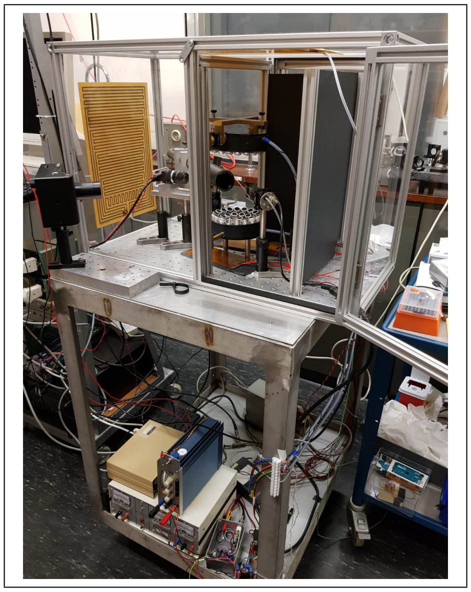
Universiteit Utrecht

Opleiding Natuur- en Sterrenkunde

Finding the Viscosity of Liquid Droplets using the Acoustic Levitator

Bachelor Thesis

Stijn Helsloot



Supervisors:

B. Yeroshenko MSc
Debye Institute for nanomaterials science

Dr. ir. Sanli Faez
Debye Institute for nanomaterials science

13 June 2019

Abstract

Viscosity is a way to measure the internal resistance of a fluid to deformation and it can be seen as the friction between the different layers of the fluid. Contact with a surface can therefore make it harder to measure the viscosity. Because of this we will use acoustic levitation, so that we can remotely measure the viscosity of a liquid droplet without it being in contact with a surface. During our experiments, we will change different variables that have influence on this viscosity. We will change the temperature, so that we can test the accuracy of our method. We will also change the volume fraction of ethylene glycol in demi-water, once again to test the accuracy. In our third experiment, we will look at the viscosity of water droplets containing nano particles because we also want to find out if we are able to see the colloidal glass transition.

Contents

1 Introduction

In physics, we quite often try to understand a complex model by first looking at the simple version of this model. In the field of nano materials, this often means that we want to study properties of these nano materials without interference of other variables (or at least as little interference as possible). A good way to study these properties is to use remote measuring. If we would be doing this for single particles, we might consider using optical tweezers. The latest groundbreaking work with optical tweezers was done by Arthur Ashkin[1], where he found that a laser could be used in order to capture small particles. For bigger particles/-droplets however, the power of the laser would have to be enormous. And since we are not interested in small particles, but big droplets, we have decided to go for another method that has this same useful property that there is no contact with a surface: acoustic levitation.

Acoustic levitation is the process in which some object is levitated with the use of sound waves. These days acoustic levitation is mainly used for containerless processing. This is how you transport a substance if you don't want the substance to come into contact with it's surroundings. [2] And this is sort of what we are using it for, or at least for the not touching it surroundings part. What we want to do is study the dynamic properties of fluid droplets. However, contact with a surface can sometimes change these properties, thus the acoustic levitator is a great way to remove contact without changing these properties.

One of the many studies that has been done in this field is from Kramer *et all.*. [3] Here they've tried to calculate the viscosity and surface tension of a droplet of liquid by using the change of the radius of the droplet inside of an acoustic field. This is done by having the surface of the droplet enter an oscillation. Now, though their final data show that their way is indeed both possible and very reasonable, we've decided that we wanted to do it differently. Kramer *et al.* actually only use the change in both the x- and y-axis of the radius. We, however, will be using how the entire surface of the droplet changes during this surface oscillation. We then calculate the viscosity of the liquid droplet by using the damping that this oscillation experiences.

In this thesis I'll start by giving you an introduction to the theory behind our work. In section 2, I'll explain to you how acoustic levitation works, what oscillation we are looking at, what the Cauchy-Lorentz Distribution is, how we go from the damping to the viscosity and what the data we acquire will look like. This section is ended by me giving you a little theory on what colloids are. In section 3.1 I'll show you what the setup of our experiment looks like as well as explain what equipment we have used and how it works. Section 3.2 then tells you about the measurement procedure, while in section 3.3 I explain how we process the data that we gain from our experiments. In section 4 I'll then also give you the results that we have acquired for our different experiments, while in section 5 I'll discuss what these results mean. Here I'll also mention what can be done to improve the results, such as optimizing, getting different equipment or switching to a new method.

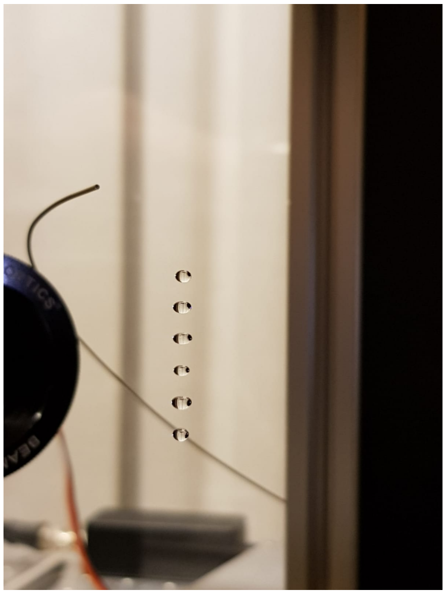
2 Theory

This section starts of with me explaining how acoustic levitation works. Then I'll show you what the oscillation we get looks like, what the Cauchy-Lorentz distribution is and why we use it and how we go from damping to viscosity. In section 2.5 I'll show you what the data that we gain will look like, while in section 2.6 I'll tell you what the colloidal glass transition exactly is.

2.1 Acoustic levitation

The main idea of this project is to use acoustic levitation to remotely measure the viscosity of a droplet. Acoustic levitation is a process in which some object is levitated with the use of sound waves. The way this works is because sounds are mechanical waves that contain momentum. This momentum comes from the fact that these mechanical waves create a pressure difference, which then exert a force on the object, which, if done correctly, are strong enough to levitate the object (see figure 1).

Figure 1: Multiple droplets levitating in the acoustic levitator.



This momentum is carried in standing waves, which are created by the actuators in our acoustic levitator. These objects can only float at points where the nodes of the standing waves are at. For a single standing wave, we know:

$$L = \frac{n}{2}\lambda \tag{1}$$

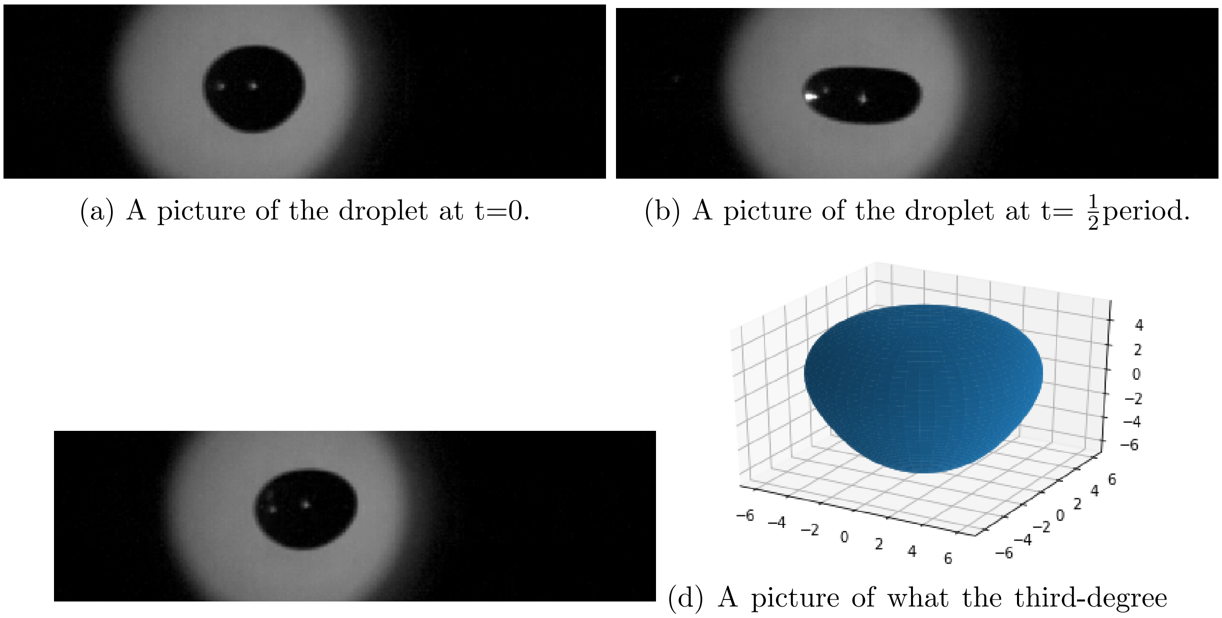
where L is the length of the "box" where the standing wave is in, n the node of the wave we are looking at and λ is the wave length. This formula tells us that the nodes of a single standing wave will be formed after each $\frac{1}{2}\lambda$. Now using the wave-equation and filling in the fact that we have a frequency of 40 kHz and a wave speed which is equal to the speed of sound (at room temperature), we get:

$$\lambda = \frac{v_{sound}}{f} = 8.6 * 10^{-3}\ m \tag{2}$$

Formula 2 tells us that, if we had been looking at a one-dimensional field, we could make an object levitate each 4.3 *mm*. However, we aren't looking at a one-dimensional field. Instead, we have many different actuator, which all produce a standing wave. Now, we have focused our actuators in such a way that we have about 6 stable point where we can levitate objects of a couple of milligrams in total. Another reason why we can't put in more than 6 object is because, whenever you put an new object in, it will interfere with the field and also weaken the field.

2.2 Explaining the oscillation

In figure 2 you can see three snapshots of the droplet at three different moments during the oscillation. Picture 2a and picture 2c actually look alike. This is because between these pictures exactly one period of the oscillation has passed. Now, picture 2b actually differs exactly half a period with both of these pictures, so if this would be an ordinary and simple oscillation, you would expect this one to be the exact opposite of the other two. However, though this picture does indeed differ from the other two, it isn't the exact opposite. This is because there are actually a couple of different oscillations going on at the same time (see 2.5). However, from the way the droplet looks in picture 2a and 2c, we see that it matches with a sphere where there would be a third-degree spherical harmonic on it's surface (see figure 2d).



(a) A picture of the droplet at t=0.

(b) A picture of the droplet at t= $\frac{1}{2}$period.

(c) A picture of the droplet at t = One period.

(d) A picture of what the third-degree spherical harmonic should look like.

