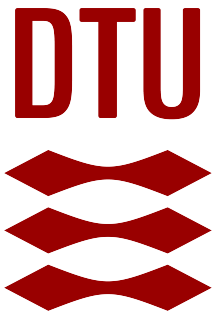
Experimental acoustofluidics for automated handling of cells



Bachelor project
Philip Jensen (s102772) and Joakim Edsen Pihl (s203921)

Supervised by
Professor Henrik Bruus
Associate Professor Massimiliano Rossi

Department of Physics
Technical University of Denmark
19th of January 2024

Contents

1 Abstract **1**

2 Preface **2**

3 Introduction **3**

4 Theory **4**
4.1 Notation . . . 4
4.2 Governing equations . . . 4
4.3 Acoustofluidics and perturbation theory . . . 4
4.4 Resonance frequency . . . 7
4.5 Resistance . . . 8
4.6 Sedimentation . . . 10
4.7 Compliance . . . 11
4.7.1 Apoximation of the compliance in a silicone tube . . . 12

5 Development of the experimental setup **16**
5.1 Initial setup . . . 16
5.2 Motivational experiment . . . 16
5.3 The device holder (DH) . . . 17
5.3.1 Initial DH . . . 17
5.3.2 New DH with rotation (RDH) . . . 18
5.4 Characterization of the capillary tube (No yield experiment) . . . 19
5.5 Sedimentation . . . 21
5.6 Changes to the setup . . . 22
5.6.1 Tubing . . . 22
5.6.2 Short circuit . . . 22
5.6.3 Camera . . . 23

6 Video analysis **24**

7 Main experiments **28**
7.1 Automated flow experiment . . . 28
7.2 Estimation of uncertainty on t^* . . . 29
7.3 Resonance frequency of the capillary tube . . . 30
7.4 Correlation between voltage and t^* . . . 34
7.5 Distribution of t^* at resonance frequency . . . 36

8 Discussion **39**
8.1 Uncertainty on t^* . . . 39
8.2 First resonance mode of the system . . . 40
8.3 Correlation between voltage and t^* . . . 40
8.4 Distribution of t^* at resonance . . . 40
8.5 Separation of algae . . . 41

9 Conclusion and outlook **42**

10 References **43**

A Sedimentation **44**

B Pre-experimental routine **45**

C Uncertainty table **47**

D Impedance sweep **48**

E Scripts **49**
E.1 Compliance 49
E.2 Surviving experiment 52
E.3 AD function 55
E.4 Gauss distribution calculation 60
E.5 Pressure due to compliance calculation 65
E.6 Sedimentation calculation 68
E.7 Temperature dependence calculation 71
E.8 Valve control 74
E.9 Cetoni script 76
E.10 Video analysis script 78

1 Abstract

The bachelor thesis explores the field of acoustofluidics, a combination of microfluidics and acoustics, to automate the handling of cells. The focus is on using acoustofluidics to focus algae in a microchannel, aiming to infer biophysical characteristics of the algae-based on their response to acoustics. This is done by creating a pressure gradient throughout the microchannel with a piezoelectric transducer. The transducer vibrates according to the frequency and amplitude applied to it. The width of the microchannel determines the resonance frequency of the microchannel. If a frequency near resonance is chosen with a suitable amplitude the algae will move to (focus in) the center of the microchannel. The focus time depends on the frequency and amplitude of the signal, but also on the biophysical parameters related to the algae. It is therefore hypothesized that something can be inferred about the biology of the algae if the other parameters can be controlled. To study the focus time a light intensity model was used. The algae were situated in the microchannel with a background light. When applying the signal the algae would focus into the center of the channel making the intensity increase in the outer regions of the microchannel. The change in light intensity over time was linked by the model to the focus time of the algae

A large part of the thesis revolved around making improvements to an experimental setup to automate the handling of the cells. This included, amongst other things, implementing a new device holder for the capillary tube with rotational capabilities and an automated switch valve. Four main experiments were made using the improved setup. (1) It was found that the relative uncertainties in the focus times do not depend on the frequency of the signal. This was done by running multiple experiments at a range of frequencies, where repeated measurements of the focus time were made. It was then seen that the variation in the data did not depend on the frequency. The uncertainty on the focus time was found to be 6%. (2) A high-definition resonance frequency was found to be 1.898 MHz. This was done with a fine frequency sweep around the presumed resonance frequency. This was then fitted to a Lorentzian, and the true resonance frequency was extracted. (3) It was shown that the focus time is proportional to the inverse square on the voltage applied from the signal. This was done by doing a sweep in voltage at the resonance frequency. It was found that the behavior was in agreement with studies on inorganic particles. The result showed that the inverse focus time increased by $7.5 \times 10^{-2} \frac{1}{\mathrm{s\,V^2}}$ to $8.6 \times 10^{-2} \frac{1}{\mathrm{s\,V^2}}$ depending on the concentration of the algae. (4) By repeating the acoustic focusing of the algae 500 times, at the resonance frequency, the focus time was shown to be normally distributed. The standard deviation varied from 5.9×10^{-2} s to 3.4×10^{-2} s depending on the correction of the data.

The findings in the thesis have given new insights into how the handling and separation of the algae can be done. There are however still several obstacles to overcome before the automated handling setup can be realized. The environmental effects on the setup have to be controlled before moving forward with separation. The setup has to be able to calculate the focus times in real time for the experiment to be feasible. This is something that might require upgrading the hardware if the computational time cannot be cut. The effects on small algae vs. large algae have to be studied to see if the method can handle different algae sizes. This is something that might require a pre-selection before the acoustic sorting.

2 Preface

This bachelor's thesis is submitted as a partial fulfillment of the requirements for the Bachelor of Science degree in Physics and nanotechnology at the Technical University of Denmark (DTU). The thesis, which corresponds to 15 ECTS points, is the result of work which was carried out between the start of October 2023 and the end of January 2024. The work was carried out at the DTU Department of Physics in the Theoretical Microfluidics group (TMF), headed by Professor Henrik Bruus.

We want to thank our supervisors Henrik Bruus, and Massimiliano Rossi Associate Professor with the University of Bologna for introducing us, with great enthusiasm, to both the theoretical and experimental aspects of acoustofluidics, as well as for their guidance throughout this thesis. We would also like to thank former and current members of TMF, Gustav K. Modler, Valdemar Frederiksen, and Søren Kuhberg, for providing valuable insight into the theoretical aspects of the thesis and joyful coffee breaks. We would also like to extend our gratitude to other members of the Fluids section at the Department of Physics at DTU, especially postdoc Matthew Dominic Biviano, for providing various aspects regarding the experimental work.

We would also like to thank assistant engineer Erik Hansen, for his advisory and quick manufacturing of the new device holder.

Philip Jensen

Joakim Edsen Pihl

Department of Physics
Technical University of Denmark
19th of January 2024

3 Introduction

The future calls for renewable fuel sources to counteract climate change. A growing field in this regard is the production of bio fuels. Bio fuels can not as this point compete with conventional fuels, especially when it comes to the production cost. If bio fuels are to replace the conventional fuels, in the areas where other sources of renewable can suffice, the production cost needs to be reduced. One way of doing so is to optimize the energy consumption in the manufacturing process. The production of bio-fuel involves growing algae in large ponds. When the algae are mature for harvest lipids inside the algae are extracted. This is usually done be either mechanical- or chemical methods and both methods seek to weaken or break the membrane making the extraction possible. If the algae could be preselected in a way as to only extract from the ones with the weakest membrane, the energy expenditure in the extraction process could be reduced. It is also hypothesized that the selection could be used for a guided evolution of the algae to evolve strains of algae with weaker cell membranes.[8]

In acoustofluidics ultrasound wave can be used to focus particle in a micro channel. It has been shown in [4] that micro beads can be focused in a liquid filled micro tube by vibrating the tube at is resonance frequency. The time is takes for the micro bead to focus is calculated using an intensity model. In the project a similar approach has been utilized. Instead of micro beads unicellular algae of the species Chlorella Vulgaris are focused and their focus time are calculated using the same model. The idea behind repeating the experiment with algae is to show if the focus time of different algae can be linked to the biology of the algae itself. It is theorized that algae with a softer membrane dampen the force acting on them more than the ones with a hard membrane and thus the former focus slower. If these algae can be separated from the rest they can be examined to see if they exhibit differences in their membranes. If this is the case it could have a large impact on innovations with bio-fuel production.

The thesis is based on the development of an experimental setup and experiments done over the cause of the project. In chapter 4 "Theory" all relevant theories of acousto- and microfluidics, used in the thesis, are derived or shown. Chapter 5 "Development of the experimental setup" describes all improvements to the setup as well as experiments done at each state in the developmental process. The experiments here are catalysts for further improvements to the setup. The chapter is written in chronological order to show the timeline for the developmental- and experimental work. Chapter 6 "Video analysis" describes the main methods used for the data analysis. This is the many particle intensity approach (described above) as opposed to the single particle velocity tracking approach. In chapter 7 "main experiments" the four main experiments of the thesis are studied here. In chapter 8 "Discussion" all the result are summarized and interpreted and the implication and limitations of the findings are discussed. This is largely in regards to the four main experiments. In chapter 9 "Conclusion and outlook" gives a final summery of the findings as well as giving a prospective of the future research need to test this hypothesis. The appendix contain all relevant findings that did not make it in the thesis. This include all the code produced, but this can be easily attained from our GitHub repository as well: https://github.com/PhilPhilJ/Ultrasound_acoustofluidics

4 Theory

4.1 Notation

Throughout this bachelor's project, there will be made use of different kinds of notation. Firstly partial derivatives may be written as a single partial derivative sign along with a subscript that denotes which variable the derivative is made with regard to. A partial time derivative using this notation is seen in equation (1).

$$\frac{\partial}{\partial t} \equiv \partial_t \tag{1}$$

Vectors will be written in bold, thus the position vector $\vec{r}$, would be written as equation (2) in which $\mathbf{e_i}, i = x, y, z$ denotes the unit vectors in the x, y, and z direction.

$$\mathbf{r} = x\mathbf{e_x} + y\mathbf{e_y} + z\mathbf{e_z} \tag{2}$$

The time average of an arbitrary function $X(t)$ will be written as $\langle X \rangle$ implying equation (3) where T denotes one period of $X(t)$.

$$\langle X \rangle \equiv \frac{1}{T} \int_0^T dt X(t) \tag{3}$$

4.2 Governing equations

The world of fluid mechanics is generally governed by two key equations, the continuity equation and the Navier-Stokes equation. The continuity equation, (4), expresses the conservation of mass and it describes how the mass of an arbitrary volume, Ω, of fluid may change. As mass cannot spontaneously appear nor disappear a change in mass for a certain volume can only occur through the flux of mass through the surface of Ω.

$$\partial_t \rho = -\boldsymbol{\nabla} \cdot (\rho \boldsymbol{v}) \tag{4}$$

The second key equation is the Navier-Stokes equation, (5), which relates to the conservation of momentum. It describes how the momentum of a fluid volume, say Ω, may change.

$$\rho\big[\partial_t \boldsymbol{v} + (\boldsymbol{v} \cdot \boldsymbol{\nabla} \boldsymbol{v})\big] = -\boldsymbol{\nabla} p + \eta \boldsymbol{\nabla}^2 \boldsymbol{v} + \beta\eta \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \boldsymbol{v}) + \rho \boldsymbol{f} \tag{5}$$

For the sake of simplicity, only the adiabatic case is studied and all external body forces like gravity or electromagnetism are neglected equation (5) reduces to equation (6).

$$\rho\big[\partial_t \boldsymbol{v} + (\boldsymbol{v} \cdot \boldsymbol{\nabla} \boldsymbol{v})\big] = -\boldsymbol{\nabla} p + \eta \boldsymbol{\nabla}^2 \boldsymbol{v} + \beta\eta \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \boldsymbol{v}) \tag{6}$$

4.3 Acoustofluidics and perturbation theory

To further study acoustics in fluids, like water, perturbation theory has to be applied. For now, a first-order perturbation will be applied to each of the major fluid fields pressure, p, density, ρ, and velocity, $\boldsymbol{v}$, describing a quiescent fluid. This yields equations (7). The introduced constant $c_0 = (\partial p / \partial \rho)_0$ is the speed of sound in the liquid medium, this can be derived by Taylor expanding the thermodynamic equation of state of the pressure.

$$\rho = \rho_0 + \rho_1 \qquad p = p_0 + c_0^2 \rho_1 \qquad \boldsymbol{v} = \boldsymbol{v}_1 \tag{7}$$

Substituting these into the continuity equation (4) yields the first-order perturbed continuity equation 8.

$$\partial_t \rho_1 = -\rho_0 \boldsymbol{\nabla} \cdot \boldsymbol{v}_1 \tag{8}$$

Continuing this approach with equation (6) yields the first-order perturbed Navier-Stokes equation (9).

$$\rho_0 \partial_t \boldsymbol{v}_1 = -c_0^2 \boldsymbol{\nabla} \rho_1 + \eta \nabla^2 \boldsymbol{v}_1 + \beta \eta \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \boldsymbol{v}_1) \tag{9}$$

Taking the time derivative of equation (8) gives $\partial_t^2 \rho_1 = -\boldsymbol{\nabla} \cdot (\rho_0 \partial_t \boldsymbol{v_1})$ in which equation 9 can easily be substituted in yielding a single equation for ρ_1, (10).

$$\partial_t^2 \rho_1 = c_0^2 \nabla^2 \rho_1 - (1+\beta)\eta \nabla^2 (\boldsymbol{\nabla} \cdot \boldsymbol{v}_1) = c_0^2 \Big[1 + \frac{(1+\beta)\eta}{\rho_0 c_0^2} \partial_t \Big] \nabla^2 \rho_1 \tag{10}$$

Assuming harmonic time dependence of the fluid fields allows for further study of this phenomenon. It is to be noted that ω denotes the angular frequency of the fields, $\omega = 2\pi f$, where f is the frequency of the fields.

$$\rho_1(\boldsymbol{r}, r) = \rho_1(\boldsymbol{r}) \mathrm{e}^{-i\omega t} \qquad p_1(\boldsymbol{r}, t) = c_0^2 \rho_1(\boldsymbol{r}) \mathrm{e}^{-i\omega t} \qquad \boldsymbol{v}_1(\boldsymbol{r}, t) = \boldsymbol{v_1}(\boldsymbol{r}) \mathrm{e}^{-i\omega t} \tag{11}$$

Taking the time derivatives of equation (10) while simultaneously substituting ρ_1 for p_1 using (11) yields equation (12).

$$\Big[1 - i \frac{(1+\beta)\eta}{\rho_0 c_0^2} \omega \Big] \nabla^2 p_1(\boldsymbol{r}, t) = -\frac{\omega^2}{c_0^2} p_1(\boldsymbol{r}, t) \tag{12}$$

A constant, $k_0 = \omega / c_0$ is now defined. As the fraction within the brackets on the left side of equation (12) is, $\frac{(1+\beta)\eta}{\rho_0 c_0^2} \omega \approx 10^{-6} \ll 1$, it allows for the use of the approximation, $(1 - i2\Gamma) \approx (1 + i\Gamma)^{-2}$, defining Γ as, $\Gamma = \frac{(1+\beta)\eta}{2\rho_0 c_0^2} \omega$. This allows for rewriting equation (12) into equation (13).

$$\nabla^2 p_1(\boldsymbol{r}, t) = -k_0^2 (1 + i\Gamma)^2 p_1(\boldsymbol{r}, t) \tag{13}$$

It can be made clearer that equation (13) is Helmholtz equation by introducing the constant, $k = k_0(1 + i\Gamma)$. This results in equation (14) and thus it is seen that the theory allows for acoustic waves in fluids.

$$\nabla^2 p_1 = -k^2 p_1 \tag{14}$$

By studying the inviscid limit i.e. $\eta = 0$, equation (12) can be rewritten as (15).

$$\nabla^2 p_1 = -\frac{\omega^2}{c_0^2} p_1 \tag{15}$$

Each instance a time derivative is made of a complex exponential it results in the complex exponential multiplied by $-i\omega$ like so $\partial_t \exp(-i\omega t) = -i\omega \exp(-i\omega t)$. From this, it can be deduced that $\partial_t \to -i\omega$ implying that $\omega \to i\partial$. Using this with equation (15), one arrives at equation (16). Equation (16) is the general wave equation, this acts as a confirmation of c_0 being the speed of sound in the medium as was asserted at the beginning of section (4.3).

$$\nabla^2 p_1 = -\frac{1}{c_0^2} \partial_t^2 p_1 \tag{16}$$

In the inviscid limit the first-order perturbed Navier-Stokes equation (9) reduces to equation (17).

$$\rho_0 \partial_t \boldsymbol{v}_1 = -c_0^2 \boldsymbol{\nabla} \rho_1 \tag{17}$$

Using the assumption of harmonic fields from (11)

$$-i\omega\rho_0 \boldsymbol{v}_1 = -c_0^2 \boldsymbol{\nabla} \rho_1 \to \boldsymbol{v}_1 = -i \frac{c_0^2}{\omega \rho_0} \boldsymbol{\nabla} \rho_1 \tag{18}$$

Substituting ρ_1 for p_1, using $\rho_1 = (1/c_0^2)p_1$, yields equation (19).

$$\boldsymbol{v}_1 = -i\frac{1}{\omega\rho_0}\boldsymbol{\nabla}p_1 \tag{19}$$

Using (19) along with the relations from (11), it clearly shows that $\boldsymbol{v}_1$ can be described by the gradient of a potential field $\phi_1 = -\frac{i}{\omega\rho_0}p_1$ leading to equation (20).

$$\boldsymbol{v}_1 = \boldsymbol{\nabla}\phi_1 \tag{20}$$

Slower processes are best described using time-averaged continuum fields. The first-order harmonic fields time-average to equal zero and hence second-order perturbation theory becomes necessary. The continuum fields will now be described as seen in (21).

$$\rho = \rho_0 + \rho_1 + \rho_2 \qquad p = p_0 + p_1 + p_2 \qquad \boldsymbol{v} = \boldsymbol{v}_1 + \boldsymbol{v}_2 \tag{21}$$

Like p_1 in equation (7), p_2 can be expressed in terms of c_0, ρ_1, and ρ_2 by taking the Taylor expanding the thermodynamic equation of state to the second-order. This leads to expression (22)

$$p_2 = c_0^2\rho_2 + \frac{1}{2}(\partial_\rho c_0^2)\rho_1^2 \tag{22}$$

Substituting (21) into the continuity equation (4) and the Navier-Stokes equation (6) yields the second-order perturbed continuity equation (23) and Navier-Stokes equation (24).

$$\partial_t\rho_2 = -\rho_0\boldsymbol{\nabla}\cdot\boldsymbol{v}_2 - \boldsymbol{\nabla}\cdot(\rho_1\boldsymbol{v}_1) \tag{23}$$

$$\rho_0\partial_t\boldsymbol{v}_2 = -\boldsymbol{\nabla}p_2 + \eta\nabla^2\boldsymbol{v}_2 + \beta\eta\boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{v}_2) - \rho_1\partial_t\boldsymbol{v}_1 - \rho_0(\boldsymbol{v}_1\cdot\boldsymbol{\nabla})\boldsymbol{v}_1 \tag{24}$$

As a result of taking the time average of (23) and (24) no terms vanish as all terms containing a second-order field or a product of first-order fields time-averages to something non-zero. Thus the time-averaged continuity (25) and the Navier-Stokes equation (26) are.

$$\rho_0\boldsymbol{\nabla}\cdot\langle\boldsymbol{v}_2\rangle = -\boldsymbol{\nabla}\cdot\langle\rho_1\boldsymbol{v}_1\rangle \tag{25}$$

$$\eta\nabla^2\langle\boldsymbol{v}_2\rangle + \beta\eta\boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\langle\boldsymbol{v}_2\rangle) - \boldsymbol{\nabla}\langle p_2\rangle = \langle\rho_1\partial_t\boldsymbol{v}_1\rangle + \rho_0\langle(\boldsymbol{v}_1\cdot\boldsymbol{\nabla})\boldsymbol{v}_1\rangle \tag{26}$$

The velocity term $\langle\boldsymbol{v}_2\rangle$ is the acoustic streaming, which occurs when the bulk fluid moves steadily in time. Even though no part of this project has the objective of studying this phenomenon it is commonly observed after acoustofluidic experiments have been running for longer time spans, hence why it is mentioned in this theory section.
Another important term is $\langle p_2\rangle$, this gives rise to the acoustic radiation force from which acoustophoretic motion arises.

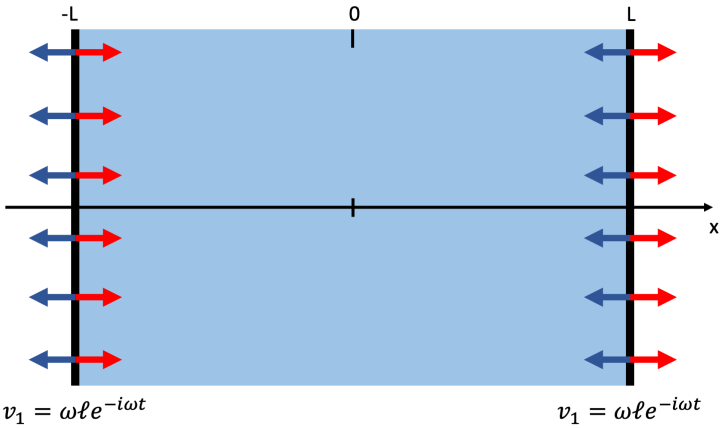


Figure 4.1: The sketch shows a microchannel at profile view. The two walls are placed at a distance L from the center of the channel. The walls are modeled as being in phase as indicated by the arrows, driven by a velocity of $v_1 = \omega\ell \exp(-i\omega t)$.

4.4 Resonance frequency

In the following a simplified setup will be studied to analytically derive an expression of the resonance frequency in a microchannel of width, $W = 2L$. A simple sketch of the overall setup is seen in figure (4.1), where two large parallel walls are placed at a distance of W apart. In this model, the two walls of the system, placed at $x = \pm L$, are driven at an angular frequency of ω. The boundary conditions at $x = \pm L$ are defined by the velocity field, v_1, in equation (27).

$$v_1(-L,t) = \omega\ell \exp(-i\omega t) \qquad v_1(L,t) = \omega\ell \exp(-i\omega t) \tag{27}$$

For a general 1D standing wave, $\boldsymbol{v}_1(x,t) = f(x)\exp(-i\omega t)\mathbf{e}_x$, the curl will be 0 i.e. $\boldsymbol{\nabla} \times \boldsymbol{v}_1 = \mathbf{0}$. As $\boldsymbol{\nabla} \times \boldsymbol{v}_1 = \mathbf{0}$, it implies that $\boldsymbol{v}_1$ is a potential flow and thus can be written in terms of the gradient of a potential, ϕ_1, i.e. $\boldsymbol{v}_1 = \boldsymbol{\nabla}\phi_1$. To find an expression for ϕ_1, one makes use of $\nabla^2\boldsymbol{v}_1 = \boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{v}_1)$, and the divergence of $\boldsymbol{v}_1$ (28) can furthermore be derived from (8).

$$\partial_t\rho_1 = -\rho_0\boldsymbol{\nabla}\cdot\boldsymbol{v}_1 \to \boldsymbol{\nabla}\cdot\boldsymbol{v}_1 \to = -\frac{1}{\rho_0 c_0^2}\partial_t p_1 \to \boldsymbol{\nabla}\cdot\boldsymbol{v}_1 = i\frac{\omega}{\rho_0 c_0^2}p_1 \tag{28}$$

Using this with the first-order perturbed Navier-Stokes (9) one can find ϕ_1 to be (29).

$$\phi_1(\boldsymbol{r},t) = \frac{-i}{\omega\rho_0(1+i\Gamma)^2}p_1(\boldsymbol{r},t) \tag{29}$$

From equation (29) one sees that as $\phi_1 \propto p_1$ and thus ϕ_1 must also satisfy the Helmholtz equation from (14), equation (30) from this with $k = k_0(1+i\Gamma), k_0 = \omega/c_0$.

$$\nabla^2\phi_1 = -k^2\phi_1 \tag{30}$$

The solution to the Helmholtz equation (30) is a superposition of complex exponentials, thus ϕ_1 is found to be (31).

$$\phi_1(x,t) = \left[\phi_+ e^{ikx} + \phi_- e^{-ikx}\right] e^{-i\omega t} \tag{31}$$

Taking the partial derivative of ϕ_1 with regards to x, yields the velocity field v_1 (32).

$$v_1(x,t) = \partial_x\phi_1(x,t) = ik\left[\phi_+ e^{ikx} - \phi_- e^{-ikx}\right] e^{-i\omega t} \tag{32}$$

By using the boundary condition for the system one finds that $\phi_+ = -\phi_-$. Using this along with boundary conditions the following values of ϕ_+ and ϕ_- are found (34).

$$v_1(L,t) = ik\phi_+ \left[\mathrm{e}^{ikL} + \mathrm{e}^{-ikL}\right] \mathrm{e}^{i\omega t} = \omega\ell \mathrm{e}^{i\omega t} \rightarrow i2k\cos(kL)\phi_+ = \omega\ell \tag{33}$$

Isolating ϕ_+ in (33) leads to the following values of $\phi_\pm$ (34).

$$\phi_+ = -\frac{i\omega\ell}{2k\cos(kL)} \qquad\qquad \phi_- = \frac{i\omega\ell}{2k\cos(kL)} \tag{34}$$

Inserting the values for $\phi_\pm$ found in (34) into equation (32) leads to the following expression for $v_1(x,t)$ (35).

$$v_1(x,t) = \omega\ell \frac{\cos(kx)}{\cos(kL)} \mathrm{e}^{-i\omega t} \tag{35}$$

A more useful expression for v_1 is found by Taylor expanding equation (35) in kL around $k_0 L$, this leads to equation (36).

$$v_1(x,t) \approx \omega\ell \frac{\cos(k_0 x) - i\Gamma k_0 x \sin(k_0 x)}{\cos(k_0 L) - i\Gamma k_0 L \sin(k_0 L)} \mathrm{e}^{-i\omega t} \tag{36}$$

Resonance is achieved when the denominator of (36) reaches a minimum, which would be reached at $\cos(k_0 L) = 0$ thus resonance will be achieved at $k_n = (n + 1/2)\pi/L$. In terms of frequency, the first resonance mode will be at $\omega_1 = c_0 k_0 \rightarrow f_1 = c_0 k_0/(2\pi) = c_0/(4L)$. Inserting values of $c_0 = 1.5 \times 10^3\,\mathrm{m\,s^{-1}}$ [1] for water and $L = 200\,\mu\mathrm{m}$, since the channel width in this project is $W = 2L = 400\,\mu\mathrm{m} \rightarrow L = 200\,\mu\mathrm{m}$, the resonance frequency becomes (37).

$$f_{res} = \frac{c_0}{4L} = 1.875\,\mathrm{MHz} \tag{37}$$

4.5 Resistance

As described in [1] chapter 4, microfluidic systems are analogous to electrical circuits in the sense that Kirchhoff's circuit laws hold. The properties of electrical resistors in series and parallel connection translates to hydraulic resistors.

