

Praktikanten:

Gerasimov, V. & Reiter, L.

Betreuer:

Wachs, D.

Versuchsdurchführung:
10. Dezember, 2019

Inhaltsverzeichnis

	Seite
1 Einführung	2
2 Versuchsaufbau, Literaturwerte & Vorbereitung	2
3 Durchführung	4
3.1 Sonnenspektrum	4
3.2 Qualitativer Vergleich einfacher Lichtquellspektren	4
3.3 Natriumspektrum	4
4 Messergebnisse	4
5 Auswertung mit Python	4
5.1 Source Code & Input	4
5.2 Output	10
5.3 Qualitativer Vergleich einfacher Lichtquellspektren	11
6 Fazit	12
7 Abbildungen	12
Gemessenes Sonnenspektrum mit und ohne Fenster	13
Absorption von Glas	14
Sonnenspektrum mit Fraunhoferlinien	14
ganzes Natriumspektrum	14
linkes Natriumspektrum	15
rechteres Natriumspektrum	15
1. Nebenserie des Na-Atoms	16
2. Nebenserie des Na-Atoms	16
Gemessenes Spektrum der Energiesparlampe und Glühlampe	17
Gemessenes Spektrum der Halogenlampe und des Lasers	18
Gemessene und normierte Spektren der LEDs	18

1 Einführung

In diesem Versuch¹ führen wir die Aufnahme des gestreuten Himmelslichts (weil die Sonne am Versuchstag verdeckt war) hinter einer Fensterscheibe und ohne Fensterscheibe durch. Danach ordnen wir die starken Fraunhoferlinien im Spektrum zu und messen die Absorption von Fensterglas durch Vergleich des Sonnenlichts ohne und mit Glasscheibe.

Wir nehmen noch Zusätzlichkeit die Spektren von anderen Lichtquellen auf und vergleichen sie Sonnenlicht, Glühlampe, LEDs, Energiesparlampe.

Als letztes führen wir einen Spektroskopie durch:

- Aufnahme des Spektrums einer Natriumlampe mit hoher Auflösung.
- Zuordnung der Serienübergänge der Hauptserie und von zwei Nebenserien.
- Bestimmung der vom Drehimpuls abhängigen Korrekturen der Energieniveaus im Natrium.

2 Versuchsaufbau¹, Literaturwerte & Vorbereitung

- Gitterspektrometer, Ocean Optics USB4000
- verschiedene Lichtquellen: LEDs, Laser, Energiesparlampe, Halogenlampe, Glühlampe, Natriumdampflampe

¹Dr. J.Wagner - Physikalisches Anfängerpraktikum - V. 1.1, Stand 1/2018, Versuch 234

3 Durchführung²

3.1 Sonnenspektrum

Registrieren und speichern Sie nacheinander die auf den Dunkelstrom korrigierten Spektren von:

- Himmelslicht bei geöffneten Fenster. Wir richten das Objektiv möglichst zum blauen Himmel aus.
- Himmelslicht hinter Fensterglas.
- Die Sonne war bei uns mit dichten Wolken bedeckt, deswegen konnten wir kein direktes Sonnenlicht aufnehmen.

3.2 Qualitativer Vergleich einfacher Lichtquellspektren

Wir nehmen fünf folgende Lichtquellen das Spektrum auf und speichern diese als interne SpectraSuite-Datei mit der Endung *.ProcSpec und als TXT-Datei:

- Glühlampe
- Halogenlampe
- Energiesparlampe
- 4 verschiedene LEDs (blau-rot, orange & weiß)
- Laser.

3.3 Natriumspektrum

Wir schließen die Natriumlampe an das Vorschaltgerät (Drossel) an und schalten die Drossel ein. Die Lampe braucht längere Zeit um zu zünden und bis sie stabil brennt. Wir richten das Objektiv auf die Lichtquelle aus. Es ist sofort klar, dass die gelbe Natriumlinie alle anderen Linien weit überstrahlt und wählen den Abstand bzw. die Blendeneöffnung so, dass die gelbe Linie bei einer Integrationszeit von 5 ms gerade in Sättigung geht. Die Aufnahme der Linien kleiner Intensität und des Spektrums der intensiven Linien nehmen wir jeweils einzeln vor und korrigieren den Dunkelstrom bei jeder Messung separat.

4 Messergebnisse

Messergebnisse wurden durch Ablesen der Abbildungen 3-6 abgeschätzt und in die Tabellen 1-3 übertragen.

5 Auswertung mit Python

5.1 Source Code & Input

λ_D und $\lambda_{1,3}$ entnehmen wir der Messung aus Tabelle 2. Wir benutzen folgende Formeln zur Näherung der Wellenlängen der Seiten:

$$\begin{aligned} E_{Ry} &= -13.605 \text{ eV} \\ hc &= 1.2398 \times 10^3 \text{ nm eV} \\ \lambda_{1,3} &= (819.10 \pm 1.10) \text{ nm} \\ \lambda_D &= (590.11 \pm 1.00) \text{ nm} \end{aligned}$$

