop=False, labelright=True)

print("best amplitude: ", est_amp)
print("best frequency: ", est_freq)
print("best phase: ", est_phase)
print("best mean: ", est_mean)

return fig
```

## Decreasing sinus regression

```
def decreasingSinusRegression(xAxis, yAxis, guessAmp, guessPower,
    guessFreq, guessPhase):

    guess_amp = guessAmp #0.007
    guess_power = guessPower #0.5
    guess_freq = guessFreq #3.5
    guess_phase = guessPhase #1.4
    guess_mean = np.mean(yAxis)

    # first estimates
```

```

data_first_guess = guess_amp/(xAxis**guess_power)*np.sin(
    guess_freq*xAxis+guess_phase) + guess_mean

# define the function to optimize, minimizing the difference
# between the actual data and our guessed parameters
optimize_func = lambda x: x[0]/(xAxis**x[1])*np.sin(x[2]*xAxis+x
    [3]) + x[4] - yAxis
est_amp, est_power, est_freq, est_phase, est_mean = leastsq(
    optimize_func, [guess_amp, guess_power, guess_freq,
    guess_phase, guess_mean])[0]

# recreate the fitted curve using the optimized parameters
data_fit = est_amp/(xAxis**est_power)*np.sin(est_freq*xAxis+
    est_phase) + est_mean

# define the geometry in which to place the subplots
grid = plt.GridSpec(1, 1)

# create the figure environment for the plots
fig = plt.figure(figsize=(12, 7))

# set the plot location plots
ax = fig.add_subplot(grid[0,0])

plt.plot(xAxis, yAxis, ".")
plt.plot(xAxis, data_first_guess, label='first guess')
plt.plot(xAxis, data_fit, label='best fit')

# set the label for the bottom subplot
ax.set_xlabel('calibrated time from peak (s)')

# create lines for the x-axes
#axhline(xmin=0, xmax=1, linewidth=0.5, c='k')

# turn off some spines
ax.spines['top'].set_color('none')

# create arrays with the automated locations of the ticks
xlocs = ax.get_xticks()
ylocs = ax.get_yticks()

# create arrays with the boundary values of the ticks
xTicksMinMax = (xlocs[1], xlocs[-2])
yTicksMinMax = (ylocs[1], ylocs[-2])

ax.spines['left'].set_bounds(yTicksMinMax[0],yTicksMinMax[1])
ax.spines['right'].set_bounds(yTicksMinMax[0],yTicksMinMax[1])
ax.spines['bottom'].set_bounds(xTicksMinMax[0],xTicksMinMax[1])

ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('both')

# set the tick labels
ax.tick_params(labeltop=False, labelright=True)

```

```
print("best amplitude: ", est_amp)
print("best power: ", est_power)
print("best frequency: ", est_freq)
print("best phase: ", est_phase)
print("best mean: ", est_mean)

return fig
```

## 7.2.5 Graphical display

### Top all - bottom selected plot

```
def topAllBottomSelectedPlot(filename, dataTop, dataBottom, axisBottom,
    sharexYesNo):

    # update the filenames by adding the study axis at the end
    filename = filename + "_" + axisBottom.upper()

    # set the typical axes colors
    xColor = 'b'
    yColor = 'g'
    zColor = 'r'
    axesColors = [xColor, yColor, zColor]

    # Define the framework for the plots
    grid = plt.GridSpec(2, 1, wspace=0.4, hspace=0.3)

    # create the figure environment for the plots
    fig = plt.figure(figsize=(16, 7))

    # top row for the superposed plots
    ax1 = fig.add_subplot(grid[0,0])

    # bottom row for the selected plot
    if sharexYesNo == 'Yes':
        ax2 = fig.add_subplot(grid[1,0], sharex=ax1)

    if sharexYesNo == 'No':
        ax2 = fig.add_subplot(grid[1,0])

    # set a title for the figure
    fig.suptitle(filename)

    # create a list to get the axis' index
    coordinates = ['x', 'y', 'z']

    #data.iloc[:, coordinates.index(i)]

    # run a for-loop over the 3 axes
    for i in coordinates:

        # plot the x acceleration
        ax1.plot(dataTop.index, dataTop['{} AxisAcceleration '.
            format(i)], label='{} acceleration '.format(i), c=
            axesColors[coordinates.index(i)], alpha=0.4)
        ax1.plot(dataTop.index, dataTop['filtered {}
            AxisAcceleration '.format(i)], label='filtered {}
```

