

Spezifische_Ladung_des_Elektrons.nbconvert

January 25, 2024

1 Fakultät für Physik

1.1 Physikalisches Praktikum P1 für Studierende der Physik

Versuch P1-71, 72, 73 (Stand: Dezember 2022)

[Raum F1-14](#)

2 Spezifische Ladung des Elektrons

Name: *surname1* Vorname: *name1* E-Mail: *mail1*

Name: *surname2* Vorname: *name2* E-Mail: *mail2*

Gruppennummer: *groupnumber*

Betreuer:

Versuch durchgeführt am: *date*

Beanstandungen:

Testiert am: _____ Testat: _____

```
[1]: # Hilfsfunktionen. Nicht weiter beachten
from IPython.core.display import Markdown
from IPython.display import display
def M_helper_func(txt):
    #print(txt)
    #print('fr'+txt.replace("$", "$").replace("{", "{{").replace("}", "}}").
    ↪replace("[", "{").replace("!", "!")+''')
    excreturnvalue = ""
    loc = {}
    exec('excreturnvalue = fr"<div style=\'margin-left: 8px;\'>'+txt.
    ↪replace("{", "{{").replace("}", "}}").replace("[", "{").replace("!", "!")+</
    ↪div>''', globals(), loc)
    display(Markdown(loc["excreturnvalue"]))
    return loc["excreturnvalue"]

from IPython.core.magic import (register_line_magic,
                                register_cell_magic)

from IPython.core import magic

def register_line_magic(f):
    setattr(f, magic.MAGIC_NO_VAR_EXPAND_ATTR, True)
    return magic.register_line_magic(f)

@register_line_magic
def m(line):
    M_helper_func(line)
```

```

import os
import numpy as np
from numpy import pi
import matplotlib.pyplot as plt
import matplotlib as mpl
import scipy as sci
import kafe2
import pandas as pd
import uncertainties
from uncertainties import ufloat, unumpy
from uncertainties.unumpy import uarray as uarr
from uncertainties.unumpy import nominal_values, nominal_values as nomv, u
↳std_devs, std_devs as stdv
from uncertainties.umath import sqrt as usqrt, sin as usin,exp as uexp
import scipy
import matplotlib.image as mpimg
from io import StringIO

mpl.rcParams['figure.dpi'] = 100
class ng:
    def __init__(self,x=None,stat=None,syst=None):
        if x == None:
            self.syst = None
            self.stat = None
        else:
            self.syst = ufloat(x,0)
            self.stat = ufloat(x,0)
            if syst != None:
                self.syst = ufloat(x,syst)
            if stat != None:
                self.stat = ufloat(x,stat)
    def SetSyst(self,syst):
        self.syst = ufloat(nomv(self.syst),syst)
    def SetStat(self,stat):
        self.stat = ufloat(nomv(self.stat),stat)
    def __str__(self):
        return f'{nomv(self.stat):.2}± {stdv(self.stat):.2}± {stdv(self.syst):
↳.2}'
    def __add__(self,o):
        res = ng()
        if type(o) == ng:
            res.syst = self.syst + o.syst
            res.stat = self.stat + o.stat
        else:
            res.syst = self.syst + o
            res.stat = self.stat + o
        return res

```

```

def __iadd__(self,o):
    return self.__add__(o)
def __sub__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst - o.syst
        res.stat = self.stat - o.stat
    else:
        res.syst = self.syst - o
        res.stat = self.stat - o
    return res
def __isub__(self,o):
    return self.__sub__(o)
def __mul__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst * o.syst
        res.stat = self.stat * o.stat
    else:
        res.syst = self.syst * o
        res.stat = self.stat * o
    return res
def __imul__(self,o):
    return self.__mul__(o)
def __truediv__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst / o.syst
        res.stat = self.stat / o.stat
    else:
        res.syst = self.syst / o
        res.stat = self.stat / o
    return res
def __itruediv__(self,o):
    return self.__truediv__(o)
def __pow__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst**o.syst
        res.stat = self.stat**o.stat
    else:
        res.syst = self.syst**o
        res.stat = self.stat**o
    return res
def __ipow__(self,o):
    return self.__pow__(o)
def sin(o):

```

```

    if type(o) == ng:
        res = ng()
        res.syst = usin(o.syst)
        res.stat = usin(o.stat)
        return res
    else:
        return np.sin(o)
def exp(o):
    if type(o) == ng:
        res = ng()
        res.syst = uexp(o.syst)
        res.stat = uexp(o.stat)
        return res
    else:
        return np.exp(o)
def sqrt(o):
    if type(o) == ng:
        res = ng()
        res.syst = usqrt(o.syst)
        res.stat = usqrt(o.stat)
        return res
    else:
        return np.sqrt(o)

def uplot(x,y,*args, **kwargs):
    utypes = (uncertainties.core.
Variable,uncertainties.core.AffineScalarFunc)
    xplt = x
    yplt = y
    if type(x[0]) in utypes:
        xplt = nomv(x)
        kwargs["xerr"] = stdv(x)
    if type(y[0]) in utypes:
        yplt = nomv(y)
        kwargs["yerr"] = stdv(y)
    plt.errorbar(xplt,yplt,*args,**kwargs)

def plting(path):
    try:
        img = mpimg.imread(path)
        imgplot = plt.imshow(img)
        plt.axis('off')
        plt.show()
    except Exception as e:
        print("Image Not Found")
        print(e)

```

3 Durchführung

3.1 Aufgabe 1: Fadenstrahlrohr

Hinweise zu allen hier durchzuführenden Messungen finden Sie in der Datei [Hinweise-Aufgabe-1.md](#).

3.1.1 Aufgabe 1.1: Magnetfeld im Fadenstrahlrohr

- Schätzen Sie das Magnetfeld B , entlang der Mittelebene zwischen H_1 und H_2 im Inneren der Spulen, mit Hilfe einer baugleichen, weiteren Helmholtzspule H_3 , einer, an verschiedenen Stellen mit Bohrungen versehenen, Holzplatte M , und einer [Hall-Sonde](#) ab. Dabei bezeichnet r im Folgenden den Abstand von der Symmetrieachse des Spulenpaares.
- Kalibrieren Sie die Hall-Sonde mit Hilfe einer weiteren, langen Spule, deren Magnetfeld Sie über das [Ampèresche Gesetz](#) bestimmen können.
- Diskutieren Sie die Homogenität von $B(r)$ für die vorliegende Spulenanordnung.
- Vergleichen Sie Ihre Messung für $r = 0$ mit Ihrer Erwartung.

```
[2]: %m In diesem Versuch geht es darum, das B-Feld innerhalb der Helmholtzspulen mit  
↪ einem baugleichen Spulenpaar abzuschätzen. Dazu wird das B-Feld an  
↪ vorgegebenen Positionen mit einer Hall-Sonde gemessen. Die vorgegebene  
↪ Positionen werden folgendermaßen nummeriert:
```

```
plting("Löcher.png")
```

```
a11 = ""
```

```
Messpunkt,Strom I in A,Spannung U in mV
```

```
0,1.008,25.69
```

```
1,1.008,27.37
```

```
2,1.008,28.12
```

```
3,1.008,27.94
```

```
4,1.008,28.50
```

```
5,1.008,28.87
```

```
6,1.008,28.69
```

```
7,1.008,28.5
```

```
8,1.008,29.06
```

```
9,1.008,28.69
```

```
10,1.008,27.56
```

```
11,1.008,24.00
```

```
0,1.504,39.00
```

```
1,1.504,41.44
```

```
2,1.504,42.00
```

```
3,1.504,41.81
```

```
4,1.504,42.56
```

```
5,1.504,42.75
```

```
6,1.504,42.75
```

```
7,1.504,42.38
```

```

8,1.504,42.94
9,1.504,42.56
10,1.504,40.50
11,1.504,34.88
0,2.007,51.56
1,2.007,54.19
2,2.007,55.13
3,2.007,54.19
4,2.007,55.31
5,2.007,55.88
6,2.007,56.06
7,2.007,55.69
8,2.007,56.44
9,2.007,55.88
10,2.007,53.63
11,2.007,46.13
"""#am Ende nochmal Messung des 0 Wertes am Ende zum Vergleich mit vor der
↳Messung 0,2.007,54.19

radiusTranslation = np.array([9,6,3,6,3,0,3,6,3,6,9,12])

%m Die Messung mit der Hallsonde liefert folgende Messergebnisse:
mpg = pd.read_csv(StringIO(a11))
display(mpg)

%m Beim Vergleich vom Anfangsergebnis mit dem erneuten Messen des
↳Anfangspunktes zeigt sich, dass die thermische Abweichung über die Messreihe
↳bei unter 6% liegt, was in diesem Kontext noch vernachlässigbar ist.

n, I, U = np.genfromtxt(StringIO(a11), delimiter=',')[1:].transpose()
U = 1e-3*U
n = np.array(n, dtype=int)
r = radiusTranslation[n]

uU = uarr(U, 0.00019/2)
uI = uarr(I, 0.004)

%m Für die Unsicherheit auf die Spannung wird 0.095mV gewählt, da die
↳Messschrittweite 0.19mV beträgt. Die Unsicherheit auf den Strom beträgt nach
↳der Schwankung des Werts 4mA.

%m Zur Kalibrierung werden 10 Wertepaare von Strom und Hallspannung auch in
↳einer langen Spule gemessen. Diese Messung ergibt folgende Messwerte:
#200mA bis 600mA
a11="""