$$E_{3p} = \frac{E_{Ry}}{3^2} - \frac{hc}{\lambda_{1,3}} \tag{1}$$

$$E_{3s} = E_{3p} - \frac{hc}{\lambda_D} \tag{2}$$

$$\Delta_s = 3 - \sqrt{\frac{E_{Ry}}{E_{3s}}} \tag{3}$$

Tabelle 1: Messung der Fraunhoferlinien

Symbol	Wellenlänge λ [nm]		Element/ Molekül
	Literaturwert ²	Messwert	
<i>A</i>	759.4 ± 0.1	760.59 ± 1.00	<i>O</i> ₂
<i>B</i>	686.7 ± 0.1	687.56 ± 1.00	<i>O</i> ₂
<i>C</i>	656.3 ± 0.1	656.78 ± 1.00	<i>H</i> α
<i>D</i> ₁	589.6 ± 0.1	590.11 ± 1.00	<i>N</i> <i>a</i>
<i>D</i> ₂	589.0 ± 0.1		
<i>D</i> ₃	587.6 ± 0.1	586.60 ± 1.00	<i>H</i> <i>e</i>
<i>E</i>	527.0 ± 0.1	527.90 ± 1.00	<i>F</i> <i>e</i> und <i>C</i> <i>a</i>
<i>b</i> ₁	518.4 ± 0.1	518.10 ± 1.00	<i>M</i> <i>g</i>
<i>F</i>	486.1 ± 0.1	487.48 ± 1.00	<i>H</i> β
<i>G</i>	430.8 ± 0.1	432.55 ± 1.00	<i>F</i> <i>e</i> und <i>C</i> <i>a</i>
<i>H</i>	396.8 ± 0.1	399.22 ± 1.00	<i>C</i> <i>a</i> ⁺
<i>K</i>	393.4 ± 0.1	395.69 ± 1.00	<i>C</i> <i>a</i> ⁺

¹ $\Delta\lambda$ der Messwerte folgt aus der abgeschätzten Halbwertsbreite der Peaks in Abbildung 3.

² Atomic Spectra Database:
<https://www.nist.gov/pml/atomic-spectra-database>

Tabelle 2: Messung des D-Linie und 1.Nebenserie

λ	Wellenlänge [nm]		Abbildung
	Näherung	Messwert	
λ_D	589.0	590.11 ± 1.00	4
$\lambda_{1,3}$	589.10	891.10 ± 1.10	6
$\lambda_{1,4}$	570.03	569.53 ± 1.00	4
$\lambda_{1,5}$	499.70	499.60 ± 1.10	5
$\lambda_{1,6}$	468.32	468.39 ± 1.20	5
$\lambda_{1,7}$	451.23	451.44 ± 1.20	5
$\lambda_{1,8}$	440.79	440.98 ± 0.80	5
$\lambda_{1,9}$	433.90	435.22 ± 1.30	5
$\lambda_{1,10}$	429.11	428.84 ± 1.20	5
$\lambda_{1,11}$	425.63	nicht gefunden	
$\lambda_{1,12}$	423.02	422.03 ± 1.60	5

¹ $\Delta\lambda$ der Messwerte folgt aus der abgeschätzten Halbwertsbreite der Peaks in entsprechender Abbildung.

$$\Delta_p = 3 - \sqrt{\frac{E_{Ry}}{E_{3p}}} \quad (4)$$

1. Nebenserie

$$\lambda_{1,m} \approx \frac{hc}{m^2 - E_{3p}} \quad \text{mit } m = 3, 4, \dots, 12 \quad (5)$$

2. Nebenserie

$$\lambda_{2,m} \approx \frac{hc}{(m - \Delta_s)^2 - E_{3p}} \quad \text{mit } m = 4, 5, \dots, 9 \quad (6)$$

Hauptserie

$$\lambda_{0,m} \approx \frac{hc}{(m - \Delta_p)^2 - E_{3s}} \quad \text{mit } m = 5, 6 \quad (7)$$

Fehler berechnen wir hier nicht, da wir uns nur für die ungefähren Werte von λ_m interessieren.

Header:

```

1 %matplotlib inline
2 import matplotlib.pyplot as plt
3 import numpy as np
4 from scipy.stats import chi2
5
6 def comma_to_float(valstr):
7     return float(valstr.decode("utf-8").replace(',','.'))

```

Mess- und Literaturdaten aus Tabellen 1, 2 und 3:

```

1 lambda_FH = np.array([759.4,666.7,666.3,589.6,589.0,587.6,527.0,518.4,486.1,430.8,396.8,393.4])
2
3 symbol_FH = np.array(['A','B','C','D_1','D_2','D_3','E','b_1','F','G','H','K'])
4
5 E_Ry = -13.605
6
7 hc = 1.2398e3
8
9 lambda_1 = 819.10
10 lambda_2 = 590.11

```

Näherung von E_{3p} , E_{3s} , Δ_s und Δ_p nach (1), (2), (4) bzw. (3):

```

1
2 E_3p = (E_Ry/(3**2))-(hc/lambda_1)
3 E_3s = (E_3p - (hc/lambda_2))
4 Delta_s = 3*np.sqrt(E_Ry/E_3s)
5 Delta_p = 3*np.sqrt(E_Ry/E_3p)

```

Näherung der 1. Nebenserie nach (5):

```

1 l_1 = np.zeros(13)
2 for m in range(3,13):
3     l_1[m]=hc/(E_Ry/(m**2-E_3p))
4     print('m={m:2d}, lambda_1={1:6.2f}'.format(m=m, l=l_1[m]))
5
6 print('\n')

```

Näherung der 2. Nebenserie nach (6):

```

1 l_2 = np.zeros(10)
2 for m in range(4,10):
3     l_2[m]=hc/(E_Ry/(m-Delta_s)**2-E_3p)
4     print('m={m:2d}, lambda_2={1:6.2f}'.format(m=m, l=l_2[m]))
5
6 print('\n')

```

Näherung der Hauptserie nach (7):

```

1 l_3 = np.zeros(6)
2 for m in range(4,6):
3     l_3[m]=hc/(E_Ry/(m-Delta_p)**2-E_3s)
4     print('m={m:2d}, lambda_0={1:6.2f}'.format(m=m, l=l_3[m]))

