

## Inhaltsverzeichnis

<b>1</b>	<b>Ziel des Versuches</b>	<b>2</b>
<b>2</b>	<b>Grundlagen</b>	<b>2</b>
2.1	Temperaturstahler . . . . .	2
<b>3</b>	<b>Versuchsaufbau</b>	<b>4</b>
3.1	Materialen und Geräte . . . . .	4
3.2	Aufbau . . . . .	4
<b>4</b>	<b>Messung und Auswertung</b>	<b>5</b>
4.1	Messprotokoll . . . . .	5
4.2	Auswertung . . . . .	6
<b>5</b>	<b>Zusammenfassung und Diskussion</b>	<b>20</b>
5.1	Zusammenfassung . . . . .	20
5.2	Diskussion . . . . .	21
<b>6</b>	<b>Quellen</b>	<b>26</b>
<b>7</b>	<b>Anhang</b>	<b>28</b>

# 1 Ziel des Versuches

In diesem Versuch werden wir das Spektrum verschiedener Lichtquellen untersuchen. Dabei sind unsere wichtigsten Zielen die Fraunhofer und Balmer Linien bei Sonnenlicht zu analysieren, die Neben- und Hauptserien von Natrium zu untersuchen und damit die Rydberg- und Serienenergie bestimmen.

## 2 Grundlagen

### 2.1 Temperaturstahler

Jeder Körper emittiert Strahlung in Form von elektromagnetischen Wellen, deren Intensität von der Temperatur des Körpers abhängt. Je höher die Temperatur, desto höher die Intensität. Um diese Intensität zu beschreiben geht man von einem Schwarzkörperstrahler aus. Bei einem solchen Modell betrachtet man einen Körper, der all die auf ihn einfallende Strahlung absorbiert. Die ausgehende Strahlung wird durch das Planck'sche Gesetz beschrieben

$$M_\lambda(\lambda, T) dA d\lambda = \frac{2\pi hc^2}{\lambda^5} \cdot \frac{1}{\exp\left(\frac{hc}{\lambda kT}\right) - 1} dA d\lambda. \quad (1)$$

$M_\lambda$  ist hier die Strahlungsleistung. Die Intensitätsverteilung für verschiedene Temperaturen ist auf Abbildung 1 dargestellt.

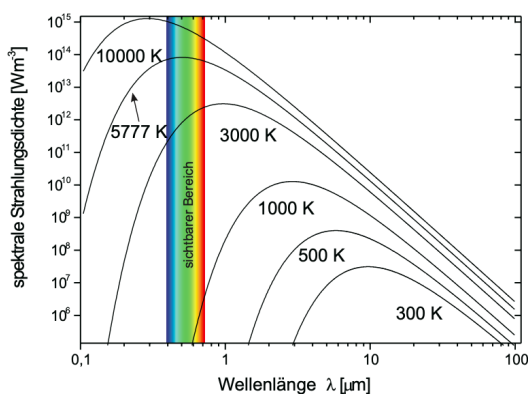


Abbildung 1: Intensitätsverteilung für verschiedene Temperaturen

Bei etwa 5500 K werden alle Wellenlängen des sichtbaren Spektrums etwa mit gleicher Intensität emittiert und das Licht wirkt dadurch weiß.

Die dort beobachtete Verschiebung lässt sich durch das Wiensche Verschiebungsgesetz erklären

$$\lambda_{\max} = \frac{2897,8 \mu\text{m} \cdot K}{T}. \quad (2)$$

Mit zunehmender Temperatur wird immer wieder mehr Energie in Form von Licht kleiner werdenden Wellenlänge emittiert, das heißt auch energetischeres Licht. Es wird also zuerst rotes Licht, dann irgendwann ein grüner Anteil.

### 2.1.1 Sonnenspektrum

Wenn das warme Licht der Sonne auf die Atmosphäre trifft, so wird dieses durch Elementen gestreut, und zwar wird aufgrund der wellenlängeabhängigen Rayleigh-Streuung blaues Licht deutlich stärker gestreut als rotes Licht, weshalb der Himmel auch blau erscheint. Bei der Vermessung des Sonnenspektrums auf der Erde erkennt man allerdings spezifische "Absorptionslinien", die dadurch entstehen, weil sowohl als auf der Sonnen- als auch auf der Erdatmosphäre das Licht von verschiedenen Elementen absorbiert wird. Zum Beispiel absorbiert Ozon ein großer Teil der ultravioletten Strahlung. In den Fraunhofer Linien kann man dieses Spektrum wieder erkennen.

### 2.1.2 Nichttemperaturstrahler

Bei Nichttemperaturstrahlern erfolgt die Ausstrahlung von Licht durch die Anregung der Atomzuständen in Gasen oder Festkörpern, oder zum Beispiel durch die Rekombination von Elektron-Loch-Paare in Halbleitern. Diese Lichtquellen besitzen ein diskretes Spektrum was materialabhängig ist. Zum Beispiel finden wir Gasentladungslampen, Leuchtdioden oder LASER.

Bei LED basiert die Emission von Licht auf Rekombination von Elektron-Loch-Paare in pn-Übergänge in einem Halbleiter. Meistens bestehen diese Halbleiter aus Galliumkombinationen wie z.B. Galliumarsenid GaAs (IR), AlGaAs (rot, IR), GaAsP (rot, orange, gelb), GaP (grün) oder zum Beispiel SiC (blau).

### 2.1.3 Natriumspektrum

Gasentladungslampen untersuchen wir anhand des Natriumspektrums. Alkaliatome besitzen ähnlich zum Wasserstoff nur ein Leuchtelektron. Nah zum Kern sieht das Leuchtelektron als das kugelsymmetrische Potential

$$V(r) = -\frac{Ze^2}{4\pi\epsilon_0 r} \quad (3)$$

und weit entfernt

$$V(r) = -\frac{e^2}{4\pi\epsilon_0 r}. \quad (4)$$

Die Energieniveaus an dieser Stelle hängen nicht mehr ausschließlich von der Hauptquantenzahl  $m$  ab, sondern auch vom Drehimpuls  $l$

$$E_{m,l} = E_{\text{Ry}} \cdot \frac{1}{(m - \Delta_{l,m})^2} \quad (5)$$

wobei der Korrekturterm  $\Delta_{l,m}$  nur wenig von  $m$  abhängt.

Im Grundzustand befindet sich das Elektron im  $3s$  Zustand. Die Hauptserie entspricht dem Übergang  $mp \rightarrow 3s$ , die erste Nebenserie  $md \rightarrow 3p$  und die zweite Nebenserie  $ms \rightarrow 3p$ . Das entstehende Spektrum ist Druck- und Feldstärkeabhängig und ist also insbesondere nicht konstant.

## 3 Versuchsaufbau

### 3.1 Materialien und Geräte

- Gitterspektrometer, Ocean Optics USB4000
- verschiedene Lichtquellen: LED, LASER, Energiesparlampe, Halogenlampe, Glühbirne, Natriumdampfampe
- PC

### 3.2 Aufbau



Abbildung 2: Versuchsaufbau nach Praktikumsskript

## 4 Messung und Auswertung

### 4.1 Messprotokoll

Die Messdaten für dieses Auswertung wurden aus der offiziellen Praktikumsseite und stammen aus dem Sommersemester 2021. Für die Durchführung wurde stattdessen das Video, was von Herrn Wagner aufgenommen wurde benutzt. In diesem ist die Aufnahme der gegebenen Messungen enthalten.

Zur Aufgabe 2: Qualitativer Vergleich einfacher Lichtquellspektren besitzen wir diverse Bilder:



Abbildung 3: Spektrum der Energiesparlampe

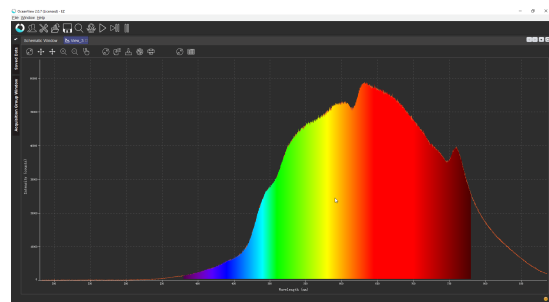


Abbildung 4: Spektrum der Glühlampe

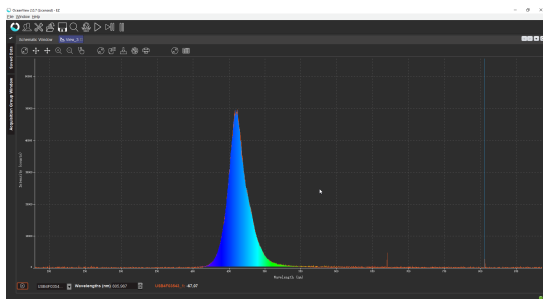


Abbildung 5: Spektrum der blauen LED

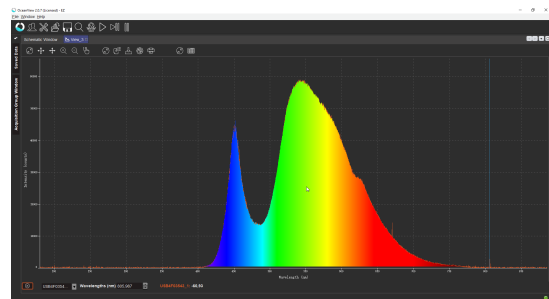


Abbildung 6: Spektrum der kaltweißen LED

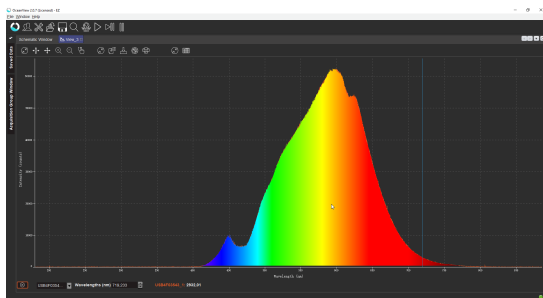


Abbildung 7: Spektrum der LED Lampe

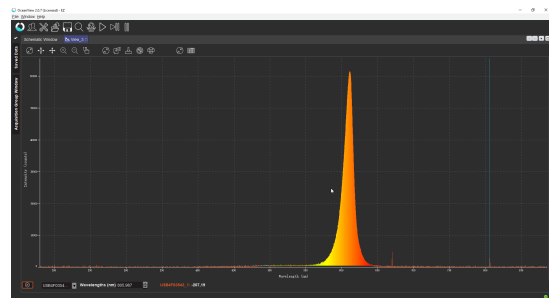


Abbildung 8: Spektrum der orangenen LED

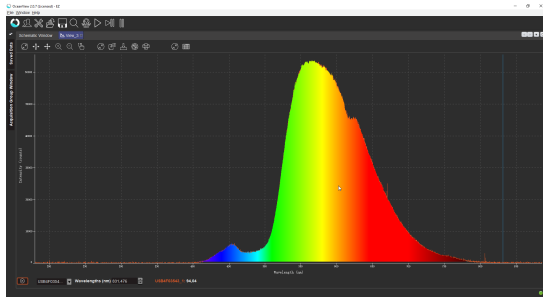


Abbildung 9: Spektrum der ersten warmweißen LED

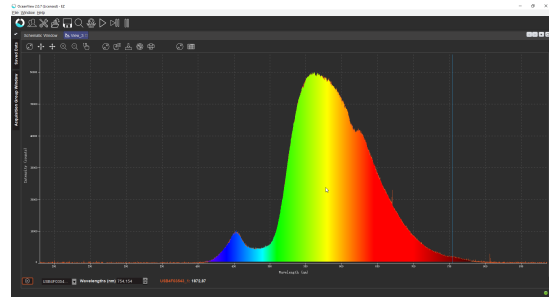


Abbildung 10: Spektrum der zweiten warmweißen LED

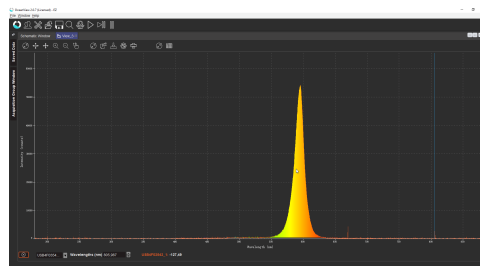


Abbildung 11: Spektrum der roten LED

## 4.2 Auswertung

Zur Auswertung dieses Versuches werden aufgrund von der Nicht-Teilnahme am Präsenzversuch die öffentlichen Messwerten des Sommersemesters 2021.

Im Folgenden wird bei der Fehleranalyse wenn nicht anders explizit angegeben die Gaußsche Fehlerfortpflanzung benutzt um die Fehlern der Größen zu bestimmen. Diese wird explizit in der digitalen Auswertung durch Python und wird in trivialen Fällen nicht nochmal bei der Ausarbeitung vorkommen.

### 4.2.1 Vergleich verschiedener Lichtquellen

Aus den verschiedenen Lichtquellen wollen wir zwei Vergleiche durchführen. Als erstes werden wir den Unterschied zwischen einer Energiesparlampe, einer Glühlampe und einer LED-Lampe durchführen. Dies erfolgt mithilfe von Diagramm 1

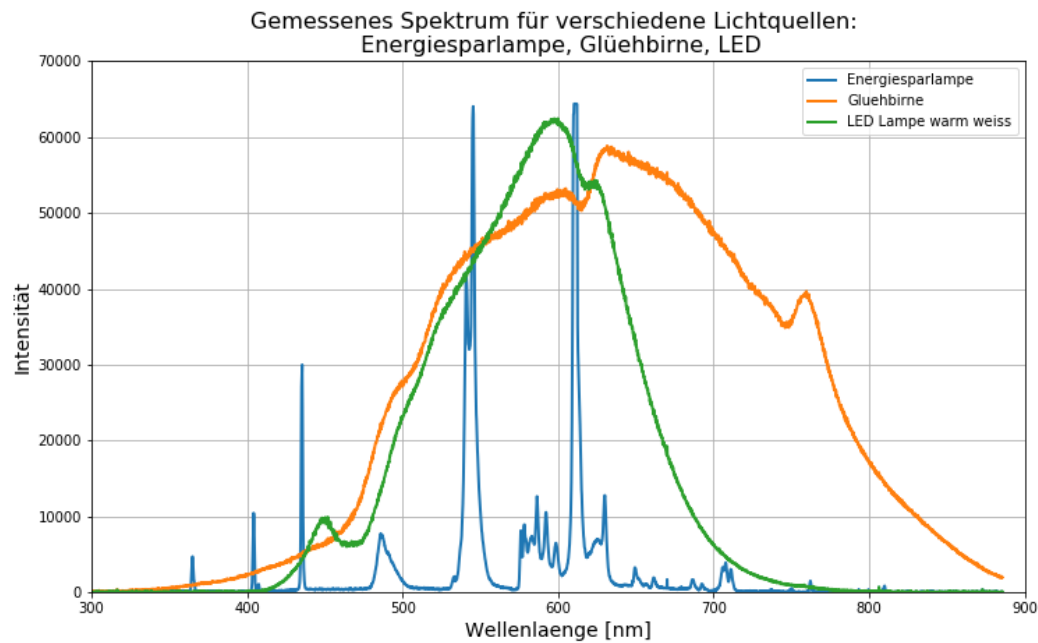


Diagramm 1: Vergleich verschiedener Lichtquellen: Energiesparlampe, Glühbirne und LED

Wir können qualitativ sehen, weshalb die Energiesparlampe, ja, Energie spart. Die Intensität ist ja nur in ganz kleinen Peaks "strategisch" verteilt. Stattdessen sind die Glühbirne und die LED weit ausgebreitet über ein kontinuierliches Spektrum. Die Glühbirne geht schon sogar weit in den infraroten Bereich.

Bei den verschiedenen farbigen LED's entdecken wir genau was wir erwarten, und zwar vereinzelte Peaks bei den wohl definierten Farben.

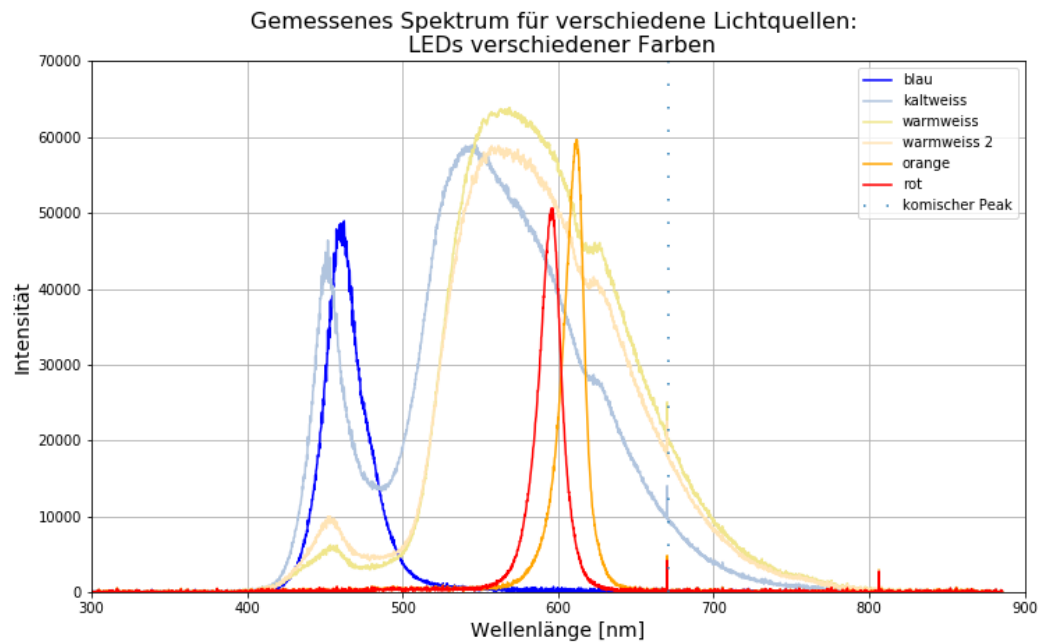


Diagramm 2: Vergleich verschiedener Lichtquellen: LEDs verschiedener Farben

Im Fall der kalt- bzw. warmweißen LEDs sehen wir auch ein weiter ausgebreitetes Spektrum. Für kaltweißes Licht ist dieses Spektrum weiter in Richtung violett verschoben und besitzt ein hohes Peak bei den blauen Wellenlängen. Interessante Bemerkung ist, dass bei allen LEDs finden wir einen sehr kleinen Peak um die Linie der 671 nm. Diese Linie können wir nicht erklären, aber wäre eine interessante Untersuchung. Unsere Hypothesen werden wir kurz in der Diskussion erläutern.