Figure 2: In this figure you can see 3 different stages of the oscillation plus how theory predicts that a third-degree spherical harmonic should look like.

This is actually a big difference with Kramer *et all.*. In their article they show that they have a second-degree spherical harmonic on the surface of their droplet. For a second-degree spherical harmonic it suffices to only look at the x and y-radius, because this oscillation can be computed as if it is indeed only in these directions. In our case however this won't suffice, because of the asymmetry of our droplet and thus the fact that, unlike what Kramer *et all.* had, the droplet doesn't always look like an ellipse, which we can fit easily (see section 8 for how fitting is done).

This oscillation of the surface actually comes from the fact that we break the stable equilibrium we have created between gravity and the force of the acoustic levitator (see 3.1.2 for how we do this). When we break this equilibrium, the droplet will start to oscillate around the equilibrium point it was in before. Now, if this object had been a solid object, the only oscillations that would exist would be those of a translational movement. However, our droplet is made from liquid, thus there are also surface oscillations. Because of the fact that our liquid has a viscosity (and thus an internal friction), this oscillation will however die out after a certain period of time. It's because of this dying out that these surface oscillations can be seen as damped harmonic oscillations.

2.3 The Cauchy-Lorentz distribution

We are actually not just looking at a ordinary damped harmonic oscillator. This is because, in order to get our surface to oscillate, we need to give it a kick. This can be seen as looking at a driven underdamped harmonic oscillator, for which the kick can be approximated as a delta function. Now, by giving the surface this kick, it will start oscillating with it's own resonant frequency [4].

Frequencies further away from this resonant frequency will actually be influenced less by this kick, since the surface of the droplet prefers to oscillate with it's resonant frequency. To give this phenomena a name, a special term was devised called the quality factor of the damped harmonic oscillator. For an underdamped harmonic oscillator, this quality factor Q is given by:

$$Q = \frac{\omega_0}{\Delta\omega} = \frac{\omega_0}{2\delta} = \frac{1}{\tau} \tag{3}$$

where ω_0 is the resonant frequency ,$\Delta\omega$ the Full Width at Half Maximum (FWHM) of the resonant frequency (also called the resonant width), δ a scale parameter and τ the damping constant of the frequency. This FWHM is the point at which the amplitude of the frequency has gone down by 50 percent. A higher Q will actually mean a smaller damping constant and thus a weaker damped system (and thus more underdamped).

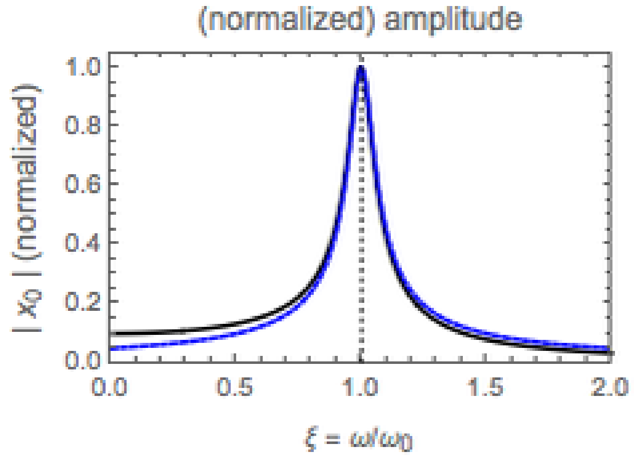


Figure 3: An example of how a lorentzian curve should look like.

If you would look at the amplitude of a driven underdamped harmonic oscillator (where the drive is a kick) and would plot this vs the different frequency's within the system, one would see a big peak at the resonant frequency which, if you move away from this frequency, falls off to a value close to zero (figure 3)[5]. Such a curve is called a Cauchy-Lorentz Curve. This curve is given by the following formula:

$$f(\omega, A, \delta, \omega_0) = \frac{A}{\pi} \frac{\delta}{(\omega - \omega_0)^2 + \delta^2}, \tag{4}$$

where f is the height of the peak, A the amplitude of the oscillation, ω_0 the resonant frequency (the center of the peak), ω the frequency and δ again a scale parameter.

Since we are indeed looking at a driven underdamped harmonic oscillator ourselves, it will indeed be this curve that we are looking for in our amplitude-spectra. Now, the correct way to get the values that you need from a spectra, is to fit a known function to your data and then get the values from there. And this is indeed what we'll be using the Cauchy-Lorentz Curve for throughout our experiments.

2.4 From Damping to Viscosity

As said, we want to calculate the viscosity of the droplet. To do this, we are gonna use a formula found by professor Horace Lamb. He has found a relation between the viscosity of a spherical droplet and the damping constant of the fluid, given by:

$$\mu = \frac{R^2}{(n-1)(2n+1)\tau}[6], \tag{5}$$

where τ is the damping constant, R is the radius of the sphere in undisturbed state, n is the mode and μ is the viscosity.

The problem that we have with this formula however, is the fact that, unlike professor Lamb, we don't have spherical droplets, but ellipsoidal droplets. However, if we lower the force done by the acoustic field, our droplets would indeed become less ellipsoidal and more spherical again, just as he had. These droplets will actually give the same result for the oscillation. However, because the acoustic field has become weaker, keeping such a droplet levitated is quite hard, thus in the end we still do the experiment with ellipsoidal droplets. And, since kramer *et al.* have also used formula 5, we assume that we can use it as well. So what we do is calculate the volume of our ellipsoidal droplet and then act as if it had been a spherical one to calculate R, so using:

$$R = \sqrt[3]{\frac{3 * V}{4\pi}}, \tag{6}$$

where V is the volume of our ellipsoidal droplet. We can now use formula 3 and 6 to rewrite formula 5. Before we do this, we first rewrite formula 3 in such a way that we can plug it in, so this becomes:

$$\tau = \frac{2}{\Delta\omega} = \frac{1}{\pi\Delta f}, \tag{7}$$

where we have used that $\Delta\omega = 2\pi\Delta f$, or in words: $\Delta\omega$ is the FWHM in radians per second, while Δf is the FWHM in Hz . Now, using this formula for τ and the fact that we are looking at a third-degree spherical harmonic (so $n = 3$), we get that the viscosity is given by:

$$\mu = \frac{R^2\pi\Delta f}{14}, \tag{8}$$

2.4.1 Dynamic vs Kinematic viscosity

A last thing that needs to be said about the viscosity, before we can continue, is what kind of viscosity we are actually looking at. Later on in the graphs and tables, you'll be reading the term "dynamic viscosity" a lot. This dynamic viscosity isn't the same as kinematic viscosity. The dynamic viscosity is also called the absolute viscosity, since it's only measured by the internal resistance of a fluid, while the kinematic viscosity differs with a factor $^1/_\rho$, where ρ is the density of the fluid.

Now, we have chosen for the dynamic viscosity, since calculating the density of our droplet would make it a lot harder without giving us a lot of extra knowledge. However, in the text (above and below) we often don't specify that we are looking at the dynamic viscosity and not kinematic viscosity. So, now you know that whenever there is written "viscosity", it should have said "dynamic viscosity".

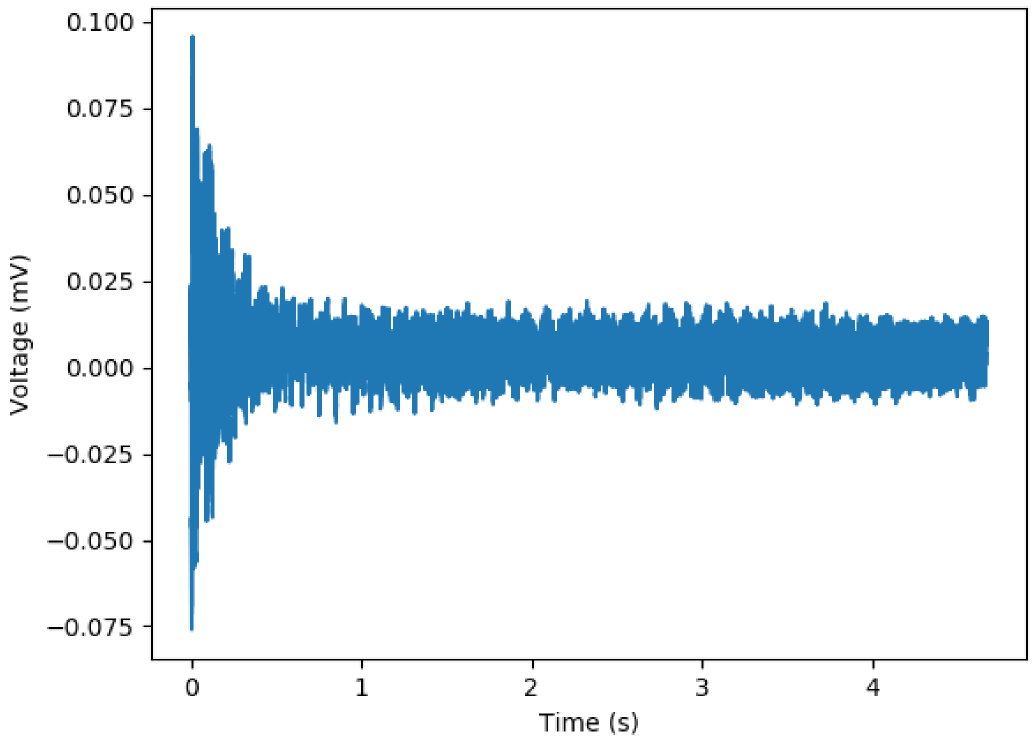


Figure 4: An example of how the data of the diode could look like. On the x-axis you can see the amount of measurements taken, while on the y-axis the amount of voltage the diode measures. In total we have done $7 * 10^5$ measurements with a frequency of $15 * 10^4$ Hz.

2.5 Analyzing the spectrum of the diode

In figure 4 you can see what the oscillation of a droplet looks like, looking at the raw intensity data we get from our measuring equipment. This is indeed the figure one would expect to see for a underdamped harmonic oscillator. The question that now arises is: ”What values do we want to get out of this data in order to calculate the viscosity?”

In order to calculate the viscosity, we will need to know both the size of the droplet as well as the FWHM of the frequency of the oscillation. The size of the droplet is a value that we won't be able to get out of this data. However, the value that we can get out of this data is the FWHM. In order to get the FWHM, we need to fit the Cauchy-Lorentz Curve from section 2.3 to the amplitude vs frequency spectra of our oscillation. The way to get this spectra is by taking the Fourier Transform from the data you see in figure 4. The Fourier transform is a way to decompose a function that depends on time into the frequencies of it's change. In our case, this will give us a frequency spectrum like the one you can see in figure 5. In this figure, you can actually see 8 different physically explainable peaks (you might think that you see more, but the rest is just noise). This leads to another question: What peak are we actually interested in? Cause we only need to know the FWHM of the resonance frequency of the droplet.