```

Abbildung 1 wird erstellt:

```

1 lamb_Himmel_freie, inten_Himmel_freie=np.loadtxt('data/Himmel_freie.txt', skiprows=17,
2 converters={0:comma_to_float, 1:comma_to_float},
3 comments='#', unpack=True)
4
5 lamb_Himmel_Fenster, inten_Himmel_Fenster=np.loadtxt('data/Himmel_Fenster.txt', skiprows=17,
6 converters={0:comma_to_float, 1:comma_to_float},
7 comments='#', unpack=True)
8
9 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
10 plt.plot(lamb_Himmel_freie, inten_Himmel_freie, label='ohne Fenster', color='C0')
11 plt.plot(lamb_Himmel_Fenster, inten_Himmel_Fenster, label='mit Fenster', color='C1')
12 plt.title('Gemessenes Sonnenspektrum mit und ohne Fenster')
13 plt.xlabel('Wellenlänge [nm]')
14 plt.ylabel('Intensität [a.u.]')
15 plt.legend()
16 plt.grid()
17 plt.ylim((0,600000))
18 plt.xlim((250,900))
19 plt.savefig('figures/234_Fig1.pdf', format='pdf')

```

Abbildung 2 wird erstellt:

```

1 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
2 A=inten_Himmel_Fenster/inten_Himmel_freie
3 plt.plot(lamb_Himmel_Fenster, A, color='C0')
4 plt.title('Absorption von Glas')
5 plt.xlabel('Wellenlänge [nm]')
6 plt.ylabel('Absorption [a.u.]')
7 plt.ylim((0,1))
8 plt.xlim((320,800))
9 plt.savefig('figures/234_Fig2.pdf', format='pdf')

```

Abbildung 3 wird erstellt:

```

1 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
2 plt.plot(lamb_Himmel_freie, inten_Himmel_freie, color='C0')
3 plt.title('Sonnenspektrum')
4 plt.xlabel('Wellenlänge [nm]')
5 plt.ylabel('Intensität [a.u.]')
6
7 for x in range(0, lambda_FH.size):
8     plt.axvline(x=lambda_FH[x], color='C'+str(x%10), linestyle='--', label=symbol_FH[x])
9
10 plt.legend()
11 plt.ylim((0,600000))
12 plt.xlim((350,800))
13 plt.savefig('figures/234_Fig3.pdf', format='pdf')

```

Abbildung 4 wird erstellt:

```

1 lamb_Na_1, inten_Na_1=np.loadtxt('data/linien_intensiv.txt', skiprows=17,
2 converters={0:comma_to_float, 1:comma_to_float},
3 comments='#', unpack=True)
4
5 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
6 for x in range(3, 13):
7     plt.axvline(x=l_1[x], color='C1', linestyle='--', label='$l_1$ $', m='+str(x))
8     for x in range(5, 10):
9         plt.axvline(x=l_2[x], color='C2', linestyle='--', label='$l_2$ $', m='+str(x))
10        for x in range(4, 6):
11            plt.axvline(x=l_3[x], color='C3', linestyle='--', label='$l_3$ $', m='+str(x))
12        plt.legend()
13        plt.plot(lamb_Na_1, inten_Na_1, color='C0')
14        plt.title('ganzes Natriumspektrum')
15        plt.xlabel('Wellenlänge [nm]')
16        plt.ylabel('Intensität [a.u.]')
17        pltyscale('log')
18        plt.xlim((300,850))
19        plt.savefig('figures/234_Fig4.pdf', format='pdf')

```

Abbildung 5 wird erstellt:

```

1 lamb_Na_2, inten_Na_2=np.loadtxt('data/linien_schwach_links.txt', skiprows=17,
2 converters={0:comma_to_float, 1:comma_to_float},
3 comments='#', unpack=True)

```

```

4 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
5 plt.plot(lamb_Na_2, inten_Na_2, color='C0')
6 for x in range(3*2, 13):
7     plt.axvline(x=l_1[x], color='C1', linestyle='--', label='$l_1$ $', m='+str(x))
8     for x in range(4, 10):
9         plt.axvline(x=l_2[x], color='C2', linestyle='--', label='$l_2$ $', m='+str(x))
10        for x in range(5, 6):
11            plt.axvline(x=l_3[x], color='C3', linestyle='--', label='$l_3$ $', m='+str(x))
12        plt.legend()
13        plt.title('linkes Natriumspektrum')
14        plt.xlabel('Wellenlänge [nm]')
15        plt.ylabel('Intensität [a.u.]')
16        plt.yscale('log')
17        plt.xlim((250,560))
18        plt.savefig('figures/234_Fig5.pdf', format='pdf')

```

Abbildung 6 wird erstellt:

```

1 lamb_Na_3, inten_Na_3=np.loadtxt('data/linien_schwach_rechts.txt', skiprows=17,
2 converters={0:comma_to_float, 1:comma_to_float},
3 comments='#', unpack=True)
4
5 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
6 plt.plot(lamb_Na_3, inten_Na_3, color='C0')
7 for x in range(3, 4):
8     plt.axvline(x=l_1[x], color='C1', linestyle='--', label='$l_1$ $', m='+str(x))
9     for x in range(5, 6):
10        plt.axvline(x=l_2[x], color='C2', linestyle='--', label='$l_2$ $', m='+str(x))
11        plt.legend()
12        plt.title('rechtes Natriumspektrum')
13        plt.xlabel('Wellenlänge [nm]')
14        plt.ylabel('Intensität [a.u.]')
15        plt.yscale('log')
16        plt.xlim((611,850))
17        plt.savefig('figures/234_Fig6.pdf', format='pdf')