Figure (4.2a) shows a sketch of the experimental setup where R_1 is a bypass channel for R_2 which is the capillary tube. It is assumed that $\mathrm{R}_2 \gg \mathrm{R}_1$ and thus R_1 will let close to all the flow run through it. It can therefore be used to equalize the pressure through R_2 and make particles stand still in the capillary tube. As it can be seen from (4.2b) that the two resistors are in parallel and the equivalent resistance is given by:

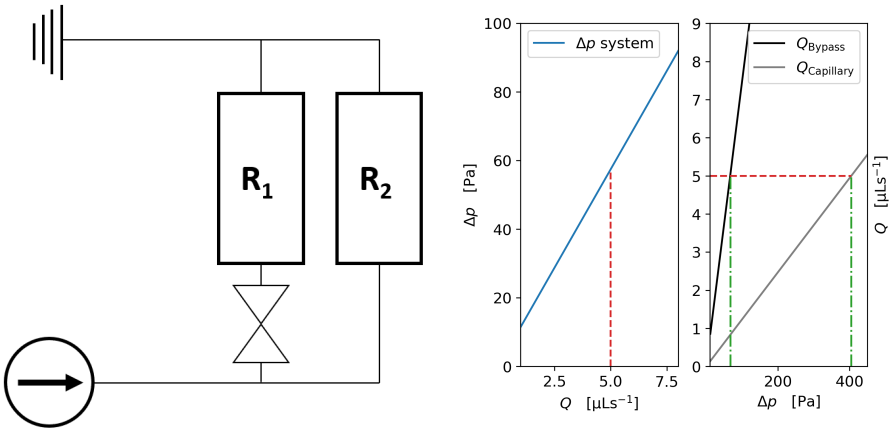
$$R_{\mathrm{e,para}} = \left(\frac{1}{R_1} + \frac{1}{R_2}\right)^{-1} = \frac{R_1 R_2}{R_1 + R_2} \tag{38}$$

$$\tag{39}$$

However both R_1 and R_2 are in them self series connection of different tubes. R_1 consists of two different types of circular tubing and R_2 of circular and square tubes. The equivalent circuit is given as:

$$R_{\mathrm{e,seri,i}} = R_{\mathrm{i},1} + R_{\mathrm{i},2} \tag{40}$$

From table 4.1 from [1] hydraulic resistances for straight channels with different cross-sections can be found. The bypass (R_1) is made with circular plastic tube, see table (1), with radius



(a) (b)

Figure 4.2: (a) "circuit-diagram" analogy of the experimental setup. R_1 and R_2 represents the bypass and capillary tube respectively. In the initial setup R_1 was not implemented forcing all flow to equalize through R_1. The bypass allowed ca. five times more flow than the capillary tube acting like a quick pressure equalizer. (b)(left)(blue line) shows the total pressure difference, in Pa, over the system. i.e. over the equivalent resistor R_e (50). (b)(right) are the flow rates, in $\mu L\,s^{-1}$ over the bypass and capillary tube (black and grey lines) respectively. The red dash line in both (left) and (right) is a sample flow of $5\,\mu L\,s^{-1}$. The green dash-dot lines show the pressure difference needed for that flow in the both types of canal illustrating a vast difference even at moderates flows. This is given that no volume builds up (4.7)

$a_1 = 0.038\,\text{mm}$ and has a length $L_{1.1} = 100\,\text{mm}$ plus a circular silicon tube with radius $a_2 = 0.5\,\text{mm}$ and has a length $L_{1.2} = 28\,\text{mm}$. The resistance of a circular tube is given by:

$$R = \frac{8}{\pi}\eta L_1 \frac{1}{a^4} \tag{41}$$

$$\tag{42}$$

Thus the resistance of R_1 is:

$$R_1 = \frac{8}{\pi}\eta\left(\frac{L_{1.1}}{a_1^4} + \frac{L_{1.2}}{a_2^4}\right) = \frac{8\,\text{mPa s}}{\pi} \times \left(\frac{100\,\text{mm}}{(0.038\,\text{mm})^4} + \frac{28\,\text{mm}}{(0.5\,\text{mm})^4}\right) \tag{43}$$

$$R_1 = 1.34 \times 10^{10}\,\frac{\text{Pa s}}{\text{m}^3} \tag{44}$$

R_2 consists of the capillary tube with a square cross-section with sides $h = 400\,\mu\text{m}$ and length $L_{2.1} = 50\,\text{mm}$ plus sections of the aforementioned plastic and silicon tubes tube with lengths $L_{2.2}$, $L_{2.2} = 193\,\text{mm}, 48\,\text{mm}$ respectively. The Resistance of a square tube is given by:

$$R = 28.4\eta L_2 \frac{1}{h^4} \tag{45}$$

The resistance for R_2 is then given by:

$$R_2 = \eta\left(\frac{8}{\pi}\frac{L_{2.2}}{a_1^4} + \frac{8}{\pi}\frac{L_{2.3}}{a_2^4} + 28.4\frac{L_{2.1}}{h^4}\right) \tag{46}$$

$$R_2 = 1\,\text{mPa s}\left(\frac{8}{\pi}\frac{48\,\text{mm}}{0.038\,\text{mm}^4} + \frac{8}{\pi}\frac{193\,\text{mm}}{0.5\,\text{mm}^4} + 28.4 \times \frac{50\,\text{mm}}{400\,\mu\text{m}^4}\right) \tag{47}$$

$$R_2 = 8.10 \times 10^{10}\,\text{Pa s m}^{-3} \tag{48}$$

The equivalent resistance is therefore given as:

$$R_{\mathrm{e}} = \left(\frac{1}{1.34 \times 10^{10}\,\mathrm{Pa\,s\,m^{-3}}} + \frac{1}{8.10 \times 10^{10}\,\mathrm{Pa\,s\,m^{-3}}} \right)^{-1} \tag{49}$$

$$= 1.15 \times 10^{10}\,\mathrm{Pa\,s\,m^{-3}} = 11.5\,\mathrm{Pa\,s\,\mu L^{-1}} \tag{50}$$

The hydrodynamic pressure difference, Δp, due to an applied flow[1] through the equivalent resistor can be determined by Hagen-Poiseuille's law:

$$\Delta p = R_{\mathrm{hyd}} Q = R_{\mathrm{e}} Q \tag{51}$$

With this relation Δp can be determined for various flow rates. The flow rates used in the setup are typically between $0.01\,\mu\mathrm{L\,s^{-1}}$ to $5\,\mu\mathrm{L\,s^{-1}}$[2].
As it can be seen from the figure(4.2b) the flow rate $Q_2 \ll Q_1$. In this theoretical estimation, the Δp is solely due to the hydrodynamic pressure difference. It is not unlikely that if no flow rate is applied there could still be a flow due to hydro-static pressure difference, compliance (4.7) or some other unforeseen effect. It is assumed that these are much smaller than the hydrodynamic pressure difference. A sample flow is presented in figure(4.2b)

A simple but important calculation explains why a bypass channel is implemented. For a pressure difference Δp across R_1 and R_2 the following relations must hold (52) due to Hagen-Poiseuille's law (51).

$$\Delta p = R_1 Q_1 \qquad\qquad \Delta p = R_2 Q_2 \tag{52}$$

Equating the relations in (52) results in equation (53).

$$R_1 Q_1 = R_2 Q_2 \rightarrow Q_2 = \frac{R_1}{R_2} Q_1 \tag{53}$$

Using the values for R_1 and R_2 which were found in (44) and (48), the following relation between Q_1 and Q_2 is (54).

$$Q_2 = 0.17 Q_1 \tag{54}$$

Thus from (54) one sees that when the bypass is opened only 17% of the fluid flows through R_2. It is also simple to see that in order to decrease the flow through R_2 one would have to either decrease the hydraulic resistance through the bypass channel, R_1, or increase the hydraulic resistance through the capillary tube, R_2.

4.6 Sedimentation

As described in section(4.5) a flow can persist even if the valve to R_1 is open. This flow have been observed to run in both directions. If there is a back flow this could be due to sedimentation. To check this a theoretical expression for the sedimentation velocity must be obtained. In an ideal scenario, one algae particle is situated in a water-filled tube with no flow in any direction. It is assumed that the algae cell is far enough away from the walls so the no-slip boundary condition of the wall can be neglected. The particle has mass, m, and radius, r. The water density, ρ, and viscosity, η. The forces on the particle are: Gravity ($\mathrm{F_g}$), buoyancy ($\mathrm{F_b}$), and stokes-drag ($\mathrm{F_d}$). Since the particle is moving at constant velocity Newton's law gives:

$$F_g + F_b + F_d = 0 = -mg + \rho g V + 6\pi\eta r v \tag{55}$$

$$v = \frac{(m - \rho V)g}{6\pi\eta r} \tag{56}$$

[1] Usually a pressure difference give rise of a flow, not vice versa, but the Cetoni Elements works by choosing a flow not a pressure and therefore it was a natural state it

[2] This is for the experiments working with flow (5.4). For flushing higher flow were used (B)

The parameters for an algae cell vary a lot, but according to [9] the mass and size are given as, $m = 1.54 \times 10^{-14}\,\mathrm{kg}$, and $r = 2.165 \times 10^{-6}\,\mathrm{m}$. with this, the velocity becomes:

$$v = \frac{(1.54 \times 10^{-14}\,\mathrm{kg} - 998\,\mathrm{kg\,m^{-3}} \times \frac{4\pi}{3} \times 2.165 \times 10^{-6}\,\mathrm{m^3}) \times 9.82\,\mathrm{m\,s^{-2}}}{6\pi \times 10^{-3}\,\mathrm{Pa\,s} \times 2.165 \times 10^{-6}\,\mathrm{m}} \tag{57}$$

$$= -6.50\,\frac{\mu\mathrm{m}}{\mathrm{s}} \tag{58}$$

With the minus sign being the direction. It is worth noting that the Stokes-drag used here assumes that the Reynolds number, $Re = \eta r v/\rho \ll 1$. An approximation of the Reynolds number yields the following:

$$Re = \frac{\eta r v}{\rho} \approx \frac{10^{-3} \times 10^{-6} \times 10^{-6}}{10^{3}} = 10^{-18} \tag{59}$$

Which satisfy the assumption that the Reynolds number is small.

4.7 Compliance

Flow persistence in the system could be due to compliance. The tubes used in the experimental setup can not be assumed to be completely rigid. The following deviation is based on [1] section on compliance.

In the experimental setup, pressure is applied for some time in order to move algae to the

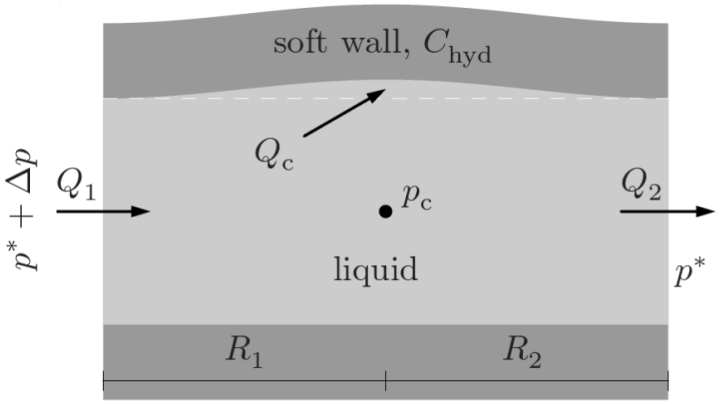


Figure 4.3: A channel consisting of two sub-channels with resistances R_1 and R_2. p^* is the pressure outside the system i.e. the reference pressure, Δp is the pressure applied to the system and p_{c} is pressure due to the compliance. Q_1 is the flow at the inlet, Q_2 is the flow at the outlet and Q_{c} is the flow from the compliance. In the present case, nothing is fluxing into the system and Q_1 should be seen as the flow in the region of R_1. From the conservation of mass, the flow into and out of the system must be the same: $Q_1 = Q_{\mathrm{c}} + Q_2$. Image taken from [1] figure 4.9.

camera view and/or flush sedimented algae. Because of the resistance in the capillary tube, pressure will build up in the system. As the pressure builds the flow increases in the capillary tube until the system reaches a steady state. It is assumed that the pressure is applied for long enough for the system to reach a steady state. The pressure at this point can be found using (4.49) from [1]:

$$p_{\mathrm{c}}(t) = p^* + \left(1 - \exp\left[-\tau_1^{-1} - \tau_2^{-1}\right]t\right) \frac{\tau_2}{\tau_1 + \tau_2} \Delta p \tag{60}$$

where $p_{\mathrm{c}}(t)$ is the pressure due to the expansion of the tube (i.e the compliance), p^* is the pressure outside the system (the reference pressure), Δp is the pressure applied to the system,

$\tau_1 = R_1 C_{\text{hyd}}$, and $\tau_2 = R_2 C_{\text{hyd}}$ are the characteristic times respectively. The compliance of a system like the one examined is defined to be (61).

$$C_{\text{hyd}} \equiv \frac{\mathrm{d}V}{\mathrm{d}p} \tag{61}$$

Equation (60) describes the build-up of pressure in the system and letting $t \to \infty$ will give the steady-state pressure:

$$p_{\text{c}}(t \to \infty) = p^* + k\Delta p \tag{62}$$

where $k = \frac{\tau_2}{\tau_1+\tau_2}$. As described on figure (4.3) $Q_1 = Q_{\text{c}} + Q_2$, where $Q_1 = \frac{p^*+\Delta p-p_{\text{c}}}{R_1}$, $Q_2 = \frac{p_{\text{c}}-p^*}{R_2}$. Q_{c} is the rate of volume change: $Q_{\text{c}} = -\partial_t V = C_{\text{hyd}}\partial_t p_{\text{c}}$. This gives:

$$\frac{p^* + \Delta p - p_{\text{c}}}{R_1} = C_{\text{hyd}}\partial_t p_{\text{c}} + \frac{p_{\text{c}} - p^*}{R_2} \tag{63}$$

$$\partial_t p_{\text{c}} = \frac{1}{C_{\text{hyd}}}\left(\frac{p^* + \Delta p - p_{\text{c}}}{R_1} - \frac{p_{\text{c}} - p^*}{R_2}\right) \tag{64}$$

$$\partial_t p_{\text{c}} = \frac{p^* + \Delta p - p_{\text{c}}}{\tau_1} - \frac{p_{\text{c}} - p^*}{\tau_2} \tag{65}$$

$$\partial_t p_{\text{c}} = \left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right) p^* - \left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right) p_{\text{c}} + \frac{\Delta p}{\tau_1} \tag{66}$$

The solution to this ODE is:

$$p_{\text{c}} = -\frac{\left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right) p^* + \frac{\Delta p}{\tau_1}}{-\left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right)} + C_1 \exp\left[-\left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right) t\right] \tag{67}$$

$$= p^* + \left(1 + \frac{\tau_2}{\tau_1}\right)\Delta p + C_1 \exp\left[-\left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right) t\right] \tag{68}$$

The condition (62) is the initial condition for the ODE and this gives:

$$p^* + k\Delta p = p^* + \left(1 + \frac{\tau_2}{\tau_1}\right)\Delta p + C_1 \tag{69}$$

$$= \Delta p\left(k - 1 - \frac{\tau_2}{\tau_1}\right) \tag{70}$$

$$C_1 = \Delta p\left(\frac{\tau_2}{\tau_1 + \tau_2} - 1 - \frac{\tau_2}{\tau_1}\right) \tag{71}$$

The final solution is then:

$$p_{\text{c}} = p^* + \left(1 + \frac{\tau_2}{\tau_1}\right)\Delta p + \Delta p\left(\frac{\tau_2}{\tau_1 + \tau_2} - 1 - \frac{\tau_2}{\tau_1}\right) \exp\left[-\left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right) t\right] \tag{72}$$

$$= p^* + \Delta p\left(\frac{\tau_2}{\tau_1 + \tau_2}\right) \exp\left[-\left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right) t\right] \tag{73}$$

With this expression, a theoretical time scale for the pressure due to compliance can be obtained.

4.7.1 Apoximation of the compliance in a silicone tube

From an examination of figure (4.4) one notes that each time the fluid inside of the tube exerts an amount of work given by $W_{fluid} = F_{fluid}\Delta r$, on the walls, the change in the potential

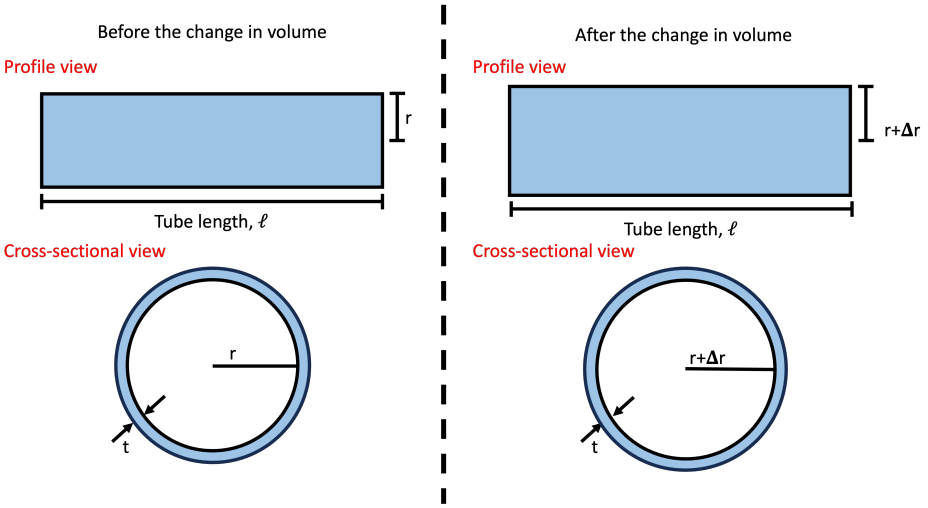


Figure 4.4: The figure shows a sketch of an example tube with an initial radius, r, a length, ℓ, and a thickness, t. The pressure increase, Δp, in the tube then leads to an increase of the tube volume through a change in the radius, Δr.

energy of the wall is equivalent to the magnitude of work done by the fluid, and given by $U_{wall} = -2\pi F_{wall}\Delta r$. This can be written as in equation (74).

$$W_{fluid} + U_{wall} = 0 \rightarrow F_{fluid}\Delta r = 2\pi F_{wall}\Delta r \tag{74}$$

F_{fluid} can be written as the change in pressure Δp, giving rise to the force, multiplied with the internal surface area of the tube, given by $A_{internal} = 2\pi r\ell$. Similarly, can F_{wall} be written as some internal pressure, p_{wall}, multiplied by the cross-sectional area of the wall, given by $A_{wall} = t\ell$. Substituting this into equation (74) yields equation (75)

$$2\pi r\ell\Delta p = 2\pi t\ell p_{wall} \rightarrow r\Delta p = tp_{wall} \tag{75}$$

Young's modulus, $E = \sigma/\epsilon$, is a measure of the relation between stress, σ, and strain, ϵ. It can also be written as $E = p \times L/\Delta L$, in this case, $p = p_{wall}$, $L = 2\pi r$ is the starting circumference of the tube, and ΔL is the change in the tube circumference. Solving the latter equation of Young's modulus for p_{wall} yields the following relation $p_{wall} = E \times \Delta r/r$. Substituting into equation (75) yields equation (76).

$$r\Delta p = \frac{Et}{r}\Delta r \rightarrow \Delta p = \frac{Et}{r^2}\Delta r \tag{76}$$

To make further progress the volume change, ΔV, that occurs as a result of the pressure change, Δp, is studied. The volume change in a tube like the one drawn in figure (4.4) is given by equation (77). As Δr is assumed to be small, the following approximation is used $2r\Delta r + \Delta r^2 \approx 2r\Delta r$.

$$\Delta V = \pi\ell(r + \Delta r)^2 - \pi\ell r^2 \approx 2\pi r\ell\Delta r \rightarrow \Delta r = \frac{\Delta V}{2\pi r\ell} \tag{77}$$

Substituting equation (77) into equation (76) results in equation (78).

$$\Delta p = \frac{Et}{2\pi\ell r^3}\Delta V \rightarrow \frac{\Delta V}{\Delta p} = 2\pi\frac{\ell r^3}{Et} \tag{78}$$

Making use of the approximation that $\frac{\Delta V}{\Delta p} \approx \frac{\mathrm{d}V}{\mathrm{d}p}$ yields equation (79) which is the final expression for the compliance of a cylindrical tube.

$$C_{hyd} = 2\pi\frac{\ell r^3}{Et} \tag{79}$$

It is important to note that expression (79) is only an approximation. A full and analytically correct expression for the compliance of such a tube is found as equation 73 in [2], it is also written in this thesis as (80). In (80), $\bar{\nu}$ is the Poisson ratio, δ is the thickness ratio $\delta = t/r$ with t being the thickness of the wall, r is the inner radius of the tube, and ℓ is the length of the tube.

$$C_{hyd} = 2\pi(1+\bar{\nu})\frac{(1-2\bar{\nu})+(1+\delta)^2}{(1+\delta)^2-1}\frac{r^2\ell}{E} \tag{80}$$

In the experimental setup, a mix of silicone and plastic tubes is used in addition to the glass capillary. From figure(4.3) is can be seen that the compliance of a soft wall consists of two tubes. For simplicity let R_1 be the tubing before the GCTH and R_2 be the capillary tube. This means that R_1 is a series of plastic- and silicon tubes. It is assumed that the plastic has very little compliance compared to the silicone and this can be checked by inserting parameters for the two into (61).

For silicone rubber Young's modulus is: $E = 0.001\,\text{GPa}$ [10][3], The radius is: $r = 0.5\,\text{mm}$ and the thickness is: $t = 1\,\text{mm}$ (1). This gives a compliance per length of:

$$\frac{C}{\ell} = 2\pi \times \frac{0.5\,\text{mm}^3}{0.001\,\text{GPa} \times 2\,\text{mm}} = 3.92 \times 10^{-13}\,\frac{\text{m}^2}{\text{Pa}} \tag{81}$$

For plastic Young's modulus is: $E = 586\,\text{GPa}$ [11], The radius is: $r = 0.38\,\text{mm}$ and the thickness is: $t = 0.42\,\text{mm}$ (1). This gives a compliance per length of:

$$\frac{C}{\ell} = 2\pi \times \frac{0.38\,\text{mm}^3}{586\,\text{GPa} \times 0.83\,\text{mm}} = 7.09 \times 10^{-19}\,\frac{\text{m}^2}{\text{Pa}} \tag{82}$$

As it can be seen the compliance per length is 6 orders of magnitude higher for silicone. In order to ease the calculation the tubing will be approximated to only consist of silicone as the contribution of the plastic tubing is negligible.

Going back to equation (73) one notice that p^* is a constant reference pressure and will not affect the pressure difference, and $\Delta p\left(\frac{\tau_2}{\tau_1+\tau_2}\right)$ is just the pressure at $t = 0$. Let $\tau^{-1} = \frac{1}{\tau_1} + \frac{1}{\tau_2}$. The only thing affecting the time it takes for the pressure to equalize is τ^{-1}. Therefore a value of $\tau^{-1} \approx 1$ will give about 1 second for the pressure to equalize to p^*. Let the length of silicone tube be: $\ell = 20\,\text{cm}$. From (60) and (4.5) τ_1 and τ_2 are given as:

$$\tau_1 = C_{\text{hyd}} R_{\text{silicone}} = 3.92 \times 10^{-13}\,\frac{\text{m}^2}{\text{Pa}} \times 20 \times 10^{-2}\,\text{m} \times \frac{10^{-3}\,\text{Pa s} \times 8 \times 20 \times 10^{-2}\,\text{m}}{\pi \times 0.5 \times 10^{-3}\,\text{m}^4} \tag{83}$$

$$= 6.4 \times 10^{-4}\,\text{s} \tag{84}$$

$$\tau_2 = C_{\text{hyd}} R_{\text{capillary}} = 3.92 \times 10^{-13}\,\frac{\text{m}^2}{\text{Pa}} \times 20 \times 10^{-2}\,\text{m} \times \frac{10^{-3}\,\text{Pa s} \times 28.4 \times 50 \times 10^{-3}\,\text{m}}{400^4 10^{-24}\,\text{m}^4} \tag{85}$$

$$= 4.4 \times 10^{-3}\,\text{s} \tag{86}$$

$$\frac{1}{\tau} = \frac{1}{6.4 \times 10^{-4}} + \frac{1}{4.4 \times 10^{-3}} = 1792\,\text{s} \tag{87}$$

The approximation is checked by calculating the compliance using (80):

$$C_{hyd} = 2\pi(1+0.48)\frac{(1-2\times0.48)+(1+4.65\times10^{-8})^2}{(1+0.48)^2-1}\frac{(0.5\times10^{-3}\,\text{m})^2 0.2\,\text{m}}{0.001\times10^9\,\text{Pa}} \tag{88}$$

$$\approx 5.25 \times 10^{-13}\,\text{m}^3\,\text{Pa}^{-1} \tag{89}$$

[3] Young's modulus varies from 0.001 GPa to 0.05 GPa from this source. The Poisson ratio for silicone rubber is $\bar{\nu} = 0.48$ ([10]). The former is used to try and maximize the compliance

This is very close to the estimated value of $1.57 \times 10^{-13}\,\mathrm{m}^3\,\mathrm{Pa}^{-1}$ showing that the approximation holds.