```

        acceleration '.format(i), linewidth=0.5, c=axesColors[
        coordinates.index(i)], alpha=0.95)

    # select the axis to work with
    if axisBottom == i:
        selectedColumn = coordinates.index(i)

# set the label for the top subplot
ax1.set_xlabel('global time of measurements (s)')

# plot the original selected acceleration
ax2.plot(dataBottom.index, dataBottom['{} AxisAcceleration '.format(
axisBottom)], label='{} acceleration '.format(axisBottom),
        linewidth=0.3, alpha=0.3, c=axesColors[selectedColumn])

# plot the filtered selected acceleration
ax2.plot(dataBottom.index, dataBottom['filtered {} AxisAcceleration
'.format(axisBottom)], label='filtered {} acceleration '.format
(axisBottom), linewidth=0.8, c=axesColors[selectedColumn])

# set the label for the bottom subplot
ax2.set_xlabel('calibrated time from peak (s)')

# create lines for the x-axes
ax1.axhline(xmin=0, xmax=1, linewidth=0.5, c='k')
ax2.axhline(xmin=0, xmax=1, linewidth=0.5, c='k')

# turn off some spines
ax1.spines['top'].set_color('none')
ax2.spines['top'].set_color('none')

# create arrays with the automated locations of the ticks
x1locs = ax1.get_xticks()
x2locs = ax2.get_xticks()

y1locs = ax1.get_yticks()
y2locs = ax2.get_yticks()

# create arrays with the boundary values of the ticks
x1TicksMinMax = (x1locs[1], x1locs[-2])
x2TicksMinMax = (x2locs[1], x2locs[-2])

y1TicksMinMax = (y1locs[1], y1locs[-2])
y2TicksMinMax = (y2locs[1], y2locs[-2])

# set the bounds for the displayed spines
ax1.spines['left'].set_bounds(y1TicksMinMax[0], y1TicksMinMax[1])
ax1.spines['right'].set_bounds(y1TicksMinMax[0], y1TicksMinMax[1])
ax1.spines['bottom'].set_bounds(x1TicksMinMax[0], x1TicksMinMax[1])

ax2.spines['left'].set_bounds(y2TicksMinMax[0], y2TicksMinMax[1])
ax2.spines['right'].set_bounds(y2TicksMinMax[0], y2TicksMinMax[1])
ax2.spines['bottom'].set_bounds(x2TicksMinMax[0], x2TicksMinMax[1])

# set the ticks for both axes
ax1.xaxis.set_ticks_position('bottom')
ax2.xaxis.set_ticks_position('bottom')

```



```
ax1.yaxis.set_ticks_position('both')
ax2.yaxis.set_ticks_position('both')

# set the tick labels
ax1.tick_params(labeltop=False, labelright=True)
ax2.tick_params(labeltop=False, labelright=True)

# add a legend showing the labels of the plots
ax1.legend(ncol=3)
ax2.legend()

return fig, filename, axisBottom
```

### Top all - bottom selected plot solo

```
def topAllBottomSelectedPlotSolo(filename, dataTop, dataBottom, axisBottom
, sharexYesNo):

    # update the filenames by adding the study axis at the end
    filename = filename + "_" + axisBottom.upper()

    # set the typical axes colors
    xColor = 'b'
    yColor = 'g'
    zColor = 'r'
    axesColors = [xColor, yColor, zColor]

    # Define the framework for the plots
    grid = plt.GridSpec(2, 1, wspace=0.4, hspace=0.3)

    # create the figure environment for the plots
    fig = plt.figure(figsize=(12, 7))

    # top row for the superposed plots
    ax1 = fig.add_subplot(grid[0,0])

    # bottom row for the selected plot
    if sharexYesNo == 'Yes':
        ax2 = fig.add_subplot(grid[1,0], sharex=ax1)

    if sharexYesNo == 'No':
        ax2 = fig.add_subplot(grid[1,0])

    # set a title for the figure
    #fig.suptitle(filename)

    # create a list to get the axis' index
    coordinates = ['x', 'y', 'z']

    #data.iloc[:, coordinates.index(i)]

    # run a for-loop over the 3 axes
    for i in coordinates:

        # plot the x acceleration
```

---

```

ax1.plot(dataTop.index, dataTop['{} AxisAcceleration '.format(i)], label='{} acceleration '.format(i), c=axesColors[coordinates.index(i)], alpha=0.95)