```

```

Strom I in A,Spannung U in mV
0.2,19.87
0.24,24.37
0.276,28.31
0.316,32.06
0.357,36.19
0.398,39.75
0.439,43.88
0.48,47.63
0.516,51.0
0.562,55.13
0.598,58.31
"""# am Ende bei 0.2: 20.06

mpg = pd.read_csv(StringIO(a11))
display(mpg)

I_cal, U_cal = np.genfromtxt(StringIO(a11), delimiter=',')[1:].transpose()
U_cal = 1e-3 *U_cal
uU_cal = uarr(U_cal, 0.00019/2)
uI_cal = uarr(I_cal, 0.004)

%m Im folgenden werden nun die Parameter einer Eichgeraden durch die für die
↳Kalibrierung gemessenen Punkte bestimmt. Es soll ein Zusammenhang  $B =$ 
↳ $m \cdot (U - c)$  bestimmt werden mit dem freien Parameter m wobei B in der
↳Kalibrierungsmessung über den Strom als  $\mu_0 \frac{N I}{l}$  bestimmt
↳werden kann. c ist hierbei der Spannungsoffset der Hallsonde.

uN = ufloat(750, 7.5)
ul = ufloat(0.3, 2e-3)
u0 = ufloat(1.25663706212, 0.00000000000019)*1e-6

def model(I, m=0.032, N=1, l=1, c=1):
    B = u0*N*I/l
    U = B/m
    return U + c

xy_data = kafe2.XYContainer(x_data=nomv(uI_cal), y_data=nomv(uU_cal))
xy_data.add_error(axis='x', err_val=stdv(uI_cal))
xy_data.add_error(axis='y', err_val=stdv(uU_cal))

lin_fit = kafe2.Fit(data=xy_data, model_function=model)
lin_fit.add_parameter_constraint(name='0', value=u0.n, uncertainty=u0.s)
lin_fit.add_parameter_constraint(name='N', value=uN.n, uncertainty=uN.s)
lin_fit.add_parameter_constraint(name='l', value=ul.n, uncertainty=ul.s)
lin_fit.do_fit()

```

```

p = kafe2.Plot(lin_fit)
p.x_label = 'Kalibrierstrom in A'
p.y_label = 'Kalibrierhallspannung in V'
p.plot()
p.show()

%m Mit den so bestimmten Werten von m und c können nun die vorherigen Messungen
↳in Magnetfeldstärken umgerechnet werden.

%m Werden die Magnetfeldstärken, die im äußeren Helmholtzspulenpaar gemessen
↳werden, über den Abstand vom Mittelpunkt aufgetragen, entsteht folgende
↳Grafik:

m,_,_,c = lin_fit.parameter_values
Δm,_,_,Δc = lin_fit.parameter_errors
uc = ufloat(c,Δc)
um = ufloat(m,Δm)

uB = (uU - uc)*um

args = {"fmt":"x", "capsize":3}
uplot(r[I==1.008], uB[I==1.008],label="1A", **args)
uplot(r[I==1.504], uB[I==1.504], label="1.5A", **args)
uplot(r[I==2.007], uB[I==2.007], label="2A", **args)

plt.legend()
plt.ylim(bottom=0)
plt.ylabel("Magnetfeld in T")
plt.xlabel("Radius in cm")
plt.show()

I_theo = uarr([1.008, 1.504, 2.007], 0.004)
B_theo = 8*u 0*130/(np.sqrt(125)*0.15)*I_theo

```

%m Es ist zu erkennen, dass das Magnetfeld bis zu einem Radius von 6cm im
 ↳Rahmen der Messunsicherheiten konstant bleibt. Der erwartete Wert für diese
 ↳Spulenkonguration ist $[\mu\text{B_theo}[0].n*1e6:.2f!]\mu\text{H} \pm [\mu\text{B_theo}[0].s*1e6:.2f!]\mu\text{H}$ für $I=[\mu\text{I_theo}[0].n:.2f!]$, $[\mu\text{B_theo}[1].n*1e6:.2f!]\mu\text{H} \pm [\mu\text{B_theo}[1].s*1e6:.2f!]\mu\text{H}$ für $I=[\mu\text{I_theo}[1].n:.2f!]$ und $[\mu\text{B_theo}[2].n*1e6:.2f!]\mu\text{H} \pm [\mu\text{B_theo}[2].s*1e6:.2f!]\mu\text{H}$ für $I=[\mu\text{I_theo}[2].n:.2f!]$. Diese Werte sind nicht im Rahmen ihrer Unsicherheit miteinander
 ↳Kompatibel. Folglich entspricht das Spulenpaar nicht ausreichend einem
 ↳idealen Helmholtzpaar oder die Messergebnisse der Hallsonde waren nicht
 ↳ausreichend gut kalibriert. Im folgenden wird mit einem in Abhängigkeit vom
 ↳Radius konstanten Magnetfeld weitergerechnet. Die Magnetfeldstärke wird in
 ↳Abhängigkeit vom Spulenstrom durch eine Gerade approximiert, um für das
 ↳zweite Experiment einen Zusammenhang von Spulenstrom und Magnetfeld für
 ↳beliebige Ströme zu erhalten.

```
def model(I, a, b):
    return I*a+b

xy_data = kafe2.XYContainer(x_data=I[r<=6], y_data=nomv(uB)[r<=6])
xy_data.add_error(axis='y', err_val=stdv(uB[r<=6]))
xy_data.add_error(axis='x', err_val=stdv(uI[r<=6]))

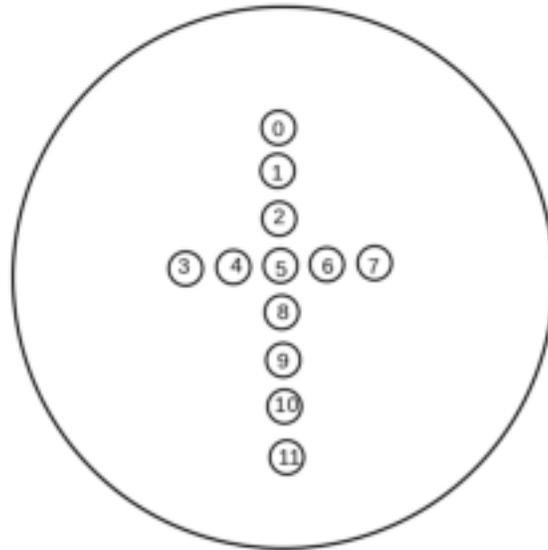
lin_fit = kafe2.Fit(data=xy_data, model_function=model)
lin_fit.do_fit()

p = kafe2.Plot(lin_fit)
p.x_label = 'Strom in A'
p.y_label = 'Magnetfeld in T'
p.plot()
p.show()

a,b = lin_fit.parameter_values
Δa,Δb = lin_fit.parameter_errors
ua = ufloat(a,Δa)
ub = ufloat(b,Δb)

bfunc = lambda i: i*ua+ub
#print(stdv(uB[r<=6]))
```