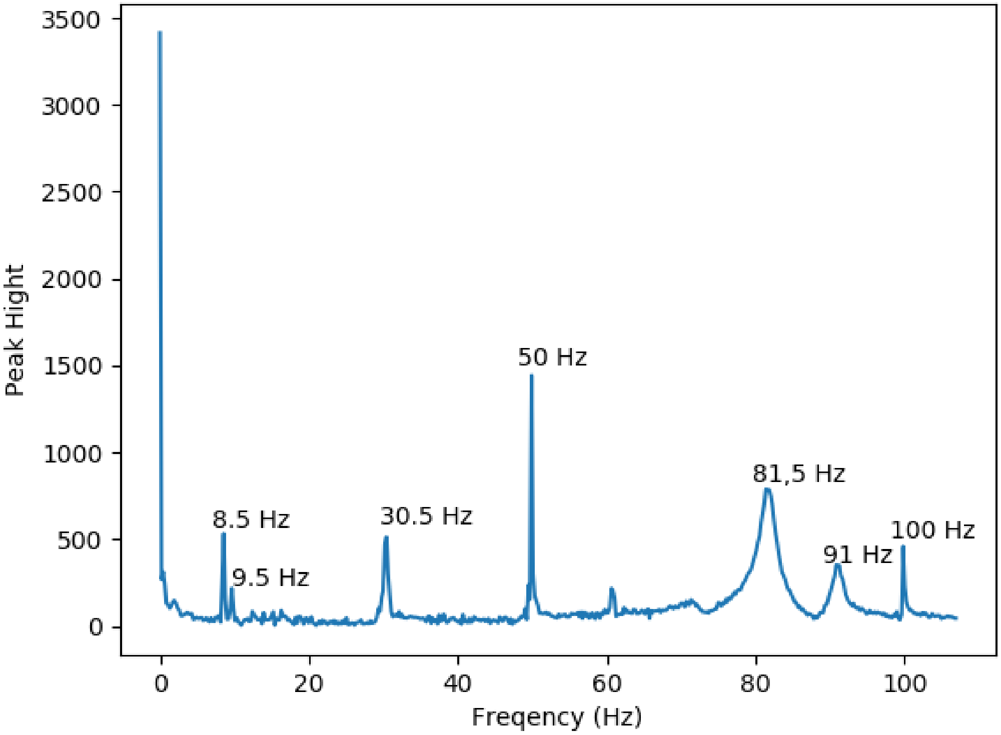


Figure 5: An example of how the frequency spectrum could look like (this spectrum matches with the diode-data from above). On the y-axis you can see the peak height. It's important to add here that this isn't always the same as the amplitude of the oscillation, because of the fitting program that we use.

For the full analysis of the spectrum and what all of the different oscillations are that the peaks represent, you should take a look at 4.1, but for now I'll tell you that only the peaks at 81.5 and 91 Hz actually have anything to do with the surface oscillation, where in the end, we'll only use the FWHM of the first peak (81.5 Hz).

2.6 The colloidal glass transition

Let's start with a quick recap of our chemistry classes. Colloids (or colloidal suspensions) are mixtures between solutions and particles ranging between 1 and a 1000 nm in diameter. Colloids exhibit a so called colloidal glass transition. [7] This is a transition between a colloidal droplet (liquid) that you start with and a solid drop that you end with. This transition happens because the liquid in which the colloidal particles were originally mixed has evaporated. Because of the fact that the liquid evaporates, while the particles stay, the viscosity rises during this transition. [8] The way that the viscosity should rise is exponentially, which starts off with a slow slope in the beginning and ends with a steep slope up until the point when the droplet has become so solid that we can no longer measure the viscosity.

3 Methods

In the following section, we'll start by telling you what equipment we have used and what each part does. We'll then explain the experimental procedure and how we process the data that we achieve.

3.1 The equipment used

In figure 6 you can see the setup of our experiment. Some of the more important parts have been listed below, with a short explanation of what they do.

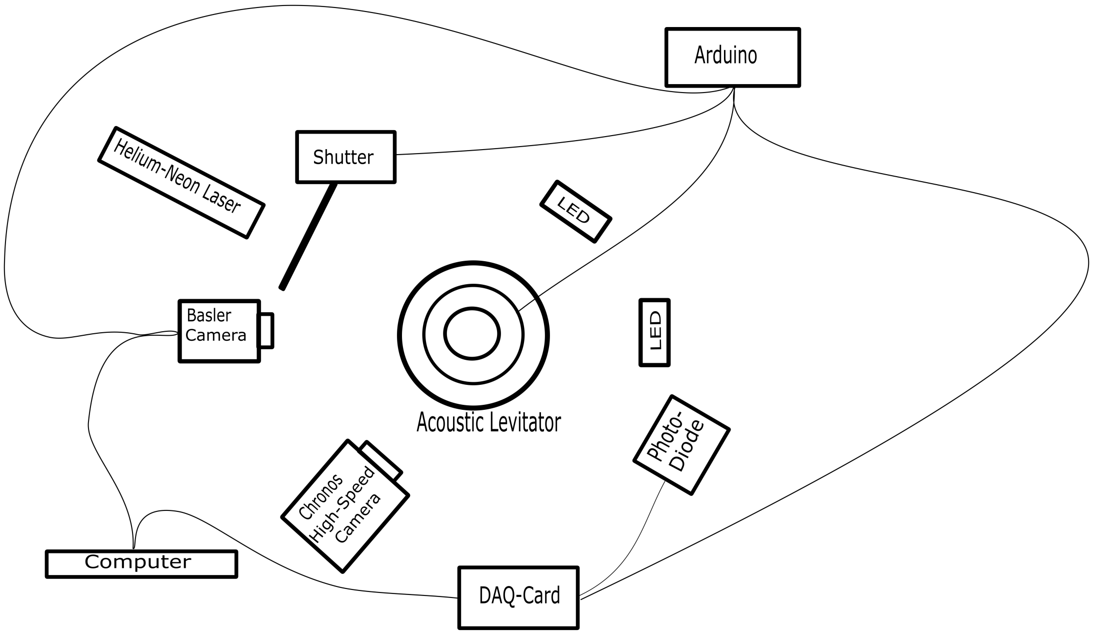


Figure 6: A block diagram of our setup. Each block represent a piece of equipment. What it is, is written inside of the block. The wires in this figure represent a data transfer from one piece of equipment to another.

3.1.1 The acoustic levitator

Let's once again start with the acoustic levitator. Our acoustic levitator has been based on the so called "TinyLev" [9]. It consists of two curved planes with a radius of 110.5 *mm*. On these domes we have placed 72 actuators, which all have a sharp resonance at around 40 kHz. These actuators are connected to a signal generator, which outputs a square wave of 40 kHz with a peak to peak amplitude of 4.8 V, a voltage amplifier and a external power supply which supplies 12 V.

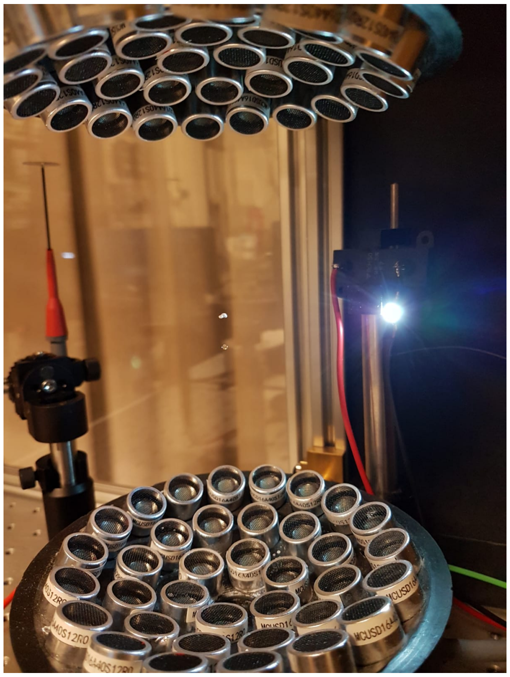


Figure 7: A close-up of the acoustic levitator so you can clearly see what it looks like and how the actuators and both domes have been placed.

3.1.2 The arduino

If you look at figure 7, you can see that the actuators on each curved plane have been placed into three different rings. The middle ring has been connected to an arduino. An arduino is a single-board microcontroller. Arduino's are used to both sense and send digital signals. We use our arduino to turn our actuators on/off, to tell our DAQ Card (see 3.1.5) when to measure and our camera when to take a picture (see 3.1.4).

3.1.3 The laser and the photodiode

We decided that the best way to measure the movement of the surface of the droplet was to use the change of intensity that a photodiode measures. This is accomplished by using a Helium-Neon laser which operates at an wavelength of 623.8 *nm*. This laser is pointed towards both the droplet and the photodiode. Because of the fact that the droplet blocks and deflects some of the incoming light, we can measure movement by the change in intensity. This change in intensity is then transformed into a change in voltage by the diode, which is then send to our DAQ Card.

3.1.4 The camera's

Since we also want to know some import physical features of the droplet, we also need to regularly take pictures from the droplet. This is done by using a camera called the Basler dart daA1280-54um, which is a form of camera which can directly be connected to a computer. This means that it can easily be controlled with code instead of having to try to control it by hand. Because of the fact that we can control it with code, making the measurement automated is a lot easier.

We have also made use of a camera called the Chronos 1.4. This is a high-speed camera which has a maximum speed of 38565 frames per second. This high-speed camera was used for the comparing the frequency spectra from figure 5 with the actual movement from the droplet. This way, we were able to tell how each peak could be physically explained.

3.1.5 The data acquisition device

The instrument that we used in order to get and process the acquired data is the keysight U2331A Data Acquisition Device. This machine has been programmed in such a way that it does $7 * 10^5$ measurements with a frequency o $15 * 10^4$ *Hz*. To make sure that the DAQ-device can store the data, it first saves the data as bits. Once it's done with the measurements, it sends the acquired data to the computer in these bits. We can than convert these bits to Volts by using the following formula:

$$\text{Converted Value (Volts)} = \left(\frac{2 * \text{Non} - \text{Converted Value (Bits)}}{2^{16}}\right) * 0.2 \qquad (9)$$

Where the 16 comes from the fact that this is multitude of bits in which our machine measures and the 0.2 from the fact that this is the range of expected input voltages.