# select the axis to work with
if axisBottom == i:
    selectedColumn = coordinates.index(i)

# set the label for the top subplot
ax1.set_xlabel('global time of measurements (s)')

# plot the original selected acceleration
ax2.plot(dataBottom.index, dataBottom['{} AxisAcceleration '.format(axisBottom)], label='{} acceleration '.format(axisBottom), linewidth=0.3, alpha=0.5, c=axesColors[selectedColumn])

# plot the filtered selected acceleration
ax2.plot(dataBottom.index, dataBottom['filtered {} AxisAcceleration '.format(axisBottom)], label='filtered {} acceleration '.format(axisBottom), linewidth=0.8, c=axesColors[selectedColumn])

# set the label for the bottom subplot
ax2.set_xlabel('calibrated time from peak (s)')

# # create lines for the x-axes
# ax1.axhline(xmin=0, xmax=1, linewidth=0.5, c='k')
# ax2.axhline(xmin=0, xmax=1, linewidth=0.5, c='k')

# turn off some spines
ax1.spines['top'].set_color('none')
ax2.spines['top'].set_color('none')

# create arrays with the automated locations of the ticks
x1locs = ax1.get_xticks()
x2locs = ax2.get_xticks()

y1locs = ax1.get_yticks()
y2locs = ax2.get_yticks()

# create arrays with the boundary values of the ticks
x1TicksMinMax = (x1locs[1], x1locs[-2])
x2TicksMinMax = (x2locs[1], x2locs[-2])

y1TicksMinMax = (y1locs[1], y1locs[-2])
y2TicksMinMax = (y2locs[1], y2locs[-2])

# set the bounds for the displayed spines
ax1.spines['left'].set_bounds(y1TicksMinMax[0], y1TicksMinMax[1])
ax1.spines['right'].set_bounds(y1TicksMinMax[0], y1TicksMinMax[1])
ax1.spines['bottom'].set_bounds(x1TicksMinMax[0], x1TicksMinMax[1])

ax2.spines['left'].set_bounds(y2TicksMinMax[0], y2TicksMinMax[1])
ax2.spines['right'].set_bounds(y2TicksMinMax[0], y2TicksMinMax[1])
ax2.spines['bottom'].set_bounds(x2TicksMinMax[0], x2TicksMinMax[1])

# set the ticks for both axes
ax1.xaxis.set_ticks_position('bottom')
```

```
ax2.xaxis.set_ticks_position('bottom')
ax1.yaxis.set_ticks_position('both')
ax2.yaxis.set_ticks_position('both')

# set the tick labels
ax1.tick_params(labeltop=False, labelright=True)
ax2.tick_params(labeltop=False, labelright=True)

# add a legend showing the labels of the plots
ax1.legend(ncol=1)
ax2.legend()

return fig, filename, axisBottom
```

### Left/right all - bottom selected plot

```
def leftRightAllBottomSelectedPlot(filenameL, filenameR, dataTopLeft,
    dataMiddleLeft, dataTopRight, dataMiddleRight, axisMiddleLeft,
    axisMiddleRight):

    # update the filenames by adding the study axes at the end
    filenameL = filenameL + "_" + axisMiddleLeft.upper()
    filenameR = filenameR + "_" + axisMiddleRight.upper()

    # set the typical axes colors
    xColorL = 'b'
    yColorL = 'g'
    zColorL = 'r'
    axesColorsL = [xColorL, yColorL, zColorL]
    xColorR = 'c'
    yColorR = 'y'
    zColorR = 'm'
    axesColorsR = [xColorR, yColorR, zColorR]

