

RUBY

Trevor Smith, Alex Storrer*
Northeastern University
(Dated: June 7, 2021)

A green laser diode was used to excite a ruby crystal and examine its properties of transmission, absorption, and fluorescence. The transmission of photons with a wavelength of 700 nm was measured at 0.91 ± 0.05 , which is higher than the expected value of 0.851. The absorption length for this sample was measured at 4.9 ± 0.1 mm. The R-line emission was measured with a wavelength of 693.2 ± 0.2 nm, which is slightly lower than the accepted value of 694 nm. The fluorescence decay constant τ was measured at 3.52 ± 0.04 ms, which agrees exactly with the expected value of 3.5 ± 0.5 ms.

INTRODUCTION

In the first phase of this experiment, we will analyze the room lighting, background lighting, and calibrate the spectrometer. The calibration will be done by comparing the measured wavelength of a green laser diode to its expected wavelength of 532 nm. We will then block out the background light with a black cloth, and measure the transmission and absorption spectrum of our ruby crystal sample by measuring the intensity of a wide spectrum white light and dividing by the intensity of the white light after it passes through the ruby.

In the next phase we will investigate ruby fluorescence. Ruby fluorescence is key in its ability to function as a laser, where, after being excited by an incident light source, the electrons will quickly enter a meta-stable state from which they will more slowly exit, emitting a photon. In this lab we will be observing the wavelength of this emittance, which corresponds to the difference in energy levels, and the lifetime of this meta-stable state. The latter will be measured by exciting the ruby with the same laser, but powered by a function generator in a square wave pattern. The output of the ruby will then be collected with a photodiode and measured on an oscilloscope.

APPARATUS

The apparatus consisted of the following.

- Aluminum optical breadboard with 1/4-20 tapped holes
- Green laser diode
- Spectrometer, Ocean Optics USB4000FL, OceanView software, USB cable
- Lens, 200mm focal length
- Lens, 25mm focal length
- Mirror in adjustable x-y mount with rotational micrometer

- Photodiode (PD) detector
- Neutral-Density (ND) optical filter
- Long-Pass optical filter
- Optical Fibers, 50 and 600 micron core
- Black cloth
- White light illuminator
- Ruby Crystal, $\text{Al}_2\text{O}_3\text{:Cr}$, approximately 0.05% Cr
- Oscilloscope, Tektronix TDS1052B
- Signal Generator, GW Instek GFG-8216A
- BNC Cables and T adapter

ROOM LIGHT SPECTRUM AND CALIBRATION

Procedure

First, a baseline for background noise in measurements made pointed at the breadboard was established. The OceanView software was used to tune the Scans to Average setting, which controls the amount of time the spectrometer collects incoming photons, to maximize the usage of the input bits of that device. This step would be performed for each spectrometer measurement.

Next, the spectrometer was calibrated using the 532 nm green laser, because the basic function of the laser will remain unchanged from its factory settings due to the nature of how laser light is produced, while a spectrometer as a measurement device relies on mirrors which can likely change their adjustment over time. The laser was connected directly to the fiber-optic cable, and then to the spectrometer, which removed any effect of the background noise.

Finally, the fiber-optic cable was pointed directly at the lab lights to analyze the spectral signature of the fluorescent lighting and other light sources.

Results

The background measurements shown in fig. 1 were found not to be useful, as the two other measurements made without the black cloth covering would be substantially brighter than the background noise, and the black cloth would be used for all future, more carefully controlled tests. This measurement may still be relevant if any of the peaks sneak into any of the the other data they shouldn't be in.

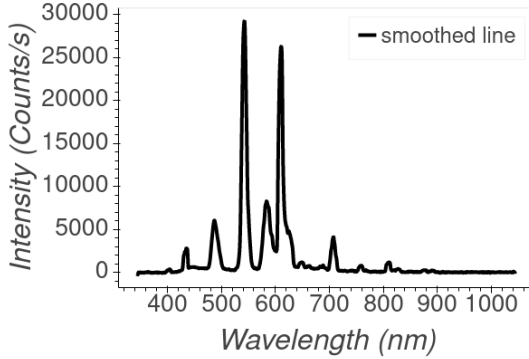


FIG. 1. Background light that reaches the test setup even while it is not pointed at the room.

The spectrometer measurement of the green laser was indeed found to be just over 533 nm, meaning that all wavelength measurements using the spectrometer must be adjusted by 1 nm. This was done proactively when loading the data from .txt file for the rest of the lab involving the spectrometer, so all reported graphs and numbers have been corrected by this constant amount of 1.094 nm.

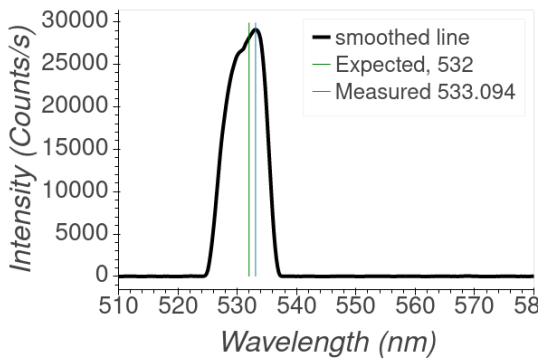


FIG. 2. The calibration of the spectrogram, performed with a green laser of known wavelength.

Finally the lab light spectrum was analyzed, and is given in fig. 3. The 6 largest peaks were found, and their characteristic linewidth calculated. The narrowest one was at 435.3 nm, which corresponds to a violet color,

with a linewidth of 1.9 nm.

It is interesting to briefly explore why there are separate peaks. Flourescent lights are percieved as white, but the white light source used later in the lab will have a significantly different spectral signature in spite of being percieved in much the same way. This is due to the nature of the photo-receptors in our eyes. The three color-receptors in fact overlap in the wavelengths they percieve, and the brain compensates for these overlaps by just adding them together. This adding together of percieved light is why, as we see in the table , orange + green + yellow + violet + red light = white light to our eyes. Flourescent bulbs create the illusion of white light by producing just a few different color lights, with a few different materials.

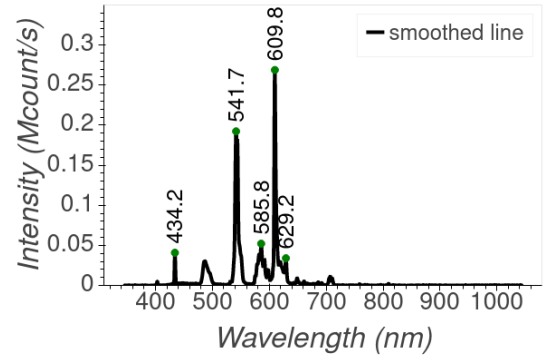


FIG. 3. The spectral signature of the laboratory, especially the flourescent lighting.

Wavelength (nm)	Intensity (cps)	Linewidth (nm)	Color
609.8	268854	3.3	Orange
541.7	192323	6.7	Green
585.8	52107	8.3	Yellow
434.2	40834	1.9	Violet
629.2	33929	2.2	Red

ABSORPTION SPECTRUM

Procedure

In this phase of the lab the absorption spectrum of the ruby sample is measured. Here we will use a white light source with a smooth output spectrum, and compare the spectral signature of white light that does not pass through the ruby and white light that does. The white light source was placed about 10 cm away from the FO cable connected to the spectrometer, and the ruby was mounted between them on a swivel to easily

move in and out of the way of the beam. After everything was secured and aligned, the black cloth was placed over the measurement setup to block out background noise. A measurement was made with and without the ruby crystal in the beam's path. Measurements on the spectrogram were calibrated to optimize the use of its measurement range, and the wavelengths measured were corrected by the above specified amount.

Results

The background noise was subtracted from each spectrum, and both are plotted in fig. 4.

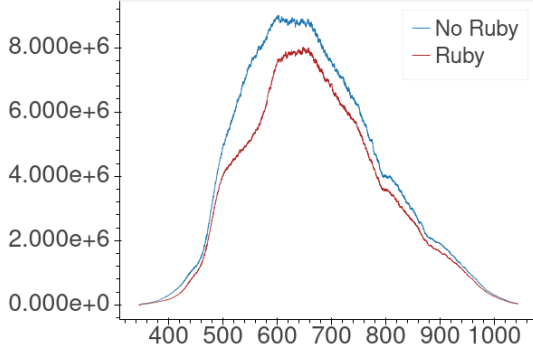


FIG. 4. A comparison of the spectral signature of the white light before and after passing through the ruby crystal.