3.2 The experimental procedure

Now on to explaining how the experiment is actually done. We start by turning on all of our equipment and running a few quick tests to see if everything is working the way it should. This means that the laser and the LED's will be on at all time. Once this is done, we start by running the code of our arduino, so that all of the equipment starts working synchronized. We also choose the order in which our equipment works, by sending a triggers with the arduino at the correct timing. The sending of these triggers have all been automated with the code. We can now start running our main code, which is called DAQControlDIGitize (see Appendix C for the actual code).

Now the experiment goes as followed:

Step 1: The first thing that we need to know, before we measure the oscillation of the droplet, is the size of the droplet. This means that we have to take a picture of our droplet and process it. Before we take this picture, the arduino actually first makes the shutter close, so that it blocks the laser. It stays there for 1 second, after which the arduino opens the shutter again. In this time the arduino tells our basler camera to take a picture of the droplet. The reason why we need this shutter to block the laser, is because otherwise this laser will give an extra reflection, which will make it harder to find the size of the droplet.

Step 2: Now that we have taken a picture, we can start the measurement. Our arduino has been programmed in such a way that every 14997 milliseconds (*ms*), the middle ring of the acoustic levitator is turned off for 3 *ms*. In the time that this ring is turned off, the levitator will no longer exert enough force on the droplet for it to levitate, thus it falls. The moment this ring is turned back on again, the droplet is once again pushed upwards into it's equilibrium position. However, since this "re-trapping" isn't a instant process but happens gradually, the droplet will first oscillate around this equilibrium point.

Step 3: Now that both our droplet and it's surface are oscillating, the arduino tells our DAQ-device that it is time to start gathering data. So our photodiode measures the change of the intensity (thus how the droplet oscillates), our DAQ-device temporarily stores this data and once it has indeed gathered $7 * 10^5$ data point, it sends this data to the computer where our python file is able to convert the bits to volts and then saves this data in the correct folder.

3.3 Processing the data acquired

There are two different kind of data sets that we actually get from one single measurement. We get a picture of the droplet, from which we want to calculate the size, and we get a list of intensity values that the diode has measured from which we, ultimately, want to calculate the damping constant of the oscillation.

3.3.1 The droplet size

In fig 8 you can see how we measure the size of the droplet. As you can see here, we have fitted an ellipse to the edge of the droplet. This is done by running the code found in Appendix A. By running this code, we'll actually get 8 different values for the ellipse. Many of these values are connected with each other. The first 4 are actually found by using mathematics to fit an ellipse to the edge. We then use these 4 values to calculate the latter 4 by using the formulas:

$$a = y\cos(\frac{\pi}{2})\sin(\theta_0) - x\sin(\frac{\pi}{2})\cos(\theta_0) \tag{10}$$

$$b = y\cos(0)\sin(\theta_0) - x\sin(0)\cos(\theta_0) \tag{11}$$

$$V(\text{Volume Ellipse}) = \frac{4}{3}\pi \mathrm{a}^2\mathrm{b} \tag{12}$$

where the last equation is quite self-explanatory. However, Equation 10 and 11 might not seem that obvious, but these come from the standard formula's for an ellipse, where we have actually already filled in the values for the angles (since we wanted to know the a and b length of our ellipse). A quick side note here is that θ_0 is the angle that the ellipse makes in comparison to the x- and y-axis.

Now, the value that we need to calculate the viscosity, can be found by using the formula in 6. One could naturally assume here that instead of using a sphere of the same volume

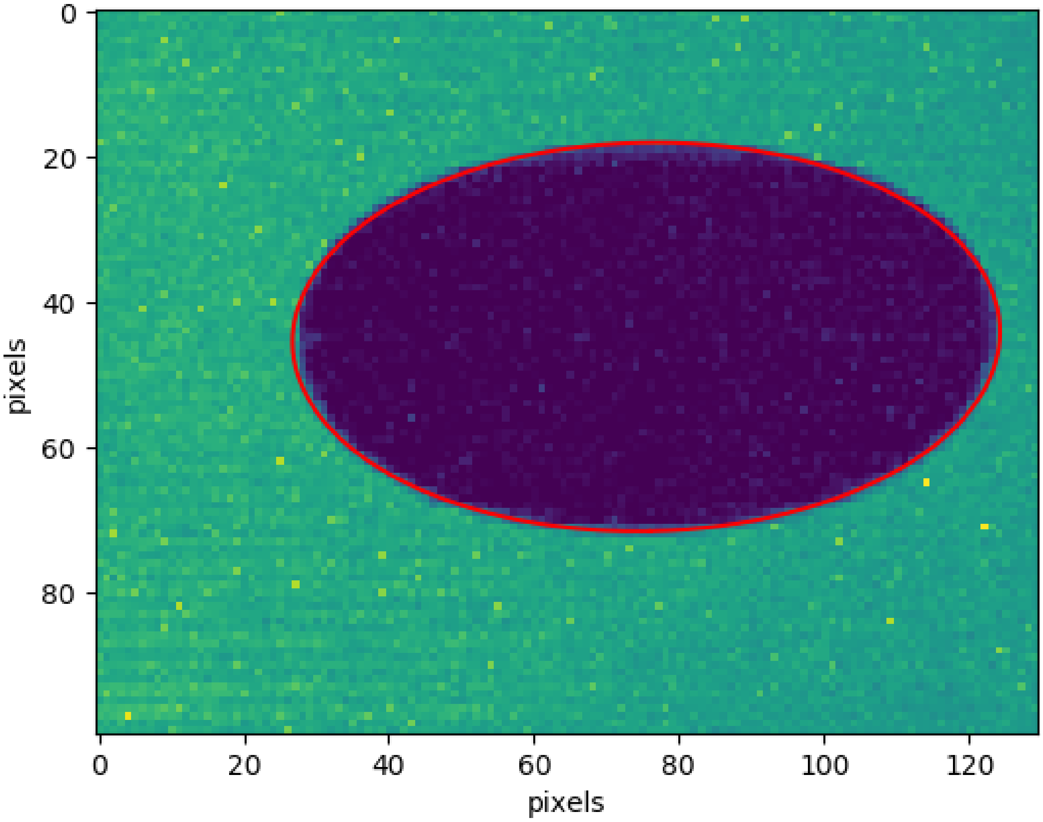


Figure 8: A picture of the droplet taken by the basler. The red ellipse is our fit of the edge of the droplet, which we do so that we can find the size of the droplet

to calculate R, one could also use a circle of the same area. This however is a physically incorrect assumption, since Lambs analysis and also future analysis was based on a sphere. But even it hadn't been a incorrect assumption, we could still see that it's wrong because of the data this method gives us. If we do multiple measurements in quick succession, one should find that, due to evaporation, the R should shrink. If we calculate this R by using the volume, this is indeed the case. However, if we use the area of the ellipse, this doesn't always happen. The reason that the area of the ellipse can grow while the volume shrinks, is because off the fact that there is a third dimension that the area doesn't take into account. And since we assume that the droplet is spherically symmetric, not taking this dimension into account can cause big fluctuations in R.

3.3.2 The full width at half maximum

In 2.5, we have said that we only need to look at one single peak, so now we can start trying to get the values that we need from the data. The proper way to do this is by fitting a function to the important data and then taking the values from there. For us this will mean that we will have to try and fit a peak. As explained in 2.3, we have chosen the Cauchy-Lorentz distribution.

Now, since we only need to look at one peak, you might think that we would also only need to fit one function and that would be it. However, while doing analysis we have learned two things. First off, in order to cancel out the noise that we have, we also need to fit a baseline. The reason we do this is to cancel out the Gaussian noise that we have throughout our spectrum. Secondly, because of the fact that the 2 or 3 peaks that come from one peak, they are always quite close to each other. Because of this, we fit all of them, to make the fit of the important one better (For the code that we use to fit, see B). In the end, the fitting will look something like the fit in figure 9.

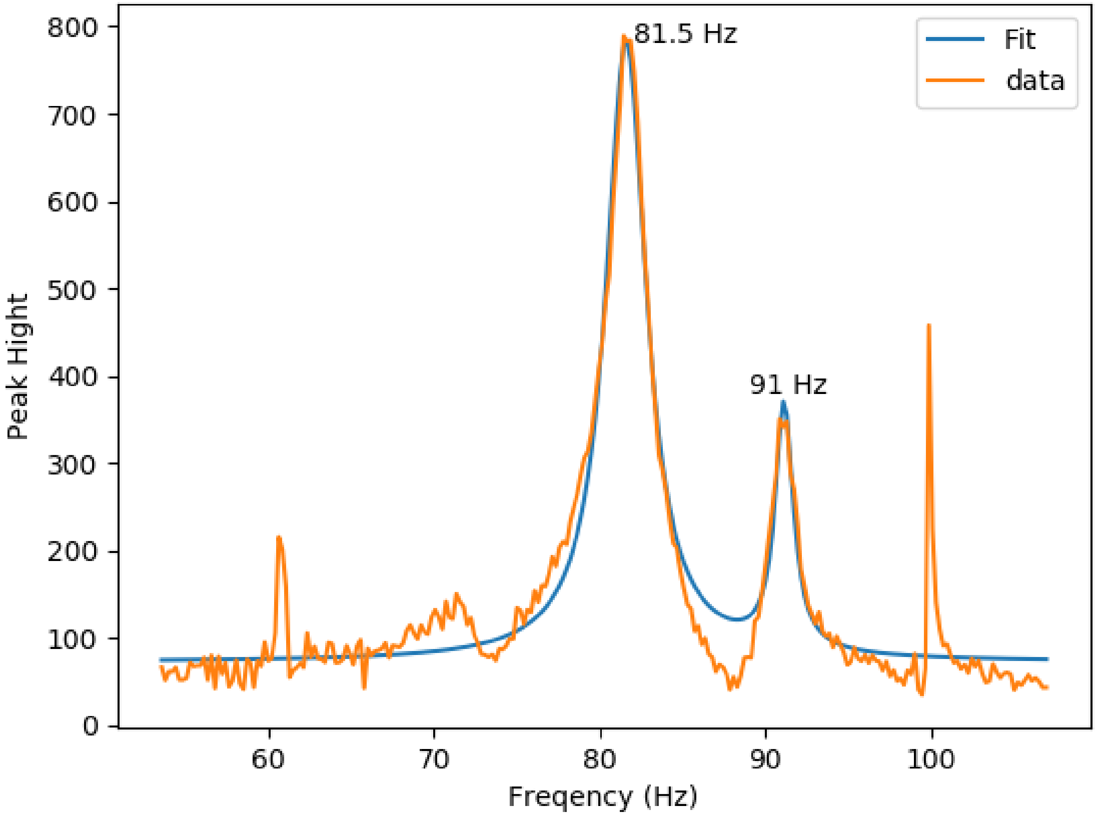


Figure 9: Same frequency spectrum as before, but now zoomed on the frequency range from approximately 50 to 110 Hz. Also plotted here is the Cauchy-Lorentz curve that we fitted to the peaks at 81.5 and 91 Hz.

