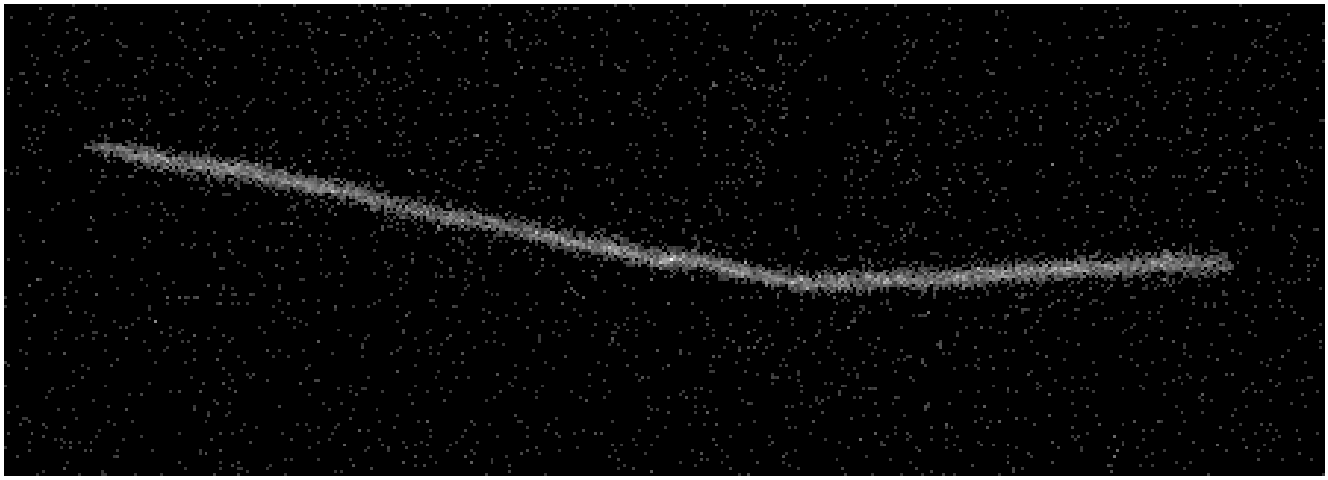
Universiteit Utrecht

Opleiding Natuur- en Sterrenkunde

Relation between surface charge and the flow rate within microfluidic channels

Bachelor Thesis

Lorenzo Sierra Perez



Supervisors:

Dr. Sanli Faez

Debye Institute for Nanomaterials Science

Drs. Dashdeleg Baasanjav

Debye Institute for Nanomaterials Science

13 June, 2018

Abstract

I created a highly adaptable, easy-to-make microfluidic cell, in which one can change the flow rate via a pressure-driven flow. By using electro-osmotic flow, which relates to the surface charge of a channel, I looked at the relation between surface charge of the channel and the flow rate. The literature suggests that there is a causal relation. I created a proper method to research this, and found that there is indeed a (mostly) linear relation between the two.

Contents

1 Introduction

Electrokinetic phenomena can be loosely defined as all those phenomena involving tangential fluid motion adjacent to a charges surface. Owing to numerous advantages, electrokinetic flow is often utilised in microfluidic devices to transport buffer solutions and to manipulate sample solutes.[1, 2]. Examples include microfluidic pumping [3], flow control [4, 5] mixing and reacting reagents [6, 7], injecting or dispensing samples [8, 9], capillary electrophoresis-based chemical separations [8, 10, 11], chromatograph [12, 13], etc. Together with electrophoresis, electro-osmosis is one of the most useful of these phenomena, and also one of the earliest discovered; both electrophoresis and electro-osmosis have been observed for the first time in 1809, by Russian professor Ferdinand Frederic Reuss [14].

Electro-osmosis represents the movement, due to an applied electric field, of an electrolyte solution relative to a stationary charged surface (i.e., a capillary tube or porous media) [15]. This surface charge influences the velocity of the electro-osmotic flow. Literature suggests [16, 17] that a flow along a surface changes the interfacial chemistry -the latter suggests that the surface charge in the presence of a flow also depends on the distance from the inlet-, and as such, the surface charge. In this thesis, I will research this further, and my research question is then as follows: How does the surface charge of the walls of a microfluidic channel depend on the flow rate in the channel?

I will firstly handle the theory behind both the electro-osmotic flow and a pressure driven flow, and discuss the relation between electro-osmotic velocity and surface charge, in Sec. 2. I then discuss the process of designing my flow cell, and describe my setup and measurement method in Sec. 3. After this, in Sec. 4 I discuss the data analysis and show the results. Finally, I draw my conclusions in Sec. 6.

2 Theory

In this section I unpack the different kinds of effects I apply on the liquid.

I introduce two different kind of flows in my channel, namely an electro-osmotic flow and a pressure-driven flow. As will be discussed in Sec. 3.1, I designed the microfluidic cell such that these two flows are perpendicular to each other; the pressure-driven flow acts along the length of the channel, and the electro-osmotic flow acts along the width of the channel. As the two do not influence each other, one can split the total displacement of the fluid into two.

2.1 Particle displacement via pressure-driven flow

To change the flow rate without changing the electrical field, I introduce a pressure-driven flow along the channel.

In general, the motion of incompressible fluids is described by the following Navier-Stokes equations [18]:

$$\frac{\delta \mathbf{u}}{\delta t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \nabla^2 \mathbf{u} = -\frac{1}{\rho_0}\nabla p + \mathbf{g} \tag{1}$$

where

- $\mathbf{u}$ is the flow velocity,
- t is time
- ν is the kinematic viscosity
- ρ_0 is the density
- p is the pressure
- $\mathbf{g}$ represents body accelerations acting on the fluid (e.g. gravity)

As my channel is a long channel with a rectangular cross-section, and as I will only be looking at the very middle of the width of the channel, I can approximate my channel as two infinitely long parallel plates, separated by a distance h. Furthermore, one can ignore any body accelerations acting on the fluid. Lastly, I regard the flow as stationary and fully developed. Then, the Navier-Stokes equations reduce to the following:

$$0 = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u_\mathrm{p}}{\partial y^2}\right) \tag{2a}$$

$$0 = -\frac{1}{\rho}\frac{\partial p}{\partial y}, \tag{2b}$$

where the pressure gradient is constant as the flow is stationary. As the velocity of the flow must be zero when either $y = h$ or $y = 0$, I get for u_p:

$$u_{\mathrm{p}}(y) = \frac{-\frac{dp}{dx}}{2\mu} y(h-y) \tag{3}$$

,

where μ is the dynamic viscosity. This is a plane Poiseuille flow. As I want to compare my velocities with other data [17] at different depths, it's important to know the shear stress. This is

$$\tau = \mu \frac{du_{\mathrm{p}}}{dy} \tag{4a}$$

$$= -\frac{dp}{dx}\left(\frac{h}{2} - y\right) \tag{4b}$$

The volume flow rate q per unit width of the channel is:

$$q_{\mathrm{p}} = \int_0^h u_{\mathrm{p}} \mathrm{d}y = \frac{-\frac{dp}{dx}}{2\mu} \int_0^h y(h-y)\mathrm{d}y = \frac{-\frac{dp}{dx} h^3}{12\mu}. \tag{5}$$

2.2 Particle displacement via electro-osmosis

Electro-osmosis is the motion of a liquid in a microchannel or capillary tube under the influence of an applied electrical field.

I show a simple example (seen in figure 1) and explain how this is analogous to the EO flow in my setup.

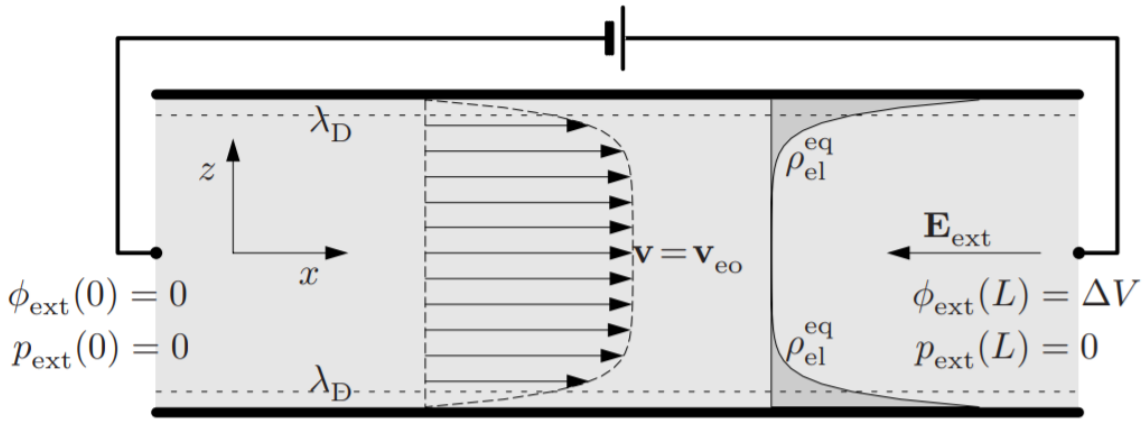


Figure 1: The velocity profile $\mathbf{v}$ (dashed line and arrows) and the negative Debye-layer charge density profile ρ_{el}^{eq} (dark gray and full line) in an ideal electro-osmotic (EO) flow inside a cylindrical channel of radius a and positively charged walls (thick horizontal lines).[19]

As depicted above, there are electrodes situated at each end of the channel. In the channel, charge separation at the walls has led to a Debye layer. If I apply a potential difference $\Delta V = \Delta\phi_{\mathrm{ext}}$ over the two electrodes, the resulting electrical field

$$\mathbf{E}_{\text{ext}} \equiv -\nabla\phi_{\text{ext}}. \tag{6}$$

I will follow Bruus' train of thought. [19] I first define an ideal EO flow as a flow that complies with the following four conditions:

i The ζ potential is constant along the wall

ii The electrical field is homogeneous

iii The flow is in a steady state i.e. the behaviour of the flow is unchanging in time

iv The Debye length is much smaller than the half-width a of the channel , so $\lambda_{\text{D}} \ll a$.

Similarly to the Poiseuille flow, I can approximate my channel with two, infinite parallel plates, as the width and length of the channel are much larger than the depth, and as I'm only interested in the behaviour at the very middle of my channel.

So let us put two infinitely long, positively charged plates parallel to the xy-frame at $z = h/2$ and $z = -h/2$. I apply the external electrical field in the negative x direction, so $\mathbf{E} = (-E, 0, 0)$.

Because of the symmetry of this "channel", I find the following structures:

$$\nabla\phi_{\text{ext}}(\mathbf{r}) = -\mathbf{E} = (E, 0, 0) \tag{7a}$$
$$\nabla p_{\text{ext}}(\mathbf{r}) = 0 \tag{7b}$$
$$\mathbf{v}(\mathbf{r}) = (v_x(z), 0, 0). \tag{7c}$$

Then, the only non-trivial component of the steady-state Navier-Stokes equations is the x-component, which is as follows:

$$0 = \nu\delta_z^2 v_x(z) + \left[\epsilon\delta_z^2\psi_{\text{eq}}(z)\right]E, \tag{8}$$

where ϵ is the dielectric constant. I re-write this equation as follows:

$$\delta_z^2\left[v_x(z) + \frac{\epsilon E}{\nu}\psi_{\text{eq}}(z)\right] = 0 \tag{9}$$

Now, I use the boundary conditions

$$v_x\left(\pm\frac{h}{2}\right) = 0 \tag{10}$$

so I get the following solution:

$$v_x(z) = \left[\zeta - \psi_{\text{eq}}(z)\right]\frac{\epsilon E}{\nu} \tag{11}$$

The parallel-plate two-wall potential ψ_{eq} is given by the following equation:

$$\psi_{\text{eq}}(z) = \zeta \cdot \frac{\cosh\left(\frac{z}{\lambda_{\text{D}}}\right)}{\cosh\left(\frac{h}{2\lambda_{\text{D}}}\right)} \qquad (-\frac{h}{2} < z < \frac{h}{2}). \tag{12}$$

Combining this with Eq. 11, we get

$$v_x(z) = \left[1 - \frac{\cosh\left(\frac{z}{\lambda_{\text{D}}}\right)}{\cosh\left(\frac{h}{2\lambda_{\text{D}}}\right)}\right] v_{\text{eo}}, \tag{13}$$

where I have introduced the EO velocity v_{eo} defined by the Helmholtz-Smoluchowski relation as

$$v_{\text{eo}} \equiv \frac{\epsilon\zeta}{\nu} E. \tag{14}$$

I define the EO mobility μ_{eo} as

$$\mu_{\text{eo}} \equiv \frac{v_{\text{eo}}}{E} = \frac{\epsilon\zeta}{\nu}. \tag{15}$$

For an ideal EO flow, I obtain the simple velocity profile

$$\mathbf{v}(\mathbf{r}) \approx v_{\text{eo}}\mathbf{e}_{\text{x}} = -\mu_{\text{eo}} * \mathbf{E}, \quad \text{for} \quad \lambda_{\text{D}} \ll \frac{1}{2}h. \tag{16}$$

Now, it should be noted that the above is derived for electro-osmotic flow in an open microchannel, while I'm in reality working with a closed microchannel. However, the important thing to take away from the above is that the velocity is linear with ζ, which it is for a closed channel as well [15].

