Physikalisches Anfängerpraktikum II für Lehramtsstudierende

Name: Marius Pfeiffer Matrikel-Nr.: 4188573
E-Mail: marius.pfeiffer@stud.uni-heidelberg.de

Betreut durch: Anselm de Jonge 29.10.2024

Versuch 243: Bestimmung der Boltzmannkonstante mit Hilfe von thermischem Rauschen

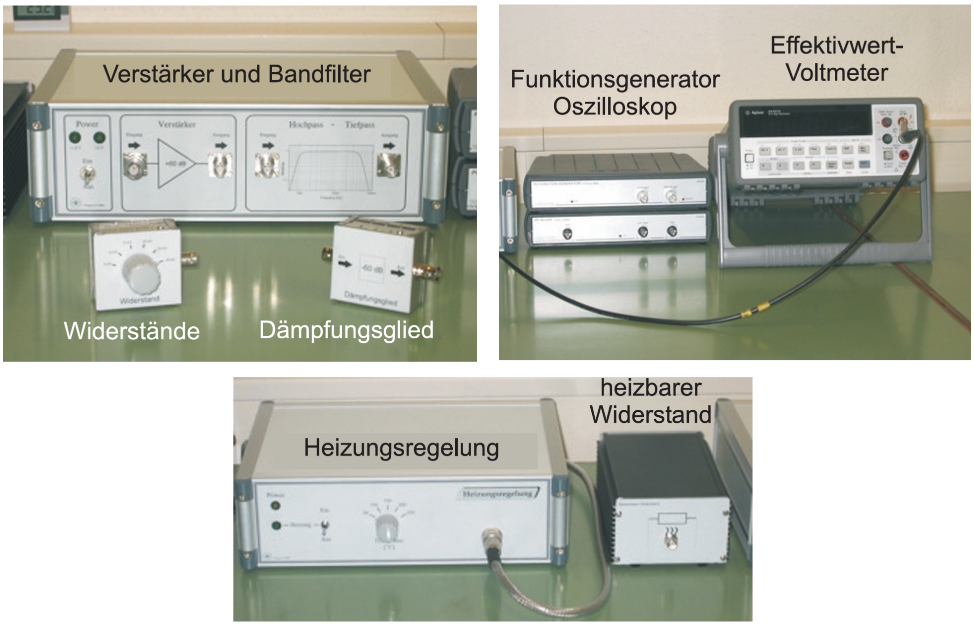


Abbildung 1: Versuchsaufbau

Inhaltsverzeichnis

1 Einleitung

Ziel des Versuchs 243 ist es, das thermische Rauschen eines elektrischen Widerstandes auszunutzen, um die Boltzmannkonstante zu bestimmen.

1.1 Physikalische Grundlagen

Sobald die Temperatur eines elektrischen Leiters über 0K ist, führen die Ladungsträger, auch ohne das Anliegen einer äußeren Spannung, zufällige Zick-Zack-Bewegungen aus. Diese Bewegungen führen zu einer Ladungstrennung und somit einer Rauschspannung, welche statistisch um einen Mittelwert $\langle U_r \rangle$ schwankt. Ohne anliegende Spannung ist die Position der Ladungsträger im Leiter gleichverteilt, es gilt also im zeitlichen Mittel

$$\langle U_r \rangle = \lim_{t' \to \infty} \frac{1}{t'} \int_0^{t'} U_r(t)\,\mathrm{d}t = 0. \tag{1}$$

Der Mittelwert verschwindet somit. Daher wird zur Berechnung der Boltzmannkonstante auf den Effektivwert oder auch Root Mean Square der Rauschspannung

$$\sqrt{\langle U_r^2 \rangle} = \sqrt{\lim_{t' \to \infty} \frac{1}{t'} \int_0^{t'} U_r^2(t)\,\mathrm{d}t}. \tag{2}$$

zurückgegriffen. Dieser steht nach der *Nyquist-Beziehung*

$$\langle U_r^2 \rangle = 4kTR\Delta f \tag{3}$$

in direkter Relation zur Boltzmannkonstante k und hängt dazu lediglich von der Temperatur T, dem Widerstand R des Leiters und der Bandbreite Δf der Messelektronik ab.

Da bei der Messung ein Verstärker verwendet wird, welcher selbst wiederum eine Rauschquelle darstellt, setzt sich der gemessene Effektivwert aus dem Rauschen des Widerstands $\langle U_R^2 \rangle$ und des Verstärkers $\langle U_V^2 \rangle$, welcher im Nachhinein abgezogen werden muss, zusammen.

Im späteren Verlauf des Versuchs wird die für Δf die Rauschbandbreite B des Messsystems eingesetzt. Die Rauschbandbreite ist definiert durch das Integral

$$B = \int_0^{\infty} g(f)^2\,\mathrm{d}f \tag{4}$$

über das Quadrat des Frequenzgangs $g(f)$. Dieser ergibt sich wiederum aus dem Verhältnis der Ein- und Ausgangsspannung bei einer bestimmten Frequenz, nach

$$g(f) = \left.\frac{U_\mathrm{aus}}{U_\mathrm{ein}}\right|_f . \tag{5}$$

1.2 Versuchsdurchführung

Die Versuchsdurchführung setzte sich aus drei Teilschritten zusammen.