4 Results

In total we have done three different experiments. The results of these three experiments can be found below. All of these experiments were predecessed by the following question: "What happens to the viscosity if we change ....?

4.1 Defining the peaks

Before we can do any of the experiments, it's important that we fully understand what we are looking at. Because of this, we started with a full analysis of the movement of the droplet. This is done by using the Chronos High Speed Camera. Using this camera and many of it's recordings, we've coupled the peaks that we can see in our frequency spectrum to different movements that we can see on our recordings. So let's go back to figure 5. Of the 8 different peaks we see, we'll actually only use the FWHM of 1 peak. This doesn't mean that we can just ignore all of the other peaks, cause they are there with a reason. So let's name all of them, explain why they are there and why we will/will not use them.

0 Hz: The 0 peak in a Fourier spectrum is actually the DC-component (so an mean value). Since this peak isn't interesting to us, we have tried to make it smaller by using a DC-filter, to filter out some of the noise. However, this DC-filter isn't perfect so we still get a high peak here. However, since it doesn't have anything to do with the oscillation of the surface of the droplet, we just ignore it.

8.5, 9.5 Hz: These two peaks both represent translational motion. The one at 8.5 Hz represents the motion from left to right (compared to the diode) while the one at 9.5 Hz a motion parallel to the direction of the laser. These two peaks don't mean anything for the viscosity of the droplet, so we can ignore them. It's even so that, the more stable the oscillation of the droplet, the lower these two peaks. So in a perfect world, these wouldn't even have been here in the first place.

30.5 Hz: Another translational motion, but this time it's the motion of the droplet up and down. Also not important for the viscosity, so ignored, though we do use this peak as a check. If we don't see this peak, something has gone wrong, cause it should always be there, since having the droplet fall down is our way of getting both the surface and the droplet to oscillate.

50, 100 Hz: These peaks actually don't represent any type of movement of the droplet. What they do represent is the movement of the electrons in our wires (so electrical noise). This movement is always with a frequency of around 50 Hz and it's multiples (so 100 Hz, but also 150 Hz, 200 Hz, etc). Again not important, so dropped.

81.5 Hz: This peak is actually the peak that we are going to use to calculate the viscosity.

91 Hz: If we would have had a simple model, this peak wouldn't have been here. However, because of peak-splitting, the resonant frequency peak has been split into 2 peaks (can sometimes also be split into 3 peaks). The reason that this happens is because of the aspherical shape of the droplet.[10]

4.2 Viscosity as a function of temperature

Whenever one thinks about the viscosity of a liquid, it's usually quite quickly coupled to the temperature of that liquid, because temperature has such a big influence on the viscosity. So it's quit natural, that we'd start there as well. To do this measurement, we'd first crank up the heat of our setup and than let it cool down. During this cooling down period we would do the measurements. The reason we did it like this, is because otherwise we would have to much extra convection currents to deal with, which would make the processing of the acquired data much harder.

In total, we've done 18 measurements going from 33.822 °C to 39.587 °C. Our results can be seen in figure 10. As you can see, there aren't any error bars within this graph (the explanation for this can be found in section 5.1).

As you can see, the line fitted to our data is quite close to the line predicted by theory. However, the slope of our line is actually way off compared to the original. It's actually because of these results that we don't truly trust our temperature reader. So from here on out, whenever we talk about a temperature that we measured, you should definitely keep in mind that it could also be 2 to 3 degrees higher or lower.

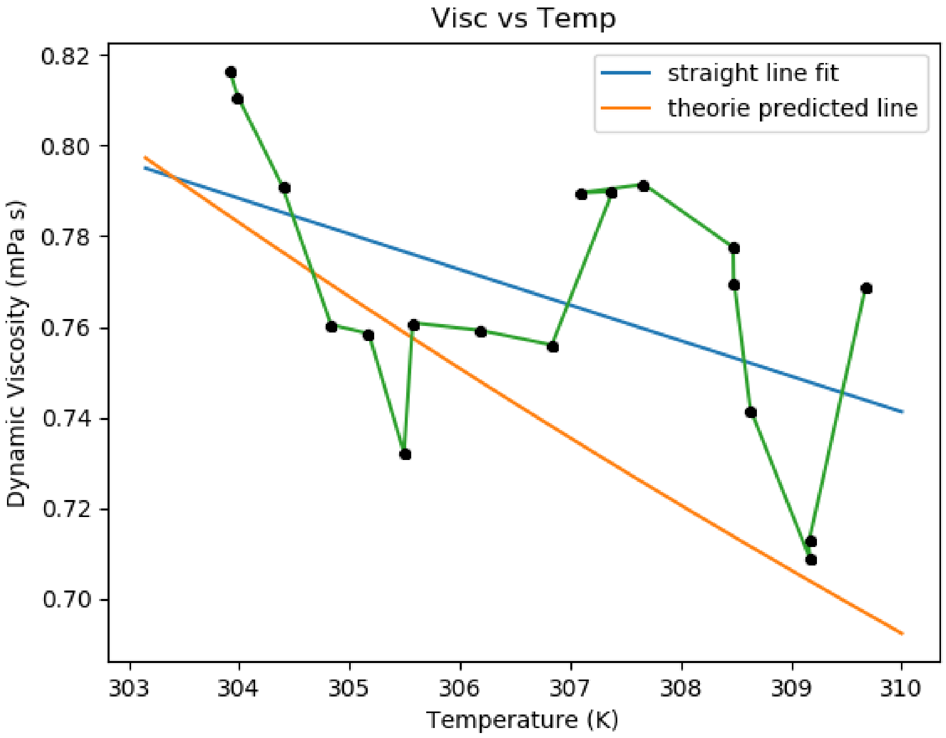


Figure 10: In this figure you can see the data that we acquired when calculating the viscosity of a drop of demi-water for different temperatures. We've also fitted a straight line through our data and plotted this one, as well as the line predicted by theory.

Volume Fractions $({}^{EthGl}\!/\!{}_{Water})$	Dynamic Viscosity $({}^{Ns}\!/\!{}_{m^2} * 10^{-3})$	Error $({}^{Ns}\!/\!{}_{m^2} * 10^{-3})$
0	0.8863	± 0.1016
$^1\!/\!_{178}$	1.1816	± 0.0723
$^1\!/\!_{64}$	1.0625	± 0.0781
$^1\!/\!_{32}$	1.1609	± 0.0769
$^1\!/\!_{16}$	1.0210	± 0.0933
$^1\!/\!_{8}$	1.4902	± 0.0842
$^1\!/\!_{4}$	1.8354	± 0.1411
$^1\!/\!_{2}$	2.8001	± 0.3217

Table 1: The values that we have found for the viscosity for different volume fractions of Ethylene glycol vs demi-water.

4.3 Viscosity as a function of volume fraction

The second experiment that we have done is what we call the ”Volume Fraction vs Viscosity”-experiment. Here we have taken 8 different volume fractions of ethylene Glycol vs Demi-Water. Of all 8 different volume fractions, we have placed a small droplet within the acoustic levitator. For each droplet we have then done 8 different measurements, all around 20 seconds after each other. We than average over these 8 values (or sometimes over 7 or 6, if measurements have gone wrong) and give them as one single point.

The different volume fractions that we used plus all of the values we get for the viscosity's for these volume fractions are listed in Table 1. All of these values below to a temperature of around 23 °C.

The way that we have calculated this errors are by using the formula:

$$\text{Error} = \mu\sqrt{\frac{\sigma_\tau}{\tau}^2 + \frac{\sigma_\mathrm{R}}{\mathrm{R}}^2} \tag{13}$$

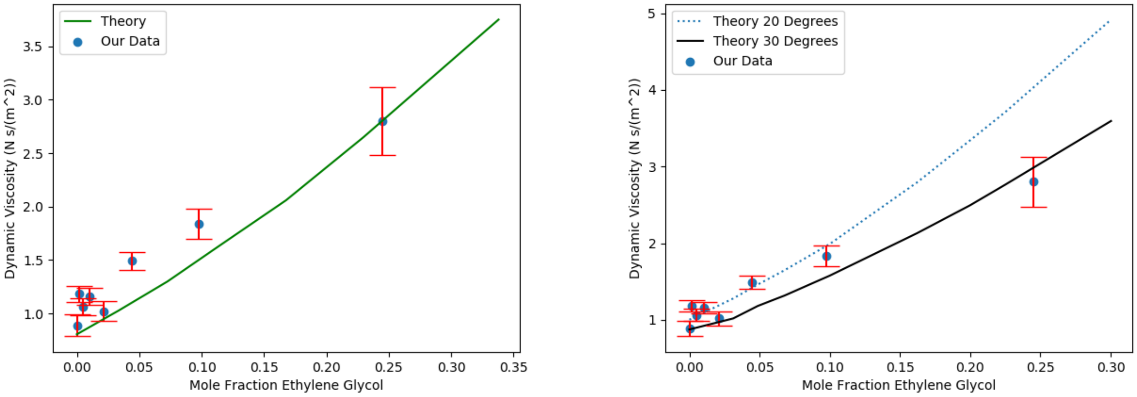
where σ_τ is the error in the damping constant, σ_r the error in the R of the droplet, R the radius of the sphere of equal volume as the ellipsoidal droplet, τ the damping constant and μ the calculated dynamic viscosity.

Because of the fact that we also want to see if our values match with the theoretical values, we need to make one little tweak. The theoretical values that Dizechi *et al.*[11] found are not plotted vs the volume concentration, but vs the mole-fraction. However, we can calculate this by using:

$$Mole\ Fraction = \frac{0.017935}{0.017935 + \frac{0.9957*Parts\ Water}{18.0153}} \tag{14}$$

where the 0.017935 comes from the fact that we always have only 1 part ethylene glycol, the 0.9957 is the density of demi-water at 30 °C and 18.0153 is the mole mass of demi-water. Using this formula, our plot becomes the one in figure 11a. Our results show that for a low concentration, we can't really say what the correlation between concentration and viscosity is. However, once the concentration reaches a certain point, it starts to look like there is a linear correlation between the concentration and the viscosity. You can also see that the error bar becomes larger once the viscosity increases while it stays the same when the viscosity stays the same.