In diesem Versuch geht es darum, das B-Feld innerhalb der Helmholtzspulen mit einem baugleichen Spulenpaar abzuschätzen. Dazu wird das B-Feld an vorgegebenen Positionen mit einer Hall-Sonde gemessen. Die vorgegebene Positionen werden folgendermaßen nummeriert:



Die Messung mit der Hallsonde liefert folgende Messergebnisse:

	Messpunkt	Strom I in A	Spannung U in mV
0	0	1.008	25.69
1	1	1.008	27.37
2	2	1.008	28.12
3	3	1.008	27.94
4	4	1.008	28.50
5	5	1.008	28.87
6	6	1.008	28.69
7	7	1.008	28.50
8	8	1.008	29.06
9	9	1.008	28.69
10	10	1.008	27.56
11	11	1.008	24.00
12	0	1.504	39.00
13	1	1.504	41.44
14	2	1.504	42.00
15	3	1.504	41.81
16	4	1.504	42.56
17	5	1.504	42.75
18	6	1.504	42.75
19	7	1.504	42.38
20	8	1.504	42.94
21	9	1.504	42.56
22	10	1.504	40.50
23	11	1.504	34.88
24	0	2.007	51.56

25	1	2.007	54.19
26	2	2.007	55.13
27	3	2.007	54.19
28	4	2.007	55.31
29	5	2.007	55.88
30	6	2.007	56.06
31	7	2.007	55.69
32	8	2.007	56.44
33	9	2.007	55.88
34	10	2.007	53.63
35	11	2.007	46.13

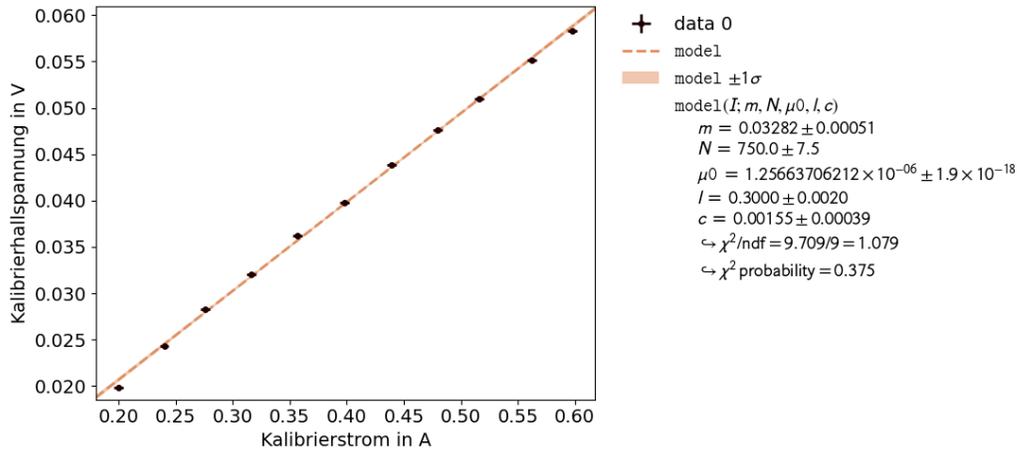
Beim Vergleich vom Anfangsergebnis mit dem erneuten Messen des Anfangspunktes zeigt sich, dass die thermische Abweichung über die Messreihe bei unter 6% liegt, was in diesem Kontext noch vernachlässigbar ist.

Für die Unsicherheit auf die Spannung wird 0.095mV gewählt, da die Messschrittweite 0.19mV beträgt. Die Unsicherheit auf den Strom beträgt nach der Schwankung des Werts 4mA.

Zur Kalibrierung werden 10 Wertepaare von Strom und Hallspannung auch in einer langen Spule gemessen. Diese Messung ergibt folgende Messwerte:

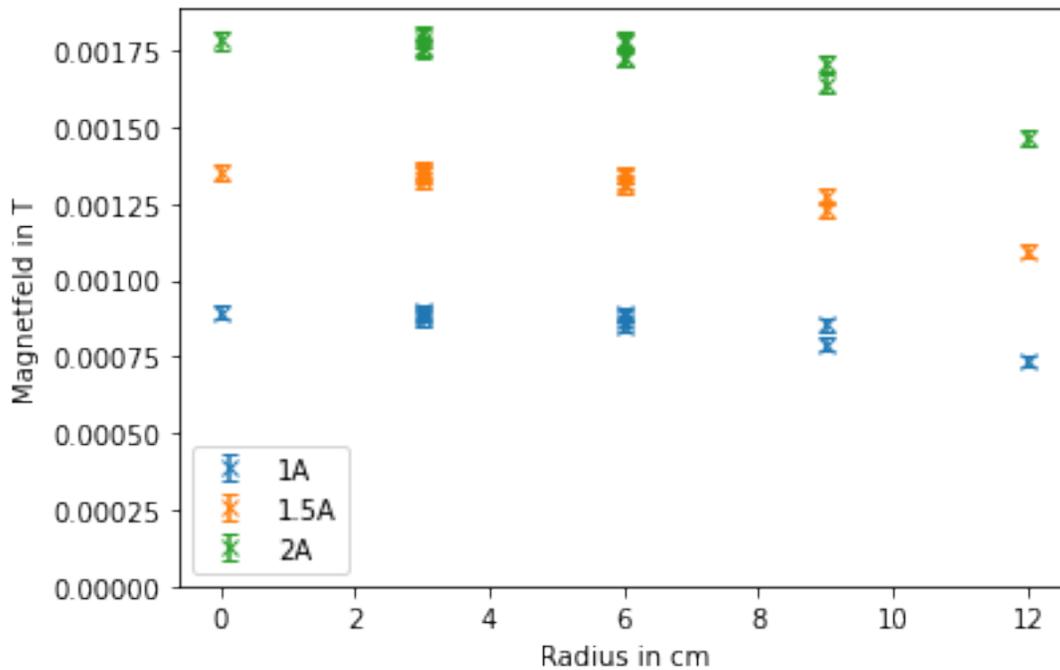
	Strom I in A	Spannung U in mV
0	0.200	19.87
1	0.240	24.37
2	0.276	28.31
3	0.316	32.06
4	0.357	36.19
5	0.398	39.75
6	0.439	43.88
7	0.480	47.63
8	0.516	51.00
9	0.562	55.13
10	0.598	58.31

Im folgenden werden nun die Parameter einer Eichgeraden durch die für die Kalibrierung gemessenen Punkte bestimmt. Es soll ein Zusammenhang $B = m(U - c)$ bestimmt werden mit dem freien Parameter m wobei B in der Kalibrierungsmessung über den Strom als $\mu_0 \frac{NI}{l}$ bestimmt werden kann. c ist hierbei der Spannungsoffset der Hallsonde.



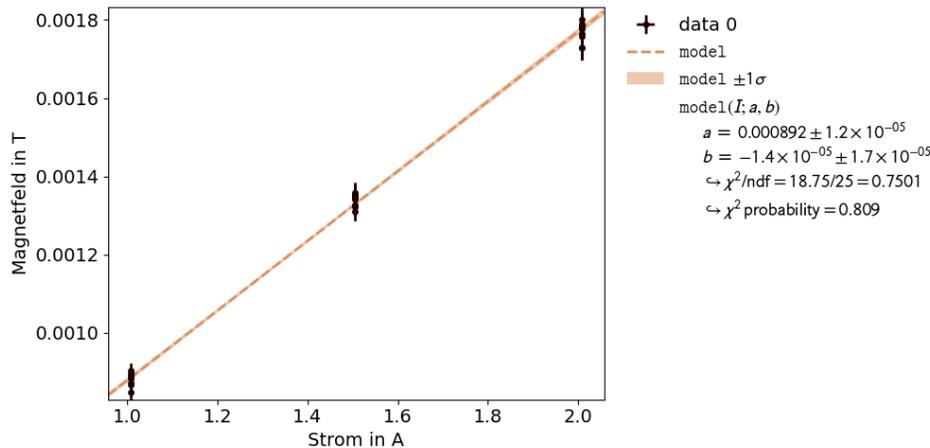
Mit den so bestimmten Werten von m und c können nun die vorherigen Messungen in Magnetfeldstärken umgerechnet werden.