Since $1/\tau \gg 1$ the times it takes for the pressure to equalize is on the scale of milliseconds. As it can be seen the values for the parameters used here are exaggerated to maximize compliance. However, the compliance is still small and this shows that a flow duo to the soft wall is unlikely. The computations were made using a script which is seen in (E).

5 Development of the experimental setup

5.1 Initial setup

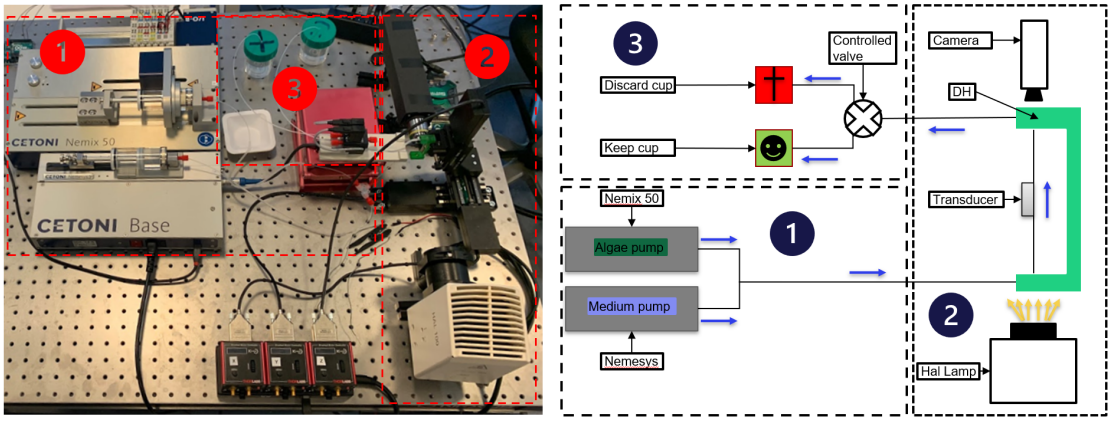


Figure 5.1: The figure shows the initial setup (used for the motivational experiment, see section (5.2)). The sketch on the right is with the selected element from the picture on the left. An exhaustive list and description can be found in section (5.1). The sketch is divided into three regions for the sake of clarity - the blue arrows show the direction of the flow. Region 1 has the Cetoni Nemix 50, and the Nemesys S syringe pumps which contain algae water and algae medium, respectively. They are both controlled via Cetoni Elements software. Region 2 has the Hal Lamp that illuminates the device holder (DH, it has a [400 µm × 400 µm] capillary tube with a Pz26 ceramic plate transducer i.e. "the device", see more in section (5.3)) and the camera for recording the algae in the capillary tube. On the sketch the DH is viewed in profile - see the picture for true orientation. Region 3 contains the switch valve that sends the water to either the discard-cup, keep-cup, or to both. The blue arrows indicate the direction of flow.

This section is meant to give an overview of the setup used for all experiments. The main workings of the setup can be seen in section (5.2).

All experiments were based on an existing acoustofluidic setup. All the equipment was fastened to a Thorlabs NEXUS-table. The Cetoni Base was connected to the Cetoni Nemesys S, and the Cetoni Nemesys S syringe pump was further connected to a [1/16 x .020]-T-junction. The Cetoni Nemix 50 was likewise connected to the T-junction. The T-junction was then connected to the device holder (DH). [1/16 x .030]-Plastic tubing was inserted in both ends of the DH. A Pz26 ceramic plate transducer with a length of 12.7 mm, a width of 6.35 mm, and a thickness of 1 mm was glued to the square capillary tube with [ID;OD] of [400;800] µm. The transducer was connected to an Analog Discovery 2 USB-oscilloscope through a Wide Band DC-5MHz amplifier. The capillary tube was illuminated by a HAL 12V, 100W max lamp and was recorded using a Basler Camera with a ZEISS 10X/0.25 lens situated on a Thorlabs Camera holder. The DH was fastened to Thorlabs x,y,z-stage. The DH was connected to a PC-controlled junction valve which was in turn connected to two collection cups. Everything controlled by the PC either via a direct USB-connection or through an Arduino. An exhaustive list of all hardware can be seen in Table(1):

5.2 Motivational experiment

As described in (4.4) particles can be acoustically focused into the center node of a microchannel if resonance frequency is used. This was tested using the initial setup see section (5.1). The transducer (Pz26) used was designed to have a resonance frequency at 2.0 MHz. This was tested by using the Analog Discovery 2 to do an impedance sweep (D). The resonance frequency of the transducer lies close to the theoretical resonance frequency of the square capillary tube, derived in section (4.4) allowing for efficient acoustic energy transfer into the capillary tube. The resonance frequency of the transducer could be off due to manufacturing flaws or damage, but also from

Name	Cetoni Nemix 50	Cetoni Base	Cetoni NemesysS	T-junction
Manufacturer	Cetoni	Cetoni	Cetoni	IDEX
Type	Pump	Base	Pump	1/16 x .020 thru
S/N	NA	CET-009845-2301	NA	P-632
Quantity	1	1	1	1
Name	Transducer	Capillary tube	Junction valve	x,y,z-stage-base
Manufacturer	Megitt A/S	CM Scientific LTD	Burkart	Thorlabs
Type	Ceramic/plate	Square Boro Tubing	PC-controlled Valve	Base
S/N	Pz26	NA	00175653	KCH 301
Quantity	1	1	1	1
Name	Brushed Motor controller	x,y,z-stage	Camera-holder	Camera
Manufacturer	Thorlabs	Thorlabs	Thorlabs	Basler
Type	kube-controller	Stage	Holder	Camera
S/N	KDC101	MTS50(A/B/C)	CP42M	40239628
Quantity	3	3	1	1
Name	Lens	Lamp	Power supply	Analog Discovery 2
Manufacturer	ZEISS	HAL	Tektromix	Digilent
Type	10x/0.25	12V, 100W max	DC	USB-oscilloscope
S/N	EC EPIPLAN	HAL 100	PWS2326	210-321
Quantity	1	1	1	1
Name	Amplifier	Arduino	Plastic tube	Silicone tube
Manufacturer	Toellner	Arduino	IDEX	IDEX
Type	WIde Band DC - 5MHz	UNC	1/16 x .030 [ID;UD] [0.76 mm;1.59 mm]	[ID;UD] [1.0 mm;3.0 mm]
S/N	TCE 7607	NA	1514L	FB50853
Quantity	1	1	NA	NA

Table 1: List of equipment in initial setup

being attached to or submerged into various substances. In this case, the transducer is attached to a borosilicate-glass-square-capillary-tube see table(1) using NOA 86H glue.

The Cetoni Nemix 50 was loaded with an undiluted algae solution from population 1.2. The Cetoni Nemesys S was filled with pure water. Pure water was chosen instead of algae medium to save the medium as all algae from the experiment would be discarded. The Hal lamp was turned on and set to 10 V and 3 A via the DC-power supply. The amplifier was set to 20 dB. The Cetoni base was turned on and connected via Cetoni Elements on the PC. On the PC the Python-script "surviving experiment" (E) was utilized to initiate the camera feed. The Thorlabs cubes were powered on and connected to the PC via Thorlabs Kinesis. Thorlabs Kinesis was used to move the DH attached to the Thorlabs stage to a desired position at which the optical focus would be in the middle of the channel. In Waveforms a sinusoidal wave of frequency 1.91 MHz (1.91 MHz was known from a previous project to work well with the given transducer and capillary tube) was set to run and the algae was observed to focus in the channel.
This experiment served as proof of concept and was a natural stepping stone towards an automated process for handling algae. The experiment and following replication experiments also served to shed light on various obstacles regarding the setup. See sections (5.3, 5.6, 5.5). It also led to the creation of a standardized initialization procedure which can be seen in section (B).

5.3 The device holder (DH)

5.3.1 Initial DH

As already stated in section (5.1) the initial setup came with a DH (5.2) to hold the capillary tube in place and attach it to the x,y,z-stage. The original DH was 3D-printed in a Elegoo Photo Polymer Resin in the lab. The capillary tube was glued directly into the DH which was a problem because it could not be replaced without breaking the tube nor the DH itself. In addition, it was not possible to rotate the the capillary tube without moving either the camera or the x,y,z-stage. Lastly, the DH had small cavities below and above where the capillary tube was inserted, which were suspected to act as "reservoirs" for algae that caused algae to remain in the system even after flushing with pure water. This however proved not to be the case, and it will discussed in more detail later.

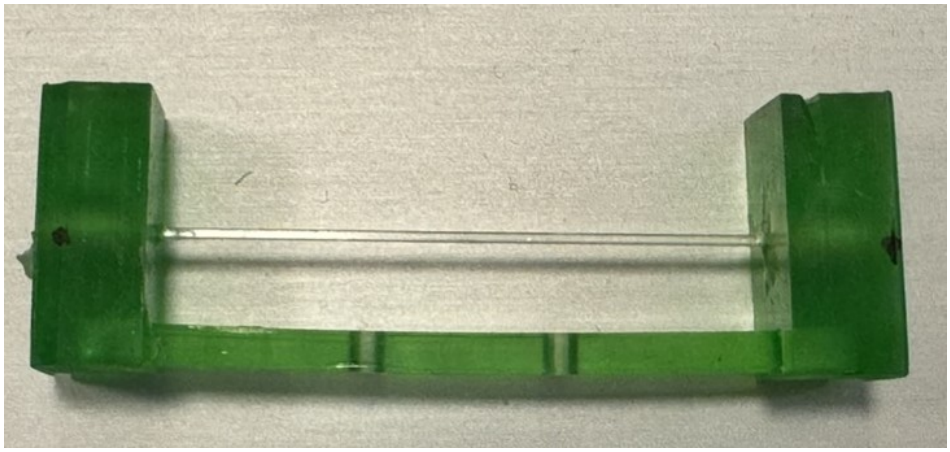


Figure 5.2: A picture of the initial DH. Small plastic tubes would be inserted at the top of each block at the end. The capillary tube was inserted into small holes and glued in place. This meant that the capillary tube could not be exchanged. The DH had no rotational capabilities which meant that either the x,y,z-stage had to be rotated or the camera moved if the capillary tube sides were not perpendicular to the viewing angle

5.3.2 New DH with rotation (RDH)

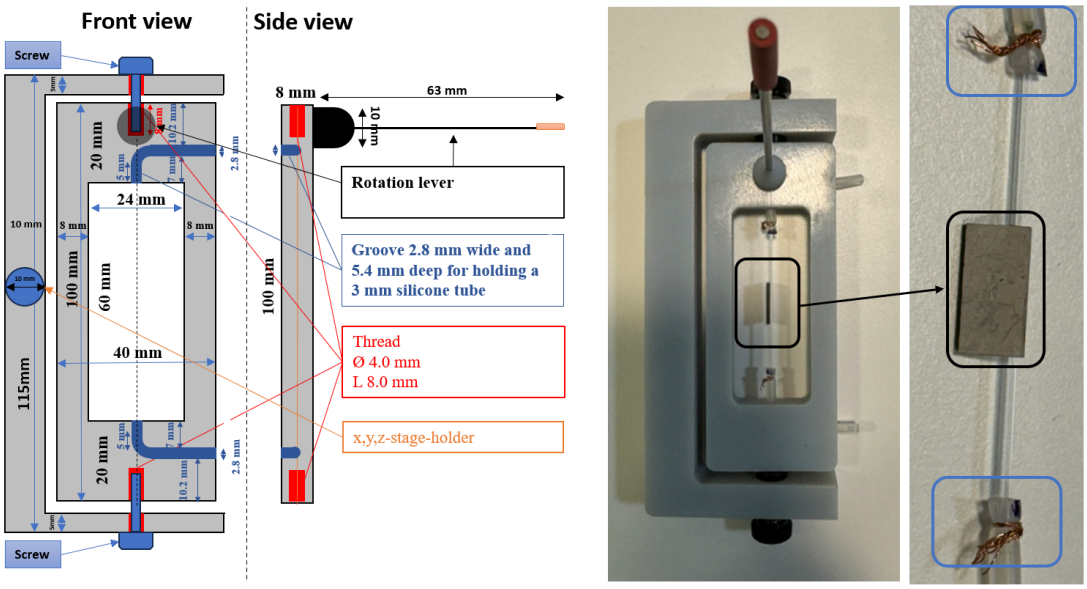


Figure 5.3: (Left) A sketch of the new device holder with rotational capabilities (RDH). The lever could be used to rotate the square capillary tube to be perpendicular to the viewing angle. The RDH also had grooves that the silicone tubes would fit into. The grooves held the silicone tubing in place, making it easy to change the capillary tube if needed. (Right) Pictures of the RDH and a close-up of the capillary tube with the transducer attached to it. The blue box shows how the capillary tube was fastened in the silicone tubes. A few strands of copper wire from a regular electrical wire were spun around the silicone tube until the capillary was fastened securely. This was a good method as it did not require glue which tended to flow into the channel.

Due to the issues stated in section (5.3.1) it was decided that a new DH was needed for the experiment. Prototypes were made from miscellaneous items found in the lab, but due to the sensitivity of the experiment, it was ultimately decided to have it properly made in the DTU Physics workshop. A plan was made for which needs the new DH should fulfill for the setup, these were based on the experience from the prototypes. The schematic for this can be seen in figure(5.3). The initial idea was to construct the new DH by milling grooves into an aluminum

plate because of the limited deformability of the metal, but after consultation with the workshop, it was decided that 3D-printing it in a hard Vero Stratasys J55 polymer would suffice. The new DH consisted of an outer "c-shaped" holder from where two screws were inserted in the top and bottom allowing for rotation of the inner rectangular-shaped holder for the capillary tube. The screws had a tightening mechanism allowing for adjustment of the resistance when rotating as well as firmly securing the position once a desirable angle was achieved. The inner part had 2.8 mm grooves which were 0.2 mm smaller than the silicone tube that went into them. This was for securing them as well as having an easy way to change the capillary tube if needed.

The capillary tube was then inserted into the silicone tube, but due to the difference in shape and size, there was a residual space between the capillary- and silicone tube which needed to be sealed. Various glues and silicone sealers were tested with limited success and eventually, the gaps were sealed by simply tightening the silicone tube to the capillary tube using four strands of cobber wire from a regular electrical wire. This turned out to work very well, and in addition, the capillary tube could be changed without breaking it or attempting to dissolve the glue. The new DH will be referred to as RDH (rotating device holder) from hereon.

5.4 Characterization of the capillary tube (No yield experiment)

The acoustic energy density (E_{ac}) can vary throughout the capillary tube as described in [3]. This could give regions, in the capillary tube, with little to no acoustic energy density or regions with larger energy densities. It was wanted to know where these energy discrepancies may be located for future experiments. The idea for the experiment was to have the Thorlabs x,y,z-stage move to a starting position, optically focus on the middle of the channel (in the depth of the channel, that is the x-direction defined by the placement of the Thorlabs x,y,z-stage) and wait. At the same time a constant flow of algae water would be started manually via Cetoni Elements. The constant flow was chosen because a flow would persist even after the flow from the syringe was stopped (4.5)(4.6)(4.7). The wave generator would then excite the capillary tube in a sweep of voltages, and a sweep of frequencies to check if the resonance frequency would change as well as the acoustic energy density. After the sweeps had been done, the x,y,z-stage would move to the next position and repeat the process as described above. This could be done for any wanted length of the capillary tube. In order to do this, a Python script was made that would be able to automatically carry out the experiment as it would be cumbersome to do manually.

First a Python-driven method for moving x,y,z-stage was created. For the experiment the following functions were created: "Position" returns the position of the stage. "Home" moved the stage to home position and defines that position as 0. "CheckHome" returned if the stage was homed. "MoveRel" moved the stage a distance from the current position. "MoveAbs" moved the stage to a given position. This was created individually for x, y and z direction as each worked as a separate device. With these functions, the stage could be moved in space as desired. However, when wanting to move the stage backwards (moving to a position with a lower value than the current position value) the stage would interpret the command as out of bounds and move the stage backward at max speed until reaching the lower limit of the stage. The problem was never identified and fixed, despite extensive efforts. Eventually a workaround, using the Home-function in addition with planning the experiment to have the stage mostly run forward, was implemented.

The second task was to acoustically focus the algae in the channel as in (5.2) experiment. Here four functions for controlling the analog Discovery were made: Connect() to connect to the Analog Discovery 2, Funcgen() to generate an acoustic signal at a given frequency and amplitude, Funcstop() to stop the acoustic signal, and disconnect() to disconnect from Analog Discovery 2.

These functions were used to create a script (E) that would carry out the following procedure:

1. Home the stage
2. Move the stage to the starting position

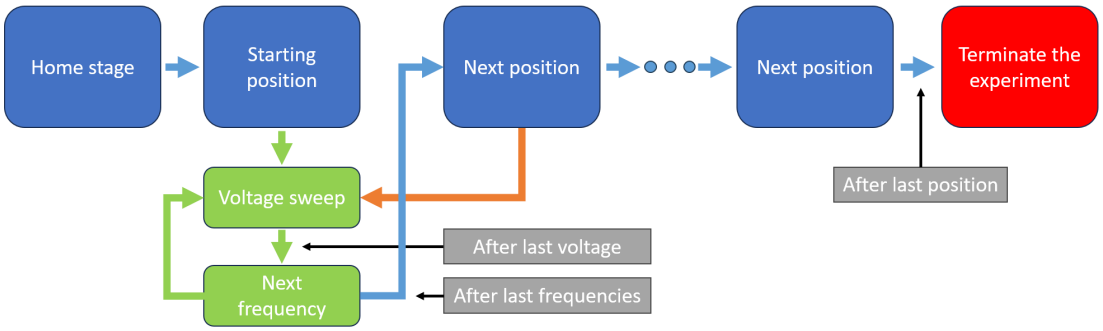


Figure 5.4: A flow chart of the experiment. The blue boxes indicate the positions of the capillary tube throughout the experiment. The green boxes are the sweeps done in frequency and voltage. For every position a sweep in frequency, and a sweep in voltage is done, e.g. 10 positions at 5 frequencies, and at 20 voltages would give 1000 runs.

   (a) Do a voltage-sweep at the first frequency

   (b) Switch to next frequency

   (c) Repeat (2a-2b)

3. Move to the next position
4. Repeat from 2a
5. Terminate once the desired length of the tube had been analysed

Doing the testing of the script multiple problems were encountered. Because the camera was very close to the capillary tube the camera would crash into the RDH if the movement range was not carefully implemented in the script. Retrospectively the rotation lever for the RDH should have been placed such that it pointed away from the camera, as it could collide with the camera enclosure limiting the movement in the z-direction. The camera could at no point give a higher frame rate than 7.4 frames per second which was too low for analysing the acoustic energy density at or near the resonance for, or at high voltages. The post-analysis was difficult because video recording was the only data - later timestamps for the frames were implemented. There was also tilt of the capillary tube in the x-direction that caused the camera to come out of focus throughout the length of the capillary tube. The problems were fixable and some were fixed, but the method was eventually scrapped due to two main problems:

1. The constant flow of algae introduced noise into the data and it was not possible to extract any meaningful results from it. It was suspected that the cause was that too much algae was flowing in or out of the frame. This would mean that the algae coming into the frame from the entrance of the capillary tube had been subject to far less acoustic focusing than the algae flowing out and vice versa at the other end of the tube. In addition, the constant flow was problematic because on one hand, the flow rate had to be low enough such that analysis of the focusing was possible. On the other hand, a slow flow would take a long time because the "focused algae" would stay in the middle even after the acoustic focusing was complete, and the entire length of the tube would have to be flushed with new algae before a new focusing could be done.
2. The entire design of the setup was not suited for this kind of experiment. The tube had regions where the clip for applying the signal to the transducer blocked light from the Hal lamp making it impossible to do analysis in those regions. The region where the transducer was attached vs. the rest of the tube had a different light intensity and this would be a problem for the analysis using the intensity method (6).

Even though the experiment did not yield any result it did illuminate numerous problems that would have occurred in later experiments had precautions not been taken.

5.5 Sedimentation

In this section an experimental value for sedimentation velocity of the algae is fund. This will be used to see if the sedimentation could explain the observed back flow.

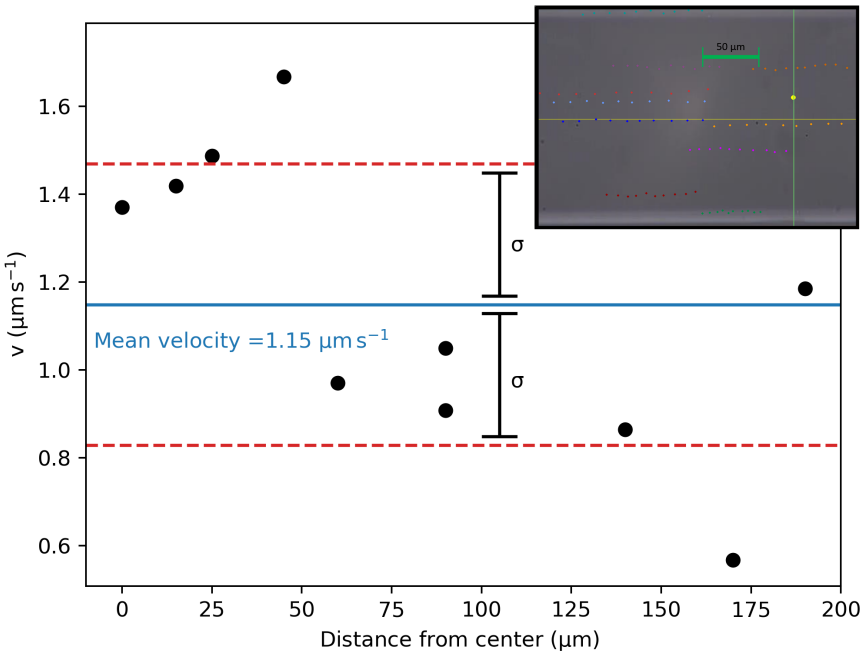


Figure 5.5: The sedimentation velocities in the square capillary tube with side length 400 µm. The main figure shows sedimentation velocities at different distances from the center of the capillary tube. The mean velocity (blue line) is ± one standard deviation (red dotted lines) $\sigma = 0.32\,\mu\mathrm{m\,s^{-1}}$. The inner figure shows the last used frame from the Logger Pro analysis. The yellow line defines the center of the capillary tube and the green line defines the width. Note that the image is tilted 90°.

As mentioned in section (4.6) a back flow could be due to sedimentation. The theoretical velocity of settling algae is in the order of $1\,\mu\mathrm{m\,s^{-1}}$. To check this the following experiment was conducted. First, the system was flushed as described in the section (B). After this the valve for the shortcut was closed and the algae solution were pumped into the capillary tube at flow rates of $1\,\mu\mathrm{L\,s^{-1}}$ over a duration of 10 s. While the algae solution was pumped into the system, the capillary tube was observed and when the algae started to appear in the frame the flow of algae was stopped. This was because the concentration of algae would otherwise be too high to clearly distinguish the algae from each other. The valve to the capillary tube was closed and the forward flow completely stopped. From hereon a video recording was started with a frame rate of one frame per second and it was left running for 1 hour, 25 minutes, and 35 seconds. The algae used was generation 3.1 and the frame size was $400\,\mu\mathrm{m} \times 700\,\mu\mathrm{m}$

After the experiment was done the video was compressed using VLC media player so it could be analysed using Logger Pro 3.16.2. The high resolution camera was used for this recording and therefore the video needed to be compressed in order for Logger Pro to be able to load it. This was justified because the individual algae were still clearly visible. Four different algae were chosen from the beginning of the video, four broadly from the middle of the video, and two from the end of the video. A scale was set over the width of the video to be 400 µm and each chosen particle was tracked using Logger Pro's point tracking with an advancement of 20 frames (20

seconds) after each position mark. The Y-positions were plotted against time with a linear fit. The sedimentation velocity was simply read off as the slope of the fits. See (A).

The velocities can be seen in figure (5.5) and vary from $-0.5426\,\mu m\,s^{-1}$ to $-1.595\,\mu m\,s^{-1}$. There are several uncertainties in this analysis e.g. the size and density of the algae may vary [9], which could also affect the sedimentation drift near the walls.

From figure (5.5) it can be seen that the last point is an outlier in contrast to the second to last. The two points are at opposite walls of the channel and their distances from the wall could be different than they appear due to a small rotation of the capillary tube or an optic effect because the left side of the tube has the transducer glued to it.

The sedimentation velocities are in agreement with theoretical values found in section (4.6) and it can be concluded that the back flow observed is not due to sedimentation of the algae.

5.6 Changes to the setup

5.6.1 Tubing

The initial setup came with a mix of silicone- and plastic tubing. In order to reduce the flow distance from the syringes to the RDH, excess tubing was cut. It was desirable to reduce the length of the tubing in the system as the parabolic flow profile of laminar flow, which was in the system, would lead algae to spread out through the length of the system as algae near the walls would move slower than the ones near the center. The silicon tubing was replaced with the plastic tubing where the silicone was not specifically needed. It was done because it was suspected that the silicone added substantial compliance to the system. See more on this in (4.7)

5.6.2 Short circuit

When running the initial tests with the new GCTH it was noticed that the algae flow persisted and the hypothesis about "algae-reservoirs" (5.3) was discarded. The flow could simply be due to hydrostatic pressure differences (4.5). The hydrostatic pressure difference could theoretically be canceled out by changing the elevation of the collecting cups. However, this turned out to be cumbersome to control. Instead, a short circuit over the RDH was implemented. From the theory, see (4.5), the flow through the capillary tube would be lower after the implementation of a tube with a much lower resistance running in parallel to it. This would help to equalize the pressure in the system to atmospheric pressure see (4.7).

First, a manually controlled short-circuit was made by inserting a t-junction before and after the RDH see (5.6). The two t-junctions were connected using the plastic tubing with a small section of the silicon tubing in the middle. The middle section served as the cut-off valve because the silicone could be clamped shut thus directing all flow to the RDH. To clamp the silicone tubing, a simple document clip was used. A flow could then be applied with a shut valve directing all flow to the RDH. Then the flow could be stopped and the valve could be opened directing (almost)[4] all the flow through the short-circuit. This proved to be a very useful method for keeping the algae still in the capillary tube. However, it was very manual and touching the setup doing the experiments often led to vibrations in the video recording.