Lustig ist ebenfalls, dass der orangefarbene Peak weiter in die roten Wellenlängen als der rote Peak liegt.

#### 4.2.2 Sonnenspektrum

Wir tragen die Intensität gegen die Wellenlänge des Sonnenlichtes einmal bei geöffnetem und einmal bei geschlossenem Fenster auf folgendem Diagramm auf.



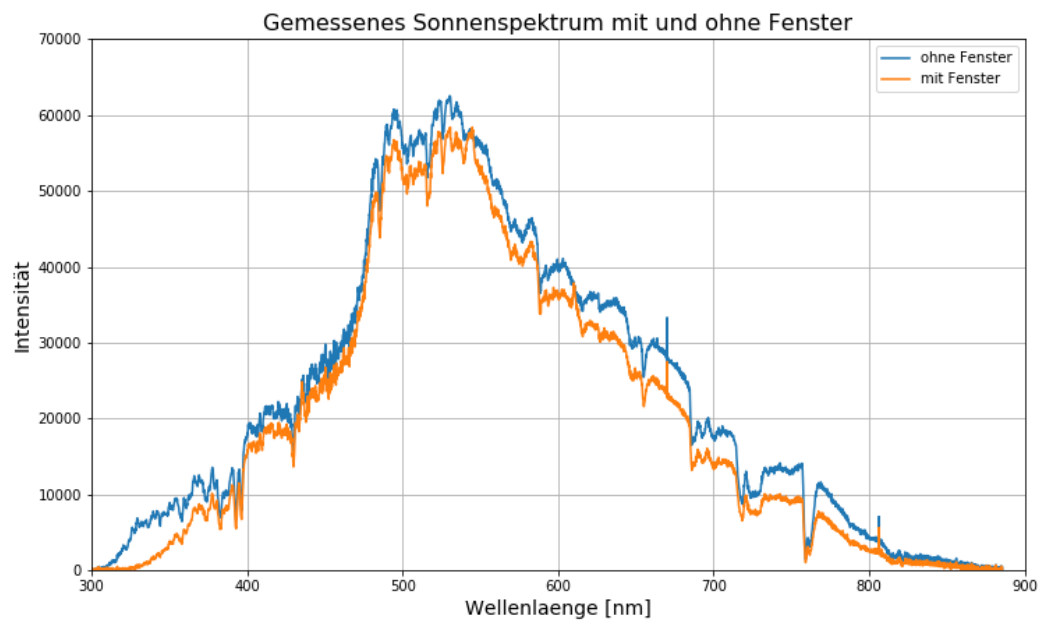


Diagramm 3: Sonnenspektrum bei geöffnetem und geschlossenem Fenster

Als Unterstützung berechnen wir die Absorption von Glas gemäß

$$A_{\text{Glas}} = 1 - \frac{I_{mg}}{I_{og}} \quad (6)$$

wobei  $mg$ ,  $og$  für mit und ohne Glas stehen. Die Absorption tragen wir wieder in Abhängigkeit der Wellenlänge auf, ausschließlich im sichtbaren Bereich.

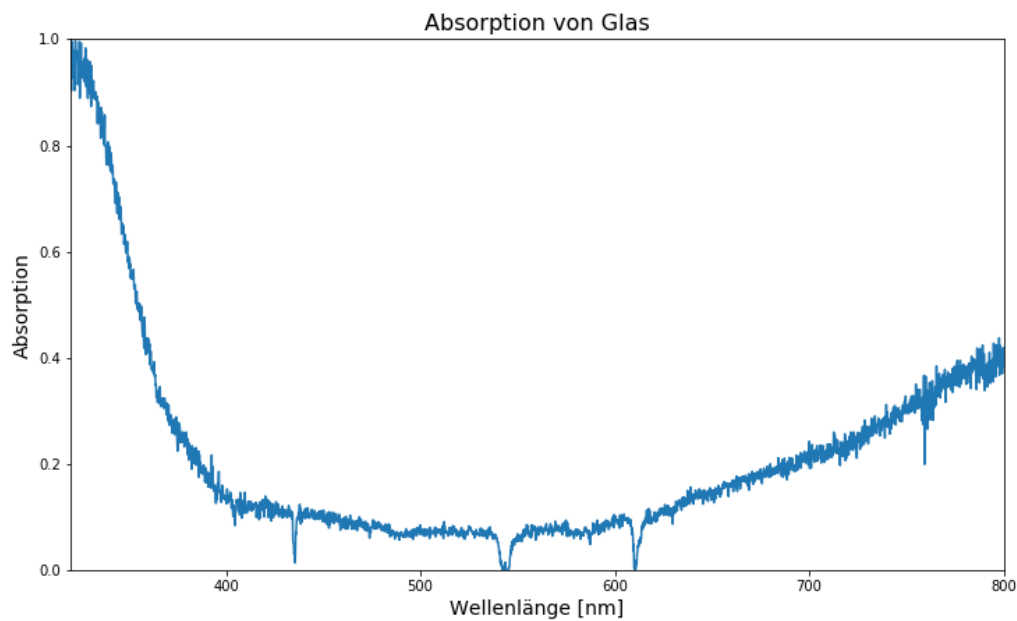


Diagramm 4: Absorption im sichtbaren Bereich des Lichts

Hier erkennt man, dass im sichtbaren Bereich die Absorption drastisch abstürzt und gegen die ultravioletten und infraroten Wellenlängen wieder steigt. Dies stimmt mit den Erwartungen zu, denn Glas ist ja durchsichtig für sichtbares Licht, aber ultraviolettes und infrarotes Licht wird davon absorbiert. Ein Teil des sichtbaren Lichts wird aber mit Sicherheit absorbiert, oder wenn nicht unbedingt absorbiert, gestreut und reflektiert, weshalb wir im sichtbaren Bereich trotzdem eine kleine Absorption erkennen.

Als nächstes betrachten wir das reine Sonnenspektrum ohne Glas und versuchen die zugehörigen Fraunhofer Linien und die der Balmerserie zu finden. Nicht alle lassen sich erkennen, also beschränken wir uns darauf die die wir sehen können zu kennzeichnen. Auf Diagramm 5 sind neben der markierten Linien auch die Namen und abgeschätzten Wellenlängen beschriftet, diese sind in aufsteigender Reihenfolge untereinander, denn alles weitere würde zu unübersichtlich sein.

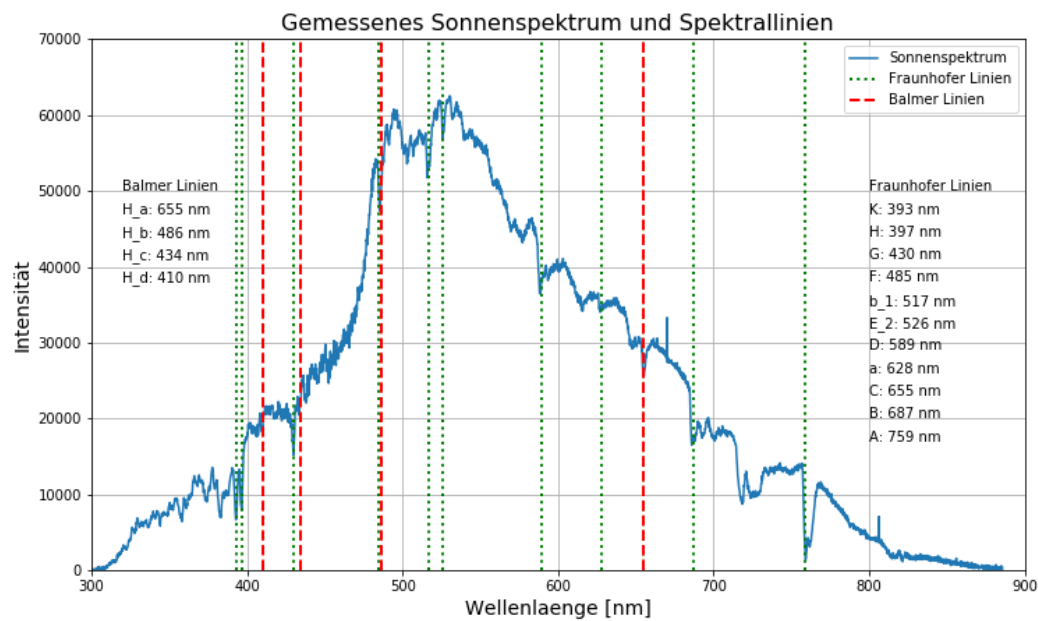


Diagramm 5: Sonnenspektrum mit Fraunhofer und Balmer Linien

Wir tragen die von uns gefundenen Linien mit den Literaturwerten (Wagner, 2022) in folgender Tabelle auf. Eine genauere Angabe der Nachkommastellen als um 1 nm finden wir nicht sinnvoll, denn durch genaues reinzoomen haben wir keinen Unterschied gemerkt. Ergebnisse wären möglicherweise vom Wissen der Lage der Literaturwerte verfälscht, also sind all die graphisch bestimmte Werte bis auf  $\Delta\lambda = 1$  nm angegeben. Im Fall der *D*-Linie erkennen wir keinen Unterschied.

Tabelle 1: Fraunhofer Linien

Linie	$\lambda_{\text{lit}}$ [nm]	$\lambda_{\text{gr}}$ [nm]
A	759,4	759
B	686,7	687
C	656,3	655
a		628
$D_1$	589,6	-
$D_2$	589	589
$D_3$	587,6	-
$E_2$	527,0	526
$b_1$	518,4	517
F	486,1	485
G	430,8	430
H	396,8	397
K	393,4	393

Für die Balmer-Serie erhalten wir

Tabelle 2: Balmer Serie

Linie	$\lambda_{\text{lit}}$ [nm]	$\lambda_{\text{gr}}$ [nm]
$H_{\alpha}$	656,3	655
$H_{\beta}$	486,1	486
$H_{\gamma}$	434,0	434
$H_{\delta}$	410,1	410

Die Fehlern der graphisch bestimmen Wellenlängen schätzen wir als  $\Delta\lambda_{\text{gr}} = 1 \text{ nm}$ . Da die Wellenlängen im Praktikumsskript auf die erste Nachkommastelle gerundet wurden, fügen wir hier zusätzlich einen Fehler von  $\Delta\lambda_{\text{lit}} = 0,5 \text{ nm}$  hinzu.

### 4.2.3 Natriumspektrum

Zur Auswertung des Natriumspektrums liegen verschiedene Messdaten zu Verfügung: Einerseits haben wir das Spektrum bei einer geringen Intensität, gezoomt auf die Wellenlängen  $\lambda \in [300, 500] \text{ nm}$ . Hier finden wir nach Angaben des Videos die "schwachen" Linien, allerdings ist es nicht klar, welche überhaupt die schwachen Linien sind, also haben wir vorsichtshalber alle auf Diagramm 6 gekennzeichnet. Auf Diagramm 7 sind die schwachen Linien im Bereich  $\lambda \in [600, 870] \text{ nm}$ . Anschließend ist unter jedem Diagramm eine Tabelle wo wir wieder die gefundenen Länge aufschreiben.

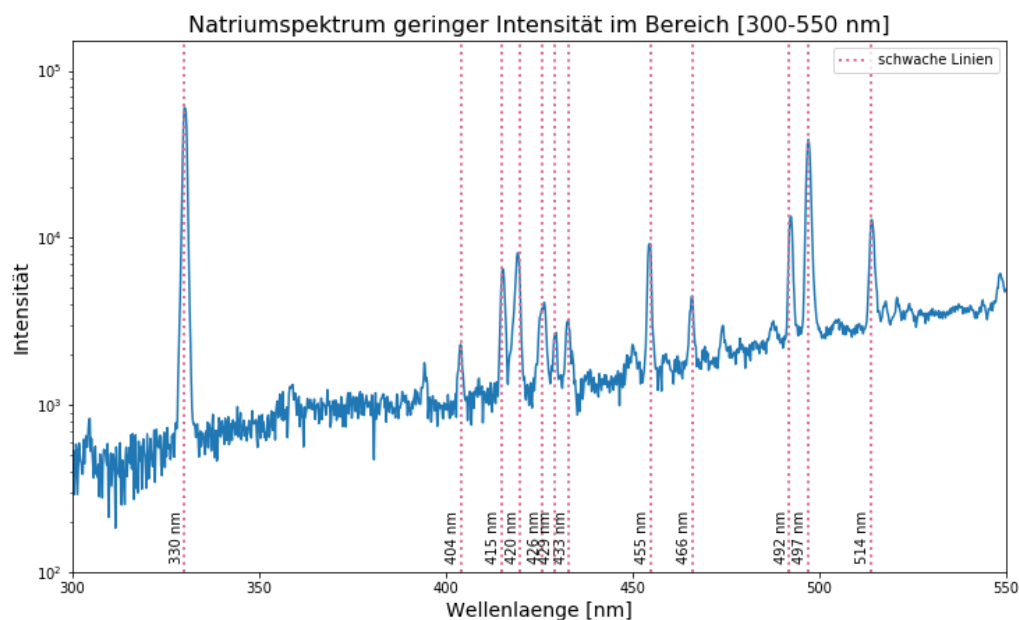
Diagramm 6: Natriumspektrum geringer Intensität  $\lambda \in [300, 500] \text{ nm}$

Tabelle 3: Wellenlänge  $\lambda \in [300, 500]$  nm

#	1	2	3	4a	4b	4c	5	6	7	8	9
$\lambda$ [nm]	330	415	420	426	429	433	455	466	492	497	514

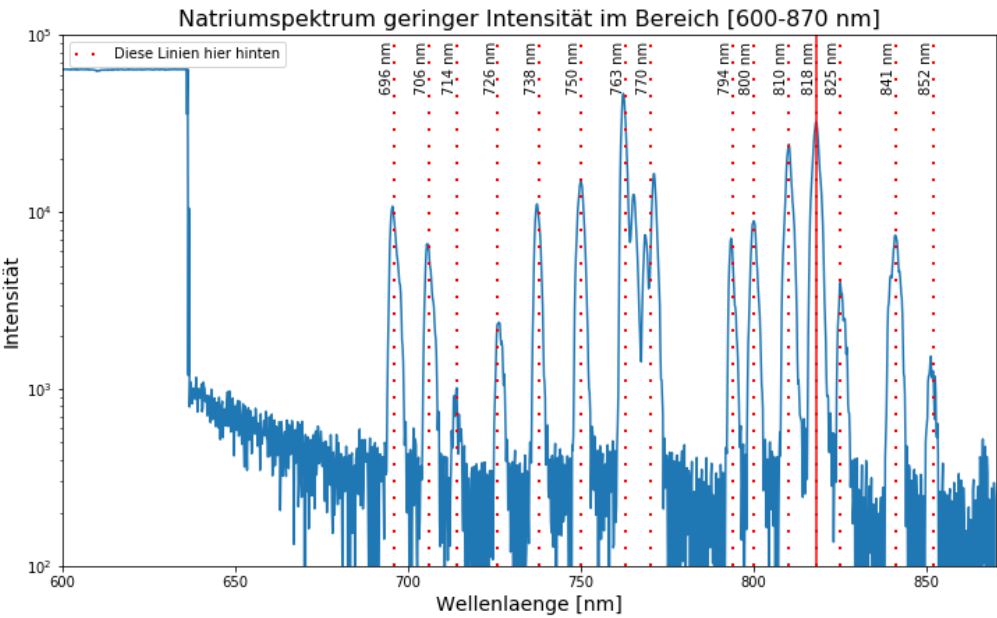
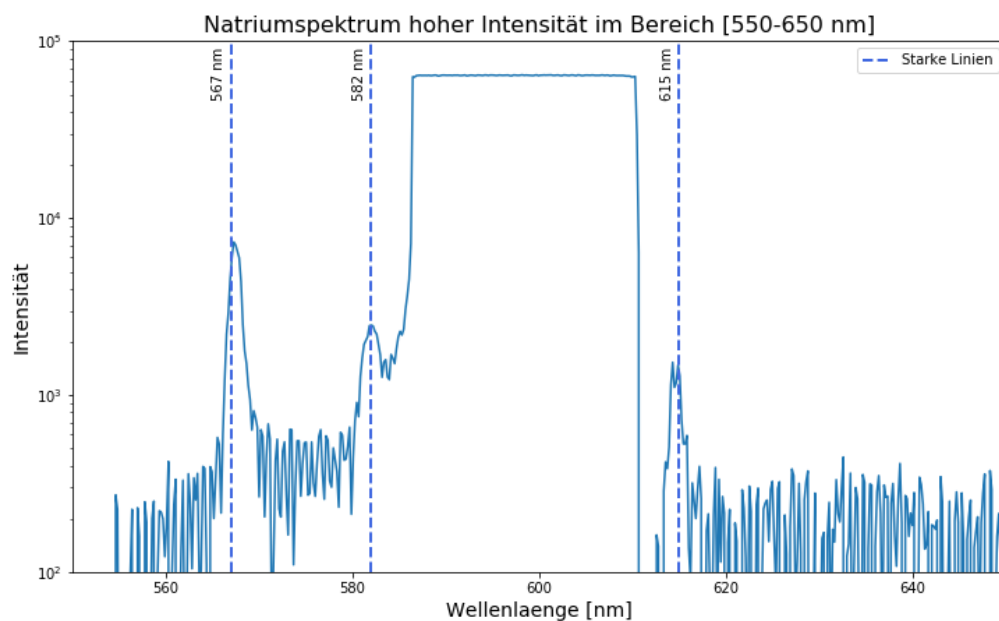


Diagramm 7: Natriumspektrum geringer Intensität  $\lambda \in [600, 870]$  nm

Tabelle 4: Wellenlänge  $\lambda \in [600, 870]$  nm

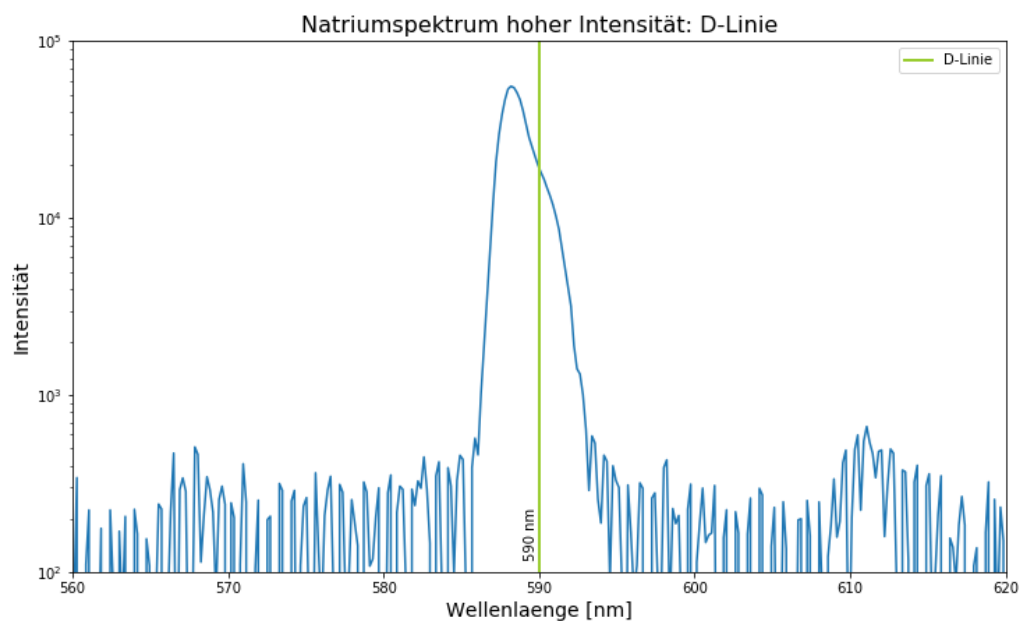
#	10	11	12	13	14	15	16	17	18	19	20	21	22	23
$\lambda$ [nm]	696	706	714	726	738	763	770	794	800	810	818	825	841	852

Auf Diagramm 8 befinden sich die als im Video bezeichneten "starke Linien" gekennzeichnet.

