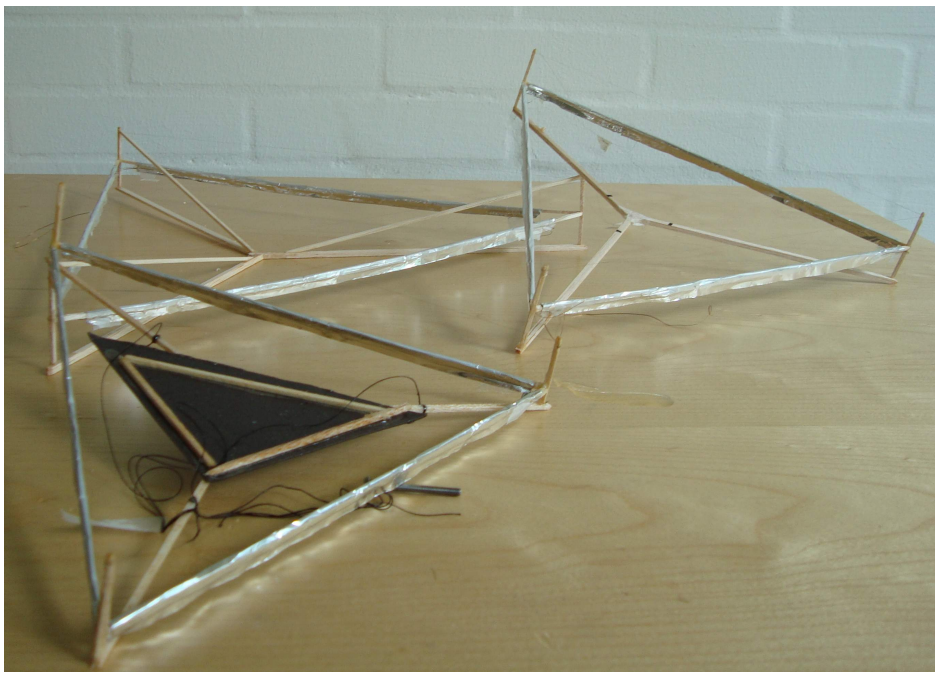
Nick P. Andersen & Kasper F. Larsen
s062117 & s062102

The Electrostatic Levitation Unit



Technical University of Denmark: FYS
Special Project, 10064
Councillor: Claus Schelde Jacobsen, Ole Trinhammer & Robert Jensen
June 23, 2008
All have contributed equally to all parts of the report.

Contents

List of Figures

List of Tables

Glossary

Notation	Description
α	Townsend coefficient
δ	Density factor specified by temperature and pressure
$\mathbf{E}$	The electric field
ϵ	The permittivity of the media
$\mathbf{F}$	The force on a particle or system
g_0	The electric breakdown for a media
g_v	The visual critical potential gradient
$\mathbf{J}$	The current density
N_e	Number of electrons in a given space
ρ	The space charge density
Φ	The potential in 3D
q	The charge of a particle
STP	Standard Temperature and Pressure, 25 °C, 1 atm
V	The potential in 2D
$\mathbf{v}$	The natural flow of a gas
V_c	The threshold potential, i.e. critical visual corona point

Motivation 1

The electric wind has been a subject of much fascination and confusion. Ever since its first description by Francis Hauksbee in 1709, [1], this phenomenon has fascinated with its promise of a simple and energy effective way to convert electrical energy into mechanical energy. Many exotic applications have been suggested; among them reduction of skin friction on aeroplanes, CPU cooling and levitational devices. Whether or not any of these applications are feasible is another discussion.

We have in this report focused on the investigation of a levitational unit — the electrostatic lifter ("lifter" hereafter). It is a simple construction made with a triangular frame and a thin wire cathode (corona wire) suspended over a grounded aluminum foil electrode wrapped around balsa sticks, separated by an air gap over which a large potential difference is applied, see Fig. 1 where the dashed line represents the corona wire, and the grayed out area is the aluminum foil. This causes the air to be ionized and then a downwards movement of the ions which impact other air molecules resulting in an upward thrust due too Newtons 3th law. The corona discharge is the term used for ion creation.

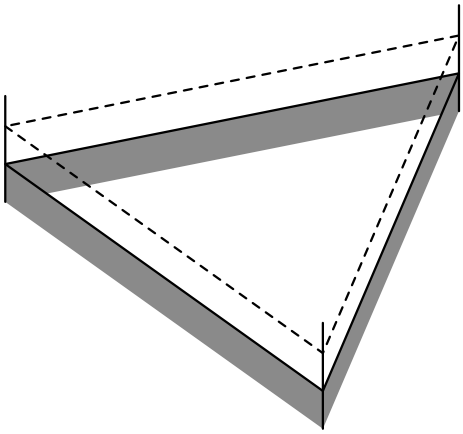


Figure 1: *A model of the lifter. The dashed line represents the cathode, and the grayed out area represents the aluminum foil anode.*

We wanted to investigate and, if possible, optimize the following, recognizing that in our case thrust is directly proportional to the corona discharge:

- Aluminum height
- Air gap between corona wire and aluminum
- Side length of the lifter
- Thrust characteristics of different dielectric media, oil and air
- V-I properties
- Resistance properties of air and oil
- Thrust vs. voltage
- The reproducibility of our measurements

In the process we have tried to optimize the above mentioned properties in order to generate the maximum amount of thrust, and as a sub project, we set up an automated system for making measurements using LabVIEW, and thoroughly investigated the properties of a "home made" high voltage transformer whose characteristics we needed for further data treatment. We have used electrostatic theory and some "common knowledge" about airflow in order to interpret our results and use them in the optimization process. In addition we have also tried to make a simple numeric simulation of the electric field present around our lifter. We have used information from [2], [3], [4], [5], [6] to build and optimize our lifter.

We have investigated the properties of the electrostatic lifter and found it to be significantly more efficient in oil than air, and that the efficiency does not depend as much on the size of the lifter as on the quality of the construction. Furthermore the relative humidity of air and flow patterns seem to have an effect.

Theory 2

Field Theory 2.1

In order to understand the mechanisms that will be used later, we now make a short introduction to the basic theory.

The exact setup of the device, which will be described later, will be approximated as a configuration of two infinitely long wires acting as electrodes, one very thin and the other relatively thick. A theory exactly matching these conditions has not been found. Which is why we will presume it to be a configuration two wires with equal radii. As an electric potential builds up over the gap between the two, it gives rise to a capacitance, thus creating an electric field. The potential will at some point reach the limit where there is a visual creation of ions. This is heard as a hissing noise and will be seen as a pale violet light in a total dark room. The value of the potential will be referred to as the corona onset voltage or *critical visual corona point*, V_c. At this voltage an electric field will have reached a certain strength, known as the *threshold electric field*, g_v.

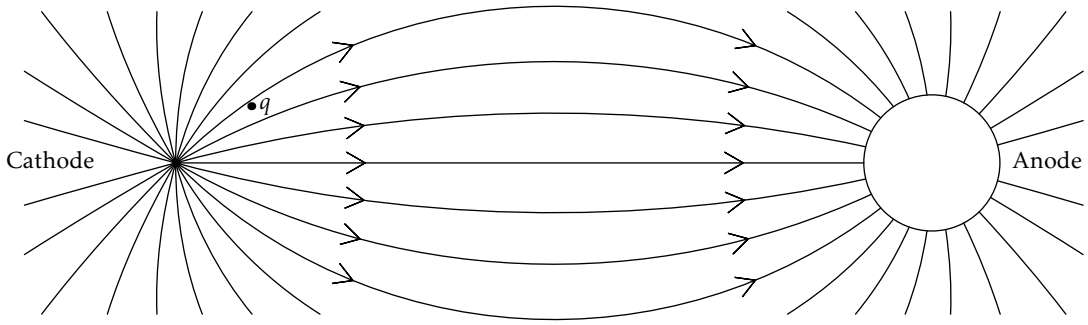


Figure 2: *A cross section of how the electric field lines lie in a thin-thick wire configuration.*

A figure of the model is seen in Fig. 2. The electric field lines are shown. The electric field is proportional to the "density of field lines", thus having larger values as the density increase. Notice that the field lines are very dense around the thin wire, the cathode. The corona discharge consist of gas-ionization which happens when molecules are present in high enough electric fields. What actually happens is that the molecules in the air are stripped of one or more electrons, thus making positive ions in proximity of the positive wire electrode (cathode), or have electrons added to them, making negative ions if it was a negative electrode (anode). This is due to the electrostatic forces acting in opposite directions on the negative electron and the positive nucleus, thus giving rise to an increasing dislocation of the two and eventually ending in total separation leaving a positive ion and a free electron. This creates a space charge of ions near the thin electrode wire adding their own electric field in the same direction as the existing field between the wires, which in effect increases the width of the conducting wire, [7]. The ions are now subjected to the forces of the electric field, creating a current in the air gap while the electrons create current in the wires.

Because the field is very intense around a "sharp" object, like the thin wire, the corona discharge can happen when a significantly less intense "average" field is present between the wires. For example, ionization of air happens at 30 kV/cm and if we have 2 cm between the wires one might expect that we would need

$$30\,\text{kV/cm} \cdot 2\,\text{cm} = 60\,\text{kV} \tag{2.1}$$

over the two wires. The field is far from linear, it's concentrated around the thin wire as seen in Fig. 2, which also will be shown in our simulation. Thus we can attain a *local field* very close to the thin wire of the necessary magnitude to ionize the air, when applying as little as 6 kV over the two wires, giving an average field of only 3 kV/cm. This will be shown in more detail in Sec. 2.4 and shown experimentally in Sec. 7.3. Surprisingly it turns out that the voltage needed to initiate the corona discharge, called corona onset voltage, V_c, is approximately independent of the distance between the two wires, except for very small distances, [4]. The reason why is made clear in Eq. (2.16).

If the electric field reaches the dielectric breakdown of air it will cause a lightning to jump. This causes large currents and actually creates a path of conducting plasma in the media. This is not a desirable situation as it means no resistance in the moment of lightning and a very large current.

EHD Flow 2.2

After the ionization of the molecules, they will be in the high electric field, subjected to a Coulomb-force, [8]

$$\mathbf{F} = q\mathbf{E} \tag{2.2}$$

where q equals the charge of the particle in the field $\mathbf{E}$. In our case the ions that have been created by the corona discharge. As the ions are accelerated and move towards the grounded (negative) electrode, in our case the thick wire, they will collide with air-molecules. This will result in a momentum-transfer from the ions to the air-molecules. Giving rise to a netto flow of air: the so called *ElectroHydroDynamic* (EHD) flow, which in the past was, and sometimes still is, referred to as "the electric wind" or "corona wind", [1]. The EHD flow is not a well-described quantity since a complete description would include both the influences of the static field caused by the potential difference between the wires, the static field caused by the space charge, the dynamic field caused by the flow of electrons, i.e. the current density, and the various hydrodynamic effects at play in airflows. An analytical treatment would be very complex, and even a numerical analysis would be difficult and is beyond the scope of this report.

The force exerted on the ions in order to accelerate them is countered by an equal and opposite force acting on the cathode and anode. In the case of the lifter the two wires are physically but not electrically connected in the lifter. According to Newton's third law the total force on the ion and the total opposite force on the lifter will be exactly the same.

Townsend Avalanche 2.3

As mentioned, the field, and thus the force on the created ions will be very large close to the thin wire, resulting in great acceleration giving the ions relatively large speeds before impact. This means that their kinetic energy occasionally becomes high enough to knock off electrons from the molecules they hit, which ionizes them too. These new ions will be subjected to the coulomb force themselves and begin accelerating. Assuming that we're still close to the electrode they might also reach velocities high enough to ionize more molecules and so on. This effect is known as the Townsend avalanche. This only takes place close to the electrode where the field is still strong enough to accelerate the created ions to the necessary speeds before impact. Accordingly this will result in an exponentially increase in ions.

$$N_e = c\exp\left(\int \alpha(r)\,\mathrm{d}r\right) \tag{2.3}$$

Where α is the Townsend ionization coefficient, c is a proportionality factor and r is the distance from the original ion. In Sec. 8.4 we will assess the quantity of ions created in this process.

Electric Field Model 2.4

We now proceed to a more detailed description of the equations providing the basis of our EHD-flow. In order to describe the flow we need (ignoring the hydrodynamics of air) to find the potential and the space charge. The electric field between the corona electrodes is governed by the Poisson equation

$$\nabla^2\Phi = -\nabla\mathbf{E} = -\frac{\rho^2}{\epsilon} \tag{2.4}$$

where, Φ is the electrostatic potential in space, ρ describes the space charge density, $\mathbf{E}$ the electric field and ϵ is the permittivity of the ambient media.

The ions in the electric field creates a current-density, here referenced as $\mathbf{J}$, taking both the ambient gas velocity and its drift into consideration we get following equation.

$$\mathbf{J} = \rho(K\mathbf{E} + \mathbf{v}) - D\nabla\rho \tag{2.5}$$

where K is the mobility of the ions, $\mathbf{v}$ is the gas velocity and D is the diffusion coefficient of the gas. With steady-state conditions the charge density must be conserved, giving

$$\nabla\cdot\mathbf{J} = 0. \tag{2.6}$$

When assuming that the velocity of the gas is half the magnitude of the ions drift velocity, the term $\mathbf{v}$ can be neglected, [3]. Taking into consideration that the electric force on the ions must be much greater than the ions diffusion constant, therefore neglecting it, we get from Eqs. (2.5), (2.6) and (2.4) using regular divergence computation relations that

$$\begin{aligned}
\nabla \cdot \mathbf{J} = 0 &= \nabla \cdot [\rho(K\mathbf{E} + \mathbf{v}) - D\nabla\rho] && (2.7)\\
&= K\nabla\rho \cdot \mathbf{E} + K\rho\nabla \cdot \mathbf{E} && (2.8)\\
&= -K\nabla\rho \cdot \nabla\Phi - K\rho\nabla \cdot \nabla\Phi && (2.9)\\
&= -\nabla\rho \cdot \nabla\Phi - \rho\nabla^2\Phi && (2.10)\\
&= -\nabla\rho \cdot \nabla\Phi - \rho\left(-\frac{\rho}{\epsilon}\right) && (2.11)\\
&= -\nabla\rho \cdot \nabla\Phi + \frac{\rho^2}{\epsilon}. && (2.12)
\end{aligned}$$

Now we have a set of partial differential equations which can be used to find the space charge distribution and the potential. (2.4) being a linear second order equation for the potential, and (2.12) a non-linear first order equation for the space charge density. To solve these we need boundary conditions. The conditions for the potential are pretty straight forward, given the potential difference between the corona cathode and the anode, Φ_0, and zero voltage on the ground electrode. On the other hand, the boundary conditions for the space charge, ρ, are a bit more complicated. Kaptzovs' hypothesis is used, which suggests that the electric field is proportional to the potential difference applied, for voltages less than V_c. But that it can be approximated as being constant after the corona discharge is initiated, even if the voltage applied increases further, [5]. This gives us the boundary condition for the space charge on the cathode, i.e. a maximum just as the voltage reaches visual critical potential, V_c. The voltage gradient, being the magnitude of the electric field, however, is not constant throughout the space between the cathode and anode.