To account for the aforementioned unwanted side effects a computer-controlled method for controlling the valve was wanted. Instead of the silicone/clamp method an automated switch valve (MX Series II^{TM} MXP9900-000 2-Position, 6-port - SiSW) was implemented. The switch valve has 6 ports that are connected as follows:

Position 1: 6-1, 5-4, 3-2. Position 2: 6-5, 4-3, 2-1

[4] At least equalize the pressure much faster

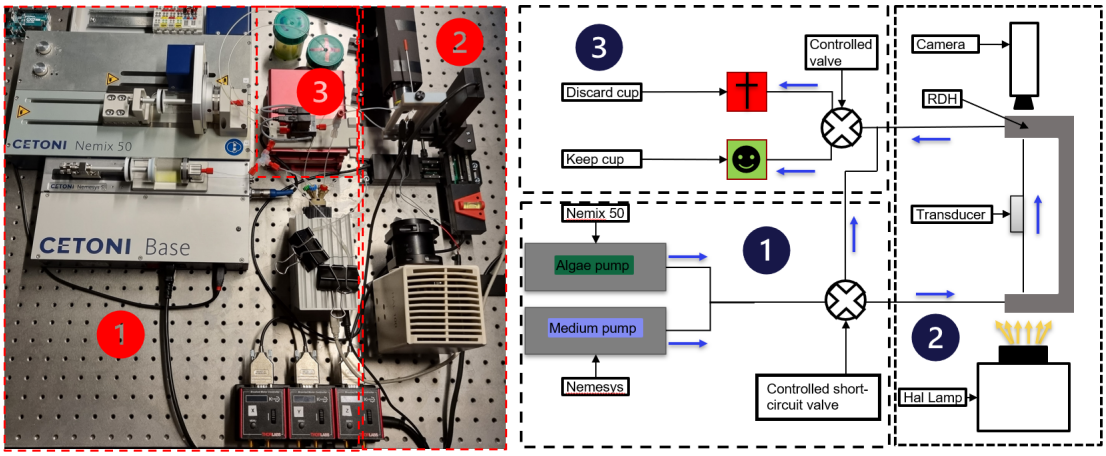


Figure 5.6: The final experimental setup. In addition to(5.1) the automated switch-valve has been implemented and the GCTH is now in gray to symbolize the the GCTH see (5.3.2)

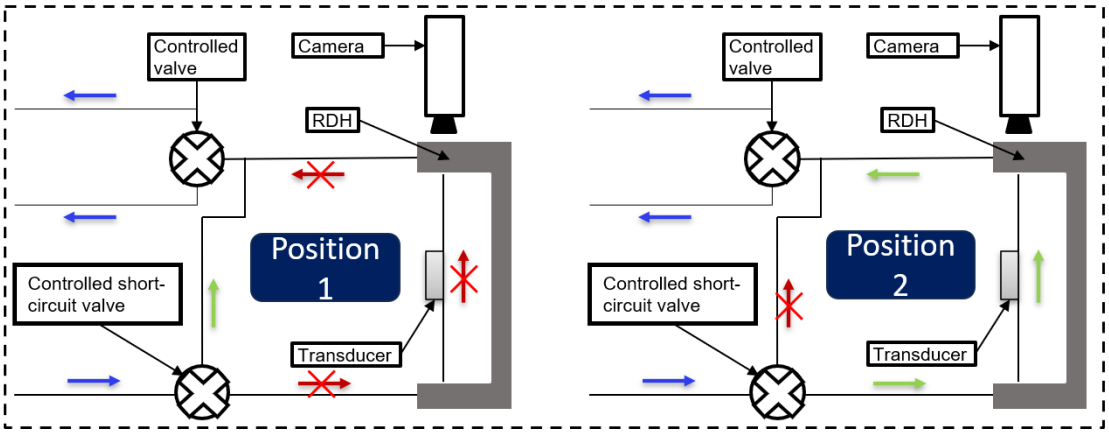


Figure 5.7: The diagram shows a zoomed in view of figure (5.6) how the controlled short-circuit valve directs flow through the system. In position 1 (open) the flow it let through the short-circuit and the flow is blocked to the RDH. In position 2 (closed) the flow is let through the RDH and the flow is blocked to the short-circuit

The inlet was then connected to port 1 connecting it to the short-circuit in "open" (position 1) through port 6. Port 1 was connected to the RDH in "closed" (position 2) through port 2. Ports 3 and 5 were clamped off because otherwise, the system would be open at the outlets of these ports, see diagram in figure (5.7). The switch valve was connected to the computer through USB. A small Python script was made to control the switch between positions, see (E). The valve could now be controlled manually and automatically through Python.

5.6.3 Camera

The camera used for (5.2) and (5.4) was a Basler a2A4504-18ucBAS which had a high-resolution of 20.2 MP, and a low frame rate. The maximum frame rate of the camera was 18 FPS though in practice only about 7.3 FPS were achieved. This proved to be too low for near resonance experiments because the slope of the curves in the t^*-fits would have too few data points for a reliable analysis (6). It was decided to change it to a Basler acA1440-220uc which had a low resolution of 1.6 MP, and a high frame rate. The lower resolution did not seem to affect the videos or data analysis in any noticeable way, and a maximum frame rate of 227 FPS could be achieved with this camera. However, a capped frame rate of 20 FPS was chosen to keep the rate constant as it was enough for the near-resonance experiments.

6 Video analysis

A key part of this project was to develop a reliable and preferably also computationally light way of analyzing the experimental data. Experimental data were gathered in the form of videos. The videos generally proceeded as shown in figure (6.1), at times before $t \leq 0\,\mathrm{s}$, the algae were uniformly distributed in the channel, at $t = 0\,\mathrm{s}$, the acoustic focusing was started which made the algae move towards the center of the channel through acoustophoretic motion, and when approximately all of the algae had moved to the center of the channel the video was stopped allowing for the start of a new video. Multiple approaches were considered for this analysis

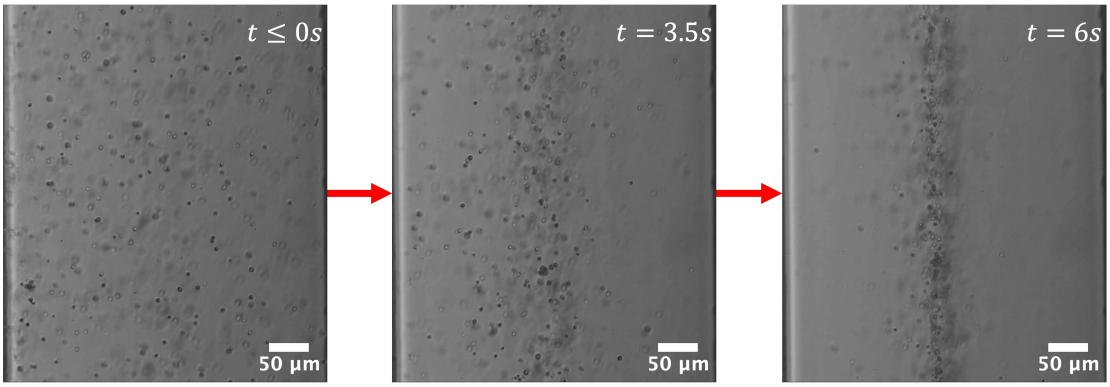


Figure 6.1: The figure shows three frames of an acoustic focusing at a frequency of 1.898 MHz, a signal amplitude of 1 V, and a signal amplification of 10 dB. $t = 0\,\mathrm{s}$ denotes the start of the acoustic focusing. The leftmost picture shows the channel at $t \leq 0\,\mathrm{s}$, where the algae are seen to be uniformly distributed throughout the channel. The second image shows an acoustic focusing at an intermediate stage, the algae are seen moving toward the middle of the channel. The third image shows the end of an acoustic focusing where almost all of the algae moved to the center of the channel.

of the recorded videos, but the chosen technique was one described in the following paper [4]. The paper describes a technique in which an expression of the acoustic radiation force is used to derive a function for the path traced by a particle undergoing acoustophoretic motion. This expression is then used along with observations of how many particles affect the light intensity in the interrogation area, seen in figure (6.2), during acoustic focusing to relate intensity changes in the interrogation area to the acoustic energy density. This results in the derivation of equation (90). It is important to note that a slightly different definition of alpha was used in this project resulting in the following conversion $\alpha' = 1 - \alpha$.

$$I_{Norm} = 1 - \frac{2R}{\alpha'\pi} \arctan\left[\tan\left(\frac{\pi}{2}\alpha'\right)\mathrm{e}^{-t/t^*}\right] \tag{90}$$

A key parameter in equation (90) is the characteristic time, t^*. This parameter was of particular interest as it is related to the acoustic energy density, E_{ac}, in the channel to the focusing time through equation (91).

$$t^* = \frac{3\eta}{4\Phi\frac{\pi a}{W}}\frac{1}{E_{ac}} \tag{91}$$

The different parameters in (91) apart from t^* and E_{ac} are the viscosity of the fluid, η, the acoustic contrast factor, Φ, the particle radius, a, and the channel width, W. As $E_{ac} \propto 1/t^*$, it means that the smaller a value of t^* that was recorded during a measurement the larger was acoustic energy density in the channel.

The relative reduction in light intensity from expression (90), R, could be approximated using the following approximation (92) which was given in [4].

$$R \approx 1 - \frac{I(t = 0\mathrm{s})}{I(t = \infty\mathrm{s})} \tag{92}$$

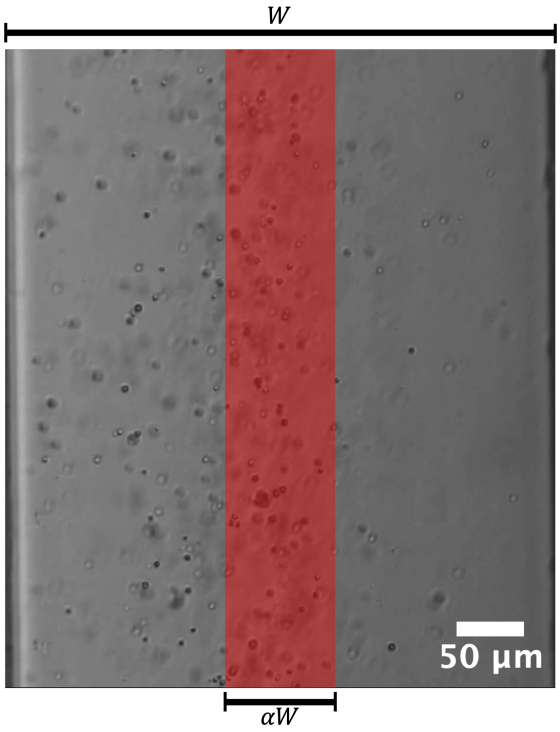


Figure 6.2: The picture shows a frame of an intermediate step of an acoustic focusing. W denotes the width of the channel, which was $W = 400\,\mu m$ for all experiments. The so-called alpha, $\alpha = 0.2$, has been highlighted with red in the center of the picture. Everything outside the alpha region will be mentioned as the interrogation area.

First $I(t = 0s)$ would be defined as the mean intensity in the interrogation area of the frame captured at time, $t = 0\,s$, thus it would have been generated by the leftmost image in figure (6.1). Secondly, $I(t = \infty s)$, also mentioned as I_{max}, was defined to be the mean intensity in the interrogation area of the last frame of each video, i.e. I_{max} would be defined from the rightmost image in figure (6.1).
The normalized intensity for each frame was then computed by finding the mean intensity in the interrogation area and then dividing it with I_{max}. From this, an array containing the normalized intensities of all frames in a video was constructed. This array could then be plotted versus the time, t, and function (90) could be fitted to it. An example of such a plot is seen in figure (6.3).

Initially, only the characteristic time, t^*, was used as a fitting parameter, and a value for the relative intensity reduction, R, was computed using equation (92). However, it was quickly found that the fits would match the data points more closely if R was also used as a fitting parameter. It was justified to use R as a fitting parameter as how much light was blocked by different algae, even in the same population, varied a great deal.

This approach to measuring the value of t^* relied on the intensity of light coming through the interrogation area thus it was sensitive to the concentration of algae in the solution. Throughout most of the project, some weak algae solutions like the one seen in figure (6.4a) were used. These solutions were of algae population 1.2 and 2.3. This weak concentration solution yielded data plots like figure (6.3) in which the increase in normalized intensity is a mere 0.5%.

The signal strength being as low as it was meant that the measurements were especially susceptible to noise and systematic errors as will be seen in section (7.3). Even though this signal strength was not great it still yielded usable measurements at a time when each experiment consisted of a relatively small amount of videos. As the setup became more autonomous it became more important for the signal strength to be stronger. This was achieved by creating a high-

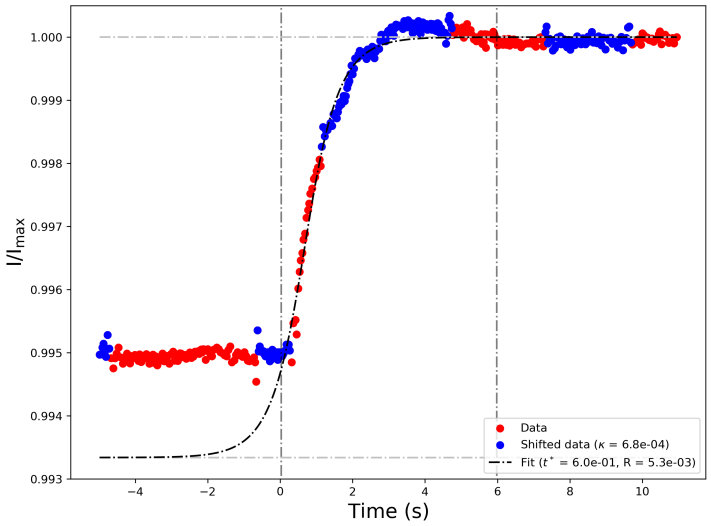
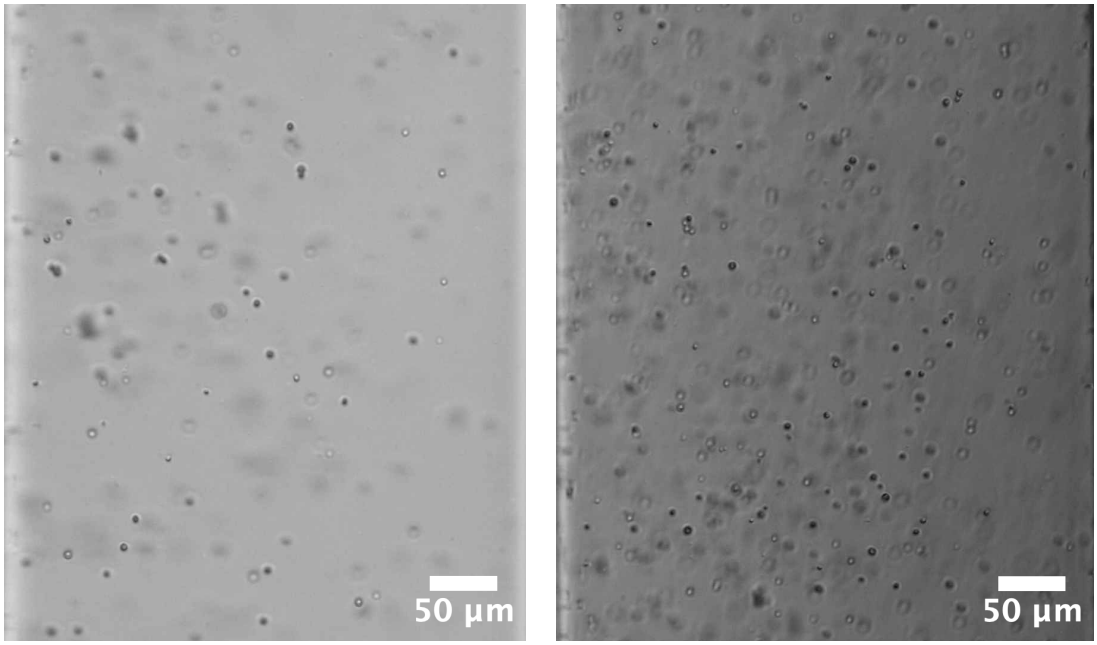


Figure 6.3: The figure shows a plot of the normalized intensities versus time during a video. The ultrasound frequency was set at 1.898 MHz. Function (90) is fitted to the data using t^* and R as fitting parameters. The blue points have all been shifted by a constant, κ, this will be discussed in section (7.3). A low-concentration algae solution was used for this experiment.



(a) (b)

Figure 6.4: (a) Low-concentration algae population (2.3). (b) High-concentration Algae population (1.1, 3.2, 3.21, 2.3). The figure shows that there was a clear difference in the algae concentration of each solution. The switch to a high-concentration helped to improve the signal to noise ratio, that otherwise gave problems with the low concentrations, see (7.4).

concentration algae solution which was done by combining algae from several populations into the same solution. It is important to note that there at this point were no differences between the different algae populations. To make the high-concentration solution the following populations were used, 1.1, 3.2, 3.21, and 2.3. They all sat undisturbed for several days allowing the algae to completely sediment. Approximately 90 % of the medium from the top of each cup containing

different populations was removed with a plastic syringe. Then each cup was thoroughly swirled as the algae tended to adhere to the bottom. After this, all populations were mixed into one cup while a little medium was added allowing the algae to reproduce.

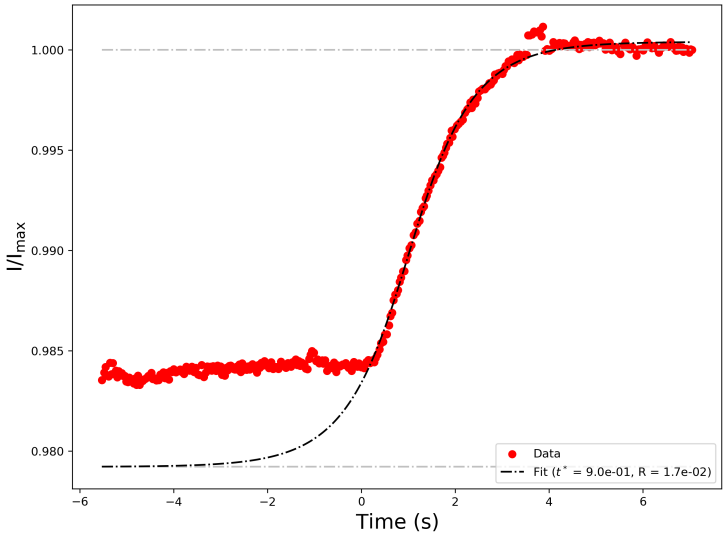


Figure 6.5: The figure shows a plot of the normalized intensities versus time during a video. The ultrasound frequency was set at 1.898 MHz. Function (90) is fitted to the data using t^* and R as fitting parameters. A high-concentration algae solution was used for this experiment.

The effects of utilizing a higher concentration of algae solution were substantial and the increase in signal strength as seen in figure (6.5) are clear. The increase in the normalized intensity of figure (6.5) is 1.6%.

It was observed that algae in the high-concentration solution tended to clump together. This was unwanted as the large clumps of algae sediment faster. This could mean that an algae clump could move in or out of frame creating a large jump continuities in the intensity. Algae clumps also focused much faster and affect the analysis making the acoustic energy density seem higher than it was.

The script used for the video analysis can be further view in appendix (E).

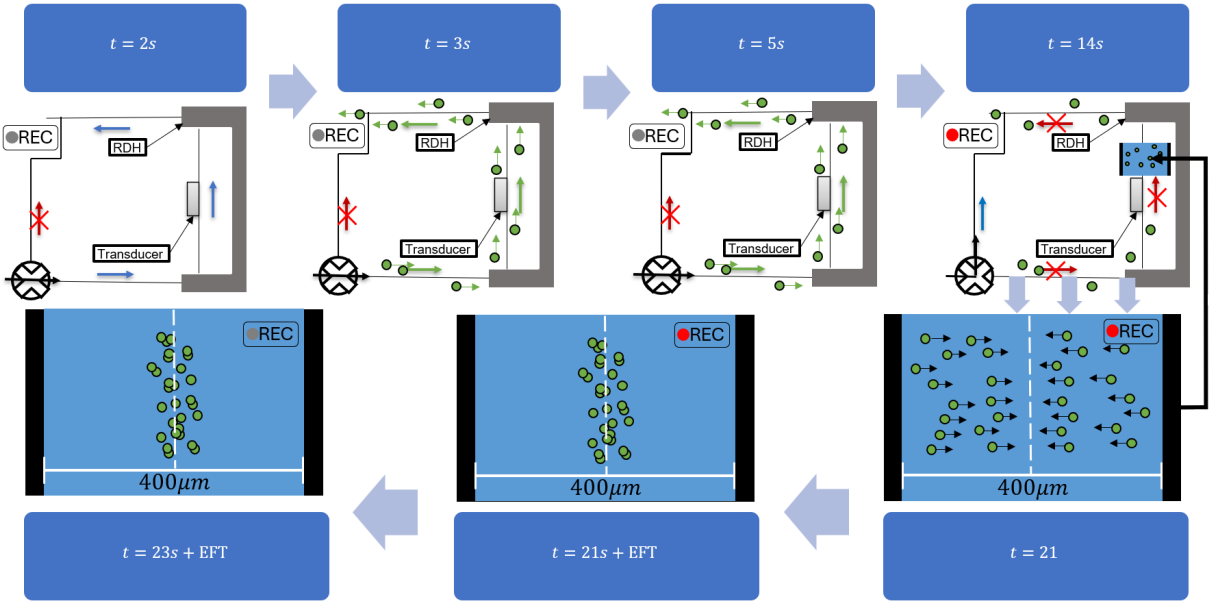


Figure 7.1: The flow chart shows the automated flow experiment also described in the process flow in this sub chapter. (Top) The arrows indicate the direction of flow. Blue arrows indicate that a flow can be applied, green arrows indicate that a flow is being applied and red arrows mean that the path is cut off. (bottom) On the bottom 4 figures, the camera view is shown and the presence of the small green arrow here means that a signal is being applied. The REC button shows if the camera is recording.

7 Main experiments

7.1 Automated flow experiment

With the changes made to the setup described in (5.6) and the experiences from experiments (5.2) and (5.4) a plan for a more automated experiment was drafted. After the start-up procedure (B) a desirable spot on the capillary tube was chosen for the experiment. With the x,y,z-stage the capillary tube was centered in the image. The optical focus was set to be approximately in the middle (depth-wise) of the channel. Tests were carried out to check that the part of the channel was not acoustically dead i.e. little to no acoustic energy density would be observed. This spot was kept throughout all of the following experiments. A Python script was made (E) that controlled all the automated experiments. The script worked with a sub-script in Cetoni Elements which could not be implemented in Python.

The Cetoni Elements-script would, when given a command through the Arduino, start a flow from Cetoni Nemix 50 (algae syringe see (5.6)) by initializing a loop that would continuously start flows at $7\,\mu L\,s^{-1}$ from the algae solution until $50\,\mu L$ had been dosed. Before the flow was terminated a new command would be sent through the Arduino terminating the loop sequence making the Cetoni Elements script wait for a new initialization after the dose was completed. The loop was set to run for 10,000 iterations to make sure the Cetoni Elements sub-script would not terminate before the Python main-script.

For the Python main-script the Arduino, Analog Discovery 2, and the camera were first initialized. Secondly, the image-feed was initialized. The script would then run the following loop for as many frequencies and repetitions on that frequency as desired. The different steps are initialized at times $t = t_0$.

1. $t = 2\,s$, the switch-valve was turned to "open", opening the flow to the RDH.
2. $t = 3\,s$, the initialization command was sent through the Arduino to initialize the Cetoni

Elements sub-script.

3. $t = 5\,\text{s}$, the termination command was sent through the Arduino to terminate the Cetoni Elements sub-script.
4. $t = 14\,\text{s}$, the switch-valve was turned to "closed" closing the flow to the RDH and at the same time initializing the camera recording.
5. $t = 21\,\text{s}$, the Analog Discovery applied the signal to the transducer at the specified frequency and voltage.
6. $t = (21 + \text{EFT})\text{s}$, the signal was terminated (EFT = estimated focus time).
7. $t = (23 + \text{EFT})\text{s}$, the recording was stopped, all data was saved and the loop was reset for the next run.

From step 3 to step 4 the wait time is longer than the flow time (approximately 7 s). Due to the flow rate being so high that the system needed to "calm down" because of compliance (4.7) making the added volume actually run to the capillary tube and not be equalized through the short circuit. At step 4 the pressure would be equalized through the short-circuit and the wait time to step 5 was to make sure that the algae were not flowing anymore. In step 6 the "EFT" was found before the experiment for the given frequency to limit the amount of data recorded after full focusing. For the experiment running sweeps of wide frequency bands the "EFT" would change depending on the frequency and was set to be a function: $\text{EFT}(x) = -|x| + x_0$ in the interval $[x_0 - x_f; x_0 + x_f]$ where x_0 was the focus time at the resonance frequency and x_f was the focus time at the last frequency in the sweep. Obviously, the focusing times are not linear in frequencies, but for the "EFT" it was sufficient. The EFT varied from $(5 - 40)$s from resonance to tail, adding substantial duration to the experiment running off resonance.

7.2 Estimation of uncertainty on t^*

In order to gain a proper estimate of the uncertainty on the parameter t^* and thus also on $1/t^*$ a separate experiment was carried out. This experiment was made utilizing the final setup and the high-concentration algae solution.
It was determined to estimate the uncertainty at 11 different frequencies, some of which would be near the resonance frequency and some would be off-resonance. This was done to ensure that if there was a difference in the uncertainty at resonance and off of the resonance frequency then this would not be overlooked. At each of the 11 frequencies, 20 measurements were made, and the average t^* value and the associated standard deviation were calculated. These can be found in appendix (C). The distribution of these uncertainties can be viewed in figure (7.2).