```

Fit an die 1. Nebenserie und Abbildung 7 wird erstellt:

```

1 wellenl=np.array([819.10,569.53,499.60,468.39,451.44,440.98,435.22,428.84,422.03])
2 fehler=np.array([1.1,1.0,1.1,1.2,1.2,0.8,1.3,1.2,1.6])
3 quantenz=np.array([3,4,5,6,7,8,9,10,12])
4
5 from scipy.optimize import curve_fit
6
7 def fit_func(m,E_Ry,E_3p,Delta_d):
8     return hc/(E_Ry/(m-Delta_d)**2-E_3p)
9
10 para = [-13.6,-3,-0.02]
11 popt, pcov = curve_fit(fit_func, quantenz, wellenl,
12                        sigma=fehler, p0=para)
13
14 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
15 plt.errorbar(quantenz, wellenl, fehler, fmt='.', color='k')
16 plt.xlabel('Quantenzahl')
17 plt.ylabel('Wellenlänge / nm')
18 plt.title('1. Nebenserie des Na-Atoms')
19 plt.errorbar(quantenz, wellenl, fehler, fmt='.', color='k')
20 plt.xlabel('Quantenzahl')
21 plt.ylabel('Wellenlänge [nm]')
22 plt.title('1. Nebenserie des Na-Atoms')
23 x=np.linspace(2.8,12.2,100)
24 plt.plot(x, fit_func(x,*popt), color='C0')
25 plt.savefig('figures/234_Fig7.pdf', format='pdf')
26
27 chisquare =np.sum((fit_func(quantenz,*popt)-wellenl)**2/fehler**2)
28 dof=len(quantenz)-3 #dof:degrees of freedom, Freiheitsgrad
29 chisquare_red = chisquare/dof
30 prob = round(1-chi2.cdf(chisquare,dof),2)*100
31
32 print('E_Ry=", popt[0], ", Standardfehler=", np.sqrt(pcov[0][0]))
33 print('E_3p=", popt[1], ", Standardfehler=", np.sqrt(pcov[1][1]))
34 print('Delta_d=", popt[2], ", Standardfehler=", np.sqrt(pcov[2][2]))
35 print('Chi-Quadrat =', chisquare)
36 print('Freiheitsgrade =', dof)
37 print('Chi-Quadrat reduziert =', chisquare_red)
38 print('Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten =', prob, '%')

```

Für an die 2. Nebenserie und Abbildung 8 wird erstellt:

```
1 wellenl_2=np.array([616.73,516.49,476.73,456.55,443.85])
2 fehler_2=np.array([1.0,1.1,2.0,2.0,2.0])
3 quantenz_2=np.array([5,6,7,8,9])
4
5 from scipy.optimize import curve_fit
6
7 def fit_func(m,E_Ry,E_3p,Delta_s):
8     return hc/(E_Ry/(m-Delta_s)**2-E_3p)
9
10 para = [-13.6,-3,-0.02]
11 popt, pcov = curve_fit(fit_func, quantenz_2, wellenl_2,
12                        sigma=fehler_2, p0=para)
13
14 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
15 plt.errorbar(quantenz_2,wellenl_2,fehler_2,fmt=".", color='k',)
16 plt.xlabel('Quantenzahl')
17 plt.ylabel('Wellenlänge [nm]')
18 plt.title('2. Nebenserie des Na-Atoms')
19 x=np.linspace(3.8,12.2, 100)
20 plt.plot(x, fit_func(x,*popt), color='C0')
21 plt.savefig('figures/234_Fig8.pdf', format='pdf')
22
23 chisquare =np.sum((fit_func(quantenz_2,*popt)-wellenl_2)**2/fehler_2**2)
24 dof=len(quantenz_2)+3 #dof:degrees of freedom, Freiheitsgrad
25 chisquare_red = chisquare/dof
26 prob = round(1-chi2.cdf(chisquare,dof),2)*100
27
28 print("E_Ry=",popt[0], ", Standardfehler=", np.sqrt(pcov[0][0]))
29 print("E_3p=",popt[1], ", Standardfehler=", np.sqrt(pcov[1][1]))
30 print("Delta_s=",popt[2], ", Standardfehler=", np.sqrt(pcov[2][2]))
31 print('Chi-Quadrat =', chisquare)
32 print('Chi-Freiheitsgrade =', dof)
33 print('Chi-Quadrat reduziert =', chisquare_red)
34 print('Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten =', prob, '%')
```

Abbildung 9 wird erstellt:

```
1 lamb_1, inten_1=np.loadtxt('data/Energiesparlampe.txt', skiprows=17,
2                               converters={0:comma_to_float, 1:comma_to_float},
3                               comments='>>', unpack=True)
4
5 lamb_2, inten_2=np.loadtxt('data/Glühlampe.txt', skiprows=17,
6                               converters={0:comma_to_float, 1:comma_to_float},
7                               comments='>>', unpack=True)
8
9 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
10 plt.plot(lamb_1, inten_1, label='Energiesparlampe', color='C0')
11 plt.plot(lamb_2, inten_2, label='Glühlampe', color='C1')
12 plt.title('Gemessenes Spektrum der Energiesparlampe und Glühlampe')
13 plt.xlabel('Wellenlänge [nm]')
14 plt.ylabel('Intensität [a.u.]')
15 plt.legend()
16 plt.grid()
17 plt.ylim((0,60000))
18 plt.xlim((250,900))
19 plt.savefig('figures/234_Fig9.pdf', format='pdf')
```