Diagramm 8: Natriumspektrum hoher Intensität  $\lambda \in [550, 650]$ Tabelle 5: Wellenlänge  $\lambda \in [550, 650]$  nm

#	24	25	26
$\lambda$ [nm]	567	582	615

Schließlich finden wir eine Datei wo die  $D$ -Linie ungesättigt zu erkennen ist, diese ist tatsächlich drei verschiedene Linien in einem langen Mantel verkleidet, weshalb sich nicht erkennen lässt wo die eine aufhört und die andere anfängt. Wir haben zu diesen Zwecken nur eine Linie drauf markiert und berücksichtigen einen viel größeren Fehler. Abgeschätzt ist die Halbwertsbreite hier etwa  $\text{FWHM} = 10 \text{ nm}$ , woraus ein Fehler von aufgerundet  $\Delta\lambda = 5 \text{ nm}$ .

Diagramm 9: Natriumspektrum bei ungesättigter *D*-LinieTabelle 6: Wellenlänge *D*-Linie

#	27
$\lambda$ [nm]	590
$\Delta\lambda$ [nm]	5

Zusammenfassend befindet sich auf Diagramm 10 und das gesamte Natriumspektrum im Bereich  $\lambda \in [300, 870]$  nm und die von uns in den letzten 4 Diagramme gefundenen Linien sind darauf überlagert. Anhand dieses Diagramms sollte man die starken Linien erkennen und darauf kennzeichnen, aber wir können hier beobachten, dass tatsächlich alle von uns gefundenen "schwachen" Linien sehr gut erkennbar sind. Zusätzlich zu den hier dargestellten Diagrammen waren unter den Daten 3 weitere Spektren, die aber nur eine schlechtere Auflösung dieses letzten sind, also haben wir hier nicht berücksichtigt.

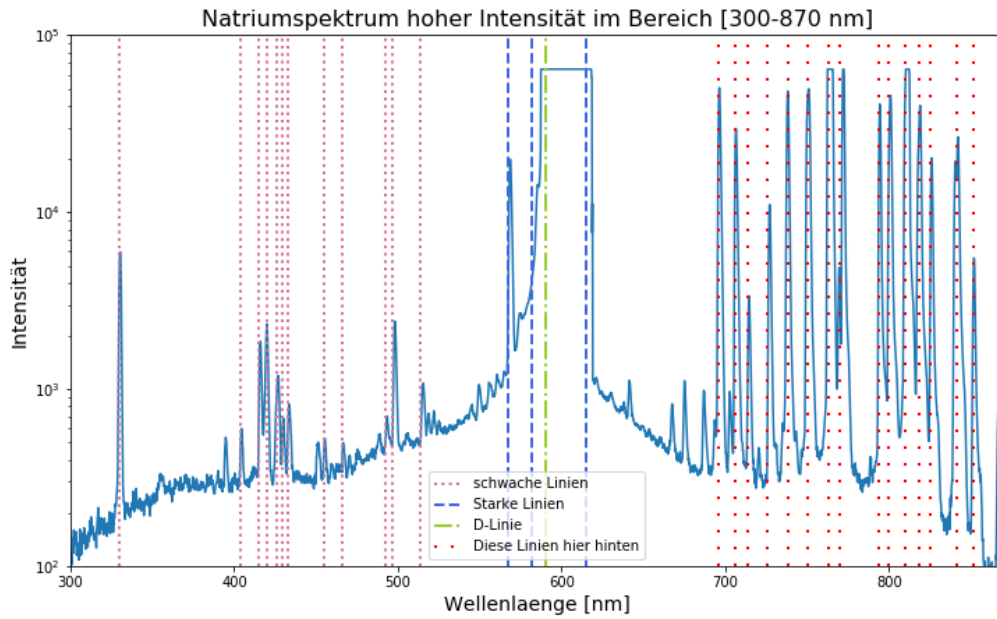


Diagramm 10: Natriumspektrum hoher Intensität mit überlagerten gefundenen Linien im Bereich  $\lambda \in [300, 870]$  nm

#### 4.2.4 Zuordnung der gefundenen Linien zu Serien

Die von uns manuell gefundenen Peaks wären gerne mit ihren Brüdern und Schwestern der selben Serie wieder zusammen. Dafür wollen wir aus verschiedenen Serien die erwarteten Wellenlängen ausrechnen und ordnen die von uns gefundenen Peaks einer passenden Linie zu.

##### Erste Nebenserie $md \rightarrow 3p$

Die erste Nebenserie erfolgt beim Übergang  $md \rightarrow 3p$ . Wir nehmen an dass der Korrekturterm für die  $d$ -Energieniveaus vernachlässigbar ist und somit lässt sich für die Wellenlänge  $\lambda_m$  der einzelnen Übergänge wie folgt berechnen:

$$\lambda_m = \frac{hc}{\frac{E_{\text{Ry}}}{m^2} - E_{3p}} \quad (7)$$

mit der Ordnung  $m$ , der Planck-Konstante  $h$ , der Lichtgeschwindigkeit  $c$ , der Rydberg-Energie  $E_{\text{Ry}}$  und der Energie des  $3p$  Zustands  $E_{3p}$ . Wir ordnen der Linie im Bereich  $\lambda = 819$  nm der Ordnung  $m = 3$  zu (auf Diagramm 7 mit einer dicken roten Linie markiert) und dadurch lässt sich  $E_{3p}$  berechnen als

$$E_{3p} = \frac{E_{\text{Ry}}}{3^2} - \frac{hc}{\lambda_3} = -3,027(2) \text{ eV}. \quad (8)$$



Hiermit berechnen wir die Wellenlänge für diesen Übergang für die Quantenzahlen  $m \in \{3, 12\}$ :

Tabelle 7: Wellenlänge für Übergang  $md \rightarrow 3p$

m	3	4	5	6	7	8	9	10	11	12
$\lambda_m$ [nm]	818	569,5	499,3	468,0	450,9	440,5	433,6	428,8	425,3	422,7
$\Delta\lambda$ [nm]	1	0,5	0,54	0,3	0,3	0,3	0,3	0,3	0,3	0,3
#	20	24	8	6			4c	4b	4	3
$\lambda_{\#}$ [nm]	818	567	497	466			433	429	425	420
$\Delta\lambda_{\#}$ [nm]	1	1	1	1			1	1	1	1

Mit einiger Kreativität könnte man für die Linie bei  $m = 7$  einen dicken kleinen Peak erkennen, aber dies hätten wir mit Sicherheit dem Rauschen zugeordnet. Für  $m = 8$  ist überhaupt kein Peak zu erkennen

### Zweite Nebenserie $ms \rightarrow 3p$

Für die zweite Nebenserie fahren wir analog fort und benutzen die  $D$ -Linie bei  $\lambda = 590$ , zugehörig zum Übergang von  $3p \rightarrow 3s$  und bestimmen die Energie  $E_{3s}$  des Grundzustands. Wir berechnen diese Energie gemäß

$$E_{3s} = E_{3p} - \frac{hc}{\lambda} = -5,129(18) \text{ eV}. \quad (9)$$

Zur Berechnung der Wellenlängen benötigen wir an dieser Stelle einen Korrekturterm für die Quantenzahl  $m$ . Dieser lässt sich durch die Energie  $E_{3s}$  ausrechnen laut

$$\Delta_s = 3 - \sqrt{\frac{E_{\text{Ry}}}{E_{3s}}} = 1,371(3). \quad (10)$$

Die Wellenlängen werden ebenfalls analog ausgerechnet, nur an dieser Stelle berücksichtigen wir den Korrekturfaktor

$$\lambda_m = \frac{hc}{\frac{E_{\text{Ry}}}{(m-\Delta_s)^2} - E_{3p}}. \quad (11)$$

Die Wellenlängen für  $m \in \{4, 9\}$  sind ebenfalls auf folgender Tabelle aufgelistet und eine graphisch bestimmte Linie wurde diesen zugeordnet.

Tabelle 8: Wellenlänge für Übergang  $ms \rightarrow 3p$ 

m	4	5	6	7	8	9
$\lambda_m$ [nm]	1171	621,7	518,2	477,2	456,2	443,8
$\Delta\lambda$ [nm]	5	0,8	0,4	0,3	0,3	0,3
#		26	9		5	
$\lambda_{\#}$ [nm]		615	514		455	
$\Delta\lambda_{\#}$ [nm]		1	1		1	

Bei der zweiten Nebenserie erkennen wir deutlich weniger Peaks, und diese sind ziemlich weit von den gefundenen entfernt.

### Hauptserie $mp \rightarrow 3s$

Wir berechnen analog den Korrekturfaktor  $\Delta_p$

$$\Delta_p = 3 - \sqrt{\frac{E_{\text{Ry}}}{E_{3p}}} = 0,8801(6) \quad (12)$$

und bestimmen für  $m \in \{4, 5\}$  die zugehörigen Wellenlängen gemäß

$$\lambda_m = \frac{hc}{\frac{E_{\text{Ry}}}{(m-\Delta_p)^2} - E_{3s}} \quad (13)$$

Tabelle 9: Wellenlänge für Übergang  $mp \rightarrow 3s$ 

m	4	5
$\lambda_m$ [nm]	332,3	286,5
$\Delta\lambda$ [nm]	1,6	1,2
#	1	
$\lambda_{\#}$ [nm]	330	
$\Delta\lambda_{\#}$ [nm]	1	

Die letzte Wellenlänge für  $m = 5$  befindet sich außerhalb des betrachteten Bereiches.

#### 4.2.5 Bestimmung der Serienenergien und $l$ -abhängigen Korrekturfaktoren

Als nächstes wollen wir unsere Ergebnisse bezüglich der Rydberg-Energie  $E_{\text{Ry}}$ , der Energie  $E_{3p}$  und der Korrekturen  $\Delta_p$  und  $\Delta_s$  überprüfen, indem wir die von uns graphisch bestimmte, zugeordnete Wellenlängen (#) gegen ihre Quantenzahl  $m$  auftragen und darüber eine Anpassung bezüglich  $m$  für ihre zugehörige Wellenlängenfunktion durchführen. Bei-

spielsweise passen wir die Funktion

$$\lambda(m) = \frac{hc}{\frac{E_{\text{Ry}}}{(m-\Delta_d)^2} - E_{3p}} \quad (14)$$

auf unsere Messdaten aus Tabelle 8 an für den Übergang  $md \rightarrow 3p$ .

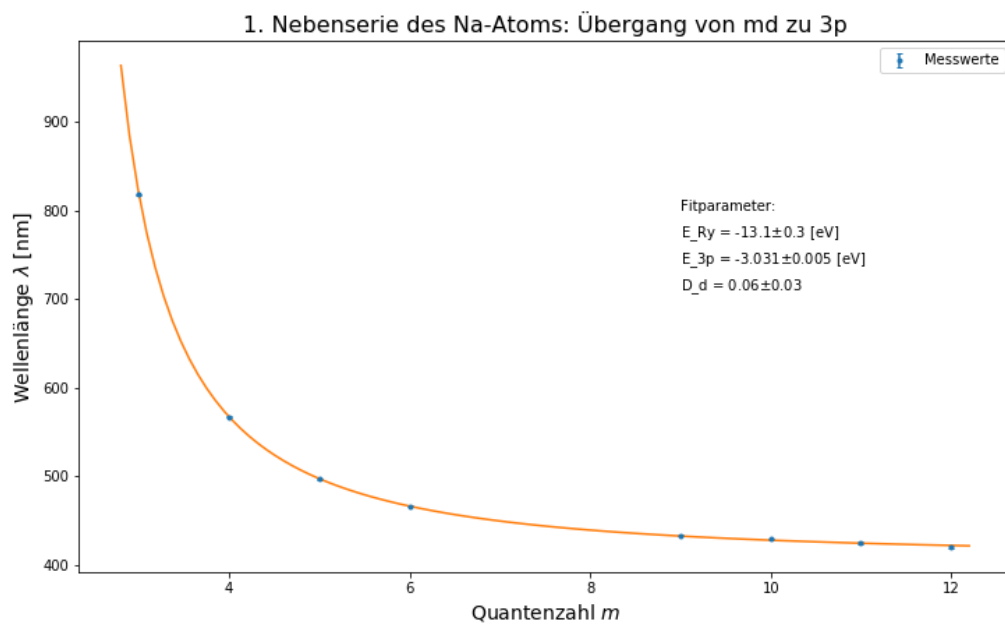


Diagramm 11: Wellenlänge als Funktion der Quantenzahl für 1. Nebenserie

Aus der Information dieses Diagramms entnehmen wir die Rydberg-Energie  $E_{\text{Ry}}$ , die Serienenergie  $E_{3p}$  und der Korrekturterm  $\Delta_d$ :

$$E_{\text{Ry},1} = -13,1(3) \text{ eV}, \quad (15)$$

$$E_{3p,1} = -3,031(5) \text{ eV}, \quad (16)$$

und

$$D_d = 0,06(3). \quad (17)$$

Die Güte dieses Fits sowie all die Vergleiche werden in der Diskussion tiefer untersucht.

Für die zweite Nebenserie plotten wir diese Abhängigkeit wieder, dieses Mal mit dem Unterschied, dass wir nicht die Korrektur  $\Delta_d$  sondern  $\Delta_s$  bestimmen.

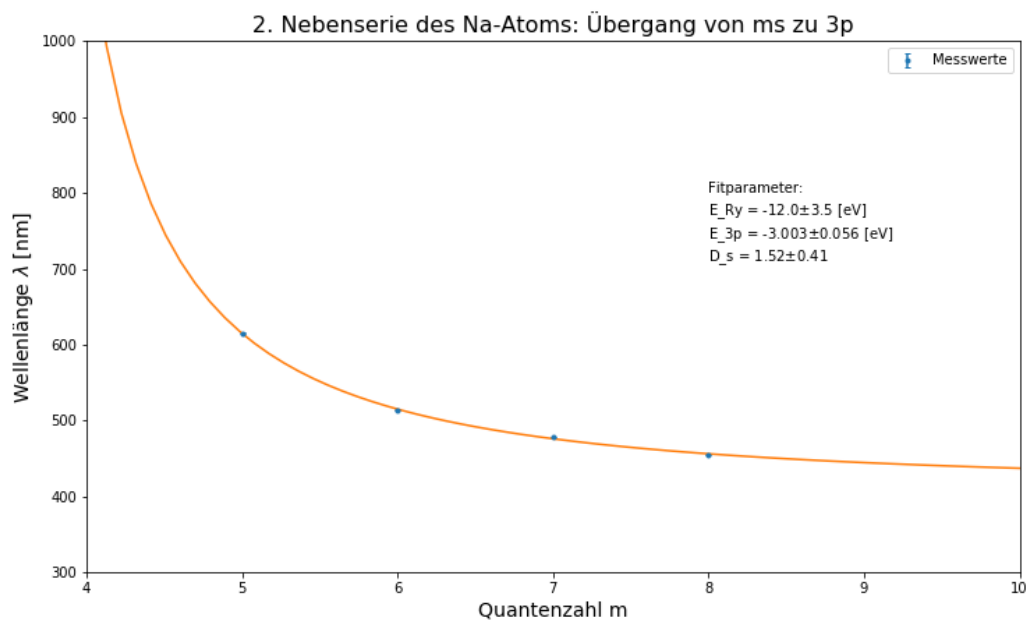


Diagramm 12: Wellenlänge als Funktion der Quantenzahl für 2. Nebenserie

Aus diesem Diagramm entnehmen wir folgende Werte:

$$E_{\text{Ry},2} = -12,0(3,5) \text{ eV}, \quad (18)$$

$$E_{3p,2} = -3,00(6) \text{ eV} \quad (19)$$

und

$$\Delta_s = 1,5(4). \quad (20)$$

## 5 Zusammenfassung und Diskussion

### 5.1 Zusammenfassung

In diesem Versuch wurde, wie der Name gewiss verrät, das Spektrum unterschiedlicher Lichtquellen untersucht. Tiefgründig wurde dies anhand des Sonnenlichts und anhand einer Natrium-Dampflampe gemacht, aber auch qualitativ für verschiedene Lichtquellen wie LED's, Glühlampen und andere Arten von Lampen.