Werden die Magnetfeldstärken, die im äußeren Helmholtzspulenpaar gemessen werden, über den Abstand vom Mittelpunkt aufgetragen, entsteht folgende Grafik:



Es ist zu erkennen, dass das Magnetfeld bis zu einem Radius von 6cm im Rahmen der Messunsicherheiten konstant bleibt. Der erwartete Wert für diese Spulenkonguration ist $785.52\mu H \pm 3.12\mu H$ für $I=1.01$, $1172.05\mu H \pm 3.12\mu H$ für $I=1.50$ und $1564.03\mu H \pm 3.12\mu H$ für $I=2.01$. Diese Werte sind nicht im Rahmen ihrer Unsicherheit miteinander kompatibel. Folglich entspricht das Spulenpaar

nicht ausreichend einem idealen Helmholtzpaar oder die Messergebnisse der Hallsonde waren nicht ausreichend gut kalibriert. Im folgenden wird mit einem in Abhängigkeit vom Radius konstanten Magnetfeld weitergerechnet. Die Magnetfeldstärke wird in Abhängigkeit vom Spulenstrom durch eine Gerade approximiert, um für das zweite Experiment einen Zusammenhang von Spulenstrom und Magnetfeld für beliebige Ströme zu erhalten.



3.1.2 Aufgabe 1.2: Elektronenkreisbahn

Bestimmen Sie den Durchmesser d der Elektronenbahn im Fadenstrahlrohr in zwei Messreihen:

- Als Funktion der Anodenspannung U (z.B. mit fünf bis sechs Messpunkten 100; 120; ... 200 V) für zwei Spulenströme (z.B. 1 A und 2 A).
- Als Funktion des Spulenstroms I (z.B. mit zehn bis elf Messpunkten 1,0; 1,1; ... 2,0 A) für zwei Anodenspannungen (z.B. 140 V und 160 V).

[3]: `%m` In diesem Experiment soll der Radius des auf die Kreisbahn gezwungenen `↳` Elektronenstrahls bestimmt werden und mit Hilfe der Ergebnisse aus den `↳` vorherigen Messungen das Magnetfeld bestimmt werden. Bei der Durchführung `↳` werden folgende Radien bestimmt:

```
a11 = ""
Spannung in V, Strom in A, Durchmesser in cm
130.2, 0.992, 8.1
141.0, 0.996, 8.8
150.4, 0.994, 9.3
160.5, 0.991, 10
170.3, 0.991, 10.9
180.4, 0.977, 11.2
108.7, 2.058, 4
121.4, 2.057, 4.3
141.2, 2.08, 4.9
```

```

160.8, 2.081, 5.1
181, 2.082, 5.6
200.0, 2.078, 5.9
160, 1.017, 10.3
160, 1.106, 9.2
160, 1.208, 8.3
160, 1.303, 7.2
160, 1.404, 7.0
160, 1.497, 6.7
160, 1.606, 6.2
160, 1.698, 5.8
160, 1.806, 5.7
160, 1.900, 5.3
160, 2.006, 5.1
140.9, 2.007, 4.9
140.9, 1.901, 5.2
140.9, 1.804, 5.5
140.9, 1.708, 5.4
140.9, 1.600, 6
140.9, 1.508, 6.1
140.9, 1.405, 6.6
140.9, 1.296, 7
140.9, 1.202, 7.5
140.9, 1.100, 8.1
140.9, 0.992, 9
"""

U, I, r = np.genfromtxt(StringIO(a11), delimiter=',')[1:].transpose()
r = 1e-2*r/2
uU = uarr(U, 0.1)
uI = uarr(I, 0.004)
ur = uarr(r, 0.002)
uB = bfunc(uI)
#uB = 8*u 0*130*uI/(sqrt(125)*0.15)

uem = 2*uU/(uB**2 * ur**2)

mpg = pd.read_csv(StringIO(a11))
mpg[r"Magnetfeldstärke in T"] = nomv(uB)
mpg[r"$\frac{e}{m}$"] = nomv(uem)
mpg[r"Unsicherheit von $\frac{e}{m}$"] = stdv(uem)
display(mpg)

```

%m Aufgetragen über Spannung und Strom ergibt sich folgende Grafik, die klar
↳ den proportionalen Zusammenhang von Spannung und Radius sowie den
↳ antiproportionalen Zusammenhang von Strom und Radius ersichtlich macht.

```

fig = plt.figure(figsize=(12, 12))
ax = fig.add_subplot(projection='3d')
ax.scatter(nomv(uI), nomv(uU), nomv(ur))
np.transpose([nomv(ur), np.zeros(ur.shape)])
for i in range(len(ur)):
    ax.plot([nomv(uI[i]), nomv(uI[i])], [nomv(uU[i]), nomv(uU[i])],
    ↪ [nomv(ur[i]), 0], '--k')
ax.set_zlim(bottom=0)
ax.set_xlabel("Spannung in V")
ax.set_ylabel("Strom in A")
ax.set_zlabel("Kreistradius in m")
plt.show()

%m Die Unsicherheit für die Beschleunigungsspannung beträgt 100mV nach der
↪ Anzeigegenauigkeit des Messgeräts. Die restlichen Unsicherheiten sind
↪ äquivalent zum vorherigen Aufgabenteil.

%m Aus dem in der vorherigen Aufgabe bestimmten Zusammenhang zwischen I und B
↪ kann die Magnetfeldstärke bestimmt werden und damit wiederum die Spezifische
↪ Ladung mit der Formel  $\frac{e}{m_e} = \frac{2 U}{B^2 r^2}$ , wie sie in der
↪ Dokumentation hergeleitet ist. Wir haben uns explizit dafür entschieden,
↪ nicht den theoretischen Wert für das Magnetfeld zu verwenden sondern den
↪ tatsächlich gemessenen. Die entsprechenden Ergebnisse sind in der Tabelle
↪ oben aufgeführt. Mit dem Likelihood-Verfahren wird jetzt ein gewichteter
↪ Mittelwert für die Spezifische Ladung bestimmt.

filter = np.where(1-np.isnan(nomv(ur)))
def model(n, em=1.7):
    return em

%m Messreihe I  $\approx$  1A:
xy_data = kafe2.XYContainer(x_data=U[filter][0:6], y_data=nomv(uem[filter][0:
↪ 6])*1e-11)
xy_data.add_error(axis='y', err_val=stdv(uem[filter][0:6])*1e-11)

lin_fit = kafe2.Fit(data=xy_data, model_function=model)
lin_fit.do_fit()

p = kafe2.Plot(lin_fit)
p.x_label = 'Spannung in V'
p.y_label = r' $\frac{e}{m} \cdot 10^{11}$ '
p.plot()
p.show()

%m Messreihe I  $\approx$  2A:
xy_data = kafe2.XYContainer(x_data=U[filter][6:12], y_data=nomv(uem[filter][6:
↪ 12])*1e-11)

```

```

xy_data.add_error(axis='y', err_val=stdv(uem[filter][6:12])*1e-11)

lin_fit = kafe2.Fit(data=xy_data, model_function=model)
lin_fit.do_fit()

p = kafe2.Plot(lin_fit)
p.x_label = 'Spannung in V'
p.y_label = r'\frac{e}{m}\cdot 10^{\{11\}}$'
p.plot()
p.show()

%m Messreihe U = 160 A:
xy_data = kafe2.XYContainer(x_data=I[filter][12:23], y_data=nomv(uem[filter][12:
↪23])*1e-11)
xy_data.add_error(axis='y', err_val=stdv(uem[filter][12:23])*1e-11)

lin_fit = kafe2.Fit(data=xy_data, model_function=model)
lin_fit.do_fit()

p = kafe2.Plot(lin_fit)
p.x_label = 'Strom in A'
p.y_label = r'\frac{e}{m}\cdot 10^{\{11\}}$'
p.plot()
p.show()

%m Messreihe U = 140.9$:
xy_data = kafe2.XYContainer(x_data=I[filter][23:34], y_data=nomv(uem[filter][23:
↪34])*1e-11)
xy_data.add_error(axis='y', err_val=stdv(uem[filter][23:34])*1e-11)

lin_fit = kafe2.Fit(data=xy_data, model_function=model)
lin_fit.do_fit()

p = kafe2.Plot(lin_fit)
p.x_label = 'Strom in A'
p.y_label = r'\frac{e}{m}\cdot 10^{\{11\}}$'
p.plot()
p.show()

%m Gesamtmessung:
xy_data = kafe2.XYContainer(x_data=np.arange(*uem[filter].shape),
↪y_data=nomv(uem[filter])*1e-11)
xy_data.add_error(axis='y', err_val=stdv(uem[filter])*1e-11)

lin_fit = kafe2.Fit(data=xy_data, model_function=model)
lin_fit.do_fit()