Abbildung 10 wird erstellt:

```
1 lamb_1, inten_1=np.loadtxt('data/Halogenlampe.txt', skiprows=17,
2                               converters={0:comma_to_float, 1:comma_to_float},
3                               comments='>>', unpack=True)
4
5 lamb_2, inten_2=np.loadtxt('data/Laser.txt', skiprows=17,
6                               converters={0:comma_to_float, 1:comma_to_float},
7                               comments='>>', unpack=True)
8
9 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
10 plt.plot(lamb_1, inten_1, label='Halogenlampe', color='C0')
11 plt.plot(lamb_2, inten_2, label='Laser', color='C1')
12 plt.title('Gemessenes Spektrum der Halogenlampe und des Lasers')
13 plt.xlabel('Wellenlänge [nm]')
14 plt.ylabel('Intensität [a.u.]')
15 plt.legend()
16 plt.grid()
```

```
17 plt.ylim((0,60000))
18 plt.xlim((250,900))
19 plt.savefig('figures/234_Fig10.pdf', format='pdf')
```

Abbildung 11 wird erstellt:

```
1 lamb_1, inten_1=np.loadtxt('data/LED_blaue.txt', skiprows=17,
2                               converters={0:comma_to_float, 1:comma_to_float},
3                               comments='>>', unpack=True)
4
5 lamb_2, inten_2=np.loadtxt('data/LED_rot.txt', skiprows=17,
6                               converters={0:comma_to_float, 1:comma_to_float},
7                               comments='>>', unpack=True)
8
9 lamb_3, inten_3=np.loadtxt('data/LED_veif.txt', skiprows=17,
10                               converters={0:comma_to_float, 1:comma_to_float},
11                               comments='>>', unpack=True)
12
13 lamb_4, inten_4=np.loadtxt('data/LED_orange.txt', skiprows=17,
14                               converters={0:comma_to_float, 1:comma_to_float},
15                               comments='>>', unpack=True)
16
17 inten_1= inten_1/max(inten_1)
18 inten_2= inten_2/max(inten_2)
19 inten_3= inten_3/max(inten_3)
20 inten_4= inten_4/max(inten_4)
21
22 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
23 plt.plot(lamb_1, inten_1, label='blaue LED', color='C0')
24 plt.plot(lamb_2, inten_2, label='rote LED', color='C3')
25 plt.plot(lamb_3, inten_3, label='weiße LED', color='C2')
26 plt.plot(lamb_4, inten_4, label='orangene LED', color='C1')
27 plt.title('Gemessene und normierte Spektren der LEDs')
28 plt.xlabel('Wellenlänge [nm]')
29 plt.ylabel('Intensität (normiert auf Maximum) [a.u.]')
30 plt.legend()
31 plt.grid()
32 plt.ylim((0,1))
33 plt.xlim((250,900))
34 plt.savefig('figures/234_Fig11.pdf', format='pdf')
```

5.2 Output

Ergebnisse zur Näherung der Wellenlängen der 1. Nebenserie des Na-Atoms in Nanometer (Tab.2):

1	m= 3, lambda_1=819.10
2	m= 4, lambda_1=570.03
3	m= 5, lambda_1=499.70
4	m= 6, lambda_1=468.32
5	m= 7, lambda_1=451.23
6	m= 8, lambda_1=440.79
7	m= 9, lambda_1=433.90
8	m=10, lambda_1=429.11
9	m=11, lambda_1=425.63
10	m=12, lambda_1=423.02

Ergebnisse zur Näherung der Wellenlängen der 2. Nebenserie des Na-Atoms in Nanometer (Tab.3):

1	m= 4, lambda_2=1172.92
2	m= 5, lambda_2=622.30
3	m= 6, lambda_2=518.66
4	m= 7, lambda_2=477.60
5	m= 8, lambda_2=456.53
6	m= 9, lambda_2=444.13

Ergebnisse zur Näherung der Wellenlängen der Hauptreihe des Na-Atoms in Nanometer (Tab.3):

1	m= 4, lambda_0=332.46
2	m= 5, lambda_0=286.66

Ergebnisse zur Anpassung an die 1. Nebenserie des Na-Atoms in Elektronenvolt:

1	E_Ry= -13.430802196005462 , Standardfehler= 0.196903735602265
2	E_3p= -3.022605790583195 , Standardfehler= 0.00315816182613189

```

3 Delta_d= 0.0165722750831767 , Standardfehler= 0.019318306201450464
4 Chi-Quadrat = 1.5979187130426997
5 Freiheitsgrade = 6
6 Chi-Quadrat reduziert = 0.26631978550711494
7 Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten = 95.0 %

```

Ergebnisse zur Anpassung an die 2. Nebenserie des Na-Atoms in Elektronenvolt:

```

1 E_By= -13.126297627419813 , Standardfehler= 0.2742691385630246
2 E_Sp= -3.0182055025822163 , Standardfehler= 0.004344244415603463
3 Delta_s= 1.3912144565425775 , Standardfehler= 0.030426992519524913
4 Chi-Quadrat = 0.05378510189855742
5 Freiheitsgrade = 2
6 Chi-Quadrat reduziert = 0.02689255094927871
7 Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten = 97.0 %

```

Die Messung der 1. Nebenserie liefert uns also:

$$\begin{aligned}
 E_{Ry,1} &= -13.43(20) \text{ eV} \\
 E_{3p,1} &= -3.0226(32) \text{ eV} \\
 \Delta_d &= 0.016(19) \\
 \chi^2_{red,1} &= 0.02689255094927871
 \end{aligned}$$

mit einer Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten von 95.0%.

Die Messung der 2. Nebenserie liefert uns also:

$$\begin{aligned}
 E_{Ry,2} &= -13.13(27) \text{ eV} \\
 E_{3p,2} &= -3.0182(43) \text{ eV} \\
 \Delta_s &= 1.391(30) \\
 \chi^2_{red,2} &= 0.05378510189855742
 \end{aligned}$$

mit einer Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten von 97.0%.