The uncertainty on t^* for each frequency examined was estimated to be the standard deviation of all the t^* values for that frequency, the relative uncertainty was then obtained by dividing that with the mean t^* value for that frequency. To convert an uncertainty for t^* to an uncertainty on $1/t^*$ one has to make use of the law of error propagation. The conversion function is derived in equation (93). The uncertainty for a parameter is denoted with a δ symbol in front of the parameter symbol and the sample mean of a parameter is denoted with a bar above the parameter symbol.

$$\delta\Big(\frac{1}{t^*}\Big) = \Big|\frac{\mathrm{d}}{\mathrm{d}t^*}\Big(\frac{1}{t^*}\Big)\Big|\delta t^* = \frac{1}{t^{*2}}\delta t^* \tag{93}$$

All of the absolute and relative uncertainties are shown in table (2). The relative standard deviation on t^* and $1/t^*$ appears to be independent of frequency and randomly distributed around the mean, as seen in figure (7.2). It was therefore decided that 6% would be a good estimate of the relative uncertainty on measured t^* and $1/t^*$ values.
A lot of effort has been put into improving the experimental setup throughout this project and it is therefore worth noting that while this estimate of the uncertainty on t^* might be accurate for the most recent experiments, it might be too small of an uncertainty for the initial experiments.

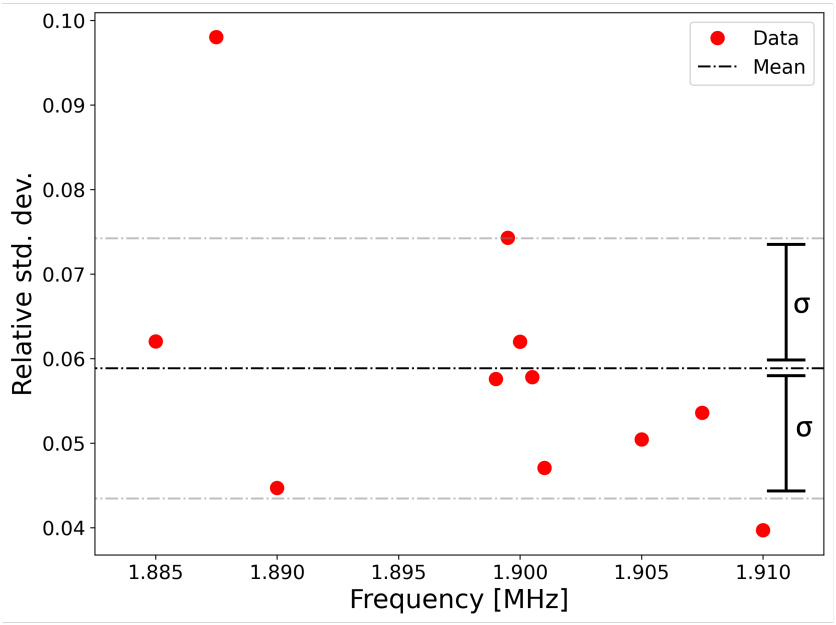


Figure 7.2: The relative standard deviation of t^* for each frequency is shown as the red dots in the plot. The black stippled line shows the mean relative standard deviation which is 0.06, and the grey stippled lines as well as the sigmas show one standard deviation from the mean relative standard deviation which is 0.02.

7.3 Resonance frequency of the capillary tube

One of the objectives of this project was to experimentally determine the first resonance mode of a capillary tube with cross-sectional dimensions of 400 µm × 400 µm. In section (4.4) a theoretical frequency of 1.875 MHz was determined to be the the first resonance mode. However several simplifying assumptions are made in this derivation, amongst these is the assumption that the walls are infinitely stiff, do not undergo any displacement, and that the channel has exact dimensions without uncertainties attached. While these assumptions allow for the derivation of an analytical expression for the resonance modes they also mean that the actual resonance modes will differ slightly from the theoretical ones. Previous work done by the "Theoretical Microfluidics Group" showed that the resonance frequency of the setup might lie at 1.91 MHz. In order to determine the actual resonance frequency of the capillary tube it was decided to carry out a so-called frequency sweep. The idea behind this was to make several measurements of t^* where the algae were acoustically focused at a range of different frequencies. The measured values of t^* would be small close to the resonance frequency, indicating a large acoustic energy density, and they would be large further away from the resonance frequency, indicating a low acoustic energy density.

The general methodology for analyzing videos of acoustic focusing is described in section (6). The acoustic energy density, E_{ac}, in the channel, increases the closer one is to the resonance frequency of the system. From article [4] it is known that the acoustic energy density, in the capillary tube, is inversely proportional to the characteristic time, t^*, meaning that $E_{ac} \propto 1/t^*$. Thus by utilizing the methodology from article [4] to generate a plot of relative intensities in the channel versus the time and then fitting function (90) to the data it is possible to extract a parameter value which describes E_{ac}. The fitting parameters that were used were t^* and the relative intensity reduction, R.

Several attempts to make the frequency sweep were made throughout the time frame of this

project. Early attempts made use of the initial setup which demonstrated several difficulties including a lack of temporal resolution, and low signal strength.
The first frequency sweep that resulted in the production of a plot displaying $1/t^*$ versus frequency, will be mentioned as a resonance plot, consisting of 17 measurements at frequencies between 1.86 MHz and 1.935 MHz. The experiment made use of algae population 2.3, which was a low-concentration algae solution, and the high-resolution camera mentioned in section (5.6.3). The result of this experiment is seen in the right plot of figure (7.3), and even though the plot and data are lacking in several ways, it hinted that the actual resonance frequency of the system might not have been 1.91 MHz, as was previously thought, but that the resonance frequency may be at a lower frequency.

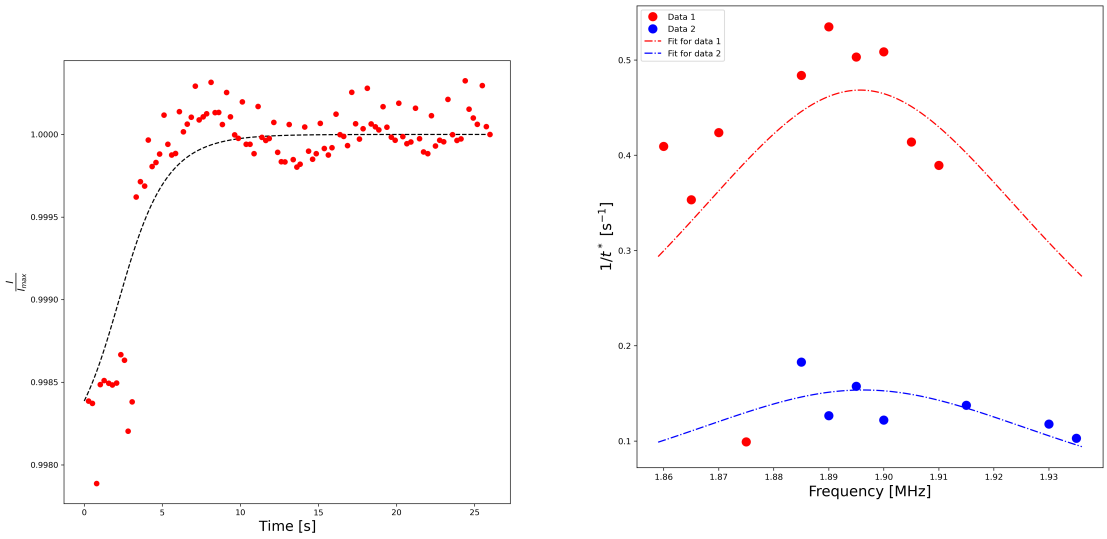


Figure 7.3: The left plot shows one of the first generated plots of the relative intensity of the interrogation area versus the time. The right plot shows a Lorentzian fitted to two data sets which were collected at the beginning of the project. Both plots were generated from data stemming from an early iteration of the experimental setup, and both plots are lagging at several points, regarding e.g. the temporal and spectral resolution of the measurements.

The left plot in figure (7.3) shows that the collected data does appear to follow function (90) and thus that the methodology described in article [4] would be a viable option in terms of analyzing the acoustic focusing of the algae. The plot also highlights a couple of areas in which the experimental setup could be improved. Firstly the temporal resolution of the setup appears to be too low as few or no data points were collected at the sloped part of the measurement. Secondly one notices that the signal strength is low as the difference between the maximum and minimum relative intensity is approximately 0.15% making the system susceptible to the interference of noise.

Multiple experiments were made afterwards, all of which were carried out after changes to the experimental setup had been made. The first experiment which resulted in a finished $1/t^*$ versus frequency plot consisted of 31 videos that were recorded at frequencies between 1.884 MHz and 1.906 MHz. The limiting factor with regards to the frequency resolution of this experiment was that the data collection had not been made fully autonomous yet, thus a human had to be present for the entire duration of the experiment in order to start and stop each measurement. Thus limiting the viable amount of measurements that could be made.

The second experiment utilized a different video camera, the low-resolution one mentioned in section (5.6.3), which greatly improved the temporal resolution from the first experiment. This is visible as data points are now recorded at the entire length of the slope. The algae solution utilized was still the low-concentration algae population 2.3. It made the issue regarding a low

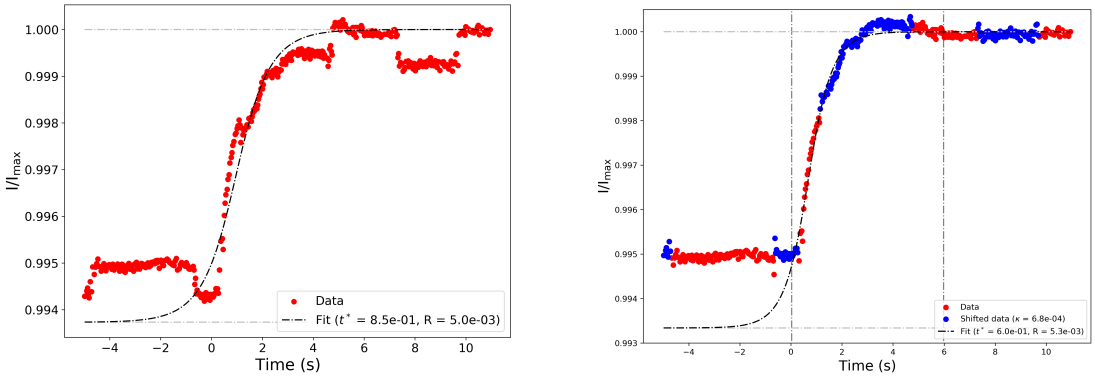


Figure 7.4: Both plots were generated from data stemming from an iteration of the experimental setup where the temporal resolution has been improved but where the signal strength was still weak. Both plots show the relative intensity measurements from a video showing acoustic focusing at a frequency of 1.898 MHz. Both plots have functions (90) fitted, shown as the stippled line. Systematic artifacts are seen in the left plot which has fitting parameters $t^* = 0.85 \pm 0.05$s, and $R = 0.005$. These artifacts have corrected for as a constant $\kappa = 6.8 \times 10^{-4}$ have been added to the blue points in the right figure, having fitting parameters of $t^* = 0.72 \pm 0.04$s, and $R = 0.004$.

signal strength is made apparent through the artifacts seen in the left plot of figure (7.4). These artifacts are visible on every measurement made during that experiment. Why the artifacts are present in all of the measurements is still unknown, however, the suspicion that the artifacts are systematic is confirmed by the fact that they occur almost exactly the same times in each measurement, only deviating by up to a few hundred of a second between measurements. A further confirmation of the artifacts being systematic is that they can be and were corrected for in every measurement by adding or subtracting a constant denoted as κ, with a value of $\kappa = 6.8 \times 10^{-4}$. After manually correcting all of the measurements a plot of $1/t^*$ versus the frequency was made resulting in figure (7.5).

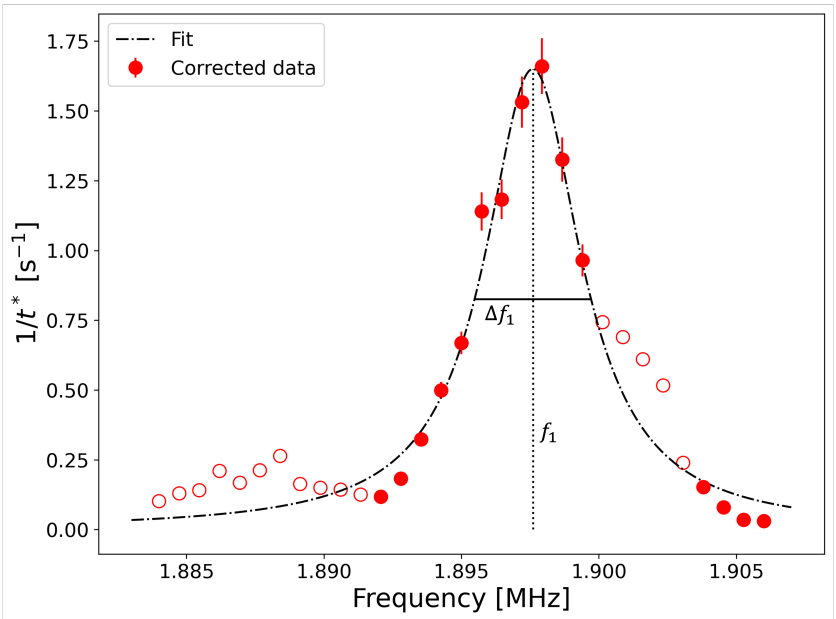


Figure 7.5: The plot shows $1/t^*$ versus the frequency. The t^* values are extracted from the measurement where the artifacts had been corrected. The peak frequency is found from the fit to be $f_1 = 1.898$ MHz, the full width at half max (FWHM) is $\Delta f_1 = 0.004$ MHz, and the quality factor of the fit is 449. A Lorentzian function has been fitted to the data and only the red-filled dots have been used to create the fit.

The improved temporal resolution made a big difference in how accurate and precise the measurements of t^* were. Furthermore, the spectral resolution of the plot was greatly improved compared to figure (7.3) thus allowing one to see that the resonance frequency lies below the previously thought 1.91 MHz. Even though the spectral resolution of figure (7.5) had gotten to a good enough level for the first resonance mode of the system to be determined it was still desirable to increase the spectral resolution of the system. This was desirable as the bunch of excluded data points between 1.9 MHz to 1.903 MHz could indicate that the peak in figure (7.5) would comprise of two peaks very spectrally close to each other. It was clear that the experimental setup had to be made fully autonomous for better spectral resolution to be achieved.

Fully autonomous data collection was achieved with the final iteration of the setup. However, the lack of signal strength became apparent yet again as the artifacts discussed concerning figure (7.4) continued to be present in all of the measurements taken. As a much larger quantity of measurements were collected to improve the spectral resolution the artifacts could not be corrected manually as was previously done. The solution to this was improving the concentration of the algae solution, as discussed in section (6), this served to improve the signal strength. As can be seen in figure (7.6) the signal strength was increased from the previous 0.15% to 1.7%.

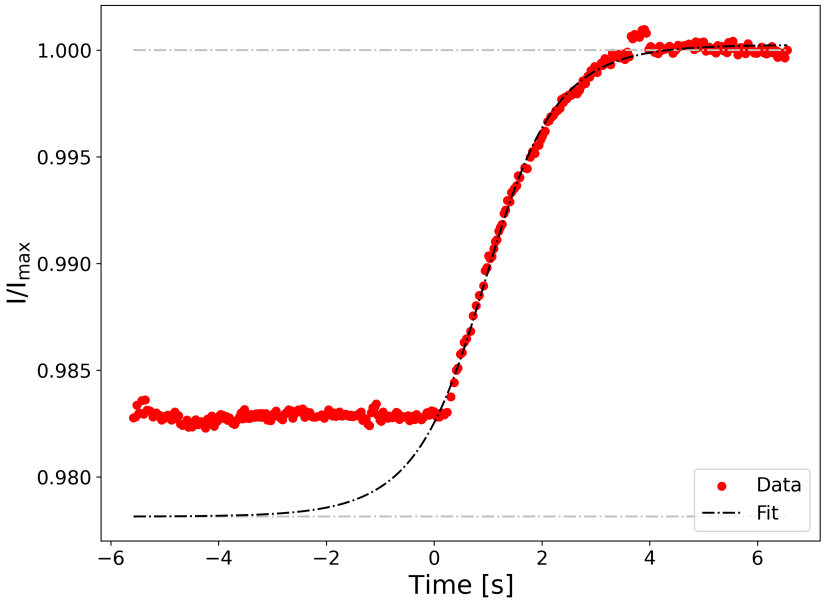


Figure 7.6: The plot shows relative intensity data an acoustic focusing at 1.898 MHz made in the third experiment. Function (90) was fitted to the data yielding values of $t^* = 0.84 \pm 0.05$s and $R = 0.02$. An artifact like the ones previously described is seen at around 3 s to 4 s, however, its influence on the measurement is much smaller as the signal strength is much larger.

As with the previous experiments related to figuring out the resonance frequency of the system, a plot of $1/t^*$ versus the frequency is made. The constructed plot is seen in figure (7.7).

As can be seen in figure (7.7) a two-peak Lorentzian fit models the data very well, confirming the suspicion that two peaks might have been close to each other. From the model, one can further see that the two peaks lie only 4 kHz apart and that the peak at 1.902 MHz is much smaller than the one at 1.898 MHz. Figure (7.7) further confirms that the resonance frequency of the system lies at 1.898 MHz. A possible reason for why a double peak Lorentzian would fit the data well is that if the channel cross-section is slightly off-square then two resonance modes would be found close to each other rather than just a single mode.

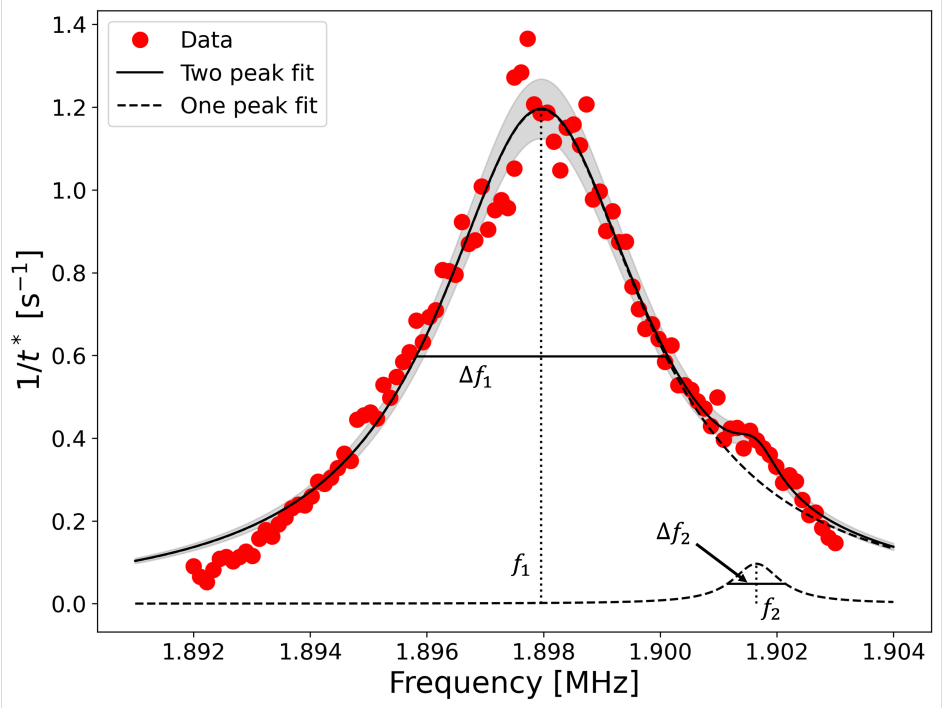


Figure 7.7: The plot shows $1/t^*$ versus frequency. A two peak Lorentzian function, as well as two one-peak Lorentzian function has been fitted to the data. From the plot a resonance frequency of $f_1 = 1.898\,\text{MHz}$ is found, the one peak fit around f_1 has a FWHM of $\Delta f_1 = 0.004\,\text{MHz}$, and a quality factor of $Q_1 = 442$. The second peak in the fit, f_2, has a peak frequency of $f_2 = 1898\,\text{MHz}$, a FWMH of $\Delta f_2 = 0.001\,\text{MHz}$, and a quality factor of $Q_2 = 1898$. The two peak fit has a $r^2 = 0.98$. The grey band in the plot shows $\pm 6\%$ deviation from the two-peak fit.

7.4 Correlation between voltage and t^*

For a general microfluidic setup, like the one utilized in this project, a piezo-transducer is attached to a capillary tube and is driven by a driving voltage, U_{pp}. From article [5] it is known that the acoustic energy density, E_{ac}, is proportional to the square of driving voltage of the piezo-transducer meaning $E_{ac} \propto U_{pp}^2$. As is known from article [4] the parameter, t^*, is proportional to the acoustic energy density meaning that $E_{ac} \propto 1/t^*$. Using both of the relations involving acoustic energy density one finds the relation seen in equation (94).

$$\frac{1}{t^*} \propto U_{pp}^2 \tag{94}$$

The objective of this part of the project is to investigate whether this proportionality holds for the setup used in this project. Two experiments were conducted throughout this project to study this. The first experiment was conducted a few days after the second resonance experiment was made, and thus the same experimental setup was used. Meaning that the temporal resolution of each measurement was good, however, the signal strength of the first experiment was not ideal as can be seen from figure (7.8).

The first experiment in which a voltage sweep was conducted, consisted of 20 measurements at linearly spaced output voltages going from 0.5 V through 8.9 V. The second experiment consisted of 100 measurements at driving voltages between 1.6 V through 8.9 V, though only the last 76 measurements were utilized in the analysis. The reason for excluding the first 24 data points of the second experiment is that the acoustic focusing was too slow for the video analysis to properly capture the change in intensity of the interrogation area meaning that

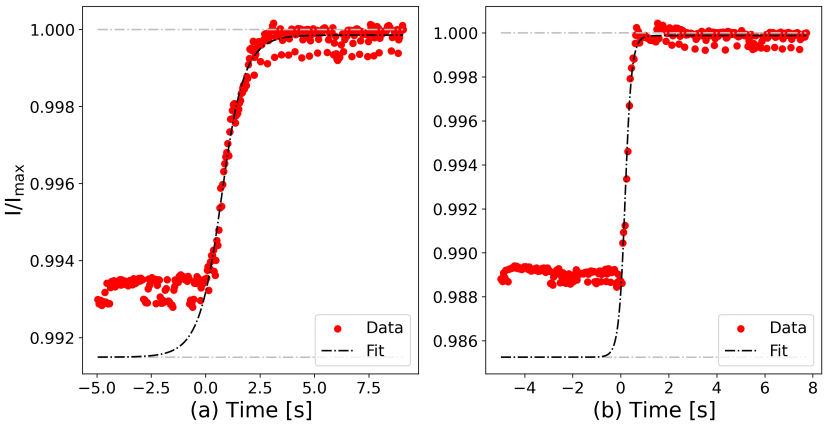


Figure 7.8: Both figures were made during the first voltage sweep experiment. The frequency used was held constant at 1.897 MHz. Figure (a) used a driving voltage of 4.5 V, yielding $t^* = 0.68 \pm 0.04$s. Figure (b) used a driving voltage of 8.9 V, yielding $t^* = 0.17 \pm 0.01$s. By examination of both plots, one sees that while figure (a) is well resolved even along the sloped region, figure (b) has reached the limit of what could be properly resolved with the at that time present setup, as the sloped region is starting to be poorly resolved.

this experiment was effectively a voltage sweep from 3.2 V and through 6.3 V. For the first experiment, a low-concentration algae solution was used along with manual stoppage of the flow in the capillary tube, as automated experiments had not been achieved at this time of the project. The second experiment was made using a high-concentration algae solution and the achievement of an automated experimental setup allowed for much more measurements to be captured. The experimental setup allowed for driving voltages up to 22.4 V to be used. The reason for not going beyond a driving voltage of 8.9 V was that the experimental setup was not able to properly resolve acoustic focusing at that quick time scale. After extracting t^* values from all of the measurements two plots were constructed. These are seen in figure (7.9). Both figures display a slope of 2.1 thus confirming the suggested proportionality of equation (94).

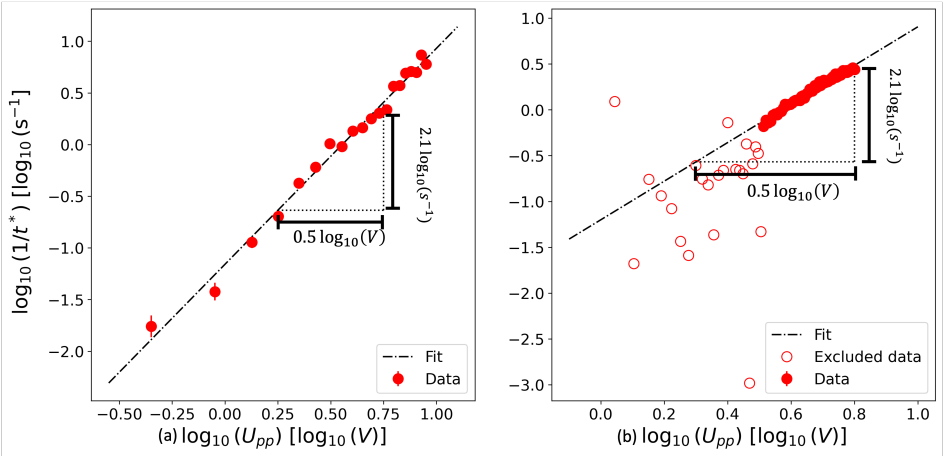


Figure 7.9: The left plot The plot shows log-transformed data with a line fitted to it, $y = ax + b$. The numeric value of a is found to be $a = 2.1$ for both plots and the numeric value of b is $b = -1.16$ for the left plot and $b = -1.20$ for the right plot. Only filled points have been used to generate the fit.