```

```

p = kafe2.Plot(lin_fit)
p.x_label = 'Datenpunkte nebeneinander'
p.y_label = r'\frac{e}{m}\cdot 10^{\{11\}}'
p.plot()
p.show()

em = lin_fit.parameter_values
Δem = lin_fit.parameter_errors
uem_fin = ufloat(em, Δem)

%m Damit ergibt sich  $\frac{e}{m_e} = -([\text{uem\_fin.n:.3f!}]\text{pm } [\text{uem\_fin.s:.3f!}$ 
  ↳]) $\cdot 10^{\{11\}} \frac{C}{kG}$ $.
%m Dieses Ergebnis liegt, der Qualität des Messaufbaus entsprechend, nah an dem
  ↳Literaturwert für  $\frac{e}{m}$ $. Er ist aber nicht im Rahmen der
  ↳Messunsicherheit mit den Messergebnissen kompatibel. Es lässt sich aber an
  ↳der regelmäßigen Struktur der Abweichungen der Messpunkte vom Schätzwert
  ↳erkennen, dass es systematische Fehler gibt, für die nicht korrigiert wurde.
  ↳Beispielsweise wäre die Inhomogenität des Magnetfeldes hier eine mögliche
  ↳systematische Abweichung.

```

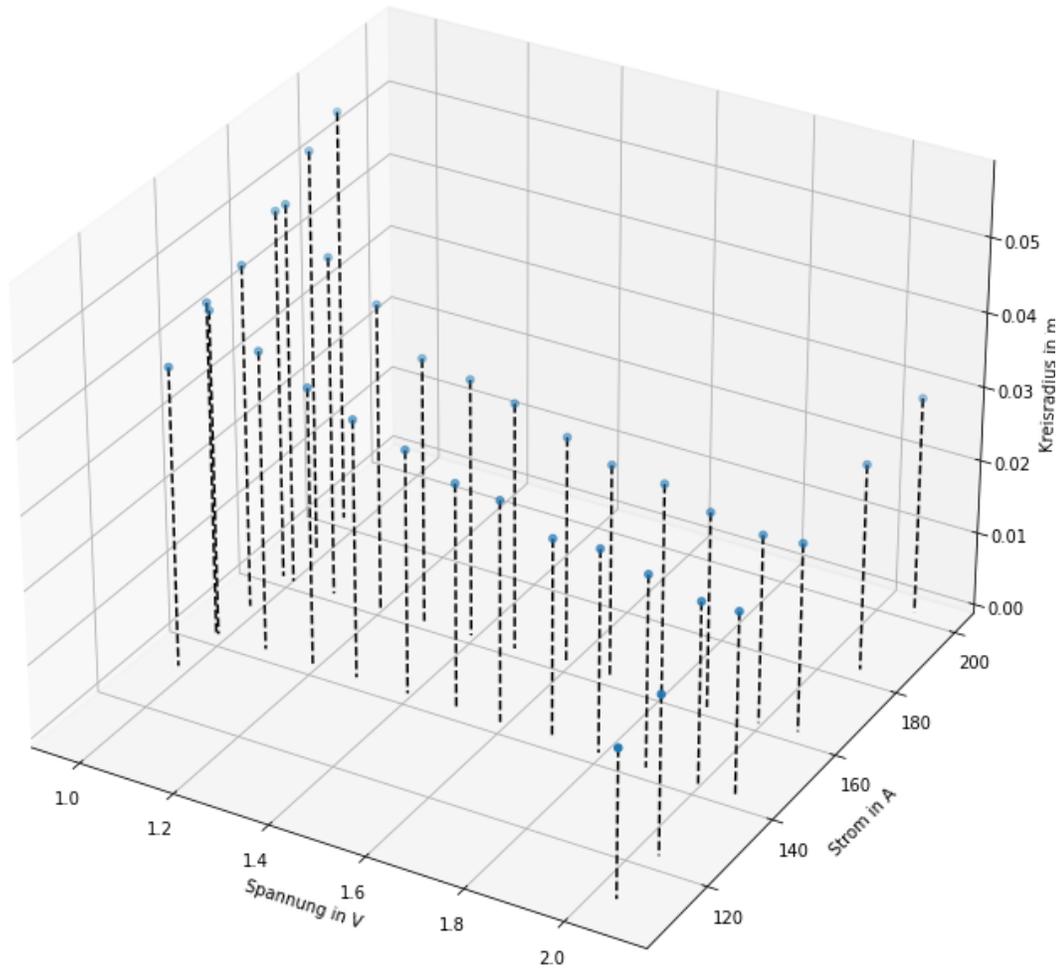
In diesem Experiment soll der Radius des auf die Kreisbahn gezwungenen Elektronenstrahls bestimmt werden und mit Hilfe der Ergebnisse aus den vorherigen Messungen das Magnetfeld bestimmt werden. Bei der Durchführung werden folgende Radien bestimmt:

	Spannung in V	Strom in A	Durchmesser in cm	Magnetfeldstärke in T \
0	130.2	0.992	8.1	0.000871
1	141.0	0.996	8.8	0.000874
2	150.4	0.994	9.3	0.000873
3	160.5	0.991	10.0	0.000870
4	170.3	0.991	10.9	0.000870
5	180.4	0.977	11.2	0.000857
6	108.7	2.058	4.0	0.001822
7	121.4	2.057	4.3	0.001821
8	141.2	2.080	4.9	0.001841
9	160.8	2.081	5.1	0.001842
10	181.0	2.082	5.6	0.001843
11	200.0	2.078	5.9	0.001839
12	160.0	1.017	10.3	0.000893
13	160.0	1.106	9.2	0.000972
14	160.0	1.208	8.3	0.001063
15	160.0	1.303	7.2	0.001148
16	160.0	1.404	7.0	0.001238
17	160.0	1.497	6.7	0.001321
18	160.0	1.606	6.2	0.001418
19	160.0	1.698	5.8	0.001501
20	160.0	1.806	5.7	0.001597
21	160.0	1.900	5.3	0.001681
22	160.0	2.006	5.1	0.001775

23	140.9	2.007	4.9	0.001776
24	140.9	1.901	5.2	0.001682
25	140.9	1.804	5.5	0.001595
26	140.9	1.708	5.4	0.001509
27	140.9	1.600	6.0	0.001413
28	140.9	1.508	6.1	0.001331
29	140.9	1.405	6.6	0.001239
30	140.9	1.296	7.0	0.001142
31	140.9	1.202	7.5	0.001058
32	140.9	1.100	8.1	0.000967
33	140.9	0.992	9.0	0.000871