Als erstes wurde das Sonnenspektrum analysiert, indem man mit einem offenen Fenster und einem geschlossenen das Himmelslicht vom Spektrometer aufgenommen wurde. Bei

der Durchführung schien allerdings nicht die Sonne und war eher wölkgig, mögliche Effekte werden bald bei der Diskussion untersucht werden.

Zunächst wurden die folgenden Lichtquellenspektren und ihre Spektren beobachtet:

- Glühlampe
- Halogenlampe
- Energiesparlampe
- verschiedene LED's
- Laser.

Hierzu wurde das Spektrum aufgenommen.

Schließlich wurde das Spektrum der Natriumdampflampe untersucht. Insbesondere wurde bei niedriger Intensität das Spektrum beobachtet und auf verschiedene Bereiche gezoomt, um die verschiedenen Abschnitte des sichtbaren Lichts und ihre Peaks gut erkennbar wären. Zusätzlich wurde auch bei hoher Intensität auf die intensiven Peaks geschaut.

## 5.2 Diskussion

Für die Diskussion werden wir zusammenfassend unsere Ergebnisse angeben und die relevanten Vergleiche für eine Größe  $G$  gemäß

$$\frac{|G_1 - G_2|}{\sqrt{(\Delta G_1)^2 + (\Delta G_2)^2}} \quad (21)$$

ausrechnen.

### 5.2.1 Vergleich verschiedener Lichtquellen

Zum Vergleich der verschiedenen Lichtquellen lässt sich nur noch wenig sagen, aber wir wollen kurz auf dieses interessante Phänomen eingehen, was auf Diagramm 2 auftaucht, und zwar der kleine Peak bei allen LEDs um die Wellenlänge 670 nm. Eine mögliche Hypothese wäre eine Spektrallinie aus dem Element woraus die Kristalle der LEDs bestehen. Nach unserer Recherche bestehen die meisten LEDs aus Galliumarsid. Wir haben zu diesem Zweck die starken Spektrallinien von Gallium (NIST, o.D.-a) nachgeschaut, aber keine davon befand sich in diesem Bereich. Die nächsten befanden sich um die 640 nm und waren nicht besonders stark. Dies war ebenfalls der Fall für Arsen (NIST, o.D.-b). Eine weitere Hypothese, für welche unsere technische Kompetenz nicht ausreicht ist eine Art Unterschrift des Herstellers, analog wie auf Straßenkarten von Kartenherstellern

inexistente Straßen bezeichnet wurden, um sich somit vom Plagiat zu schützen, könnte diese eine Linie eine Unterschrift des Herstellers sein. Allerdings, da ich den Versuch nicht selber durchgeführt habe, und selbst dann nicht auf den Hersteller aufgepasst hätte, bleibt diese Hypothese leider unbegründet.

### 5.2.2 Sonnenspektrum

Für Sonnenlicht wurde einmal das Sonnenspektrum mit geöffnetem Fenster, auf Diagramm 3 mit blau gekennzeichnet und einmal mit geschlossenem Fenster (orange) aufgenommen. Bei einer gelungenen Messung würden wir hier nämlich zwei Sachen erwarten: a) Glas soll durchsichtig sein, also sollten die Messungen bei geöffnetem und geschlossenem Fenster im sichtbaren Bereich des Lichts nicht signifikant voneinander abweichen, b) das infrarote und ultraviolette Spektrum sollte bei der Messung mit Fenster deutlich kleiner sein als ohne Fenster. Dies lässt sich an diesem Diagramm nicht deutlich erkennen, denn wir uns nur das sichtbare Spektrum angeschaut haben. Im Video der Durchführung ist dies aber klar der Fall. Wo wir dieses Phänomen besser beobachten ist auf Diagramm 4, wo wir die Absorption von Glas gegen die Wellenlänge auftragen. Wir beobachten beim violetten bis ultravioletten Bereich eine nahezu vollständige Absorption. Im sichtbaren Bereich reduziert sich die Absorption drastisch und steigt langsam für die infraroten. Über die Messung lässt sich zumindest anhand dieser Diagrammen sagen, dass sie erfolgreich durchgeführt wurde. Unreinigkeiten und andere Lichtphänomene wie Beugung und Streuung am Glas können möglicherweise dazu geführt haben, dass die Intensität des orangenen Spektrums kleiner ist als die des blauen, weshalb wir trotz der Lichtdurchlässigkeit von Glas trotzdem eine nicht vernachlässigbare Absorption im sichtbaren Bereich erkennen.

Auf Diagramm 5 mit dem ungefilterten Sonnenspektrum wurden die Literaturwerte für die Balmer Serie und die Fraunhofer Linien identifiziert. Wir begründen die Rundung der Fehlern auf Basis eines kleinen Auflösungsvermögens des Spektrometers, mit einem Abstand von etwa 0,2 nm pro Messung<sup>1</sup>. Da die Cursor- und Zoom-Funktion im Jupyter-Kernel des KIP nicht möglich ist heißt das ein manuelles Anpassen der Bildbedingungen und ein neuer Lauf des Programms ist für jede Änderung notwendig. Durch Vorwissen der Lage der Literaturwerten wäre eine genauere Messung als bis auf 1 nm vielleicht sogar durch dieses Vorwissen beeinflusst. Auf folgender Tabelle geben wir einen Vergleich der Literaturwerten und graphisch bestimmten Werten an. Da die Literaturwerten ebenfalls gerundet sind, könnten wir einen Fehler betrachten, dies wäre aber im Vergleich zur Rundung über die graphischen Werten vernachlässigbar.<sup>2</sup>

<sup>1</sup>Insgesamt  $\approx 3650$  Messungen im Bereich  $\lambda \in [180, 886]$  nm

<sup>2</sup>Für die Linie  $a$  wurde der Vergleichswert aus (Wikipedia, 2022) genommen.

Tabelle 10: Vergleich der Fraunhofer Linien

Linie	$\lambda_{\text{lit}}$ [nm]	$\lambda_{\text{gr}}$ [nm]	$\lambda_{\text{gr}} - \lambda_{\text{lit}}$ [nm]	$\sigma$
A	759,4	759	-0,4	0,4
B	686,7	687	-0,7	0,3
C	656,3	655	-1,3	1,3
a	628,3	628	0,3	0,3
$D_1$	589,6	589	-0,6	0,6
$D_2$	589	589	0	0
$D_3$	587,6	589	1,4	1,4
$E_2$	527,0	526	-1	1
$b_1$	518,4	517	-0,3	0,3
F	486,1	485	-1,1	1,1
G	430,8	430	-0,8	0,8
H	396,8	397	0,1	0,1
K	393,4	393	- 0,4	0,4

An dieser Tabelle erkennen wir als erstes, dass all unsere Abschätzungen in positive bis sehr positive  $\sigma$ -Bereiche befinden, mit den wenigsten knapp über  $1\sigma$ . Außerdem erkennen wir, dass die meisten Differenzen zwischen graphischem Wert und Literaturwert negativ sind, weshalb meistens die graphischen Schätzen offensichtlich unterschätzt wurden. Dies ist nur nicht der Fall bei der  $a$ -Linie, welche nicht als Vergleichswert im Praktikumsskript steht, bei der  $D_3$  Linie, wo es offensichtlich der Fall sein sollte, denn wir für die  $D$ -Wellenlänge in die Mitte dieser drei geschaut haben und bei der  $H$ -Linie. Dies ist insbesondere auffällig, denn das heißt dass es nur eine bedeutende Linie gibt ( $H$ ), wo wir die Wellenlänge *überschätzt* haben. Dies könnte daran liegen, dass der Spektrometer nur leicht falsch kalibriert ist, oder dass wir konsistent einen systematischen Fehler durchgeführt haben. Bei so vielen Daten lässt sich aber schwer auf Zufall zurücklehnen.

Für die Balmer-Serie erhalten wir

Tabelle 11: Vergleich der Balmer Serie

Linie	$\lambda_{\text{lit}}$ [nm]	$\lambda_{\text{gr}}$ [nm]	$\lambda_{\text{gr}} - \lambda_{\text{lit}}$ [nm]	$\sigma$
$H_\alpha$	656,3	655	-1,3	1,3
$H_\beta$	486,1	486	-0,1	0,1
$H_\gamma$	434,0	434	0	0
$H_\delta$	410,1	410	-0,1	0,1

Hier lässt sich ebenfalls ein ähnliches Muster erkennen, wobei die Wellenlängen bezüglich der Literaturwerten unterschätzt worden sind. Immerhin erhalten wir positive  $\sigma$ -Abweichungen. In diesem Fall und im letzten hätte man bei den Abweichungen von 0 noch mehrere Nachkommastellen von Genauigkeit mitberücksichtigen können, aber an so

einer großen Menge an Daten haben wir dies nicht entscheidend empfunden.

### 5.2.3 Natriumspektrum

#### 1. Nebenserie

Am Natriumspektrum haben wir die verschiedenen Peaks bei niedriger und hoher Intensität identifiziert und sie den theoretischen Werten bzgl. der ersten und zweiten Nebenserie und der Hauptserie zugeordnet.

Tabelle 12: Vergleich der Wellenlänge für Übergang  $md \rightarrow 3p$

m	3	4	5	6	7	8	9	10	11	12
$\lambda_m$ [nm]	818	569,5	499,3	468,0	450,9	440,5	433,6	428,8	425,3	422,7
$\Delta\lambda$ [nm]	1	0,5	0,54	0,3	0,3	0,3	0,3	0,3	0,3	0,3
#	20	24	8	6			4c	4b	4	3
$\lambda_{\#}$ [nm]	818	567	497	466			433	429	425	420
$\Delta\lambda_{\#}$ [nm]	1	1	1	1			1	1	1	1
$\lambda_{\#} - \lambda_m$ [nm]	0	-2,5	-2,3	-2			-0,6	0,2	-0,3	-2,7
$\sigma$	0	2,2	2,1	1,9			0,6	0,18	0,3	2,6

Für die  $m = 3$  Linie erhalten wir selbstverständlich eine Abweichung von Null, denn wir diese als Referenz für all die weiteren theoretisch bestimmten Wellenlänge benutzt haben. Alle andere Wellenlängen liegen innerhalb eines akzeptablen  $3\sigma$ -Bereiches, aber die Genauigkeit ist hier um einiges ungenauer als beim Sonnenspektrum. Gründe dafür sind zum Einen, dass man für die Bestimmung der theoretischen Linien die *graphisch* bestimmte Wellenlänge für  $m = 3$  benutzt hat. Dies ist ein systematischer Fehler der sich bei all den weiteren Messungen ausgebreitet hat. Diese Linie sollte im Bereich von 819 nm liegen, und die wir gefunden haben war wohl 1 nm zu klein. Dies führt zu einer leicht kleineren Zustandsenergie  $E_{3p}$  (vgl. (9)) und dementsprechend größere Wellenlängen  $\lambda_m$ . Ob dieser Effekt signifikant ist benötigt eine ausführlichere Besprechung. Zum Zweck der Genauigkeit haben wir ebenfalls die theoretischen Wellenlängen für den übertriebenen Fall mit  $\lambda_3 = 820$  nm und erhalten zwischen beiden Messungen eine Abweichung von  $2\sigma$ -Bereiche bei der Wellenlänge. Für größerer Anfangswert als  $\lambda = 818$  nm werden allerdings die Abweichungen zu den zugeordneten Linien  $\lambda_{\#}$  sogar größer. Auf dieser Basis begründen wir unsere Wahl für  $\lambda_3 = 818$  nm und für die weiteren Anfangswerte für die Bestimmung der Serien.

An den Differenzen der graphischen und theoretischen Wellenlängen erkennen wir ebenfalls das selbe Muster, dass die graphischen Wellenlängen konsistent kleiner sind als die theoretischen. Dies unterstützt unsere Hypothese, dass der Spektrometer leicht falsch kalibriert ist.



## 2. Nebenserie

Wir betrachten nun die Abweichungen für die zweite Nebenserie im Übergang  $ms \rightarrow 3p$

Tabelle 13: Vergleich der Wellenlänge für Übergang  $ms \rightarrow 3p$

m	4	5	6	7	8	9
$\lambda_m$ [nm]	1171	621,7	518,2	477,2	456,2	443,8
$\Delta\lambda$ [nm]	5	0,8	0,4	0,3	0,3	0,3
#		26	9		5	
$\lambda_{\#}$ [nm]		615	514		455	
$\Delta\lambda_{\#}$ [nm]		1	1		1	
$\lambda_{\#} - \lambda_m$ [nm]		-6,7	-4,2	-	-1,2	
$\sigma$		5	4	-	1,1	

An dieser Stelle treten zum ersten Mal signifikante Abweichungen auf. Dies liegt höchstwahrscheinlich daran, dass die # Wellenlängen falsch zugeordnet worden sind. Hieran lässt sich leider nur wenig dagegen unternehmen, denn um die gegebenen Wellenlängen haben wir keine merkbare Peaks gefunden.

## Hauptserie

Schließlich betrachten wir die Hauptserie aus Tabelle 9. Hier haben wir leider nur eine Linie zuordnen können. Wir erhalten für die Differenz  $\lambda_{\#} - \lambda_m = -2,3 \text{ nm}$  und eine  $\sigma$ -Abweichung von 1,2.

### 5.2.4 Serienenergien und $l$ -Abhängigkeit der Korrekturfaktoren

Nun wird betrachtet, ob unsere Zuordnungen, wenn rückgängig auf die entsprechenden Funktionen angepasst werden, akzeptable Werte für die Serienenergien ergeben. Hierzu muss erwähnt werden, dass für die zweite Nebenserie einer der Messdaten künstlich hinzugefügt werden musste. Die Wellenlänge bei 477,2 nm wurde zum Zweck einer möglichen Anpassung in unsere Daten hinzugefügt, obwohl wir um diesen Bereich keinen Peak gefunden hatten. Dies war nötig, denn mit drei freien Parametern hätte die Anpassung mindestens 4 Messpunkten gebraucht.

Tabelle 14: Vergleich der Serienenergien und Korrekturterme

Größe	Lit. / Theo.	1. NS	2. NS	$\sigma_{\text{lit, 1. NS}}$	$\sigma_{\text{lit, 2. NS}}$	$\sigma_{\text{1. NS, 2. NS}}$
$E_{\text{Ry}}$ [eV]	-13,605	-13,1(3)	-11,9(2,7)	1,5	0,6	0,4
$\frac{\Delta E_{\text{Ry}}}{E_{\text{Ry}}}$ [%]		2,5	23			
$E_{3p}$ [eV]	-3,027(2)	-3,031(5)	-3,00(6)	0,6	0,5	0,6
$\frac{\Delta E_{3p}}{E_{3p}}$ [%]	0,06	0,18	1,5			
$\Delta_d$	0	0,06(3)		1,7		
$\Delta_s$	1,371(3)		1,5(3)		0,5	
$\Delta_p$	0,8801(6)					

Nun betrachten wir auch die Ergebnisse der  $\chi^2$  Analyse für die zwei durchgeführten Fits. Für die erste Nebenserie erhalten wir eine Fitwahrscheinlichkeit von 34%, was obwohl nicht optimal für Praktikumsstandards ziemlich akzeptabel ist. Man hätte diese wahrscheinlich mit genaueren Abschätzungen noch besser machen können, aber wir haben uns dagegen entschieden.

Die zweite Nebenserie ergibt dagegen eine Fitwahrscheinlichkeit von 6%, was schon kritisch ist. Dies lässt sich ebenfalls darauf zurückführen, dass wahrscheinlich einige Linien falsch zugeordnet worden sind.

Die Ergebnisse dieser Tabelle scheinen den Ergebnissen der  $\chi^2$ -Analyse zu widersprechen, zumindest insofern dass die Tabelle ganz akzeptable  $\sigma$ -Abweichungen liefert, zum größten Teil sogar unter 1, aber alle unter 2, aber die  $\chi^2$  Analyse ist dagegen weniger beeindruckend. Allerdings ist genau das wahrscheinlich der Grund, weshalb wir so gute  $\sigma$ -Abweichungen bekommen. Beispielsweise erhalten wir anhand des Fits an die 2. Nebenserie die kleinste Abweichung von allen für die Rydberg-Energie von 0,4 aber ebenfalls die größte absolute Differenz gegenüber des Literaturwertes. Dies liegt daran dass der Fehler der Rydberg-Energie sehr groß abgeschätzt werden musste, da nur so wenige Daten zu Verfügung standen. Wir erhalten da einen sehr großen relativen Fehler von über 20%, was bei für eine so kleine Abweichung sorgt.

Insgesamt ließen sich die erwarteten Verhältnisse anhand unserer Messdaten ziemlich gut überprüfen und man kann von einer gelungen Durchführung reden.

## 6 Quellen

Wagner, J., Universität Heidelberg (2021). Physikalisches Praktikum PAP 2.1 für Studierende der Physik B.Sc..