Starting with only an assumption of $1/t^*$ being related to the driving voltage through a power law yields equation (95).

$$\frac{1}{t^*} = b \times {U_{pp}}^a \tag{95}$$

By taking the logarithm of equation (95), a linear relation is obtained. This is seen in equation (96).

$$\log_{10}\left(\frac{1}{t^*}\right) = a\log_{10}(U_{pp}) + \log_{10}(b) \tag{96}$$

Fitting equation (96) to the log-transformed data would then give that the value of a equals the power of U_{pp} which $1/t^*$ is proportional to. As can be seen from figure (7.9) suggest that $1/t^* \propto {U_{pp}}^2$ would describe the data well.

This suggestion is further confirmed by figure (7.10) where $1/t^*$ versus ${U_{pp}}^2$ is plotted for both experiments. From the plot it is seen that the fitted line describes the data very well, suggesting that a linear relation between $1/t^*$ and U_{pp}^2 is accurate.

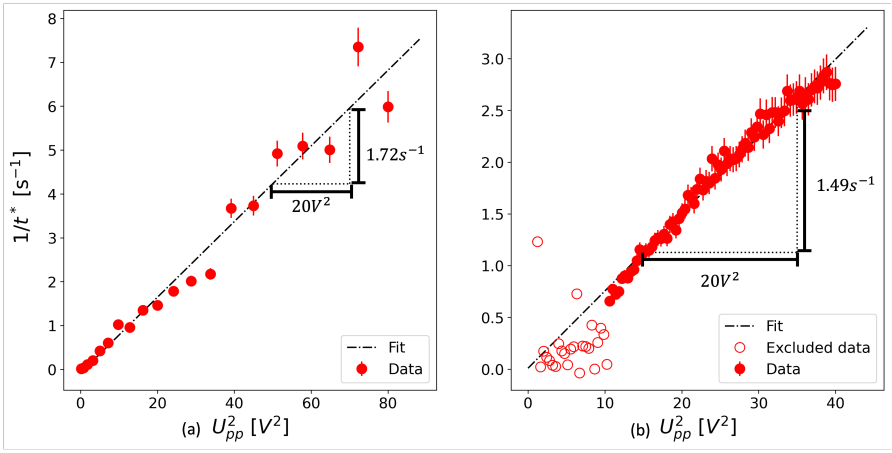


Figure 7.10: The fit to the data in figure (a) has a slope of $0.09\,\mathrm{s}^{-1}\,\mathrm{V}^{-2}$, and it intersects the y-axis at $-0.08\,\mathrm{s}^{-1}$. The fit to the data in figure (b) has a slope of $0.07\,\mathrm{s}^{-1}\,\mathrm{V}^{-2}$, and it intersects the y-axis at $0.009\,\mathrm{s}^{-1}$. As the data follows the fitted lines closely, it shows that the relation $1/t^* \propto {U_{pp}}^2$ holds.

7.5 Distribution of t^* at resonance frequency

For this final experiment, the hypothesis stated in (3) was checked using the automated setup (7.1). It was expected that $t^* \sim \mathcal{N}(\mu, \sigma^2)$ and given enough data point this could be shown. The setup was set to run for 500 runs at resonance (1.898 MHz). The experiment was however cut short due to time limitations and ended up being 404 runs, but this should still be at least an adequate amount of data points if not a good amount. The data was fitted for t^* as described in (6) and the distribution is shown in figure(7.11a) with the data from figure (7.11b). All data points in figure(7.11b) are used figure (7.11a)

The data points need to be independent of each other to infer anything physical based on a single run of algae. When looking at figure (7.11b) it is clear that this is not the case. In the figure it can be seen there are two linear tendencies which are fitted by a purple and green line respectively. It was suspected that this was caused by an increase in temperature throughout the experiments. The experiment was lit up using a Hal lamp that had a halogen light bulb set to 10 V and 3.5 A giving a power of 35 W. Halogen light bulbs are generally considered to have an efficiency of around 10-20%. It is therefore not unlikely that a lot of heat was generated in the setup. The jump at run 275 could be due to other people using the lab during the experiment e.g. having the door or a window opened at that time. Each run took ca. 30 seconds meaning that the six runs between the linear fits would amount to three minutes and it is not unlikely that some event happened that could alter the temperature over that time.

From [7] it is known that the propagation speed of sound in water is dependent on temperature.

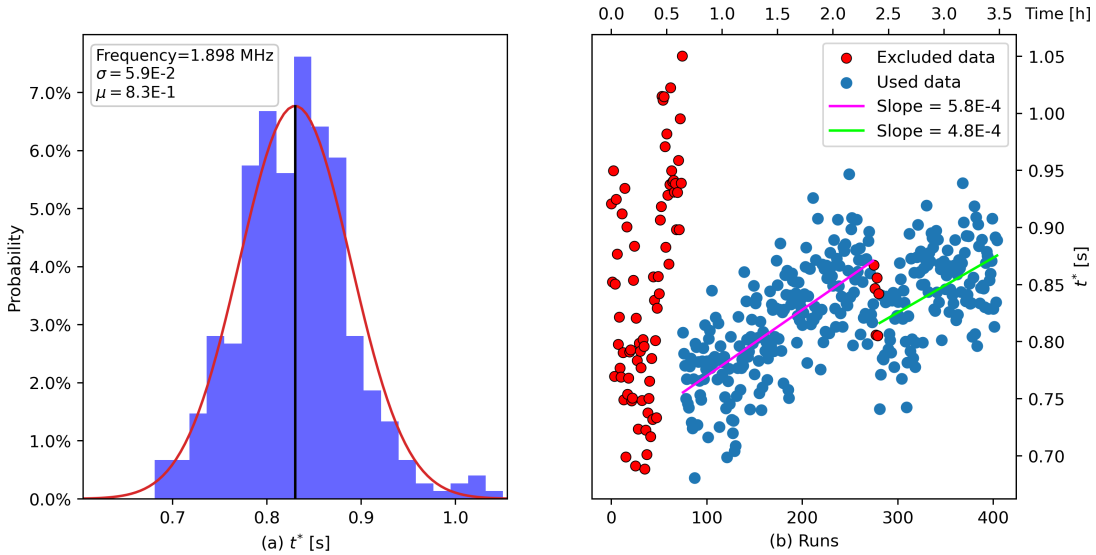


Figure 7.11: (a) shows the distribution of t^* over 420 runs at resonance frequency (1.898 MHz) of uncorrected data. (b) shows the raw data of t^* over the runs/time in chronological order. The data points are dependent on temperature and are shown corrected in (7.13). The red points are excluded from further analysis. The slopes of the purple and green lines are almost identical suggesting that the data points in the green line region are shifted by a constant value for all points indicating a temperature drop at the jump continuity.

It is shown that a fifth-degree polynomial model the speed as a function of temperature in the range $283\,\mathrm{K} - 323\,\mathrm{K}$ and is given as:

$$P_5(T) = C_0 + C_1\widetilde{T} + C_2\widetilde{T}^2 + C_3\widetilde{T}^3 + C_4\widetilde{T}^4 + C_5\widetilde{T}^5, \widetilde{T} = \frac{T}{1\mathrm{K}} \tag{97}$$

The numerical values for C_n are found in [6]. With this, the increase in speed was plotted for an increase of 1 K at room temperature (295 K)(7.12). The speed is seen to increase $2.88\,\mathrm{m\,s^{-1}}$ from $1488\,\mathrm{m\,s^{-1}}$. The resonance is assumed to be 1.898 MHz at 295 K and therefore it must hold that the wavelength is:

$$\lambda = \frac{1488\,\mathrm{m\,s^{-1}}}{1.898\,\mathrm{MHz}} = 784\,\mu\mathrm{m} \tag{98}$$

given that the frequency is proportional to the speed with λ^{-1} the frequency is seen to increase from 3.7 kHz to 1.902 MHz. Using figure(7.7) 1.902 MHz lies on the secondary peak off resonance. Given that this was only for a 1 K increase this suggests that the system is very temperature sensitive.

Using the purple fit from (7.11b) the first and last t^* values are extracted to check the decrease in $1/t^*$.:

$$\frac{1}{t^*_{\mathrm{diff}}} = \frac{1}{t^*_f} - \frac{1}{t^*_i} = \frac{1}{1.15} - \frac{1}{1.32} \approx -0.18 \tag{99}$$

Again looking at (7.7) a drop in t^* by 0.18 from the peak would be a small increase in temperature. This supports the hypothesis that the lack of temperature control is the cause of the upward tendency of t^*

Assuming the hypotheses above the data was corrected. All the red data points were removed (7.11b). The first 75 points were because they did not resemble tendencies in the purple and

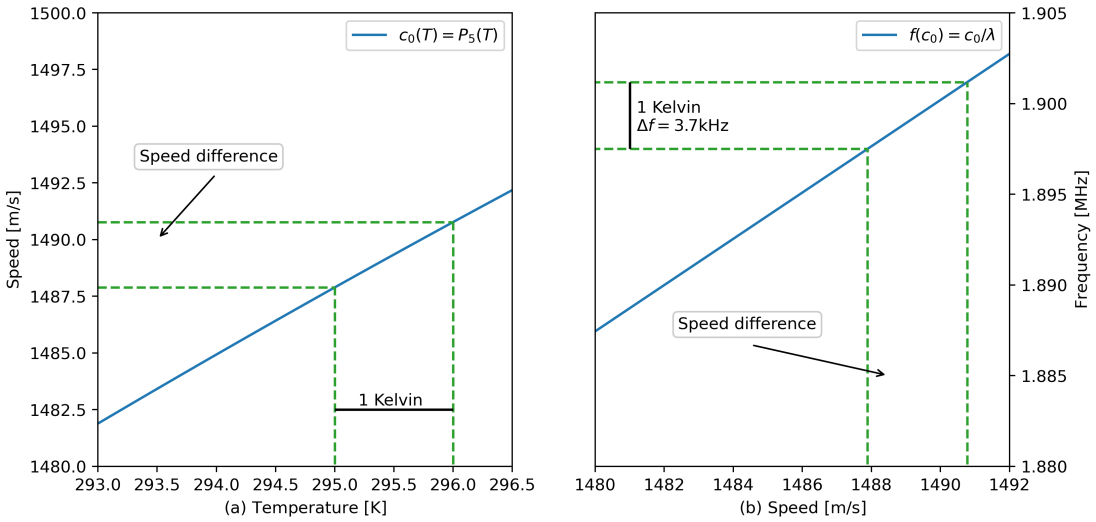


Figure 7.12: (a) shows $c_0(T) = P_5(T)$ (blue curve) which is the propagation speed of sound in water as a function of temperature in kelvin (a fifth-degree polynomial zoomed in). The dashed green lines protruding from the first axis indicate a temperature increase of 1 K from room temperature (22 °C). This gives the speed increase shown on the second axis (2.88 m s^{-1}). (b) shows the same increase in speed on the first axis which in turn gives rise to an increase in the resonance frequency of 3.7 kHz shown on the second axis

green fits. It could be explained by activity in the lab, but the cause is unknown. The points 275-281 were removed because they were in the jump continuity. Six points were chosen as that would give the closest slope fit of the green line compared to the purple line. This was justified as it was assumed that the green and purple had to have the same slope. The jump continuity was assumed to be because of a temperature disturbance. The data was then subtracted from the fits giving normal distributed data with $\mu = 0$. The t^* value is never zero and since the experiment shows the t^*, as the above mentioned function of temperature, the constant valued t^* was added to the mean of the first 5 data points in the purple fit, as this was assumed to be close to room temperature.

The result is shown in figure(7.13). In (b) the corrected data is shown and exhibits only the constant valued tendency. The data in (b) was used to create the histogram in (a) clearly showing that $t^* \sim \mathcal{N}(\mu, \sigma^2)$. (7.11a) and (7.13a) share the same range on the first axis clearly showing a smaller variance in the corrected data. Nothing should be inferred from comparing the means of the two as the uncorrected data mean does not reflect the t^* value at a given temperature whereas the corrected data mean does.

From the original motivation for the thesis, algae wanted to be separated based on an acoustofluidic principle. It was especially interesting to be able to separate the slowest-focusing algae from the rest of the population to see if the slow focusing could be correlated with some biophysical parameter of the algae. On figure(7.13a) a region is shaded green. This region contains the runs $< \mu - 2\sigma$ and the algae here could be analyzed if a method of separating them from the rest of the population exited. This is discussed further in (8).

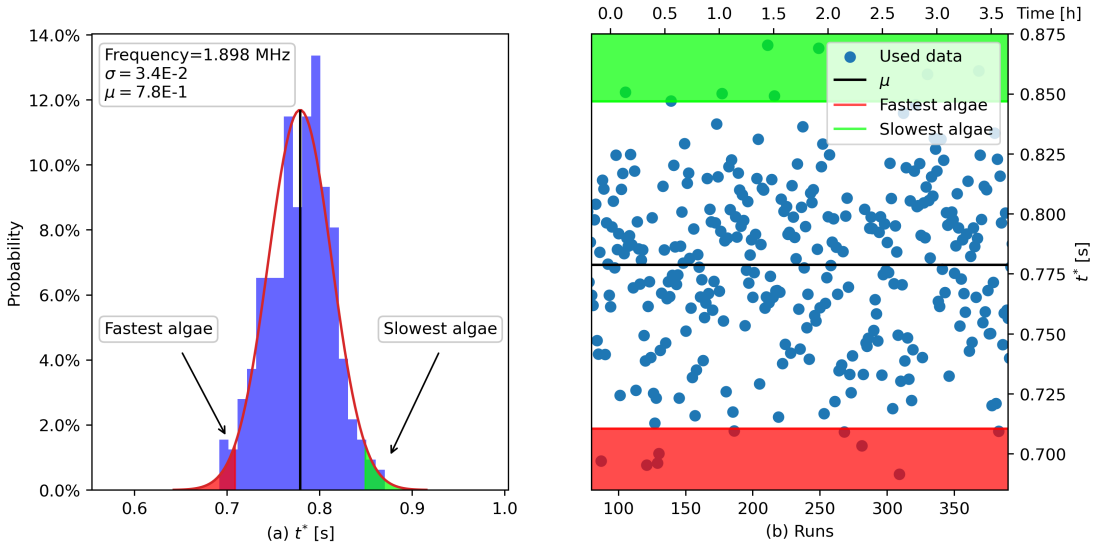


Figure 7.13: Shown in (a) is the Gaussian distribution of the corrected data. The green shaded region on the left tail indicates the region where $t^* > \mu - 2\sigma$ and where the slowest focusing times are found. The region is also shown in (b) where the specific runs can be seen. In (b) the raw corrected data can be seen. It shows that the data does not tend up or down over the runs anymore as was the case in (7.11b)

8 Discussion

8.1 Uncertainty on t^*

A large part of this project was spent on improving the experimental setup as well as the data analysis. This was done such that a value of t^* at a given frequency could be determined with a high degree of accuracy and precision. In section (7.2) a relative uncertainty for both t^* and $1/t^*$ was estimated to be 6%. This uncertainty estimate was made using statistical analysis of experimental data in which the sample standard deviation was used as a measure of the uncertainty on t^*.

The standard error on the mean is a measure of how precise the sample mean is to the true mean of the system. This measure is proportional to one over the square root the number of samples collected, $\sigma_{\bar{x}} \propto 1/\sqrt{N}$, where N is the number of samples used to compute the mean. This relation implies that the more measurements the better the estimate of the mean is. It could therefore be desirable to increase the number of samples gathered in the uncertainty experiment.

The estimated uncertainty on t^* indicates that precise and accurate measurements of t^* are made using the final experimental setup along with a high-concentration algae solution. While the uncertainty result has been extrapolated to earlier iterations of the experimental setup and measurements thereof, one ideally would have made an uncertainty experiment at each iteration of the experimental setup. This would have allowed for insight into how the accuracy and precision of the setup varied through the course of the project. Other suggestions for improvements could be to incorporate the uncertainty measurements into e.g. the experiment in which the resonance frequency of the system was determined. This could have been done by doing several repeated measurements at a constant frequency and then using the mean value of that sample as the best estimate of the t^* value at that frequency and simultaneously using the standard error on the mean of that to estimate an uncertainty of that t^* value.

8.2 First resonance mode of the system

One of the main objectives of this project was to characterize the experimental setup. Part of this involved experimentally verifying the first resonance mode of the system. Simplified analytical derivations in section (4.4) showed that the first resonance mode was at a frequency of 1.875 MHz, while previous work done by the "Theoretical Microfluidics Group" showed that the actual resonance frequency lied at 1.91 MHz. It was decided to carry out a designated sweep over a range of difference frequencies where the value of t^* was measured at each frequency, as discussed in section (7.3). This sweep resulted in the first resonance mode of the system being determined to be at a frequency of 1.898 MHz, as can be seen from figure (7.7). The general result of this good, and further analysis could involve studying the temperature dependence of the resonance frequency. The no-yield experiment which was discussed in section (5.4) would be beneficial to do as it would reveal if the first resonance mode had positional dependency in the channel.

8.3 Correlation between voltage and t^*

From the theory described in [5] $\frac{1}{t^*} \propto U_{pp}^2$. This was examined by conducting an experiment where algae were focused at resonance frequency (1.898 MHz) with driving voltages between 0.35 V and 2.0 V with 100 steps (7.4). The first 24 data points were excluded because of too little signal strength. The remaining data points were plotted as U_{pp}^2 vs. $1/t^*$ giving a linear fit with intersect at $0.0\,\mathrm{s}^{-1}$ in agreement with what would be expected. The slope was found to be $0.7\,\mathrm{V}^{-2}\,\mathrm{s}$

More experiments off resonance should be done to see if the model needs correction or if the setup needs to be controlled better.

8.4 Distribution of t^* at resonance

From (3) it was hypothesized that the focus time would vary with each algae package. It was initially wanted to see if this variation could be linked to some bio-physical parameter of the algae cell. In particular, the hypothesis was that algae cells with weaker cell membranes would focus slower. The focus time is related to the acoustic energy density and it must be constant for a given frequency and amplitude meaning that variations in focus time could be linked to the algae itself.

In (7.5) 323 runs at resonance frequency (1.898 MHz) and with an amplitude of 10 V was fitted for t^*. t^* was shown to be normally distributed with $\sigma = 3.4 \times 10^{-2}$ s supporting the hypothesis. In figure (7.13a) two regions are marked "slowest" and "fastest". In these two regions the the algae are hypothesized to exhibit a small variation in their cell membranes. If a method for separating the slowest algae from the fastest is implemented in the setup. The hypothesis, on the link between algae membrane and focus time, could be checked.

The t^* values were expected to be normally distributed around some mean value but were seen to increase linearly in time. It was shown in (7.5) that this could be due to a temperature increase over time. Initially, the experiment ran for 404 runs (7.11), but 81 runs were excluded mainly because they did not fit this tendency. This gave the result of 323 runs. If this approach is used in the future the temperature dependence has to be controlled before any separation of algae can be made. The method is promising and if the separation can be achieved it can the hypothesis could be checked potentially opening paths for future research. At this stage, nothing should be inferred about the algae membrane, but the result shows that some could be if the temperature could be controlled

8.5 Separation of algae

The thesis shows that a separation of the algae could be done. To do this, first the incorporation of a real-time evaluation of the algae in the experiments needs to be made. This could prove to be computationally heavy and/or slow. Second, the selection of what proportion of the algae to separate needs to be chosen. Either the population needs to be split in half dividing between the slowest- and fastest half, or a fraction of the slow half (as shown in (7.13)) needs to be separated from the rest. Either method should be repeated or an iterative process should be implemented. It would be interesting to check both methods to see if they yield different results. Throughout this project, it has been empirically observed that the large algae (or even clumps of algae) are more susceptible to the radiation force, and that the smaller ones are more prone to the radiation scattering effect. This is in agreement with the theoretical foundation. It would be interesting to see if the implemented method for separating the algae would instead separate based on the size of the algae and if so study if this method is useful for sorting based on the size of the algae.

It is known that the algae are exposed to the largest energy density at the anti-notes of the capillary tube. If the algae could be focused for long enough to the second-order resonance frequency nodes, the following focus to the first-order resonance frequency (center) node would give the focusing algae the same starting condition. This would leave a trail of the slowest algae behind the fast-focusing ones. This is known to be possible from a small experiment conducted where the amplifier was set to max amplification and the frequency was set to 2 times the first-order resonance node. This would make the algae focus on the second-order node. The amplification could then be lowered again and the frequency set back to resonance of the first order. From here the algae would focus on the center node. This experiment was not well suited for the transducer at used as it was fabricated to have a resonance at the first-order resonance frequency of the capillary tube. A way to fix this problem could be to have two capillary tubes in parallel before the main capillary tube. The algae would focus on their respective nodes and join to make two streams of algae in the main capillary tube. The problem with this is that the focusing process can not be done more than ones. A method for having two transducers on the same capillary tube or one that could tune the resonance would be preferable as the same sample of algae could be focused multiple times.

9 Conclusion and outlook

The overall objective of this project was to develop an experimental setup that allowed for automated handling of cells, hereunder algae. Throughout the project, five main experiments have been completed successfully with many more being part of the continuous development process of the experimental setup.

Many milestones have been reached in terms of the experimental setup. Starting with the replacement of the old device holder with a new version which enabled rotational movement of the capillary tube. This made fine-tuning of the captured video quality much easier allowing for better quality data to be recorded. Further improvements concerning the experimental setup were the introduction of the bypass channel allowing for almost stopping the flow of algae solution through the capillary, and later with the introduction of the automated switch valve full stoppage of the algae solution in the capillary tube at the desired times were made possible. Most recently full automation of the experimental setup was achieved with the development of a script in the Cetoni software.

The many improvements to the experimental setup have paved the way for many interesting experiments to be made. Amongst these are the experiments where the resonance frequency of the setup was determined. The resonance frequency was determined to be 1.898 MHz in two separate experiments as seen in both figure (7.5) and (7.7). With the determination of the resonance frequency of the setup, an important parameter was characterized.
Further characterization of the setup was made with the estimation of the uncertainty on the key parameter t^*, linking how quickly an acoustic focusing occurred to the acoustic energy density, E_{ac}. As described in section (7.2) it was discovered that the relative uncertainty on t^* was independent of the frequency, and thus the relative uncertainty on t^* was determined to be 6 %.

Another objective of this project became to test the hypothesis that E_{ac} is proportional to the square of the driving voltage, U_{pp}, implying that $1/t^* \propto U_{pp}^2$. It was in section (7.4) verified that this hypothesized proportionality does indeed hold, however fully understanding why a large difference in the proportionality constant was found in the two experiments would require further studies.

A final experiment showing the potential of the setup in its current iteration was made. From the experiment described in section (7.5), such a large quantity of measurements of t^* with a high degree of accuracy and precision, were made that it allowed one to see these values being normally distributed. If more time had been available for the project this data could have been utilized in an attempt to sort the algae, based on the value of t^* a given algae sample gave rise to.

The prospects are many and the ambitions are high for the future of this setup. The next step could involve going in various directions.
A next step could be to characterize the resonance frequency along the entire length of the capillary tube allowing one to find the ideal position to acquire video data. Another logical next step could be to try and make the video analysis computationally light enough so that it could be made while an experiment was running. This would allow for the sorting of algae based on the t^* value of a sample, during an experiment. For the future finding ways to improve signal strength would also be beneficial, as it reduces how susceptible the system is to noise. One could also imagine different ways of detecting the particle movement during acoustic focusing, such methods could be particle tracking. The benefits of such methods could be studied and one could see if they presented improvements over the currently utilized technique.

10 References

[1] Henrik Bruus. *Theoretical microfluidics*. Oxford Master Series in Physics. Oxford University Press, 2007. ISBN: 0191528587.

[2] Henrik Bruus. "Acoustofluidics 1: Governing equations in microfluidics". In: *Lab Chip* 11 (22 2011), pp. 3742–3751. DOI: 10.1039/C1LC20658C. URL: http://dx.doi.org/10.1039/C1LC20658C.

[3] Per Augustsson; Rune Barnkob; Steven T. Wereley; Henrik Bruus and Thomas Laurell. "Automated and temperature-controlled micro-PIV measurements enabling long-term-stable microchannel acoustophoresis characterization". In: *Lab Chip* 11.4152 (2011). DOI: 10.1039/c1lc20637k.

[4] Rune Barnkob et al. "Measuring acoustic energy density in microchannel acoustophoresis using a simple and rapid light-intensity method". In: *Lab Chip* 12 (13 2012), pp. 2337–2344. DOI: 10.1039/C2LC40120G.

[5] Henrik Bruus. "Acoustofluidics 7: The acoustic radiation force on small particles". In: *Lab Chip* 12 (6 2012), pp. 1014–1021. DOI: 10.1039/C2LC21068A. URL: http://dx.doi.org/10.1039/C2LC21068A.

[6] Peter Barkholt Muller and Henrik Bruus. "Numerical study of thermoviscous effects in ultrasound-induced acoustic streaming in microchannels". In: *Phys. Rev. E* 90 (4 Oct. 2014), p. 043016. DOI: 10.1103/PhysRevE.90.043016. URL: https://link.aps.org/doi/10.1103/PhysRevE.90.043016.

[7] Peter Barkholt Muller* and Henrik Bruus †. "Numerical study of thermoviscous effects in ultrasound-induced acoustic streaming in microchannels". In: *PHYSICAL REVIEW E* 90.Issue 4 (2014), p. 043016. DOI: 10.1103/PhysRevE.90.043016.

[8] Hector De la Hoz Siegler. "Chapter 22 - Process intensification for sustainable algal fuels production". In: *Handbook of Algal Biofuels*. Ed. by Mostafa El-Sheekh and Abd El-Fatah Abomohra. Elsevier, 2022, pp. 503–521. ISBN: 978-0-12-823764-9. DOI: https://doi.org/10.1016/B978-0-12-823764-9.00011-X. URL: https://www.sciencedirect.com/science/article/pii/B978012823764900011X.

[9] WENNA HU. *Dry Weight and Cell Density of Individual Algal and Cyanobacterial Cells for Algae*. URL: https://mospace.umsystem.edu/xmlui/handle/10355/46477#:~:text=It%20is%20obvious%20that%20the%20dry%20weight%20of,g%2Fcm%20%5Bsuperscript%203%5D%20for%20algae%20and%20cyanobacteria%2C%20respectively..

[10] AZO materials. *Silicone Rubber*. URL: https://www.azom.com/article.aspx?ArticleID=920.