In order to determine the threshold strength of the electric field at the corona onset we use Peeks formula, [7]. Peeks formula assumes that the two wires have an identical radius. We were not able to find a formula for two different radii. We will have to settle for a minimum and maximum theoretical value of V_c , by calculating V_c for two thin and V_c for two thick wires. V_c is given by

$$V_c = m_v g_\mathrm{v} \delta r \ln\left(\frac{S}{r}\right) \tag{2.13}$$

where S and r are the distance between the electrodes and the radius of the wires respectively. m_v is an irregularity factor which accommodates the condition of the wire, for thin smooth wires $m_v = 1$. g_v is *the visual critical potential gradient* or the critical electric field when the corona first appears

$$g_\mathrm{v} = g_0\delta\left(1 + \frac{0.0301\,\mathrm{m}^{1/2}}{\sqrt{\delta r}}\right) \quad \text{, where } \mathrm{m}^{1/2} \text{ is a unit} \tag{2.14}$$

where g_0 represents the electrical breakdown for a media between two infinite plate electrodes. In pure air this is approximately 3 MV/m. δ is a density factor defined as (specifically for air), [7]

$$\delta = \frac{0.002\,94\,\mathrm{K/Pa} \cdot p}{T} \tag{2.15}$$

Be aware that the constant 0.002 94 K/Pa only represents pure air. For other media different values replace it.
It is noted that at STP (Standard Temperature and Pressure) we have $\delta \approx 1$.
In the before mentioned equations all parameters should be in SI-units. With the assumptions of STP and a perfect wire Eq. (2.13) is reduced to

$$V_c = r g_0\left(1 + \frac{0.0301\,\mathrm{m}^{1/2}}{\sqrt{r}}\right)\ln\frac{S}{r} \tag{2.16}$$

It is now possible to determine the threshold strength of the electrical potential, above which the corona discharge initiates. In the formula the radius, r, represents radii of both wires.

As seen in Eq. (2.16) V_c has a dependence of $\ln(S/r)$ which is why there is little difference in V_c when changing the distance between the electrodes, S, or the radii r, as long as $S \gtrapprox 200r$. But in our experiment $S \approx 10r$ for the thick wire, and $S \approx 800r$ for the thin. This is why why now calculate V_c for both two thin wires and two thick, and expect the actual V_c for a thin-thick configuration to be in between. V_c will be lower for two thin wires than a thick and thin, because they both will be creating ions which in turn will enhance the field and lower the V_c.

If we first consider two "thin" smooth wires of the same radii $r = 0.05/2\,\text{mm}$ and the distance between them $S = 2\,\text{cm}$, as was the case in our experiment, and then insert in Eq. (2.16) while assuming STP, we get

$$g_v = 3\,\text{MV/m}\left(1 + \frac{0.0301\,\text{m}^{1/2}}{\sqrt{0.025 \times 10^{-3}\,\text{m}}}\right) \tag{2.17}$$

$$= 21.1\,\text{MV/m} \tag{2.18}$$

which is the threshold electric field. Then we find V_c

$$V_c = r_c \cdot g_v \ln\left(\frac{S}{r}\right) = 0.025 \times 10^{-3}\,\text{m}\,21.1\,\text{MV/m}\ln\left(\frac{0.02\,\text{m}}{0.025 \times 10^{-3}\,\text{m}}\right) \tag{2.19}$$

$$= 3.5\,\text{kV} \tag{2.20}$$

So we have corona discharge at $V_c = 3.5\,\text{kV}$ and a threshold electric field of $g_v = 21.1\,\text{MV/m}$. In order to get an upper limit on our expected field we now calculate V_c for two "thick wires" of the same radii as the width of the aluminum wrapped around the balsa sticks on the lifter (the exact construction and properties of the lifter will be given in Sec. 3). Then Eq. (2.16) gives a higher V_c than needed to initiate the corona discharge. V_c for two thick wires with radii of 2 mm is thus calculated again using the same equations and gives 8.6 kV, and a threshold electric field of $g_v = 9.4\,\text{MV/m}$. Therefore we expect to find an actual V_c between 3.5 kV and 8.6 kV.

Lifter Construction 3

To build the lifter we choose a lightweight and easy-to-built construction. It's seen in Fig. 3. As a frame for the lifters we chose a simple triangular design consisting of a horizontal triangle made of $2\,\text{mm} \times 2\,\text{mm}$ balsa wood, glued together in each corner with a vertical beam. We used a kanthal wire of 0.05 mm diameter, going between the tops of the three vertical beams (barely seen on the figure) called the corona wire since it's were the corona discharge occurs. The distance between the wire and foil was 2 cm. Each horizontal beam was wrapped in aluminum foil and connected in each corner with small wires. All of the aluminum was then connected to ground, see the top left corner on the figure. It is also shown how the "foot" of the lifter was attached, the foot is marked with three black marks on it.

Resistance through the aluminum foil was in the area of $20\,\Omega$, and the corona wire had a known resistance of $695.0\,\Omega/\text{m}$. Because of the high voltages and low currents this seemingly high resistance didn't have any effect on the lifter. As is seen by the following example with a wire of length 3 m and a current 1 mA.

$$695.0\,\Omega/\text{m} \cdot 3\,\text{m} \cdot 1\,\text{mA} = 2\,\text{V} \ll \text{kV} \tag{3.1}$$

Thus the resistance in the corona wire and aluminum foil can easily be ignored.

As described in the theory the potential difference between the top cathode corona wire, and the grounded anode aluminum foil, will result in a downward EHD airflow. Since the cathode and anode are physically connected in the lifter they will be subject to the described upward force, however as some of the air molecules and the ions impact with the anode, they will transfer some of their momentum back to it and thus generating a downward force on the lifter. But as long as the total momentum transfered to the air is greater than the momentum transfered back, there will be an upward thrust.

In this report we use gram (g) as a measure of thrust, as it was the unit measured. We note that $1\,\text{g} \approx 0.01\,\text{N}$.

We wanted to keep V_c constant over the whole length of the sides of the lifter, both to make sure that the lift was somewhat even and to make a more stable lifter. We also wanted to avoid V_c being so low locally that an "ion channel" could form with ions running from the corona wire to the aluminum, seen as a purple glow, without colliding with the air, thus "wasting" their momentum. This meant trying to have as smooth a surface on the aluminum foil as possible and a corona wire without "kinks" in it. Any small "pointy" structures would lower the local V_c as seen in formula Eq. (2.16) since "pointy" structures have smaller r. This glow was frequently seen at the ends of the aluminum. It was the most common place we saw the glow. In addition a too low V_c also made it

easier for sparks to jump from the corona wire to a small tip at the end of the aluminum. As it turned out it was quite difficult to apply the foil without wrinkling it. We invariably had a lot of charge losses due to the effect just described. Some of this could be recovered by looking for the "telltale" purple corona discharge glow on the lifter in a dark room and cover the spots with glue. Despite our best efforts it turned out that there was still a significant difference among the different lifters as to how efficient they were.

Figure 3: *This is how the actual lifter looks like. Notice the "foot" which is used when measuring the lift. The corona wire can barely be seen on the figure,*

Simulation of Potential 4

In dealing with corona discharge we needed to assess the quantitative characteristics of the electrostatic field, which determines if the corona discharge initiates. We wanted a simple model of the voltage contours between the corona wire and the ground electrode, and used the well known *method of relaxation*, [8, Sec. 3.1.3]. For simplicity the computations are entirely done in 2D. The method is based on the following.

The value of V at one point in a mesh of calculation points is equivalent to the mean of the surrounding points. Therefore an integral along a circle around the point divided by it's area equals the mean around the point.

$$V(x,y) = \frac{1}{2\pi R} \oint V \, \mathrm{d}l \tag{4.1}$$

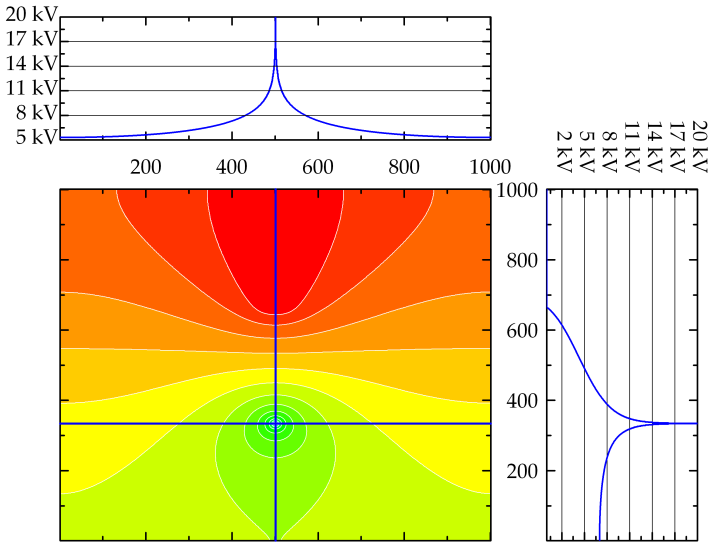
where R is the radius of the integral circle. Using Eq. (4.1) and with the boundary of our mesh *loose*, i.e. without boundary conditions on the edge of the mesh, and calculating only for 4 points on the circle, we write the following

$$V(x,y) = \frac{1}{4}\left[V(x+1,y) + V(x-1,y) + V(x,y+1) + V(x,y-1)\right]. \tag{4.2}$$

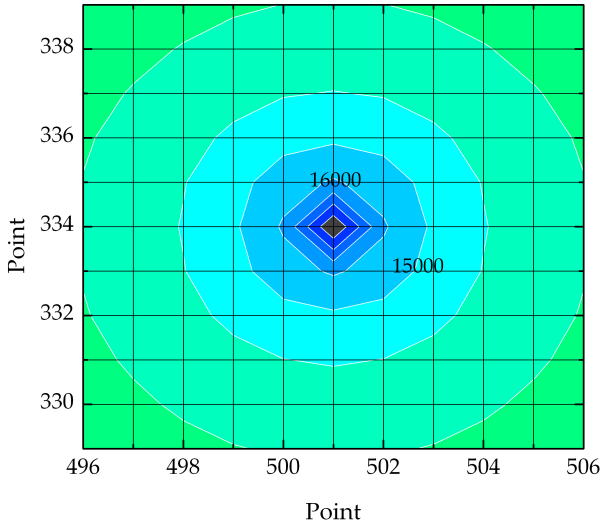
We now approximate the circle by a square, and expand the equation to hold the coordinates $V(x \pm 1, y \pm 1)$ as well, multiplying each adjacent point by 1/6 and each diagonal point by 1/12

$$\begin{aligned} V(x,y) = {} & \frac{1}{6}\left[V(x+1,y) + V(x-1,y) + V(x,y+1) + V(x,y-1)\right] \\ & + \frac{1}{12}\left[V(x+1,y+1) + V(x+1,y-1) + V(x-1,y+1) + V(x-1,y-1)\right] \end{aligned} \tag{4.3}$$

The 9 points are thought to give a more accurate picture of the situation than Eq. (4.2). The entire program is seen in Appendix D, where the computations takes place in the class Poisson, D.2.1. The mesh is a 1000×1000 grid which has a length between two points of 60 µm. This is calculated on the basis that the distance between the corona wire and the ground electrode is 0.02 m separated by 333 points. And by thus having the corona wire filling one point it will have a diameter of 0.06 mm, which corresponds very well to our real diameter of the wire, 0.05 mm. The iteration process has predefined boundary conditions at the ground electrode with a voltage level equal to 0 V, and at the corona wire 20 kV.



(a) *After one million iterations the voltage drop is shown in the mesh. The horizontal and vertical blue lines indicate the top and right plot respectively. Each contour line is separated by* 1000 V.



(b) *The field's properties are shown around the corona wire. Each cross in the grid represents a point in the mesh. Notice that the voltage drops to just above* 16 kV *at the points next to the corona wire. The above grid represents an area of* 0.6 mm × 0.6 mm.

Figure 4: *The mesh is a* 1000 × 1000 *grid. The distance between each point represents* 60 µm*, calculated on the basis of a total distance of* 0.02 m *between the ground electrode and the corona wire. Thus the total area of the grid is* 60 mm × 60 mm.

Fig. 4 shows how the situation looks after one million iterations. Each contour line represents a difference of 1 kV and can thereby give us an indication of the field strength around the corona wire. In Fig. 4a we see the entire picture. It includes a profile plot of the contours of the voltage in the horizontal and vertical direction across the corona wire. The top plot is a profile along the horizontal line, and the right corresponds to a profile along the vertical line. It is clear that the gradient close to the corona wire is very large. Almost an instantaneously drop of several kV. The approximation of the gradient very close to the wire is seen in Fig. 4b, where we have zoomed in on the high voltage electrode. It is readily shown that the electric field (voltage gradient) is about 4000 V/60 µm ≈ 66 MV/m in the near vicinity of the wire, by going one point to the left which decreases the voltage of 4000 V, and knowing that the distance is 60 µm. This is of course a significant approximation, but for our purposes it is accurate enough.

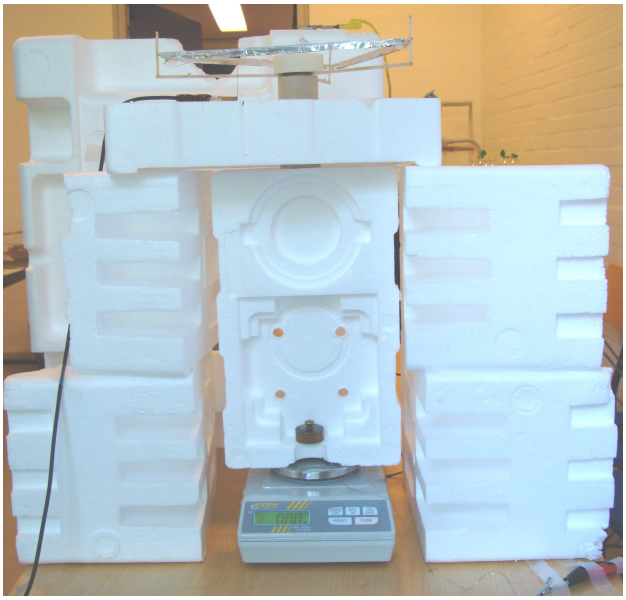
If we compare with the results of the theory, Eq. (2.20), we know that the real value of visual critical potential gradient, g_v, should be in the range between 9.4 MV/m and 21.1 MV/m. Therefor the above simulation shows that the visual critical potential gradient should be reached well below a potential difference of 20 kV. We also showed, in the theory, that the potential across the cathode and anode should have a value in the range of 3.5 kV and 8.6 kV.