In figure 11a, you can see our data being compared to the values that Dizechi *et al.* found for the viscosity as a function of the mole fraction. During there experiment, the temperature was 30 °C.



(a) The green line is the correlation between viscosity and mole fraction found by Dizechi *et al.* for a temperature of 30 °C. The blue points are the data that we found for the viscosity as a function the mole fraction, while the error bars are given in red.

(b) The blue points are the data that we found for the viscosity, while their error bars are given in red. The striped line is what Tsierkezos *et al.* found for the viscosity vs mole fraction, while the temperature was 20 °C. The black line is what they found while it was 30 °C.

Figure 11: In this figure you can see the viscosity's of different fractions of ethylene glycol vs demi-water (given in the different Ethylene Glycol Mole Fractions). Each point on the graph is actually composed out of 8 different measurements, over which was then averaged. This is done for a temperature of around 23 °C. Here we have compared our own data with two different researches.

In figure 11b you can see our data being compared to another research done by Tsierkezos *et al.*.[12] They also measured the viscosity for different ethylene glycol mole fractions, but they have also done this for multiple temperatures (20 and 30 °C). And, since we don't entirely trust our temperature machine, plotting our own data next to theirs can give us a more definite answer as to whether we are in the correct ballpark for the viscosity or not.

	PMMA		TiO2	
Time	Dynamic Viscosity	Error	Dynamic Viscosity	Error
(ms)	$(^{Ns}/_{m^2} * 10^{-3})$	$(^{Ns}/_{m^2} * 10^{-3})$	$(^{Ns}/_{m^2} * 10^{-3})$	$(^{Ns}/_{m^2} * 10^{-3})$
0	1.4677	± 0.2459	2.0280	± 0.1535
198			2.3114	± 0.2216
199	1.6651	± 0.1860		
382			2.7111	± 0.3529
383	1.9636	± 0.2764		
566			4.3177	± 0.2585
567	2.5275	± 0.5568		
750	7.4356	± 0.3776	11.4194	± 0.1230

Table 2: Our values for the viscosity and there errors of both PMMA and TiO2 in time.

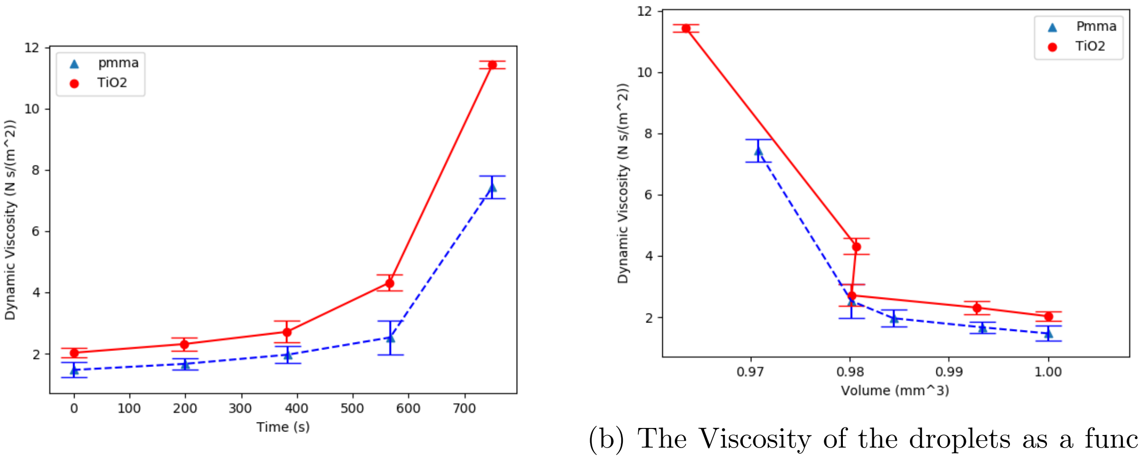
4.4 Finding the colloidal glass transition

For our last experiment, we have looked at how the viscosity of different colloids looks as a function of both time and the size of the droplet. Here we did 1 measurement once every 180 seconds for both PMMA and TiO2. The results of this you can see in table 2.

As you can see, the times between different points don't always match for the PMMA and TiO2. This is because the computer sometimes needs a little bit longer to process the data.

The points that you see in Table 2 are also once again given in a graph (see figure 12a). Here you can see that though the absolute value of the two different solutions is different, they do follow the same kind of correlation between time and viscosity, which appear to be some form of exponentially growing function.

We've also plotted these same viscosity's vs the size of the droplet at that point (figure 12b). And as you can see, the functions once again look the same. For TiO2, one point isn't where we would expect it to be. However, this point is an error, since the size of the droplet can't grow during measurements without us adding more colloid. So this point is there because of an fitting problem to the image that the basler camera had made. However, apart from this one point, you once again find the same exponential function as before, now just mirrored.



(a) The viscosity of the droplets as a function of the time. The blue line gives the viscosity of PMMA, while the red line gives the viscosity of TiO2.

(b) The Viscosity of the droplets as a function of their size (compared with the size at which they began.) The blue line gives the viscosity of Pmma, while the red line gives the viscosity of TiO2. For the TiO2, one point is off from the expected exponential curve.

Figure 12: Two figures in which we have plotted the viscosity of PMMA and TiO2. In both figures you can see that they both posses the characteristics of an exponential curve. The temperature for which we did these measurements was around 23 °C.

5 Discussion

. Now that we have showed you the data that we have acquired, let's discuss whether this data is correct and if our methods should be improved.

5.1 Viscosity as a function of temperature

A big difference actually between this experiment and all of the later experiments is the fact that both our setup and our analysis hadn't been optimized yet, so the results of this experiment aren't as thrust worthy as we would wish them to be.

Let's go back to the graph in figure 10. There are 2 reason why we haven't used any error bars in this graph.

- The first reason is because of the fact that we hadn't yet optimized both the setup and the analysis. This meant that a lot of the values that we had to use, were gained by just looking at the frequency spectra by eye. Because of this, we can't really put any reasonable error bars on our results.
- The second reason is because we don't really trust our temperature machine, because of the weird jumps it can sometimes make. What we mean by this is the fact that, sometimes, the temperature between measurements can just randomly go up a few degrees without us directing any extra heat to our setup. An example of this phenomena can be seen when, inside of our graph, you look at 307.5 K. There you can see that the temperature has randomly jumped up again.

The first problem with this experiment (the optimizing) can be solved quite easily, since we've already done this. So for better result, we'd just have to redo the measurement. However, this won't solve the problem we have with the temperature machine. And to be truthful, we don't really yet understand why it has these weird jumps. So that's definitely where the focus for this experiment should lie in the future.

5.2 Upgrading the experiment

During this thesis, the setup has been improved and upgraded quite a lot. One of these improvements is the fact that we have stopped using the high-speed camera for getting the data to calculate the viscosity. We have still used this high-speed camera to find which degree our spherical harmonics had and which peak belonged to which oscillation. For the data needed to calculate the damping constant however, we have started using the Diode, Laser and DAQ-device. Another improvement was finding the correct degree of the spherical harmonic. My predecessor's experiment [13] had been based around trying to get the damping constant via the radius of the droplet during the oscillation using the camera, since this is what kramer *et al.* did as well. However, the "flaw" that was made is in the fact that, to get the radius of the droplet, the assumption was made that the droplet would still be an ellipse during the oscillation. In the case of kramet *et al.*, this was a correct assumption, since they had a second-degree spherical harmonics on the surface of their droplet. For us however, this assumption wasn't correct, since we have a third-degree spherical harmonic.

Another inconvenience that working with the camera brought, was the fact that processing it's data took way to long. Saving the clip that the camera had made already took around 50 minutes, than cutting this to frames takes 30 minutes and fitting an ellipse to all of the frames costs another 45. This means that doing 1 measurement takes you more than 2 hours. Using the DAQ however, we get preciser results in a shorter amount of time , which is around 20 seconds for a single measurement.

5.3 Viscosity as a function of volume fraction

In figure 11a we saw that, for concentrations of ethylene glycol in Demi-Water above 10 percent (mole fraction of 0.05), the correlation we get seems to be a linear one. However, for concentrations below this value, it seems that we get a straight line (a constant value). This leads to us asking ourselves: Which of these two correlations is correct?

To answer this question, it's always a good idea to first start by looking at what other people have found with different methods. In figure 11a, we had also plotted the viscosity's found by Dizechi *et al.*. Looking at their data and than back to ours, we can see that we are in the right ballpark. It does however look like we are always a little bit to high. This is probably because of the fact that they did their measurements at exactly 30 °C, whilst ours temperature wasn't as stable, nor was it 30 °C. Since our temperature reader wasn't the best, we can't be exactly sure, but our temperature was somewhere around 23 °C. Now, a lower temperature is indeed cause to a higher viscosity, so this would meant that our data do follow the correct trend. It's because of this that we have also compared our data to the data of another research (figure 11b). They did the same measurement, but they also did it for 2 different temperatures (20 and 30 °C). And here you can indeed see that most of our data is between these 2 lines, with once again proves that we are in the correct ballpark.

Having answered this question, you might already be able to guess as to why we aren't able to get the correct correlation below concentrations of 10 percent. This is because our set up isn't yet precise enough for this regime. However, the fact that we are able to find values that are all so close to each other, while the mole fraction only changes from 0 to 0.02, shows that we already on the right track precision wise.

This immediately leads to the question of where this project could be going in the future. We should definitely be trying to find the correct correlation for concentrations below 10 percent. The obvious way to do this is to improve the part of the setup that acquires the data, but there are actually also other ways that this can be done.

- A good way to make the setup preciser would be to try and lower the noise. This can be done by enlarging the signal gained, but than filtering out the parts that aren't needed (we used a DC-filter). Another good way to lower the noise would be to try to do this mathematically. The problem that we found however was the fact that, once doing it mathematically, you also lose some data that you sometimes do want to keep.
- Another good way would be to try and see if we can enlarge the oscillation itself. As we have told in the beginning, we had chosen for a drop of 3 *ms*. We had done this because

this seemed like the safest option. For drops which were much lower, the oscillation sometimes wouldn't even be there, because the droplet was just to viscous. For drops which were much higher, we noticed that, if the droplet had a high density, there was always a big chance that it would just fall out entirely. So we just chose a number that worked and stuck with it, but to improve the setup, this might be something to take a look at.