2.3 Surface Charge

The charge of a liquid is calculated by the following equation:

$$Q_{\text{liq}} = \int_0^\infty dz \rho_{\text{el}}(z) = \int_0^\infty dz \Big[-\frac{\epsilon\zeta}{\lambda_{\text{D}}^2} e^{-\frac{z}{\lambda_{\text{D}}}} \Big] = -\frac{\epsilon}{\lambda_{\text{D}}}\zeta. \tag{17}$$

As the surface charge will be the opposite of the charge of the liquid, the surface charge must be

$$Q_{\text{surf}} = -\epsilon E = -\epsilon\delta_{\text{z}}\psi(0) \tag{18a}$$

$$= -Q_{\text{liq}} = \frac{\epsilon}{\lambda_{\text{D}}}\zeta. \tag{18b}$$

So then I can write the EO velocity as

$$v_{\text{eo}} = E \cdot Q_{\text{surf}} \cdot \frac{\lambda_{\text{D}}}{\nu} \tag{19a}$$

$$\propto Q_{\text{surf}} \tag{19b}$$

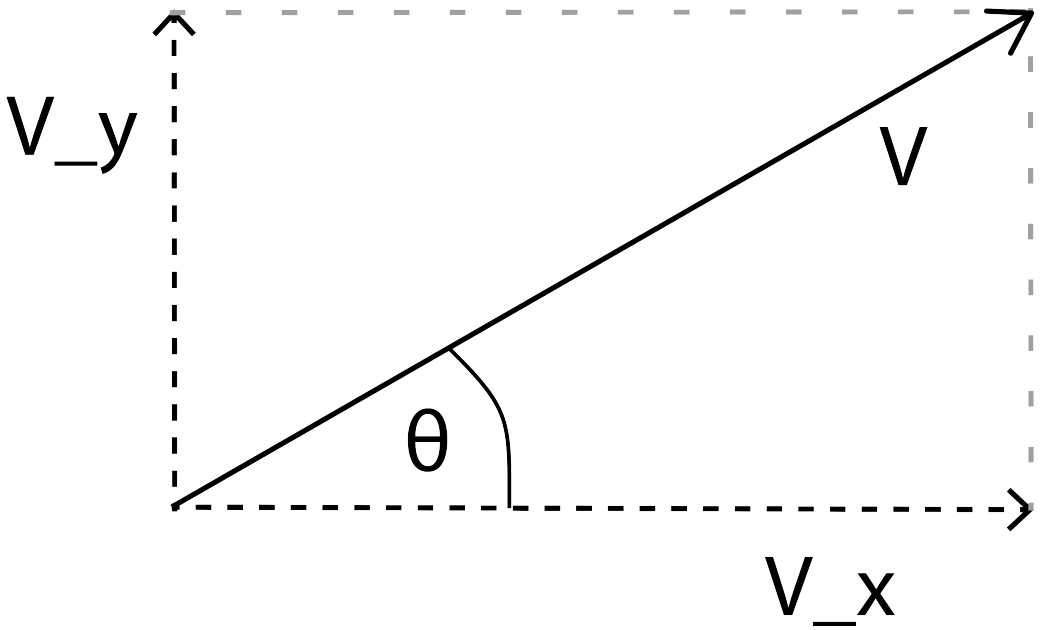


Figure 2: A simple schematic to show the decomposition of vectors into two, linearly independent vectors.

So, as I keep the electric field constant, if the EO velocity changes, the surface charge must have changed as well. If a flow would alter the surface charge, as the literature suggests [16], then there would be a noticeable change in the EO velocity.

2.4 Finding electro-osmotic velocity

The above means that I must find the EO velocity from my data to see if the surface charge is changing. By design of the channel, the electro-osmotic flow and pressure-driven flow are perpendicular to each other, so I can decompose the actual velocity vector in my channel into the electro-osmotic velocity vector and the pressure-driven velocity vector, as seen in Fig. 2.

If I know the length of the vector in the x-direction and the time it took to reach that length, then I can find the velocity in the x-direction by simply dividing the two. If I know the slope of the actual vector as well, then multiplying the slope with the length in the x-direction will give me the length of the vector in the y-direction. If I know the time it took to reach that length, then I can calculate the velocity in the y-direction by simply dividing the length in the y-direction by the time it took. This is how I find the pressure-driven velocity, and the electro-osmotic velocity.

In my experiment, I multiply the slope by half the length in the x-direction to find the amplitude, as the full length corresponds with one complete oscillation, and a peak-to-peak is done in half the time it takes for one full oscillation (see Sec. 4.1).

3 Experimental Method

I split my experimental setup in two distinct parts: the design of the flow cells used in the experiment, and the setup itself. I will describe both, starting with the flow cell design. After that, I'll discuss the exact steps taken to reach my measurements.

3.1 Flow Cell Design

To accurately measure/track the particles, I needed to create a flow cell fit for this purpose. This means that I need to be able to create a pressure-driven flow, an electrical field tangential to this flow, and illuminate the channel such that I can use a high enough exposure time for the camera to register an entire oscillation of a particle in a single frame, but as little background noise as possible

Ideally, I want to impose some other conditions on my flow cell, namely that it has to be able to be cleaned and re-used, that it should be easy to create, and that the design can be easily changed depending on specific needs/different experiments.

To get an idea of a typical sample, see Fig. 3 and Fig. 6. In Fig. 3 we see a picture of a finished sample, where I accentuated the hard-to-see parts of the sample. We can see the optical fibre, responsible for the illumination, entering the channel from the left. The copper triangles are serving as electrodes, and are placed perpendicular to the channel. In Fig. 6 we see a 3D schematic of my sample. The top layer and middle layer are made out of Polyethylene terephthalate (PET), and the bottom layer is made out of glass.

The aforementioned conditions impose certain limitations: to create a sample that is easily re-made with a changed design, you'd want either something that is easily made from scratch, or a flow cell where you could switch parts in and out. As the latter should pose problems concerning leakage, I focused on the former, and looked at possibilities of using a vinyl cutter to create the flow cell. Here, one could create their design in graphic design software, and then upload their design to the vinyl cutter, which would then cut out their design.

This idea has been expanded upon and used by Dr. Fernando Ontiveros[20] to build microfluidic flow cells constructed out of multiple layers of PET laminating plastic, where the layers stick to each other by use of the ethylene-vinyl acetate (EVA) coating, resulting in cells such as seen in figure 4. This layer-based method of fabrication made for microfluidic channels that were made by exclusive use of consumer-grade components and equipment, where the fabrication time is in the order of minutes, and the product is easily re-made when in need for a different channel.

The problem with the Ontiveros-cell is that they're entirely made of plastic, which results in too much scattering of the laser light to see the channel. Another problem is that all layers were glued to each-other at once, leaving no time to properly glue either electrodes or an optical fiber. Furthermore, a problem with the use of laminating plastic in general is that one could have spilling of the EVA layer into the channel, or the layer retracting under the heat.

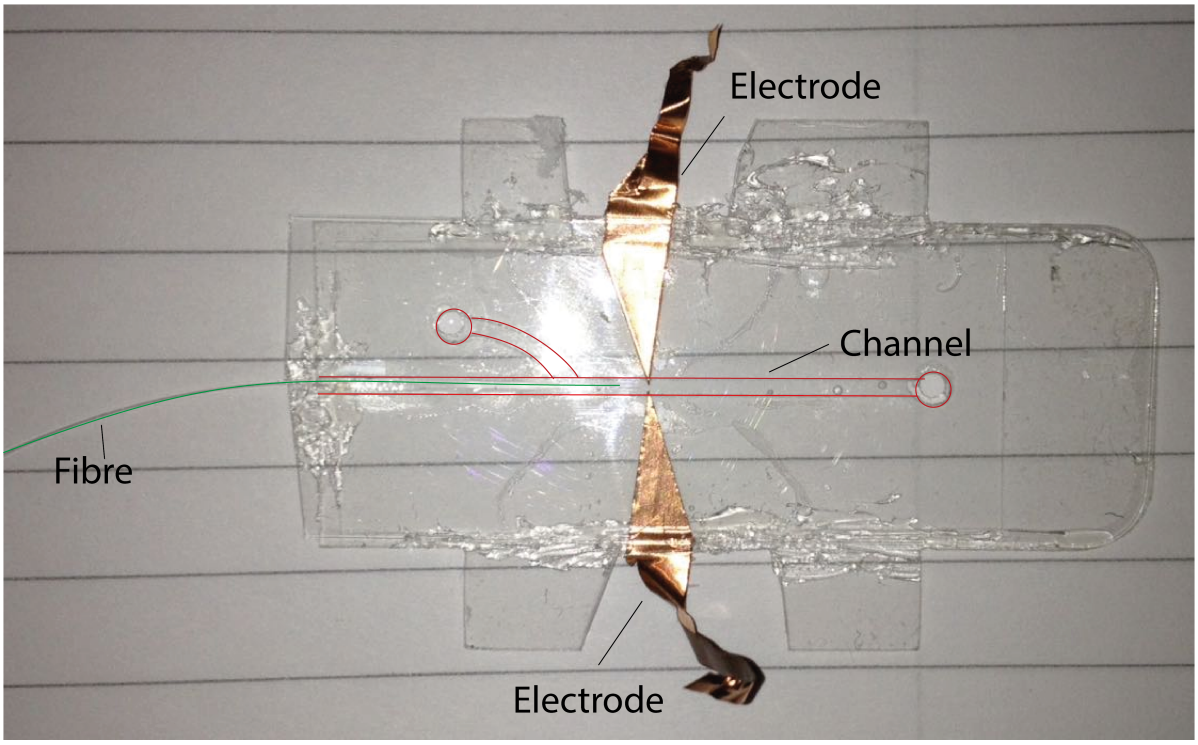


Figure 3: This is how a finished sample looks like. The copper triangles are entering the channel, the sample is completely sealed, and one can see the optical fibre entering the channel from the left side.

I kept the layered design that Ontiveros used, but changed the bottom layer to a glass coverslip, and started using optically clear double-sided acrylic adhesive tape with a thickness of 0.050mm to glue the three layers to each-other.

To create an electrical field tangentially to the direction of pressure-driven flow, I tried using a middle layer made out of PET with an Indium-Tin-Oxide (ITO) coating on one side. The entire channel would then be made out of this material, with a narrowing of the channel at the point of interest. After attaching this layer to the top layer, the sides would be cut off (see dashed red lines in Fig. 5), resulting in two electrodes, one at either side of the channel. The outer sides extend, giving the sample a cross-shape, making it easier to attach the electrodes to the sample, and making alignment in my sample holder easier (see Sec. 3.2).

The idea of using the ITO was that I wouldn't have to use glue apart from the optical double-sided tape, and the field lines at the narrowing of the channel would be tangential to the channel. Unfortunately this wasn't the case when tested, and instead the electro-osmotic flow was along the length of the channel. Because of this, I had to think of a different way to introduce an electrical field tangential to the length of the channel. The flow cell did keep his cross-shape, as it proved very useful during alignment in the sample holder.