Experimental Setup 5

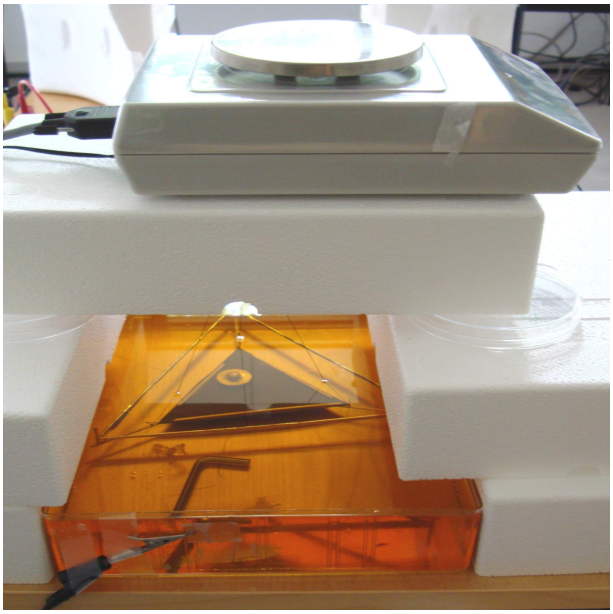
Air 5.1

After having seen that the electrostatic lifter was able to fly we wanted to get more quantitative measurements of the thrust generated. Our setup of the experiments was as follows, see Fig. 5a. We measured the thrust by attaching a "foot" consisting of balsa wood beams extending from under each corner pole and meeting up in the middle of the triangle, see Fig. 3. The foot was placed on a cardboard tube standing on a 40 cm polystyrene slab resting on a scale. Then we placed a plate supported by "legs" on the table, with a hole in it for the cardboard tube, between the lifter and the scale. This was done to avoid having the airflow generated by the lifter pushing down on the scale which would distort the readings. The 40 cm polystyrene slab was there to distance the electrical field generated by the lifter, and in particular the field generated by possible sparks, from the electronic scale. The interference

from the fields in the scale turned out to be a significant yet systematic error source which we accounted for in our treatment of the data. On top of the lifters foot we placed a relatively heavy object so that the lifter was at all times sitting on the cardboard tube. This meant that when the lifter started lifting, we would measure a weight on the scale equal to the lift.



(a) *Our setup of the lifter in the air test. It's seen how we have separated the scale from the lifter with polystyrene. This was to avoid high electric fields in the vicinity of the scale.*



(b) *It's here seen how the setup is in oil. The oil is regular canola oil sold in supermarkets. Underneath the scale there was a screw that the lifter was suspended from using thread, which made it possible to weigh it.*

Figure 5: *It's shown displayed how our setup was in the air and oil testing-environments.*

Oil 5.2

As we wanted to experiment with the lifter in other dielectrical media than air, we thought of oil. We came up with the setup shown in Fig. 5b. There we have the scale on a polystyrene box with a hole in it. Underneath it there is a screw attaching the object. We wound three threads around the screw, which were attached to the lifter in each corner so that it was leveled in a horizontal position. The lifter was then submerged in oil.

As balsa wood is quite light the lifter was buoyant so it needed a lot of weight to keep it down, approximately 100 g. As the lifter started lifting the scale would measure the change in weight.

Electrical Circuit and Measurement Precision 5.3

The setup for measuring the different voltages and currents was as follows, see Fig. 6. We had a data card supply a voltage between 0 and 10 V, V_{PC}. Next there was an amplifier, V_{amp}, which had an amplification approximately a factor of 2. The current was then measured by running the output from the amplifier through a 0.4 Ω resistance and measuring the voltage difference over the resistance, V_{amp}, then calculating the current dividing by 0.4 Ω by using Ohms law. Due to limitations on the amplifier, which was only able to amplify up to a voltage of 20 V, we then ran the signal through another voltage supply, V_{raise}, that increased the voltage by a pre-set amount. This was in order to take advantage of the transformers input range of 0 to 24 V. Just before the transformer the signal was split into two parallel paths one sending the signal through a 20 kΩ then a 10 kΩ resistor in series and on to ground. We measured the voltage difference over the 10 kΩ resistor with the data card, V_{in}. Since this voltage is one half of the voltage over the 20 kΩ resistance we then knew what the voltage was in the other path relative to ground, because of the voltage distribution of resistors in parallel coupling. The other path sent the signal to the transformer where it was transformed with a factor of about 1000. On the high voltage side of the transformer the current was split into two paths. One is called the monitor side which was used to find the exact value of the high voltage, and the other was the path to the lifter.

Instruments used are listed in Appendix A.

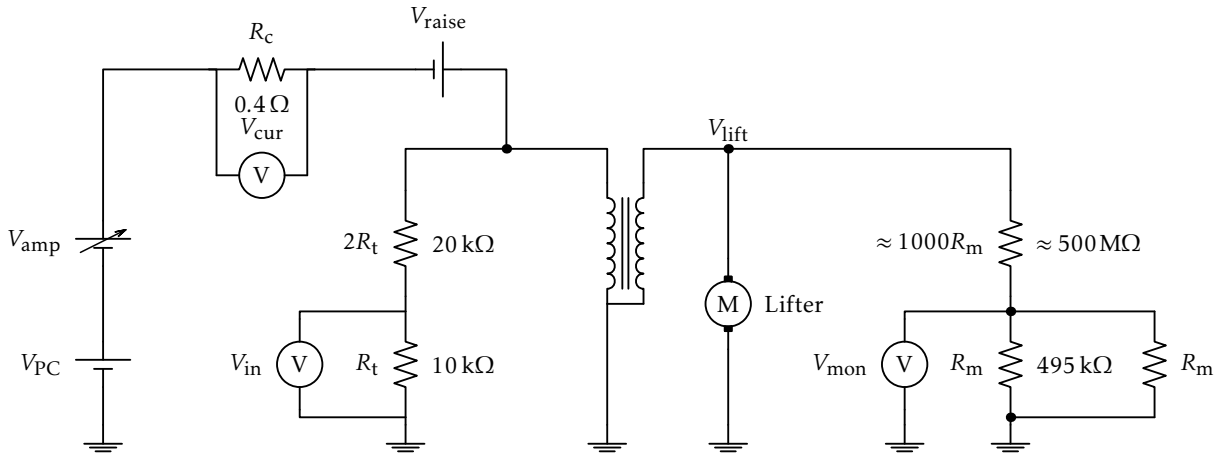


Figure 6: *The electronic circuit of our setup. The* M *represents the lifter as a motor.*

In our data treatment we have assumed that the errors on our measured values are independent, so we have used the Pythagorean relations between the errors. These error analysis computations, [9], will not be explained in detail but the relations used are listed here.

$$Z = A \pm B \qquad (\Delta Z)^2 = (\Delta A)^2 + (\Delta B)^2 \tag{5.1}$$

$$Z = A \cdot B \qquad \left(\frac{\Delta Z}{Z}\right)^2 = \left(\frac{\Delta A}{A}\right)^2 + \left(\frac{\Delta B}{B}\right)^2 \tag{5.2}$$

$$Z = \frac{A}{B} \qquad \left(\frac{\Delta Z}{Z}\right)^2 = \left(\frac{\Delta A}{A}\right)^2 + \left(\frac{\Delta B}{B}\right)^2 \tag{5.3}$$

Our further calculations using the above mentioned equations have assumed that the errors on the measurements of the data card are so small compared to all other errors that we can consider the data card readings exact for all practical purposes. The uncertainties on the various measured values are listed in Tbl. 1. The resistors and the scale have their error printed on them. We have based V_{lift} on the relative difference between measured values and the readings on the monitor side. These errors are shown in future graphs by error bars.

Table 1: *Our known errors on specific components in our setup.*

Part	Known error range
R_c	$\pm 2\,\text{m}\Omega$
R_t	$\pm 0.1\,\% \approx \pm 0.01\,\Omega$
R_m	$\pm 0.1\,\% \approx \pm 0.5\,\text{k}\Omega$
V_{lift}	$\pm 1.5\,\% \leq \pm 300\,\text{V}$
W	$\pm 0.01\,\text{g}$

Data Gathering 6

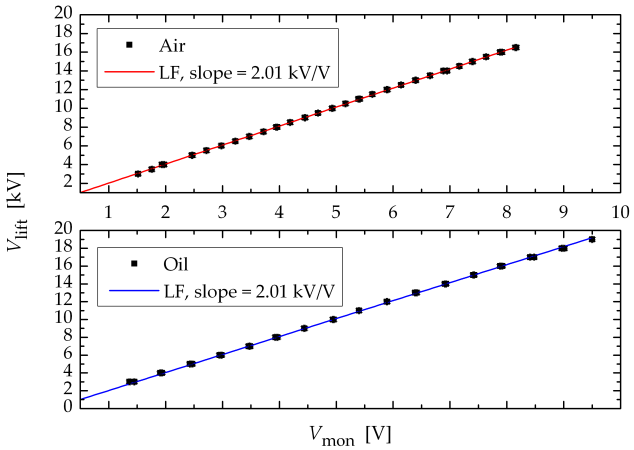
Transformer 6.1

As seen in our electric circuit we had a transformer which supplied a high voltage. This consisted of a "black box" from an old computer monitor's high voltage supply. It was assumed to contain a large number of wire coils with a wire wrapped around it a number of times. Some electronics were placed in the transformer to avoid undesired effects like voltage backfeeding. The transformer got a supply voltage V_{in} and gave a high voltage output, V_{lift}, parallel with a monitor output, V_{mon} which was $\approx 1/2000 V_{lift}$, see Fig. 6. The transformer had a maximum allowed V_{in} of 24 V.

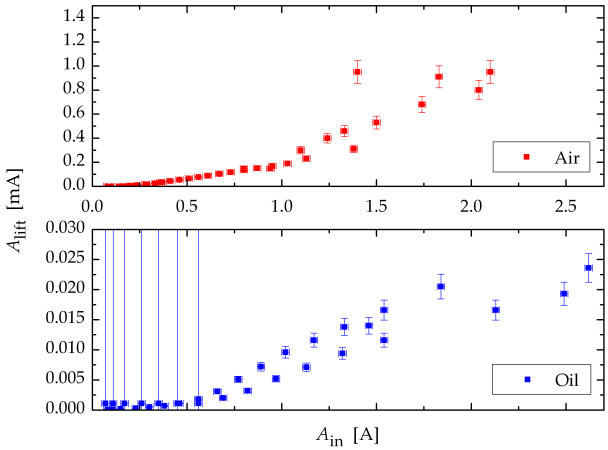
In order to analyze the lifters we had to characterize the transformer. By measuring the monitor voltage V_{mon}, the monitor current, A_{mon} , the high voltage over the lifter, V_{lift}, and the current through the lifter A_{lift} we could establish a relation for the transmission coefficients throughout the entire range of input.

In this section the following measurements were taken with a lifter connected to the transformer, so we would have the exact same situation as when gathering future data. The voltage and current readings were manually read from an (inexpensive) amp meter. Because there was a possibility of enormous currents running through the equipment if sparks jumped on the lifter, which consequently would destroy the amp meter. We would therefore not risk the rather expensive instruments in these measurements. Sparks and lightning occur when the electric field reaches the dielectric breakdown of air, as is mentioned in the theory.

Ideal transformers are completely linear in terms of current and voltage input/output. In our case the transformer was not linear. We measured the output of the monitor side, V_{mon}, to be close to 1/2000 of the high voltage output, V_{lift}. The resistance on the monitor side was approximately 500 MΩ so that absolutely no current should run. As it will be shown later the resistance in air was measured to 10 MΩ and for oil 400 MΩ, through the lifter. Thus the current through the monitor side, A_{mon}, was negligible for air but almost equal to A_{lift} in oil.



(a) *Voltage across lifter, V_{lift}, vs. the monitor voltage, V_{mon}. It is readily shown that a perfect linearity is present. With slopes just above 2.0 kV/V, which is very satisfactory. Note that the errorbars are so small that they are hardly visible. This allows us to rely on the monitor voltage to be exactly 1/2000 of V_{lift}.*



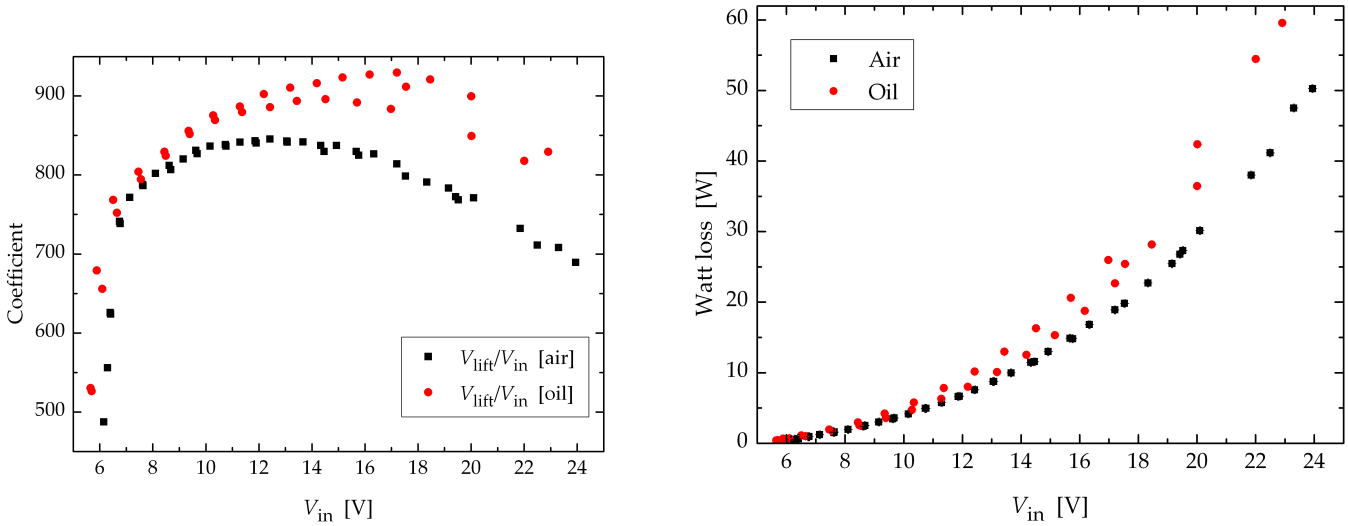
(b) *The current through the lifter, A_{lift}, vs. the current into the transformer, A_{in}, displayed for both oil and air. Notice the difference in y-axis scale which clearly demonstrates that the current through the lifter in oil is much less than in air. It is approximately 40 times less current in oil.*

Figure 7: *The voltage and current characteristics of the lifter. Notice in (b) that the current through the lifter, A_{lift} , in oil is much less than in air. This is because the resistance on the monitor side is similar to that of oil, so the current through the monitor, A_{mon}, is of a similar magnitude to A_{lift}.*

In Fig. 7 we have voltage and current characteristics for the lifter, Fig. 7a and Fig. 7b respectively. These plots gives us direct information as to how the transformer responds before and after the corona discharge initiates. The first graph shows the linearity between the monitor voltage, V_{mon}, and high voltage output through the lifter, V_{lift}, both for oil and air. It is easily seen that a linearity exists between the two values, and that the slope corresponds to the value of 2.0 kV/V. Linear fits will be referenced to *LF* in graphs. We found that the value V_{lift} is about 2000 times the value of V_{mon}. So by measuring V_{mon} we could calculate V_{lift}.