Eq. 1 was used to calculate various facets of the expected and measured spectral signature of the ruby. First, we examine the expected transmission at wavelength 700 nm, assuming that $\alpha = 0$ and using the accepted value for n , 1.7. The exponent evaluates to 1, and thus we get $(1 - R)^2 = 0.870$. This value corresponds to the baseline (or maybe ceiling line) for how high transmission can be at any wavelength regardless of absorption at that wavelength, due to the portion of the signal that will be reflected due to the mismatch in n .

$$I = I_0(1 - R)^2 e^{-\alpha L} \quad (1)$$

Eq. 1 relates the intensity of light traveling through air I to the intensity of light traveling through a material I_0 , a reflection coefficient R , the absorption of the material α , and the length of the material L .

The transmission spectrum, given by eq. 2, was computed and plotted in 5

$$T(\lambda) = I(\lambda)/I_0(\lambda) \quad (2)$$

where λ is the wavelength of light, and I and I_0 are functions of λ that relate to the medium.

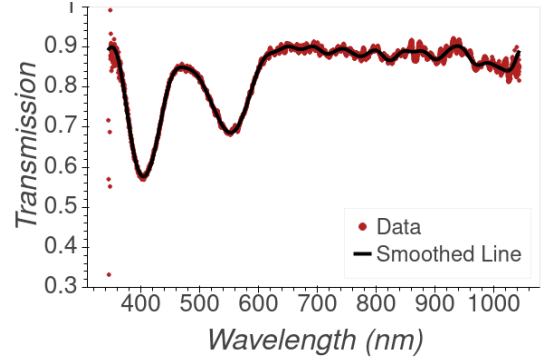


FIG. 5. The transmission rate through a ruby sample.

The spectrum of the absorption coefficient as a function of wavelength, $\alpha(\lambda)$ was computed via eq. 3, which was derived from eq. 1, and plotted in fig. 6.

$$\alpha = \frac{-1}{L} \cdot \ln \left(\frac{T}{(1 - R)^2} \right) \quad (3)$$

This equation is derived from 1 and has the same constants.

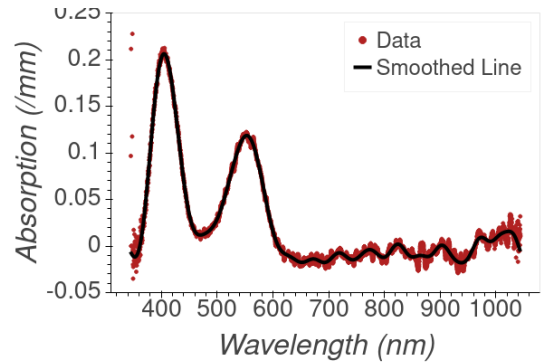


FIG. 6. The absorption of a ruby crystal, $\alpha(\lambda)$

The absorption peaks are centered at 400 and 550 nm, which correspond to the wavelengths of light that the ruby absorbs. These correspond to, essentially, the other colors that aren't red, showing that the ruby appears red due to the red light passing through it. The wavelength, widths and absorption lengths are given in table I. The uncertainty displayed in the table was found by averaging the difference between the raw, noisy data, and the smoothed line (smoothed using a simple lowpass filter in the frequency domain).

Peak	Wavelength (nm)	Width (nm)	alpha (m^{-1})	1/alpha (m)
1	404	56.0	0.206+/-0.005	4.9 +/- 0.1
2	552	66.5	0.118+/-0.005	8.4 +/- 0.4

TABLE I.

RUBY FLUORESCENCE SPECTRUM

Procedure

The optical setup was arranged to measure the emission spectrum of the ruby spectrum, excited by a green diode laser, shown in Fig. 7. All optical components were aligned by their height above the breadboard, and the 600 μm fiberoptic cable was mounted pointing toward the ruby crystal. The beam's intensity was reduced by a neutral density filter, and a convex collection lens was used to focus any output from the ruby towards the optical fiber. The lens, of focal length 25 mm, was placed such that the image of the ruby would hit the FO with a magnification of 1, which requires that both objects be set an equal distance from the lens.

$$\frac{1}{f} = \frac{1}{o} + \frac{1}{i} \quad (4)$$

Using the thin lens formula, where the focal length f is related to object distance and image distance, given by 4, the optimal distances were calculated to be 50 mm away from the lens.

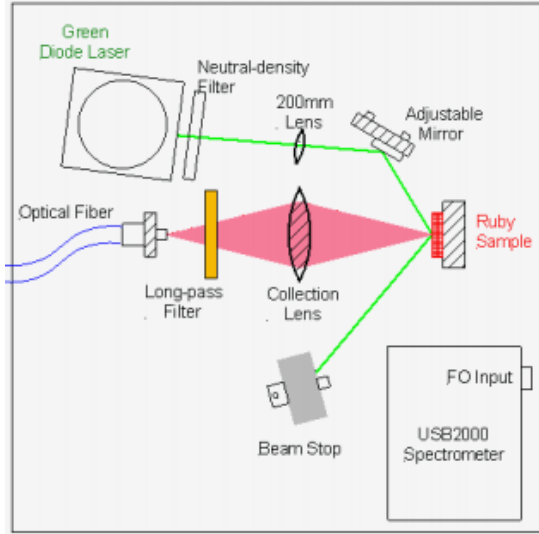


FIG. 7. Test setup

The setup was aligned by replacing the spectrogram at the opposite end of the collection FO cable with a

white light source, and aligning the image of the light, as well as the green laser, directly on the ruby crystal. By replacing the input with an output, we can be sure that once the spectrogram is hooked up, it is perfectly aligned to take data from the ruby crystal. After this alignment, the spectrogram was hooked up to the FO cable and the black cloth was placed over the apparatus.

Results

The measured emission spectrum is shown in 8. The most prominent peak, the R-line emission of the crystal, is located at $693.2 \pm 0.4 \text{ nm}$. This uncertainty was calculated based on 2x the measurement precision of the spectrometer. This is given considering that, while the low-pass filter is effective at removing the most significant source of noise, high-frequency electronic noise, it is still true that different filtering techniques will yield different values, and could reasonably correspond to a range of 5 “markings” on the measurement device. A brief exploration of this filtering is shown in the appendix. This value is significant because it corresponds to the difference between the energy level of the metastable state available in ruby crystals, and the ground state, and with a theoretical value of 694 nm our measurement is slightly below the accepted value. The other peak at 532 nm is due to the green laser reflecting off the surface of the ruby.

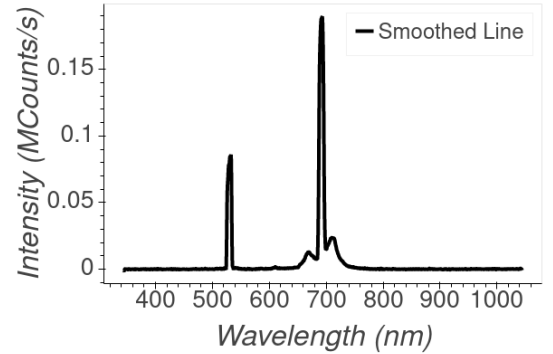


FIG. 8. Emission of a ruby sample excited by a green laser light.