	$\frac{e}{m}$	Unsicherheit von $\frac{e}{m}$
0	2.093769e+11	2.291284e+10
1	1.905416e+11	1.949965e+10
2	1.827225e+11	1.791651e+10
3	1.696887e+11	1.575822e+10
4	1.515443e+11	1.322062e+10
5	1.565086e+11	1.343663e+10
6	1.637897e+11	3.318339e+10
7	1.584471e+11	2.992060e+10
8	1.387754e+11	2.309415e+10
9	1.457455e+11	2.333907e+10
10	1.359350e+11	1.990662e+10
11	1.358432e+11	1.893244e+10
12	1.512758e+11	1.368067e+10
13	1.599186e+11	1.557132e+10
14	1.642986e+11	1.723951e+10
15	1.872975e+11	2.209812e+10
16	1.703687e+11	2.052422e+10
17	1.633479e+11	2.041326e+10
18	1.655025e+11	2.215601e+10
19	1.689976e+11	2.404041e+10
20	1.545040e+11	2.230914e+10
21	1.613196e+11	2.493437e+10
22	1.561563e+11	2.501903e+10
23	1.488204e+11	2.477720e+10
24	1.474220e+11	2.320423e+10
25	1.464623e+11	2.187583e+10
26	1.696649e+11	2.581632e+10
27	1.568050e+11	2.164501e+10
28	1.709889e+11	2.328841e+10
29	1.685247e+11	2.141196e+10
30	1.764121e+11	2.134914e+10
31	1.789949e+11	2.049311e+10
32	1.836915e+11	1.985890e+10
33	1.835328e+11	1.846510e+10

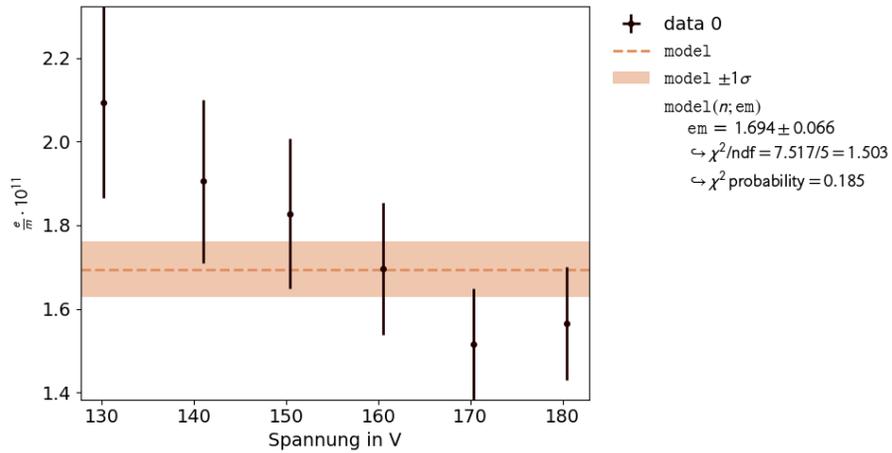
Aufgetragen über Spannung und Strom ergibt sich folgende Grafik, die klar den proportionalen Zusammenhang von Spannung und Radius sowie den antiproportionalen Zusammenhang von Strom und Radius ersichtlich macht.



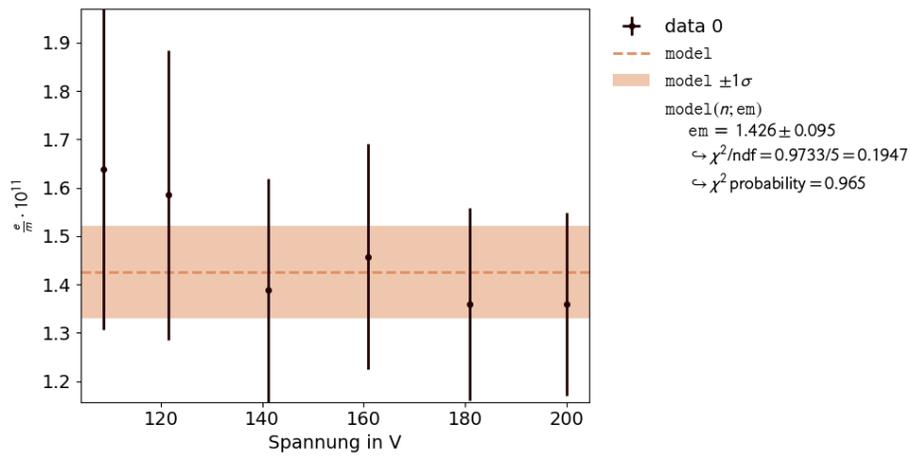
Die Unsicherheit für die Beschleunigungsspannung beträgt 100mV nach der Anzeigenauigkeit des Messgeräts. Die restlichen Unsicherheiten sind äquivalent zum vorherigen Aufgabenteil.

Aus dem in der vorherigen Aufgabe bestimmten Zusammenhang zwischen I und B kann die Magnetfeldstärke bestimmt werden und damit wiederum die Spezifische Ladung mit der Formel $\frac{e}{m_e} = \frac{2U}{B^2 r^2}$, wie sie in der Dokumentation hergeleitet ist. Wir haben uns explizit dafür entschieden, nicht den theoretischen Wert für das Magnetfeld zu verwenden sondern den tatsächlich gemessenen. Die entsprechenden Ergebnisse sind in der Tabelle oben aufgeführt. Mit dem Likelihood-Verfahren wird jetzt ein gewichteter Mittelwert für die Spezifische Ladung bestimmt.

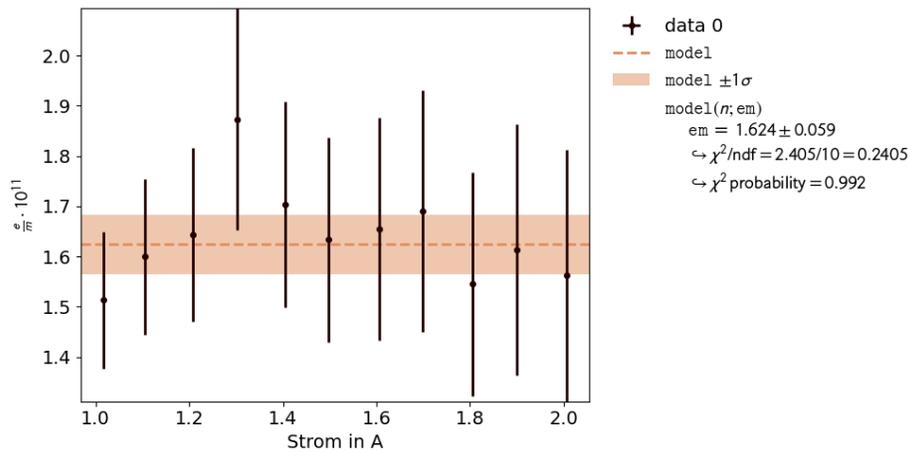
Messreihe I \approx 1A:



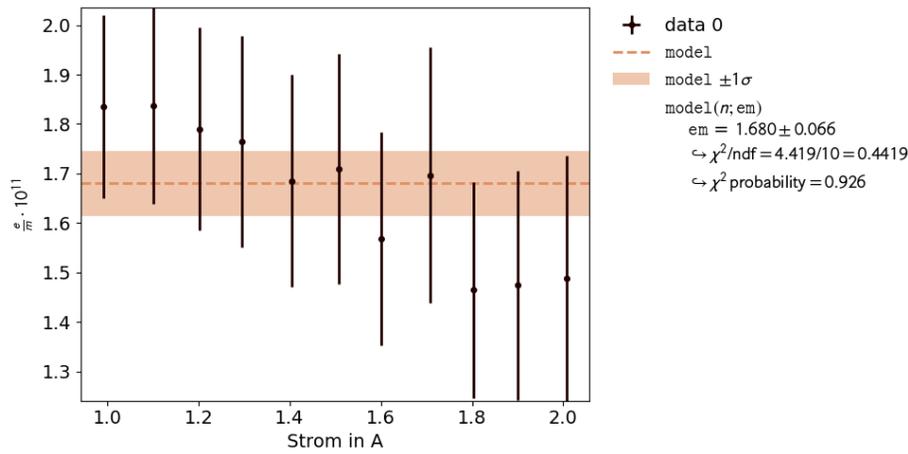
Messreihe I $\approx 2A$:



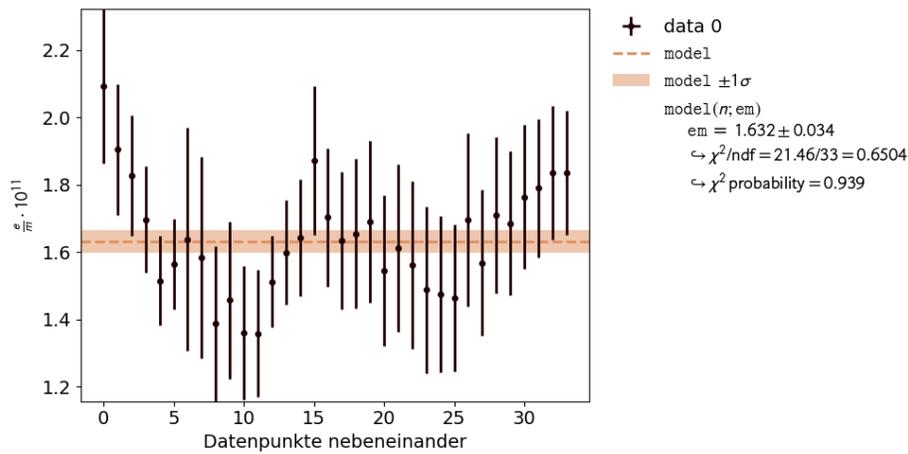
Messreihe U = 160 A:



Messreihe U = 140.9\$:



Gesamtmessung:



Damit ergibt sich $\frac{e}{m_e} = -(1.632 \pm 0.034) \cdot 10^{11} \frac{C}{kG}$.

Dieses Ergebnis liegt, der Qualität des Messaufbaus entsprechend, nah an dem Literaturwert für $\frac{e}{m}$. Er ist aber nicht im Rahmen der Messunsicherheit mit den Messergebnissen kompatibel. Es lässt sich aber an der regelmäßigen Struktur der Abweichungen der Messpunkte vom Schätzwert erkennen, dass es systematische Fehler gibt, für die nicht korrigiert wurde. Beispielsweise wäre die Inhomogenität des Magnetfeldes hier eine mögliche systematische Abweichung.

3.2 Aufgabe 2: Methode von Busch

Hinweise zu allen hier durchzuführenden Messungen finden Sie in der Datei [Hinweise-Aufgabe-2.md](#).

3.2.1 Aufgabe 2.1: Vorbereitung der Messung

Machen Sie sich mit der Methode zur Bestimmung der von e/m_e nach der Methode von Busch vertraut. Verändern Sie hierzu bei vorgegebener Beschleunigungsspannung U_z den Spulenstrom, und erklären Sie Ihre Beobachtungen mit eigenen Worten.

```
[4]: %m Idee der Methode von Busch:
%m Zunächst werden die aus einem Glühdraht austretende Elektronen linear in
↳eine Richtung beschleunigt (Beschleunigungsspannung über das rote Rädchen
↳einstellbar). Nachdem dieser Strahl mit dem blauen Rädchen fokussiert und
↳seine Helligkeit mit dem gelben Rad eingestellt wurde, wird er durch zwei
↳y-Ablenkplatten gelenkt, zwischen denen eine Wechselspannung anliegt. Diese
↳führt dazu, dass die Elektronen von ihrer geradlinigen Bahn abgelenkt werden
↳und als Strich auf dem Schirm erscheinen (da die Spannung zwischen  $-U_0$ 
↳und  $U_0$  variiert, variiert auch das Auftreffen einzelner Elektronen auf
↳dem Schirm vom Auftreffpunkt bei einer Spannung vom Minimalwert bis zum
↳Maximalwert). Zusätzlich wird nun ein B-Feld in Richtung des unabgelenkten
↳Elektronenstrahls angelegt. Dieses sorgt dafür, dass eine Ablenkung durch
↳die angelegte Spannung an den Platten zu einer Ablenkung der Elektronen auf
↳eine Helixbahn führt. Für einen festen Wert von U lässt sich dann das B-Feld
↳(über den Spulenstrom) so einstellen, dass das Auftreffen auf dem Schirm mit
↳dem vollenden einer ganzzahligen Anzahl an Kreisdurchläufen zusammenfällt,
↳also nur noch ein Punkt in der Ruhelage zu beobachten ist.
%m Beobachtungen:
%m Beim Einstellen des Stroms rollt sich der Strich scheinbar zusammen,
↳zwischendurch wird er S-förmig. Dies liegt daran, dass bei unpassenden
↳Verhältnissen nicht alle Elektronen einen vollen Kreis während der Helixbahn
↳durchlaufen und so an verschiedenen Stellen auftreffen, diese Punkte aber
↳immer mehr zusammenliegen.
```

Idee der Methode von Busch:

Zunächst werden die aus einem Glühdraht austretende Elektronen linear in eine Richtung beschleunigt (Beschleunigungsspannung über das rote Rädchen einstellbar). Nachdem dieser Strahl mit dem blauen Rädchen fokussiert und seine Helligkeit mit dem gelben Rad eingestellt wurde, wird er durch zwei y-Ablenkplatten gelenkt, zwischen denen eine Wechselspannung anliegt. Diese führt dazu, dass die Elektronen von ihrer geradlinigen Bahn abgelenkt werden und als Strich auf dem Schirm erscheinen (da die Spannung zwischen $-U_0$ und U_0 variiert, variiert auch das Auftreffen einzelner Elektronen auf dem Schirm vom Auftreffpunkt bei einer Spannung vom Minimalwert bis zum Maximalwert). Zusätzlich wird nun ein B-Feld in Richtung des unabgelenkten Elektronenstrahls angelegt. Dieses sorgt dafür, dass eine Ablenkung durch die angelegte Spannung an den Platten zu einer Ablenkung der Elektronen auf eine Helixbahn führt. Für einen festen Wert von U lässt sich dann das B-Feld (über den Spulenstrom) so einstellen, dass das Auftreffen auf dem

Schirm mit dem vollenden einer ganzzahligen Anzahl an Kreisdurchläufen zusammenfällt, also nur noch ein Punkt in der Ruhelage zu beobachten ist.

Beobachtungen:

Beim Einstellen des Stroms rollt sich der Strich scheinbar zusammen, zwischendurch wird er S-förmig. Dies liegt daran, dass bei unpassenden Verhältnissen nicht alle Elektronen einen vollen Kreis während der Helixbahn durchlaufen und so an verschiedenen Stellen auftreffen, diese Punkte abber immer mehr zusammenliegen.