We expected the current into the transformer, A_{in}, and the current through the lifter, A_{lift}, to be linear. But as seen in Fig. 7b we find that a linearity is not present. A_{lift} is related to the corona discharge, because of the current being ions moving from cathode to anode. In the graph it looks as if the ohmic resistance in air seems to fall. This is in accordance with the Townsend Avalanche effect as discussed previously. Which means an exponential growth of ions. We see that around $A_{\text{in}} = 0.9$ A for air, and $A_{\text{in}} = 0.6$ A for oil, the growth pattern changes. This is, as just mentioned, due to the fact that at around these voltages we saw a force on the lifter indicating corona discharge. The current transmission coefficient can therefore be defined in two separate cases, one before the corona discharge and no ions resulting in very small currents, and one with corona discharge creating greater currents. As to why there is a small current running before the corona discharge is reached is hard to answer. A good suggestion would be charged carried away by the humidity in air, or water molecules dissolved in the oil. Other suggestions might be that there are several kinks in the aluminum and corona wire that locally decreases the corona onset voltage, or maybe the balsa wood's resistance is not that different from 500 MΩ, although it is thought of as an insulator. In any case, we are mostly interested in what happens after corona discharge.

The transmission coefficient is remarkably different at different input voltages and is shown in Fig. 8a. It is clearly noticeable that the coefficients are not constant. We get an optimum transformation around an input voltage of $V_{\text{in}} = 12\,\text{V}$ in air.



(a) *Transmission coefficients for $V_{\text{lift}}/V_{\text{in}}$ vs. V_{in}. It gives a maximum coefficient at $V_{\text{in}} = 12\,\text{V}$ in air. This shows the non-linearity of our transformer.*

(b) *Power loss of the transformer in watts, with respect to the voltage V_{in}.*

Figure 8: *It is shown how the transmission coefficient is for oil and air in (a).*

The mentioned properties should not surprise since the transformer was not built to be neither linear or efficient. It was specifically built just to get a high voltage output, no matter what. The inefficiency can also be seen in the power loss of the transformer as seen in Fig. 8b. The watt loss is calculated with the following equations

$$W_{\text{in}} = V_{\text{in}} A_{\text{in}} \tag{6.1}$$

$$W_{\text{out}} = V_{\text{mon}} A_{\text{mon}} + V_{\text{lift}} A_{\text{lift}} \tag{6.2}$$

$$W_{\text{loss}} = W_{\text{in}} - W_{\text{out}}. \tag{6.3}$$

It is shown in the graphs that the power loss in the lifter is close to 0 W for less than $V_{\text{in}} = 6\,\text{V}$, but rises as the V_{lift} increases. This indicates a relatively low zero load loss in the transformer. It should be noted that the power loss happens when the corona discharge initiates, around $V_{\text{in}} = 6\,\text{V}$, which is when the lifter starts producing force. This happens because, before the resistance in air and oil is immense before corona onset, so essentially no current flows. As the corona discharge creates ions and causes them to move in the air-gap, the ohmic resistance drops as the space-charge in the region increases. This causes current to flow through the lifter. As explained by the Townsend Avalanche. It is also seen later on in Fig. 14, that the measured resistance does drop significantly as the corona discharge increases.

LabVIEW Instrumentation 6.2

When dealing with experiments one often encounters the problem of measuring different variables at the same time in a precise manner. Consequently such experiments are best handled by an automatic computerized setup. We quickly discovered that attaining precise measurement data was almost impossible. For example the manual setting of the DC-supply was simply too inaccurate. Another issue was that the corona discharge created an amount of ozone significantly beyond what was healthy, so it was beneficial for our health to have an automated system allowing us to leave the room during measurements. We created a program for data collection. This was done in the graphical program-language LabVIEW, that allows one too quickly create a usable and very precise program for data gathering.

The LabVIEW implementation was an optimal setup for gathering sufficiently precise data. The instruments are listed in Appendix A. We used an NI-data card which could be expanded with voltmeters. On the data card we attached three input channels and one output channel and we connected the scale separately through the serial

port, so that all parameters could be measured simultaneously from within LabVIEW. In Tbl. 2 it's shown how the expected output/input ranges vary as well as the possible measurement ranges. All the parameters are in the circuit Fig. 6 except the weight which was not part of the electric circuit, but was still a measurable quantity. V_{PC} was limited by the computer output. V_{amp} and V_{lift} were preset by the user so that there was total control of the input into the transformer. All of the other ranges are determined by limitations on the instrument or variables nested in the coded program. The resistors' precision are listed in Tbl. 1. The possible errors on the measurements on the data card was negligible.

Table 2: *All data acquisition is listed here in their expected and possible measurement ranges.*

Part	Expected value ranges	Capable measurement range
V_{PC}	0–10 V	–
V_{amp}	$\approx 2 \times V_{\mathrm{PC}}$	–
V_{cur}	0–1.12 V	−1.5 V – 1.5 V
V_{raise}	0–10 V	–
V_{in}	0–8.3 V	0 – 10 V
V_{mon}	0–8 V	0 – 14 V
W	–	0 g – 600 g

As it is seen in the circuit, Fig. 6, we have one output-channel, V_{PC}, and at three other locations in the circuit we connected the input channels. The three input-channels called, V_{cur}, W and V_{mon}. V_{cur} was for measuring the current into the transformer. This was done in the program by using that

$$V = R_{\mathrm{c}} I \qquad \text{, and that } R_{\mathrm{c}} = 0.4\,\Omega. \tag{6.4}$$

Knowing the current we lifted the potential with a manually set voltage, V_{raise}, using another voltage supply. Because of the limitations of the transformer the voltage on the input should not exceed 24 V. To precisely know what voltage was supplied to the transformer we used a configuration of two resistors. One of the resistor's resistance was double that of the smaller resistor R_{t}. We then knew that the total voltage supplied to the transformer, V_{in}, was three times the value measured over R_{t}. The diagram of the transformer can be seen in Appendix B. On the high voltage side the transformer supplied a voltage, V_{lift}, which will be used quite frequently throughout the data treatment and analysis. However V_{lift} was far to high to be safely measured by the data card. To be able to measure these high voltages we needed a path parallel to the one through the lifter, and one with very large resistors. The parallel resistors' total resistance was approximately 500 MΩ. That ensured a large enough potential drop over the resistors to allow us to measure V_{mon} with the data card. We found that V_{mon} is 1/2000 of V_{lift} by:

$$V_{\mathrm{lift}} = I\left[1000 R_{\mathrm{m}} + \left(\frac{1}{R_{\mathrm{m}}} + \frac{1}{R_{\mathrm{m}}}\right)^{-1}\right] \tag{6.5}$$

$$= I\left(1000 R_{\mathrm{m}} + \frac{R_{\mathrm{m}}}{2}\right) \tag{6.6}$$

$$= 1000 R_{\mathrm{m}} I + \frac{R_{\mathrm{m}}}{2} I \tag{6.7}$$

We can now find the relative difference, by defining $V_R = 1000 R_{\mathrm{m}} I$

$$\frac{\frac{R_{\mathrm{m}}}{2} I}{1000 R_{\mathrm{m}} I} = \frac{V_{\mathrm{mon}}}{V_R} \qquad \Rightarrow \qquad V_R = 2000 V_{\mathrm{mon}} \tag{6.8}$$

and using Eq. (6.7)

$$V_{\mathrm{lift}} = 2001 V_{\mathrm{mon}} \approx 2000 V_{\mathrm{mon}} \tag{6.9}$$

Thus we were able to calculate the voltage over the lifter, V_{lift}, see Fig. 7a. We wanted the program to continuously vary the voltage output V_{lift} in small increments. Thus increasing V_{in} in steps until the limitations of the transformer was reached and then decreasing it again, while continuously collecting data. In this way the data should be more reliable, than manual measurements, and provide more information about the lift in relation to V_{lift}. The implementation of automatic readings drastically increased the amount of data collected, thus creating statistically

better results. We implemented a function that gave us the ability to set the number of voltage cycles passed, with a user setting of the V_{in} increments.

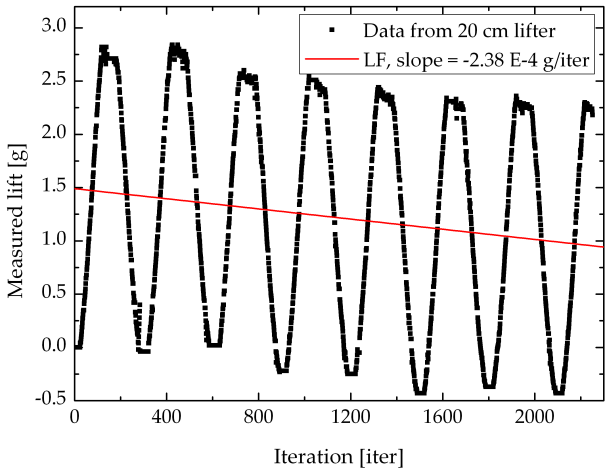
We quickly discovered that the sampling rate of the scale was rather slow. Therefore the program is designed to accommodate the limitations of the scale, this resulted in waiting approximately 600 ms between each voltage step. The program did 100 measurements of all input channels (except the scale which only took one) every 600 ms with a sampling rate of 200 pr. sec., giving the mean of these hundred data points as output. This meant that the precision of the data could only be increased by lowering the step size of V_{in} due to the slow sampling rate of the scale.

After collecting the data a file was created with all the measured data.

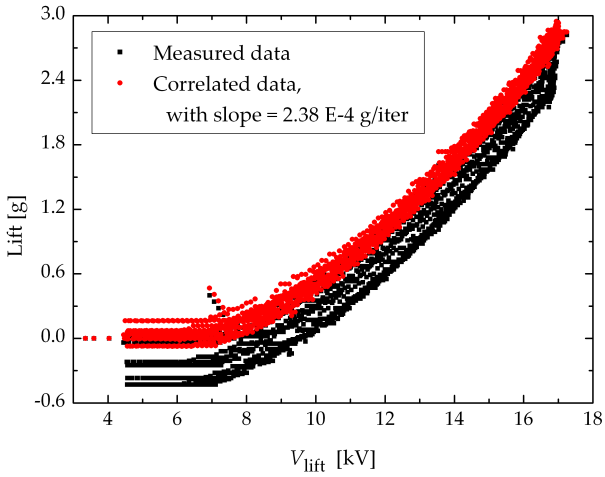
Data Correction 6.3

After the data collection we had a large amount to be treated. In examining the data we found that the zero point of the scale decreased with each measurement. This leads to the conclusion of an interfering electric field disturbing the scale. When increasing the distance from the scale to the lifter we experience a less significant offset. This confirmed our beliefs.

On Fig. 9a we see a measurement of the medium size lifter (26 cm), the other size lifters exhibited the same tendency. As the iterations rose we also saw a net decrease in the zero point of weight. If we assume this offset decreases linearly with the iterations, we can eliminate the vertical offset by fitting a linear function to it and adding it to the measurements. Even if the offset might not be perfectly linear this still seems like the best option. The fitted line has a slope of -2.38×10^{-4} g/iter which is shown by the red line in Fig. 9a. On Fig. 9b we see the correction of the measurements as well as the real measurements. Notice the small spread that emerges from the correction of the weight. These new values are seen to have less deviation from the mean, and we will therefore assume that our treatment is a good correction. All subsequent figures or data remarks will be using the correlated weight.



(a) *Lift against iterations. As it is seen there is a tendency that the zero point of the scale wanderers down. Therefore we correct the measurements by fitting a linear function to the decrease, which we add to the measurements, causing the plot to be aligned at $y = 0$.*



(b) *The corrected lift displayed with the lift against the voltage, V_{lift}. The black is the actual lift measured and the correlated weight is the red. It's clearly seen how the correction improves the standard deviation.*

Figure 9: *The above figures show how the lift is corrected, first by fitting a line to the decreasing offset, and then adding the line to the data.*

As seen on the graph the thrust increases with a slightly exponential tendency as V_{lift} increases. As expected the thrust starts around 7 kV, which is of the magnitude predicted by the theory Sec. 2 and our simulation Sec. 4. Of course this is since no corona discharge occurs at lower voltage differences, and no lift is created.

Range Problem 6.4

As we were almost done with all data gathering we discovered an error in our data-ranges. We initially thought that the card reading of V_{mon} had a capable upper limit of 20 V. But we realized that the actual range of measuring was $0-14$ V. And because V_{mon} went from $0-20$ V and thus over the capable range we had to rethink the circuit. It could be solved by measuring V_{mon} over two resistors in parallel connection instead of one. Eq. (6.7) assumes two resistors in parallel connection. This is the reason why we have several thrust measurements for $V_{\text{lift}} = 14$ kV on figures Fig. 15, Fig. 11b and Fig. 12. The max thrust measured is still correct, albeit at a wrong V_{lift}. The result of the range being to low was that the card thus just gave the upper limit of measurement, why V_{lift} turns constant around the 14 kV. This error did not have an effect other than giving the wrong V_{mon} value, and thus a wrong V_{lift} value. It should be noted that the setup in Fig. 6 is the corrected one. We have explicitly mentioned where the described error applies in the figure captions.

We did not have enough time to remeasure all the data, so we were forced to use the incorrect data.

Optimization of Lifter 7

As we wanted as much as possible thrust from the lifter, we wanted to optimize several parameters of the lifter. These being

- Air-gap between the corona wire and the aluminum foil
- Height of the aluminum foil
- The side length of lifter

Optimal Air Gap 7.1

From Eq. (2.16) we expected greater lift due to the higher field gradient for smaller distances, which we also saw in our experiments. However to avoid problems with sparks we decided on a fixed length of 2 cm assuming the spark breakdown voltage in dry air is 30 kV/cm. The breakdown voltage varies with humidity, and can increase with up to 6%, [10]. The breakdown voltage also varies with the smoothness of the aluminum foil. This is due to the discussion made in the theory. A significant increase in the field gradient increases the number of ions, making it easier for sparks to jump. We also used this seemingly large distance because the wire was drawn towards the foil due to the Lorentz force. As it turned out even with a distance of 2 cm we still experienced occasional sparks jumping between the wire and foil. So the optimum distance was dictated by practicality.

Optimal Aluminum Foil Height 7.2

To determine the optimal height of the aluminum foil we did a simple comparison of the maximum thrust possible with the same lifter by using different heights of aluminum foil. At the time we did not have access to a scale, so we set up a lifter with foil on one side only, and then balanced the lifter so that it would tip over when the same amount of gross thrust was present in each trial. We did the test with different heights. This way we also took into account the weight of the foil as it had to lift "it's own weight" in order to tip the lifter. We wrote down the magnitude of the current when the lifter tipped over, assuming that an equal thrust generated at a lower current indicated a higher possible thrust, and thus a greater efficiency.