    # define the geometry in which to place the subplots
    grid = plt.GridSpec(3, 2, wspace=0.4, hspace=0.3)

    # create the figure environment for the plots
    fig = plt.figure(figsize=(16, 7))

    # left column for the L plots
    ax1 = fig.add_subplot(grid[0,0])
    ax2 = fig.add_subplot(grid[1,0])

    # right column for the R plots
    ax3 = fig.add_subplot(grid[0,1])
    ax4 = fig.add_subplot(grid[1,1])

    # bottom row for the superposed plots
    ax5 = fig.add_subplot(grid[2,:])

    # create an array with all the axes
    ax = [ax1, ax2, ax3, ax4, ax5]

    # set a title for the figure
    ax1.set_title(filenameL)
    ax3.set_title(filenameR)
```

```

# create a list to get the axis' index
coordinates = ['x', 'y', 'z']

# run a for-loop over the 3 axes for the L plots
for i in coordinates:

    # plot all the L accelerations
    ax1.plot(dataTopLeft.index, dataTopLeft['{ }
              AxisAcceleration '.format(i)], label='{ } acc '.format(i)
              , c=axesColorsL[coordinates.index(i)], alpha=0.4)
    ax1.plot(dataTopLeft.index, dataTopLeft['filtered { }
              AxisAcceleration '.format(i)], label='filt { } acc '.
              format(i), linewidth=0.5, c=axesColorsL[coordinates.
              index(i)], alpha=0.95)

    # select the axis to work with
    if axisMiddleLeft == i:
        selectedColumnL = coordinates.index(i)

# plot the original selected L acceleration
ax2.plot(dataMiddleLeft.index, dataMiddleLeft['{ } AxisAcceleration
        '.format(axisMiddleLeft)], label='{ } acc '.format(axisMiddleLeft
        ), linewidth=0.3, alpha=0.3, c=axesColorsL[selectedColumnL])

# plot the filtered selected L acceleration
ax2.plot(dataMiddleLeft.index, dataMiddleLeft['filtered { }
        AxisAcceleration '.format(axisMiddleLeft)], label='filt { } acc
        '.format(axisMiddleLeft), linewidth=0.8, c=axesColorsL[
        selectedColumnL])

# run a for-loop over the 3 axes for the R plots
for i in coordinates:

    # plot all the R acceleration
    ax3.plot(dataTopRight.index, dataTopRight['{ }
              AxisAcceleration '.format(i)], label='{ } acc '.format(i)
              , c=axesColorsR[coordinates.index(i)], alpha=0.4)
    ax3.plot(dataTopRight.index, dataTopRight['filtered { }
              AxisAcceleration '.format(i)], label='filt { } acc '.
              format(i), linewidth=0.5, c=axesColorsR[coordinates.
              index(i)], alpha=0.95)

    # select the axis to work with
    if axisMiddleRight == i:
        selectedColumnR = coordinates.index(i)

# plot the original selected R acceleration
ax4.plot(dataMiddleRight.index, dataMiddleRight['{ }
        AxisAcceleration '.format(axisMiddleRight)], label='{ } acc '.
        format(axisMiddleRight), linewidth=0.3, alpha=0.3, c=
        axesColorsR[selectedColumnR])

# plot the filtered selected R acceleration
ax4.plot(dataMiddleRight.index, dataMiddleRight['filtered { }
        AxisAcceleration '.format(axisMiddleRight)], label='filt { } acc

```

```
        '.format(axisMiddleRight), linewidth=0.8, c=axesColorsR[
        selectedColumnR])

# create a plot with both the L and R original and filtered
# selected accelerations
ax5.plot(dataMiddleLeft.index, dataMiddleLeft['{} AxisAcceleration
'.format(axisMiddleLeft)], label='{} acc '.format(axisMiddleLeft)
), linewidth=0.3, alpha=0.8, c=axesColorsL[selectedColumnL])
ax5.plot(dataMiddleLeft.index, dataMiddleLeft['filtered {}
AxisAcceleration '.format(axisMiddleLeft)], label='filt {} acc
'.format(axisMiddleLeft), linewidth=0.8, alpha=0.7, c=
axesColorsL[selectedColumnL])
ax5.plot(dataMiddleRight.index, dataMiddleRight['{}
AxisAcceleration '.format(axisMiddleRight)], label='{} acc '.
format(axisMiddleRight), linewidth=0.3, alpha=0.8, c=
axesColorsR[selectedColumnR])
ax5.plot(dataMiddleRight.index, dataMiddleRight['filtered {}
AxisAcceleration '.format(axisMiddleRight)], label='filt {} acc
'.format(axisMiddleRight), linewidth=0.8, alpha=0.7, c=
axesColorsR[selectedColumnR])