Jens Wagner. (2021, 8. Februar). Versuch 234 - Lichtquellen [Video]. YouTube. [https://www.youtube.com/watch?v=1b5w4vYi1vI&list=PLnD\\_bpaogbGeIrF4Wd25x-yujWv65Q05I&index=25](https://www.youtube.com/watch?v=1b5w4vYi1vI&list=PLnD_bpaogbGeIrF4Wd25x-yujWv65Q05I&index=25)

National Institute of Standards and Technology. (o. D.-a). Strong Lines of Arsenic ( As ). Physics.Nist.Gov. Abgerufen am 1. April 2022, von <https://physics.nist.gov/PhysRefData/Handbook/Tables/arsenictable2.htm>

National Institute of Standards and Technology. (o. D.-b). Strong Lines of Gallium ( Ga ). Physics.Nist.Gov. Abgerufen am 1. April 2022, von <https://physics.nist.gov/PhysRefData/Handbook/Tables/galliumtable2.htm>

Physikalisches Institut Heidelberg & Wagner, J. (o. D.). Infos zum Praktikum für Physiker im SS 2021. [physi.uni-heidelberg](https://www.physi.uni-heidelberg.de/Einrichtungen/AP/info/Corona/Corona.php). Abgerufen am 1. April 2022, von <https://www.physi.uni-heidelberg.de/Einrichtungen/AP/info/Corona/Corona.php>

Wikipedia contributors. (2022, 1. Februar). Fraunhofer lines. Wikipedia. Abgerufen am 1. April 2022, von [https://en.wikipedia.org/wiki/Fraunhofer\\_lines](https://en.wikipedia.org/wiki/Fraunhofer_lines)

# Experiment Lichtquellen

1. April 2022

## 7 Anhang

```
[1]: import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
%matplotlib inline
import numpy as np
from numpy import exp, sqrt, log, pi
from scipy.optimize import curve_fit
from scipy.stats import chi2
from scipy import odr
from scipy.integrate import quad
from scipy.signal import find_peaks
from mpl_toolkits.axes_grid1.inset_locator import inset_axes

def fehler(name, G, sig_G, G_lit, sig_G_lit):
    print(name)
    print('Relativer Fehler:      ', sig_G / G * 100)
    print('Rel. Fehler (Vergleich):', sig_G_lit / G_lit * 100)
    print('Absoluter Fehler:      ', G - G_lit)
    # print('Verhältnis:', G / G_lit)
    print('Sigma-Abweichung:      ', np.abs(G - G_lit)
          / sqrt(sig_G ** 2 + sig_G_lit ** 2), '\n')

def fehler_small(name, G, sig_G):
    print(name)
```

```

    print('Relativer Fehler: ', sig_G / G * 100)

def ergebnis(name, G, sig_G, komma, einheit):
    print(name + ' =', np.round(G, komma), '+/-' , np.round(sig_G,
    ↪komma), einheit)

def ergebnis_large(name, G, sig_G, komma, einheit):
    print(name + ' =', np.round(G, komma))
    print('+/-' .rjust(len(name) + 2), np.round(sig_G, komma), einheit)

def fitparameter(name, G, sig_G, komma, einheit):
    return name + ' = ' + str(np.round(G, komma)) + '$\pm$' + str(np.
    ↪round(sig_G, komma)) + einheit

def chi_sq(test_func, x_val, y_val, sig_y_val, popt, number):
    chi2_ = np.sum((test_func(x_val, *popt) - y_val) ** 2
                    / sig_y_val ** 2)
    dof = len(y_val) - number
    chi2_red = chi2_ / dof

    print("chi2 =", chi2_)
    print("chi2_red =", chi2_red)

    prob = np.round(1 - chi2.cdf(chi2_,dof), 2) * 100
    print("Wahrscheinlichkeit =", prob, "%\n")

```

```

[2]: def comma_to_float(valstr):
    return float(valstr.replace(',', '.'))

```

## Auswertung der unterschiedlichen Lichtquellen

```
[3]: def extract_line(data):
    file = open('data/234/' + str(data), 'r')

    for i in range(17):
        trash = file.readline()

    lamb = np.array([])
    inten = np.array([])

    for line in file:
        line = line.strip()
        line = line.split()

        lamb = np.append(lamb, comma_to_float(line[0]))
        inten = np.append(inten, comma_to_float(line[1]))

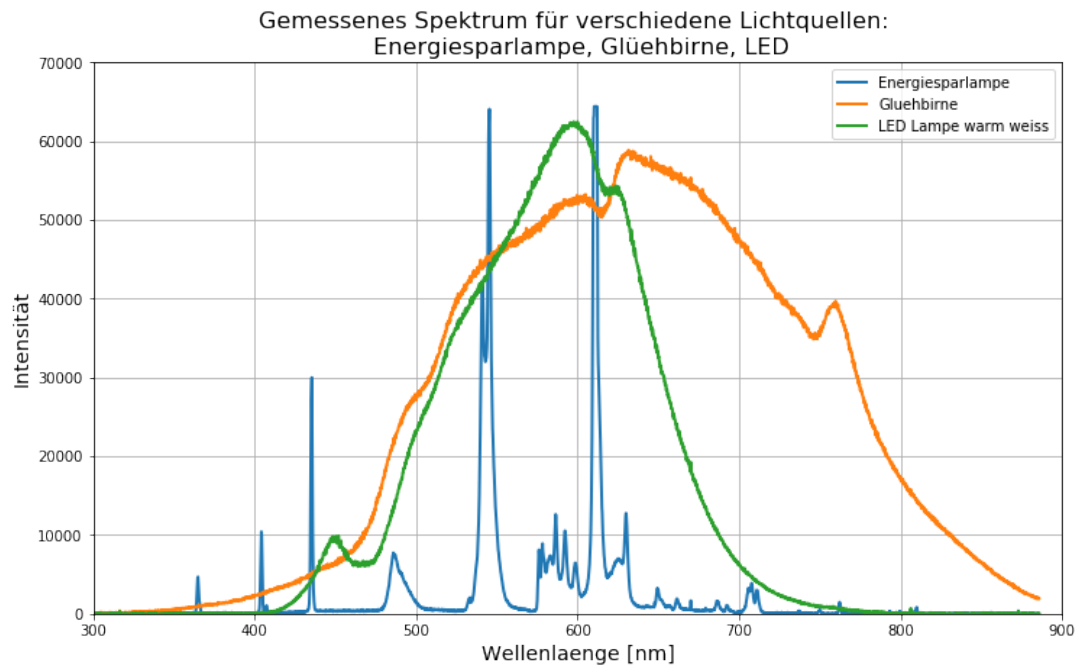
    return lamb, inten

lamb_spar, inten_spar = extract_line('Energiesparlampe.txt')
lamb_glueh, inten_glueh = extract_line('Gluehbirne.txt')
lamb_led_lww, inten_led_lww = extract_line('LEDLampewarmweiss.txt')

# Plot ES, GB, LED
plt.figure(figsize = (12, 7))
plt.plot(lamb_spar, inten_spar, label = 'Energiesparlampe', linewidth = 2)
plt.plot(lamb_glueh, inten_glueh, label = 'Gluehbirne', linewidth = 2)
plt.plot(lamb_led_lww, inten_led_lww, label = 'LED Lampe warm weiss', linewidth = 2)
plt.title('Gemessenes Spektrum für verschiedene Lichtquellen: '
          + ' \n Energiesparlampe, Glühbirne, LED', size = 16)
plt.xlabel('Wellenlaenge [nm]', size = 14)
```

```
plt.ylabel('Intensität', size = 14)
plt.legend()
plt.grid()
plt.ylim((2,70000))
plt.xlim((300,900))

plt.savefig('images/234/V234Diagramm0.png')
```



```
[4]: # Lamb, inten
lamb_blaue, inten_blaue = extract_line('LEDblau.txt')
lamb_kaltweiss, inten_kaltweiss = extract_line('LEDkaltweiss.txt')
lamb_orange, inten_orange = extract_line('LEDorange.txt')
lamb_rot, inten_rot = extract_line('LEDrot.txt')
lamb_warmweiss, inten_warmweiss = extract_line('LEDwarmweiss.txt')
lamb_warmweiss2, inten_warmweiss2 = extract_line('LEDwarmweiss2.txt')

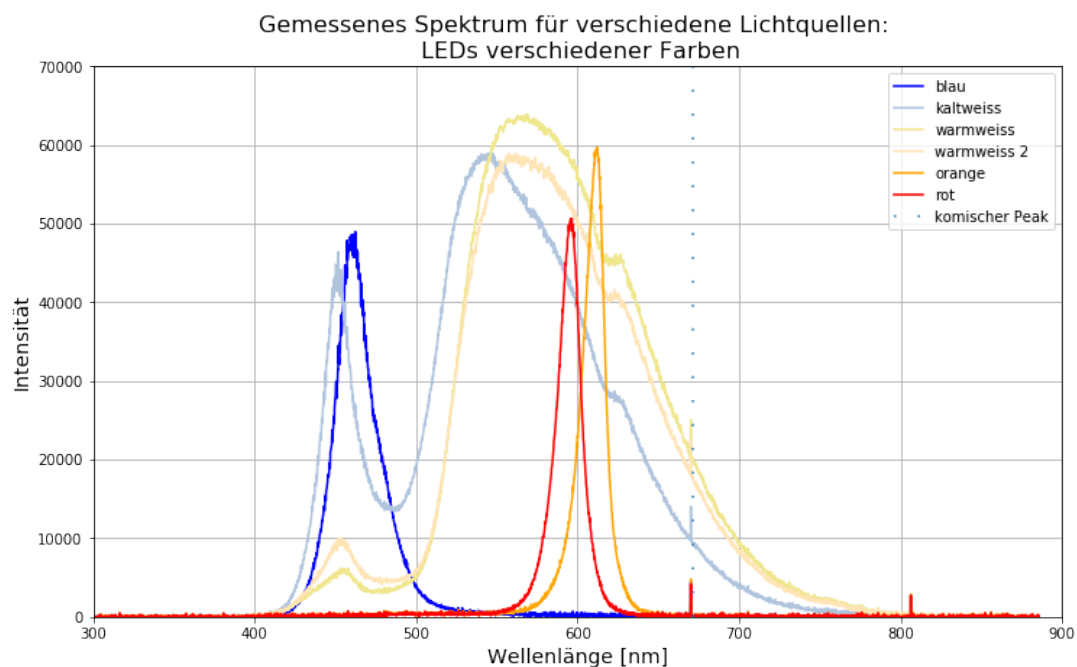
# Plot Farben
plt.figure(figsize = (12, 7))
plt.plot(lamb_blaue, inten_blaue, color = 'blue', label = 'blau')
```

```

plt.plot(lamb_kaltweiss, inten_kaltweiss, color = 'lightsteelblue',
         label = 'kaltweiss')
plt.plot(lamb_warmweiss, inten_warmweiss, color = 'khaki',
         label = 'warmweiss')
plt.plot(lamb_warmweiss2, inten_warmweiss2, color = 'moccasin',
         label = 'warmweiss 2')
plt.plot(lamb_orange, inten_orange, color = 'orange', label = 'orange')
plt.plot(lamb_rot, inten_rot, color = 'red', label = 'rot')
plt.axvline(671, label = 'komischer Peak', linestyle = (0, (1, 10)))
plt.title('Gemessenes Spektrum für verschiedene Lichtquellen:'
         + ' \n LEDs verschiedener Farben', size = 16)
plt.xlabel('Wellenlänge [nm]', size = 14)
plt.ylabel('Intensität', size = 14)
plt.legend()
plt.grid()
plt.ylim((2,70000))
plt.xlim((300,900))

plt.savefig('images/234/V234Diagramm0b.png')

```





## Auswertung des Sonnenspektrums

```
[6]: file_offen = open('data/234/untitled.txt', 'r')
file_zu = open('data/234/Fenster_geschlossen.txt', 'r')

for i in range(17): # Entferne erste 17 Zeilen
    trash = file_offen.readline()
    trash_2 = file_zu.readline()

lamb_og = np.array([])
inten_og = np.array([])

for line in file_offen: # Extract lamb und inten
    line = line.strip()
    line = line.split()

    lamb_og = np.append(lamb_og, comma_to_float(line[0]))
    inten_og = np.append(inten_og, comma_to_float(line[1]))

lamb_mg = np.array([])
inten_mg = np.array([])

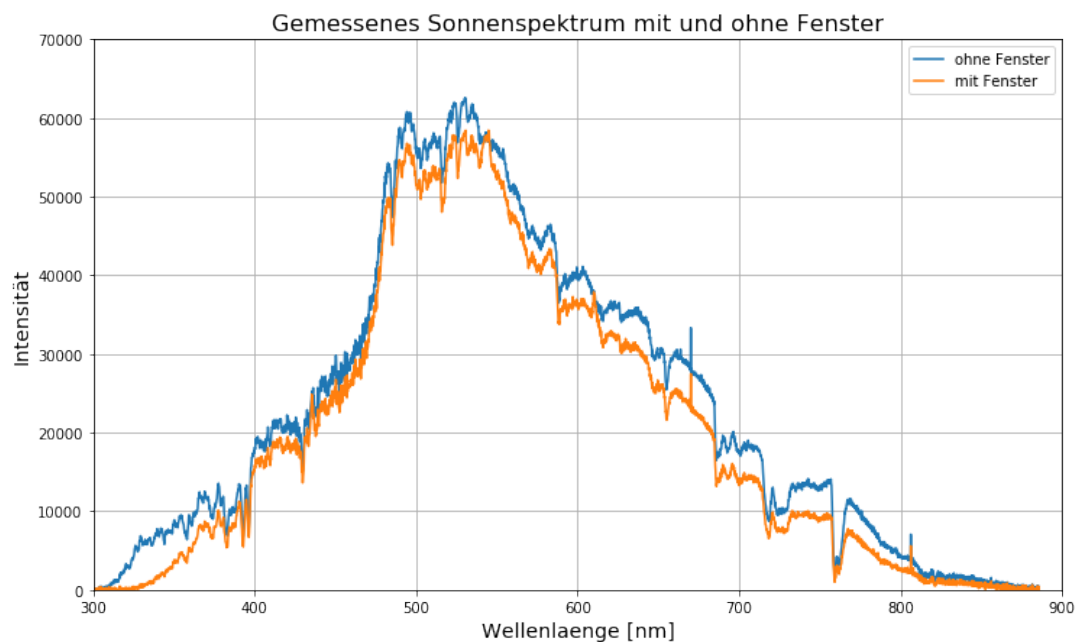
for line in file_zu:
    line = line.strip()
    line = line.split()

    lamb_mg = np.append(lamb_mg, comma_to_float(line[0]))
    inten_mg = np.append(inten_mg, comma_to_float(line[1]))
```

```
[7]: # Plot
plt.figure(figsize = (12, 7))
plt.plot(lamb_og, inten_og, label='ohne Fenster')
plt.plot(lamb_mg, inten_mg, label='mit Fenster')
```

```
plt.title('Gemessenes Sonnenspektrum mit und ohne Fenster', size = 16)
plt.xlabel('Wellenlaenge [nm]', size = 14)
plt.ylabel('Intensität', size = 14)
plt.legend()
plt.grid()
plt.ylim((2,70000))
plt.xlim((300,900))

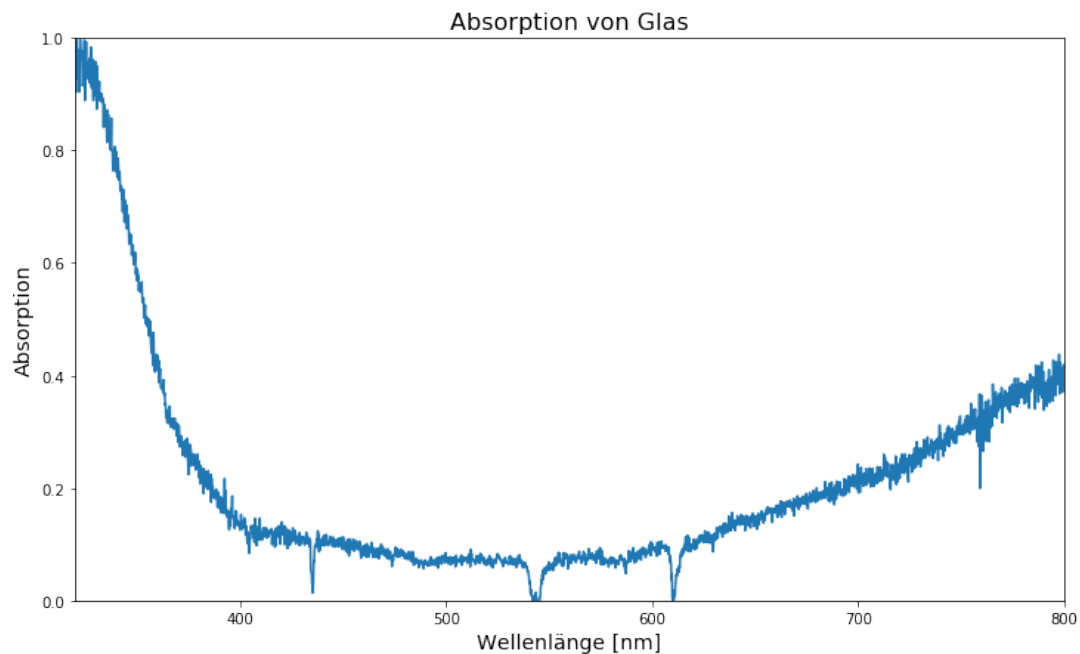
plt.savefig('images/234/V234Diagramm1.png')
```



```
[8]: # Absorption von Glas
A = 1 - inten_mg / inten_og

plt.figure(figsize = (12, 7))
plt.plot(lamb_mg, A)
plt.title('Absorption von Glas', size = 16)
plt.xlabel('Wellenlänge [nm]', size = 14)
plt.ylabel('Absorption', size = 14)
plt.ylim((0,1))
plt.xlim((320, 800))
```

```
plt.savefig('images/234/V234Diagramm2.png')
```



```
[9]: # Linien in der Sonne
def lines(serie_name, serie, col, style):
    plt.axvline(serie[0], label = serie_name, color = col,
                linestyle = style, linewidth = 2)
    for i in serie[1:]:
        plt.axvline(i, color = col, linestyle = style, linewidth = 2)

# Fraunhofer
fraunhofer_names = np.array(['K', 'H', 'G', 'F', 'b_1', 'b_2',
                              'E_2', 'D_1', 'D_2',
                              'D_3', 'a', 'C', 'B', 'A'])
fraunhofer        = np.array([393, 397, 430, 485, 517, 526, 589, 589,
                              589, 628, 655, 687, 759])
sig_lambda = 1

fraunhofer_lit = np.sort(np.array([759.4, 686.7, 656.3, 627.7, 589.
    ↪ 6, 589,
```

```

587.6, 527.0, 517.3, 486.1, 430.8,
↪396.9,
393.4]))

sig_lambda_lit = 0.1

balmer_names = np.array(['H_a', 'H_b', 'H_c', 'H_d'])
balmer = np.array([655, 486, 434, 410])

balmer_lit = np.array([656.3, 486.1, 434, 410.1])

plt.figure(figsize = (12, 7))
plt.plot(lamb_og, inten_og, label = 'Sonnenspektrum')
lines('Fraunhofer Linien', fraunhofer, 'green', ':')
plt.text(800, 50000, 'Fraunhofer Linien')
for i in range(len(fraunhofer)):
    plt.text(800, 50000 - 3000 * (i + 1), fraunhofer_names[i]
            + ': ' + str(fraunhofer[i]) + ' nm')
lines('Balmer Linien', balmer, 'red', '--')
plt.text(320, 50000, 'Balmer Linien')
for i in range(len(balmer)):
    plt.text(320, 50000 - 3000 * (i + 1), balmer_names[i]
            + ': ' + str(balmer[i]) + ' nm')

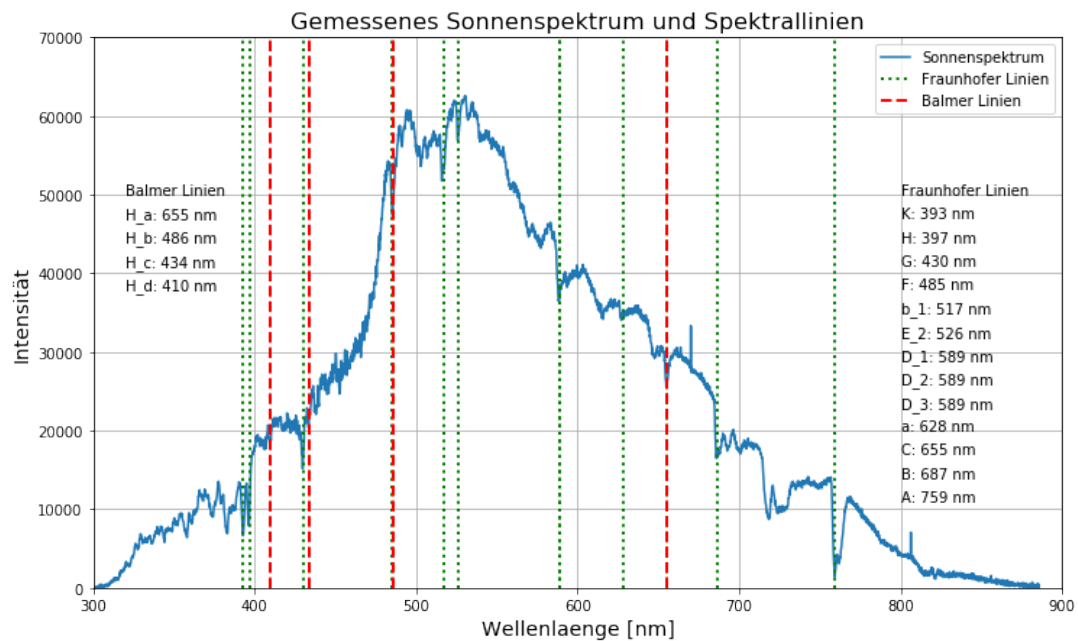
plt.title('Gemessenes Sonnenspektrum und Spektrallinien', size = 16)
plt.xlabel('Wellenlaenge [nm]', size = 14)
plt.ylabel('Intensität', size = 14)
plt.legend()
plt.grid()
plt.ylim((2,70000))

a = 300
b = 900

plt.xlim((a, b))

plt.savefig('images/234/V234Diagramm3.png')