Initially we assumed that the height of 2.5 cm as suggested in [2] and [3], would be optimal, but after trying a shorter length it turned out that it had better thrust. We then thought that an absolute minimum of foil would be better, but as shown in Fig. 10 it turned out that a height of 0.8 cm was optimal. We suggest this is due to the fact that there is less aerodynamic turbulence when the foil goes beneath the balsa stick acting as an air foil, which might allow more air to pass the beam without transferring its momentum to it by colliding with it, or at least changing the angle of impact so it transfers less momentum. An increasing height of aluminum foil would mean a disproportional additional amount of weight compared to how much airflow is improved, thus our optimum height was found and applied.

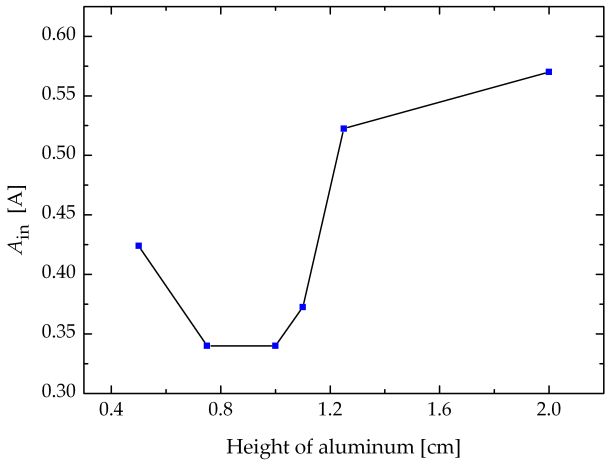


Figure 10: *The optimal aluminum height is found. A simple test were the lifter had to lift itself as well as the aluminum, vs. the current running in the transformer, generated the result of an optimal height of* 0.8 cm.

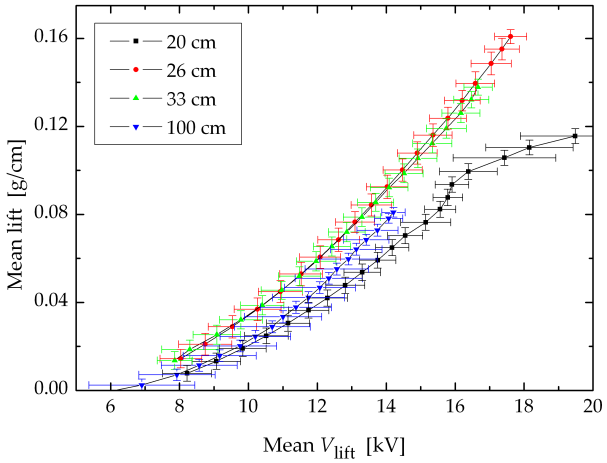
Size Dependence 7.3

We had a theory that a larger lifter would be either more or less efficient in terms of grams of thrust pr. cm than a smaller lifter. We proposed that the extra surface area of a large lifters' corona wire would mean a larger space of high electric field gradients, which should result in more corona discharge and a greater thrust which might or might not make up for the additional weight of the extra wood and aluminum. However as seen on Fig. 11a the expected trend was not there. This indicates that the effects on the lift from the irregularities of the foil and wire on each lifter were greater than any possible effects of the length of the sides. Otherwise we would have expected the biggest lifter to be either the best or worst and the opposite for the smallest. It should be noted that the biggest lifter never had a spark jump. In some way this indicates a more stable lifter. The reason for this is unknown. The only thing we've noticed is that the corona wire was pulled more toward the aluminum than for the other lifters, lowering the corona onset, but also making the distribution of the voltages along the wire change. Thus not allowing the sparks to jump at the ends of the aluminum, as explained in Sec. 3. Another interesting thing to note is that the thrust, and therefore the corona discharge, initiates at $V_{\text{lift}} = 6\,\text{kV} = V_c$ over the lifter, which is right in the middle of the expected range of 3.5–9.4 kV which we had calculated for our lifters configuration in Sec. 2.4. This confirmed that the approximations we made were valid, since the experimental value is right in the middle of the expected range. We were very pleased considering the rather significant assumptions we made.

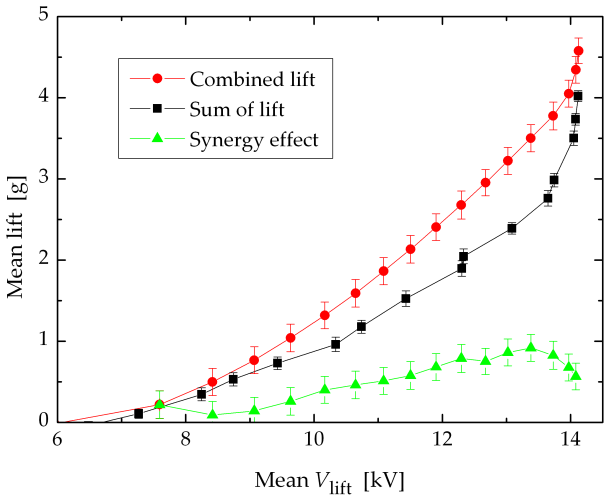
Synergy Effects of Concentric Lifters 7.4

Another theory that we wanted to test, was possible synergy effects of having a smaller lifter inside a bigger one. All data was thus measured while they were attached to each other. This should decrease the unknown effects of being attached to each other.

As seen in Fig. 11b we actually measured a synergy effect increasing approximately linearly with V_{lift} from $0 - 0.9\,\text{g}$. The error bars on the voltage has been left out to increase legibility. Our theory for this effect is, that the presence of two airflows creates a more laminar airflow with less turbulence. This allows the air to pass more freely and thus a slightly smaller amount of momentum is lost. We cannot say whether or not the mere presence of another lifter increased the lift, since they were together for all three measurements. Their individual thrust was unfortunately not measured at the same time, and as will be shown in Sec. 8.1 we cannot compare measurements separated by too much time. Having two airflows gave an increased lift. The absence of error bars on Fig. 11b is due to the fact that we had very few measurements, and thus the error bars are huge and have been removed for better legibility. Of course we would be required to make more measurements in order to get the shown effect greater than the uncertainty, but we still hypothesize that an effect was present. No conclusions can be made.



(a) *In the above figure the four different sized lifters are compared with respect to their relative lift pr. cm. No clear tendency as to which size lifter is optimal.*



(b) *The 20 cm lifter was connected to the 26 cm lifter enabling us to determine any synergy effects. The combined lift is the thrust of the two lifters together, while the sum of lift is the sum of the two independent lifters' thrust. The reason for several thrust measurements being at 14 kV is because the V_{lift} measurements were out of the input range of the data card as described in Sec. 6.4 therefore the value of V_{lift} is only credible in the range of $V_{\text{lift}} = 8 - 13$ kV.*

Figure 11: *The size of lifters compared to their lift is shown in (a). No clear tendency is seen. On the other hand in (b) we see a indication that attaching two lifters together would increase the total lift.*

Data Analysis 8

Reproducibility 8.1

In order to find out if the measurements could be reproduced day to day we took a series of measurements. We expected the only varying parameter to be the relative humidity. We measured it to range from 40% to 60% which at 23 °C corresponds to a partial pressure of water vapor of 1.12 kPa and 1.68 kPa respectively. According to Peek this would give us a lowering of V_c with approximately 3% and 4% respectively, [7]. If no other parameters affected the setup, we should find that the difference between the lowest and highest potential would differ no more han 0.97% of the lowest V_c. This is calculated assuming the lowest V_c is 3% higher than the V_c when no water vapor is present. Setting V_0 as the threshold potential with no water vapour we show

$$V_{\text{highest}} = V_{\text{lowest}} + V_{\text{lowest}} \frac{V_0 \cdot 104\% - V_0 \cdot 103\%}{V_0 \cdot 103\%} \tag{8.1}$$

$$= V_{\text{lowest}} + V_{\text{lowest}} \frac{1\%}{103\%} \tag{8.2}$$

$$= V_{\text{lowest}} + V_{\text{lowest}} 0.97\%. \tag{8.3}$$

As seen on Fig. 12 the lowest V_c was 7 kV and the highest V_c was 9 kV, but 0.97% of 7 kV is much less than the 2 kV difference measured. Thus the deviation in V_c due to the humidity of the air is much less than other effects. Even though the humidity did appear to have an effect: On humid days the lifter was more likely to create a spark. We can explain this, because the water molecules in the air are easier to ionize, and thus an electric current was more likely to be created. This enabled the spark to initiate at lower voltages.

The deviations could also be related to the aluminum foil, whose surface could change when we touched the lifter. That meant that kinks in the foil could appear, or disappear, resulting in change of the corona threshold voltage. This on the other hand could explain the deviation, because of the large difference in radii of a small kink compared to the actual radii of the foil.

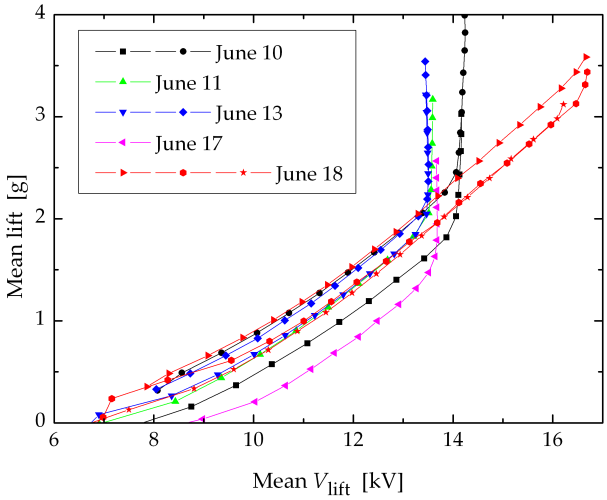


Figure 12: *Plot of several thrust-V_{lift} measurements of the same size lifter, 26 cm, in the same setup on different dates. It is seen that V_c varies between 7 and 9 kV. These deviations do not lie within the range explained by variations in the relative humidity. Note that several measurements have many lift readings for 14 kV, this is due to the out of range error described in Sec. 6.4 and is not a physical phenomenon. It is seen that we had solved the problem for the measurements on June 17th and 18th, all of the other measurements should have had same qualitative characteristics.*

Oscillation of Corona Wire 8.2

A buzzing sound was audible when V_{lift} exceeded approximately 13 kV. It is not to be confused with the hissing sound of the actual corona discharge. The corona wire could be seen oscillating with an amplitude of several mm. At the same time we saw an oscillation in the current output from our amplifier with an amplitude of 1.5 – 2 A, depending on A_{in}. The fact that these oscillations do not seem to have influenced our measurements, might be because the oscillations were so fast that the scale could not keep up with the change in the lift. Therefore the oscillations do not appear in the weight measurements. We simply did not have the ability to measure it. Even if the lifter was flying the lifter moved as if no oscillation was present. The oscillation did, however, stress our corona wires and caused them to snap prematurely.

The oscillation had an frequency of 100 Hz as seen in Fig. 13. This is calculated on the basis of 10 periods passing in 0.1 s. It's also seen on the figure that there exhibits a capacitor like decrease in the curve, but not a capacitor like increase. This is hard to explain as the actual components in the transformer are not known. We investigated the oscillation further and found that the frequency was independent of the lifter size or increase in V_{lift} . The fact that the frequency was independent of further increase in V_{lift} after onset, and that it was exactly twice that of the net frequency, leads us to think it is an oscillation induced by one of our components in the electric circuit.

V-I Properties 8.3

Unfortunately, we were only able to measure the current through the lifter on the high voltage side "manually," by putting a (cheap) ampere meter in a serial connection between the transformer and the lifter. This meant that we only made one series of measurements in oil and air. We knew the voltage both from the known factor between the monitor voltage V_{mon} and V_{lift}, and from measuring it directly on the high voltage side with an electro-meter. An electro-meter is a voltmeter based on coulomb force attraction. As shown by two graphs in Fig. 14 we see that the voltage almost increases as a linear function of the current, in the range from 0 to 13 kV and 200 µA in air, and from 0 to 11 kV and 3.1 µA in oil, and another linearity in the ranges above for both oil and air. Since the resistance of the medium is the derivative of the VI graph we get that the resistance for oil drops in a very quickly from 1.6 GΩ/cm to around 200 MΩ/cm (as recalled, we measured the resistance over a distance of 2 cm) and the resistance stayed approximately constant at 200 MΩ for higher voltages, as indicated by the linear fit. This is interesting because it shows that the lifter acts approximately as an ohmic resistance, both in oil and air, in average fields over 6.5 kV/cm in air and 5.5 kV/cm in oil. The steep decay in resistance is possibly caused by the *Avalanche Effect* which takes place from corona onset voltages and up, and corresponds to the change in resistance. The sudden change to the approximately constant resistance could be explained, by the limited volume in which ionization takes place

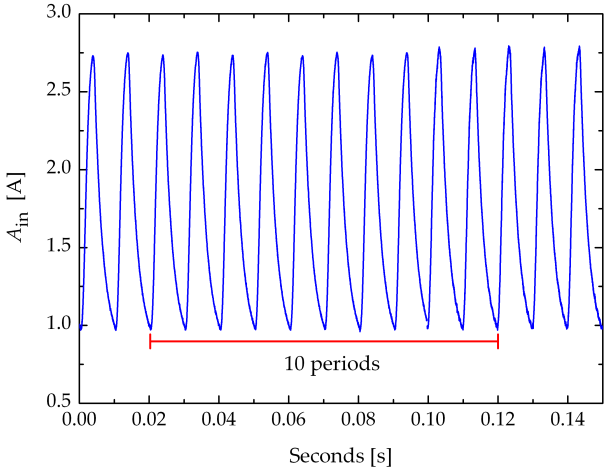


Figure 13: *The lifter showed a frequency of about* 100 Hz. *There was a clear sound which easily could be heard.*

around the corona wire, and the limited diffusion of air. This sets an upper limit on the number of moving ions, i.e. current.