# set the label for the bottom subplot
ax5.set_xlabel('calibrated time from peak (s)')

# create lines for the x-axes
for i in range(len(ax)) :
    ax[i].axhline(xmin=0, xmax=1, linewidth=0.5, c='k')

# turn off some spines
for i in range(len(ax)):
    ax[i].spines['top'].set_color('none')

# create arrays with the automated locations of the ticks

x1locs = ax1.get_xticks()
x2locs = ax2.get_xticks()
x3locs = ax3.get_xticks()
x4locs = ax4.get_xticks()
x5locs = ax5.get_xticks()

y1locs = ax1.get_yticks()
y2locs = ax2.get_yticks()
y3locs = ax3.get_yticks()
y4locs = ax4.get_yticks()
y5locs = ax5.get_yticks()

# create arrays with the boundary values of the ticks
x1TicksMinMax = (x1locs[1], x1locs[-2])
x2TicksMinMax = (x2locs[1], x2locs[-2])
x3TicksMinMax = (x3locs[1], x3locs[-2])
x4TicksMinMax = (x4locs[1], x4locs[-2])
x5TicksMinMax = (x5locs[1], x5locs[-2])

y1TicksMinMax = (y1locs[1], y1locs[-2])
y2TicksMinMax = (y2locs[1], y2locs[-2])
y3TicksMinMax = (y3locs[1], y3locs[-2])
y4TicksMinMax = (y4locs[1], y4locs[-2])
```

```

y5TicksMinMax = (y5locs[1], y5locs[-2])

xTicksMinMax = [x1TicksMinMax, x2TicksMinMax, x3TicksMinMax,
                x4TicksMinMax, x5TicksMinMax]
yTicksMinMax = [y1TicksMinMax, y2TicksMinMax, y3TicksMinMax,
                y4TicksMinMax, y5TicksMinMax]

# set the bounds for the displayed spines
for i in range(len(ax)):
    ax[i].spines['left'].set_bounds(yTicksMinMax[i][0],
                                   yTicksMinMax[i][1])
    ax[i].spines['right'].set_bounds(yTicksMinMax[i][0],
                                     yTicksMinMax[i][1])
    ax[i].spines['bottom'].set_bounds(xTicksMinMax[i][0],
                                      xTicksMinMax[i][1])

# set the ticks for both axes
for i in range(len(ax)):
    ax[i].xaxis.set_ticks_position('bottom')
    ax[i].yaxis.set_ticks_position('both')

# set the tick labels
for i in range(len(ax)):
    ax[i].tick_params(labeltop=False, labelright=True)

# add a legend showing the labels of the plots
ax1.legend(ncol=3)
ax2.legend()
ax3.legend(ncol=3)
ax4.legend()
ax5.legend(ncol=2)

return fig, filenameL, filenameR, axisMiddleLeft, axisMiddleRight

```

## 7.2.6 Miscellaneous

## 7.2.7 Libraries and main code

At the start of the main code :

```

import numpy as np

import matplotlib
matplotlib.use('Qt5Agg')
import matplotlib.pyplot as plt
from matplotlib import gridspec

import pandas as pd

import scipy
from scipy import signal
from scipy.optimize import leastsq

```

At the end of the main code :

```

# show the plots
plt.show()

```

```
# close the current figure , axes and figure window
plt.clf()
plt.cla()
plt.close()
```

## Save figure

```
def saveFigure(filename , fig , saveFigureYesNo , doubleYesNo , filenameDouble
):
    if doubleYesNo == "Yes":
        # take whatever is common in the two filenames and save it
        common = ""
        for i in filename[:-3]:
            if i in filenameDouble[:-3]:
                common += i

        # define a new name with the common parts and the
        # calibration and study axes for each of the two files
        # together
        commonFilename = common + 'L' + filename[-3:-2] + filename
        [-1:] + '_' + 'R' + filenameDouble[-3:-2] +
        filenameDouble[-1:]

        # remove the double '_' that arised from the common-filter
        # operation
        commonFilename = commonFilename[:commonFilename.find('__')]
        + commonFilename[commonFilename.find('__')+1:]
        print('\n' + 'A common filename has been created out of
        the two initial filenames')

        # save the created figure as a file with the common name
        if saveFigureYesNo == "Yes":
            fig.savefig(commonFilename + ".jpg")
            print('\n' + "The file has been saved.")
        if saveFigureYesNo == "No":
            print('\n' + "The file has not been saved.")

    if doubleYesNo == "No":
        # save the created figure as a file with the same name
        if saveFigureYesNo == "Yes":
            fig.savefig(filename + ".jpg")
            print('\n' + "The file has been saved.")
        if saveFigureYesNo == "No":
            print('\n' + "The file has not been saved.")