5.3 Qualitativer Vergleich einfacher Lichtquellspektren

Abschätzte mittlere Wellenlängen:

Glühlampe:	≈ 630 nm
Halogenlampe:	≈ 620 nm
Energiesparlampe:	≈ 580 nm
blaue LED:	≈ 460 nm
rote LED:	≈ 630 nm
weiße LED:	≈ 510 nm
orangene LED:	≈ 590 nm
Laser:	≈ 530 nm

Die Energiesparlampe und die weiße LED haben mehrere Peaks, d.h. sie erzeugen die wahrgenommene Farbe durch Ausnutzung der menschlichen Wahrnehmung - unsere Netzhaut misst nicht die Wellenlänge des auftretenden Lichts, sondern nur wie stark die 3 verschiedenen Rezeptoren von ihnen jeweils angeregt werden. Wenn also Licht verschiedener Wellenlängen die Netzhaut trifft, erzeugt das Gehirn den Eindruck einer Farbe. Diese Farbe kann so aussehen wie die wahrgenommene Farbe zu einer bestimmten Wellenlänge, oder auch Farben erzeugen, die keiner spezifischen einzelnen Wellenlänge zugeordnet werden können. Diese Farben können nur als „Mischfarbe“ existieren (z.B. Grün).

Wir nehmen bläuliche Lichtquellen als „kalt“ und eher rötliche als „warm“ wahr. Das ist evolutionär und kulturell bedingt, da normalerweise Temperaturstrahlter genau andersrum erscheinen (wärmer → höhere mittlere Wellenlänge → erscheint bläulicher)

6 Fazit

Die Werte für E_{Ry} sind relativ nah zu einander und zum Literaturwert:

$$\begin{aligned}
 E_{Ry,1} &= -13.43(20) \text{ eV} \\
 E_{Ry,2} &= -13.13(27) \text{ eV} \\
 E_{Ry} &= -13.605693122994(26) \text{ eV}
 \end{aligned}$$

mit

$$\begin{aligned}
 E_{Ry,2} - E_{Ry,1} &\approx 0.9 \sigma \\
 E_{Ry,1} - E_{Ry} &\approx 0.9 \sigma \\
 E_{Ry,2} - E_{Ry} &\approx 1.7 \sigma
 \end{aligned}$$

Die Werte für E_{3p} sind auch nah zu einander :

$$\begin{aligned}
 E_{3p,1} &= -3.0226(32) \text{ eV} \\
 E_{3p,2} &= -3.0182(43) \text{ eV}
 \end{aligned}$$

mit

$$E_{3p,2} - E_{3p,1} \approx 0.8 \sigma$$

Somit konnten wir innerhalb der Ergebnisse als auch zwischen ihnen und den Literaturwert für E_{Ry} keine signifikanten Unterschiede feststellen. Die durchgeführten Korrekturen waren innerhalb unserer Messunsicherheiten ausreichend um alle signifikanten systematischen Fehler zu beseitigen. Es ist zudem möglich, dass die Fehler für die Wellenlängen zu groß abgeschätzt wurden, da sie Chi-Quadrat-Werte

$$\begin{aligned}
 \chi^2_{red,1} &= 2.7 \times 10^{-2} \\
 \chi^2_{red,2} &= 5.4 \times 10^{-2}
 \end{aligned}$$

kleiner sind als erwartet.

7 Abbildungen

Es folgen alle Abbildungen .

Abbildung 2

$\lambda_{2,m}$	Wellenlänge [nm]		Abbildung
	Näherung	Messwert	
$\lambda_{2,4}$	1172.92	außerhalb von unserem Spektrum	
$\lambda_{2,5}$	622.30	616.73 ± 1.00	6
$\lambda_{2,6}$	518.66	516.49 ± 1.10	5
$\lambda_{2,7}$	477.60	476.73 ± 2.00	5
$\lambda_{2,8}$	456.53	456.55 ± 2.00	5
$\lambda_{2,9}$	444.13	443.85 ± 2.00	5

¹ $\Delta\lambda$ der Messwerte folgt aus der abgeschätzten Halbwertsbreite der Peaks in entsprechender Abbildung.