3.2.2 Aufgabe 2.2: Bestimmung von e/m_e

Messen Sie für Beschleunigungsspannungen von $U = 200 \dots 450$ V (in Schritten von 25 V) den nötigen Spulenstrom I , um auf dem Schirm einen Signalpunkt zu erzeugen. Gehen Sie dabei, für jeden Messpunkt so, wie in Aufgabe 2.1 vor. Tragen Sie U geeignet über I^2 auf und ermitteln Sie daraus e/m_e .

```
[5]: #d2 verwendet 7cm
###Datenblattangaben
L = ufloat(200e-3,0.5e-3)
R = ufloat(45e-3,0.5e-3)
N = 3000
dS = ufloat(150e-3,0)
d2 = dS-ufloat(70e-3,1e-3)
mu_0 = ufloat(1.25663706212e-6,1.5e-10) # nach https://en.wikipedia.org/wiki/
↳Vacuum_permeability, abgerufen am 12.12.23 13:30
K = 0.567/
↳(dS-d2)*(usqrt(R**2+dS**2)-usqrt(R**2+d2**2))+usqrt(R**2+(L-d2)**2)-usqrt(R**2+(L-dS)**2))
B_02 = K * N*mu_0/L

###Messungen
file = "A2.csv"
mpg = pd.read_csv(file)
mpg.rename(columns={'U': 'Spannung in V', 'Ia': 'unterer nötiger Strom in
↳mA', 'Ib': 'oberer nötiger Strom in mA'}, inplace=True)
U,Ia,Ib = np.genfromtxt(file, delimiter=',')[1:].transpose()
I = np.array([1e-3*0.5*(Ib[i]+Ia[i]) for i in range(len(Ia))])
ΔI = [1e-3*0.5*(Ib[i]-Ia[i]) for i in range(len(Ia))]
I_square=[i**2 for i in I]
ΔI_square=[i**2 for i in ΔI]
%m Vermessung des nötigen Spulenstroms zum Erzeugen eines Signalpunktes auf dem
↳Schirm für gegebene Beschleunigungsspannungen (Es wurden jeweils die oberen
↳und unteren Stromgrenzen angegeben, für die ein Punkt erkennbar war):
display(mpg)
ΔU = 1 #Abschätzen

#Fitten
```

```

%m Das erzeugte Magnetfeld wird durch einen geometrischen Faktor genähert, der
↳ die nicht vernachlässigbare Länge der Spule berücksichtigen soll. Dieser ist
↳  $K = \frac{0.567 \{d_S - d_2\} \int_{d_2}^{d_S} \left( \frac{a}{\sqrt{R^2 + a^2}} + \frac{L-a}{R^2 + (L-a)^2} \right) da}{0.567 \{d_S - d_2\} (\sqrt{R^2 + d_S^2} - \sqrt{R^2 + d_2^2} + \sqrt{R^2 + (L-d_2)^2} - \sqrt{R^2 + (L-d_S)^2})}$ 
↳  $K$  mit  $B_0 = K \frac{N \mu_0}{L}$  und damit das Magnetfeld als  $B = B_0 I$  ergibt.

%m An die Daten wird nun ein Modell der Form  $I^2 = -8 \pi^2 \frac{U}{B^2 L^2} \frac{e}{m}$  gefittet.

%m Dafür werden für den Strom jeweils die Hälfte des Intervalls und für die
↳ Spannung ein Fehler von  $\Delta U$ ,  $V$  angenommen. Als gemessener Stromwert
↳ wird der Mittelwert der Intervalle verwendet. Der Fehler auf  $B_0$  ergab sich
↳ aus den Fehlern der angegebenen Längen und verwendeten Konstanten mit ihren
↳ aktuellen Unsicherheiten, der auf die Länge ebenfalls aus den im Datenblatt
↳ verwendeten.

def BModeld(U, k=-1.758e11, B_0=nomv(B_02), l=nomv(dS-d2)):
    return (-8*np.pi**2*(B_0)**(-2)*U*l**(-2)*1/k)

xy_data = kafe2.XYContainer(x_data=U, y_data=I_square)
line_fit = kafe2.Fit(data=xy_data, model_function=BModeld)

line_fit.add_error("x", ΔU)
line_fit.add_error("y", ΔI)
line_fit.add_parameter_constraint(name='l', value=nomv(dS-d2),
↳ uncertainty=stdv(dS-d2))
line_fit.add_parameter_constraint(name='B_0', value=nomv(B_02),
↳ uncertainty=stdv(B_02))
line_fit.do_fit()
p = kafe2.
↳ plot(line_fit, parameter_names=dict(k=r"\frac{e}{m}"), x_label='angelegte
↳ Spannung in V', y_label = r"nötiger Strom in  $\{mA\}^2$ ")
p.show()

###Bewertung
#Literaturwert  $e/m = -1.75882001076e11 \pm 0.53 \text{ C/kg}$  von https://en.wikipedia.org/wiki/Mass-to-charge\_ratio
↳ aufgerufen am 14.12.2023

%m Die  $\chi^2$ -Wahrscheinlichkeit ist mit 1 zu gut, besonders da
↳  $\frac{\chi^2}{\text{ndof}}$  so gering ist. Auffällig ist ebenfalls, dass die
↳ kleineren Werte stärker von der Gerade abweichen als die späteren. Um dies
↳ zu weiter zu untersuchen, wären weitere Messungen bei geringeren Spannungen
↳ sehr hilfreich und vielleicht eine Markierung des erhofften Punktes auf der
↳ Linse hilfreich, um die Grenzen jeweils gleichmäßiger messen zu können.

```

$\frac{e}{m} = (-1.75882001076 \times 10^{-11}) \text{ pm} \cdot \text{kg}^{-1}$ damit immer noch im $2\text{-}\sigma$ -Intervall, die Messungen sind also gut kompatibel.

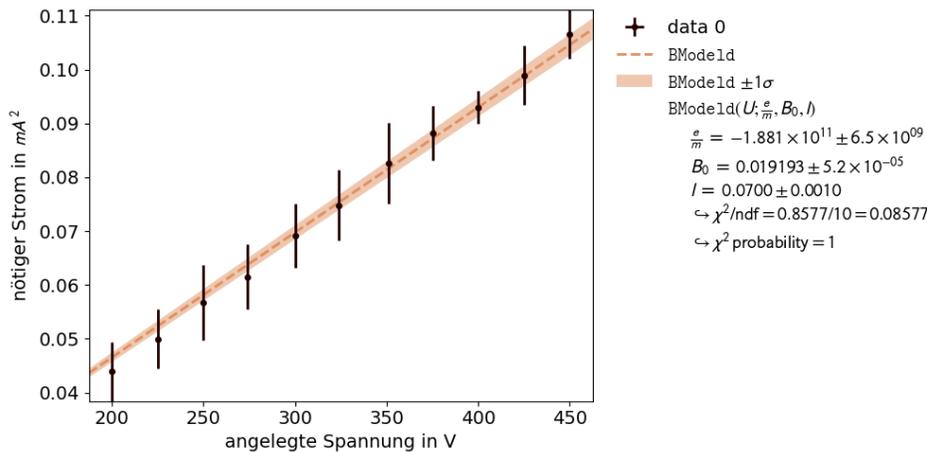
Vermessung des nötigen Spulenstroms zum Erzeugen eines Signalpunktes auf dem Schirm für gegebene Beschleunigungsspannungen (Es wurden jeweils die oberen und unteren Stromgrenzen angegeben, für die ein Punkt erkennbar war):

	Spannung in V	unterer nötiger Strom in mA	oberer nötiger Strom in mA
0	200	204	215
1	225	218	229
2	250	231	245
3	274	242	254
4	300	257	269
5	324	267	280
6	351	280	295
7	375	292	302
8	400	302	308
9	425	309	320
10	450	322	331

Das erzeugte Magnetfeld wird durch einen geometrischen Faktor genähert, der die nicht vernachlässigbare Länge der Spule berücksichtigen soll. Dieser ist $K = \frac{0.567}{d_S - d_2} \int_{d_2}^{d_S} \left(\frac{a}{\sqrt{R^2 + a^2}} + \frac{L-a}{R^2 + (L-a)^2} \right) da = \frac{0.567}{d_S - d_2} \left(\sqrt{R^2 + d_S^2} - \sqrt{R^2 + d_2^2} + \sqrt{R^2 + (L - d_2)^2} - \sqrt{R^2 + (L - d_S)^2} \right) = (1.02 \pm 0.00225)$, womit sich $B_0 = K \frac{N \mu_0}{L} = (0.019 \pm 5.2e - 05) \frac{\text{kg}}{\text{s}^2}$ und damit das Magnetfeld als $B = B_0 * I$ ergibt.

An die Daten wird nun ein Modell der Form $I^2 = -8\pi^2 \frac{U}{B^2 L^2} \frac{e}{m}$ gefittet.

Dafür werden für den Strom jeweils die Hälfte des Intervalls und für die Spannung ein Fehler von 1 V angenommen. Als gemessener Stromwert wird der Mittelwert der Intervalle verwendet. Der Fehler auf B_0 ergab sich aus den Fehlern der angegebenen Längen und verwendeten Konstanten mit ihren aktuellen Unsicherheiten, der auf die Länge ebenfalls aus den im Datenblatt verwendeten.



Die χ^2 -Wahrscheinlichkeit ist mit 1 zu gut, besonders da $\frac{\chi^2}{ndof}$ so gering ist. Auffällig ist ebenfalls, dass die kleineren Werte stärker von der Gerade abweichen als die späteren. Um dies zu weiter zu untersuchen, wären weitere Messungen bei geringeren Spannungen sehr hilfreich und vielleicht eine Markierung des erhofften Punktes auf der Linse hilfreich, um die Grenzen jeweils gleichmäßiger messen zu können.

Trotzdem liegt der Literaturwert von $\frac{e}{m} = (-1.75882001076 * 10^{11} \pm 0.53) \frac{C}{kg}$ damit immer noch im 2- σ -Intervall, die Messungen sind also gut kompatibel.