    commonFilename = ""

    return commonFilename
```

## Plot informations

```
def plotInformations(filename , data , samplingFrequency ,
    calibratedSamplingFrequency , calibratedCutoffFrequency ,
    calibratedFilterOrder , cutoffFrequency , filterOrder , calibrationAxis ,
```

studyAxis, cutFromStart, cutFromFinish, saveTextYesNo, doubleYesNo, commonFilename, appendTruncate):

```
# create a banner for the information text
informationsText = "Informations on the initial dataset:"

# put together all the informations to print
informationsTextTotal = '\n' + informationsText + '\n' + "-" * len(
    informationsText) + '\n' + "Filename: " + filename + '\n \n'
    + "Original sampling frequency: " + str(samplingFrequency) +
    '\n' + "Original Butterworth cut-off frequency: " + str(
    cutoffFrequency) + '\n' + "Original Butterworth filter order:
    " + str(filterOrder) + '\n \n' + "Calibrated sampling
    frequency: " + str(calibratedSamplingFrequency) + '\n' + "
    Calibrated Butterworth cut-off frequency: " + str(
    calibratedCutoffFrequency) + '\n' + "Calibrated Butterworth
    filter order: " + str(calibratedFilterOrder) + '\n \n' + "[
    xMin —äxMax] = [" + str(data.xAxisAcceleration.min()) + " —
    " + str(data.xAxisAcceleration.max()) + "]" + '\n' + "[yMin —
    äyMax] = [" + str(data.yAxisAcceleration.min()) + " — " + str
    (data.yAxisAcceleration.max()) + "]" + '\n' + "[zMin —äzMax]
    = [" + str(data.zAxisAcceleration.min()) + " — " + str(data.
    zAxisAcceleration.max()) + "]" + '\n \n' + "Calibration axis:
    " + calibrationAxis + '\n' + "Axis of interest: " + studyAxis
    + '\n \n' + "Cut from start: " + str(cutFromStart) + '\n' + "
    Cut from finish: " + str(cutFromFinish) + '\n'

# print informations on plot
print('\n',
      informationsTextTotal,
      '\n')

if saveTextYesNo == "Yes":
    if doubleYesNo == "No":
        # create file with the printed informations
        informationsTextFile = open('{}.txt'.format(
            filename), appendTruncate)
        informationsTextFile.write(informationsTextTotal)
        informationsTextFile.close()
        print('\n' + "The text file with the plot
            informations has been created.")
    if doubleYesNo == "Yes":
        # create file and append printed informations to
        it
        informationsTextFile = open('{}.txt'.format(
            commonFilename), appendTruncate)
        informationsTextFile.write(informationsTextTotal +
            '\n')
        informationsTextFile.close()
        print('\n' + "The text file with the plot
            informations has been created.")
if saveTextYesNo == "No":
    print('\n' + "The text file with the plot informations has
        neither been created nor has overwritten the previous
        one.")

return
```



## Map and position of the camp in Vallunden

```
from mpl_toolkits.basemap import Basemap
import numpy as np
import matplotlib.pyplot as plt

fig=plt.figure()
ax=fig.add_axes([0.1,0.1,0.8,0.8])

# setup mercator map projection.
m = Basemap(llcrnrlon=16.1, llcrnrlat=77.76, urcrnrlon=17.2, urcrnrlat=77.95,
            resolution='h', projection='merc')

m.drawcoastlines()
m.fillcontinents()

#SveaCamp
lat1 = 77.88617
lon1 = 16.77295

#77.88617
#16.77295

x1, y1 = m.projtran(lon1, lat1)

# draw parallels
m.drawparallels([77.76, lat1, 77.9499999], labels=[1,0,0,0])

# draw meridians
m.drawmeridians([16.1, lon1, 17.1999999], labels=[0,0,0,1])

m.plot(x1, y1, 'ro', marker=(4,2,0), markersize=10)