- Something that we could also try, since we have ignored it from the beginning, is the peak-splitting. We could use math to combine the different peaks to one. This might solve our precision problem as well.

- A last place that we could look for improving the setup would be to try and see if changing the voltage we send to the actuators gives any improvements. This voltage after all determines how elliptical the droplet will be, which influences the oscillation.

5.4 The colloidal glass transition

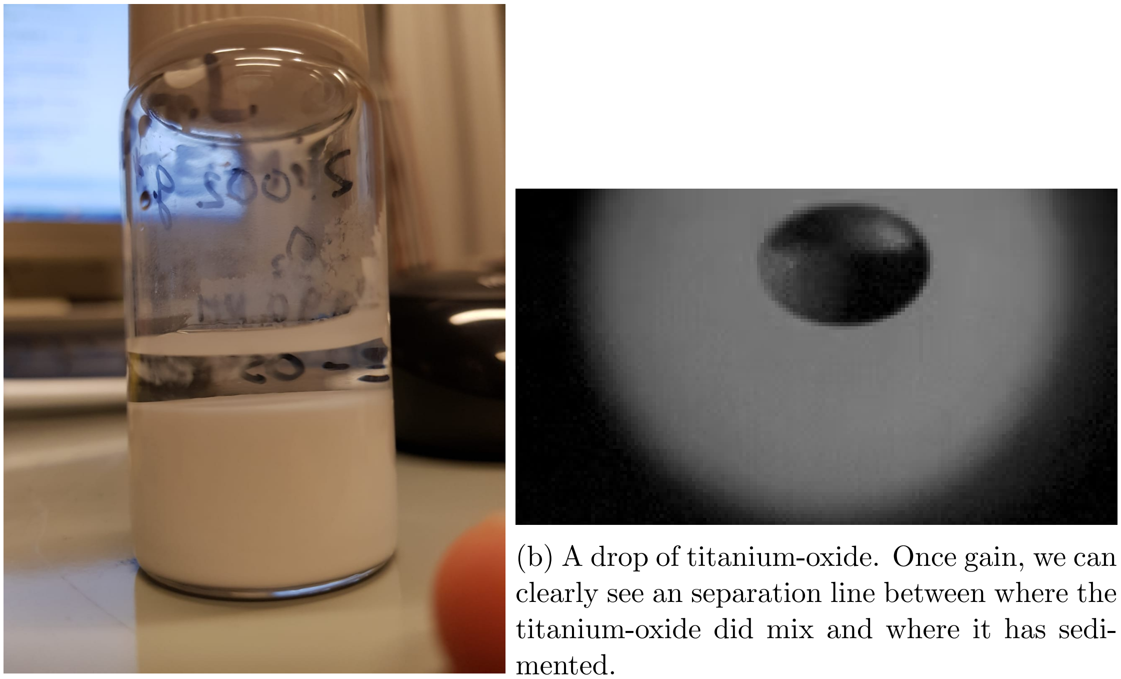
In figure 12 you can see 2 different ways to plot the viscosity of PMMA vs TiO2. All of these lines appear to posses the same exponential characteristics (time and size give a mirrored exponential, but they still look alike). This means that for both the PMMA and for the TiO2 we were able to see the colloidal glass transition happen.

For the TiO2, we didn't really know whether or not to expect this. That's because the TiO2 that we have doesn't really want to be solved in demi-water, neither in the container, nor in the droplet (see figure 13a). However, as you can see, the part of the droplet that's still viscous still gave us the oscillation that we were looking for. So this shows that both TiO2 and PMMA have the same structural design once dissolved in demi-water, since they both end up going through the colloidal glass transition.

There are two places where this experiment can go to in the future:

1. Right now, we have only looked at 2 different colloids. However, when making a line for example, it's always said that you need at least 3 different points. In our case this means that, if we really want to know if our setup can always find the colloidal glass transition, we would need to do this for more colloids. It would also be interesting to see if, by enlarging the fall of the droplet, we might be able to measure higher viscosity's for the colloids that we have already look at (PMMA and TiO2)

2. The other way we could go is by no longer looking at just colloids, but start looking at mixtures of colloids with polymers. Adding polymers should change the way that the viscosity changes in time [14] and it would be interesting to find out if we could see this for ourselves.



(a) Titanium-Oxide (TiO2) in it's original container. As you can see, it's doesn't want to be mixed within all of the demi-Water, so we get a clear separation line.

(b) A drop of titanium-oxide. Once gain, we can clearly see an separation line between where the titanium-oxide did mix and where it has sedimented.

Figure 13: Two different pictures of the Titanium-Oxide not wanting to solve properly in the demi-water. In both cases you can see that the TiO2 starts to sediment.

6 Conclusion

In conclusion we can say that acoustic levitation has become a good method to calculate the viscosity of liquids without it touching other surfaces. With our methods we are always able to find the correct trend that we should find, but also values that were close to the theoretical values. Improvement can still be gained however, especially in the temperature-experiment, since the errors we had were so large that we couldn't even plot them. For the "volume fraction vs viscosity"-experiment we should really try to see if we can also get the correct linear trend for lower volume-fractions, though at least all of the values are already in the correct ballpark. In our last experiment we saw that the viscosity of both our colloidal droplets did what it should have done, which was exponentially rise through time and exponentially decrease when plotting it vs the size of the droplet. This meant that we were indeed able to see the colloidal glass transition happen.

,

References

[1] Nobel Media. *Arthur Ashkin Facts.*

[2] R.R. WhyMark. Acoustic field positioning for containerless processing. *Ultrasonics*, November 1975.

[3] J. Kramer, A. Kilzer, and M. Petermann. Simultaneous measurement of surface tension and viscosity using freely decaying oscillations of acoustically levitated droplets. *Review of Scientific Instruments*, January 2018.

[4] T. Satogata. *Oscillators, Resonances, and Lorentzians*, January 2013.

[5] Antoine Weis. *Resonance Lineshapes of a Driven Damped Harmonic Oscillator*, March 2011.

[6] H. Lamb. *Hydrodynamics.* Cambridge University Press., fourth edition, 1932.

[7] E.C. Weeks. An introduction to the colloidal glass transition. *Department of Physics, Emory University, Atlanta*, November 2016.

[8] G.L.Hunter and E.C. Weeks. The physics of the colloidal glass transition. *Rep. Prog. Phys*, June 2011.

[9] A. Marzo, A. Barnes, and B.W. Drinkwater. *TinyLev: A multi-emitter single-axis acoustic levitator*, July 2017.

[10] I. Egry, H Griffard, and S schneider. The oscillating drop technique revisited. *Measurement Science and Technology*, February 2005.

[11] Manuchehr Dlzechl and Ekkehard Marschal. Viscosity of some binary and ternary liquid mixtures. *Journal of chemical engineering data*, July 1982.

[12] Nikos G. Tsierkezos and Ioanna E. Molinou. Thermodynamic properties of water + ethylene glycol at 283.15, 293.15, 303.15, and 313.15 k. *Journal of chemical engineering data*, August 1998.

[13] Milo Collaris. Remote viscosity measurement of acoustically levitated colloidal droplets. *Bachelor Thesis University Utrecht*, June 2018.

[14] W.C.K. Poon. The physics of a model colloid–polymer mixture. *Journal of Physics: Condensed Matter*, August 2002.

A Fitting an ellipse to the edge of the droplet

```
import numpy as np
from skimage import feature
from skimage import io
import cv2
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
from matplotlib.patches import Ellipse
from numpy.linalg import eig, inv
from scipy import ndimage
import os

”””
The steps that we do in this file are as followed:
– We start by importing the image, so that we can indeed fit an ellipse to
it.
– We than make sure that the middle of the droplet is also indeed black,
since otherwise the programm cannot fit to the edge of the ellipse
correctly.
– We than find the edge of the droplet.
– We now fit an ellipse to this edge and than calculate the parameters we
need (See the list below for which parameters this are).
”””

alist = []
aLength = []
bLength = []
centerlist = []
xLengthList = []
yLengthList = []
Anglelist = []
OppList = []
VolList = []

R = np.arange(0, 2*np.pi,0.01)
for i in range(24):
        ###Import the image
        image =cv2.imread(”Fileplace.png”,2)[150:400,600:1000]

        ###Here we make the middle of the image black (if we don’t do this,
        there will a white spot here, which is there because of the
        brightness
        of the led)
        image[120:170,170:230] = 12
```

```
###Find the edge of the droplet and there position
edges = cv2.Canny(image, 90, 170)
indices = np.where(edges != [0])
plt.imshow(image)

###Fitting an ellipse to the edge of the droplet
def fitEllipse(x,y):
        x = x[:,np.newaxis]
        y = y[:,np.newaxis]
        D =  np.hstack((x*x, x*y, y*y, x, y, np.ones_like(x)))
        S = np.dot(D.T,D)
        C = np.zeros([6,6])
        C[0,2] = C[2,0] = 2; C[1,1] = -1
        E, V =  eig(np.dot(inv(S), C))
        n = np.argmax(np.abs(E))
        a = V[:,n]
        return a

###Calculating the center of the ellipse
def ellipse_center(a):
        b,c,d,f,g,a = a[1]/2, a[2], a[3]/2, a[4]/2, a[5], a[0]
        num = b*b-a*c
        x0=(c*d-b*f)/num
        y0=(a*f-b*d)/num
        return np.array([x0,y0])

###Calculating the length of the ellipse in the direction of both
the x- and y-axis
def ellipse_axis_length( a ):
        b,c,d,f,g,a = a[1]/2, a[2], a[3]/2, a[4]/2, a[5], a[0]
        up = 2*(a*f*f+c*d*d+g*b*b-2*b*d*f-a*c*g)
        down1=(b*b-a*c)*((c-a)*np.sqrt(1+4*b*b/((a-c)*(a-c)))-(c+a))
        down2=(b*b-a*c)*((a-c)*np.sqrt(1+4*b*b/((a-c)*(a-c)))-(c+a))
        res1=np.sqrt(up/down1)
        res2=np.sqrt(up/down2)
        return np.array([res1, res2])

###Calculating the angle of the a and b of the ellipse compared to
the x- and y-axis
def ellipse_angle_of_rotation2( a ):
        b,c,d,f,g,a = [a[1]/2, a[2], a[3]/2, a[4]/2, a[5], a[0]]
        if b == 0:
                if abs(a) > abs(c):
                        return 0
```

```
            else:
                return np.pi/2
        else:
            if abs(a) > abs(c):
                return np.arctan(2*b/(a-c))/2
            else:
                return np.pi/2 + np.arctan(2*b/(a-c))/2


    ###Writing all of the parameters to there lists
    a = fitEllipse(indices[0],indices[1])
    alist.append( fitEllipse(indices[0],indices[1]))
    centerlist.append( ellipse_center(a))
    center = ellipse_center(a)
    yLengthList.append( ellipse_axis_length(a)[0])
    yLength = ellipse_axis_length(a)[0]
    xLengthList.append( ellipse_axis_length(a)[1])
    xLength = ellipse_axis_length(a)[1]
    Anglelist.append(ellipse_angle_of_rotation2(a))
    Angle = ellipse_angle_of_rotation2(a)
    print(i)

    aLength.append((yLength*np.cos(np.pi/2)*np.sin(Angle) +
      xLength*np.sin(np.pi/2)*np.cos(Angle))*0.0146)
    bLength.append((yLength*np.cos(0)*np.cos(Angle) -
      xLength*np.sin(0)*np.sin(Angle))*0.0146)
    VolList.append((4/3)*np.pi*aLength[i]**2*bLength[i])

    ###If we want, we can use the data below to plot the ellipse we
    just fitted so that we can see that our code actually works

    yy1 = center[0] + yLength*np.cos(R)*np.cos(Angle) -
      xLength*np.sin(R)*np.sin(Angle)
    xx1 = center[1] + yLength*np.cos(R)*np.sin(Angle) +
      xLength*np.sin(R)*np.cos(Angle)
    plt.plot(xx1,yy1, color = ’red’)
    plt.xlabel(”pixels”)
    plt.ylabel(”pixels”)
    print(aLength,bLength)
    #plt.show()


###Now we save the values that we want to save in their corresponding
files using np.savetxt(”Storage-Place”, file-name, fmt=’%1.6f’)
```