FLUORESCENCE LIFETIME OF RUBY R-LINE

Procedure

The optical setup was adjusted to measure the fluorescence of the ruby crystal in the time-domain. A function generator was used to produce 100 ms long laser burst, and this signal was used to trigger an oscilloscope on the

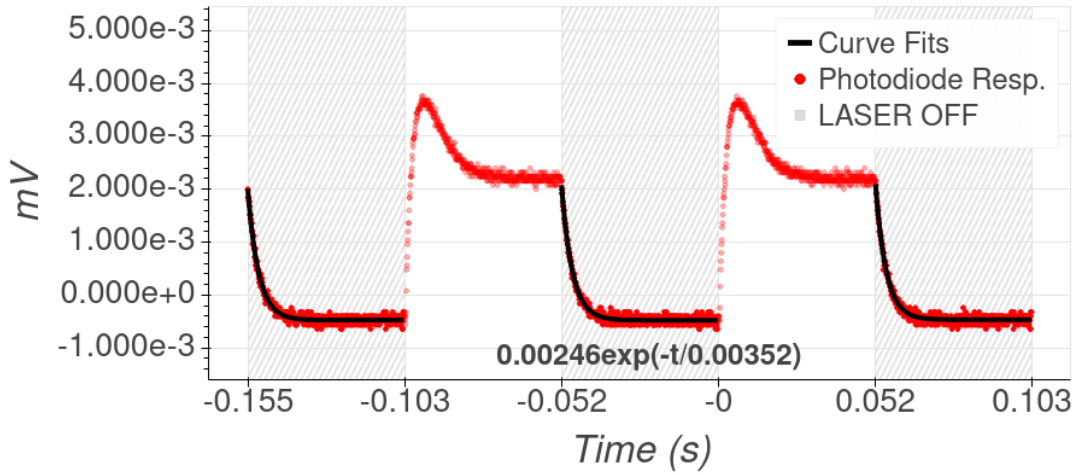


FIG. 10. Three observations of the decay time of a fluorescent ruby sample following short laser pulses, with the average fit equation given.

edge of the square wave. The photodiode (PD) was connected to the FO and used to measure the fluorescence of the ruby and produce a signal on channel 2 of the oscilloscope. A filter was placed between the lens and the FO to remove the green laser light reflected from the ruby, as discussed in the previous section.

Results

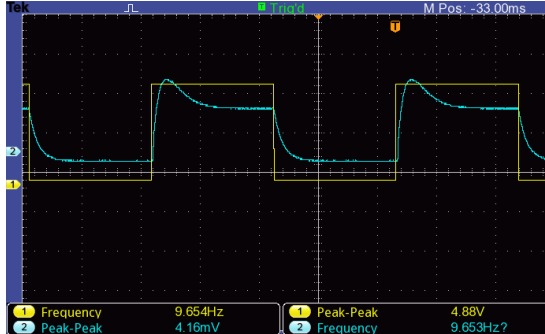


FIG. 9.

Three complete periods of the PD output were captured and recorded, with the raw data presented in fig. 9. An inverse exponential fit (with offsets that are not shown) was applied to each curve, beginning at the moment the laser was powered off. After aligning the time axis such that 0 s is the moment a given laser burst ends, and 0 mV is the asymptote of the curve, the equation is given by eq. 5, with the caveat that in our measurements N_0 is not measured, rather a signal in mV.

$$N(t) = N_0 e^{-t/\tau} \quad (5)$$

This equation relates the number of electrons in an excited state, N , to the amount of electrons originally in the excited state N_0 as that number decays through time with a decay constant τ .

The fits are shown in 10, along with the equation given by the mean of each variable. The lifetime of R-line ruby fluorescence, τ , was calculated at 3.52 ± 0.04 ms, with the uncertainty given by the standard deviation across the three fits. This value overlaps with the accepted value of $3.5 \pm .4$ ms.

SUMMARY

The absorption and fluorescence of a ruby crystal was analyzed, with values for transmission at 700 nm, peak absorption length, R-line wavelength, and fluorescence lifetime calculated.

In the first phase of the lab, a white light source was used to produce a continuous curve of transmission and absorption as a function of wavelength. Using the accepted value for n , the index of refraction for a ruby crystal, an accepted value for the transmission at 700 nm, assuming zero absorption, was calculated to be 0.851. The experimental value attained for this

transmission was 0.91 ± 0.05 , which is high. This error may stem from the shape of the white light, which was not well optimized towards the collection zone, or on the ruby itself.

An absorption curve was derived from the transmission curve, and the main peaks were analyzed. While it is difficult to find an accepted value for the peak absorption and its corresponding absorption length, $1/\alpha$, these numbers do square with the simple observation that rubies are red, and therefore absorb the wavelengths of light that correspond to colors other than red. There were definitely issues with this test setup, given the lowest values for absorption were below zero which is not possible. It is likely that some white light was still hitting the FO input even after the ruby was inserted, skewing the results as though more light was getting through than should have been able to. It may also be that this particular sample had a lower index of refraction than the accepted value, which would result in less light being reflected and would also explain our observations.

In the next phase of the lab fluorescence was measured for its R-line wavelength and lifetime. The measured value was 693.2 ± 0.4 nm, reasonably close to the expected 694 nm. Some of this error may be simply due to electronic noise. Interestingly, calibration of the spectrometer using green laser light yielded a correction of an about 1 nm subtraction. It may have been that this calibration was done incorrectly, as the green light peak did not have a sharp peak, and that the measured value would be correct if not for miscalibration.

The lifetime of ruby fluorescence was measured by performing several inverse exponential fits on the ruby emissions after short laser pulses. While the data was quite noisy at this granularity, the fit parameters were very consistent and accurate. At 3.52 ± 0.04 ms they overlapped perfectly with the accepted value of 3.5 ms. This more precise measurement may be the result of more precise controls being possible with this setup, and it also may be that the oscilloscope, measuring a net voltage from a photodiode, is a more accurate measurement device than a spectrogram, which relies on measuring very many different wavelengths and relies on precisely controlling the incoming beam.

TABLE II. Measured and accepted values of various fluorescent properties of a ruby crystal.

Property	Measured	Accepted	Ref.	Deviation
700nm Transmission	0.91 ± 0.05	0.851	[2]	2σ
Peak Absorption Length	4.9 ± 0.1 mm	-	-	-
R-Line Wavelength	693.2 ± 0.4 nm	694 nm	[3]	-2σ
Lifetime τ	3.52 ± 0.04 ms	3.5 ± 0.5 ms	[4]	0σ

-
- * smith.tr@northeastern.edu; <https://github.com/trevorm4x/>
- [1] ThoughtCo, Understanding Visible Light
<https://www.thoughtco.com/understand-the-visible-spectrum-608329>
- [2] Gem Society, Table of Refractive Indices and Double Refraction of Selected Gems
<https://www.gemsociety.org/article/table-refractive->

-
- [index-double-refraction-gems/](#)
- [3] Physics Open Lab, Ruby Crystal Fluorescence
<https://physicsopenlab.org/2020/06/15/ruby-crystal-fluorescence/>
- [4] SpringerLink, Ruby Crystal for Demonstrating Time- and Frequency-Domain Methods of Fluorescence Lifetime Measurements
<https://link.springer.com/article/10.1007/s10895-006-0123-7>

APL_2_cell

June 6, 2021

```
[365]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from uncertainties import ufloat

from jupyterthemes import jtplot
jtplot.style(theme='monokai', context='notebook', ticks=False, grid=False)

import matplotlib as mpl

# Edit the font, font size, and axes width
# mpl.rcParams['font.family'] = 'Avenir'
plt.rcParams['font.size'] = 24
plt.rcParams['axes.linewidth'] = 2

%load_ext autoreload
%autoreload 2

import sys
sys.path.insert(0, '/home/trevormjs/Documents/Science/APL/Lab')

from Helper.plotting import my_graph
```