plt.show()
```

# Bibliography

- [1] Matti Leppäranta. *The Drift of Sea Ice*. Springer, second edition, 2011.
- [2] Michel Rieutord. *Fluid Dynamics: an Introduction*. Graduate Texts in Physics. Springer, 2015.
- [3] Martin Kunz. Macroscopic description of continuous media: thermodynamics, fluid dynamics and elasticity. Lecture notes for the course Media Studies at the University of Geneva, 2015.
- [4] Gabriele Morra. *Pythonic Geodynamics: Implementations for fast computing*. Lecture Notes in Earth System Sciences. Springer, 2018.
- [5] Alexandra Techet. Potential flow theory. Lecture notes for the course 2.016 Hydrodynamics at MIT, 2005.
- [6] Alexandra Techet. Derivation of unsteady bernoulli's equation. Lecture notes for the course 2.016 Hydrodynamics at MIT, 2005.
- [7] Rinat Mukhametianov Aleksandra Visich Daniela Klein, Terese Od. At-322 report. Technical Report October, The University Centre in Svalbard, 2018.
- [8] Aleksey Marchenko. Added masses and forces on body in a flow with acceleration. Lecture notes for the course AT-322 at the University centre in Svalbard.
- [9] Takahashi Takeo Matthieu Hillairet. Collisions in three-dimensional fluid structure interaction problems. *Society for Industrial and Applied Mathematics*, 40(6):2451–2477, 2009.

- [10] Rocky Taylor Aleksey Marchenko, Vladimir Markov. Influence of water on collisions of floating ice blocks. *Proceedings of the Twenty-ninth (2019) International Ocean and Polar Engineering Conference*, 2019.
- [11] Matthew Hancock. The 1-d heat equation the 1-d heat equation. *Lecture notes for the course Linear Partial Differential Equations at MIT*, 2006.
- [12] Dytran Instruments Incorporation. Dytran 4000a1 vibration recorder operating guide. Technical report, Dynamic Transducers and Systems, 2014.
- [13] Matjaz Gams Hristijan Gjoreski. Accelerometer data preparation for activity recognition. 2011.
- [14] Weichuan Yu Chao Yang, Zengyou He. Comparison of public peak detection algorithms for maldi mass spectrometry data analysis. *BMC Bioinformatics*, 2009.
- [15] P. Pieter. Butterworth filters, 2018.
- [16] Josef Goette. An iir-filter example : A butterworth filter, 2018.
- [17] Donal Watts Douglas Bates. *Nonlinear Regression Analysis and Its Applications*. Wiley Series in Probability and Statistics. Wiley, 1988.
- [18] Peter Dourmashkin. Classical mechanics: Mit 8.01 course notes. Lecture notes for the course Physics 8.01: Classical Mechanics at MIT, 2013.
- [19] Alexandra Techet. Derivation of added mass around a sphere, cylinder. Lecture notes for the course 2.016 Hydrodynamics at MIT, 2005.
- [20] Christopher Brennen. *An Internet Book on Fluid Dynamics*. 2006.
- [21] Tracker 5.0 help. Technical report.
- [22] S. Skiaker A. Tremblett M. Hovind H. Troite, V. Kröger. Engineering properties of sea ice. Technical report, The University Centre in Svalbard, 2019.
- [23] A. Tremblett M. Hovind V Kröger, S. Skiaker. Properties of ice in engineering. Technical report, The University Centre in Svalbard, 2019.
- [24] J. Wahlin O. Johannessen V. Alexandrov, S. Sandven. The relation between sea ice thickness and freeboard in the arctic. *Cryosphere*, 4(3):373–380, 2010.
- [25] Kai Griebel Nikita Turko, Brian McDonald. Report on wave mechanics. Technical Report October, The University Centre in Svalbard, 2018.

- [26] Victor Starovoitov. Behavior of a rigid body in an incompressible viscous fluid near a boundary. *International Series of Numerical Mathematics*, 2003.
- [27] J. Morison M. Mcphee. Under-ice boundary layer. 2001.
- [28] Johannes Mosig Michael Meylan, Luke Bennetts. Dispersion relations, power laws, and energy loss for waves in the marginal ice zone. *Journal of Geophysical Research: Oceans*, 2018.
- [29] Nicolas Lecoq. Boundary conditions for creeping flow along periodic or random rough surfaces : experimental and theoretical results. *Journal of Physics: Conference Series*, 392, 2012.
- [30] David Griffiths. *Introduction to Electrodynamics*. Pearson, fourth edition edition, 2013.
- [31] Richard Timoney. Lebesgue integral. Lecture notes for the course MA 2224 at Trinity College Dublin, 2016.
- [32] Arthur Mattuck. The divergence theorem. Lecture notes for the course 18.02SC Multivariable Calculus at MIT, 2010.