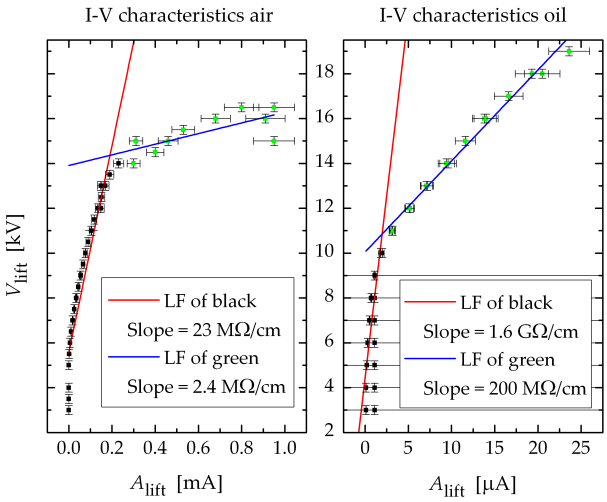


Figure 14: *In the above graphs it is shown how the approximative ohmic resistance is calculated. Notice that the resistance in oil is much higher than air. And that both media exhibit the same steep drop in resistance as the corona discharge initiates.*

Townsend Avalanche Analysis 8.4

For further analysis of the Townsend Avalanche theory we consider a N_2^+ ion in a uniform electric field, $\mathbf{E} = 66\,\mathrm{MV/m}$, this is approximately the field at the surface of the corona wire as derived from Fig. 4a, in Sec. 4. We thus assume the field to be constant in the vicinity of the ion. The characteristic length between molecules in atmospheric air can roughly be approximated with the volumetric density and molecular weight of O_2 $(2 \cdot 16\,\mathrm{g/mol})$ and N_2 $(2 \cdot 14\,\mathrm{g/mol})$:

$$\overset{\text{Nitrogen}}{0.79 \cdot 28\,\mathrm{g/mol}} + \overset{\text{Oxygen}}{0.21 \cdot 32\,\mathrm{g/mol}} = 26.9\,\mathrm{g/mol} \tag{8.4}$$

with a density of approximately $1.3\,\mathrm{kg/m^3}$ we end up with

$$\frac{1300\,\mathrm{g/m^3}}{26.9\,\mathrm{g/mol}} = 48\,\mathrm{mol/m^3} = 2.9 \times 10^{25}\,\mathrm{molecules/m^3} = 0.03\,\mathrm{molecules/nm^3} \tag{8.5}$$

We end up with a mean distance of 3 nm between molecules. We use this to calculate the mean free path, d_{air}, an ion can travel between impacts, with the formula, [11, p. 700-701]

$$d_{\text{air}} = v_{\text{air}} t_{\text{mean}} = \frac{V}{4\pi\sqrt{2}r^2 N} \tag{8.6}$$

where v_{air} is the molecules speed and t_{mean} is the mean time between collisions, N/V is molecules per unit volume, and r is the radius of the molecule. We have approximated the radius of "air molecules" to equal that of nitrogen, being 65 pm. We assume that the N_2 ions will align themselves parallel to the direction of travel. This approximation was needed in order to simplify the calculations. With these values we get the mean free path to $d_{\text{air}} = 4.6 \times 10^{-7}$ m. This justifies the assumption that the field is approximately constant over the mean free path. We then calculate the acceleration from Newton's second law and the fact that the force exerted on the ion equals the charge times the electric field. The force on the molecule is, see Eq. (2.2)

$$\mathbf{F} = q\mathbf{E} = 1.602 \times 10^{-19}\,\text{eV} \cdot 66\,\text{MV/m} = 1.06 \times 10^{-11}\,\text{N} \tag{8.7}$$

therefore the acceleration of the molecule is, according to Newton's second law

$$a_{\text{air}} = \mathbf{F}/m = \frac{1.06 \times 10^{-11}\,\text{N}}{28 \cdot 1.66 \times 10^{-27}\,\text{kg}} \approx 2.3 \times 10^{14}\,\text{m/s}^2 \tag{8.8}$$

we then solve the equations of movement, disregarding the initial velocity given by the drift velocity.

$$t_{\text{air}} = \sqrt{\frac{2d_{\text{air}}}{a_{\text{air}}}} = \sqrt{\frac{2 \cdot 4.6 \times 10^{-7}\,\text{m}}{2.3 \times 10^{14}\,\text{m/s}^2}} = 6.4 \times 10^{-11}\,\text{s} \tag{8.9}$$

we then multiply the acceleration by the time passed, and finally find the kinetic energy in electron volts

$$v_{\text{air}} = a_{\text{air}} \cdot t_{\text{air}} = 2.3 \times 10^{14}\,\text{m/s}^2 \cdot 6.4 \times 10^{-11}\,\text{s} = 14\,478\,\text{m/s} \tag{8.10}$$

$$E_{\text{kin}} = \frac{1}{2}mv^2 = \frac{1}{2}(14\,478\,\text{m/s})^2 \cdot 28 \cdot 1.66 \times 10^{-27}\,\text{kg} = 30.4\,\text{eV}. \tag{8.11}$$

We can now make a rough estimate of the number of ions one ion from the corona wire can give rise to. Assuming all impacts are elastic, and that all other molecules are at rest before impact and of same mass, we may assume no energy is lost in the particles in the collision. The kinetic energy will thus be split between the two molecules, and after impact they now both have an energy of 15 eV. This energy is still enough to ionize one more molecule each (the first ionization energy of N_2 is 14.5 eV and 13.6 eV for N_2 [12]). If we assume the electric field to go as the field of an infinity wire described by

$$\mathbf{E} = \frac{\lambda}{4\pi\epsilon r} \quad , \lambda \text{ is the line charge density, } \epsilon \text{ is the permittivity and } r \text{ the distance.} \tag{8.12}$$

Setting the field at the surface of the wire to 66 MV/m. Iterating through the field with the above calculations until the speed reaches a velocity that can't ionize Nitrogen, and counting the ions created as the loop runs. We get that one ionized molecule will give rise to 420 other ionized molecules, only including the ions that the initial ion have ionized. All of this happens within the distance of 0.19 mm from the corona wire. If also counting the ions created by the first generation there would indeed be an avalanche of ions. However we have left out a lot of factors such as ion recombination etc. so the actual number of ions created will be smaller than indicated. The code for this small loop is seen in Appendix C and runs in Maple 11 (a powerful scientific calculator).

We have a velocity of 1.4×10^4 m/s right before impact, which is several times greater than the mean velocity of molecules in air given by the diffusion constant, being $\approx 0.2\,\text{cm}^2/\text{s}$, at room temperature [11], and also significantly greater than any drift velocity possible in the air, so we were justified in disregarding it. The ions leave faster than they arrive at a certain point, and even though the whole mass of air around the corona wire eventually will be in motion, we can safely assume that the ions leave faster than new air molecules arrive. This puts an upper limit on the conductivity and thus a lower limit for the resistance. The same calculations will now be done for oil.

The density of canola oil is about $920\,\text{kg/m}^3$ and the makeup is 55% Oleic acid, 25% Linoleic acid, 10% Alpha-linolenic acid and 10% various saturated fatty acids, we have chosen stearic acid as an example. This gives us a molecular weight of:

$$\overset{\text{Oleic}}{0.55 \cdot 282.41\,\text{g/mol}} + \overset{\text{Linoleic}}{0.25 \cdot 280.45\,\text{g/mol}} + \overset{\text{Alpha–Linoleic}}{0.1 \cdot 278.43\,\text{g/mol}} + \overset{\text{Various}}{0.04 \cdot 284\,\text{g/mol}} = 264\,\text{g/mol} \tag{8.13}$$

Since oil is a liquid we can assume that the mean free path is approximately the same as the mean distance between molecules, [11]. By calculating the mean distance between molecules in the manner we did it for air, we get an intermolecular distance of $d_{oil} = 0.78 \times 10^{-9}$ m. Again, in order to simplify matters we assume the long C-H molecules are parallel to the direction of movement. We approximate the radius in the direction of movement to equal the radius of Carbon, 70 pm plus the diameter of Hydrogen, 50 pm, plus the length of a C-H bond, 144 pm thus giving a total radius of 264 pm. Employing the same calculations used for air, we find the density to be 2 molecules/nm^3. Inserting all this in Eq. (8.6) gives a kinetic energy of 0.05 eV and a velocity of 195 m/s. The low energy would indicate that the ionization in oil does not give rise to a Townsend Avalanche, when using the rather significant simplifications mentioned in the beginning of the calculations.

We also made a very crude attempt at determining the dielectric breakdown field of the oil. We simply lowered two wires with a 20 kV difference between them into the oil and moved them closer until a spark jumped. This happened at a distance of less than 0.5 mm and so the breakdown field must be greater than 40 GV/m.

Thrust Characteristics in Oil vs. Air 8.5

Since any dielectric medium suffers from dielectric breakdown at some point, and because some of these are much better insulators than the humid air in our lab, we wanted to see how a lifter would work in normal canola oil. In air the lifter lifted about 2 – 3 g. But in oil we saw up to 14 g of thrust as seen on Fig. 15. This was quite amazing.

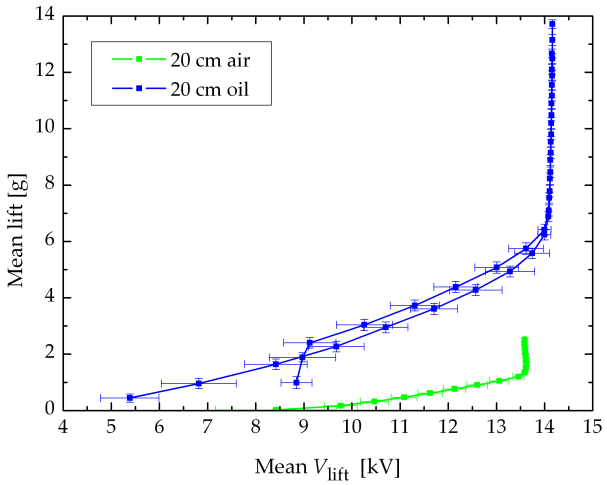


Figure 15: *It's here clearly shown that a lifter in oil produces much more lift than in air. Almost 6 times as much lift in oil as in air. Further more the problem of sparks in oil has never been encountered.*

If we look at the thrust per watt the difference becomes even greater. If we take as an example, the 26 cm lifter, which is slightly larger than the lifter in oil, at its' maximum efficiency of 0.94 g/W at 12.5 kV, 3.2 g and 10.9 W at 12 V. The lifter in oil had an efficiency of 20.7 g/W which increases to a maximum of 41.6 g/W at 17 V and 0.36 W. As shown the maximum efficiency is approximately 40 times greater in oil than in air. It is also noteworthy that a lifter in air has its maximum efficiency at 12 V V_{in}, approximately half way to the maximum of V_{in}, whereas a lifter in oil has an efficiency that increases with greater V_{in} and thus might have obtained an even greater maximum lift had our transformer allowed for greater V_{in}.

These measurements were made after the oil had been sitting in a rather rusty container for 3 months and had turned from yellow to a rusty orange. This presumably lowered the maximum thrust of the lifter, as a greater amount of charge was able to leave the corona wire through charging the iron present in the oil, using it as charge carrier. Even so, we still had a thrust more than four times as high as that of an identical lifter in air. The differences observed were far too great to be explained with inaccuracies in our measurements and differences in the lifters, aluminum foil, etc. It is also very interesting that both the resistance of oil and efficiency of the lifter in oil is approximately 40 times greater than the resistance and efficiency in air, for voltages over 13.5 kV between the electrodes, which is at least an interesting coincidence.

We have some possible explanations for why the lifter works so much better in oil than in air. As stated, the resistance is 40 times higher and the breakdown voltage is also significantly higher, which would allow for higher operational voltage differences. However, since we were not able test at higher voltages, it does not explain the

difference we measured.

We suspect that the main contributing factor is that oil has a very short mean free path. This does not allow the ions to transfer very much momentum back to the lifter on impact with the anode foil, whereas the ions in air are able to transfer about 10 times more momentum back to the lifter on impact. Somewhat like the difference between a running and a spinning wheel. This indicates a relationship between the efficiency of the lifter and the resistance of oil, but the data supports no conclusion. Further investigation would be necessary. The only conclusion is that the lifter is far more efficient in oil than air.

Prospects and Conclusion 9

Throughout this process we learned a lot about how good experimentalists are investigating, improving and changing experiments. About encountering problems and finding the best solution for the experiment. Sometimes an experimentalist must start the process all over in order to get credible results, as we did. We have also gained experience in how to analyze the gathered data.

We found an optimal air gap, an optimal aluminum height. But different sizes of the lifter didn't seem to have any pros or cons. The V-I properties of the lifter was demonstrated and it was showed that the resistance in oil was about 40 times as great as in air, with voltages above the corona onset. Giving a much greater lift in oil.

In terms of the practical applications of the principles discussed, there are many possibilities, but also, as we have seen, limitations. Although the electrostatic lifter is fairly efficient in oil, it is not in air. We suspect it would also be efficient in other dense dielectric media in terms of thrust pr. gram. The 44 g/W lift in oil translates to approximately 30 kg/HP which is not that far from the efficiency of an airplane wing.

The most likely application that comes to mind based on these results, would be to use the EHD flow in oil to make a pump with no moving parts. As we have seen EHD is much more effective in terms of thrust pr. watt in oil than air, even though it is not as efficient as a conventional pump. It could be advantageous to use EHD flow "pumps" in hard to reach places, since it has no moving parts that can break.

The lifter in air is a far cry from being useful as a means of propulsion or levitational use. The efficiency is simply to little and the size too great — our largest lifter was 0.5 m^2 and could only lift 10 g which was half its own weight. The largest lifters we have heard of in other research are several m^2 and can still only lift a few hundred grams. One could argue that the ion thrusters used in satellites are a sort of electrostatic "lifter" engine. There the high velocity ions and simplicity of the design are great advantages and the lack of thrust is less of a concern. However, one can hardly argue that such engines make use of EHD flow since they work in a vacuum.

Another application that created some interest was to use the lifter for ventilation on small scales. The viscosity of air creates problems for fans when channels in heat sinks become too small [13]. The success of these cooling devices will probably depend on how well the high voltage can be shielded from sensitive circuitry, for example in CPU cooling. The area where EHD pumps have the greatest success is as pumping devices on micrometer scale and smaller. They have been used for some time as pumps on Lab-on-a-chip devices [14] when capillary forces do not suffice. We are more likely to see EHD pumps employed in the micro laboratories of tomorrow, rather than electrostatic levitational devices making cars fly.

References

[1] Myron Robinson. A History of the Electric Wind. *American Journal of Physics*, 30(5):366–372, May 1962.

[2] Lin Zhao and Kazimierz Adamiak. Numerical Analysis of Forces in an Electrostatic Levitation Unit. *Journal of Electrostatics*, 63:729–734, March 2005.

[3] Lin Zhao and Kazimierz Adamiak. EHD Gas Flow in Electrostatic Levitation Unit. *Journal of Electrostatics*, 64:639–645, November 2005.

[4] Lin Zhao and Kazimierz Adamiak. EHD Flow in Air Produced by Electric Corona Discharge in Pin-Plate Configuration. *Journal of Electrostatics*, 63:337–350, July 2004.

[5] Kazimierz Adamiak, Vladimir M. Atrazhev, and Pierre Atten. Corona Discharge in The Hyperbolic Point-Plane Configuration: Direct Ionization Criterion Versus Approximate Formulations. *IEEE Transactions on dielectrics and electrical insulation*, 12(5):1025–1034, October 2005.

[6] Kazimierz Adamiak and Pierre Atten. Simulation of Corona Discharge in Point-Plane Configuration. *Journal of Electrostatics*, 61:85–98, January 2004.

[7] F. W. Peek. *Dielectric Phenomena in High Voltage Engineering*. McGraw-Hill, 1 edition, 1929.

[8] David J. Griffiths. *Introduction to Electrodynamics*. Prentice Hall, 3 edition, 1999.

[9] John R. Tayler. *An Introduction to Error: The Study of Uncertainties in Physical Measurements*. University Science Books, 1982.

[10] K. R. Allen and K. Philips. Effect of Humidity on the Spark Breakdown Voltage. *Nature*, 183(4655):174–175, January 1959.