0.0.1 Part A

Measurements

```
[373]: silicon_thickness = 0.54 # mm
silicon_dimensions = [32.33, 9.04]
wire_colors = {
    1: 'purple',
    2: 'blue',
    3: 'green',
    4: 'black',
    5: 'white',
    6: 'grey'
}
color_wires = {i:k for k, i in wire_colors.items()}
```



```
[374]: resistances = pd.DataFrame({
    'first': [
        'green', 'green', 'green',
        'green', 'green', 'blue',
        'blue', 'blue', 'blue',
        'grey', 'grey', 'grey',
        'white', 'white', 'black',
    ],
    'second': [
        'blue', 'grey', 'white',
        'black', 'purple', 'grey',
        'white', 'black', 'purple',
        'white', 'black', 'purple',
        'black', 'purple', 'purple',
    ],
    'resistance': [
        22.76e3, 24.55e3, 66.43e3,
        26.94e3, 28.57e3, 11.213e3,
        12.445e3, 12.06e3, 11.665e3,
        51.39e3, 8.8936e3, 9.566e3,
        171.76e3, 176.26e3, 12.296e3,
    ]
})
```

```
[375]: resistances.insert(2, '#1', [color_wires[color] for color in_
    ↳resistances['first']])
resistances.insert(3, '#2', [color_wires[color] for color in_
    ↳resistances['second']])
```

```
[376]: resistances.insert(
    5, 'wire-pair', resistances[['#1', '#2']].apply(lambda x: '-'.join(x.
    ↳astype(str).tolist()), axis = 1))
```

```
[377]: resistances.index = resistances['wire-pair']
```

```
[378]: silicon_dimensions, silicon_thickness
```

```
[378]: ([32.33, 9.04], 0.54)
```

```
[379]: print(resistances[['resistance']].to_latex())
```

```
\begin{tabular}{lr}
\toprule
{} & resistance \\
wire-pair & \\
\midrule
3-2 & 22760.0 \\
3-6 & 24550.0
```



```

3-5      &      66430.0 \\
3-4      &      26940.0 \\
3-1      &      28570.0 \\
2-6      &      11213.0 \\
2-5      &      12445.0 \\
2-4      &      12060.0 \\
2-1      &      11665.0 \\
6-5      &      51390.0 \\
6-4      &      8893.6 \\
6-1      &      9566.0 \\
5-4      &      171760.0 \\
5-1      &      176260.0 \\
4-1      &      12296.0 \\
\bottomrule
\end{tabular}

```

Data

```

[380]: SI_unit_dict = {
    -2: 'c',
    -3: 'm',
    -6: '\u03BC',
    -9: 'n',
    0: '',
    3: 'K',
    6: 'M',
    9: 'G',
    12: 'T'
}

class value_unit():
    def __init__(self, value, unit_scale, unit_name):
        self.value = value
        self.unit_scale = unit_scale
        self.unit_name = unit_name
        self.make_unit_annotation()

    def make_unit_annotation(self):
        self.unit_annotation = SI_unit_dict.get(
            self.unit_scale) + self.unit_name

    def __format__(self):
        print(self.value, self.unit_annotation)

```

```

[381]: silicon_dimensions = [9.4, 33.5]

```

```
[382]: width = value_unit(9.4, -6, 'm')
```

```
[383]: width.print()
```

```
-----  
AttributeError                                Traceback (most recent call last)  
<ipython-input-383-e320bc9f0a1e> in <module>  
----> 1 width.print()  
  
AttributeError: 'value_unit' object has no attribute 'print'
```

```
[387]: voltages = pd.DataFrame(columns=['current', 'voltage', 'resistance', 'power'])
```

```
def add_row(c, v, r=12296.0):  
    voltages.loc[voltages.shape[0], :] = [c, v, r, c*v]  
    display(voltages.iloc[-1])
```

Adding

```
[388]: add_row(0.0022, 0.0183)  
  
add_row(0.0071, 0.0836)  
  
add_row(0.0128, 0.2103)  
  
add_row(0.0201, 0.4462)  
  
add_row(0.0441, 1.0425)  
  
add_row(0.0656, 1.501)  
  
add_row(-0.0011, -0.0196)  
  
add_row(-0.0044, -0.0603)  
  
add_row(-0.0105, -0.1550)  
  
add_row(-0.0133, -0.2201)  
  
add_row(-0.0236, -0.5823)  
  
add_row(-0.0306,  
        -0.9142)  
  
add_row(-0.0352,  
        -1.1616)
```

```
add_row(-0.0383,  
        -1.3490)
```

```
add_row(-0.0424,  
        -1.5880)
```

```
add_row(-0.0482,  
        -1.9278)
```

```
voltages
```

```
current      0.0022  
voltage      0.0183  
resistance   12296.0  
power        0.00004  
Name: 0, dtype: object
```

```
current      0.0071  
voltage      0.0836  
resistance   12296.0  
power        0.000594  
Name: 1, dtype: object
```

```
current      0.0128  
voltage      0.2103  
resistance   12296.0  
power        0.002692  
Name: 2, dtype: object
```

```
current      0.0201  
voltage      0.4462  
resistance   12296.0  
power        0.008969  
Name: 3, dtype: object
```

```
current      0.0441  
voltage      1.0425  
resistance   12296.0  
power        0.045974  
Name: 4, dtype: object
```

```
current      0.0656  
voltage      1.501  
resistance   12296.0  
power        0.098466  
Name: 5, dtype: object
```

```
current      -0.0011  
voltage      -0.0196  
resistance   12296.0
```

```

power          0.000022
Name: 6, dtype: object

current        -0.0044
voltage        -0.0603
resistance     12296.0
power          0.000265
Name: 7, dtype: object

current        -0.0105
voltage        -0.155
resistance     12296.0
power          0.001628
Name: 8, dtype: object

current        -0.0133
voltage        -0.2201
resistance     12296.0
power          0.002927
Name: 9, dtype: object

current        -0.0236
voltage        -0.5823
resistance     12296.0
power          0.013742
Name: 10, dtype: object

current        -0.0306
voltage        -0.9142
resistance     12296.0
power          0.027975
Name: 11, dtype: object

current        -0.0352
voltage        -1.1616
resistance     12296.0
power          0.040888
Name: 12, dtype: object

current        -0.0383
voltage        -1.349
resistance     12296.0
power          0.051667
Name: 13, dtype: object

current        -0.0424
voltage        -1.588
resistance     12296.0
power          0.067331
Name: 14, dtype: object

current        -0.0482

```

```

voltage      -1.9278
resistance    12296.0
power         0.09292
Name: 15, dtype: object

```

```

[388]:
current voltage resistance    power
0    0.0022  0.0183    12296.0  0.00004
1    0.0071  0.0836    12296.0  0.000594
2    0.0128  0.2103    12296.0  0.002692
3    0.0201  0.4462    12296.0  0.008969
4    0.0441  1.0425    12296.0  0.045974
5    0.0656  1.501     12296.0  0.098466
6   -0.0011 -0.0196    12296.0  0.000022
7   -0.0044 -0.0603    12296.0  0.000265
8   -0.0105 -0.155     12296.0  0.001628
9   -0.0133 -0.2201    12296.0  0.002927
10  -0.0236 -0.5823    12296.0  0.013742
11  -0.0306 -0.9142    12296.0  0.027975
12  -0.0352 -1.1616    12296.0  0.040888
13  -0.0383 -1.349     12296.0  0.051667
14  -0.0424 -1.588     12296.0  0.067331
15  -0.0482 -1.9278    12296.0  0.09292

```

```

[389]: print(voltages.to_latex())