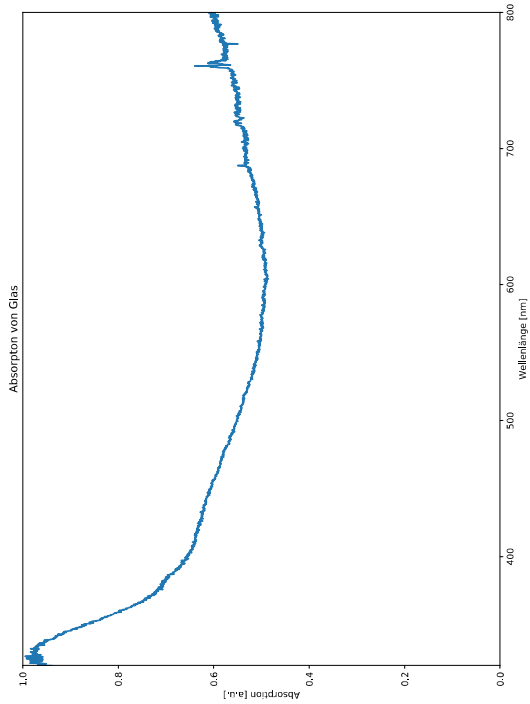


Abbildung 3

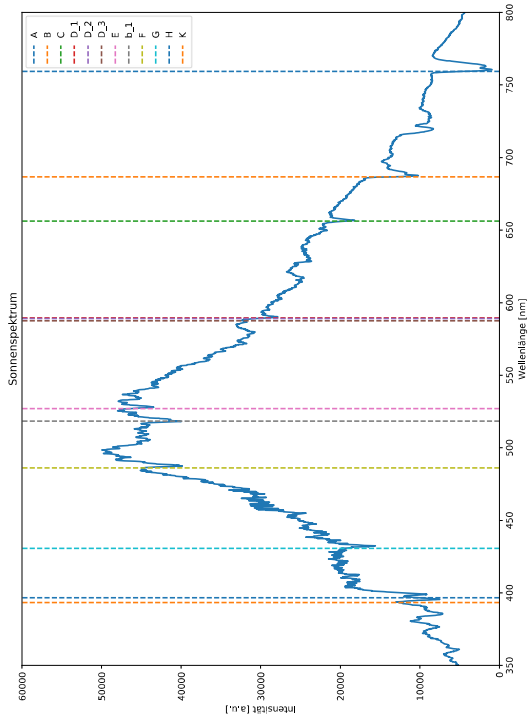
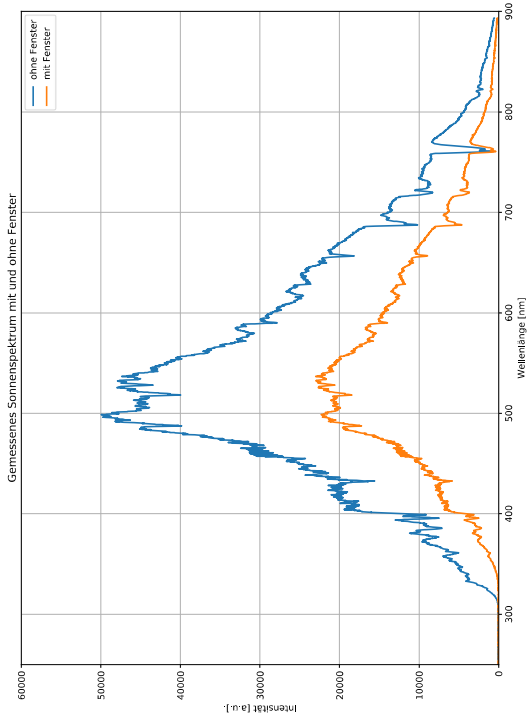