Instead, I now glue two triangles, made out of copper tape, placed on either side of the channel, to the glass cover slip on the bottom. To align the copper tape and the glass, I printed the preferred placement on a piece of paper, and put the glass slide on top of the print-

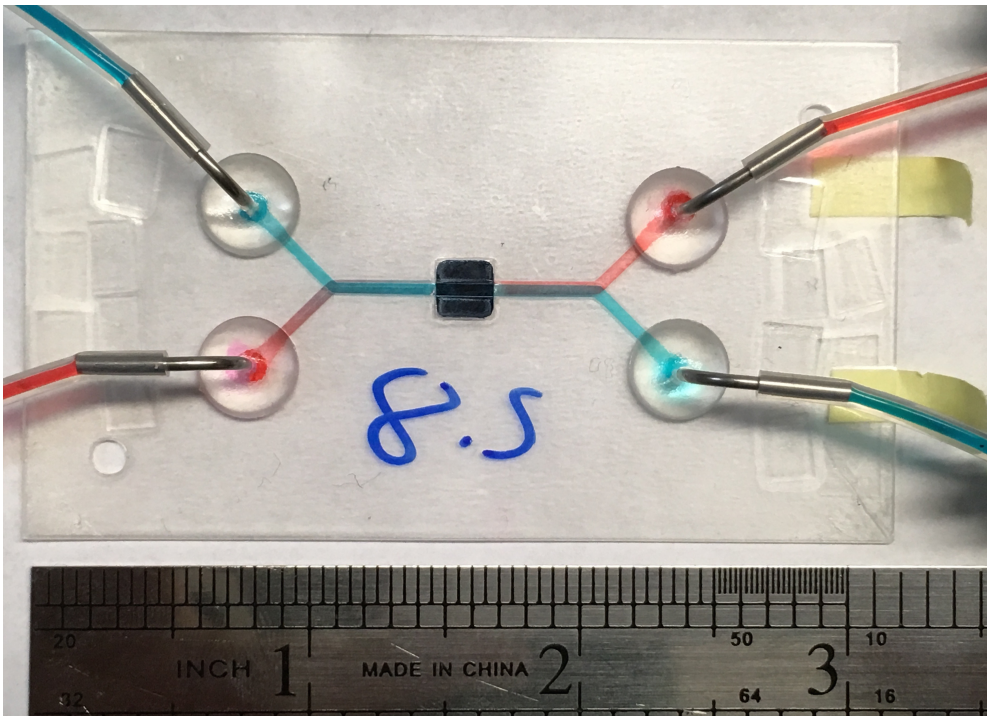


Figure 4: A microfluidic cell made with Ontiveros' method, built to study virus detection using nano membranes mounted on his microfluidic flow cells.

out, thus making alignment of the copper triangles easier. I also used two cameras to help me with the alignment. As the copper has a non-zero thickness, I needed to apply additional sealant to the sides to make the flow cell waterproof. I measured the final dimensions of the sample to be 70 mm long, 40 mm wide, and the channel is 1 mm wide, 24 mm long and 0.18 mm thick, all with an uncertainty of 0.005 mm.

As for the easiness of creating and re-creating these layers, the top layer is cut out with a vinyl cutter, and has two inlets. The middle layer and with that the layers of the optically clear double-sided tape are cut with a laser cutter, as together with the layers of tape and their respective coatings, the middle layer was too thick for the vinyl cutter. The principle of designing and cutting a layer stays the same, whether the cutting is by vinyl cutter of by laser cutter. Having these two important layers be freely designed and cut opens up design possibilities, and creating the three individual layers takes less than ten minutes total.

3.1.1 Illumination

Illumination is very important for my measurements, because my measurements consist of still frames with a high exposure time, thereby increasing any kind of background noise. The better the illumination becomes, the less exposure time needed for a good enough signal to see/track the particles clearly, which would then result in less background noise. Ideally, you'd like as little background noise as possible, so I spent a lot of time optimising the illumination.

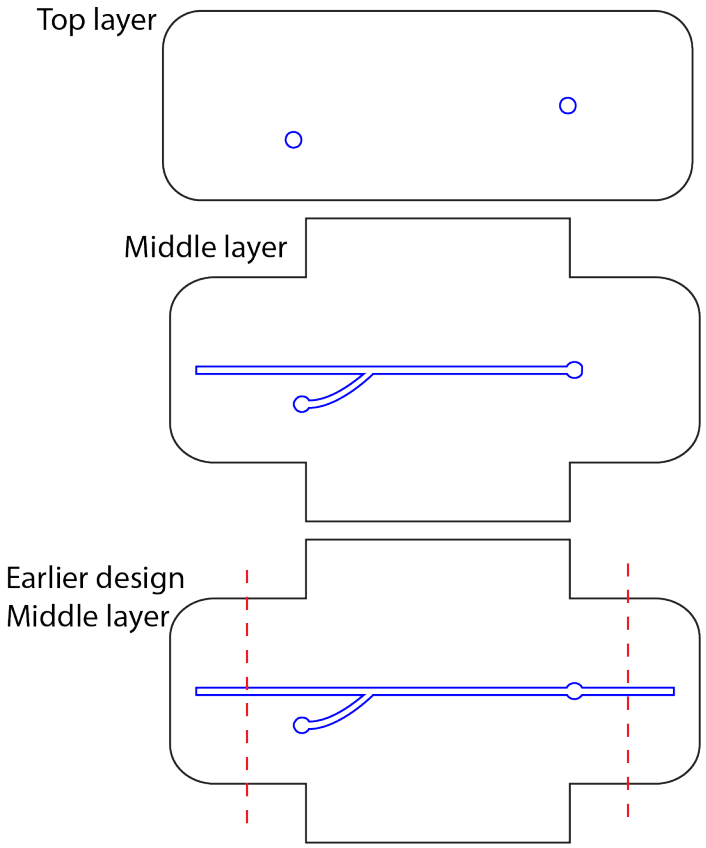


Figure 5: The designs for the top and the middle layer of my flow cell. The difference between the middle layer from the previous design and the middle layer from the new cell is that there is no need to separate the two sides of the flow cell, thus the only place where I need to cut is the entrance of the channel for inserting the optical fibre.

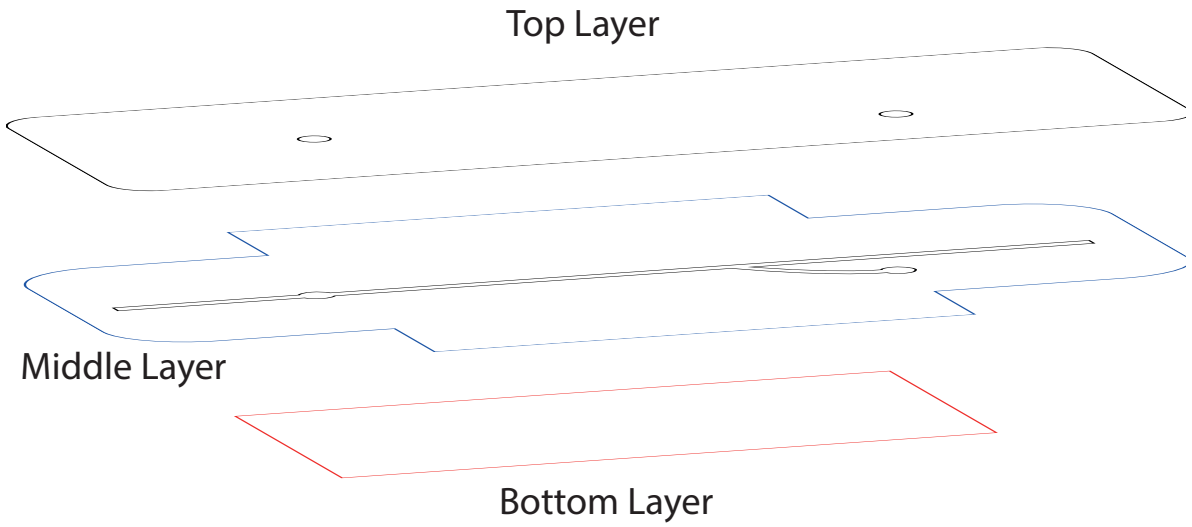


Figure 6: A schematic view of the different layers of my sample. The top layer is PET, the bottom is glass, and the middle layer is made of PET, and has a channel cut out of it. They are glued together by using an optically clear double-sided acrylic adhesive tape with a thickness of 0.050mm.

For illumination, I considered two options. The first one is to illuminate from outside the flow cell, shining a laser through the top layer into the channel. As the top layer of my flow cell is made of plastic, this resulted in a great deal of scattering. After changing to a glass top as well as a glass bottom layer, there still was enough scattering for me to consider illumination from inside the channel.

Illumination from inside the channel is done by carefully inserting an optical fibre inside the channel, close to the point of interest, and gluing it in place. I use a 50μm core multimode fibre to optimise the amount of laser light entering my sample, as it makes it easier to couple the laser light to the fibre.

So, to summarise, my final flow cell, seen schematically in Fig 5 is triple-layered: the top layer is made out of PET; the middle layer is also made out of PET, with a channel cut out; and the bottom layer is a glass cover slip with two copper electrodes. The three layers are attached to each other by 0.050 mm thick optically clear double-sided adhesive tape, and has been made waterproof by sealant. An optical fibre has entered the channel via an opening, and has been glued to the flow cell. The opening has been sealed.

3.2 Setup

In this section the setup used in this research will be discussed. In Sec. 3.2 a general overview of the setup is given, and in the sections after that I will discuss the parts of the setup piece by piece.

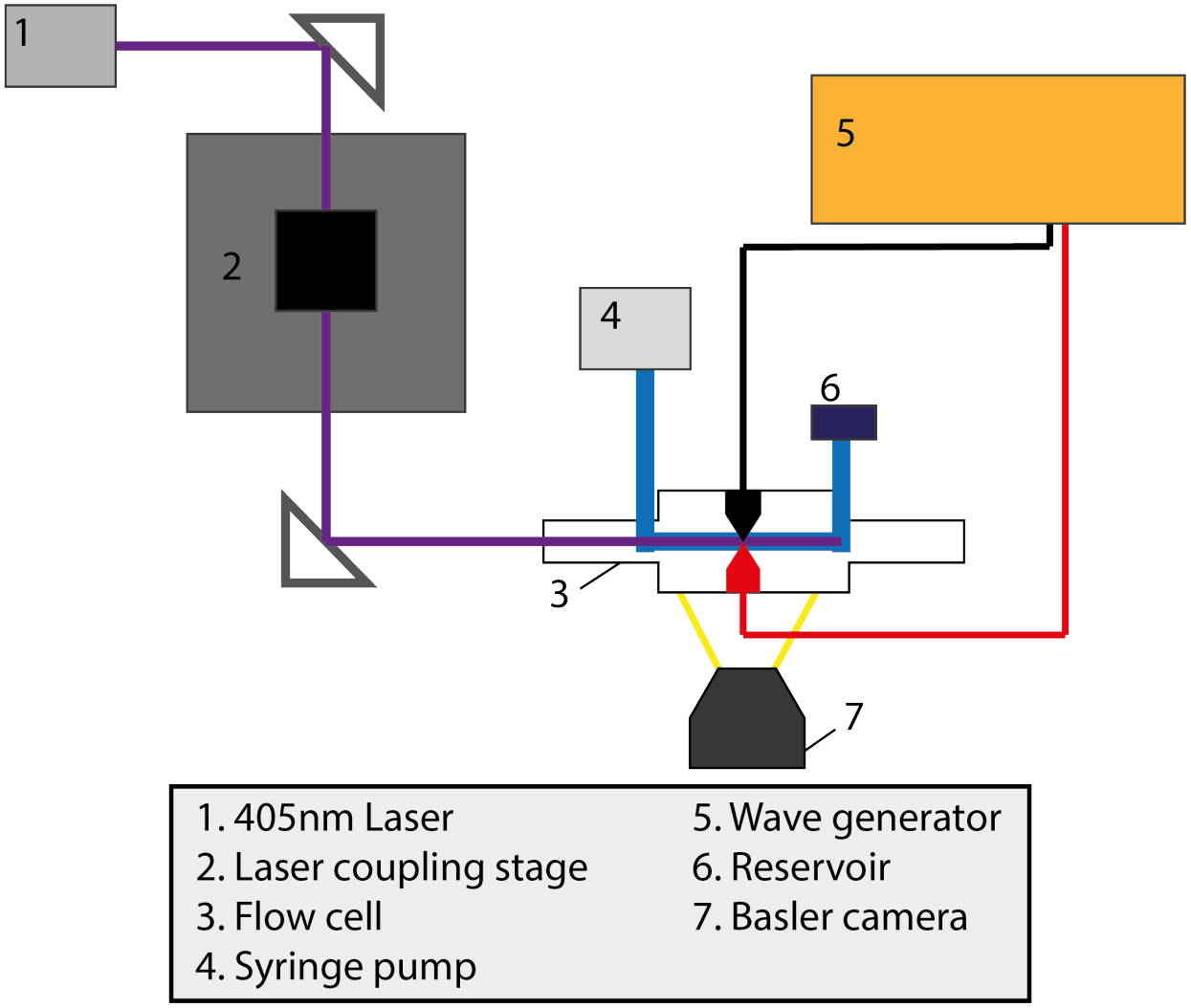


Figure 7: Schematic overview of the setup used to create both a pressure-driven flow and an electro-osmotic flow in my channel.