```

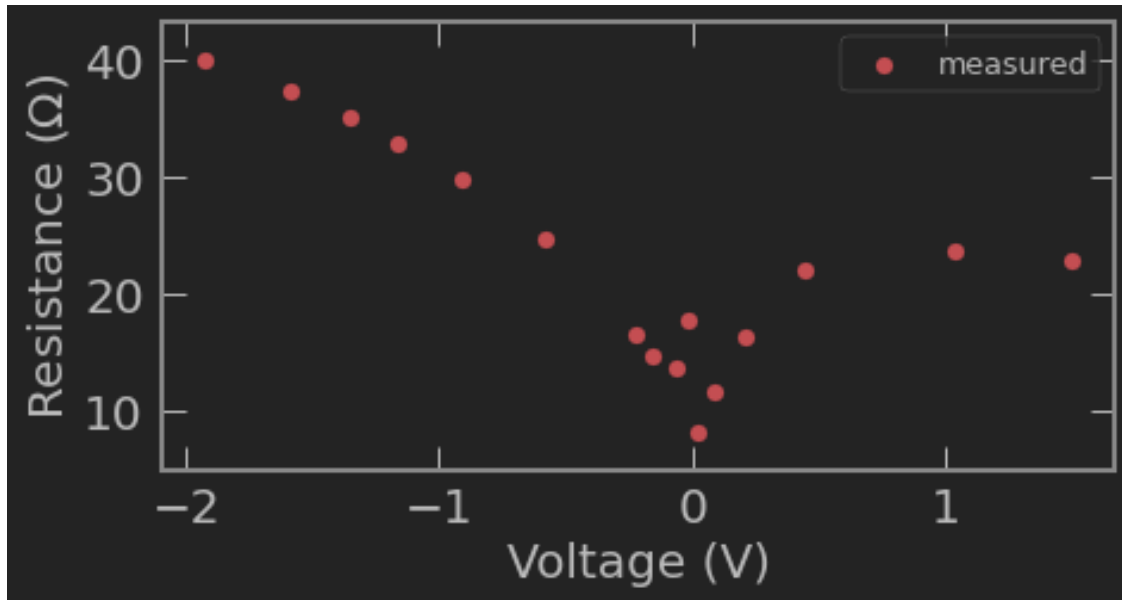
```

\begin{tabular}{llllll}
\toprule
{} & current & voltage & resistance & & power \\
\midrule
0 & 0.0022 & 0.0183 & 12296.0 & & 0.00004 \\
1 & 0.0071 & 0.0836 & 12296.0 & & 0.000594 \\
2 & 0.0128 & 0.2103 & 12296.0 & & 0.002692 \\
3 & 0.0201 & 0.4462 & 12296.0 & & 0.008969 \\
4 & 0.0441 & 1.0425 & 12296.0 & & 0.045974 \\
5 & 0.0656 & 1.501 & 12296.0 & & 0.098466 \\
6 & -0.0011 & -0.0196 & 12296.0 & & 0.000022 \\
7 & -0.0044 & -0.0603 & 12296.0 & & 0.000265 \\
8 & -0.0105 & -0.155 & 12296.0 & & 0.001628 \\
9 & -0.0133 & -0.2201 & 12296.0 & & 0.002927 \\
10 & -0.0236 & -0.5823 & 12296.0 & & 0.013742 \\
11 & -0.0306 & -0.9142 & 12296.0 & & 0.027975 \\
12 & -0.0352 & -1.1616 & 12296.0 & & 0.040888 \\
13 & -0.0383 & -1.349 & 12296.0 & & 0.051667 \\
14 & -0.0424 & -1.588 & 12296.0 & & 0.067331 \\
15 & -0.0482 & -1.9278 & 12296.0 & & 0.09292 \\
\bottomrule
\end{tabular}

```

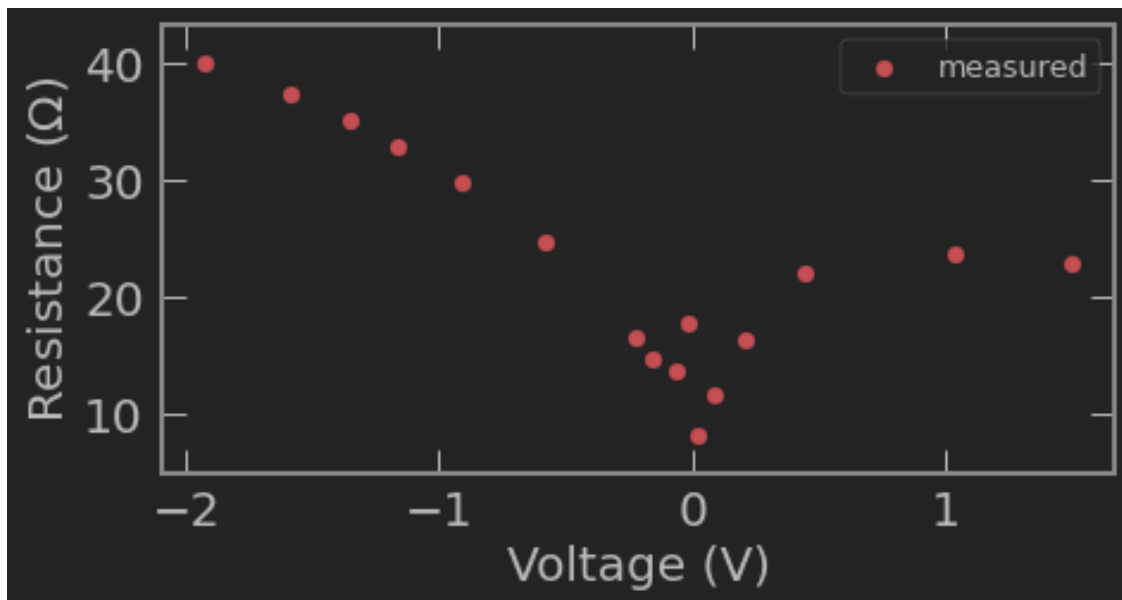
Results

```
[436]: my_graph(voltages.loc[voltages.voltage.sort_values().index].voltage, voltages.  
→loc[voltages.voltage.sort_values().index].current,  
→'linear','linear','Voltage (V)', 'Current (A)', 'l2_a_1', 'linear')
```



```
[391]: rohms = 'Resistance ($\\mathregular{\\Omega}$)'
```

```
[392]: my_graph(voltages.voltage, voltages.voltage/voltages.current,  
→'linear','linear','Voltage (V)', rohms, 'l2_a_2', '', )
```



0.0.2 Part B

Measurements

```
[495]: volts = 0.9083
      amps = 0.0396e-3

      fourwire = pd.DataFrame({
          'start': [
              'blue', 'white',
          ],
          'end': [
              'green', 'grey',
          ],
          'voltage': [
              5.84e-3, 9.33e-3
          ]
      })
```

Calculations

```
[496]: fourwire.insert(3, 'resistance', fourwire.voltage/amps)
      fourwire.insert(4, 'distance', [9.46, 11.63])
      fourwire.insert(5, 'path', [
          f"{color_wires[s]}-{color_wires[e]}" for s, e in zip(fourwire.
          ↪start, fourwire.end)])

      fourwire.index = fourwire.path

      csa = cross_sectional_area = silicon_dimensions[0] * \
          1e-3 * silicon_thickness*1e-3
      csa
```

```
[496]: 5.076000000000001e-06
```

$$R = \rho \cdot \frac{L}{A}$$
$$\rho = \frac{R \cdot A}{L}$$

```
[497]: fourwire.insert(5, 'rho', fourwire.resistance * csa / (fourwire.distance*1e-3))

      fourwire.insert(6, 'measured_R', [20.2e3, 213.6e3])

      fourwire.insert(6, 'current (mA)', amps*1e3)

      fourwire.insert(6, 'area ($m^2$)', csa)
```



```
fourwire.axes[0].name = None

fourwire.resistance = fourwire.resistance.astype(int)
```

```
[503]: print(fourwire[['resistance', 'measured_R', 'rho']].to_latex())
```

```
\begin{tabular}{lrrr}
\toprule
{} & resistance & measured\_R & rho \\
\midrule
2-3 & 147 & 20200.0 & 0.079131 \\
5-6 & 235 & 213600.0 & 0.102832 \\
\bottomrule
\end{tabular}
```

```
[397]: from Helper.numbers import print_unc
```

```
[398]: 3.5 *1e-6*1e-2
```

```
[398]: 3.5e-08
```

```
[500]: _ = print_unc(fourwire.rho.mean(), fourwire.rho.std())
```

```
0.09 +- 0.02
```

0.0.3 Hall effect, carrier concentration and mobility

Calibration of the Electromagnet

```
[400]: data = pd.read_excel('./Lab2_HALL_alpha_new.xlsx', skiprows = 1)
```

Fit

```
[401]: alpha, beta = data.iloc[3, -2:]
data = data.iloc[:,1:3]
data
```

```
[401]:
```