```



## Auswertung des Natriumspektrums

```
[10]: file_na_1 = open('data/234/Na_1.txt', 'r')

lamb_na_1 = np.array([])
inten_na_1 = np.array([])

for i in range(17): # Entferne erste 17 Zeilen
    trash = file_na_1.readline()

for line in file_na_1:
    line = line.strip()
    line = line.split()

    lamb_na_1 = np.append(lamb_na_1, comma_to_float(line[0]))
    inten_na_1 = np.append(inten_na_1, comma_to_float(line[1]))

schwach = np.array([330, 404, 415, 420, 426, 429, 433, 455, 466, 492, ↵
↵497, 514])
```

```

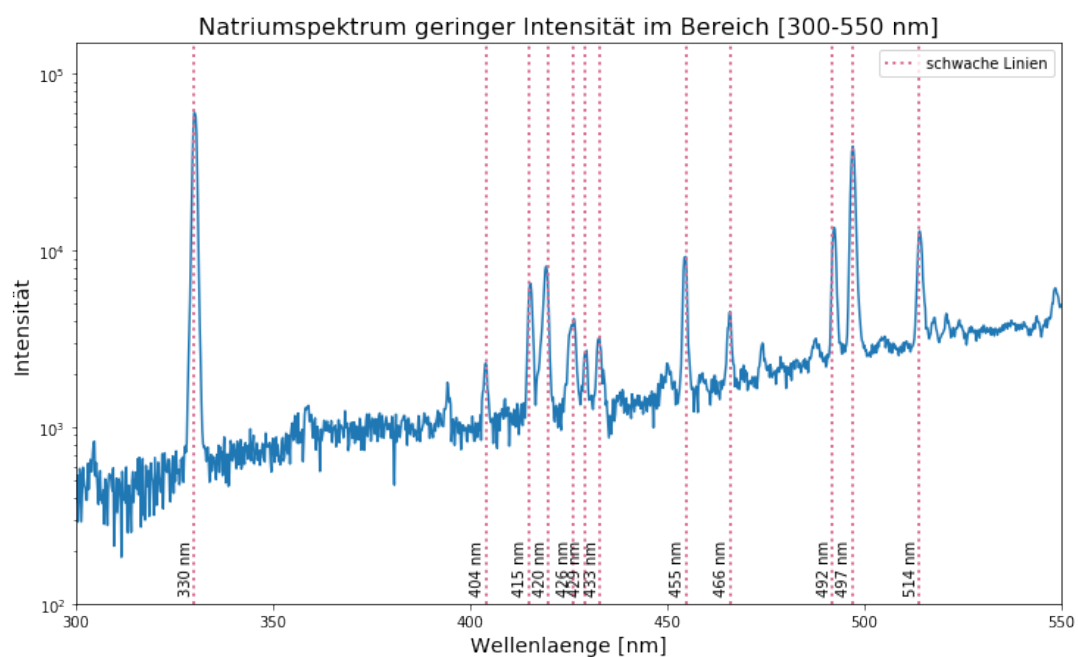
plt.figure(figsize = (12, 7))
plt.plot(lamb_na_1, inten_na_1)
lines('schwache Linien', schwach, 'palevioletred', ':')
for i in range(len(schwach)):
    plt.text(schwach[i] - 4, 2e2, str(schwach[i]) + ' nm', rotation = 90,
             ↪'vertical')

a = 300
b = 550

plt.title('Natriumspektrum geringer Intensität im Bereich ['
          + str(a) + '-' + str(b) + ' nm]', size = 16)
plt.xlabel('Wellenlaenge [nm]', size = 14)
plt.ylabel('Intensität', size = 14)
plt.yscale('log')
plt.legend()
plt.ylim((100, 1.5e5))
plt.xlim((a, b))

plt.savefig('images/234/V234Diagramm4a.png')

```



```
[11]: # Spektrum mit Zoom auf 550-650 nm, 'starke Linien'
file_na_2 = open('data/234/Na_2.txt', 'r')

lamb_na_2 = np.array([])
inten_na_2 = np.array([])

for i in range(17): # Entferne erste 17 Zeilen
    trash = file_na_2.readline()

for line in file_na_2:
    line = line.strip()
    line = line.split()

    lamb_na_2 = np.append(lamb_na_2, comma_to_float(line[0]))
    inten_na_2 = np.append(inten_na_2, comma_to_float(line[1]))

stark = np.array([567, 582, 615])

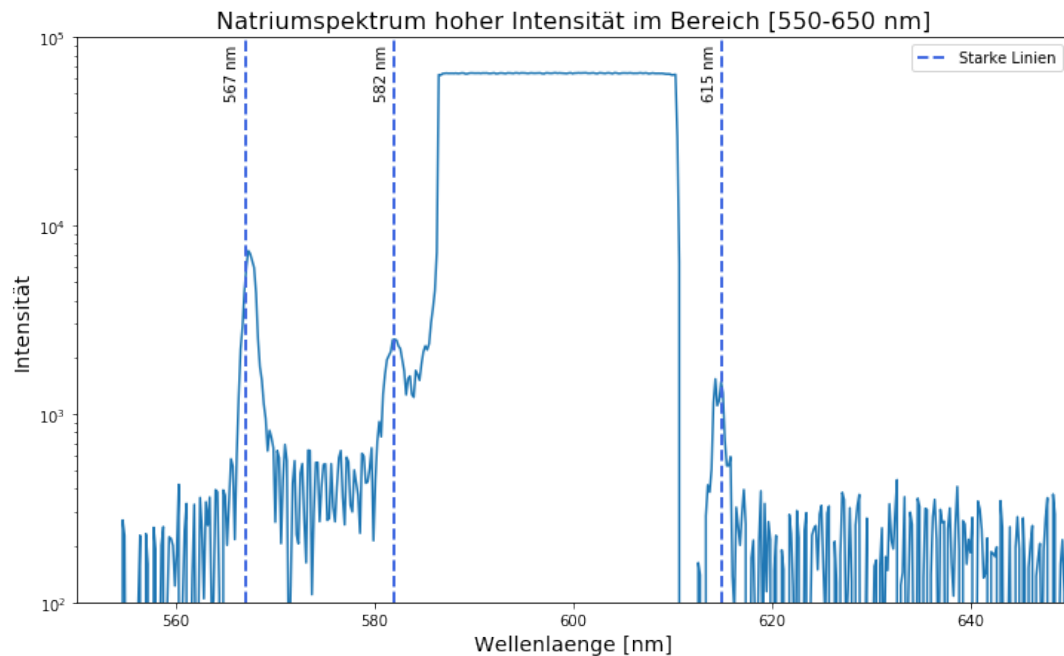
plt.figure(figsize = (12, 7))
plt.plot(lamb_na_2, inten_na_2)
lines('Starke Linien', stark, 'royalblue', '--')
for i in range(len(stark)):
    plt.text(stark[i] - 2, 8e4, str(stark[i]) + ' nm', rotation = 90,
             ↪'vertical')

a = 550
b = 650

plt.title('Natriumspektrum hoher Intensität im Bereich ['
          + str(a) + '-' + str(b) + ' nm]', size = 16)
plt.xlabel('Wellenlaenge [nm]', size = 14)
plt.ylabel('Intensität', size = 14)
plt.yscale('log')
plt.legend()
```

```
plt.ylim((100, 1e5))
plt.xlim((a, b))

plt.savefig('images/234/V234Diagramm4b.png')
```



```
[12]: file_na_3 = open('data/234/Na_3.txt', 'r')

lamb_na_3 = np.array([])
inten_na_3 = np.array([])

for i in range(17): # Entferne erste 17 Zeilen
    trash = file_na_3.readline()

for line in file_na_3:
    line = line.strip()
    line = line.split()

    lamb_na_3 = np.append(lamb_na_3, comma_to_float(line[0]))
    inten_na_3 = np.append(inten_na_3, comma_to_float(line[1]))
```

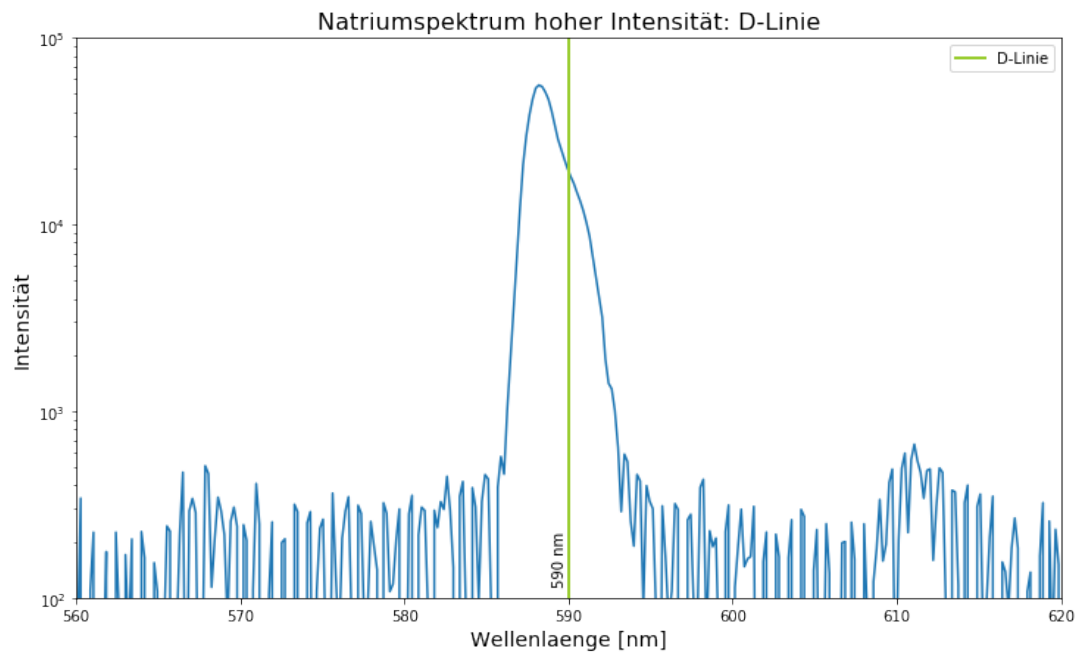


```
neben_1 = np.array([426.2, 429.41, 432.8, 449.0, 466.03, 497.33])
neben_2 = np.array([445.2, 456.32, 474.21, 514.23])
haupt = np.array([330.17, 284.15])
extra = np.array([293.8, 394.3, 404.0, 415.4, 419.46, 569])
dline = np.array([590])

plt.figure(figsize = (12, 7))
plt.plot(lamb_na_3, inten_na_3)
lines('D-Linie', dline, 'yellowgreen', '-')
for i in range(len(dline)):
    plt.text(dline[i] - 1, 2e2, str(dline[i]) + ' nm', rotation = 90,
             align='right')

plt.title('Natriumspektrum hoher Intensität: D-Linie', size = 16)
plt.xlabel('Wellenlaenge [nm]', size = 14)
plt.ylabel('Intensität', size = 14)
plt.yscale('log')
plt.legend()
plt.ylim((100, 1e5))
plt.xlim((560, 620))

plt.savefig('images/234/V234Diagramm4c.png')
```



```
[13]: file_na_4 = open('data/234/Na_4.txt', 'r')

lamb_na_4 = np.array([])
inten_na_4 = np.array([])

for i in range(17): # Entferne erste 17 Zeilen
    trash = file_na_4.readline()

for line in file_na_4:
    line = line.strip()
    line = line.split()

    lamb_na_4 = np.append(lamb_na_4, comma_to_float(line[0]))
    inten_na_4 = np.append(inten_na_4, comma_to_float(line[1]))

dieseline = np.array([696, 706, 714, 726, 738, 750, 763, 770,
                      794, 800, 810, 818, 825, 841, 852])
```

```

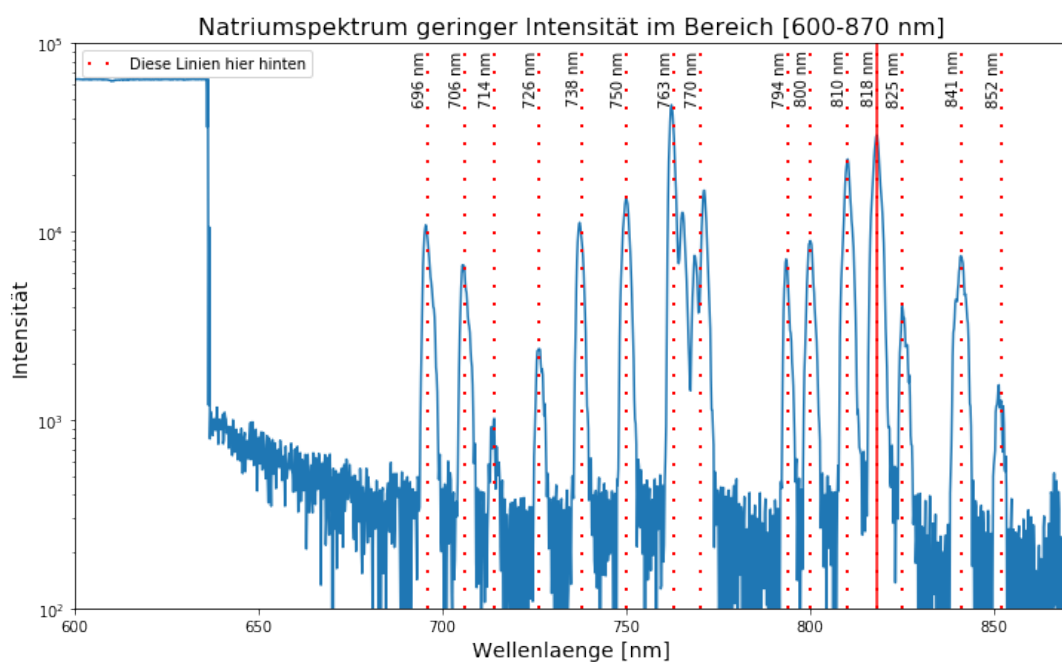
plt.figure(figsize = (12, 7))
plt.plot(lamb_na_4, inten_na_4)
lines('Diese Linien hier hinten', dieseline, 'red', (0, (1, 5)))
plt.axvline(818, color = 'red')
for i in range(len(dieseline)):
    plt.text(dieseline[i] - 4, 8e4, str(dieseline[i]) + ' nm', rotation_
↳= 'vertical')

a = 600
b = 870

plt.title('Natriumspektrum geringer Intensität im Bereich ['
        + str(a) + '-' + str(b) + ' nm]', size = 16)
plt.xlabel('Wellenlaenge [nm]', size = 14)
plt.ylabel('Intensität', size = 14)
plt.yscale('log')
plt.legend()
plt.ylim((1e2, 1e5))
plt.xlim((a, b))

plt.savefig('images/234/V234Diagramm4d.png')