B Fitting a Cauchy-Lorentz Curve to the frequency spectrum

```
from lmfit.models import LorentzianModel
from lmfit.models import ConstantModel
import numpy as np
import matplotlib.pyplot as plt



###We start by writing a function that those the fourier−transform of
the data that the diode finds
def frequency(data):
        freq = np.fft.fftfreq(data.size, d=spf)
        spect = np.abs(np.fft.fft(data))
        angle = np.angle(np.fft.fft(data))
        return freq,spect,angle

fps = 500000
spf = 1/fps

### Here we import the data
DiodeDataRaw = np.loadtxt(”File−Place.txt”)



###Here we plot the data of the diode, so that we can see if the
measurement was a succes
plt.plot(DiodeDataRaw)
plt.xlabel(”Measurement”)
plt.ylabel(”Voltage (mV)”)
plt.figure()

###Here we have our function to the fourier transform
freqdata, spectdata, angledata = frequency(DiodeDataRaw)
freqdataR = np.array(freqdata)
spectdataR = np.array(spectdata)

”””
Everything below this point is there to make a fit to the data.
–We start by finding the correct points where we tell the fit to search
a peak.
–We can than call LorentzianModel(), which uses Scipy to find a fit to
the data.
–We can than plot it to check if it has indeed worked the way that it
should.
```

```
”””

begin = 120
middlepoint2 = 140
middlepoint3 = 160
middlepoint4 = 180
middlepoint5 = 200
end = 250

Lorentzian1 = LorentzianModel(prefix = 'L1_')
parsL1 = Lorentzian1.guess(spectdataR[begin:middlepoint2],
    x=freqdataR[begin:middlepoint2])

parsL1['L1_center'].set(67, min=60, max=70)

Lorentzian2 = LorentzianModel(prefix = 'L2_')
parsL1.update(Lorentzian2.guess(spectdataR[middlepoint3:middlepoint4],
    x=freqdataR[middlepoint3:middlepoint4]))

parsL1['L2_center'].set(85, min=80, max=90)
parsL1['L2_sigma'].set(0.1, min=0.01, max=20)

Lorentzian3 = LorentzianModel(prefix = 'L3_')
parsL1.update(Lorentzian3.guess(spectdataR[middlepoint5:end],
    x=freqdataR[middlepoint5:end]))

parsL1['L3_center'].set(115, min=105, max=120)
parsL1['L3_sigma'].set(0.1, min=0.01, max=20)

Constant = ConstantModel()
parsL1.update(Constant.guess(spectdataR[begin:end],x=freqdataR[begin:end]))

modL1 = Lorentzian1 + Lorentzian3 + Constant
outL1 = modL1.fit(spectdataR[begin:end], parsL1, x=freqdataR[begin:end])
print(outL1.fit_report(min_correl=0.25)) #Gives all of the parameter values
plt.plot(freqdata[begin:end], outL1.best_fit, label = 'Fit')
plt.plot(freqdata[begin:end],spectdata[begin:end], label = 'data')
plt.xlabel(”Freqency (Hz)”)
plt.ylabel(”Peak Hight”)
plt.legend()
plt.show()
```

C Controlling the Data Acquisition(DAQ) Device

```
import sys
import os
import time
import visa
import numpy as np
import datetime
from pypylon import pylon
import matplotlib.pyplot as plt

”””
The code below is used to control the DAQ and tell him when and
how to measure:
– We start by having the computer search the DAQ and connect with it.
– We than give the DAQ it’s used parameters. These are the channel’s
which he needs to look at, the rate at which he acquires points,
the amount of points he needs to acquire and the range of the
incoming voltages (the last one is very important, since it decides how
big the bits are (how many voltage per 16 bits))
– We than tell our DAQ to only start measuring once it’s triggered
by the arduino (which it does each time the oscillation has begun)
– After the DAQ is done measuring, we calculate the measured voltages
by rewriting the incoming bits.
”””


###We have our computer search for the camera and connect with it. We
also give the camera the correct values so that it works once triggered
and etc.
os.makedirs(”C:\\Users\\linx\\My Documents\\StijnHelsloot
    \\Basler Pictures/PMMA/Droplet1”, exist_ok = True)
camera = pylon.InstantCamera(pylon.TlFactory.GetInstance().
    CreateFirstDevice())
camera.RegisterConfiguration(pylon.ConfigurationEventHandler(),
pylon.RegistrationMode_ReplaceAll, pylon.Cleanup_Delete)
camera.Open()
camera.AcquisitionMode.SetValue(’Continuous’)
camera.MaxNumBuffer = 15
camera.TriggerSelector.SetValue(’FrameStart’)
camera.TriggerMode.SetValue(’On’)
camera.TriggerSource.SetValue(’Line1’)

###We have our computer search for the DAQ and connect with it
rm = visa.ResourceManager()
device = rm.open_resource(’USB0::0x0957::0x1518::TW56100007::0::INSTR’)
```

```
channel_Temperature = 112
channel_Diode = 116

###We have our machine do one warm-up measurement, since the first
measurement always takes a little bit longer
print(device.query("MEAS:VOLT? (@112)"))

device.timeout = 500000

###We give our DAQ the correct parameters that it needs to use
device.write("ACQ:SRAT 150000")
print(device.query("ACQ:SRAT?"))
device.write("ACQ:POIN 700000")
print(device.query("ACQ:POIN?"))
device.write
("ROUT:SCAN (@112,116)")
print(device.query("ROUT:SCAN?"))
device.write("ROUT:CHAN:RANG 0.2, (@112,116)")
print(device.query("ROUT:CHAN:RANG? (@112,116)"))

numberOfImagesToGrab = 100
camera.StartGrabbingMax(numberOfImagesToGrab)

def Measurement(x):
    n = 0
    while n<x:
        TempList = []
        DiodeList = []

        # We tell it to only measure once it is given a trigger
        device.write("TRIG:SOUR EXTA")
        print(device.query("TRIG:SOUR?"))
        device.write("TRIG:ATR:COND BLOW")
        device.write("TRIG:ATR:LTHR 3")
        device.write("TRIG:TYP POST")
        timenw = datetime.datetime.now()

    ###We start by having our camera take a picture the moment the
    shutter is in front of the laser
        grabResult = camera.RetrieveResult(25000,
            pylon.TimeoutHandling_ThrowException)

        if grabResult.GrabSucceeded():
            # Access the image data.
        img = grabResult.Array
        plt.imshow(img)
        plt.savefig(”C:\\Users\\linx\\My Documents\\
                StijnHelsloot\\Basler Pictures\\PMMA\\
                     Droplet1\\Meas”+str(timenw.strftime(
                        ’%Y-%m-%d-%H-%M-%S’))+”.png”)
grabResult.Release()
print(”Done Picture”)

device.write(”DIG”)

###We make our computer fall ”asleep”, so that our DAQ
gets enough time to do the measurement
time.sleep(180)

###We calculate the incoming voltages by rewriting the
inncoming bits
meas = device.query_binary_values(”WAV:DATA?”,
    datatype = ’h’, is_big_endian = False)
measDiode = []
measTemp = []


print(len(meas))
for i in range(700000):
        measDiode.append(meas[1 + 2*i])
        measTemp.append(meas[0 + 2*i])

for i in range(700000):
        DiodeList.append((2*measDiode[i])/(2**16)*0.2)
        TempList.append((2*measTemp[i])/(2**16)*200)

###We can now save the data into the correct folder.
DiodeMeasurement = np.array(DiodeList)
TempMeasurement = np.array(TempList)
os.makedirs(”C:\\Users\\linx\\My Documents\\StijnHelsloot
    \\DataDAQ/PMMA/Droplet1”, exist_ok = True)
os.makedirs(”C:\\Users\\linx\\My Documents\\StijnHelsloot
    \\DataDAQ\\PMMA\\Droplet1/”
        +str(timenw.strftime(’%Y-%m-%d-%H-%M-%S’)),
            exist_ok = True)
np.savetxt(”C:\\Users\\linx\\My Documents\\StijnHelsloot
    \\DataDAQ\\PMMA\\Droplet1\\”
        +str(timenw.strftime(’%Y-%m-%d-%H-%M-%S’))+ ”\\
            TempData.txt”, TempMeasurement, fmt = ’%1.6f’)
```

```
                    np.savetxt(”C:\\Users\\linx\\My Documents\\StijnHelsloot
                        \\DataDAQ\\PMMA\\Droplet1\\”
                            +str(timenw.strftime('%Y−%m−%d−%H−%M−%S'))+ ”\\
                                DiodeData.txt”, DiodeMeasurement,fmt = '%1.6f')
                    n = n+1
                    print(”Measurement Done”)

###Calling the function 8 times means that the measurement is done 8
times in total
Measurement(8)
```