General overview A schematic overview of my setup is shown in Fig. 7. The laser gets coupled to the optical fibre, which enters the sample. A syringe pump pumps fluid into the channel, and a wave generator generates an electrical field between the two copper electrodes in the channel. A camera records the channel from below.

Camera The camera used in my setup is a Basler daA128-54um. It has CMOS (Complementary Metal Oxide Semiconductor) image sensor of 1280 by 960 pixels. The size of each

pixel is 3.75 μm. I calibrated the camera with a calibration target slide. Calibration shows that the effective pixel size is 226 nm. As I wanted to synchronize the framerate with the frequency of the square wave, I chose for an exposure time of 75000 μs, which I found to correspond with 10 frames per seconds on my camera.

Wave generator The electric field used to create the electro-osmosis within the channel is made by generating a square wave with a frequency of 10 Hz, with a peak-to-peak electric potential difference of 10 volt. The wave is completely symmetric around the mean, which is 0 volt.

Laser and coupling The laser used for illumination is a continuous laser (Thorlabs LP405-SF10) with a wavelength of 405 nm. The laser bundle gets directed into the stage used for coupling the light with the multimode optical fibre (Thorlabs FG050LGA) with 50μ core. The stage can be moved in the x,y, and z-axis by turning the corresponding drives.

Syringe pump The pressure inside the channel is regulated by a programmable syringe pump. The fluid being pumped is a 500-times diluted mixture of 0.05 mg/mL 100 nm Polyethylene Glycol-coated NanoXact Gold particles, and is being pumped out of a 1 mL disposable syringe. Fluid that passed through the channel will get dumped in the reservoir.

Sample holder The holder is made up of a top plate and a bottom plate, which are screwed on to each other, with the sample in between. The designs of the holder can be seen in Fig. 8. The area marked in green is where the sample resides. Both the top and bottom parts have been cut out of acrylate; the top is 4 mm thick, and the bottom is 8 mm. This is done because now the sample is slightly farther away from the microscope stage which makes for more room to attach and easier attachment of the electrodes to the copper tape. In the middle of the holder, gaps have been cut at the sides of both the top and bottom for easy access to the copper tapes, and thus to make attaching the electrodes easier. On the bottom, what's inside the green lines has been milled 2 mm deep to make a space for the sample. The darkened area seen on the bottom holder in the figure has also been milled, as to make room for the optical fibre to enter the holder. The bottom of the sides of the bottom holder have been milled 2 mm deep to fit properly on the microscope stage. The eight blue holes have been made to be able to screw the top plate on the bottom plate of the setup. Lastly, both the top and bottom of the holder have been milled around the opening in the middle for easier access of the camera and better collection of light on the camera.

The top holder has two small holes inside the green area; I put red boxes around them in Fig. 8 for clarity. These are the inlet and the outlet for the fluid, corresponding to the inlet and outlet in the sample (see Fig. 5). Two small metal tubes are glued on the top and serve as the inlet and outlet, and on top of these, rubber tubes have been fastened. To make sure that the fluid doesn't leak when entering the sample, two spacers have been glued around the

inlet and outlet to the bottom of the top holder, and two rubber o-rings are placed between the sample and the top holder.

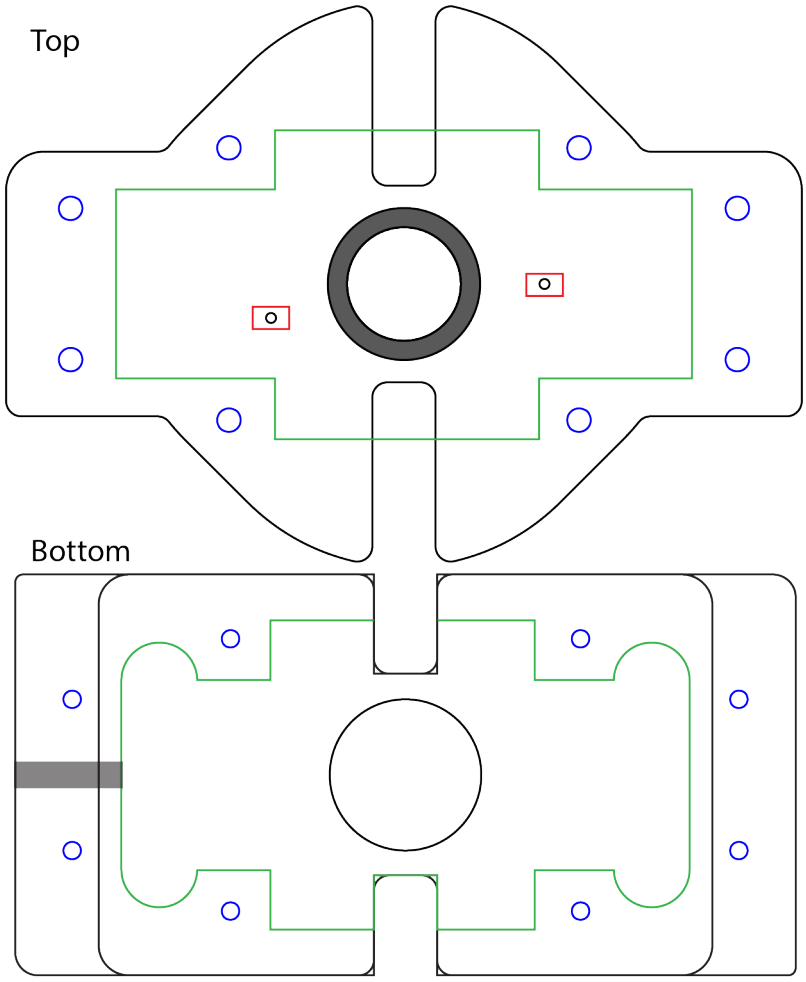


Figure 8: The designs of the top and bottom of the sample holder. The area marked in green is where the sample resides. The eight blue holes show where the screws fasten the top and bottom part together. The two small holes inside the red boxes show where the inlet and outlet are on the top part. The dark grey area on the bottom part shows where the bottom has been milled to allow for the entrance of an optical fibre.

3.3 Mode of operation

To investigate the relation of the flow rate and the surface charge, I'll vary the flow rate by using the syringe pump. I am interested in the slope of the particle under the effect of the electric field. A typical frame of a video is seen in Fig. 9.

I first start by coupling the laser and the optical fibre. When this is finished, I place the sample in the holder and position the o-rings so that the holder won't leak when fluid passes through the sample. After testing the sample for leaks, I place the holder on the microscope

stage, and focus the microscope on to the area of interest, i.e. the part of the channel where the two copper triangles are opposing each other. I do this by focusing first on the actual copper, and then slowly going away from them until I focus on a plane where particles show up. I found this to be about 0.06 mm away from the bottom, by using the measurement scale on the microscope. I attach the two electrodes to the copper tape, and refocus the microscope if needed.

I fill a 1 mL syringe with the fluid with gold particles (mentioned in Sec. 3.2) and couple it to the inlet. I place the syringe in the syringe pump, and pump the syringe with a flow rate of 0.1 mL/min out of the syringe. If there are still no leaks, I can start the measurements.

Firstly, I set the required camera settings, e.g. a frame rate of 10 fps and a exposure time of 75000 μs (complete metadata in App. A). Then, for the range of flow rates between 0.00 and 0.10 mL/min on the syringe pump, with a 0.01 mL/min increment, I first flush the channel until the flow is stationary, then I record a video without electric field to be able to correct the slope of the particles later. After this, I turn on the electric field and do the same, twice. Between recordings, I wait about 30 seconds to make sure that the particles recorded in a previous video don't have an impact on the particles that are to be recorded in the next video. Every recorded video is 50 frames long.

4 Results

I split this section in two parts: first the data analysis will be explained, and then the results of my experiments will be shown. All scripts discussed below are written in Python 3.6, and can be found in App. B.

4.1 Data Analysis

We've seen in Sec. 3.3 that each measurement consists of three videos of 50 frames each. Let us walk through the data analysis by the example of a single video, namely my first measurement at set flow rate of 0.10 mL/min.

I start with using ImageJ to cut the video in its frames, resulting in 50 images. I then use a script to calculate the most probable value of every pixel in these images, and then subtracts these from the actual pixel value in every image. Any negative values that result from this are set to 0. The effect of the removal is seen in Fig. 9.

After the background has been removed, I need to find the particles in the videos, so I can then find their slopes. As the particles have a quite unusual shape, and makes it hard to track the complete particle appropriately, I use a script to be able to select a rectangle around the particle, specifically the downward slope of the particle, which will then be cropped out, for every frame in a video. It's important I use either only upward or downward slopes, as the misalignment of the copper triangles might have an influence on the respective slopes. More about this in Sec. 5.2.2 of the Discussion.

The cropped-out particle slopes will then undergo another, heavier background-reduction, which consists of setting all particle values lower than 80 to 0. After this, I use probabilistic Hough transform to "draw" the particle again by using lines. This is seen in Fig. 10.

To find the slope of a particle, I take the slopes of all the lines that the probabilistic Hough transforms finds, and then average over them. To find the average slope of particles in a single video, I average over the individual slopes of all the used particles. Because of the probabilistic nature of probabilistic Hough transform, I need to account for some uncertainty in the found slope. This is handled by doing the detection of edges and the probabilistic Hough transform five times, and averaging over them. The result of this is an average slope and uncertainty per video.

This calculating of the slope is also done for the videos without electric field, so I can use those to correct the found slopes for a possible rotation of the axes, be it from the camera not being lined up correctly or from the flow preferring to go slightly downwards.

The new, corrected slope is calculated via the following equation:

$$S_\mathrm{c} = \arctan(\tan(S) - \tan(S_0)) \tag{20}$$

where S is the found slope that is affected by the electric field, S_0 the found slope that is not affected by the electric field, and S_c the corrected slope.

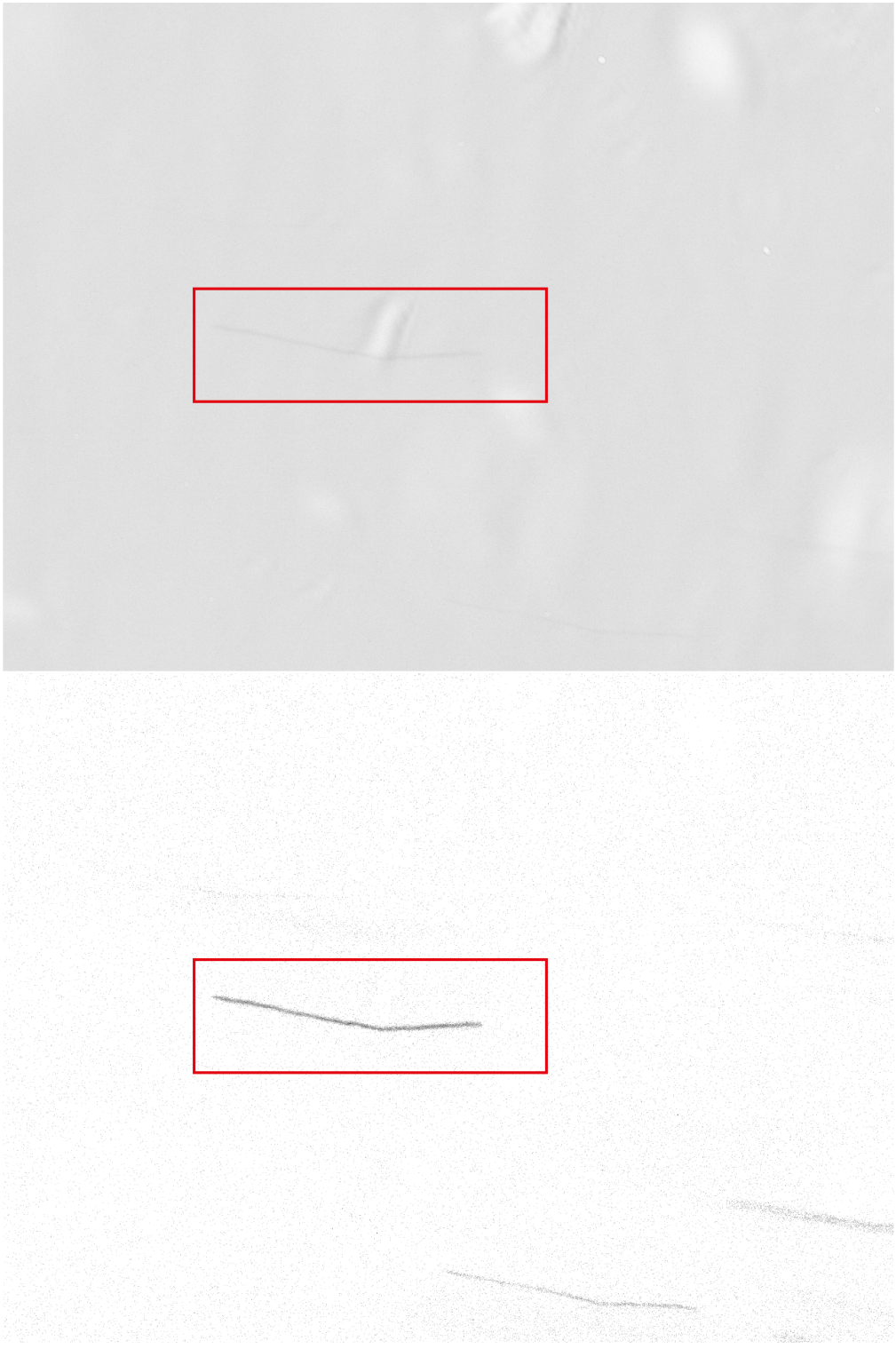


Figure 9: Here we see a frame from the first measurement of the set flow rate of 0.10 mL/min. On top, you can see the original frame of the video. On the bottom you see the same frame, but now after my background-removal script has acted upon it. Both frames have their colours inverted, and I put a red box around a particle to emphasise its location .

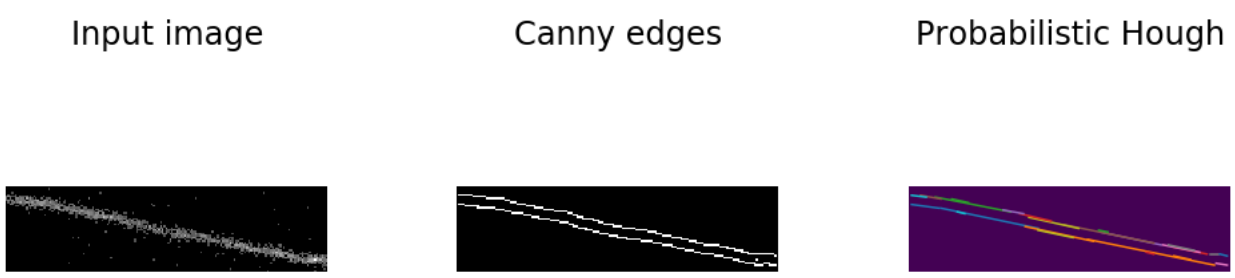


Figure 10: The slope of a particle in a frame of the first measurement of the set flow rate of 0.10 mL/min. On the left, we see the cropped slope that has been put in. In the middle, we see the edges detected by the edge-detection algorithm. On the right, we see the lines that the probabilistic Hough transform finds in the detected edges.

I find the amplitude of the oscillating behaviour of the particles that are influenced by the electrical field by the method discussed in Sec. 2.4. I first find the average length of the particles that aren't influenced by the electric field, and I multiply the slope of a measurement by half the length of the particle that's not affected by the electric field. I multiply by half the length instead of the full length, as one full oscillation is done in one frame, so a peak-to-peak is done in half the time it takes for a full oscillation. To find the electro-osmotic velocity, I simply divide the peak-to-peak by half the oscillation time.

4.2 Results

The calculated amplitudes can be found in Table 1. The uncertainties come from the probabilistic nature of the probabilistic Hough transform. The measurements of the set flow rate of 0.01 mL/min has been left out, as there weren't any videos with a downward-slope that was large enough to be able to crop out. Both the measurements are plotted in Fig. 11, where the blue corresponds with the first measurement series, and the red with the second measurement series.

The calculated EO velocity out of these amplitudes is found in Table 2. The relative EO velocity compared to no pressure-driven flow is seen in Fig. 12.

In Fig. 11 we see the amplitude of the oscillation plotted out against the pressure-driven velocity. The first measurement series is in blue, the second in red. In both measurement series we can see that the higher the pressure-driven velocity becomes, i.e. the higher the flow rate becomes, the lower the found amplitude gets. From the figures, it seems that the dependency on flow rate is most noticeable for low flow rates, and that for higher flow rates, the amplitudes stay around the same values. One could see a linear downwards trend in these data points.

In Fig. 12 we see the relative EO velocity plotted out against the pressure-driven velocity, with the amplitude at zero pressure-driven flow as 1. We can see a linear downwards trend, the higher the pressure-driven velocity becomes.

Table 1: Overview of the calculated amplitudes and their uncertainties.

Set flow rate (in mL/min)	Measurement	Amplitude (in μm)	Uncertainty (in μm)
0.00	1	5.70321	0.60207
0.02	1	4.61896	0.04886
	2	4.76666	0.04031
0.03	1	3.71839	0.05888
	2	4.97025	0.08931
0.04	1	5.56325	0.35144
	2	3.91559	0.08910
0.05	1	4.36695	0.15101
	2	3.5238	0.1352
0.06	1	4.11381	0.03979
	2	2.83647	0.03982
0.07	1	4.47626	0.08418
	2	4.71947	0.07732
0.08	1	3.19198	0.03345
	2	2.98461	0.02011
0.09	1	3.08476	0.05189
	2	2.50037	0.05246
0.10	1	3.2541	0.0366
	2	3.07199	0.01663

Table 2: Overview of the calculated EO velocities and their uncertainties.

Set flow rate (in mL/min)	EO velocity (in μm/s)	Uncertainty (in μm/s)
0.00	152.086	11.353
0.02	125.142	0.845
0.03	115.849	1.426
0.04	126.385	4.834
0.05	105.21	2.70
0.06	92.6705	0.7505
0.07	122.61	1.52
0.08	82.3545	0.5204
0.09	74.4684	0.9838
0.10	84.3478	0.5354

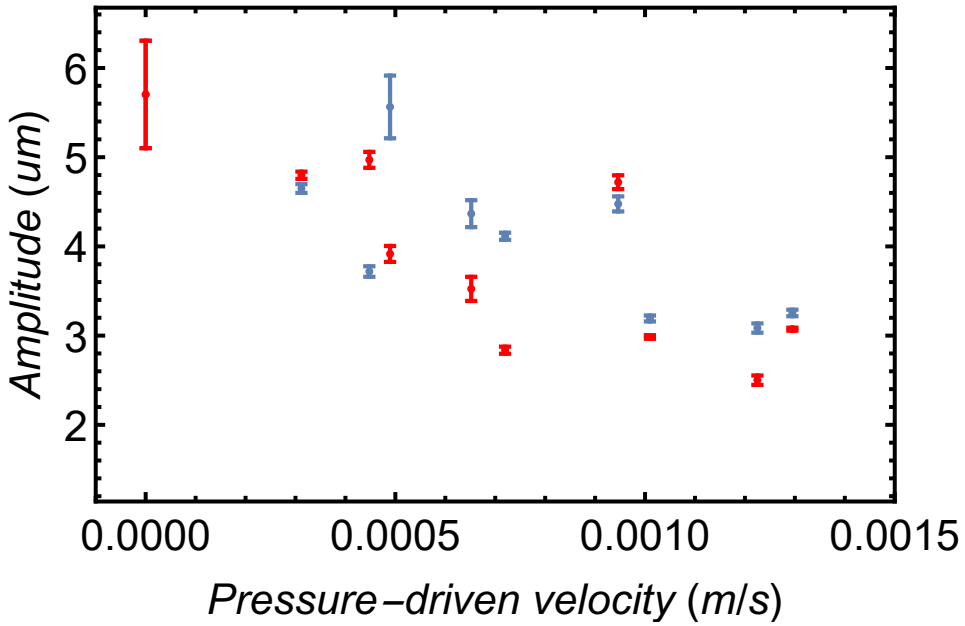


Figure 11: The amplitudes from both measurement series plotted out. The first measurement series is in blue, the second is in red.

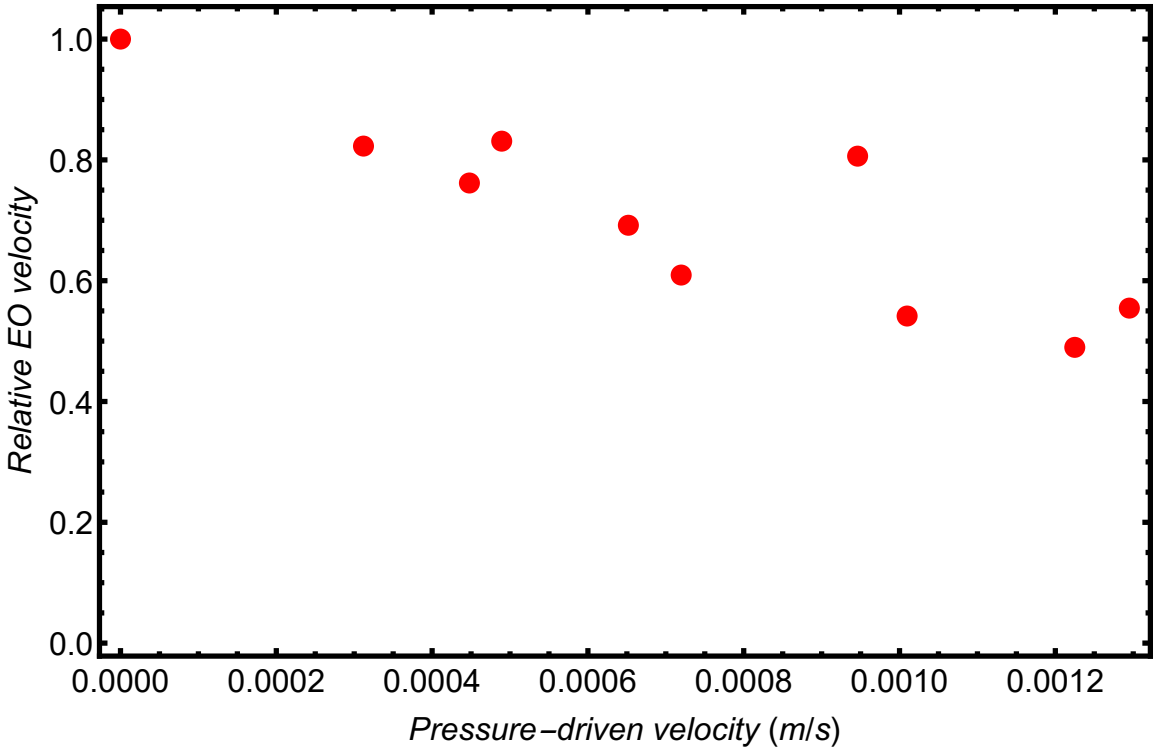


Figure 12: The relative EO velocity compared to zero pressure-driven flow is plotted out against the set flow rate.

5 Discussion and outlook

In this thesis, I looked at the relation between the flow rate and the surface charge in a microfluidic channel. I will now discuss my findings and the approximations that were made, possible improvements to the setup and further experiments.

5.1 Results

The two main figures 11 and 12 show a clear downwards trend for an increase in flow rate. It looks like the dependence on flow rate gets less and less the higher the pressure-driven velocity, but it seems like a linear relation for the lower velocities. From the paper by Mischa Bonn [16], one would expect that the difference between high enough flow rates wouldn't matter, and the amplitudes would remain the same. This might be what we're starting to see for the higher flow rates, but I don't have enough data to say for certain if this is the case or not.

As we can see, there is a sudden spike in both of my measurements around 0.0009 m/s, which corresponds with a set flow rate of 0.06 mL/min. As there is no satisfactory reason for the surface charge to be so different from the surface charge at a slightly higher and lower flow rate, I think that this behaviour is due to a measurement problem, rather than describing what's going on inside the channel accurately. Neither paper [16, 17] about the relation of surface charge and flow support this spike as well, so it's safe to say that the spike is an anomaly.

In the theory sections Sec. 2.1 and 2.2 I made the assumption that my channel could be approximated by two parallel plates, whereas it is in fact a rectangular channel. I am allowed to make such an approximation, as the width and length of the channel are much larger than the depth. Another approximation in Sec. 2.2 is that the derivation for the relation between electro-osmotic velocity and surface charge is found using the electro-osmotic velocity of an open channel, whereas in reality I'm using a closed channel. As the important finding is that the EO velocity and surface charge are proportional to each other, and as this is also the case for a closed channel, the approximation does not have an impact on my findings.

5.2 Possible improvements

During the analysis of the data, I found some areas I could improve upon. I'll briefly discuss per area what the problems are and how they can be solved.

5.2.1 Velocity of the fluid

Ideally, one would want the highest pressure-driven velocity of their fluid as possible. As the limit of the velocity of the fluid is, without taking the structure and sturdiness of the channel into account, the velocity at which the elongated particle is longer than your camera

records, and that this can be solved by simply lowering the exposure time and upping the frame rate of the camera, one could theoretically get his particles to move very fast along the channel.

This would result in a number of positives, namely that the effect of Brownian motion would be much less compared to the effects of the pressure, and that any possible changes of the surface charge that come into existence because of the flow rate would be larger, and thus easier to spot and quantify.

Another pro would be that the syringe pump would pump more smoothly, which would make for a more constant flow inside the channel. Because of the low flow rates now, especially around 0.01-0.05 mL/min, the syringe pump would pump harder than normal once in a while. This makes the flow less steady, and isn't stationary any longer. This could also be solved by using an even smaller syringe, but they weren't available during the measurements.

I didn't pursue these higher velocities all that much, apart from some explorational research, as a higher velocity would need a higher frequency of the square wave, which would lead to a loss amplitude. This would've been easily solvable by simply turning up the voltage used, but I soon found out that this resulted in the creation of bubbles by virtue of electrolysis. These bubbles pushed the liquid around and caused incredible scattering of the illumination, and thus made higher voltages unusable with my current setup. This could be solved by using a different kind of electrode, perhaps ones that are even outside of the sample entirely. Then there would be no chance of electrolysis and I can then up the voltage significantly and make the frequency of the square wave much higher.

In that case, the lighting would be better as well, as the exposure time would be less, resulting in less background noise.

5.2.2 Difference upward/downward slope

Another problem with my current sample is that the downward slope and upward slope of an oscillating particle might not be the same. Ideally, the electric field is completely perpendicular to my channel, and would thus have no influence whatsoever on the velocity of a particle that is travelling along the channel. Unfortunately, the placement of the electrodes is almost certainly imperfect, and will then result in some interference with the pressure-driven velocity. A schematic of what is meant is seen in Fig. 13.

I tried to take this into account by only using the downward slope for my calculations, so my calculations would be consistent, which is also the reason why both the measurements for the set flow rate of 0.01 mL/min weren't used, and used two small cameras to help align the copper electrodes on the glass cover slips.

Even so, this behaviour could influence my measurements by reducing one of the slopes when the pressure-driven flow gets bigger. If I would then choose the wrong slope to focus my research on, I could get higher amplitudes when there is no flow and lower amplitudes when there is flow, purely because of this offset. If this is the case, then one could construct my findings purely by misaligning the electrodes.

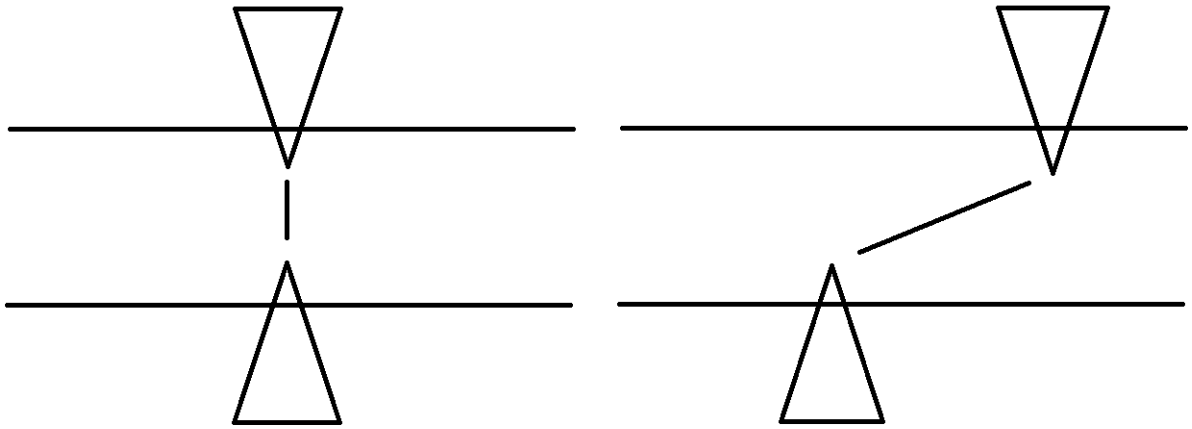


Figure 13: A schematic of the influence an imperfect placement of the electrodes has on the velocity along the channel.

Of course, every sample would still have some sort offset of the electrodes, but the total effect could be remedied by doing measurements not just with a single sample, but with a few similar samples, that would all have some offset of the electrodes. As the offset is based around 0, i.e. there is no reason why there would be more samples with the upper electrode more to the right than to the left, using more samples that are similarly made would counter the individual distortions of the offset of the electrodes.

One could also test their samples by either making sure that every set flow rate has a measurement where one could use either slope to find the amplitude, and then compare their findings. My measurements weren't suitable for this, however. Another way to check if the effect of the electrodes is significant would be to check the angle between the oscillation of the particles without any flow, and the particles that are being driven by flow but not under effects of an electric field. I did this by drawing lines through both, and finding the angle between the two lines. A schematic of this idea can be found in Fig. 14. For particles in the pressure-driven flow, I used the first measurement of the 0.09 mL/min series. By drawing lines through the samples and finding the angles, I found that the average angle between the two particles was 90.6 degrees. The difference with a perfect 90 degree angle is relatively small, so I assume correcting for this will not have much of an impact.

A third option to test this behaviour might be to simply reverse the flow and measure the slopes again.

5.2.3 Forming of vortexes

Another thing is that, because I'm working with closed channel electro-osmosis, vortexes start to show up in my channel. Because one side of my channel is made from glass and another from PET, the respective surface charges are different as well. This results in a vortex in my channel where a particle would move left at one side of my channel, and right at the other side. To circumvent this behaviour, I chose a voltage and frequency at which

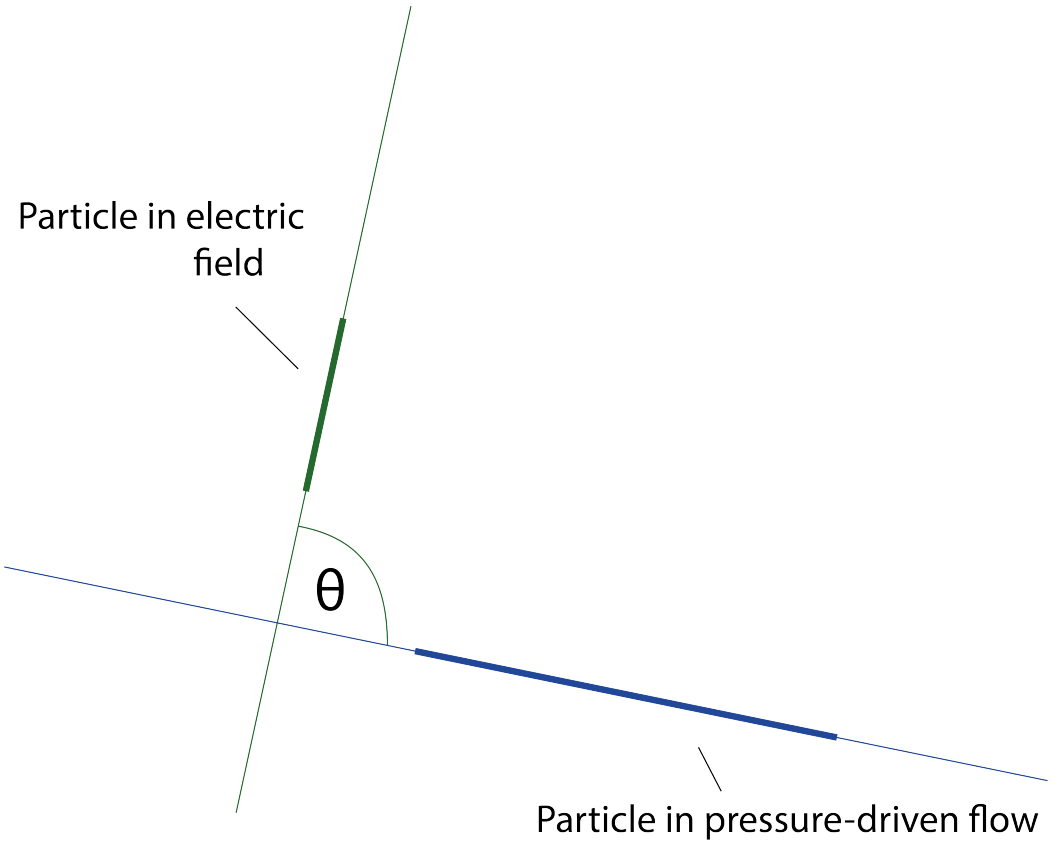


Figure 14: Caption

the effect didn't seem to show up much. Of course, I talked about using a higher voltage above, at which the problem might appear again.

The effect lessens when the frequency of the square wave is set higher as well, but as far as I've seen, it still occurred. A solution to this might be to make the channel thicker, as then the vortexes would need to be bigger and thus slower, and perhaps using electrodes that are outside of the channel would take care of this as well. On the other hand, such vortexes are inherent to the structure of the flow cell as a closed channel, and as the bottom and top are of two different materials, with different surface charges.

5.3 Further research

As stated in Sec. 3.1, the flow cell itself was designed with versatility in mind, as it can be quickly made and re-designed for another purpose.

Some ideas for further research might be to see what happens at higher velocities, as discussed in Sec. 5.2; to see what the effect is when the electric field is turned on for much longer times, such as an hour, before the measurements; to do this experiment for a range of pH's; or to see what happens in different fluids, perhaps non-Newtonian fluids.

A nice possible experiment can be based on a paper by B. L. Werkhoven, J. C. Everts, S. Samin, and R. van Roij [17]. They simulated an experiment similar to Bonn's [16] and mine, where they looked at the relation between surface charge and flow rate in a channel. They found that flow rate indeed has an effect on the surface charge, but it also depends on the position along the length of the channel. This is seen in Fig. 15 (c). The system size of their simulation is as follows: they choose the depth of the channel H as 0.5, 1, or 2 μm and the length of the channel L as 10,20,30, or 40 μm. The size of my system, as mentioned in Sec. 3.1, is H = 0.18 mm and L = 24 mm. The ratio between these is 133.33, instead of the fixed 40 as in both Bonn and Werkhoven.

As the simulation is looking close to the Stern layer, which means very close to the side of the channel. As said in Sec. 3.3, my measurements are actually some 0.06 mm away from the bottom of the channel. I can calculate the velocities at the Stern layer by using the equation for pressure driven flow and its derivative to the depth, i.e. $\frac{\tau}{\mu}$. If we look at Fig. 12, but now with the pressure-driven velocities at the Stern layer, we get Fig. 16. As can be seen, the lowest EO-velocity is around 55% of the EO-velocity without any pressure-driven flow. This is in line with the findings of Werkhoven et al.

Exploring this more accurately is definitely very interesting, and an experiment would be quite easy to set up. By adjusting the optical fibre such that one can see particles close to the Stern-layer, and by making the channel long enough, the theory can be tested without a whole lot of trouble. Either one could make a channel with two sets of electrodes, and measure at both ends of the channel, or one could put the pair of electrodes at one end of the channel, and reverse the flow (one could see this as effectively rotating the channel 180 degrees) to see what the difference is between surface charge at the beginning of the channel at a certain flow rate and at the end.

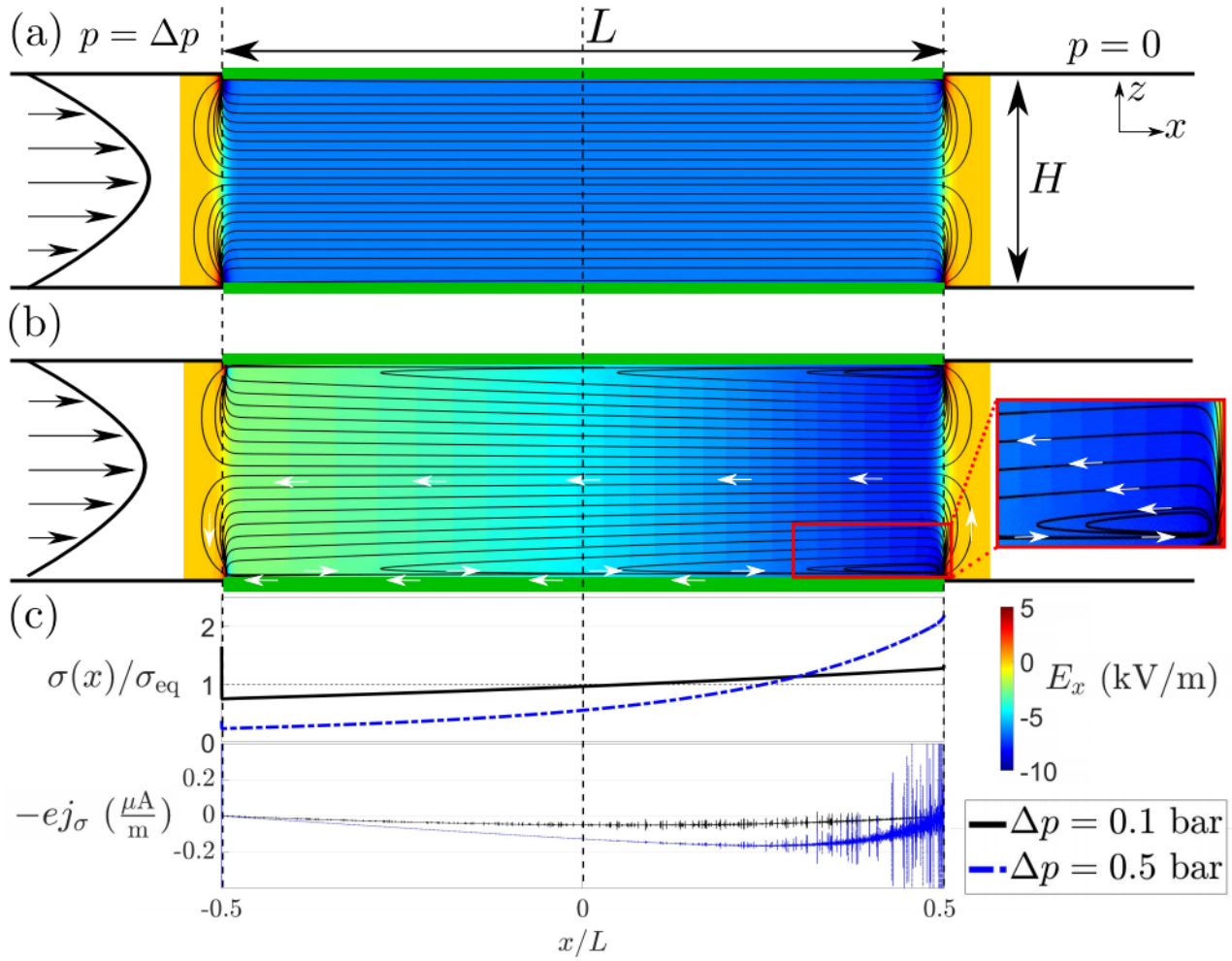


Figure 15: "Streamlines of the net charge flux and colour map of the tangential electric field $E_{\rm x}$ near the charged surfaces (green stripes) of a rectangular channel with a pressure drop $\Delta p = 0.5$ bar between in- and outlet at $x = \pm L$, (a) with vanishing Stern-layer conduction ($D_s = 0$ resulting in fixed surface charge of $-e\sigma_e q = -0.069\ e/\text{nm}^2$ that mimics silica at pH = 6.5, and in (b) with non-zero Stern-layer conductance and our dynamic charge regulation model. (c) Flow-induced heterogeneous surface charge density $\sigma(x)$ and surface charge flux $-ej_\sigma$ for $\Delta p = 0.1, 0.5$ bar for the parameters of case of (b)." [17]

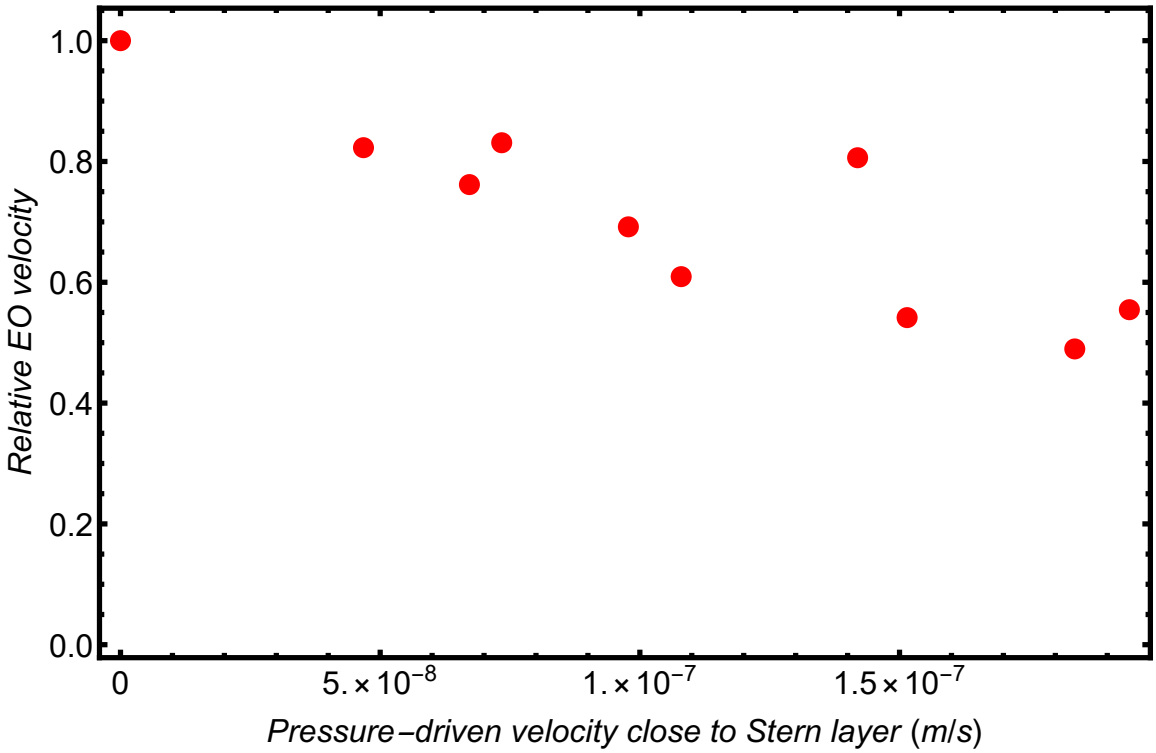


Figure 16: The relative EO velocity compared to zero pressure-driven flow is plotted out against the set flow rate. The pressure-driven velocity is now for fluid close to the Stern-layer.

6 Conclusion

In this thesis, I described my research into the relation between surface charge and the flow rate in a microfluidic channel. I found a method of sample-making, with which I can make easy-to-build suitable samples, and got proper illumination by illuminating the channel from inside. I changed the flow rate via a pressure-driven flow, and found the electro-osmotic velocity by recording the particles oscillating because of a square wave electric field, and calculating their amplitude. The amplitude is calculated by finding the slope of the particle under influence of the electric field, and the length of the particle when it's not under influence. As I've shown in Sec. 2.3, the electro-osmotic velocity is proportional to the surface charge.

The results show that there is indeed a difference in electro-osmotic velocity at different pressures. My data shows a mostly linear downwards trend in the electro-osmotic velocity versus the pressure-driven velocity, and as such, the surface charge changes as well. Even taking into account other factors discussed in the discussion, the data shows a significant relation between surface charge and flow rate. My findings agree with the findings of Bonn et al. and Werkhoven et al. [16, 17].

7 Acknowledgements

I would like to thank Sanli Faez for giving me the opportunity to work on this project. It was a very interesting topic, I learned a lot about the physics and research in general, and I really appreciated the freedom to figure things out on my own, and solving problems in an easy, creative way. Secondly I would like to thank my daily supervisor Dashka Baasanjav for his help when needed, both showing me how things worked and for helpful tips and discussions.

I'd also like to thank Peter Speets, Qianjing Tang, and Tommaso Pavolini for their input on practical problems such as sample-making, gluing, and programming, and creating a fun and lively atmosphere in the lab.

Lastly I would like to thank everyone in the Nanophotonics group for having me, for interesting discussions about physics and non-physics-related topics, and for the good times I had during my research.

References

[1] H. Chang, Can. J. Chem. Eng. p. 1 (2006).

[2] J. D. Harrison, K. Fluri, K. Seiler, Z. Fan, C. S. Efffenhauser, and A. Manz, Science p. 895 (1993).

[3] P. H. Paul and D. J. Rakestraw, *Electrokinetic high pressure hydraulic system* (2000), a U. S. Patent No. 6019882.

[4] C. T. Culbertson, R. Ramsey, and J. M. Ramsey, Anal. Chem p. 2285 (2000).

[5] S. L. R. Barker, D. Ross, M. J. Tarlov, M. Gaitan, and L. E. Locascio, Anal. Chem p. 5925 (2000).

[6] S. C. Jacobson, T. E. Mcknight, and J. M. Ramsey, Anal. Chem p. 4455 (1999).

[7] D. Erickson and D. Li, Anal. Chem p. 3208 (2004).

[8] C. T. Culbertson, S. C. Jacobson, and J. M. Ramsey, Anal. Chem p. 5814 (2000).

[9] S. V. Ermakov, S. C. Jacobson, and J. M. ramsey, Anal. Chem p. 3512 (2000).

[10] O. Salas-Solano, D. Schmalzing, L. Koutny, S. Buonocore, A. Adourian, P. Matsidaira, and D. Ehrlich, Anal. Chem p. 3129 (2000).

[11] A. T. Wooley and R. A. Mathies, Anal. Chem p. 3676 (1995).

[12] F. von Heeren, E. Verpoorte, A. Manz, and W. Thormann, Anal. Chem p. 2044 (1996).

[13] B. E. Slentz, N. A. Penner, E. Lugowska, and F. Regenier, Anal. Chem p. 361 (2003).

[14] F. F. Reuss, Mmoires de la Socit Impriale des Naturalistes de Moscou pp. 327–337 (1809).

[15] J. H. Masliyah and S. Bhattacharjee, *Electrokinetic and Colloid Transport Phenomena* (John Wiley & Sons, 2006).

[16] M. Bonn, D. Lis, E. H. G. Backus, J. Hunger, and S. H. Parekh, Science pp. 1138–1142 (2014).

[17] B. L. Werkhoven, J. C. Everts, S. Samin, and R. van Roij, PRL (2018).

[18] P. K. Kundu, I. M. Cohen, and D. R. Dowling, *Fluid Mechanics* (Academic Press, 2016), 6th ed.

[19] H. Bruus, *Theoretical Microfluidics* (Oxford University Press, Oxford, 2008).

[20] F. Ontiveros, *Instructions for making a petl microfluidic cell*, URL `https://sites.google.com/a/sjfc.edu/nanobiology/home/petl-instructions`.

[21] P. Speets, *Script for background removal*, URL `https://github.com/nanoepics/CETgraph/blob/peter/tracking/subtractBackground.py_;_https://github.com/nanoepics/CETgraph/blob/peter/tracking/track.py`.

A Camera settings

The camera settings for my Basler daA1280-54um used in my experiments are as follows:

```
Basler daA1280-54um (22148129)
  Analog Control
    Gain Selector                          All
    Gain [dB]                              7.51327
    GainRaw                                76
    Gain Auto                              Off
    Auto Gain Lower Limit [dB]             0
    AutoGainLowerLimitRaw                  32
    Auto Gain Upper Limit [dB]             18.0278
    AutoGainUpperLimitRaw                  255
    Black Level Selector                   All
    Black Level [DN]                       0
    BlackLevelRaw                          0
    GammaRaw                               1000
    Gamma                                  1
    Color Space Mode                       <not available>
  Image Format Control
    Sensor Width                           1280
    Sensor Height                          960
    Width Max                              1280
    Height Max                             960
    Width                                  1280
    Height                                 960
    Offset X                               0
    Offset Y                               0
    Binning Horizontal Mode                Average
    Binning Horizontal                     1
    Binning Vertical Mode                  Average
    Binning Vertical                       1
    Reverse X                              0
    Reverse Y                              0
    Pixel Format                           Mono8
    Pixel Size                             Bpp8
    Pixel Dynamic Range Min                0
    Pixel Dynamic Range Max                255
    Test Pattern                           Off
  Acquisition Control
    Acquisition Mode                       Continuous
    Acquisition Start                      <not readable>
    Acquisition Stop                       <not readable>
    Sensor Shutter Mode                    Global
    Overlap Mode                           Off
    Immediate Trigger Mode                 Off
    Exposure Time [us]                     75000
    ExposureTimeRaw                        75000
```