[11] *Perfluoroalkoxy alkane*. URL: https://en.wikipedia.org/wiki/Perfluoroalkoxy_alkane.

A Sedimentation

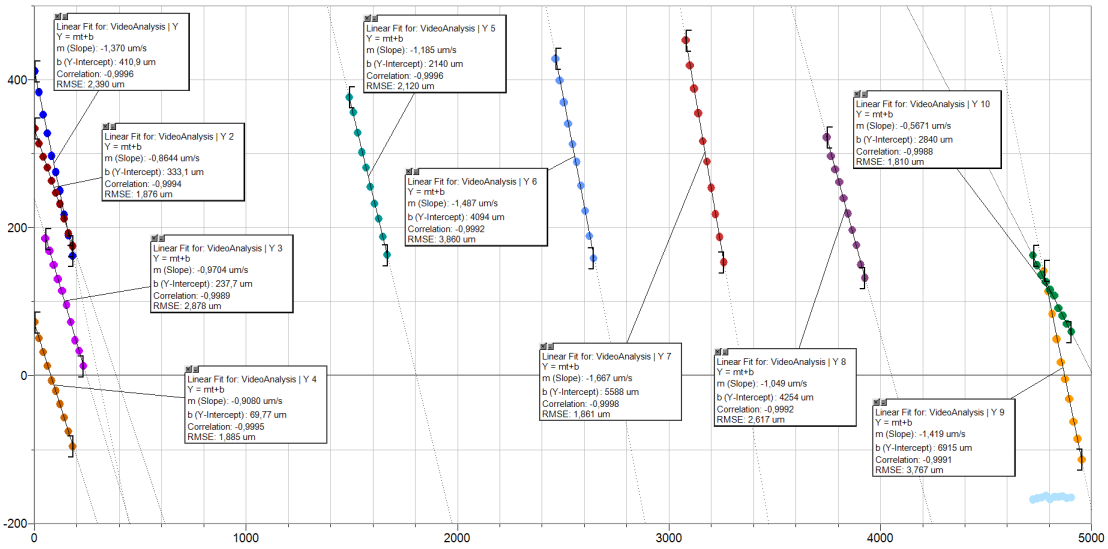


Figure A.1: The figure shows the sedimentation time for algae particles at different times over a period of 1:25:35 hours.

B Pre-experimental routine

Before any experiments were conducted at set of start-up procedures were carried out to ensure that every part of the setup behaved as intended. Unless otherwise specified gloves were worn when handling everything on the Thorlabs NEXUS-table.

1. **Turning on the experiment:**
   The experiment is a mentioned in section(5.1) a collection of multiple components and forgetting to initialize one of these could potentially ruin the experiment. Here is and exhaustive list of step to ensure correct start up.

   1.1 Turn on the power for the entire setup

   1.2 Turn on Cetoni base

   1.3 Turn on Thorlabs base

   1.4 Turn on the three individual Thorbase-cubes

   1.5 Turn on PC

2. **Loading relevant software (*optional)**

   2.1 Run anaconda navigator and activate environment (JoakimPhilip)

   2.2 Run Spyder

   2.3 Run Cetoni Elements

   2.4 Run Thorlabs Kinesis*

   2.5 Run Wave forms*

   2.6 Run Pylon*

3. **Loading algae- and filler-fluid**

   3.1 In Cetoni Elements connect to the setup

   3.2 Unscrew the screw on the syringe for the algae water (Cetoni Nemix)

   3.3 In Cetoni Elements pull back the piston of the syringe

   3.4 Once fully retracted, unscrew by hand the syringe from the holder

   3.5 Over the sink, carefully pull out the piston

   3.6 With a finger over the outlet, pour in the algae water

   3.7 Put the piston back in and push out all the air until a water bobble forms on the outlet of the syringe

   3.8 Dry off with paper towel and screw back into the holder

   3.9 In Cetoni Elements start a high flow and observe the holder closing on the syringe. Once the back of the holder touches the syringe, stop the flow.

   3.10 Screw the screw back in

   3.11 repeat steps 3.2 to 3.10 for the filler (Cetoni NemesysS) with pure water or medium depending on the experiment

4. **System flush.** If the system has been idle for some time, e.g. over night, small air bobbles can form in the tubes and algae can sediment. It is important that this is flushed out with high pressure, but since bobbles and sediment can get stuck in the tube transitions it is also important to flush with low pressure to get creeping flow arounf edges.

4.1 In Cetoni elements make sure both valves for the discard cups are open

4.2 Cut flow to the GCTH

4.3 Set filler-flow to $> 50\,\mu L\,s^{-1}$

4.4 Turn off one of the valves for the discard cups

4.5 Observe when the line to the discard cup has been completely emptied of air then close the valve and open the other.

4.6 Decrease the flow to $< 5\,\mu L\,s^{-1}$

4.7 Repeat steps 4.3 to 4.6

4.8 Cut the flow to the short circuit and open the flow to the GCTH

4.9 Repeat steps 4.3 to 4.7 with 10% of the flow

5. **Homing Thorlabs stage (optional)**. When The Thorlabs stage has been turned off it looses it position and can therefore sometimes be unable to go backwards. In the Thorlabs kinesis software simply click home on all three directions, but let the x-direction run for approximately 10 seconds before starting the others so the camera does not hit the GCTH.

6. **Checking the connection to transducer (optional)**. The transducer is connected to the wave-generator via a clip that can easily loose connection.

6.1 Start a video feed either with a python scrips or through the pylon software.

6.2 If Thorlabs stage is not in position move it to a position.

6.3 Cut the short circuit

6.4 In Cetoni elements start a algae water flow of $10\,\mu L\,s^{-1}$.

6.5 Stop the flow once the channel is filled with algae

6.6 Open the short circuit and cut the GCTH

6.7 Start a signal at resonance frequency with gain at 20 dB

6.8 The connection check is positive if the algae focus

6.9 If the algae do not focus, gently move the clip op the transducer and ones the focus happens leave it in the position and the connection check is done.

6.10 Remember to turn of signal generator.

C Uncertainty table

Frequency (MHz)	δt^*	$(\delta t^*/\bar{t^*})$	$\delta 1/t^*$	$(\delta 1/t^*)/(1/t^*)$
1.885	0.7 s	0.06	$0.006\,\mathrm{s}^{-1}$	0.06
1.8875	0.6 s	0.1	$0.02\,\mathrm{s}^{-1}$	0.1
1.89	0.1 s	0.04	$0.006\,\mathrm{s}^{-1}$	0.04
1.899	0.1 s	0.06	$0.05\,\mathrm{s}^{-1}$	0.06
1.8995	0.1 s	0.07	$0.05\,\mathrm{s}^{-1}$	0.07
1.9	0.1 s	0.06	$0.04\,\mathrm{s}^{-1}$	0.06
1.9005	0.1 s	0.06	$0.03\,\mathrm{s}^{-1}$	0.06
1.901	0.1 s	0.05	$0.02\,\mathrm{s}^{-1}$	0.05
1.905	0.8 s	0.05	$0.003\,\mathrm{s}^{-1}$	0.05
1.9075	0.5 s	0.05	$0.006\,\mathrm{s}^{-1}$	0.05
1.91	0.3 s	0.04	$0.005\,\mathrm{s}^{-1}$	0.04

Table 2: The table shows absolute and relative uncertainties on t^* and $1/t^*$ at different frequencies. 20 measurements were made in order to estimate each uncertainty.

D Impedance sweep

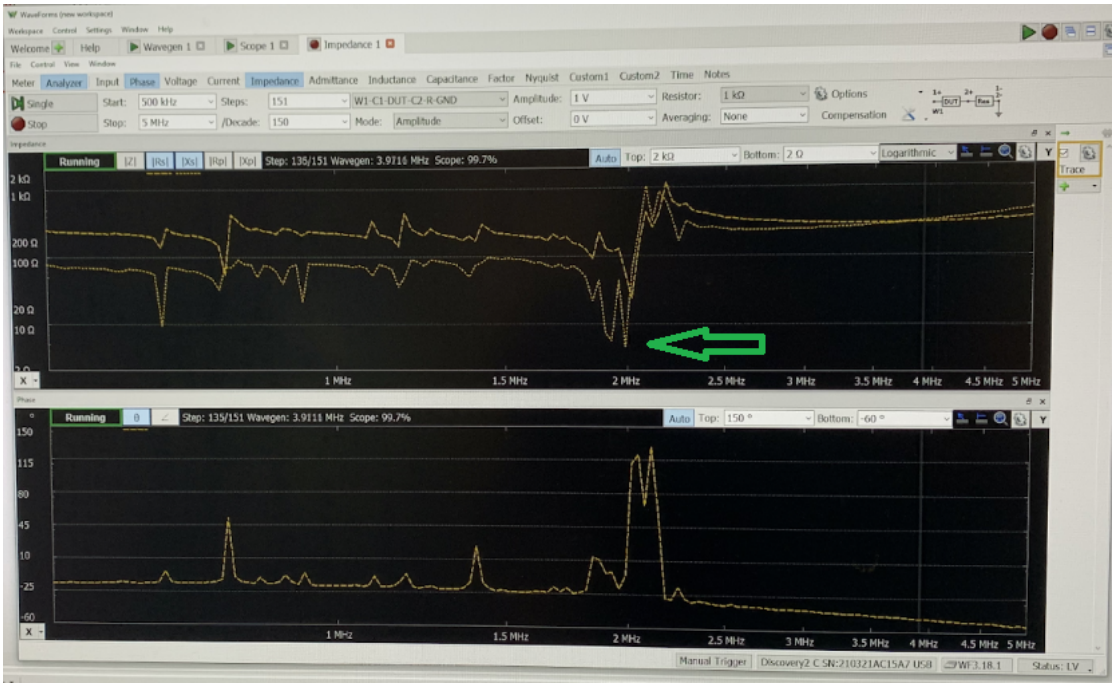


Figure D.1: Impedance sweep made in Wave forms. Shows the the resonance at around 2 MHz

E Scripts

E.1 Compliance

Untitled-1

```python
# %%
import numpy as np
import matplotlib.pyplot as plt

# %%
# calculate the compliance per length of silicone and plastic tubes

#Silicone

#Material properties. Found on https://www.azom.com/properties.aspx?ArticleID=920
E_s = np.linspace(0.001,0.05,100)*1E9 #young's modulus
E_s = E_s[0]
r_s = 0.5E-3 #radius
t_s = 1E-3 #thickness

#compliance per length
compliance_per_length_silicone = (2* np.pi * r_s**3) / (E_s*t_s)
print("The compliance per length of silicone is: ",compliance_per_length_silicone)

#Plastic

#Material properties. Found on https://en.wikipedia.org/wiki/Perfluoroalkoxy_alkane
E_p = 586E9 #young's modulus
r_p = 0.76/2 * 1E-3 #radius
t_p = (1.59 - 0.76)/2 * 1E-3 #thickness

#compliance per length
compliance_per_length_plastic = (2* np.pi * r_p**3) / (E_p*t_p)
print("The compliance per length of plastic is: ",compliance_per_length_plastic)




# %%
#calculate the compliance of silicon

#Length of the silicon tubes
L = 0.2

C = compliance_per_length_silicone * L
print("The compliance of silicon tube of", L, "m is: ",C)

bruus = 1
if bruus:
    # calculation based on the actual method for calculating the compliance:

    nu = 0.48
    delta = t_s/r_s
    top = 2 * np.pi * (1 + nu) * ((1 - 2 * nu) + (1 + delta)**2) * r_s**2 * L
    bottom = (((1 + delta)**2 - 1)*E_s)
    C = top / bottom
print("The compliance of silicon tube of", L, "m is: ",C)

# %%
#calculate the resistance

#Resistance of the silicon tubes
```

```
R_silicone = 1E-3 * (8 / np.pi) * (L/r_s**4)

R_capillary = 1E-3 * 28.4 * (50E-3/(400**4*1E-24)) #Resistance of the capillary tubes 50 mm in length with side length 400 um

# Format the number in scientific notation

print("{:.2e}".format(R_capillary))
print("{:.2e}".format(R_silicone))

# %%
#Calculate the taus
tau1 = R_silicone * C
tau2 = R_capillary * C

print("The tau of the silicone tube is: ", tau1)
print("The tau of the capillary tube is: ", tau2)

# %%
#defining the pressure function
def Compliance_pressure(t):
    return p_star + delta_p*(tau2/(tau1 + tau2))*np.exp(-(1 / tau1 + 1 / tau2)*t)

# %%
#defining the parameters
t = np.linspace(0,00.1,10000)
delta_p = 10
p_star = 0

# %%
#create subplot
fig, ax = plt.subplots()
ax.plot(t,Compliance_pressure(t))
```

E.2 Surviving experiment

```
'''
A simple Program for grabing video from basler camera and converting it to oper
Tested on Basler acA1300-200uc (USB3, linux 64bit , python 3.5)
'''
import cv2
from pypylon import pylon
import serial, time


print('Press ESC to close the window')

arduinoSer = serial.Serial('COM3',baudrate=9600,timeout=0.5)
flow_on = 0;

# conecting to the first available camera
camera = pylon.InstantCamera(pylon.TlFactory.GetInstance().CreateFirstDevice())

# Grabing Continusely (video) with minimal delay
camera.StartGrabbing(pylon.GrabStrategy_LatestImageOnly)
converter = pylon.ImageFormatConverter()

# converting to opencv bgr format
converter.OutputPixelFormat = pylon.PixelType_BGR8packed
converter.OutputBitAlignment = pylon.OutputBitAlignment_MsbAligned

while camera.IsGrabbing():
    grabResult = camera.RetrieveResult(5000, pylon.TimeoutHandling_ThrowExcepti

    if grabResult.GrabSucceeded():
        # Access the image data
        image = converter.Convert(grabResult)
        img = image.GetArray()
        cv2.namedWindow('title', cv2.WINDOW_NORMAL)
        cv2.imshow('title', img)
        k = cv2.waitKey(1)
        # print(k)
        if k == 27: # press ESC
            break
        if k == 32: # press spacebar
            if flow_on==0:
                arduinoSer.write(b'1')
                flow_on = 1
            else:
                arduinoSer.write(b'0')
                flow_on = 0
        if k == 97: # press a (alive)
            arduinoSer.write(b'3') # on
        if k == 100: # press d (dead)
            arduinoSer.write(b'2') # off



    grabResult.Release()

# Releasing the resource
```

```
camera.StopGrabbing()

arduinoSer.write(b'0')

cv2.destroyAllWindows()
arduinoSer.close()
```

E.3 AD function

AD_func.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Aug 12 17:35:47 2023

@author: joakimpihl
"""

##########
# This script contains various functions related to




import sys
sys.path.append('C:/Program Files (x86)/Digilent/WaveFormsSDK/samples/py')
#sys.path.append('/Applications/WaveForms.app/Contents/Resources/SDK/samples/py')
from ctypes import *
from dwfconstants import *
import numpy as np



### Function for connecting to the analog discovery

def Connect():
    """
    This function connects to the Analog discovery.

    Returns
    -------
    None.

    """
    #Defines global variable (dwf, hdwf) which can be used outside the function
    global dwf
    global hdwf
    global channel
    # Checks if the system is windows (win) or Mac os (darwin) and then find path to dwf: /Library/Frameworks/dwf.framework/dwf
    if sys.platform.startswith("win"):
        dwf = cdll.dwf
    elif sys.platform.startswith("darwin"):
        dwf = cdll.LoadLibrary("/Applications/WaveForms.app/Contents/Frameworks/dwf.framework/dwf")

    hdwf = c_int()
    channel = c_int(0)

    # Prevent temperature drift - Specifies Analog discovery behavior upon closure of the program
    dwf.FDwfParamSet(DwfParamOnClose, c_int(0)) # 0 = run, 1 = stop, 2 = shutdown

    #Open device
    print("Opening device...")
    dwf.FDwfDeviceOpen(c_int(-1), byref(hdwf))

```

```
    return



### Functino for disconnecting the analog discovery
def disconnect():
    dwf.FDwfDeviceCloseAll()

    return



### Function for generating a sound siganl

def funcGen(shape=funcSine,freq=1.91,Amplitude=1,v_Offset=0):
    """


    Parameters
    ----------
    shape : TYPE, optional
        Determines the output signal shape. The default is funcSine.
    freq : TYPE, optional
        Frequency of the signal in MHz. The default is 1.91 MHz.
    Amplitude : TYPE, optional
        Voltage amplitude of the signal. The default is 1V.
    v_Offset : TYPE, optional
        Voltage offset of the signal. The default is 0V.

    Returns
    -------
    None.

    """

    #if hdwf.value == hdwfNone.value:
    #    print("failed to open device")
    #    quit()

    # 0 = the device will be configured only when calling FDwf###Configure - One can for instance just change the freq and everything else will automatically be configured
    dwf.FDwfDeviceAutoConfigureSet(hdwf, c_int(0))

    dwf.FDwfAnalogOutNodeEnableSet(hdwf, channel, AnalogOutNodeCarrier, c_int(1))
    dwf.FDwfAnalogOutNodeFunctionSet(hdwf, channel, AnalogOutNodeCarrier, shape) # Sets the output function
    dwf.FDwfAnalogOutNodeFrequencySet(hdwf, channel, AnalogOutNodeCarrier, c_double(freq*10**6)) # Sets the frequency in Hz i.e. 1000 = 1kHz
    dwf.FDwfAnalogOutNodeAmplitudeSet(hdwf, channel, AnalogOutNodeCarrier, c_double(Amplitude)) # Sets the signal amplitude in volts i.e. 1 = 1V
    dwf.FDwfAnalogOutNodeOffsetSet(hdwf, channel, AnalogOutNodeCarrier, c_double(v_Offset)) # Sets the voltage offset

    print("Generating sine wave at " + str(round(freq,3)) + "MHz")
    dwf.FDwfAnalogOutConfigure(hdwf, channel, c_int(1)) #This func configures/starts the device with the specified configuration.

    return



```

```
###  Function for stopping the sound signal

def funcStop():
    """
    This function stops any signal generated by the Analog discovery 2

    Returns
    -------
    None.

    """
    print("Stopping sine wave...")
    dwf.FDwfAnalogOutConfigure(hdwf, channel, c_int(0)) # Stops the sine function
    return

######################################################################


#Figure out which frequency sweep to utilize and the clever image analysis


######################################################################

### Function for doing a frequency

def sweep(HzStart, HzStop, shape=funcSine, Amplitude=1, v_Offset=0, HzTime=5):

    secSweep = (HzStop-HzStart)/HzTime



    dwf.FDwfAnalogOutNodeEnableSet(hdwf, channel, AnalogOutNodeCarrier, c_int(1))
    dwf.FDwfAnalogOutNodeFunctionSet(hdwf, channel, AnalogOutNodeCarrier, shape)
    dwf.FDwfAnalogOutNodeFrequencySet(hdwf, channel, AnalogOutNodeCarrier,
c_double(HzStop))
    dwf.FDwfAnalogOutNodeAmplitudeSet(hdwf, channel, AnalogOutNodeCarrier,
c_double(Amplitude))
    dwf.FDwfAnalogOutNodeOffsetSet(hdwf, channel, AnalogOutNodeCarrier, c_double(v_Offset)
)

    dwf.FDwfAnalogOutRunSet(hdwf, channel, c_double(secSweep)) # Determines for how long
it stays at each frequency in seconds
    dwf.FDwfAnalogOutRepeatSet(hdwf, channel, c_int(1)) #Sets the repeat count

    return


### Function for generating a frequency sweep

def freqSweep(shape=funcSine,start=1.89,stop=1.93,by=2,Amplitude=1,v_Offset=0):
    # 0 = the device will be configured only when calling FDwf###Configure - One can for
instance just change the freq and everything else will automatically be configured
    import time
    dwf.FDwfDeviceAutoConfigureSet(hdwf, c_int(0))

    dwf.FDwfAnalogOutNodeEnableSet(hdwf, channel, AnalogOutNodeCarrier, c_int(1))
    dwf.FDwfAnalogOutNodeFunctionSet(hdwf, channel, AnalogOutNodeCarrier, shape) # Sets
the output function
    dwf.FDwfAnalogOutNodeAmplitudeSet(hdwf, channel, AnalogOutNodeCarrier,
c_double(Amplitude)) # Sets the signal amplitude in volts i.e. 1 = 1V
    dwf.FDwfAnalogOutNodeOffsetSet(hdwf, channel, AnalogOutNodeCarrier, c_double(v_Offset)
) # Sets the voltage offset

```

```
    frequencies = np.arange(start*10**6, stop*10**6+by*10**3, by*10**3)   # start and stop frequencies in MHz, the step frequency is in kHz

    for i in frequencies:
        dwf.FDwfAnalogOutNodeFrequencySet(hdwf, channel, AnalogOutNodeCarrier, c_double(i*10**6)) # Sets the frequency in Hz i.e. 1000 = 1kHz
        dwf.FDwfAnalogOutConfigure(hdwf, channel, c_int(1)) #This func configures/starts the device with the specified configuration.
        ########################################

        #Add in some clever image analysis here

        ########################################
        print('Analysis at' + str(i) + 'MHz is complete. The result is:' + 'Add the result is' + '. Going on to the next frequency...')
        time.sleep(5) #Generates frequency for 5s and then it goes on to the next frequency



    print("Generating frequency sweep starting at " + str(start) + "MHz" + " and stopping at" + str(stop) + "MHz." + "The frequency step size is: " + str(by))

    return
```

E.4 Gauss distribution calculation

Untitled-1

```python
# %%
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import csv
from scipy.stats import linregress
from decimal import Decimal
from matplotlib.ticker import PercentFormatter

# %%
# import csv-file
df = pd.read_csv(r'C:\Users\Phili\OneDrive - Danmarks Tekniske Universitet\Bachelorprojekt\Videoer\High concentration\Gauss on same frequency\Resonance-freq.csv', delimiter=',', encoding='utf-8')
t_star = df.iloc[:,2]
x_values = np.arange(0, len(t_star))

# %%
# make a gaussian
mean = np.mean(t_star)
std = np.std(t_star)
gauss = np.random.normal(mean, std, len(t_star))

# set x range
width = 0.45
xrange = [mean - width/2, mean + width/2]

# %%
# remove n data points
n = 75
x_values2 = x_values[n:]
t_star2 = t_star[n:]

# split data into two arrays
m = 200
x_values2_1 = x_values2[:m]
t_star2_1 = t_star2[:m]

x_values2_2 = x_values2[m+6:]
t_star2_2 = t_star2[m+6:]

# make linear regression
result2_1 = linregress(x_values2_1, t_star2_1)
slope2_1 = result2_1.slope
intercept = result2_1.intercept
y_hat2_1 = slope2_1 * x_values2_1 + intercept

result2_2 = linregress(x_values2_2, t_star2_2)
slope2_2 = result2_2.slope
intercept = result2_2.intercept
y_hat2_2 = slope2_2 * x_values2_2 + intercept

print(slope2_1)
print(slope2_2)

# %%
# create side by side sub plots
```

```
fig, ax = plt.subplots(1, 2, figsize=(10, 5))

## plot left side subplot
# Plot the histogram
ax[0].hist(t_star, bins=int(np.floor(np.sqrt(len(t_star)))), density=True, alpha=0.6, color='b')

# Plot the probability density function (PDF) of the Gaussian distribution
x = np.linspace(mean - 4 * std, mean + 4 * std, 100)
pdf = (1 / (std * np.sqrt(2 * np.pi))) * np.exp(-(x - mean)**2 / (2 * std**2))
ax[0].plot(x, pdf, color='tab:red')
textstr = '\n'.join((r"Frequency=1.898 MHz", r"$\sigma = $" + f"{Decimal(str(std)):.1E}", r"$\mu = $" + f"{Decimal(str(mean)):.1E}"))
ax[0].text(0.03, 0.97, textstr, transform=ax[0].transAxes, fontsize=9.4,
        verticalalignment='top', bbox=dict(boxstyle='round', facecolor='white', alpha=0.2)
)

# Add labels and title
ax[0].set_xlabel(r'(a) $t^{\mathrm{*}}$ [s]')
ax[0].set_ylabel('Probability')
ax[0].vlines(x=mean, ymin=0, ymax=6.785, color='black')
ax[0].set_xlim(xrange[0], xrange[1])
ax[0].yaxis.set_major_formatter(PercentFormatter())

## plot right side subplot

# plot the scatter plot
ax[1].scatter(x_values[:n], t_star[:n], color = "red",edgecolors='black',linewidths=0.5)
ax[1].scatter(x_values[n:n+m], t_star[n:n+m], color = "tab:blue")
ax[1].scatter(x_values[n+m+6:], t_star[n+m+6:], color = "tab:blue")
ax[1].scatter(x_values[n+m:n+m+6], t_star[n+m:n+m+6], color = "red",edgecolors='black', linewidths=0.5)
ax[1].twiny().plot([0,3.48], [0.8,0.8], color="black", alpha=0)

ax[1].plot(x_values2_1, y_hat2_1, color="magenta")
ax[1].plot(x_values2_2, y_hat2_2, color="lime")

# Add labels and title
ax[1].set_xlabel('(b) Runs')
ax[1].set_ylabel(r'$t^{\mathrm{*}}$ [s]')
ax[1].yaxis.set_label_position("right")
ax[1].yaxis.tick_right()
ax[1].legend(['Excluded data', 'Used data', "_nolegend_", "_nolegend_", "Slope = " + f"{Decimal(str(slope2_1)):.1E}", "Slope = " + f"{Decimal(str(slope2_2)):.1E}"])

ax[1].text(1.02, 1.06, "Time [h]", transform=ax[1].transAxes, fontsize=9.5,
        verticalalignment='top', bbox=dict(boxstyle='round', facecolor='white', alpha=0))
# save the plot
plt.savefig(r'C:\Users\Phili\OneDrive - Danmarks Tekniske Universitet\Bachelorprojekt\Images for Overleaf\Gauss-distribution.png', dpi=350, bbox_inches = "tight")


# %%
# merge to arrays
x_values = np.concatenate((x_values2_1, x_values2_2))
t_star = np.concatenate((t_star2_1 - y_hat2_1, t_star2_2 - y_hat2_2)) + np.mean(t_star2[:5])