	Im (A)	B (T)
0	-3.00	-0.487
1	-2.50	-0.408
2	-2.00	-0.327
3	-1.50	-0.246
4	-1.00	-0.164
5	-0.50	-0.082
6	0.00	0.001
7	0.50	0.078
8	1.00	0.160
9	1.50	0.240
10	2.00	0.321

```

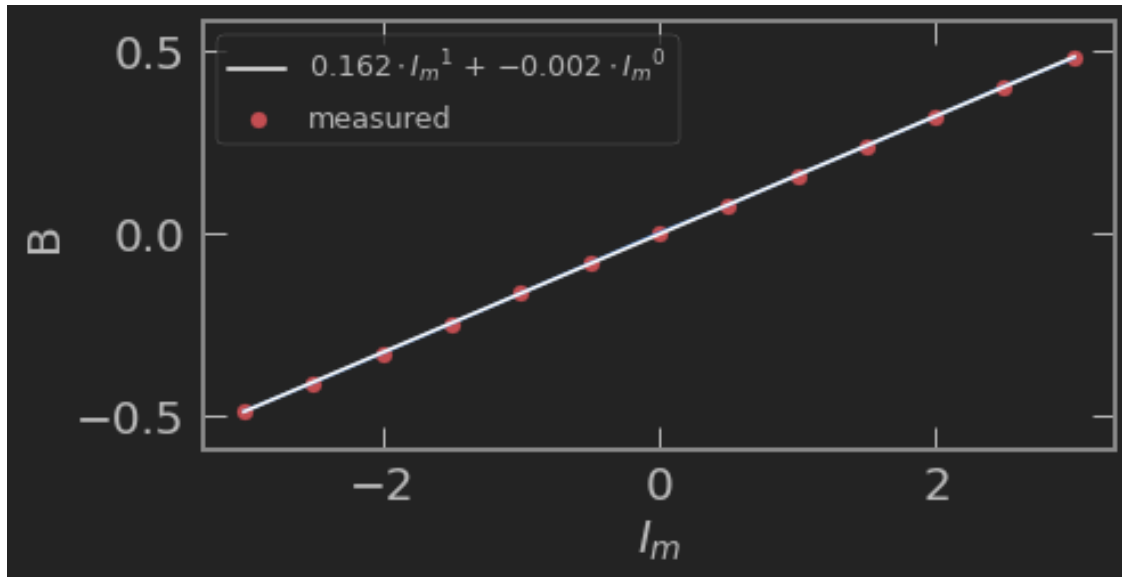
11    2.49  0.402
12    3.00  0.481

```

```

[514]: my_graph(data['Im (A)'], data['B (T)'],
               'linear', 'linear', '$I_m$', 'B', 'l2_d_4', 'linear', 1)

```



```

[514]: array([ 0.16180289, -0.00226015])

```

Cross-voltages

```

[402]: current = 7.91e-3

```

```

[403]: cross_voltage = {'grey-blue': 73.9e-3,
                       'white-green': .2906}
cross_voltage

```

```

[403]: {'grey-blue': 0.0739, 'white-green': 0.2906}

```

Measure Hall Voltage

Data

```

[404]: h_v = pd.DataFrame(columns=['Is', 'Im', 'Vh'])

def add_row(Is, Im, Vh):
    h_v.loc[h_v.shape[0], :] = [Is, Im, Vh]
    # display(h_v.iloc[-1])

add_row(7.86e-3, 0, 71.22e-3)
add_row(7.87e-3, .18, 70.23e-3)

```

```

add_row(7.87e-3, .36, 68.91e-3)
add_row(7.917e-3, .66, 68.15e-3)
add_row(7.906e-3, 1.06, 64.91e-3)
add_row(7.906e-3, 1.53, 62.28e-3)
add_row(7.912e-3, 2.01, 59.28e-3)
add_row(7.911e-3, 2.32, 57.37e-3)
add_row(7.874e-3, 2.69, 54.48e-3)
add_row(7.901e-3, 2.98, 53.01e-3)
add_row(7.913e-3, -1e-10, 72.67e-3)
add_row(7.913e-3, -.32, 74.46e-3)
add_row(7.940e-3, -.56, 76.44e-3)
add_row(7.933e-3, -.85, 77.53e-3)
add_row(7.950e-3, -1.01, 79.10e-3)
add_row(7.929e-3, -1.30, 81.22e-3)
add_row(7.918e-3, -1.67, 83.04e-3)
add_row(7.925e-3, -2.00, 85.10e-3)
add_row(7.941e-3, -2.34, 87.73e-3)
add_row(7.927e-3, -2.63, 89.43e-3)
add_row(7.952e-3, -3.00, 92.15e-3)

```

h_v

```

[404]:
      Is      Im      Vh
0    0.00786      0 0.07122
1    0.00787  0.18 0.07023
2    0.00787  0.36 0.06891
3    0.007917 0.66 0.06815
4    0.007906 1.06 0.06491
5    0.007906 1.53 0.06228
6    0.007912 2.01 0.05928
7    0.007911 2.32 0.05737
8    0.007874 2.69 0.05448
9    0.007901 2.98 0.05301
10   0.007913 -0.0 0.07267
11   0.007913 -0.32 0.07446
12   0.00794 -0.56 0.07644
13   0.007933 -0.85 0.07753
14   0.00795 -1.01 0.0791
15   0.007929 -1.3 0.08122
16   0.007918 -1.67 0.08304
17   0.007925 -2.0 0.0851
18   0.007941 -2.34 0.08773
19   0.007927 -2.63 0.08943
20   0.007952 -3.0 0.09215

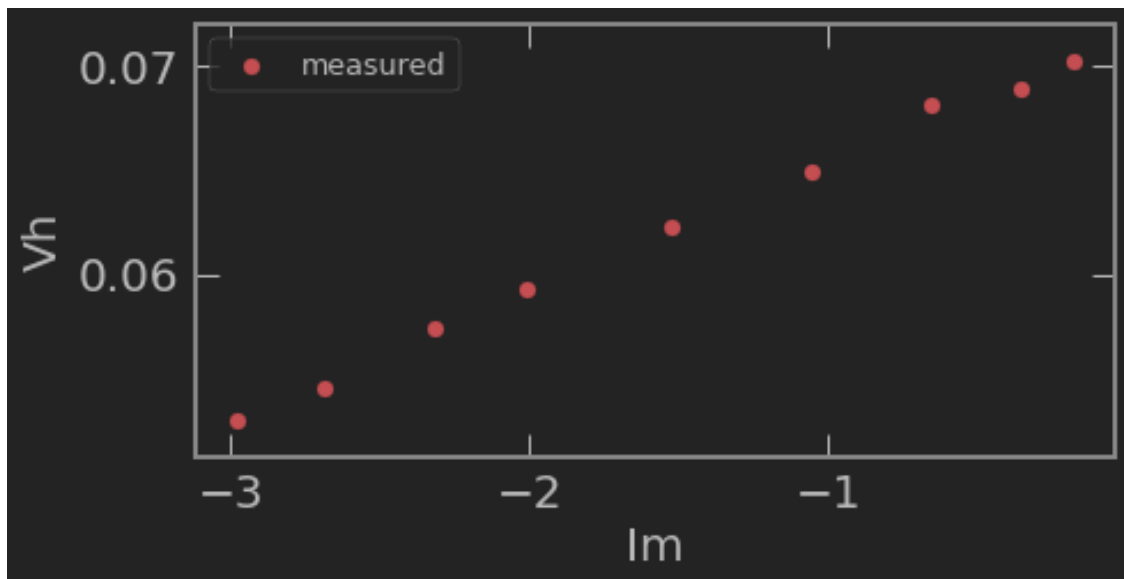
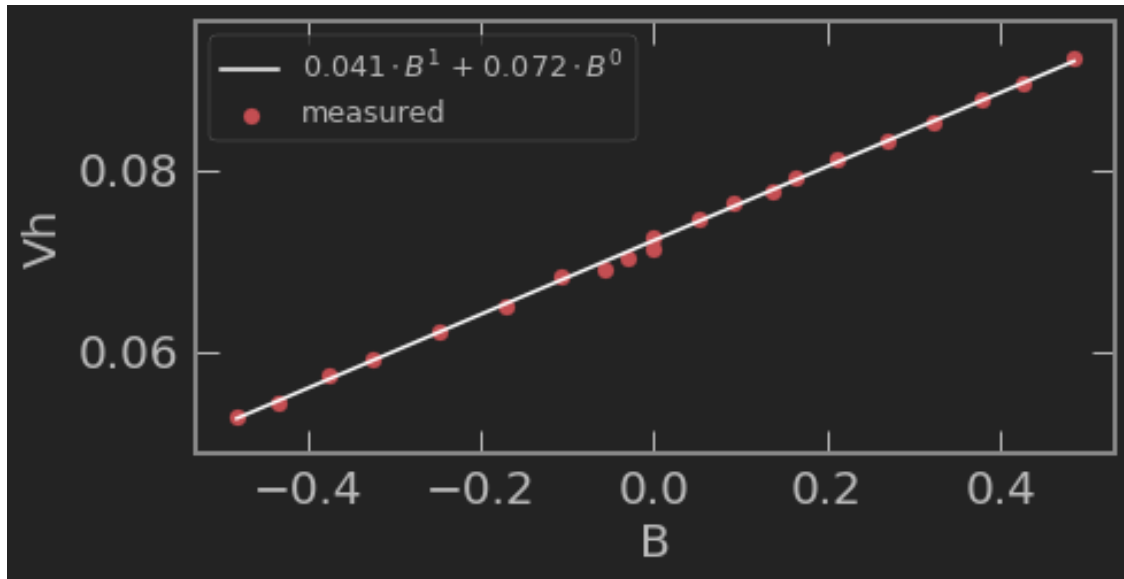
```