```



```
[14]: file_na_622 = open('data/234/Na_622.txt', 'r')

lamb_na_622 = np.array([])
inten_na_622 = np.array([])

for i in range(17): # Entferne erste 17 Zeilen
    trash = file_na_622.readline()

for line in file_na_622:
    line = line.strip()
    line = line.split()

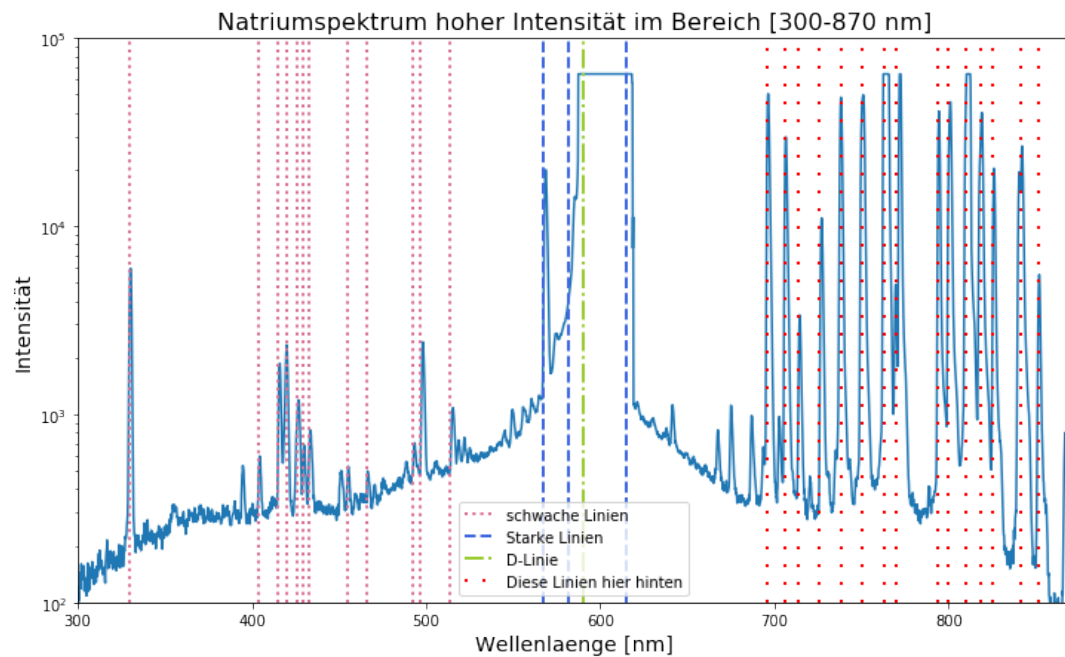
    lamb_na_622 = np.append(lamb_na_622, comma_to_float(line[0]))
    inten_na_622 = np.append(inten_na_622, comma_to_float(line[1]))

plt.figure(figsize = (12, 7))
plt.plot(lamb_na_622, inten_na_622)
lines('schwache Linien', schwach, 'palevioletred', ':')
lines('Starke Linien', stark, 'royalblue', '--')
lines('D-Linie', dline, 'yellowgreen', '-.')
lines('Diese Linien hier hinten', dieseline, 'red', (0, (1, 5)))

a = 300
b = 870

plt.title('Natriumspektrum hoher Intensität im Bereich ['
          + str(a) + '-' + str(b) + ' nm]', size = 16)
plt.xlabel('Wellenlaenge [nm]', size = 14)
plt.ylabel('Intensität', size = 14)
plt.yscale('log')
plt.legend()
plt.ylim((100, 1e5))
plt.xlim((a, b))
```

```
plt.savefig('images/234/V234Diagramm4e.png')
```



## Zuordnung der gefundenen Linien zu Serien

```
[15]: # Erwartete Linien
# Erste Nebenserie
E_Ry = - 13.605 # eV
hc = 1.2398e3 # nm eV
n = 3
l = 818 # nm
sig_l = 1

lambda_1_n = np.array([])
sig_lambda_1_n = np.array([])
q_1_n = np.array([])

# Energie m = 3 Zustand
E_3p = (E_Ry / n ** 2) - hc / l
sig_E_3p = hc * (sig_l / l ** 2)
ergebnis('E_3p', E_3p, sig_E_3p, 3, '[eV]\n')
```

```

for m in range(3,13):
    l = hc / (E_Ry / m ** 2 - E_3p)
    sig_l = hc / (E_Ry / m ** 2 - E_3p) ** 2 * sig_E_3p

    lambda_1_n = np.append(lambda_1_n, l) # Liste der Wellenlängen
    sig_lambda_1_n = np.append(sig_lambda_1_n, sig_l)
    q_1_n = np.append(q_1_n, m) # Liste der Quantenzahlen
    ergebnis('lambda_' + str(m), l, sig_l, 1, '[nm]')

# lambda_1_n: m = 7, 8 nicht gefunden, entferne für spätere sigmas
mask = np.ones(len(lambda_1_n), dtype = bool)
mask[[4, 5]] = False

lambda_1_n = lambda_1_n[mask, ...]
sig_lambda_1_n = sig_lambda_1_n[mask, ...]

```

$E_{3p} = -3.027 \pm 0.002$  [eV]

```

lambda_3 = 818.0 +/- 1.0 [nm]
lambda_4 = 569.5 +/- 0.5 [nm]
lambda_5 = 499.3 +/- 0.4 [nm]
lambda_6 = 468.0 +/- 0.3 [nm]
lambda_7 = 450.9 +/- 0.3 [nm]
lambda_8 = 440.5 +/- 0.3 [nm]
lambda_9 = 433.6 +/- 0.3 [nm]
lambda_10 = 428.8 +/- 0.3 [nm]
lambda_11 = 425.3 +/- 0.3 [nm]
lambda_12 = 422.7 +/- 0.3 [nm]

```

```

[16]: # Erste Nebenserie mit alternativer Wellenlänge (relevant in der
      ↳ Diskussion)
      l = 820 # nm

      lambda_1_n_b = np.array([])
      sig_lambda_1_n_b = np.array([])
      q_1_n_b = np.array([])

```

```

# Energie m = 3 Zustand
E_3p_b = (E_Ry / n ** 2) - hc / l
sig_E_3p_b = hc * (sig_l / l ** 2)
#ergebnis('E_3p', E_3p, sig_E_3p, 3, '[eV]\n')

for m in range(3,13):
    l = hc / (E_Ry / m ** 2 - E_3p_b)
    sig_l = hc / (E_Ry / m ** 2 - E_3p_b) ** 2 * sig_E_3p_b

    lambda_1_n_b = np.append(lambda_1_n_b, l) # Liste der Wellenlängen
    sig_lambda_1_n_b = np.append(sig_lambda_1_n_b, sig_l)
    q_1_n_b = np.append(q_1_n_b, m) # Liste der Quantenzahlen
    #ergebnis('lambda_' + str(m), l, sig_l, 1, '[nm]')

# lambda_1_n: m = 7, 8 nicht gefunden, entferne für spätere sigmas
mask = np.ones(len(lambda_1_n_b), dtype = bool)
mask[[4, 5]] = False

lambda_1_n_b = lambda_1_n_b[mask, ...]
sig_lambda_1_n_b = sig_lambda_1_n_b[mask, ...]

```

```

[17]: # Zweite Nebenserie
l_g = 590
sig_l_g = 5

lambda_2_n = np.array([])
sig_lambda_2_n = np.array([])
q_2_n = np.array([])

# Energie Grundzustand
E_3s = E_3p - (hc / l_g)
sig_E_3s = sqrt(sig_E_3p ** 2 + (hc * sig_l_g / (l_g ** 2)) ** 2)
ergebnis('E_3s', E_3s, sig_E_3s, 3, '[eV]\n')

# Korrekturterm
Ds = 3 - sqrt(E_Ry / E_3s)

```

```

sig_Ds = 0.5 * sig_E_3s * sqrt(E_Ry / (E_3s ** 3))
ergebnis('Ds', Ds, sig_Ds, 3, '\n')

for m in range(4,10):
    l = hc / (E_Ry/(m - Ds) ** 2 - E_3p)
    sig_l = sqrt((2 * E_Ry * hc / ((m - Ds) ** 3
                                * (E_Ry / (m - Ds) ** 2 - E_3p) ** 2) *
                                sig_Ds) ** 2
                + (hc / (E_Ry / (m - Ds) ** 2 - E_3p) ** 2 * sig_E_3p) ** 2)
    lambda_2_n = np.append(lambda_2_n, l)
    sig_lambda_2_n = np.append(sig_lambda_2_n, sig_l)
    q_2_n = np.append(q_2_n, m)
    ergebnis('lambda_' + str(m), l, sig_l, 2, '[nm]')

# lambda_2_n: m = 4, 7, 9 nicht gefunden, entferne für spätere sigmas
# 7 muss allerdings bleiben für Fit
mask_2 = np.ones(len(lambda_2_n), dtype = bool)
mask_2[[0, 5]] = False

lambda_2_n = lambda_2_n[mask_2, ...]
sig_lambda_2_n = sig_lambda_2_n[mask_2, ...]

```

E\_3s = -5.129 +/- 0.018 [eV]

Ds = 1.371 +/- 0.003

lambda\_4 = 1171.3 +/- 5.14 [nm]

lambda\_5 = 621.73 +/- 0.77 [nm]

lambda\_6 = 518.24 +/- 0.44 [nm]

lambda\_7 = 477.23 +/- 0.35 [nm]

lambda\_8 = 456.2 +/- 0.31 [nm]

lambda\_9 = 443.81 +/- 0.3 [nm]



```
[18]: # Hauptserie
lambda_H = np.array([])
sig_lambda_H = np.array([])
q_H = np.array([])

Dp = 3 - sqrt(E_Ry / E_3p)
sig_Dp = (3 - Dp) / 2 * np.abs(sig_E_3p / E_3p)
ergebnis('Dp', Dp, sig_Dp, 4, '\n')

for m in range(4,6):
    l = hc / (E_Ry / ((m - Dp) ** 2) - E_3s)
    sig_l = sqrt((2 * E_Ry * hc / ((m - Dp) ** 3
                                * (E_Ry / (m - Dp) ** 2 - E_3s) **
                                ↪2) *
                                sig_Ds) ** 2
                + (hc / (E_Ry / (m - Dp) ** 2 - E_3s) ** 2 * sig_E_3s) ↪
    ↪** 2 )
    lambda_H = np.append(lambda_H, l)
    sig_lambda_H = np.append(sig_lambda_H, sig_l)
    q_H = np.append(q_H, m)
    ergebnis('lambda_' + str(m), l, sig_l, 1, '[nm]')

# lambda_H: m = 5 nicht gefunden, entferne für spätere sigmas
mask_H = np.ones(len(lambda_H), dtype = bool)
mask_H[1] = False

lambda_H = lambda_H[mask_H, ...]
sig_lambda_H = sig_lambda_H[mask_H, ...]
```

Dp = 0.8801 +/- 0.0006

lambda\_4 = 332.3 +/- 1.6 [nm]

lambda\_5 = 286.5 +/- 1.2 [nm]

```
[19]: # Spektrum mit theoretischen Nebenserien
plt.figure(figsize = (12, 7))
plt.plot(lamb_na_622, inten_na_622)
```

```

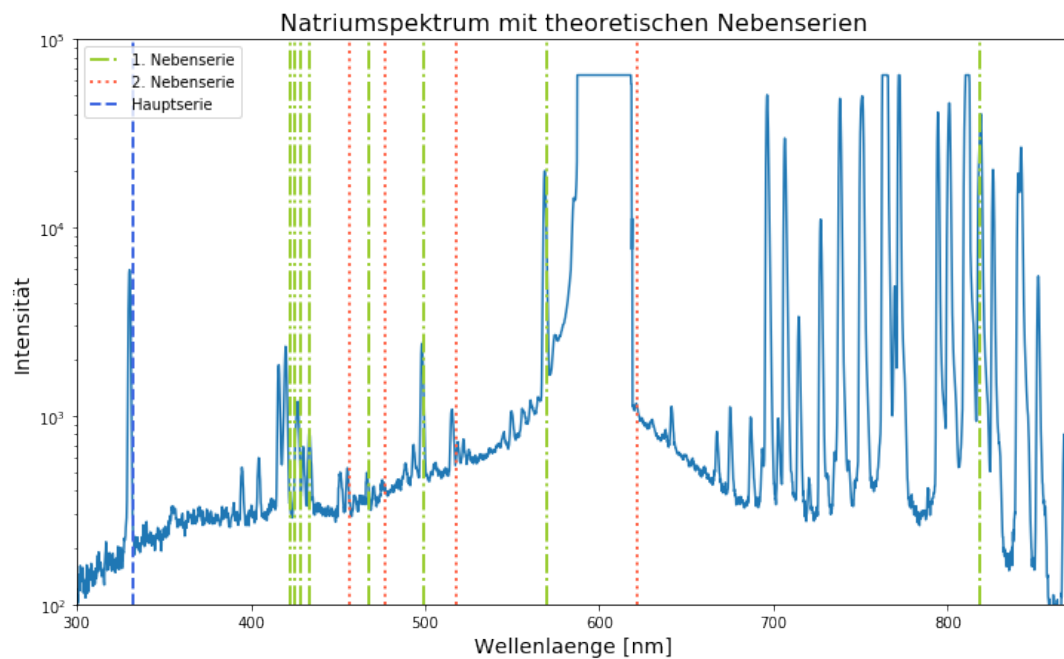
lines('1. Nebenserie', lambda_1_n, 'yellowgreen', '-.')
lines('2. Nebenserie', lambda_2_n, 'tomato', ':')
lines('Hauptserie', lambda_H, 'royalblue', '--')

a = 300
b = 870

plt.title('Natriumspektrum mit theoretischen Nebenserien', size = 16)
plt.xlabel('Wellenlaenge [nm]', size = 14)
plt.ylabel('Intensität', size = 14)
plt.yscale('log')
plt.legend()
plt.ylim((100, 1e5))
plt.xlim((a, b))

plt.savefig('images/234/V234Diagramm4f.png')

```



## Linzertorte

```
[20]: # Zutaten
ingr = np.array(['Mehl', 'Mandelmehl', 'Zimt', 'Backpulver', 'Salz',
                 'Butter', 'Brauner Zucker', 'Marmelade'])
menge = np.array([190, 50, 3, 3, 3, 125, 135, 240]) # g

# Anleitung
def ofen(temperatur):
    print('Ofen vorerhitzt auf ' + str(temperatur) + '°C\n')

def marmelade(kuelzeit):
    print('Marmelade wird bei mittlerer Hitze gekocht und ' + _
    ↪str(kuelzeit)
        + ' min kühlen gelassen\n')

def mischen(name, zutaten):
    for zutat in range(len(zutaten)):
        print(zutaten[zutat])
    print('werden in die Schüssel reingetan\n')
    return name

def kremieren(name, zutaten):
    print(zutaten[0] + ' und ' + zutaten[1]
          + ' werden brutal geschlagen bis Mischung cremig\n')
    return name

def kneten(x):
    print('Teig wird zu etwa ' + str(x) + ' cm geknetet\n')
```

```

ofen(180)
mehlmisch = mischen('Mehlmischung', ingr[0:4])
buttermisch = kremieren('Butterkreme', ingr[5:7])
mehlbuttermisch = mischen('Teig', np.array([mehlmisch, buttermisch]))
marmelade(5)
kneten(.5)

major_ticks = np.arange(-0.2, 0.2, 0.05)

fig, ax = plt.subplots()
circle2 = plt.Circle((0, 0), 0.2, color='darkmagenta')
plt.title('Bild')
ax.add_patch(circle2)
ax.set_xticks(major_ticks)
ax.set_yticks(major_ticks)
plt.grid(linewidth = 5, color = 'darkgoldenrod')
ax.set_ylim((-0.2, 0.2))
ax.set_xlim((-0.2, 0.2))
fig.show()

def backen(zeit):
    print('Kuchen wird etwa ' + zeit
          + ' min gebacken, oder bis golden braun und Marmelade_
↪blubert')

backen('30-35')

```

Ofen vorerhitzt auf 180°C

Mehl

Mandelmehl

Zimt

Backpulver

werden in die Schüssel reingetan

Butter und Brauner Zucker werden brutal geschlagen bis Mischung kremig

Mehlmischung

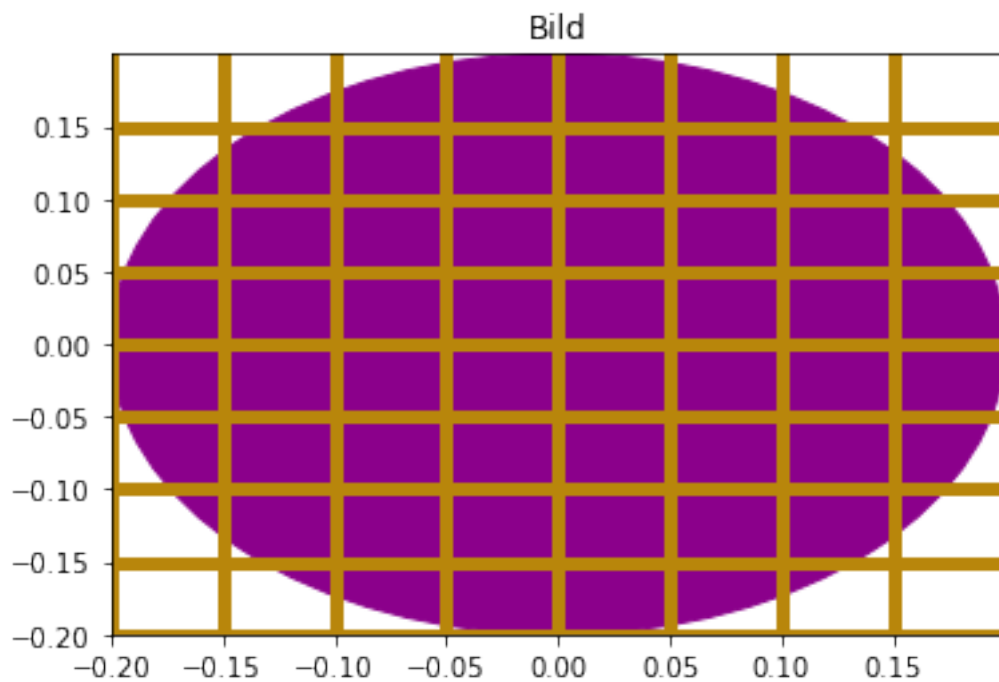
Butterkreme

werden in die Schüssel reingetan

Marmelade wird bei mittlerer Hitze gekocht und 5 min kühlen gelassen

Teig wird zu etwa 0.5 cm geknetet

Kuchen wird etwa 30-35 min gebacken, oder bis golden braun und Marmelade  
 ↳ blubert