[11] Hugh D. Young and Roger A. Freedman. *University Physics with Modern Physics*. Pearson, 11 edition, 2004.

[12] Raymond Chang. *General Chemistry*. McGraw-Hill, 4 edition, 2006.

[13] F. Yang, N.E. Jewell-Larsen, D.L. Brown, K. Pendergrass, D.A. Parker, I.A. Krichtafovitch, and A.V. Mamishev. Corona Driven Air Propulsion for Cooling of Electronics. *International Symposium on High Voltage Engineering, Netherlands*, XIII:1–4, 2003.

[14] R. Edwin Oosterbroek and Albert Berg. *Lab-on-a-chip: Miniaturized Systems for (bio)chemical Analysis*. Elsevier, 1 edition, 2003.

Appendix: Scientific instruments A

We here supply a list over used instrumentation on the acquiring of data.

- LabView 8.0
- Measurement card, input/output, NI-SC-2345
  - SCC-AI03, A_{in}
  - 2×SCC-FT01, V_{lift}, V_{PC}
  - SCC-LP02, V_{mon}
- Iso-Tech: PS IPS-2010, V_{lift}
- Frederiksen, Amplifier 2500.50, V_{amp}
- Kern 440-35A, W

Appendix: Diagram of transformer B

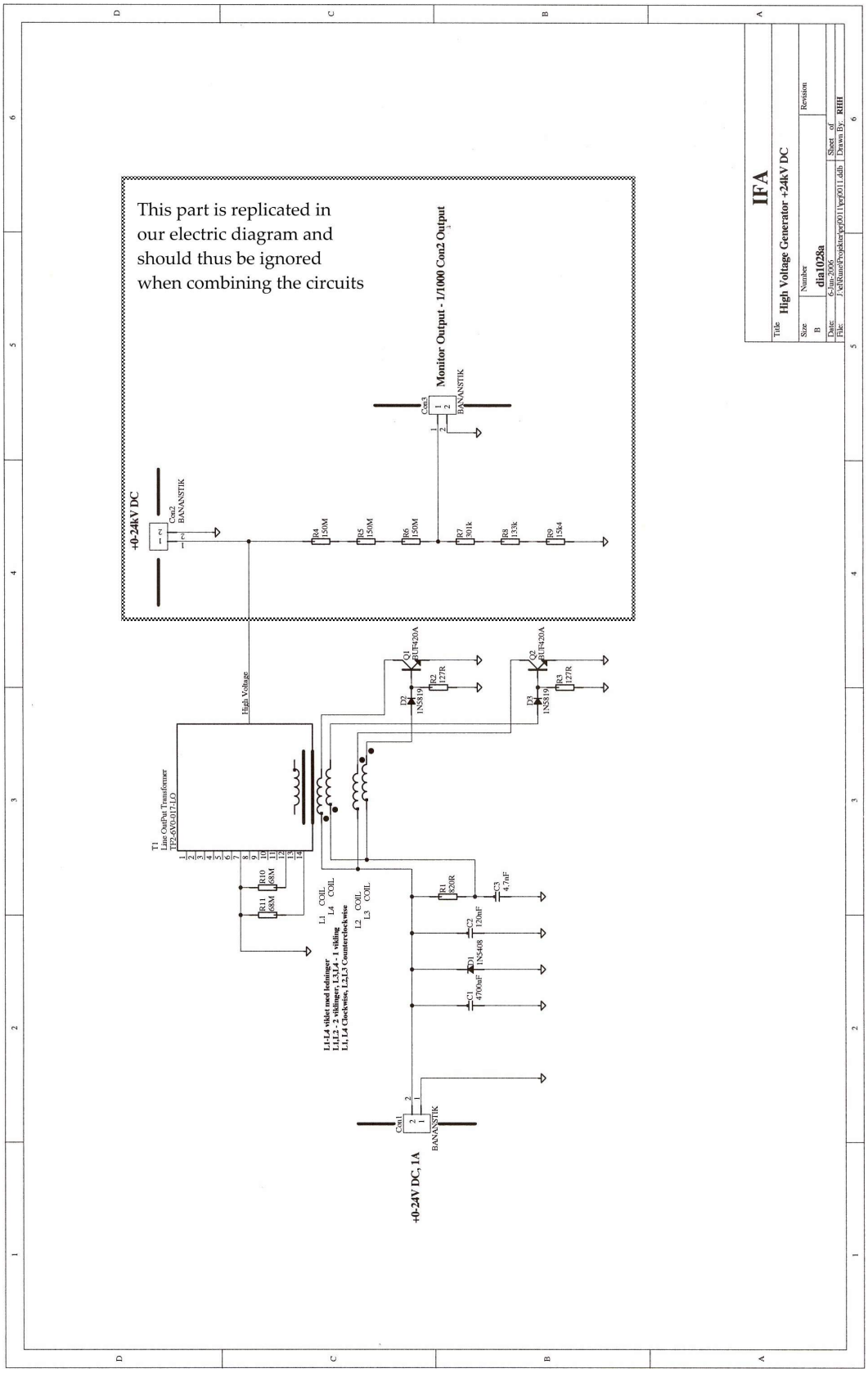


Figure 16: *Transformer diagram of our high voltage supply. Notice that the exact understanding of the transformer-part is unknown. This is due to the CRT-monitor unit inserted in the circuit.*

Appendix: Townsend ionized molecules program C

Here the program of the Townsend ionization is shown. The values are all repeated in Sec. 8.4.

```
restart;
eVLimit := 14:
r := 65e-12:
q := 1.602e-19:
m := 28*1.66e-27:
d := evalf(3.4595e-26/(4*Pi*sqrt(2)*r^2)):
b := solve(b/(0.025e-3)=66e6,b):
Efield := x -> b/(0.025e-3+x):

vnow := 0:
molecules := 1:
for i from 0.00 by d while i < 0.02 do
  a := Efield(i)*q/m;
  t0 := abs(solve(1/2*a*t^2=d,t)[1]);
  v := vnow + a*t0;
  eV := 1/2*m*v^2/q;
  if (eV >= eVLimit) then
    molecules := molecules + 1;
  else
    break;
  end if;
  vnow := v/2;
end do:
```

Appendix: Source code D

None of the below codes are supplied with comments. This is primarily due to the simplicity in the coding and for a better overview of the pages.

Main class D.1

Start D.1.1

Java-code 1: *Start.java*

```
1  import gui.InterFace;
2
3  public class Start {
4
5      public Start() {}
6
7      public static void main(String arg[]) {
8          javax.swing.SwingUtilities.invokeLater(new Runnable() {
9              public void run() {
10                 InterFace frame = new InterFace();
11                 frame.setVisible(true);
12             }
13         });
14     }
15 }
```

Package: calc D.2

Poisson D.2.1

Java-code 2: *calc/Poisson.java*

```
1  package calc;
2
3  import gui.ProgressMonitorSetup;
4  import io.GetPoisson;
5  import io.WriteData;
6
7  import java.awt.Toolkit;
8
9  import javax.swing.JOptionPane;
10 import javax.swing.SwingWorker;
11
12 public class Poisson {
13     private ProgressMonitorSetup progressMonitor;
14     private int row = 1000;
15     private int col = 1000;
16     private int N = 10000;
17     private double[][] V;
18
19     private double Vmax = 20000D;
20     private double Vmin = 0D;
21
22     private int rkSt;
23     private int progressStart = 0;
24     private int colSt;
25
26     public Poisson() {}
```

```
28      public boolean importV(final String filNavn, final int start) {
29              final GetPoisson getV = new GetPoisson();
30              final boolean allow = getV.setFile(filNavn);
31              if (allow) {
32                      JOptionPane.showMessageDialog(null, "Importere nu filen:" + filNavn, "Importering", JOptionPane.
                                INFORMATION_MESSAGE);
33                      V = new double[row + 1][col + 1];
34                      V = getV.getPoissonData(row + 1, col + 1);
35                      rkSt = Math.round(row / 3);
36                      colSt = Math.round(col / 2);
37                      progressStart = start;
38              }
39              return allow;
40      }

42      public void setIterations(final int N) {
43              this.N = N;
44      }

46      public void setVmax(final double Vmax) {
47              this.Vmax = Vmax;
48      }

50      public void setVmin(final double Vmin) {
51              this.Vmin = Vmin;
52      }

54      public void setRow(final int row) {
55              this.row = row;
56      }

58      public void setColumn(final int col) {
59              this.col = col;
60      }

62      public int getIterations() {
63              return N;
64      }

66      public double getVmax() {
67              return Vmax;
68      }

70      public double getVmin() {
71              return Vmin;
72      }

74      public int getRow() {
75              return row;
76      }

78      public int getColumn() {
79              return col;
80      }

82      public void setSize(final int row, final int col) {
83              this.col = col;
84              this.row = row;
85      }

87      public void writeBoundary() {
88              V = new double[row + 1][col + 1];

90              // initial conditions
91              for (int i = 0; i <= row; i++) {
92                      for (int j = 0; j <= col; j++) {
93                              if (i < row / 2) {
94                                      V[i][j] = Vmax / 3;
95                              } else {
96                                      V[i][j] = Vmax / 6;
97                              }
98                      }
99              }
```

```
101             rkSt = Math.round(row / 3);
102             colSt = Math.round(col / 2);
103             writeBoundaryLoop();
104             V[rkSt + 1][colSt] = Vmax / 2;
105             V[rkSt - 1][colSt] = Vmax / 2;
106             V[rkSt][colSt + 1] = Vmax / 2;
107             V[rkSt][colSt - 1] = Vmax / 2;
108         }

110         public void writeBoundaryLoop() {
111             for (int i = 0; i <= row; i++) {
112                 V[i][0] = V[i][1]; // left
113                 V[i][col] = V[i][col - 1]; // right
114             }
115             for (int j = 0; j <= col; j++) {
116                 V[0][j] = V[1][j]; // top
117                 V[row][j] = V[row - 1][j]; // bottom
118             }

120             V[rkSt][colSt] = Vmax;

122             V[2 * rkSt - 3][colSt] = Vmin;
123             V[2 * rkSt - 2][colSt + 1] = Vmin;
124             V[2 * rkSt - 2][colSt] = Vmin;
125             V[2 * rkSt - 2][colSt - 1] = Vmin;

127             V[2 * rkSt - 1][colSt - 2] = Vmin;
128             V[2 * rkSt - 1][colSt - 1] = Vmin;
129             V[2 * rkSt - 1][colSt] = Vmin;
130             V[2 * rkSt - 1][colSt + 1] = Vmin;
131             V[2 * rkSt - 1][colSt + 2] = Vmin;
132             for (int j = 2 * rkSt; j < row; j++)
133                 for (int i = -25; i < 26; i++)
134                     V[j][colSt + i] = Vmin;
135         }

137         public void runSimulation() {
138             progressMonitor = new ProgressMonitorSetup();
139             progressMonitor.setTitle("Løser Poisson's ligning");
140             progressMonitor.setMaximum(N);
141             progressMonitor.initProgressMonitor();
142             final RunPoisson task = new RunPoisson();
143             task.execute();
144         }

146         class RunPoisson extends SwingWorker<Void,Void> {
147             @SuppressWarnings("finally")
148             @Override
149             public Void doInBackground() {
150                 int progress = 0;
151                 progressMonitor.setProgress(progress);
152                 String fileName = "";
153                 final WriteData save = new WriteData();
154                 writeBoundaryLoop();
155                 try {
156                     Thread.sleep(500);
157                     while (progress < N && !progressMonitor.isDone()) {
158                         if (progress % 100 == 0)
159                             progressMonitor.setProgress(progress);
160                         if (progress % 10000 == 0) {
161                             fileName = "I[" + (progress + progressStart) + "]rk[" + row + "]col[" + col +
                                    "]N[" + N + "].txt";
162                             save.setFile(fileName);
163                             save.savePoissonData(V);
164                         }
165                         for (int i = 1; i < row; i++) {
166                             for (int j = 1; j < col; j++) {
167                                 if (j == colSt && i == rkSt) {
168                                     V[i][j] = Vmax;
169                                 } else {
170                                     V[i][j] = 0.16666666666 * (V[i - 1][j] + V[i + 1][j] + V[i][j -
                                            1] + V[i][j + 1]) + 0.0833333333
```

```
171                                                                 * (V[i - 1][j + 1] + V[i - 1][j - 1] + V[i + 1][j +
                                                                        1] + V[i + 1][j - 1]);
172                                         }
173                                     }
174                                 }
175                                 writeBoundaryLoop();
176                                 progress++;
177                             }
178                     } catch (final InterruptedException e) {
179                             JOptionPane.showMessageDialog(null, "Simuleringen blev afbrudt", "Poisson", JOptionPane.
                                        INFORMATION_MESSAGE);
180                     } catch (final Exception e) {
181                             e.printStackTrace();
182                     } finally {
183                             fileName = "I[" + (progress + progressStart) + "]rk[" + row + "]col[" + col + "]N[" + N + "
                                        ].txt";
184                             save.setFile(fileName);
185                             save.savePoissonData(V);
186                             progressMonitor.setProgress(N);
187                             return null;
188                     }
189             }

191             @Override
192             public void done() {
193                     Toolkit.getDefaultToolkit().beep();
194                     progressMonitor.close();
195                     JOptionPane.showMessageDialog(null, "Simuleringen er færdig", "Poisson", JOptionPane.
                                INFORMATION_MESSAGE);
196             }
197     }
198 }
```

Package: gui D.3

Interface D.3.1

Java-code 3: *gui/Interface.java*

```
1  package gui;

3  import java.awt.Font;
4  import java.awt.GridBagConstraints;
5  import java.awt.GridBagLayout;
6  import java.awt.Insets;
7  import java.awt.event.ActionEvent;
8  import java.awt.event.ActionListener;

10 import javax.swing.JButton;
11 import javax.swing.JFrame;
12 import javax.swing.JLabel;
13 import javax.swing.JPanel;
14 import javax.swing.JTextField;

16 public class InterFace extends JFrame {
17       private final static long serialVersionUID = 24362461L;

19       JTextField JThstep = new JTextField(15);
20       JLabel headLine = new JLabel("Du er igang med at løse Poissonligningen");
21       private PoissonGUI pGUI = new PoissonGUI();

23       JButton run = new JButton("Kør");
24       JButton finish = new JButton("Afslut");
25       Font Fover = new Font("SanSerif", Font.BOLD, 11);
26       Font Funder = new Font("Verdana", Font.PLAIN, 9);

28       public InterFace() {
```

```
29              // Startsetup
30              super("Initialisering af differentialløsning...");
31              setSize(650, 550);
32              setDefaultCloseOperation(EXIT_ON_CLOSE);

34              GridBagConstraints c = new GridBagConstraints();
35              c.insets = new Insets(5, 5, 5, 5);

37              JPanel JPMASTER = new JPanel();
38              JPMASTER.setLayout(new GridBagLayout());
39              GridCont(c, 0, 0, 2, 1, 0, 0);
40              JPMASTER.add(headLine, c);
41              GridCont(c, 0, 2, 2, 1, 0, 0);
42              JPMASTER.add(pGUI, c);

44              // Buttons and their actions.
45              finish.addActionListener(new ActionListener() {
46                      public void actionPerformed(ActionEvent arg) {
47                              System.exit(0);
48                      }
49              });
50              GridCont(c, 0, 5, 2, 1, 0, 0);
51              JPMASTER.add(finish, c);
52              JPMASTER.setSize(JPMASTER.getPreferredSize());
53              setContentPane(JPMASTER);
54              // Centering of all
55              setLocationRelativeTo(null);
56      }

58      private void GridCont(GridBagConstraints GBC, int gx, int gy, int gw, int gh, double wx, double wy) {
59              GBC.gridx = gx;
60              GBC.gridy = gy;
61              GBC.gridwidth = gw;
62              GBC.gridheight = gh;
63              GBC.weightx = wx;
64              GBC.weighty = wy;
65      }
66  }
```