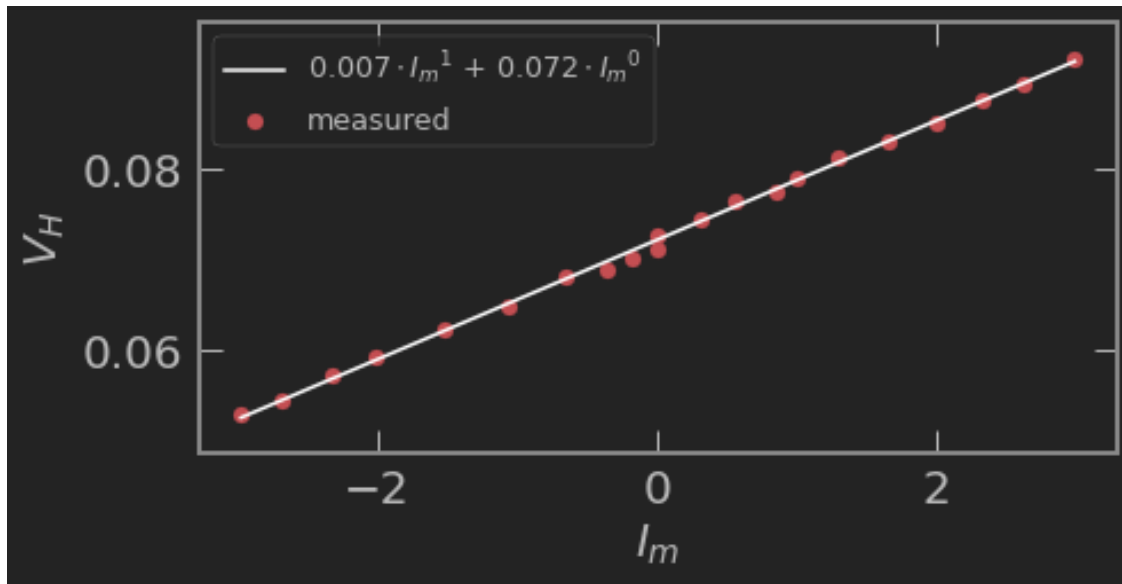
```

[405]: h_v.Im *= -1

```

```
[515]: my_graph(h_v['Im'].loc[h_v.Im >= 0],
          h_v['Vh'].loc[h_v.Im >= 0],
          'linear', 'linear', 'Im', 'Vh', 'l2_d_4a_pos', '')
my_graph(h_v['Im'].loc[h_v.Im < 0],
          h_v['Vh'].loc[h_v.Im < 0],
          'linear', 'linear', 'Im', 'Vh', 'l2_d_4a_neg', '')
my_graph(h_v['Im'], h_v['Vh'],
          'linear', 'linear',
          '$I_m$', '$V_H$', 'l2_d_4a_all', '', 1)
```





```
[515]: array([0.00655619, 0.07220566])
```

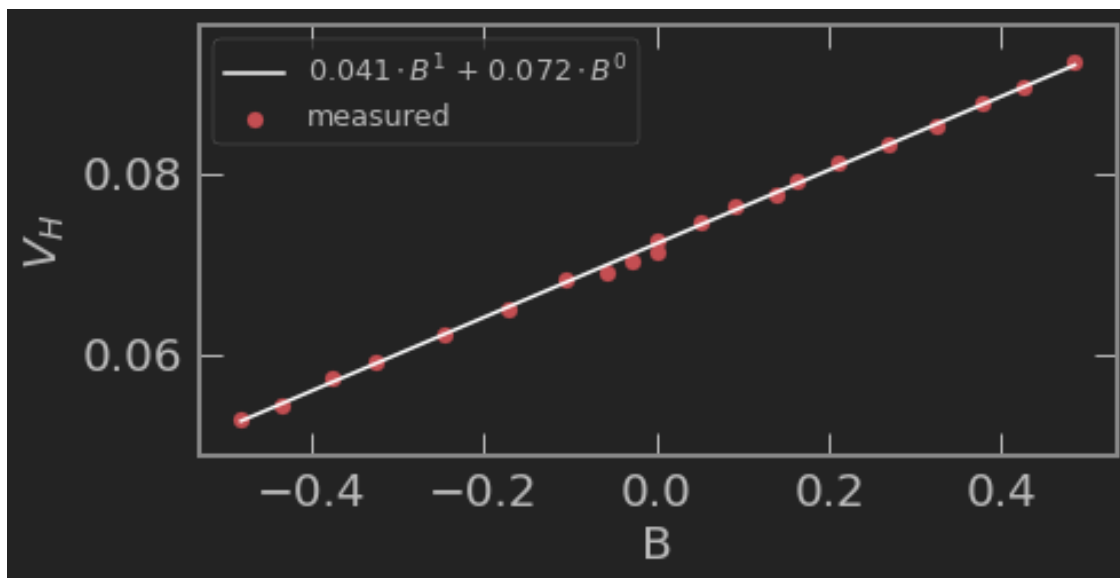
```
[407]: h_v.insert(3, 'B', alpha*h_v.Im)
h_v
```

```
[407]:
```

	Is	Im	Vh	B
0	0.00786	0	0.07122	0.0
1	0.00787	-0.18	0.07023	-0.029125
2	0.00787	-0.36	0.06891	-0.058249
3	0.007917	-0.66	0.06815	-0.10679
4	0.007906	-1.06	0.06491	-0.171511
5	0.007906	-1.53	0.06228	-0.247558
6	0.007912	-2.01	0.05928	-0.325224
7	0.007911	-2.32	0.05737	-0.375383
8	0.007874	-2.69	0.05448	-0.43525
9	0.007901	-2.98	0.05301	-0.482173
10	0.007913	0.0	0.07267	0.0
11	0.007913	0.32	0.07446	0.051777
12	0.00794	0.56	0.07644	0.09061
13	0.007933	0.85	0.07753	0.137532
14	0.00795	1.01	0.0791	0.163421
15	0.007929	1.3	0.08122	0.210344
16	0.007918	1.67	0.08304	0.270211
17	0.007925	2.0	0.0851	0.323606
18	0.007941	2.34	0.08773	0.378619
19	0.007927	2.63	0.08943	0.425542
20	0.007952	3.0	0.09215	0.485409

```
[408]: from Helper.numbers import get_leading_figure
```

```
[531]: [dvh_db, inter] = my_graph(h_v['B'], h_v['Vh'],  
    'linear', 'linear', 'B', '$V_H$', 'l2_d_4b', '', 1)  
dvh_db
```



```
[531]: 0.04051960131985295
```