## Bestimmung der Serienenergien und l-abhängigen Korrekturfaktoren

```
[21]: # Fit md -> 3p
m_md3p = np.array([3, 4, 5, 6, 9, 10, 11, 12])
lambda_md3p = np.array([818, 567, 497, 466, 433, 429, 425, 420])
sig_lambda_md3p = np.ones(len(lambda_md3p))
```

```

def fit_func(m, E_Ry, E_3p, D_d):
    return hc / (E_Ry / (m - D_d) ** 2 - E_3p)

p0 = [-13.6, -3, -0.02]
popt_1, pcov_1 = curve_fit(fit_func, m_md3p, lambda_md3p,
                           sigma=sig_lambda_md3p, p0=p0)

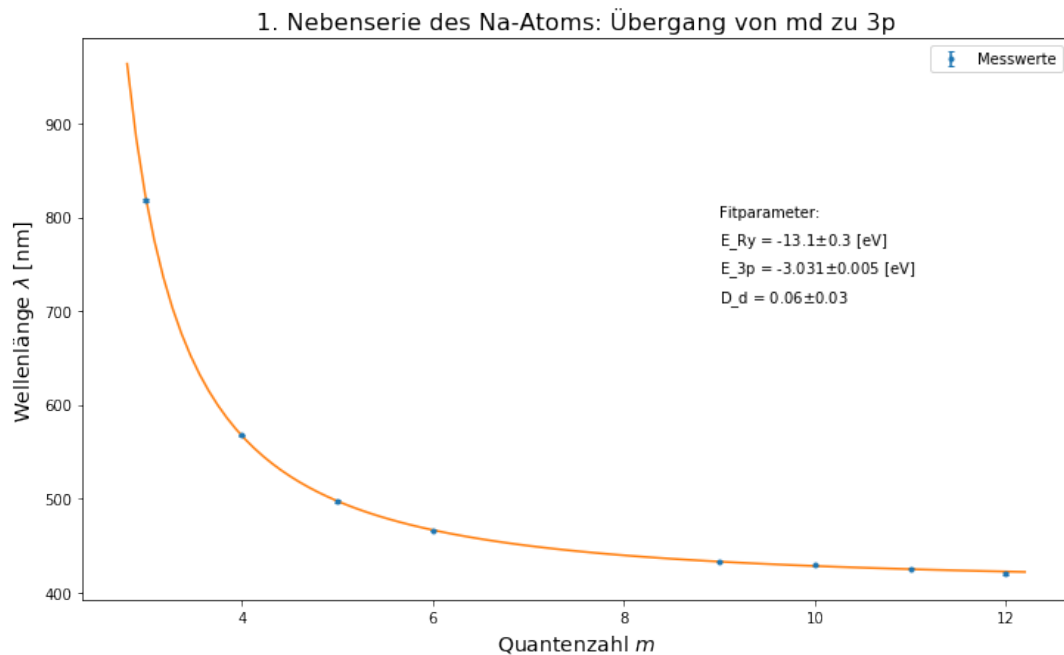
# Fitparameter
E_Ry_fit_1 = popt_1[0]
sig_E_Ry_fit_1 = sqrt(pcov_1[0, 0])
E_3p_fit_1 = popt_1[1]
sig_E_3p_fit_1 = sqrt(pcov_1[1, 1])
D_d_fit_1 = popt_1[2]
sig_D_d_fit_1 = sqrt(pcov_1[2, 2])

x=np.linspace(2.8, 12.2, 100)

# Plot
plt.figure(figsize = (12, 7))
plt.errorbar(m_md3p, lambda_md3p, yerr = sig_lambda_md3p, fmt = '.',
             ↪ capsize = 2,
             label = 'Messwerte')
plt.plot(x, fit_func(x, *popt_1))
plt.text(9, 800, 'Fitparameter:')
plt.text(9, 770, fitparameter('E_Ry', E_Ry_fit_1, sig_E_Ry_fit_1, 1, '↪
↪ [eV]'))
plt.text(9, 740, fitparameter('E_3p', E_3p_fit_1, sig_E_3p_fit_1, 3, '↪
↪ [eV]'))
plt.text(9, 710, fitparameter('D_d', D_d_fit_1, sig_D_d_fit_1, 2, ''))
plt.title('1. Nebenserie des Na-Atoms: Übergang von md zu 3p', size =
↪ 16)
plt.xlabel('Quantenzahl $m$', size = 14)
plt.ylabel('Wellenlänge $\lambda$ [nm]', size = 14)
plt.legend()

```

```
plt.savefig('images/234/V234Diagramm5a.png')
```



```
[22]: #  $\chi^2$  Analyse
chi_sq(fit_func, m_md3p, lambda_md3p, sig_lambda_md3p, popt_1, 3)
```

```
chi2 = 5.634602732918731
```

```
chi2_red = 1.1269205465837462
```

```
Wahrscheinlichkeit = 34.0 %
```

```
[23]: # Fit ms -> 3p
m_ms3p = np.array([5, 6, 7, 8])
lambda_ms3p = np.array([615, 514, 477.2, 455])
'''
Künstliches Hinzufügen der m = 7 Linie, weil sonst zu viele
↳Parameter,
nicht genug Werte
'''
sig_lambda_ms3p = np.ones(len(lambda_ms3p))

def fit_func(m, E_Ry, E_3p, D_d):
```

```

    return hc / (E_Ry / (m - D_d) ** 2 - E_3p)

p0 = [-13.6, -5, 1]
popt_2, pcov_2 = curve_fit(fit_func, m_ms3p, lambda_ms3p,
                           sigma = sig_lambda_ms3p, p0 = p0)

# Fitparameter
E_Ry_fit_2 = popt_2[0]
sig_E_Ry_fit_2 = sqrt(pcov_2[0, 0])
E_3p_fit_2 = popt_2[1]
sig_E_3p_fit_2 = sqrt(pcov_2[1, 1])
D_s_fit_2 = popt_2[2]
sig_D_s_fit_2 = sqrt(pcov_2[2, 2])

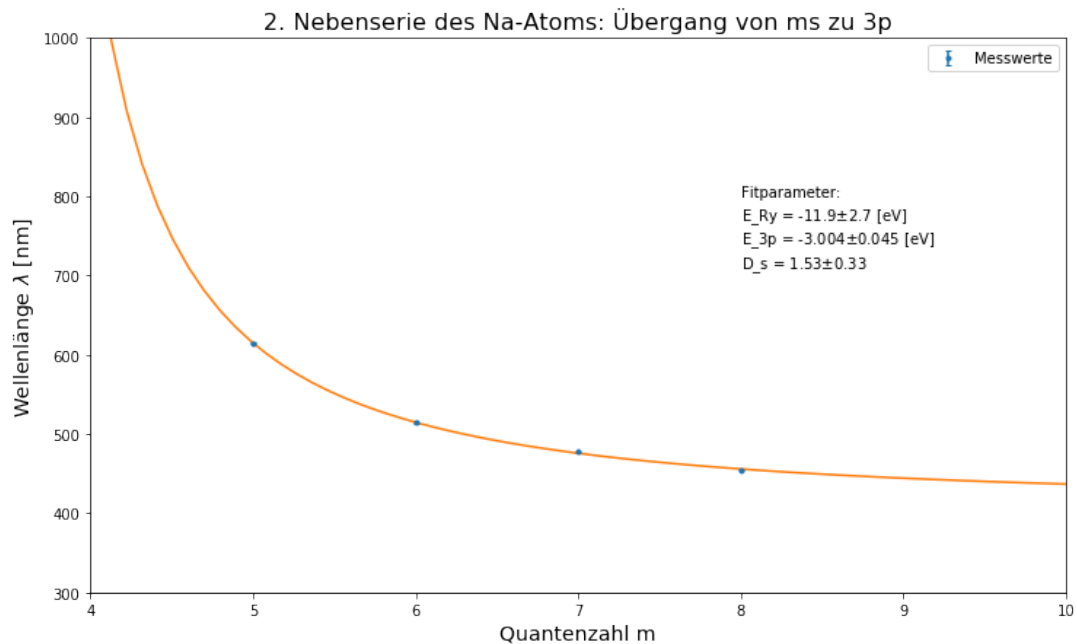
x=np.linspace(2.8, 12.2, 100)

# Plot
plt.figure(figsize = (12, 7))
plt.errorbar(m_ms3p, lambda_ms3p, yerr = sig_lambda_ms3p, fmt = '.,',
             ↪ capsize = 2,
             label = 'Messwerte')
plt.plot(x, fit_func(x, *popt_2))
plt.text(8, 800, 'Fitparameter:')
plt.text(8, 770, fitparameter('E_Ry', E_Ry_fit_2, sig_E_Ry_fit_2, 1, '↪
↪[eV]'))
plt.text(8, 740, fitparameter('E_3p', E_3p_fit_2, sig_E_3p_fit_2, 3, '↪
↪[eV]'))
plt.text(8, 710, fitparameter('D_s', D_s_fit_2, sig_D_s_fit_2, 2, ''))
plt.title('2. Nebenserie des Na-Atoms: Übergang von ms zu 3p', size = ↪
↪16)
plt.xlim((4, 10))
plt.ylim((300, 1000))
plt.xlabel('Quantenzahl m', size = 14)
plt.ylabel('Wellenlänge  $\lambda$  [nm]', size = 14)
plt.legend()

```



```
plt.savefig('images/234/V234Diagramm5b.png')
```



```
[24]: # chi^2 Analyse
chi_sq(fit_func, m_ms3p, lambda_ms3p, sig_lambda_ms3p, popt_2, 3)
```

```
chi2 = 3.5716318071014905
```

```
chi2_red = 3.5716318071014905
```

```
Wahrscheinlichkeit = 6.0 %
```

## Sigmas

Sonnenspektrum

```
[25]: # Vergleich der Fraunhofer Linien
fehler('Fraunhofer Linien', fraunhofer, sig_lambda, fraunhofer_lit, 0)
```

Fraunhofer Linien

```
Relativer Fehler: [0.25445293 0.25188917 0.23255814 0.20618557 0.
↪1934236
```

```
0.19011407
```

```
0.16977929 0.16977929 0.16977929 0.15923567 0.15267176 0.14556041
```

```
0.13175231]
```

Rel. Fehler (Vergleich): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]  
 Absoluter Fehler: [-0.4 0.1 -0.8 -1.1 -0.3 -1. 1.4 0. -0.6  
 ↪ 0.3 -1.3  
 0.3 -0.4]  
 Sigma-Abweichung: [0.4 0.1 0.8 1.1 0.3 1. 1.4 0. 0.6 0.3 1.3 0.  
 ↪ 3 0.4]

```
[26]: # Balmer Serie
fehler('Balmer Serie', balmer, sig_lambda, balmer_lit, 0)
```

Balmer Serie

Relativer Fehler: [0.15267176 0.20576132 0.23041475 0.24390244]  
 Rel. Fehler (Vergleich): [0. 0. 0. 0.]  
 Absoluter Fehler: [-1.3 -0.1 0. -0.1]  
 Sigma-Abweichung: [1.3 0.1 0. 0.1]

Natriumspektrum

```
[27]: # Erste Nebenserie
# lambda_1_n: theo
# lambda_md3p: exp
fehler('Erste Nebenserie', lambda_md3p, sig_lambda_md3p, lambda_1_n, ↪
↪ sig_lambda_1_n)
```

Erste Nebenserie

Relativer Fehler: [0.12224939 0.17636684 0.20120724 0.21459227 0.  
 ↪ 23094688  
 0.23310023  
 0.23529412 0.23809524]  
 Rel. Fehler (Vergleich): [0.12224939 0.0851111 0.0746188 0.06993552 0.  
 ↪ 06480037  
 0.06408512  
 0.063566 0.06317676]  
 Absoluter Fehler: [ 0. -2.49876465 -2.29230241 -1.95537575  
 -0.59480134 0.19109577  
 -0.3353339 -2.73084633]

```

Sigma-Abweichung:      [0.          2.24854888  2.14806371  1.85838663  0.
↪57262769
0.18426486
0.32371105  2.63837558]

```

```

[28]: # Erste Nebenserie mit alternativen Daten
fehler('Erste Nebenserie mit alternativen Daten', lambda_md3p, ↪
↪sig_lambda_md3p, lambda_1_n_b, sig_lambda_1_n_b)

```

Erste Nebenserie mit alternativen Daten

```

Relativer Fehler:      [0.12224939  0.17636684  0.20120724  0.21459227  0.
↪23094688
0.23310023
0.23529412  0.23809524]
Rel. Fehler (Vergleich): [0.03256922  0.02265815  0.01986074  0.01861249  0.
↪01724406
0.01705348
0.01691516  0.01681145]
Absoluter Fehler:      [-2.          -3.46745841 -3.03672512 -2.60922573
-1.15609846 -0.35787101
-0.87543728 -3.26435131]
Sigma-Abweichung:      [1.93227685  3.43885017  3.02185987  2.59935742  1.
↪15287208
0.35691553
0.87317459  3.25611836]

```

```

[29]: # Zweite Nebenserie
fehler('Zweite Nebenserie', lambda_ms3p, sig_lambda_ms3p, lambda_2_n, ↪
↪sig_lambda_2_n)

```

Zweite Nebenserie

```

Relativer Fehler:      [0.16260163  0.19455253  0.20955574  0.21978022]
Rel. Fehler (Vergleich): [0.12339003  0.08403458  0.07325034  0.06887503]
Absoluter Fehler:      [-6.73463591 -4.24373047 -0.0320105  -1.
↪19667814]
Sigma-Abweichung:      [5.34338059  3.89077118  0.03021739  1.14164959]

```

```
[30]: # Hauptserie
lambda_mp3s = 330
sig_lambda_mp3s = 1
fehler('Hauptserie', lambda_mp3s, sig_lambda_mp3s, lambda_H,
      ↪ sig_lambda_H)
```

Hauptserie

Relativer Fehler: 0.30303030303030304  
 Rel. Fehler (Vergleich): [0.48471199]  
 Absoluter Fehler: [-2.29852861]  
 Sigma-Abweichung: [1.21238694]

Serienenergien

```
[31]: # Rydberg-Energie
fehler('Rydberg-Energie: 1. NS - Lit ', E_Ry_fit_1, sig_E_Ry_fit_1,
      ↪ E_Ry, 0)
```

Rydberg-Energie: 1. NS - Lit  
 Relativer Fehler: -2.5896665793294975  
 Rel. Fehler (Vergleich): -0.0  
 Absoluter Fehler: 0.49275983401194523  
 Sigma-Abweichung: 1.4511575073317156

```
[32]: fehler('Rydberg-Energie: 2. NS - Lit', E_Ry_fit_2, sig_E_Ry_fit_2,
      ↪ E_Ry, 0)
```

Rydberg-Energie: 2. NS - Lit  
 Relativer Fehler: -23.01382424759472  
 Rel. Fehler (Vergleich): -0.0  
 Absoluter Fehler: 1.6736820154583114  
 Sigma-Abweichung: 0.6095309140378529

```
[33]: fehler('Rydberg-Energie: 1. NS - 2. NS', E_Ry_fit_1, sig_E_Ry_fit_1,
      ↪ E_Ry_fit_2, sig_E_Ry_fit_2)
```

Rydberg-Energie: 1. NS - 2. NS  
 Relativer Fehler: -2.5896665793294975

Rel. Fehler (Vergleich): -23.01382424759472  
 Absoluter Fehler: -1.1809221814463662  
 Sigma-Abweichung: 0.42682358964688294

```
[34]: # Serienenergie
fehler('Serienenergie E_3p: 1. NS - Lit', E_3p_fit_1, sig_E_3p_fit_1,
      ↪E_3p, sig_E_3p)
```

Serienenergie E\_3p: 1. NS - Lit  
 Relativer Fehler: -0.17716609662004004  
 Rel. Fehler (Vergleich): -0.061205080142107016  
 Absoluter Fehler: -0.0032716975585564967  
 Sigma-Abweichung: 0.5760144265709591

```
[35]: fehler('Serienenergie E_3p: 2. NS - Lit', E_3p_fit_2, sig_E_3p_fit_2,
      ↪E_3p, sig_E_3p)
```

Serienenergie E\_3p: 2. NS - Lit  
 Relativer Fehler: -1.4960454107195604  
 Rel. Fehler (Vergleich): -0.061205080142107016  
 Absoluter Fehler: 0.022925857022761065  
 Sigma-Abweichung: 0.509631165296719

```
[36]: fehler('Serienenergie: 1. NS - 2. NS', E_3p_fit_1, sig_E_3p_fit_1,
      ↪E_3p_fit_2, sig_E_3p_fit_2)
```

Serienenergie: 1. NS - 2. NS  
 Relativer Fehler: -0.17716609662004004  
 Rel. Fehler (Vergleich): -1.4960454107195604  
 Absoluter Fehler: -0.02619755458131756  
 Sigma-Abweichung: 0.5787395232335217

```
[37]: # Korrekturterme
fehler('D_s_2 - D_s_lit', D_s_fit_2, sig_D_s_fit_2, Ds, sig_Ds)
```

D\_s\_2 - D\_s\_lit  
 Relativer Fehler: 21.317823565446247

Rel. Fehler (Vergleich): 0.20732048086167204

Absoluter Fehler: 0.15392626969638168

Sigma-Abweichung: 0.47339664132005976

```
[38]: fehler('D_p_1 - D_p_lit', D_d_fit_1, sig_D_d_fit_1, 0.000000000001, 0.  
        ↪0000000000000001)
```

D\_p\_1 - D\_p\_lit

Relativer Fehler: 58.75795777993172

Rel. Fehler (Vergleich): 1.0

Absoluter Fehler: 0.058004015885407746

Sigma-Abweichung: 1.7018971349005958

```
[ ]:
```