Exposure Auto	Off
Auto Exposure Time Lower Limit [us]	100
AutoExposureTimeLowerLimitRaw	100
Auto Exposure Time Upper Limit [us]	100000
AutoExposureTimeUpperLimitRaw	100000
Trigger Selector	FrameStart
Trigger Mode	Off
Trigger Software	<not readable>
Trigger Source	Line1
Trigger Activation	RisingEdge
Exposure Mode	Timed
AcquisitionFramePeriodRaw	100000
Acquisition Frame Rate [Hz]	10
Resulting Frame Rate [Hz]	10
ResultingFramePeriod [us]	100000
Image Quality Control	
ContrastEnhancementRaw	0
Contrast Enhancement	0
Contrast Mode	Linear
BslBrightnessRaw	0
Brightness	0
BslContrastRaw	0
Contrast	0
Defect Pixel Correction Mode	On
Digital IO Control	
Line Selector	Line1
Line Mode	Input
Line Inverter	0
Line Source	UserOutput1
Line Format	LVTTL
LineDebouncerTimeRaw	0
Line Debouncer Time [us]	0
Line Status	1
Line Status All	3
User Output Selector	UserOutput1
User Output Value	0
Auto Function Control	
Target Brightness	0.5
AutoTargetBrightnessRaw	50
Backlight Compensation	0
AutoBacklightCompensationRaw	0
Auto Function Profile	Smart
Auto Function ROI Control	
ROI Selector	ROI1
Width	1280

Height	960
Offset X	0
Offset Y	0
Brightness	1
User Set Control	
User Set Selector	Default
User Set Load	<not available>
User Set Save	<not available>
User Set Default	Default
Test Control	
Test Pending Ack [ms]	0
Transport Layer Control	
Payload Size [B]	1228800
TL Params Locked	1
USB Speed Mode	SuperSpeed
Payload Transfer Size	1048576
Payload Transfer Count	1
Payload Final Transfer 1 Size	180224
Payload Final Transfer 2 Size	0
Internal_PHE	46931
Internal_LKE	0
Internal_URE	41
Internal_EPU	0
Internal_LKR	28
Internal_SQE	0
Internal_ERT	0
Internal_ERC	0
Device Control	
Device Vendor Name	Basler
Device Model Name	daA1280-54um
Device Manufacturer Info	dtx=xa3
Device Version	106681-11
Device Firmware Version	p=daA1280_54um/s=r/v=1.4.0.25(
Device Serial Number	22148129
Device User ID	
Device Scan Type	Areascan
Device SFNC Version Major	2
Device SFNC Version Minor	2
Device SFNC Version Sub Minor	0
Device Reset	<not readable>
BslFirmwareName	daA1280_54um
BslFirmwareLabel	1.4.0.250701
BslFirmwareCompatibilityID	1
BslDeviceRole	Camera
BslDeviceRoleKey	0

```
    Device Link Selector                          0
    Device Link Speed [Bps]                       500000000
    Device Link Throughput Limit Mode             On
    Device Link Throughput Limit [Bps]            163000000
    Device Indicator Mode                         Active
    Device Registers Streaming Start              <not available>
    Device Registers Streaming End                <not available>
  Calibration Control
    Defect Pixel Correction Version               0x20000
    Defect Pixel Table Size                       64
    Calibration Store                             <not available>
    Calibration Lock Challenge                    1727623110
    Calibration Lock Response                     0
Transport Layer
  Migration Mode Enable                           0
  Statistic
    Read Pipe Reset Count                         0
    Write Pipe Reset Count                        0
    Read Operations Failed Count                  0
    Write Operations Failed Count                 0
    Last Error Status                             0
    Last Error Status Text
Stream Parameters
  Maximum Number of Buffers                       10
  Maximum Buffer Size                             1228800
  Maximum Transfer Size                           1048576
  Num Max Queued Urbs                             64
  Transfer Loop Thread Priority                   15
  Transfer Timeout                                4000
  Statistic
    Total Buffer Count                            2738
    Failed Buffer Count                           0
    Last Failed Buffer Status                     0
    Last Failed Buffer Status Text
    Missed Frame Count                            0
    Resynchronization Count                       0
    Last Block Id                                 2737
Image Format Conversion
  Output Bit Alignment                            MsbAligned
  Padding X                                       0
  Output Orientation                              Unchanged
  Inconvertible Edge Handling                     SetZero
  Mono Conversion
    Mono Conversion Method                        Truncate
    Gamma                                         <not available>
```

Additional Left Shift	0

B Analysis Code

My analysis code is made up out of three important parts: a script for background removal; a script for getting the coordinates for the particles; and a script for finding the average slope of a measurement.

Removing background The code for removing the background is adapted slightly from code by Peter Speets [21]. I both use his track.py and SubtractBackground.py, but have adapted his SubtractBackground.py slightly, as seen below:

```
import numpy as np
import matplotlib.pyplot as plt
import trackpy as tp
import math
from PIL import Image
import os
import datetime
import h5py
import cv2
import pims
import pickle
import csv
import sys #gives sys.exit() for debugging
from track import Tracking


folder ="E:/New_folder/measurements180507/00/New_folder"

trackingObject = Tracking(folder, 31, 1750, 4000, 10, 0.225664, h5name = "data.h5", FPS = 10 ,useFrames =
trackingObject.currentPath = folder
maxFrames = 300
array = []
dimensions = [300,960,1280]
if(maxFrames > 0):
    dimensions[0] = np.amin([dimensions[0],maxFrames])
else:
    maxFrames = dimensions[0]

#creates array from which singular values most occuring pixels are calculated
for i in range(np.amin([len(trackingObject.frames),maxFrames])):
    array.append(trackingObject.frames[i].flatten())

u, s, v = np.linalg.svd(array,full_matrices= False)
s0 = s[0]
for i in range(1,len(s)):
    s[i] = 0

array = (u * s[..., None, :]) @ v
frames = []
temp = []

for i in range(dimensions[0]):
    for j in range(0,dimensions[1]*dimensions[2],dimensions[2]):
        temp.append(array[i][j:j+dimensions[2]])
    frames.append(temp)
```

```
    temp = []

background = frames[0]
trackingObject.saveImage(np.uint8(background), folder + "/background.png")
trackingObject.subtractBackground(background = np.uint16(background))

for i in range(len(trackingObject.frames)):
    trackingObject.saveImage(trackingObject.frames[i], folder +"/background/BG"+str(i)+".png")

print('done')
```

Finding coordinates The code for my script which makes it possible to get the coordinates of the particles found in a frame:

```
from tkinter import *
from tkinter.filedialog import askopenfilename

"""The main idea of this code is to get a list of the coordinates
of the places you clicked on , spread out over a bunch of images.
Left click on the image registers the coordinates clicked on,
releasing the left click will output those coordinates as well,
right click opens a new image and starts a new list for that image,
and middle mouse click ends the entire program and outputs
positionlist, which will be a list of lists-per-frame, which in turn give
the clicked coordinates per frame. Thus one can click and
hold over a certain point   of interest, and get the coordinates
of the two endpoints of a box around that point."""



#setting global parameters to their respective starting positions
positionlist = []
index = 0
indexlist = [index]
clicklist = []
event2canvas = lambda e, c: (c.canvasx(e.x), c.canvasy(e.y))

if __name__ == "__main__":
    root = Tk()

    #setting up a tkinter canvas with scrollbars
    frame = Frame(root, bd=2, relief=SUNKEN)
    frame.grid_rowconfigure(0, weight=1)
    frame.grid_columnconfigure(0, weight=1)
    xscroll = Scrollbar(frame, orient=HORIZONTAL)
    xscroll.grid(row=1, column=0, sticky=E+W)
    yscroll = Scrollbar(frame)
    yscroll.grid(row=0, column=1, sticky=N+S)
    canvas = Canvas(frame, bd=0, xscrollcommand=xscroll.set, yscrollcommand=yscroll.set)
    canvas.grid(row=0, column=0, sticky=N+S+E+W)
    xscroll.config(command=canvas.xview)
    yscroll.config(command=canvas.yview)
    frame.pack(fill=BOTH,expand=1)

    #adding the first image

    File = askopenfilename(parent=root, initialdir="M:/",title='Choose an image.')
    print("opening %s" % File)
    img = PhotoImage(file=File)
    can = canvas.create_image(0,0,image=img,anchor="nw")
    canvas.config(scrollregion=canvas.bbox(ALL))

    #function to be called when mouse is clicked
    def clickcoordinates(event):
        #outputting x and y coords to list for that particular frame.
```

```
        global clicklist
        clicklist = []
        cx, cy = event2canvas(event, canvas)
        clicklist.append([cx,cy])


    #function to be called when mouse is released
    def releasecoordinates(event):
        global clicklist
        cx,cy = event2canvas(event,canvas)
        clicklist.append([cx,cy])
        indexlist.append(clicklist)

    #function to be called when mouse is clicked
    def nextimage(event):
        #This puts the previous list of coordinates in the overarching
        #positionlist, and moves to the next frame.
        global index
        global indexlist
        global img2
        positionlist.append(indexlist)
        index = index+1
        indexlist =[index]
        File = askopenfilename(parent=root, initialdir="M:/",title='Choose an image.')
        print("opening %s" % File)
        img2 = PhotoImage(file=File)
        canvas.itemconfig(can, image=img2)
        canvas.config(scrollregion=canvas.bbox(ALL))

    #function to be called when mouse is clicked
    def close(event):
        #this puts the last list of coordinates in the overarching
        #positionlist, prints that and then closes the program
        global index
        global indexlist
        positionlist.append(indexlist)
        root.destroy()

    #mouseclick event
    canvas.bind("<ButtonPress-1>",clickcoordinates)
    canvas.bind("<ButtonRelease-1>",releasecoordinates)
    canvas.bind("<ButtonPress-3>", nextimage)
    canvas.bind("<ButtonPress-2>",close)

    root.mainloop()

print(positionlist)
```

Finding slopes The third script is used to find the slopes of all the particles found in a measurement, and average them:

```
import numpy as np
from PIL import Image

from skimage.transform import (hough_line, hough_line_peaks,
                               probabilistic_hough_line)
from skimage.feature import canny
from skimage import data

import matplotlib.pyplot as plt
from matplotlib import cm

"""The main idea of this script is to first crop the slopes
from the particles and then perform a probabilistic Hough
```

```
Transfrom on them. Then the slopes from every particle
in a measurement get saved in a list. I then average
these slopes and output a single average slope.

The input is the positionlist gotten from the previous
script, and this program will be run five times for
every positionlist.

We can plot all the slopes and their respective
Hough Transforms to check if the Hough Transform
is properly working, and if the coordinates of the
particles were correct."""
folder = 'E:/New_folder/measurements180507/10/Basler_daA1280-54um_(22148129)_20180507_120954627'
def cropping(plist):
    slopes=[]
    for i in range(0,len(plist)):
        img = Image.open(folder + '/background/BG' + str(i)+".png")
        i_list = plist[i]
        # input of list of coordinates
        for j in range(1,len(i_list)):
            area = (i_list[j][0][0],i_list[j][0][1],i_list[j][1][0],i_list[j][1][1])
            cropped_img = img.crop(area)
            cropped_img = np.asarray(cropped_img)
            cropped_img.setflags(write=1)
            low_values_flags = cropped_img < 80  # Where values are low
            cropped_img[low_values_flags] = 0
            np.save(folder +'/background/crops/BG' + str(i)+ '_' + str(j),cropped_img)
            edges = canny(cropped_img,2,1,25)
            lines = probabilistic_hough_line(edges, threshold=55, line_length = 4, line_gap = 3)

            arline=np.asarray(lines)
            slopelist = []

            for n in range(0,len(arline)):
                dx,dy = arline[n][0]-arline[n][1]
                if dx != 0:
                    slope = dy/dx
                if slope > 0:
                    slopelist.append(slope)
            if len(slopelist) != 0:
                slopes.append(np.sum(slopelist)/len(slopelist))
#             fig, axes = plt.subplots(1, 3, figsize=(15, 5), sharex=True, sharey=True)
#             ax = axes.ravel()
#
#             ax[0].imshow(cropped_img, cmap=cm.gray)
#             ax[0].set_title('Input image')
#
#             ax[1].imshow(edges, cmap=cm.gray)
#             ax[1].set_title('Canny edges')
#
#             ax[2].imshow(edges * 0)
#             for line in lines:
#                 p0, p1 = line
#                 ax[2].plot((p0[0], p1[0]), (p0[1], p1[1]))
#             ax[2].set_xlim((0, cropped_img.shape[1]))
#             ax[2].set_ylim((cropped_img.shape[0], 0))
#             ax[2].set_title('Probabilistic Hough')
#
#             for a in ax:
#                 a.set_axis_off()
#
#             plt.tight_layout()
#             plt.show()
    return np.sum(slopes)/len(slopes)

a,b,c,d,e = cropping(array),cropping(array),cropping(array),cropping(array),cropping(array)
print((a+b+c+d+e)/5)
print(a,b,c,d,e)
```