# %%
# make a gaussian
mean = np.mean(t_star)
```

```
std = np.std(t_star)
gauss = np.random.normal(mean, std, len(t_star))

# make x range
xrange = [mean - width/2, mean + width/2]

# %%
# create side by side sub plots
fig, ax = plt.subplots(1, 2, figsize=(10, 5))

## plot left side subplot
# Plot the histogram
ax[0].hist(t_star, bins=int(np.floor(np.sqrt(len(t_star)))), density=True, alpha=0.6, color='b')

# Plot the probability density function (PDF) of the Gaussian distribution
x = np.linspace(mean - 4 * std, mean + 4 * std, 100)
pdf = (1 / (std * np.sqrt(2 * np.pi))) * np.exp(-(x - mean)**2 / (2 * std**2))
ax[0].plot(x, pdf, color='tab:red')
ax[0].fill_between(x, 0, pdf, where=x < mean - 2 * std, color='red', alpha=0.7)
ax[0].fill_between(x, 0, pdf, where=x > mean + 2 * std, color='lime', alpha=0.7)

# create the jazz
textstr = '\n'.join((r"Frequency=1.898 MHz", r"$\sigma = $" + f"{Decimal(str(std)):.1E}", r"$\mu = $" + f"{Decimal(str(mean)):.1E}"))
ax[0].text(0.03, 0.97, textstr, transform=ax[0].transAxes, fontsize=10,
        verticalalignment='top', bbox=dict(boxstyle='round', facecolor='white', alpha=0.2)
)
textstr = "Fastest algae"
ax[0].text(0.03, 0.37, textstr, transform=ax[0].transAxes, fontsize=10,
        verticalalignment='top', bbox=dict(boxstyle='round', facecolor='white', alpha=0.2)
)
ax[0].annotate("", xy=(0.7, 1.6), xytext=(0.65, 4.3),
            arrowprops=dict(arrowstyle="->"))
textstr = "Slowest algae"
ax[0].text(0.7, 0.37, textstr, transform=ax[0].transAxes, fontsize=10,
        verticalalignment='top', bbox=dict(boxstyle='round', facecolor='white', alpha=0.2)
)
ax[0].annotate("", xy=(0.875, 1), xytext=(0.93, 4.3),
            arrowprops=dict(arrowstyle="->"))

# Add labels and title
ax[0].set_xlabel(r'(a) $t^{\mathrm{*}}$ [s]')
ax[0].set_ylabel('Probability')
ax[0].vlines(x=mean, ymin=0, ymax=11.7, color='black')
ax[0].set_xlim(xrange[0], xrange[1])
ax[0].yaxis.set_major_formatter(PercentFormatter())

## plot right side subplot

# plot the scatter plot
ax[1].scatter(x_values, t_star, color="tab:blue")
ax[1].twiny().plot([0,3.48], [0.8,0.8], color="black", alpha=0)

# Add labels and title
ax[1].set_xlabel('(b) Runs')
ax[1].set_ylabel(r'$t^{\mathrm{*}}$ [s]')
ax[1].yaxis.set_label_position("right")
ax[1].yaxis.tick_right()
ax[1].hlines(y=mean, xmin=x_values[0], xmax=x_values[-1], color='black')
ax[1].hlines(y=mean - 2 * std, xmin=x_values[0], xmax=x_values[-1], color='red', alpha=
```

```
0.7)
ax[1].hlines(y=mean + 2 * std, xmin=x_values[0], xmax=x_values[-1], color='lime', alpha=0.7)
ax[1].fill_between(x_values, 0.685, mean - 2 * std, color='red', alpha=0.7)
ax[1].fill_between(x_values, mean + 2 * std, 0.875, color='lime', alpha=0.7)
ax[1].legend(['Used data', r"$\mu$", r"Fastest algae", r"Slowest algae"])
ax[1].set_xlim(80, 390)
ax[1].set_ylim(0.685, 0.875)
ax[1].text(1.02, 1.06, "Time [h]", transform=ax[1].transAxes, fontsize=9.5,
        verticalalignment='top', bbox=dict(boxstyle='round', facecolor='white', alpha=0))
# save the plot
plt.savefig(r'C:\Users\Phili\OneDrive - Danmarks Tekniske Universitet\Bachelorprojekt\Images for Overleaf\Gauss-distribution-corrected.png', dpi=350,bbox_inches = "tight")

# %%
# calculate difference in t_star for fits
first_t_star = y_hat2_1[0]
last_t_star = y_hat2_1[-1]

oneover_first = 1/first_t_star
oneover_last = 1/last_t_star

print(oneover_first)
print(oneover_last)

# calculate decrease

diff_oneover =  oneover_last - oneover_first

print(diff_oneover)

# %%
#time duration
first_time = 17044486625899300E-7
last_time = 17044612068070100E-7

diff_time = last_time - first_time
print(diff_time/3600)

# %%
```

E.5 Pressure due to compliance calculation

Untitled-1

```python
# %%
import numpy as np
import matplotlib.pyplot as plt

# %%
# calculate the compliance per length of silicone and plastic tubes

#Silicone

#Material properties. Found on https://www.azom.com/properties.aspx?ArticleID=920
E_s = np.linspace(0.001,0.05,100)*1E9 #young's modulus
E_s = E_s[0]
r_s = 0.5E-3 #radius
t_s = 1E-3 #thickness

#compliance per length
compliance_per_length_silicone = (2* np.pi * r_s**3) / (E_s*t_s)
print("The compliance per length of silicone is: ",compliance_per_length_silicone)

#Plastic

#Material properties. Found on https://en.wikipedia.org/wiki/Perfluoroalkoxy_alkane
E_p = 586E9 #young's modulus
r_p = 0.76/2 * 1E-3 #radius
t_p = (1.59 - 0.76)/2 * 1E-3 #thickness

#compliance per length
compliance_per_length_plastic = (2* np.pi * r_p**3) / (E_p*t_p)
print("The compliance per length of plastic is: ",compliance_per_length_plastic)




# %%
#calculate the compliance of silicon

#Length of the silicon tubes
L = 0.2

C = compliance_per_length_silicone * L
print("The compliance of silicon tube of", L, "m is: ",C)

bruus = 1
if bruus:
    # calculation based on the actual method for calculating the compliance:

    nu = 0.48
    delta = t_s/r_s
    top = 2 * np.pi * (1 + nu) * ((1 - 2 * nu) + (1 + delta)**2) * r_s**2 * L
    bottom = (((1 + delta)**2 - 1)*E_s)
    C = top / bottom
print("The compliance of silicon tube of", L, "m is: ",C)

# %%
#calculate the resistance

#Resistance of the silicon tubes
```

```
R_silicone = 1E-3 * (8 / np.pi) * (L/r_s**4)

R_capillary = 1E-3 * 28.4 * (50E-3/(400**4*1E-24)) #Resistance of the capillary tubes 50 mm in length with side length 400 um

# Format the number in scientific notation

print("{:.2e}".format(R_capillary))
print("{:.2e}".format(R_silicone))

# %%
#Calculate the taus
tau1 = R_silicone * C
tau2 = R_capillary * C

print("The tau of the silicone tube is: ", tau1)
print("The tau of the capillary tube is: ", tau2)

# %%
#defining the pressure function
def Compliance_pressure(t):
    return p_star + delta_p*(tau2/(tau1 + tau2))*np.exp(-(1 / tau1 + 1 / tau2)*t)

# %%
#defining the parameters
t = np.linspace(0,00.1,10000)
delta_p = 10
p_star = 0

# %%
#create subplot
fig, ax = plt.subplots()
ax.plot(t,Compliance_pressure(t))
```

E.6 Sedimentation calculation

Untitled-1

```
# %% [markdown]
# ### Sedimantation velocities

# %%
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from matplotlib.offsetbox import (OffsetImage, AnnotationBbox)

# %%
xs = np.array([0, 140, 60, 90, 190, 25, 45, 90, 15, 170])
vs = np.array([1.370, 0.8644, 0.9704, 0.9080, 1.185, 1.487, 1.667, 1.049, 1.419, 0.5671])
mean_v = np.mean(vs)
sd_v = np.std(vs)

# %%
plt.scatter(xs, vs, color="black")

plt.xlabel(r"Distance from center ($\mathrm{\mu m}$)  ")
plt.ylabel(r"v ($\mathrm{\mu m \,s^{-1}}$)")

plt.hlines(y=mean_v, xmin=np.min(xs)-10, xmax=np.max(xs)+10)
plt.hlines(y=mean_v + sd_v, xmin=np.min(xs)-10, xmax=np.max(xs)+10, color= "tab:red", linestyle = "dashed")
plt.hlines(y=mean_v - sd_v, xmin=np.min(xs)-10, xmax=np.max(xs)+10, color= "tab:red", linestyle = "dashed")
plt.hlines(y=mean_v - sd_v + 0.02, xmin=100, xmax=110, color= "black")
plt.hlines(y=mean_v - 0.02, xmin=100, xmax=110, color= "black")

plt.hlines(y=mean_v + sd_v - 0.02, xmin=100, xmax=110, color= "black")
plt.hlines(y=mean_v + 0.02, xmin=100, xmax=110, color= "black")
plt.vlines(x=105, ymin=mean_v + 0.02, ymax=mean_v + sd_v -0.02, color="black")
plt.vlines(x=105, ymin=mean_v - sd_v +0.02, ymax=mean_v - 0.02, color="black")

plt.text(-8, 1.05, s="Mean velocity =" + str(round(mean_v,2)) + " " + r"$\mathrm{\mu m \, s^{-1}}$", color="tab:blue")
plt.text(108, 1.27, s=r"$\mathrm{\sigma}$", color="black")
plt.text(108, 0.96, s=r"$\mathrm{\sigma}$", color="black")

plt.xlim(np.min(xs)-10, np.max(xs)+10)
plt.ylim(mean_v - 2*sd_v, mean_v + 2*sd_v)


plt.savefig(r'C:\Users\Phili\OneDrive - Danmarks Tekniske Universitet\Bachelorprojekt\Images for Overleaf\sedimentation velocities.png', dpi=350, bbox_inches = "tight")

# %%
#theoretical value
m = 1.54E-14
r = (4.33/2)*1E-6
rho = 998
g = 9.82
eta = 1E-3

v = ((m - rho * 4/3 * np.pi * r**3) * g) / (6 * np.pi * eta * r)
print(v)

```

```
# %%


# %%
```

E.7 Temperature dependence calculation

Untitled-1

```python
# %%
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
from decimal import Decimal

# %%
# define the cs coeficients
c0 = -3.1420310908E4
c1 =  4.4207693183E2
c2 =  -2.4555531077
c3 =  7.0411177557E-3
c4 =  -1.0353739103E-5
c5 =  6.1949139692E-9

# %%
# define the speed function in terms of temperature
def Speed(T):
    return c0 + c1*T + c2*T**2 + c3*T**3 + c4*T**4 + c5*T**5

T = np.linspace(283, 323, 10000)
Ti = 273 + 22
Tf = Ti + 1

# define the frequency function
def Frequency(v):
    return v / lambda_

res_freq = 1.8975 #value taken from the thesis
v = np.linspace(Speed(Ti) * 0.99, Speed(Tf) * 1.01, 10000)
lambda_ = Speed(Ti)/res_freq
print(lambda_*1E-6)
print(Speed(Ti))

# %%
# crete sub plot
fig, ax = plt.subplots(1, 2, figsize=(10, 5))
ax[0].plot(T, Speed(T))
ax[0].vlines(x=Ti, ymin=Speed(T[0]), ymax=Speed(Ti), color="tab:green", linestyles="dashed")
ax[0].vlines(x=Tf, ymin=Speed(T[0]), ymax=Speed(Tf), color="tab:green", linestyles="dashed")
ax[0].hlines(y=Speed(Ti), xmin=T[0], xmax=Ti, color="tab:green", linestyles="dashed")
ax[0].hlines(y=Speed(Tf), xmin=T[0], xmax=Tf, color="tab:green", linestyles="dashed")
ax[0].set_xlim(293,296.5)
ax[0].set_ylim(1480,1500)
ax[0].hlines(y=1482.5, xmin=Ti, xmax=Tf, color="black", linestyles="solid")
ax[0].text(295.2, 1482.7, "1 Kelvin", fontsize=10)
ax[0].set_xlabel(r'(a) Temperature [$\mathrm{K}$]')
ax[0].set_ylabel(r'Speed [$\mathrm{m/s}$]')
ax[0].legend([r"$c_0(T)=P_5(T)$", "_", "_"])

# plot the boxes
ax[0].text(0.1, 0.7, "Speed difference", transform=ax[0].transAxes, fontsize=10,
        verticalalignment='top', bbox=dict(boxstyle='round', facecolor='white', alpha=0.2))
ax[0].annotate("", xy=(293.5, 1490), xytext=(294, 1492.9),
            arrowprops=dict(arrowstyle="->"))
```

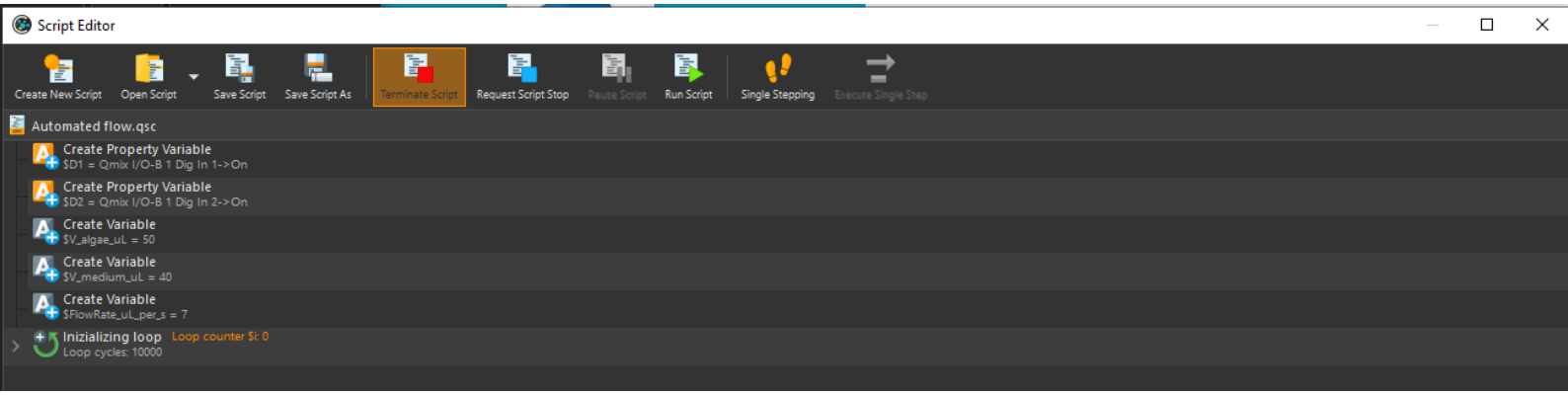
```
56
57  ax[1].plot(Speed(T), Frequency(Speed(T)))
58  ax[1].vlines(x=Speed(Ti), ymin=Frequency(Speed(T))[0], ymax=Frequency(Speed(Ti)), color="
    tab:green", linestyles="dashed")
59  ax[1].vlines(x=Speed(Tf), ymin=Frequency(Speed(T))[0], ymax=Frequency(Speed(Tf)), color="
    tab:green", linestyles="dashed")
60  ax[1].hlines(y=Frequency(Speed(Ti)), xmin=Speed(T)[0], xmax=Speed(Ti), color="tab:green",
    linestyles="dashed")
61  ax[1].hlines(y=Frequency(Speed(Tf)), xmin=Speed(T)[0], xmax=Speed(Tf), color="tab:green",
    linestyles="dashed")
62  ax[1].set_xlim(1480,1492)
63  ax[1].set_ylim(1.88, 1.905)
64  ax[1].vlines(x=1481, ymin=Frequency(Speed(Ti)), ymax=Frequency(Speed(Tf)), color="black",
    linestyles="solid")
65  ax[1].text(1481.2, 1.8985, '\n'.join((r"1 Kelvin", r"$\Delta f =$" + "3.7" + r"
    $\mathrm{kHz}$")), fontsize=10)
66  ax[1].set_xlabel(r'(b) Speed [$\mathrm{m/s}$]')
67  ax[1].set_ylabel(r'Frequency [$\mathrm{MHz}$]')
68  ax[1].yaxis.set_label_position("right")
69  ax[1].yaxis.tick_right()
70  ax[1].legend([r"$f(c_0) = c_0 / \lambda$", "_", "_"],loc="upper right")
71
72  # plot the boxes
73  ax[1].text(0.2, 0.33, "Speed difference", transform=ax[1].transAxes, fontsize=10,
74          verticalalignment='top', bbox=dict(boxstyle='round', facecolor='white', alpha=0.2))
75  ax[1].annotate("", xy=(1488.5, 1.885), xytext=(1484.5, 1.8867),
76              arrowprops=dict(arrowstyle="->"))
77
78  plt.savefig(r'C:\Users\Phili\OneDrive - Danmarks Tekniske
    Universitet\Bachelorprojekt\Images for Overleaf\Temeparature dependence.png', dpi=350,
    bbox_inches = "tight")
79
80
81  # %%
82  # speed difference
83  speed_difference = Speed(Tf) - Speed(Ti)
84  print("Speed difference =" + str(round(speed_difference,2)))
85
86  # %%
87  # increase in frequency
88  round(Frequency(Speed(Tf)),4)
89
90
91
```

E.8 Valve control

Valve_control.py

```python
# -*- coding: utf-8 -*-
"""
Created on Wed Nov 15 12:44:40 2023

@author: s102772
"""

def switchvalve():
    import MXII_valve

    port_name = "COM4"
    #if the port has been cahnges it valve does not connect uncomment next line
    #port_name = None

    #find and connect to alve
    valve_address = MXII_valve.find_address(port_name)[0]
    valve = MXII_valve.MX_valve(valve_address, ports=10, name='My_valve', verbose=True)

    #switch valve position
    port = valve.get_port()
    if port == 1:
        valve.change_port(2)
    else:
        valve.change_port(1)

    del valve

    return

def CheckOpen():
    import MXII_valve

    port_name = "COM4"
    #if the port has been cahnges it valve does not connect uncomment next line
    #port_name = None

    #find and connect to alve
    valve_address = MXII_valve.find_address(port_name)[0]
    valve = MXII_valve.MX_valve(valve_address, ports=10, name='My_valve', verbose=True)

    #switch valve position
    port = valve.get_port()
    if port == 1:
        openport = True
    else:
        openport = False
    del valve

    return openport

```

E.9 Cetoni script


Script Editor
Create New Script
Open Script
Save Script
Save Script As
Terminate Script
Request Script Stop
Pause Script
Run Script
Single Stepping
Execute Single Step
Automated flow.qsc
Create Property Variable
$D1 = Qmix I/O-B 1 Dig In 1->On
Create Property Variable
$D2 = Qmix I/O-B 1 Dig In 2->On
Create Variable
$V_algae_uL = 50
Create Variable
$V_medium_uL = 40
Create Variable
$FlowRate_uL_per_s = 7
Inizializing loop
Loop counter $i: 0
Loop cycles: 10000

E.10 Video analysis script

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Nov 29 14:06:19 2023

@author: joakimpihl
"""
#%%
#Load necessary packages
import cv2, os, sys
import numpy as np
from matplotlib import pyplot as plt
import pandas as pd
from scipy.optimize import curve_fit
sys.path.append('/Users/joakimpihl/Desktop/DTU/7. Semester/Bachelorprojekt/Ultr
# Importing functions from other files
from Moving_Average import *

def func(t, t_star, R): #Define function which data is fitted to
    return 1-R/k*np.arctan(np.tan(k)*np.exp(-t/t_star))
#%%
# Paths to files
#vid_num = 15 #Video 0 through 52
path_out ='/Users/joakimpihl/Desktop/V_sweep_LC/'
for vid_num in range(19,20):
    path_vid = '/Users/joakimpihl/Desktop/DTU/7. Semester/Bachelorprojekt/Exper
    path_imp_times = '/Users/joakimpihl/Desktop/DTU/7. Semester/Bachelorprojekt
    path_times = '/Users/joakimpihl/Desktop/DTU/7. Semester/Bachelorprojekt/Exp

    # Load files
    vid = cv2.VideoCapture(path_vid)
    imp_times = pd.read_csv(path_imp_times, delimiter=';', encoding='utf-8')['I
    timestamps = pd.read_csv(path_times, delimiter=';', encoding='utf-8')['Fram
    freq = pd.read_csv(path_times, delimiter=';', encoding='utf-8')['Frame time

    # Manipulate csv files - Timescaling, and retrieve frequency information
    t_index_0 = None
    t_index_end = None
    for i, ts in enumerate(timestamps):
        if np.isclose(ts, imp_times[1], atol=0.01, rtol=0):
            if t_index_0 is None:
                t_index_0 = i
            else:
                raise ValueError('Error! Multiple starting indices were found.'
        if np.isclose(ts, imp_times[2], atol=0.01, rtol=0):
            if t_index_end is None:
                t_index_end = i
            else:
                raise ValueError('Error! Multiple starting indices were found.'

    if t_index_0 is not None:
        imp_times = np.array(imp_times) - timestamps[t_index_0]
        timestamps = np.array(timestamps) - timestamps[t_index_0]
    else:
        raise ValueError('Error! No starting index found.')
```

```
###################################### Generate intensity matrix ##########
ret, frame = vid.read()#Reads the fist frame
total_frames = int(vid.get(cv2.CAP_PROP_FRAME_COUNT)) #Frames in video

# Scaling information
px_width = np.size(frame, axis=1) #Width of video in px
px_height = np.size(frame, axis=0) #Height of video in px
mm_per_px = 400/px_width # conversion factor: um/px
width = px_width * mm_per_px #um
height = px_height * mm_per_px #um

#if there's a missmatch between the number of frames and the number of time
if len(timestamps) != total_frames:
    raise ValueError('The number of timestamps doens\'t match the number of

#Define empty arrays to contain data
intensities = np.empty([total_frames, px_width])

plt.close('all') #Close all plots for good measure

#Generate matrix containing vertically meaned intensities
for i in range(total_frames):
    vid.set(cv2.CAP_PROP_POS_FRAMES, i)
    print('Analyzing frame number: ' + str(int(vid.get(cv2.CAP_PROP_POS_FRA
    ret, frame = vid.read() #Reads the fist frame
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #Converts the frame to
    frame_mean = np.mean(frame, axis=0)
    intensities[i] = frame_mean

#Remove the first meaned frame as values are always 'weird'
intensities = np.delete(intensities, 0, 0)
timestamps = np.delete(timestamps, 0)

#Define alpha region - The region which algae are focused into
alpha = 0.20
I_norm = np.empty([len(timestamps)]) #normalised intensity - I/I_max

#Define the 2 sections where we measure the intensity change
s1 = np.array([0, int(np.ceil(px_width * (1 - alpha) * 1/2))]) #Index for s
s2 = np.array([int(np.ceil(px_width * (1+alpha) * 1/2)), -1]) #Index for se


#Define max intensity - Max value is defined to be the sum of mean intensit
I_0 = np.sum(intensities[t_index_0,s1[0]:s1[1]]) + np.sum(intensities[t_inc
I_max = np.sum(intensities[-1,s1[0]:s1[1]]) + np.sum(intensities[-1,s2[0]:s
R_data = 1 - I_0/I_max # Relative reduction in intensity
k = np.pi*(1-alpha)/2 #Defining a constant k

#For each timestamp figure out the intensity the intensity relative to I_ma
for i in range(len(timestamps)):
    if i < np.size(intensities,axis=0):
        i_1 = np.sum(intensities[i, s1[0]:s1[1]])
        i_2 = np.sum(intensities[i, s2[0]:s2[1]])

        I_t = np.sum([i_1, i_2])
```

```
        I_norm[i] = I_t/I_max

popt, pcov = curve_fit(func, timestamps, I_norm) #Fitting - Use all data or

#Fitting parameters are defined
t_star = popt[0]
R_fit = popt[1]
delta_y=R_fit/(1-alpha)

residuals = I_norm - func(timestamps, *popt)
ss_res = np.sum(residuals**2)
ss_tot = np.sum((I_norm-np.mean(I_norm))**2)
r_squared = 1 - (ss_res / ss_tot)

#Generate fitting data to overlay experimental data
xdata_time = np.linspace(timestamps[0], timestamps[-1], 1000)
ydata = func(xdata_time, *popt)

plt.close('all')

t_star_e = format(t_star, ".1e")
R_fit_e = format(R_fit, ".1e")

fig, ax = plt.subplots(figsize=(10,7.5))
ax.hlines(y=1-R_fit/(1-alpha), xmin=timestamps[0], xmax=timestamps[-1], col
ax.hlines(y=1, xmin=timestamps[0], xmax=timestamps[-1], color='silver', alp
#ax.vlines(x=imp_times[1], ymin=0.998, ymax=1)
#ax.vlines(x=imp_times[2], ymin=0.998, ymax=1)
ax.scatter(timestamps, I_norm, marker='o', color='red', label='Data')
ax.plot(xdata_time, ydata, linestyle='dashdot', color='black', label=f'Fit'

#plt.ylim((0.9955, 1.0008))

plt.title('Run '+str(vid_num)+' and freq. ' + str(round(freq,3))+r' MHz (Fi

ax.set_xlabel(r'Time ($\mathrm{s}$)', fontsize='20')
ax.set_ylabel(r'$\mathrm{I/I_{max}}$', fontsize='20')

plt.legend(loc='lower right')


plt.savefig(path_out + 'Plots/run '+str(vid_num)+'.png', dpi=300, bbox_inch

df = pd.DataFrame({'Frequency (MHz)':[freq], 't_star':[t_star], 'alpha':[al

df_I = pd.DataFrame({'Timestamps':timestamps, 'Normalized intensity':I_norm
df_I.to_csv(path_out + 'I_norm/I_norm run '+str(vid_num)+'.csv', sep=',')

if os.path.exists(path_out + 'Resonance freq.csv') != True:
    df.to_csv(path_out + 'Resonance freq.csv', sep=',')
else:
    df.to_csv(path_out + 'Resonance freq.csv', sep=',', header=False, mode=
```