Versuchtteil 1 / Vorversuch. Im Vorversuch untersuchten wir zunächst qualitativ die Rauschspannung eines ohm'schen Widerstandes. Hierzu schlossen wir den Widerstand direkt an den Verstärker und den Verstärker ohne einen zusätzlichen Filter an das Oszilloskop an. In der Ausgabe des Oszilloskops war eine gleiche Verteilung aller Frequenzen zu beobachten. Nach Zuschalten des Bandfilters zeichnete sich im Frequenzspektrum ein Abfall bei den höheren Frequenzen ab.

Versuchsteil 2. Im zweiten Versuchsteil ging es um die Bestimmung des Effektivwertes der Rauschspannung. Der zugehörige Versuchsaufbau ist in Abbildung 2 dargestellt. Für die Widerstandswerte von 5 bis 30 kΩ führten wir jeweils eine Messreihe von 110 Punkten durch, um den Mittelwert, sowie die Standardabweichung und den fehler des Mittelwerts zu bestimmen.

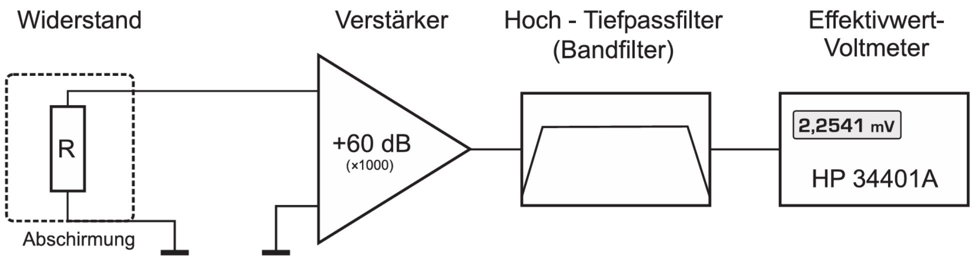


Abbildung 2: Aufbau Versuchsteil 2

Die Aufgenommenen Werte sind dem Messprotokoll zu entnehmen.

Versuchsteil 3. Der letzte Versuchsteil umfasste die Bestimmung des Frequenzgangs des Verstärkers und des Bandfilters. Um kontrolle über das Eingangssignal zu haben, verwendeten wir in diesem Versuchsteil einen Funktionsgenerator als Spannungsquelle. Dieser lieferte ein Sinussignal mit einstellbarer Frequenz und Amplitude. Abbilung 3 zeigt den Aufbau dieses Versuchsteils.

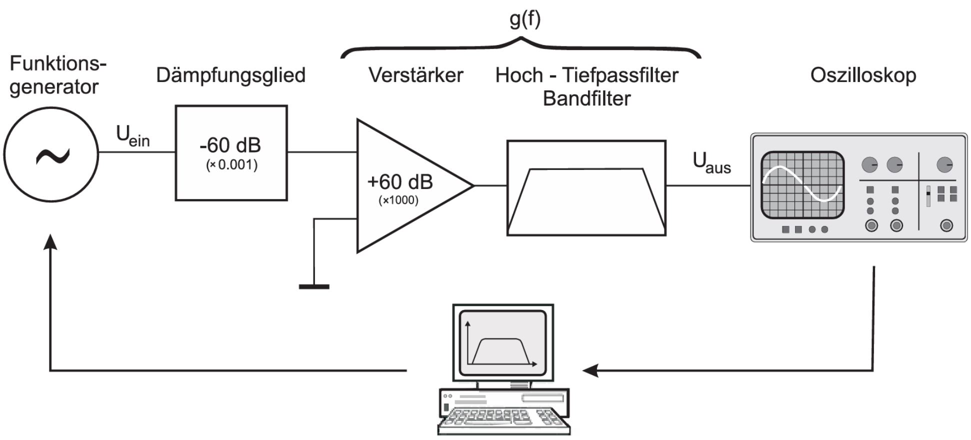


Abbildung 3: Aufbau Versuchsteil 3

Die Messungen dieses Versuchsteils wurden durch die Oszilloskopsoftware automatisiert durchgeführt. Diese zeichnete für eine Konstante Eingangsspannung $\sqrt{\langle U_{\text{ein}}^2\rangle}$ bei kontinuierlich steigender Frequenz f die Augangsspannung $\sqrt{\langle U_{\text{aus}}^2\rangle}$ auf.

2 Messprotokoll

Messprotokoll 243 Marius Pfeiffer, Robert Grosch 29.10.2024

Aufgabe 1

Umschalten Frequenzbereich 0.6 MHz → 3 MHz:
- → kein sichtbarer Abfall bei höheren Frequenzen
- → Mögliche Ursache bei Abfall:
  - ↳ Nicht mehr im Frequenzbereich des Gerätes
  - ↳ Frequenzen fallen nach oben hin ab

Nach zuschalten des Bandfilters
- → Starker Abfall bei höheren Frequenzen sichtbar

Aufgabe 2

Widerstand [kΩ]	#	Mittelwert [mV]	Stabw [mV]	F. d. Mw. [mV]
5 kΩ	110	2,3337	0,012	0.0011
10 kΩ	110	3,0061	0,0111	0.0011
15 kΩ	110	3,5534	0,0124	0,0012
20 kΩ	110	4,0292	0.0145	0,0014
25 kΩ	110	4,4614	0.0162	0,0015
30 kΩ	110	4,8555	0.0182	0,0017
0 kΩ	110	1,3368	0.0055	0.0005

Fehler Widerstände: > 0.5%
Messgenauigkeit Voltmeter: 0.3%

Zimmertemperatur (23,0 ± 0.1)°C

3 Auswertung

3.1 Bestimmung der Rauschbandbreite

Im ersten Teil der Auswertung möchten wir die in Aufgabenteil 3 aufgenommenen Daten verwenden, um die Rauschbandbreite B zu bestimmen. Wie in der Einleitung eingeführt, ist diese Definiert durch das Integral

$$B = \int_0^\infty g(f)^2 \, \mathrm{d}f \, . \tag{6}$$

Aus den aufgenommenen Daten zeigt sich ein funktionaler Verlauf des Frequenzgangs, wie in Abbildung 4. Ziel ist es nun, an diesen Verlauf eine Funktion zu fitten, welche wir im Anschluss numerisch Integrieren können, um B zu bestimmen.

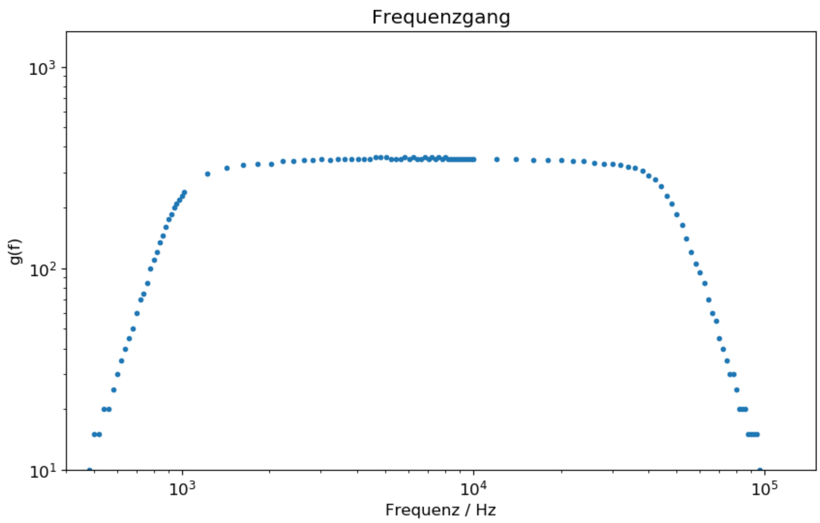


Abbildung 4: Frequenzgang $g(f)$ bei konstanter Eingangsspannung und kontinuierlicher Frequenz

Wir auch dem angefügten Python-Skript zu entnehmen, betrachten wir einen bereinigten Bereich des Frequenzgangs, ohne ein- und auslaufende Werte.

Die zu fittende Funktion ist gegeben durch

$$g(f) = \frac{V}{\sqrt{1 + 1/(f/\Omega_1)^{2n_1}}\sqrt{1 + (f/\Omega_2)^{2n_2}}} . \tag{7}$$

Diese beschreibt den Frequenzgang einer Hintereinanderschaltung eines Verstärkers, eines Hochpassfilters der Ordnung n_1 mit Grenzfrequenz Ω_1, sowie eines Tiefpassfilters der Ordnung n_2 mit Grenzfrequenz Ω_2. Als Startparameter für den Fit wählen wir

Variable	Startwert
Verstärkung V	1000
Untere Grenzfrequenz Ω_1	1000
Obere Grenzfrequenz Ω_2	50000
Filterordnung n_1, n_2	5

Tabelle 1: Startwerte für `curve_fit` der Daten an $g(f)$

Abbildung 5 zeigt das Resultat des Fits. Die genauen Werte der optimierten Parameter sind an dieser Stelle der Übersichtlichkeit halber nicht angegeben, können aber dem angehängten Python Code entnommen werden.

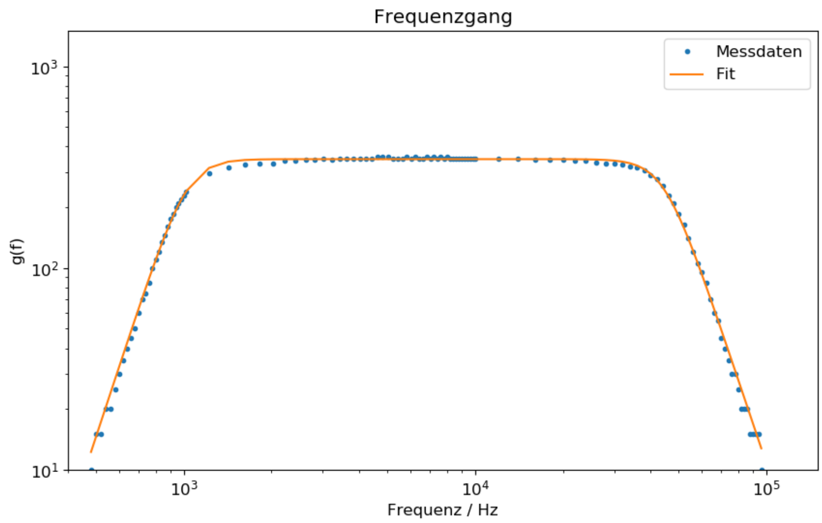


Abbildung 5: Frequenzgang $g(f)$ mit Fit

Um die Rauschbandbreite zu bestimmen, quadrieren wir zunächst die Funktion $g(f)$ und integrieren das Ergebnis numerisch im zuvor festgelegten Wertebereich der Frequenz. Die Integration liefert uns einen Wert von

$$B = (5.35 \pm 0.11) \cdot 10^9 \text{Hz}. \tag{8}$$

Der Fehler von B ist rein systematischer Natur und setzt sich aus unterschiedlichen Faktoren, wie Untergrund, Genauigkeit der Messinstrumente und auch Integrationsfehler zusammen. Zusammenfassend wählen wir daraus für die Größe einen relativen Fehler von 2%, wie es auch in der Praktikumsanleitung angegeben ist. Mit diesem Wert fahren wir zum nächsten Schritt fort.

3.2 Bestimmung der Boltzmannkostante

Für die Bestimmung der Boltzmannkonstante erinnern wir uns zunächst an die Nyquist-Beziehung

$$\langle U_r^2 \rangle = 4kTR\Delta f \tag{9}$$

für den Effektivwert der Rauschspannung U_r. Ersetzen wir die Bandbreite Δf durch die zuvor bestimmte Rauschbandbreite B, so ergibt sich ein linearer Zusammenhang zwischen dem Widerstand R und $\langle U_r^2 \rangle$ mit Steigung

$$c = 4kTB. \tag{10}$$

Um diese Steigung zu bestimmen, ziehen wir nun die Daten, aufgenommen in Aufgabenteil 2, hinzu. In Plot 6 zu sehen sind für die jeweiligen Widerstände die gemessenen Rauschspannungen U_r abzüglich der Rauschspannung der Messelektronik U_V, jeweils quadriert.

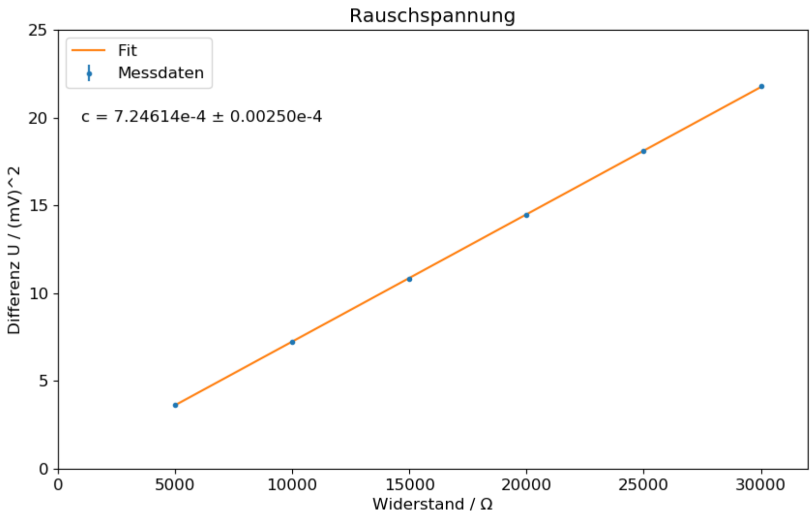


Abbildung 6: Rauschspannung der Widerstände abzüglich der Rauschspannung der Messelektronik

An diese Daten fitten wir nun eine einfache lineare Funktion der Form $f(x) = cx$. Das Ergebnis des Fits ist bereits als Gerade in Abbildung 6 zu sehen. Die bestimmte Steigung liegt bei

$$c = (7.2461 \pm 0.0026) \cdot 10^{-4} \frac{\mathrm{mV}^2}{\Omega}. \tag{11}$$

Um zusätzlich die Güte des Fits zu untersuchen, betrachten wir nun einmal die χ^2-Summe, sowie die reduzierte χ^2-Summe, χ^2_{red}.

$$\chi^2 = 83.07 \tag{12}$$
$$\chi^2_{\mathrm{red}} = 16.61 \tag{13}$$

Die Fitwahrscheinlichkeit für eine Wiederholungsmessung berechnete sich zu einem Wert von 0.0%.

Durch Umstellen der Gleichung (10) erhalten wir eine Formel für die Boltzmannkonstante

$$k = \frac{c}{4TB}. \tag{14}$$

Die bis jetzt in der Auswertung nicht erwähnte Umgebungstemperatur T ist ebenfalls dem Messprotokoll zu entnehmen. Diese lag bei $T = (23.00 \pm 0.10)^\circ\mathrm{C} = (296.15 \pm 0.10)\mathrm{K}$.

Setzen wir nun alle Werte in (14) ein so erhalten wir für die Boltzmannkonstante einen Wert von

$$k = (11.435 \pm 0.004\,(\mathrm{stat.}) \pm 0.229\,(\mathrm{sys.})) \cdot 10^{-23} \frac{\mathrm{J}}{\mathrm{K}}. \tag{15}$$

Wie der Praktikumsanleitung zu entnehmen, entspricht der statistische Fehler von k gerade dem relativen Fehler von c;

$$\Delta k_{\mathrm{stat.}} = \frac{\Delta c}{c} k. \tag{16}$$

Der systematische Fehler setzt sich zusammen aus dem Fehler der Zimmertemperatur T, sowie dem Fehler von B. Letzterer ist als eine Zusammensetzung verschiedener systematischer Fehlerquellen auf $\Delta B/B = 2\%$ abgeschätzt;

$$\Delta k_{\text{sys.}} = \sqrt{\left(\frac{\Delta T}{T}\right)^2 + \left(\frac{\Delta B}{B}\right)^2} k. \tag{17}$$

4 Zusammenfassung und Diskussion

Bestandteil dieses Versuchs war es, das Phänomen des thermischen Rauschens quantitativ zu untersuchen. Thermisches Rauschen tritt in einem elektrischen Leiter immer auf, sobald dieser eine Temperatur größer 0K hat und führt zu einer elektrischen Spannung, der Rauschspannung, im Leiter, ohne dass eine äußere Spannung angelegt ist. Der Effektivwert der Rauschspannung hängt laut der Nyquist-Beziehung, $\langle U_r^2 \rangle = 4kTR\Delta f$, von der Temperatur T, dem Widerstand des elektrischen Leiters R, der Bandbreite der Messelektronik Δf, sowie der Boltzmann-Konstante k ab. Letztere zu berechnen war das Ziel des Versuchs.

Der Versuch setzte sich aus zwei Teilen zusammen. Im ersten Versuchsteil bestimmten wir die Rauschbandbreite des Messsystems, zusammengesetzt aus einem Verstärker und einem Bandfilter. Hierzu zeichneten wir mittels eines Computers den Frequenzgang des Messsystems auf. Als Integral unter dem Quadrat des Frequenzgangs ergab sich zu Rauschbandbreite zu

$$B = (5.35 \pm 0.11) \cdot 10^9 \text{Hz}.$$

Für den zweiten Versuchsteil nahmen wir für unterschiedliche Widerstände die Effektivwerte der Rauschspannung auf. Da diese, nach der eben genannten Nyquist-Beziehung, in einem linearen Zusammenhang stehen, konnten wir mittels linearer Regression eine Geradensteigung von

$$c = (7.2461 \pm 0.0026) \cdot 10^{-4} \frac{\text{mV}^2}{\Omega}.$$

aus den gemessenen Werten bestimmen. Gemeinsam mit der zuvor ermittelten Rauschbandbreite ergab sich für die Boltzmannkonstante ein Wert von

$$k = (11.435 \pm 0.004\,(\text{stat.}) \pm 0.229\,(\text{sys.})) \cdot 10^{-23} \frac{\text{J}}{\text{K}}.$$

Der Literaturwert der Boltzmann konstante k liegt (nach *2022 CODATA recommended values*) bei

$$k = 1.380\,649 \cdot 10^{-23} \frac{\text{J}}{\text{K}}.$$

Es ist leicht zu erkennen, dass dieser Wert um etwas weniger als einen Faktor 10 von dem hier berechneten Wert abweicht.
Für die Beurteilung dieses Fehlers betrachten wir zunächst den Frequenzgang. Aus dem Plot 4, sowie den aufgezeichneten Rohdaten ist zu erkennen, dass das Plateau des Maximums bei ca. 300 bis 400 liegt. Im Vergleich zu anderen Auswertungen, sowie den Plots in der Praktikumsanleitung fällt hier ein Anstieg des Frequenzgangs bis auf einen Wert um

1000 auf. Wir können somit davon ausgehen, dass sich in unsere Messung des Frequenzgangs ein Fehler im Aufbau eingeschlichen hat. Der Fehler im Frequenzgang wirkt sich auf das Integral der Kurve und somit direkt auf die Berechnung der Boltzmannkonstante aus, was schon einmal einen großen Teil der Abweichung erklärt.
Weiter möchten wir nun den Fit an die Effektivwerte der Rauschspannung betrachten. Auch wenn der Plot es nicht vermuten lässt, so weißt der berechnete χ^2_{red}-Wert von 16.61, dieser sollte optimalerweise gegen 1 gehen, auf einen sehr schlechten Fit hin. Ebenso gilt dies für die berechnete Fitwahrscheinlichkeit von 0.0%. Grund für diese Werte könnten die statistischen Fehler sein. Als Fehler der Mittelwerte über eine Serie von ca. 110 Messungen, welche mit hoher Genauigkeit von einem Computer durchgeführt wurden, sind diese sehr klein. Als Nenner in der Berechnung der χ^2 Wert führen diese somit bereits bei geringen Abweichungen der Messwerte von den Funktionswerten zu großen Auswirkungen auf das Ergebnis.

Python Code, Hauptprogramm

auswertung

December 7, 2024

[89]:
```
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit
import scipy.integrate as integrate

%run ../lib.ipynb

plt.rcParams.update({'font.size': 12})
```

[90]:
```
f_a3 = np.loadtxt('aufgabe3_frequenzgang.txt', skiprows=1, usecols=(0,1),␣
↪unpack=True)
```

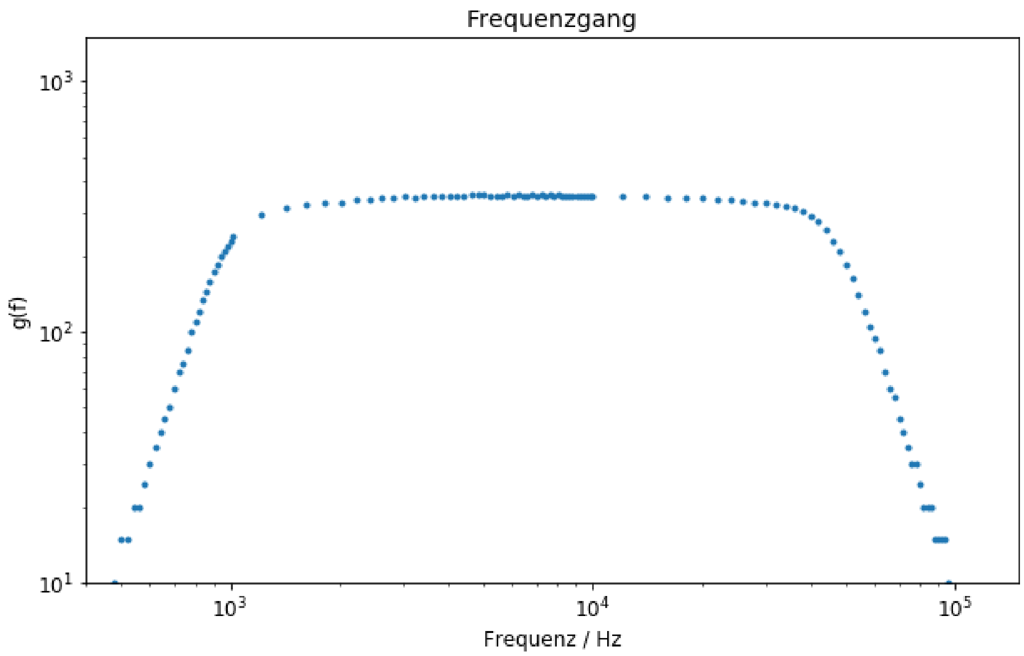
0.0.1 Konstanten

[91]:
```
D = 1e-3
U_ein = 0.2
lb = 19
ub = 13
U_aus = f_a3[1][lb:-ub]
g = U_aus / (U_ein * D)

np.max(g)
```

[91]: 354.99999999999994

0.0.2 Plot des Frequenzgangs

[92]:
```
plt.figure(figsize=(10,6))
plt.loglog(f_a3[0][lb:-ub], g, linestyle='None', marker='.')
plt.axis([4E2, 1.5e5, 10, 1.5E3])
plt.xlabel('Frequenz / Hz')
plt.ylabel('g(f)')
plt.title('Frequenzgang')
plt.savefig('freq_data.png', dpi=100)
```



0.0.3 Definition Fitfunktion

[93]:
```
def fit_func(f, V, W1, W2, n1, n2):
    hp_filter = np.sqrt(1 + 1 / (f / W1)**(2*n1))
    tp_filter = np.sqrt(1 + (f/W2)**(2*n2))
    return V / (hp_filter * tp_filter)
```

0.0.4 Startparameter Fitting

[94]:
```
verst = 1000
untere_grenzfreq = 1000
obere_grenzfreq = 50000
filterord_n1 = filterord_n2 = 5
p0 = [verst, untere_grenzfreq, obere_grenzfreq, filterord_n1, filterord_n2]
```

0.0.5 Fitting

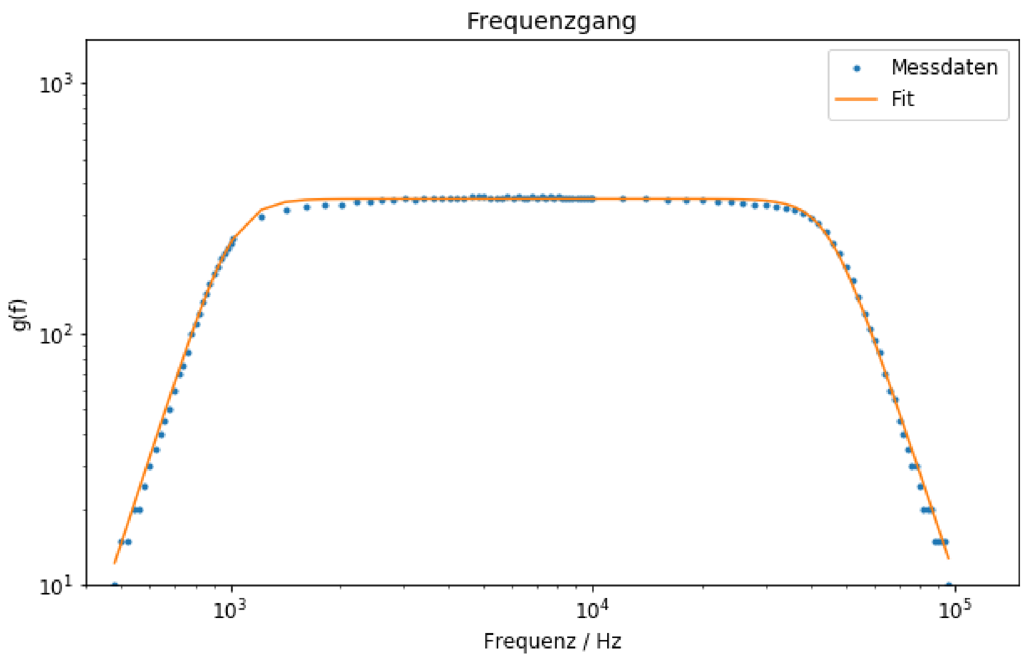
[95]:
```
popt_freq, pcov_freq = curve_fit(fit_func, f_a3[0][lb:-ub], g, p0)

V_ve = ValErr.fromFitAll(popt_freq, pcov_freq)

list(V_ve)
```

2

```
[95]: [ValErr(346.5518430884586, 0.8216377544429044),
 ValErr(1027.9483181651547, 4.764504463568716),
 ValErr(44598.41662432931, 203.70317002946283),
 ValErr(4.387039307736234, 0.09704251588395478),
 ValErr(4.301916086387025, 0.08225781876207226)]
```

0.0.6 Funktion + Fit

```
[96]: plt.figure(figsize=(10,6))
plt.loglog(f_a3[0][lb:-ub], g, linestyle='None', marker='.', label='Messdaten')
plt.loglog(f_a3[0][lb:-ub], fit_func(f_a3[0][lb:-ub], *popt_freq), label='Fit')
plt.axis([4E2, 1.5e5, 10, 1.5E3])
plt.xlabel('Frequenz / Hz')
plt.ylabel('g(f)')
plt.title('Frequenzgang')
plt.legend(loc='best')
plt.savefig('freq_data_fit.png', dpi=100)
```



0.0.7 Numerische Integration

```
[97]: def fit_func_square(f, V, W1, W2, n1, n2):
    return fit_func(f, V, W1, W2, n1, n2)**2
```

3

[98]:
```python
B_int_res = integrate.quad(fit_func_square, f_a3[0][lb], f_a3[0][-ub],
    args=tuple(popt_freq))

B = ValErr(B_int_res[0], B_int_res[0] * 0.02)
print(fr"Wert des Integrals: B = {B.strfmtf(3, 9)}")
```

```
Wert des Integrals: B = 5.349e9 ± 0.107e9
```

0.1 Bestimmung der Boltzmannkonstante

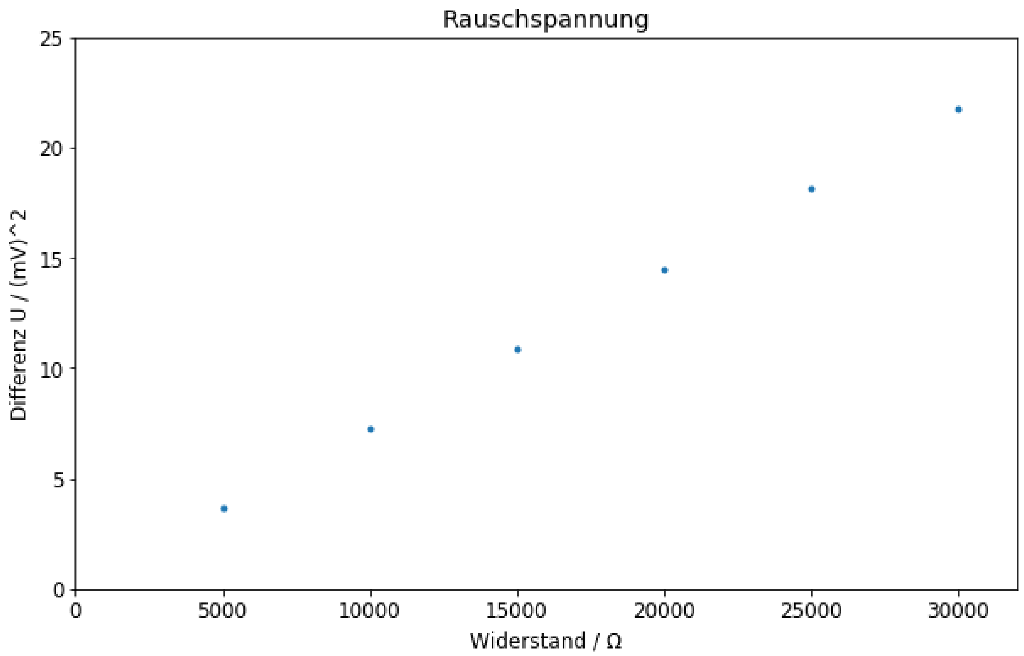
[99]:
```python
f_a2 = np.loadtxt('aufgabe2_rauschspannung.txt', skiprows=1, usecols=(0,1,2),
    unpack=True)
R = f_a2[0][:-1]
U = f_a2[1][:-1]
U_err = f_a2[2][:-1]

U_0 = ValErr(f_a2[1][-1], f_a2[2][-1])
```

0.1.1 Differez und Fehler der Differenz

[100]:
```python
def U_diff(u, u_0):
    return u**2 - u_0**2

def U_diff_err(u, u_err, u_0, u_0_err):
    return np.sqrt((u_err * 2 * u) ** 2 + (u_0_err * 2 * u_0) ** 2)

fehler_D=np.sqrt((U_err * 2 * U) ** 2 + (U_0.err * 2 * U_0.val) ** 2)

Diff = U_diff(U, U_0.val)
Diff_Err = U_diff_err(U, U_err, U_0.val, U_0.err)
```

0.1.2 Plot der Daten

[101]:
```python
plt.figure(figsize=(10,6))
plt.errorbar(R, Diff, yerr=Diff_Err, fmt='.')
plt.axis([0,3.2e4,0,25])
plt.xlabel('Widerstand / Ω')
plt.ylabel('Differenz U / (mV)^2')
plt.title('Rauschspannung')
plt.savefig('ur_data.png', dpi=100)
```



0.1.3 Linearer Fit an Daten

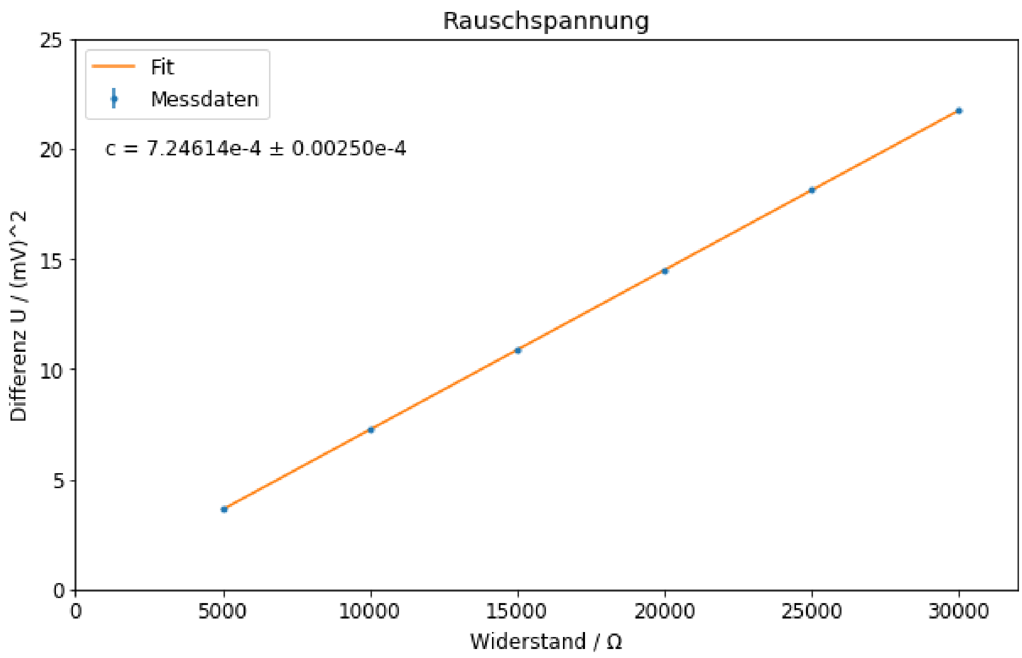
[102]:
```
def lin_fit_func(x, c):
    return x * c
```

[103]:
```
popt_lin, pcov_lin = curve_fit(lin_fit_func, R, Diff, sigma=Diff_Err,␣
↪absolute_sigma=True)

c = ValErr.fromFit(popt_lin, pcov_lin, 0)

print(f"Geradensteigung: c = {c.strfmtf(6, -4)}")
```

```
Geradensteigung: c = 7.246144e-4 ± 0.002505e-4
```

[104]:
```
plt.figure(figsize=(10,6))
plt.errorbar(R, Diff, yerr=Diff_Err, fmt='.', label='Messdaten')
plt.plot(R, lin_fit_func(R, *popt_lin), label='Fit')
plt.text(1000, 20, rf'c = {c.strfmtf(5, -4)}', horizontalalignment='left',
     verticalalignment='center')
plt.axis([0,3.2e4,0,25])
plt.xlabel('Widerstand / Ω')
plt.ylabel('Differenz U / (mV)^2')
plt.title('Rauschspannung')
plt.legend(loc='upper left')
```

```
plt.savefig('ur_data_fit.png', dpi=100)
```



0.1.4 Bestimmung der Güte des Fits

[105]:
```
chisquare=np.sum(((lin_fit_func(R,*popt_lin)-Diff)**2/Diff_Err**2))
dof=5 #degrees of freedom
chisquare_red=chisquare/dof
print(f'chi^2 = {chisquare}')
print(f'chi^2 red = {chisquare_red}')
```

```
chi^2 = 83.0708880081837
chi^2 red = 16.61417760163674
```

[106]:
```
from scipy.stats import chi2
prob=round(1-chi2.cdf(chisquare,dof),2)*100
print(f'Wahrscheinlichkeit = {prob}%')
```

```
Wahrscheinlichkeit = 0.0%
```

6

0.1.5 Ausrechnen der Boltzmannkonstante

[110]:
```
# c = 4kTB <=> k = c/(4TB),
T = 23. + 273.15
T_err = 0.1

k = c.val * (10**-6) / (4. * T * B.val)
k_stat = (c.err / c.val) * k
k_sys = np.sqrt((T_err / T)**2 + (B.err/B.val) ** 2) * k


f"k = {floatfmt(k, 5, -23)} ± {floatfmt(k_stat, 5, -23)} ± {floatfmt(k_sys, 5,␣
↪-23)}"
```

[110]:
```
'k = 11.43531e-23 ± 0.00395e-23 ± 0.22874e-23'
```

lib

December 7, 2024

[32]:
```python
def floatfmt(v, prec, exp):
    return f"{v/10**(exp):0=1.{prec}f}{f'e{exp}' if exp != 0 else ''}"
```

[36]:
```python
class ValErr:
    val: float = 0
    err: float = 0
    err_set = False

    def __init__(self, val, err=0):
        self.val = val
        if err != 0:
            self.err_set = True
            self.err = err

    def getTuple(self):
        return (self.val, self.err)

    @classmethod
    def fromTuple(self, tup):
        return ValErr(tup[0], tup[1])

    @classmethod
    def fromFit(self, popt, pcov, i):
        return ValErr(popt[i], np.sqrt(pcov[i][i]))

    @classmethod
    def fromFitAll(self, popt, pcov):
        for i in range(0, len(popt)):
            yield ValErr(popt[i], np.sqrt(pcov[i][i]))

    def strfmt(self, prec=2):
        if self.err != 0:
            return fr"{self.val:.{prec}e} ± {self.err:.{prec}e}"
        else:
            return f"{self.val:.{prec}e}"

    def strfmtf(self, prec, exp):
```

```
        if self.err != 0:
            return fr"{floatfmt(self.val, prec, exp)} ± {floatfmt(self.err,␣
↪prec, exp)}"
        else:
            return f"{floatfmt(self.val, prec, exp)}"

    def strltx(self, prec=2):
        if self.err != 0:
            return fr"{self.val:.{prec}e} \pm {self.err:.{prec}e}"
        else:
            return f"{self.val}"

    def __repr__(self):
        return f"ValErr({self.val}, {self.err})"

    def __mul__(self, num):
        return ValErr(self.val * num, self.err * num)
```

[34]:
```
def spacearound(dat, add):
    return np.linspace(dat[0] - add, dat[len(dat)-1] + add)
```

[35]:
```
def div_with_err(a, a_err, b, b_err):
    err = (1 / b) * np.sqrt(a_err**2 + (a * b_err / b)**2)
    return (a / b, err)
```

[ ]:
```

```

2