Abbildung 4

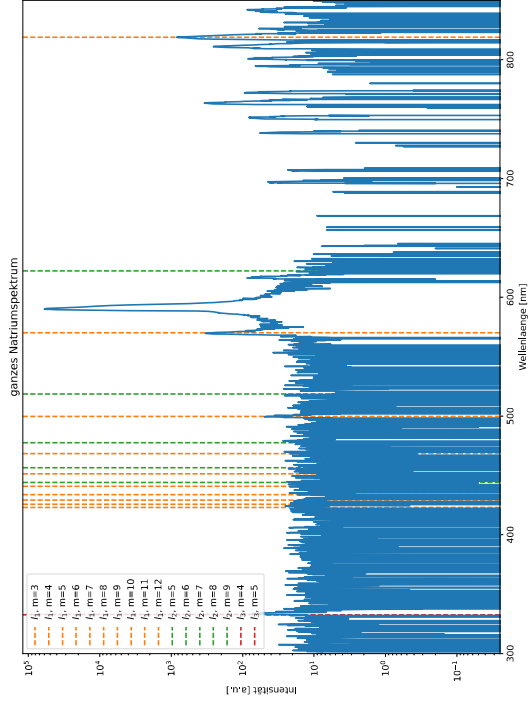


Abbildung 6

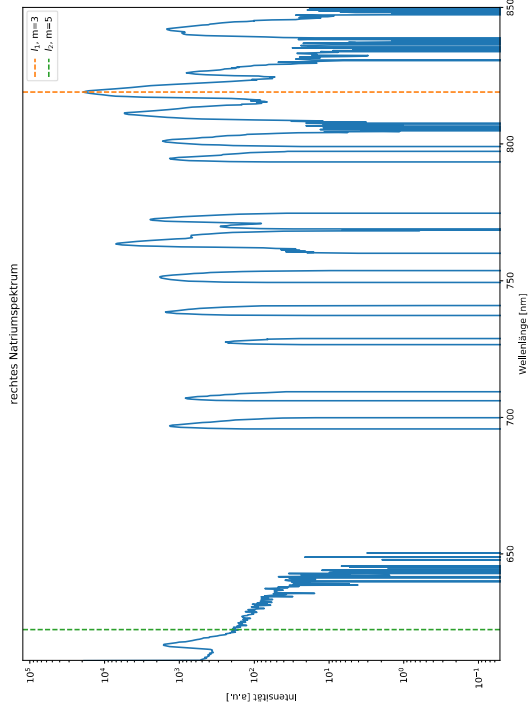


Abbildung 5

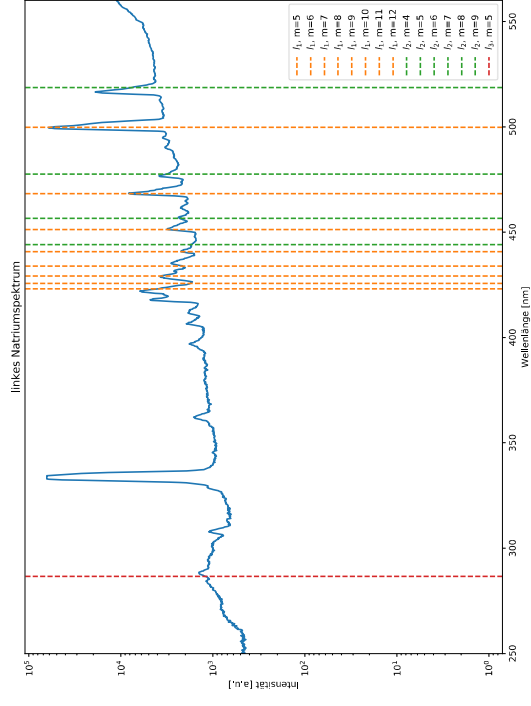


Abbildung 7

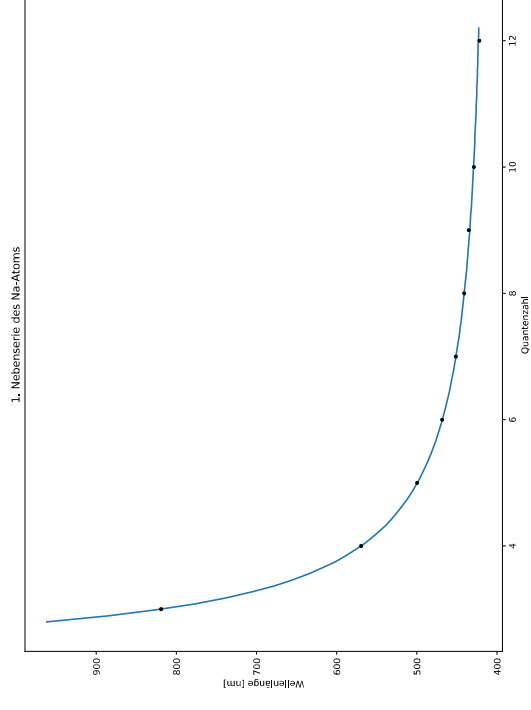


Abbildung 8

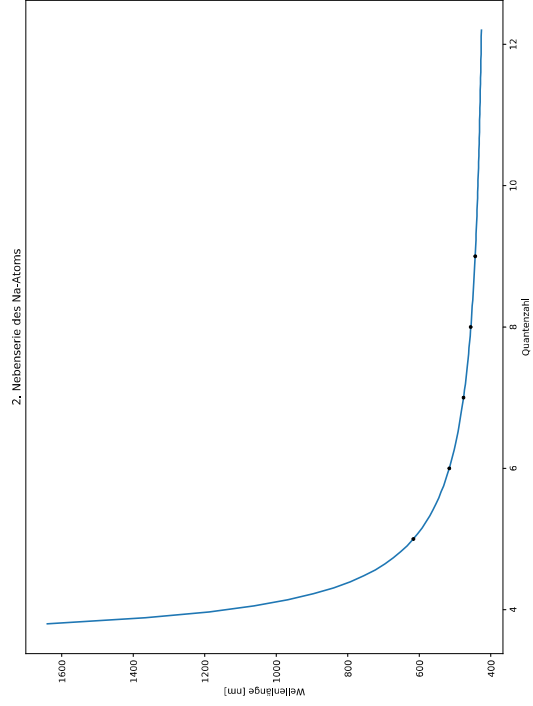


Abbildung 10

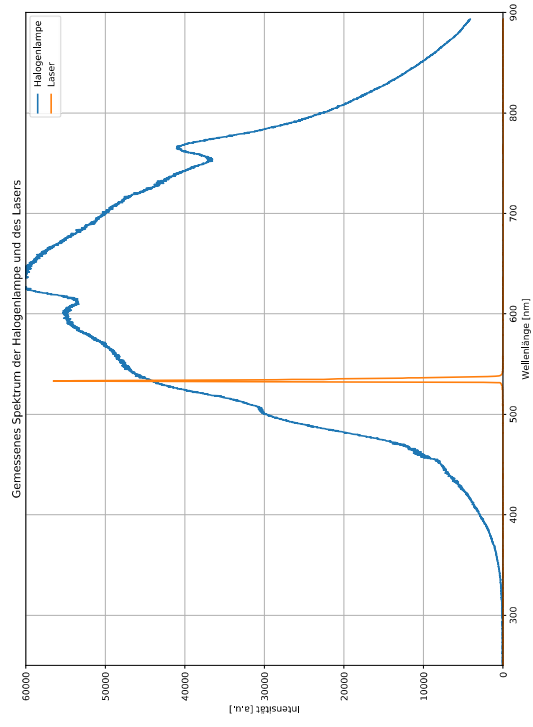


Abbildung 9

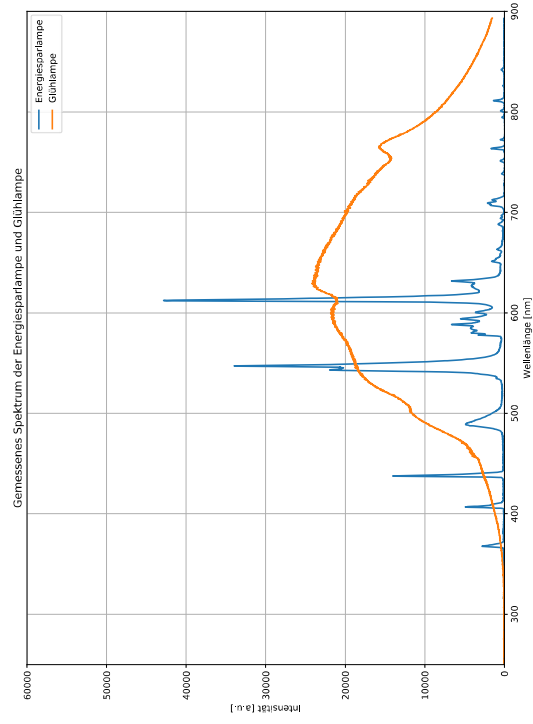


Abbildung 11