ProgressMonitorSetup D.3.2

Java-code 4: *gui/ProgressMonitorSetup.java*

```
1   package gui;

3   import java.awt.Toolkit;

5   import javax.swing.ProgressMonitor;

7   public class ProgressMonitorSetup {
8       private ProgressMonitor progressMonitor;
9       private String title;
10      private int min;
11      private int progress;
12      private int max;
13      private String note;
14      private String complete;
15      private String cancel;
16      private String completeTotal;

18      public ProgressMonitorSetup() {
19              title = "Kørsel af langt job";
20              min = 0;
21              max = 100;
22              complete = "Fremgang: ";
23              cancel = "Job afbrudt.";
24              completeTotal = "Job færdigt.";
25              note = "";
26      }
```

```
28      public ProgressMonitorSetup(String title, int min, int max, String complete, String completeTotal, String cancel
                ) {
29            this.title = title;
30            this.min = min;
31            this.progress = min;
32            this.max = max;
33            this.complete = complete;
34            this.completeTotal = completeTotal;
35            this.cancel = cancel;
36            note = "";
37      }

39      public void setTitle(String title) {
40            this.title = title;
41      }

43      public void setMinimum(int min) {
44            this.min = min;
45      }

47      public void setMaximum(int max) {
48            this.max = max;
49      }

51      public void setRunningComplete(String complete) {
52            this.complete = complete;
53      }

55      public void setComplete(String completeTotal) {
56            this.completeTotal = completeTotal;
57      }

59      public void setCancel(String cancel) {
60            this.cancel = cancel;
61      }

63      public String getCancel() {
64            return cancel;
65      }

67      public boolean isDone() {
68            if (progress >= max) {
69                  progressMonitor.close();
70                  return true;
71            }
72            return false;
73      }

75      public void close() {
76            progressMonitor.close();
77      }

79      public void initProgressMonitor() {
80            progressMonitor = new ProgressMonitor(null, title, note, min, max);
81            progressMonitor.setMillisToDecideToPopup(100);
82            progressMonitor.setMillisToPopup(100);
83            progressMonitor.setProgress(min);
84      }

86      public String setProgressMessage(int i) {
87            progress = i;
88            progressMonitor.setProgress(progress);
89            note = complete + progress + " af " + max;
90            progressMonitor.setNote(note);
91            if (progressMonitor.isCanceled() || isDone()) {
92                  Toolkit.getDefaultToolkit().beep();
93                  if (progressMonitor.isCanceled()) {
94                        progress = max++;
95                        return cancel;
96                  }
97                  return completeTotal;
98            }
99            return note;
```

```
100         }

102         public void setProgress(int i) {
103                 progress = i;
104                 progressMonitor.setProgress(progress);
105                 note = complete + progress + " af " + max;
106                 progressMonitor.setNote(note);
107                 if (progressMonitor.isCanceled() || isDone()) {
108                         Toolkit.getDefaultToolkit().beep();
109                         if (progressMonitor.isCanceled()) {
110                                 progress = max++;
111                         } else {}
112                 }
113         }
114 }
```

PoissonGUI D.3.3

Java-code 5: *gui/PoissonGUI.java*

```
1   package gui;

3   import java.awt.Font;
4   import java.awt.GridBagConstraints;
5   import java.awt.GridBagLayout;
6   import java.awt.Insets;
7   import java.awt.event.ActionEvent;
8   import java.awt.event.ActionListener;

10  import javax.swing.JButton;
11  import javax.swing.JCheckBox;
12  import javax.swing.JLabel;
13  import javax.swing.JOptionPane;
14  import javax.swing.JPanel;
15  import javax.swing.JTextField;
16  import javax.swing.SwingConstants;
17  import javax.swing.border.TitledBorder;

19  import calc.Poisson;

21  public class PoissonGUI extends JPanel {
22          private final static long serialVersionUID = 24362461L;
23          private TitledBorder titled;
24          private JButton Jrun = new JButton("Kør");
25          private JButton Jimport = new JButton("Import");
26          private JCheckBox JUseImport = new JCheckBox("Brug importerede data");
27          private JTextField JTrow = new JTextField(10);
28          private JTextField JTcol = new JTextField(10);
29          private JTextField JTN = new JTextField(10);
30          private JTextField JTVmax = new JTextField(10);
31          private JTextField JTVmin = new JTextField(10);
32          private Poisson poi = new Poisson();
33          Font Fover = new Font("SanSerif", Font.BOLD, 13);
34          Font Funder = new Font("Verdana", Font.PLAIN, 10);

36          private String[] lblNames = new String[] { "Rækker: ", "Kolonner: ", "Antal iterationer: ", "Potentialet på
                    grænsen: ", "Potentialet på jorden: " };

38          public PoissonGUI() {
39                  this.setLayout(new GridBagLayout());
40                  this.setSize(250, 200);
41                  titled = new TitledBorder("Poisson:");
42                  titled.setTitleFont(Fover);
43                  this.setBorder(titled);
44                  GridBagConstraints c = new GridBagConstraints();

46                  c.insets = new Insets(0, 0, 0, 0);

48                  ActionListener newValue = new ActionListener() {
49                          public void actionPerformed(ActionEvent evt) {
```

```
50                          Object source = evt.getSource();
51                          boolean doubleInt = false;
52                          String tekst = "";
53                          double valueD = 0.0D;
54                          int valueI = 0;

56                          if (source == JTrow) {
57                                 tekst = JTrow.getText();
58                          } else if (source == JTcol) {
59                                 tekst = JTcol.getText();
60                          } else if (source == JTN) {
61                                 tekst = JTN.getText();
62                          } else if (source == JTVmax) {
63                                 doubleInt = true;
64                                 tekst = JTVmax.getText();
65                          } else if (source == JTVmin) {
66                                 doubleInt = true;
67                                 tekst = JTVmin.getText();
68                          }
69                          try {
70                                 if (doubleInt) {
71                                        valueD = Double.parseDouble(tekst);
72                                 } else {
73                                        valueI = Integer.parseInt(tekst);
74                                 }
75                                 if (source == JTrow) {
76                                        poi.setRow(valueI);
77                                 } else if (source == JTcol) {
78                                        poi.setColumn(valueI);
79                                 } else if (source == JTN) {
80                                        poi.setIterations(valueI);
81                                 } else if (source == JTVmax) {
82                                        poi.setVmax(valueD);
83                                 } else if (source == JTVmin) {
84                                        poi.setVmin(valueD);
85                                 }
86                          } catch (NumberFormatException e) {
87                                 JOptionPane.showMessageDialog(null, "Initialiseringen af værdien er mislykkedes, prøv
                                              igen.", "Init", JOptionPane.INFORMATION_MESSAGE);
88                                 JTrow.setText("" + poi.getRow());
89                                 JTcol.setText("" + poi.getColumn());
90                                 JTN.setText("" + poi.getIterations());
91                                 JTVmax.setText("" + poi.getVmax());
92                                 JTVmin.setText("" + poi.getVmin());
93                          }
94                   }
95            };
96            Jrun.addActionListener(new ActionListener() {
97                   public void actionPerformed(ActionEvent arg) {
98                          if (!JUseImport.isSelected()) {
99                                 poi.writeBoundary();
100                         }
101                         poi.runSimulation();
102                  }
103           });

105           GridCont(c, 0, 0, 1, 1, 0, 0);
106           this.add(Jrun, c);

108           Jimport.addActionListener(new ActionListener() {
109                  public void actionPerformed(ActionEvent arg) {
110                         String fileName = JOptionPane.showInputDialog("Indtast filnavn:");
111                         String value = JOptionPane.showInputDialog("Indtast startsimuleringsnummer:");
112                         if (fileName != null && value != null)
113                                JUseImport.setSelected(poi.importV(fileName, Integer.parseInt(value)));
114                  }
115           });
116           GridCont(c, 1, 0, 1, 1, 0, 0);
117           this.add(Jimport, c);
118           GridCont(c, 0, 1, 2, 1, 0, 0);
119           this.add(JUseImport, c);
120           for (int i = 0; i < lblNames.length; i++) {
121                  JLabel jl = new JLabel(lblNames[i]);
```

```
122                     GridCont(c, 0, i + 2, 1, 1, 0, 0);
123                     this.add(jl, c);
124             }
125             GridCont(c, 1, 2, 1, 1, 0, 0);
126             this.add(TextFieldSet(JTrow, newValue), c);
127             GridCont(c, 1, 3, 1, 1, 0, 0);
128             this.add(TextFieldSet(JTcol, newValue), c);
129             GridCont(c, 1, 4, 1, 1, 0, 0);
130             this.add(TextFieldSet(JTN, newValue), c);
131             GridCont(c, 1, 5, 1, 1, 0, 0);
132             this.add(TextFieldSet(JTVmax, newValue), c);
133             GridCont(c, 1, 6, 1, 1, 0, 0);
134             this.add(TextFieldSet(JTVmin, newValue), c);
135             initVars();
136     }

138     private void initVars() {
139             JTrow.setText("" + poi.getRow());
140             JTcol.setText("" + poi.getColumn());
141             JTN.setText("" + poi.getIterations());
142             JTVmax.setText("" + poi.getVmax());
143             JTVmin.setText("" + poi.getVmin());
144     }

146     private JTextField TextFieldSet(JTextField JT, ActionListener ac) {
147             JT.setText("00000000");
148             JT.setHorizontalAlignment(SwingConstants.CENTER);
149             JT.addActionListener(ac);
150             JT.setFont(Funder);
151             return JT;
152     }

154     private void GridCont(GridBagConstraints GBC, int gx, int gy, int gw, int gh, double wx, double wy) {
155             GBC.gridx = gx;
156             GBC.gridy = gy;
157             GBC.gridwidth = gw;
158             GBC.gridheight = gh;
159             GBC.weightx = wx;
160             GBC.weighty = wy;
161     }
162 }
```

Package: io D.4

GetPoisson D.4.1

Java-code 6: *io/GetPoisson.java*

```
1  package io;

3  import java.io.BufferedReader;
4  import java.io.File;
5  import java.io.FileInputStream;
6  import java.io.IOException;
7  import java.io.InputStreamReader;
8  import java.util.ArrayList;
9  import java.util.List;

11 import javax.swing.JOptionPane;

13 public class GetPoisson {

15     private File curFile = new File("NO");
16     private double[][][] V;

18     public GetPoisson() {}
```

```
20      public boolean setFile(String fileName) {
21              curFile = new File(fileName);
22              if (!curFile.exists()) {
23                      JOptionPane.showMessageDialog(null, "Filen er IKKE importeret, den eksistere ikke. Filnavn:\n" +
                                fileName, "Importering", JOptionPane.INFORMATION_MESSAGE);
24                      return false;
25              }
26              return true;
27      }

29      public double[][] getPoissonData(int row, int col) {
30              try {
31                      BufferedReader reader = new BufferedReader(new InputStreamReader(new FileInputStream(curFile), "
                                UTF8"));
32                      V = new double[row][col];
33                      String line = "";
34                      int lines = 0;
35                      List<Double> rowArray = new ArrayList<Double>();
36                      while ((line = reader.readLine()) != null && lines != row) {
37                              rowArray = convertToList(line.split("\t"));
38                              for (int i = 0; i < rowArray.size(); i++)
39                                      V[lines][i] = rowArray.get(i);
40                              lines++;
41                      }
42                      reader.close();
43                      JOptionPane.showMessageDialog(null, "Filen er importeret", "Importering", JOptionPane.
                                INFORMATION_MESSAGE);
44              } catch (IOException e) {
45                      e.printStackTrace();
46                      JOptionPane.showMessageDialog(null, "Filen er IKKE importeret, der er sket en fejl.", "Importering"
                                , JOptionPane.INFORMATION_MESSAGE);
47              }

49              return V;
50      }

52      protected static List<Double> convertToList(String[] anArray) {
53              if (anArray == null) {
54                      return null;
55              }
56              List<Double> v = new ArrayList<Double>(anArray.length);
57              for (int i = 0; i < anArray.length; i++) {
58                      v.add(Double.parseDouble(anArray[i]));
59              }
60              return v;
61      }
62  }
```

WriteData D.4.2

Java-code 7: *io/WriteData.java*

```
1   package io;

3   import gui.ProgressMonitorSetup;

5   import java.io.BufferedWriter;
6   import java.io.File;
7   import java.io.FileOutputStream;
8   import java.io.IOException;
9   import java.io.OutputStreamWriter;

11  import javax.swing.SwingWorker;

13  public class WriteData {
14      private ProgressMonitorSetup progressWrite;
15      private double[][] VV;
16      private File curFile = new File("NO");

18      public WriteData() {}
```

```
20      public void setFile(String fileName) {
21              curFile = new File(fileName);
22              if (curFile.exists()) {
23                      curFile.delete();
24              }
25      }

27      public final void savePoissonData(final double[][] V) {
28              this.VV = V;
29              progressWrite = new ProgressMonitorSetup();
30              progressWrite.setTitle("Gemmer fil indeholdende data");
31              progressWrite.setMaximum(VV.length);
32              progressWrite.initProgressMonitor();
33              SavePoisson task = new SavePoisson();
34              task.execute();
35      }

37      class SavePoisson extends SwingWorker<Void,Void> {
38              @Override
39              public Void doInBackground() {
40                      progressWrite.setProgress(0);
41                      try {
42                              final BufferedWriter out = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(
                                        curFile), "UTF8"));
43                              String text = "";
44                              for (int i = 0; i < VV.length; i++) {
45                                      progressWrite.setProgress(i);
46                                      for (int j = 0; j < VV[1].length; j++) {
47                                              text += "\t" + VV[i][j];
48                                              if (j == 0)
49                                                      text = "" + VV[i][j];
50                                      }
51                                      text += "\n";
52                                      out.write(text);
53                              }
54                              out.close();
55                      } catch (final IOException e) {
56                              e.printStackTrace();
57                      }
58                      progressWrite.setProgress(VV.length);
59                      return null;
60              }

62              @Override
63              public void done() {
64                      progressWrite.close();
65              }
66      }
67  }
```

In total **691** lines. All inclusive. Spread on **7** files.

Index

E

F

I

K

M

P

S

T