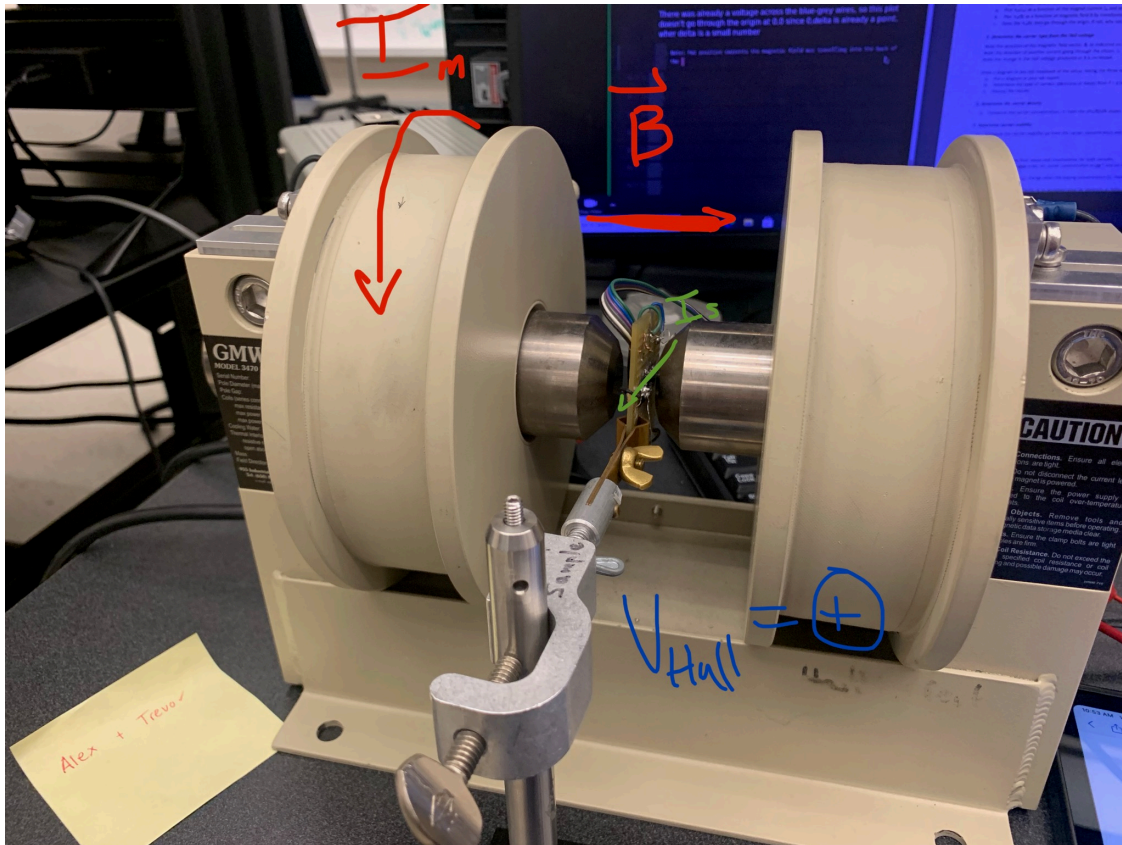
There was already a voltage across the blue-grey wires, so this plot doesn't go through the origin at 0,0 since 0,delta is already a point, wher delta is a small number

Note: For positive currents the magnetic field was travelling into the back of the chip, for negative currents the field was travelling into the top of the chip.

Current was travelling from blue to black

```
[410]: from IPython.display import Image  
Image("./messages_0.jpeg", width = 900)
```

```
[410]:
```



Determine carrier density

```
[411]: v_per_b = ufloat(.04051960131985295, 0.00035729666185760475)
```

```
[534]: errs = dvh_db*h_v['B'] - h_v['Vh'] + inter
```

```
[536]: abs(errs).mean()
```

```
[536]: 0.0003572966618576107
```

$$n = \frac{BI}{V_H e d}$$

$$n = \frac{B}{V_H} \frac{I}{e d}$$

$$n = \frac{dV_H^{-1}}{dB} \frac{I}{e d}$$

```
[414]: I_s_mean, I_s_unc = h_v.Is.mean(), h_v.Is.std()
```



```
[415]: I_s = ufloat(I_s_mean, I_s_unc)
```

```
[416]: e = 1.60217662e-19 # coulombs
```

```
[417]: d = ufloat(silicon_thickness, .02)*1e-3
print_unc(d)
```

0.00054 +- 0.00002

```
[417]: (0.00054, 2e-05, 5)
```

```
[547]: print(pd.DataFrame(
    {u: [v.n, v.s] for v, u in zip([v_per_b, I_s, d],
    ['dV_H/dB (V/T)', 'I (A)', 'd (m)'])},
    index=['Measured Value', 'Uncertainty']).T.to_latex())
```

```
\begin{tabular}{lrrr}
\toprule
{} & Measured Value & & Uncertainty \\
\midrule
dV\_H/dB (V/T) & & 0.040520 & & 0.000357 \\
I (A) & & 0.007913 & & 0.000026 \\
d (m) & & 0.000540 & & 0.000020 \\
\bottomrule
\end{tabular}
```

```
[549]: from Helper import numbers
```

```
[554]: n = v_per_b**-1 * I_s_mean / (e*d)
n.n* 10**numbers.get_leading_figure(n.n), n.s* 10**numbers.get_leading_figure(n.
↪n), f'e{numbers.get_leading_figure(n.n)}' # per m^3
```

```
[554]: (2.2571414938221857, 0.08593447778863826, 'e-21')
```

Determine Carrier Mobility

```
[420]: rho = ufloat(fourwire.rho.mean(), fourwire.rho.std())
rho
```

```
[420]: 0.3242430928548563+/-0.05972607723585084
```

$$\rho = \frac{1}{n e \mu}$$
$$\mu = \frac{1}{n e \rho}$$

```
[556]: mu = 1/(n*e*rho) # m^2/(Vs)
mu
```

[556]: 0.008528253321725519+/-0.0016041212535573492

```
[557]: mu * 100*100
```

[557]: 85.28253321725518+/-16.041212535573493

```
[559]: rho * 100, 'cm'
```

[559]: (32.424309285485634+/-5.972607723585084, 'cm')

```
[560]: print(n*1e-6)
```

(2.26+/-0.09)e+15

0.0.4 Summary

```
[425]: # mobility, resistivity, charge carrier density
# mu, rho, n
```

```
[561]: print_unc(rho)
```

0.32 +- 0.06

[561]: (0.32, 0.06, 2)

```
[562]: 44.9-470.5,
```

[562]: (-425.6,)

```
[563]: (n.n+n.s)*1e-6
```

[563]: 2343075971610824.0

```
[429]: summary = pd.DataFrame([
    ['$\rho$', '$0.09 \pm 0.02$ (cm\Omega)$', '$5.86-6.31$ (cm\Omega)$',
    '\cite{resistivity}', '$2\sigma$'],
    # ['$n$', '$2.26e15 \pm 9e13$ (cm$^{-1}$)$', '$2.02-6.31$',
    '\cite{resistivity}', '$2\sigma$'],
    ['$\mu$', '$320 \pm 60$ (\frac{cm}{Vs})$', '$455$ (\frac{cm}{Vs})$',
    '\cite{resistivity}', '$-3\sigma$'],
],
    columns=['Property', 'Measured Value',
             'Accepted Value', 'Refs.', 'Deviation'])
print(summary.to_latex())
```

```
\begin{tabular}{lllll}
\toprule
{} & Property & Measured Value & Accepted Value &
Refs. & Deviation \\
\end{tabular}
```

```

\midrule
0 & \textbackslash rho\textbackslash$ & \textbackslash$0.09 \textbackslash pm 0.02\textbackslash
(cm\textbackslash Omega)\textbackslash$ & \textbackslash$5.86-6.31\textbackslash (cm\textbackslash
Omega)\textbackslash$ & \textbackslash cite\{resistivity\} & \textbackslash$2\textbackslash sigma\textbackslash$ \textbackslash
1 & \textbackslash mu\textbackslash$ & \textbackslash$320 \textbackslash pm 60\textbackslash
(\textbackslash frac\{cm\}\{Vs\})\textbackslash$ & 455\textbackslash (\textbackslash
frac\{cm\}\{Vs\} & \textbackslash cite\{resistivity\} & \textbackslash$-3\textbackslash
sigma\textbackslash$ \textbackslash
\bottomrule
\end{tabular}

```

[]:

[]:

[]: