

# Physikalisches Anfängerpraktikum III für Lehramtsstudierende

Name: Marius Pfeiffer

Matrikel-Nr.: 4188573

E-Mail: marius.pfeiffer@stud.uni-heidelberg.de

Betreut durch: Henri Valentin Lurtz

12.02.2025

## Versuch 234: Lichtquellen und Gitterspektroskopie

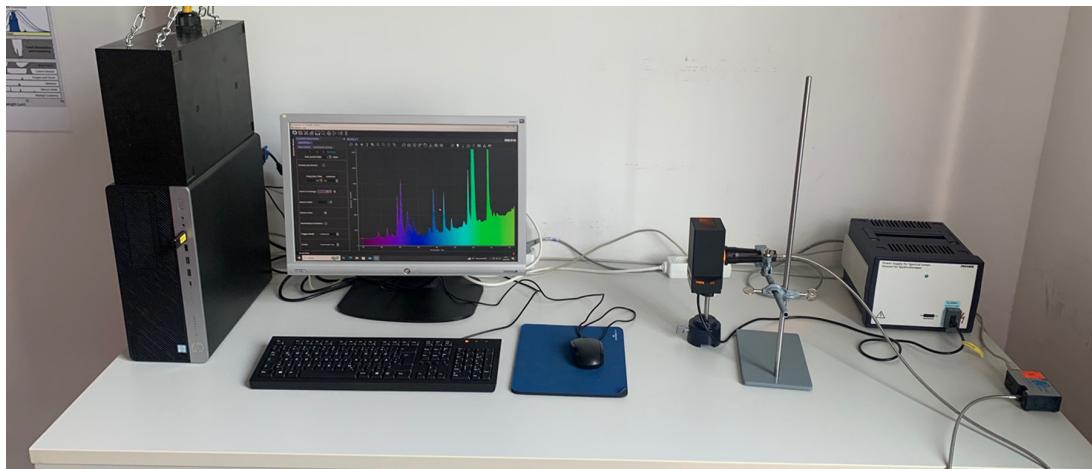


Abbildung 1: Versuchsaufbau (Quelle: Praktikumsanleitung)

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>2</b>
1.1 Physikalische Grundlagen . . . . .	2
1.2 Versuchsdurchführung . . . . .	4
<b>2 Messprotokoll</b>	<b>5</b>
<b>3 Auswertung</b>	<b>7</b>
3.1 Untersuchung des Sonnenlichtspektrums . . . . .	7
3.2 Vergleich der Spektren verschiedener Leuchtmitte . . . . .	10
3.3 Untersuchung des Natriumspektrums . . . . .	12
3.4 Bestimmung der Serienenergien und der l-abhängigen Korrekturfaktoren . .	16
<b>4 Zusammenfassung und Diskussion</b>	<b>19</b>

# 1 Einleitung

In Versuchs 234 betrachten wir verschiedene Arten von Lichtquellen und deren Spektren. Hierbei unterscheiden wir speziell zwischen Temperaturstrahlern und Nichttemperaturstrahlern, welche in ihrer Art und Weise, Licht bzw. Strahlung auszusenden auf zwei verschiedenen physikalischen Phänomenen basieren.

## 1.1 Physikalische Grundlagen

Zur Erklärung von **Temperaturstrahlern** betrachten wir zunächst einen idealisierten schwarzen Körper, also einen Körper mit dem Emissionsvermögen  $\varepsilon = 1$ . Sobald die Temperatur  $T$  dieses, und allgemein jedes Körpers, größer 0K ist, sendet dieser elektromagnetische Strahlung aus, deren Intensität dem Planck'schen Strahlungsgesetz

$$M_\lambda(\lambda, T) dA d\lambda = \frac{2\pi hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kT}} - 1} dA d\lambda \quad (1)$$

folgt.  $M_\lambda$  steht hierbei für die Strahlungsleistung, welche vom Flächenelement  $dA$  im Wellenlängenbereich  $[\lambda, \lambda + d\lambda]$  abgestrahlt wird. Die Intensitätsverteilung besitzt ein Maximum bei einer bestimmten Wellenlänge, welche nach dem Wie'schen Verschiebungsgesetz

$$\lambda_{\max} = \frac{2897.8 \mu\text{m K}}{T} \quad (2)$$

von der Temperatur des Körpers abhängt.

Durch Betrachtung der Farbe des ausgestrahlten Lichts lassen sich die Lichtquellen charakterisieren. Ein Körper erscheint bei einer Temperatur von 0K absolut schwarz. Darauf folgt bei leichter Erwärmung die Emission von zunächst rotem Licht. Bei einer weiteren Erwärmung wird zusätzlich grünes Licht emittiert, welches sich mit den Rotanteilen zu orange bis gelblich erscheinendem Licht vermischt. Weißlich wirkendes Licht erhält man, wenn sich bei einer Temperatur von etwa 5500K alle Wellenlängen des sichtbaren Bereichs mit etwa gleicher Intensität emittiert werden. Mit weiterer Erwärmung des Körpers kommen vermehrt Blauanteile hinzu, welche das Licht zunächst hellblau, dann blau bis violett erscheinen lässt. Konträr dazu bezeichnen wir Licht mit höheren Rotanteilen umgangssprachlich als „wärmer“ und Licht mit höheren Blauanteilen als „kälter“.

Die Streuung von Licht ist nach der Rayleigh-Streuung stark Frequenzabhängig, daher wird blaues Licht deutlich stärker gestreut als rotes Licht. Dies ist auch der Grund, weshalb der Himmel blau erscheint, obwohl das direkte Sonnenlicht weiß ist. Betrachtet man das kontinuierliche Spektrum des von der Sonne abgestrahlten Lichts, so sind in diesem Absorptionslinien, die sogenannten Fraunhoferlinien zu erkennen. Diese entstehen dadurch, dass die Atmosphärenschichten der Sonne und der Erde, welche das Licht passiert, Teile der Strahlung absorbieren.

Bei **Nichttemperaturstrahlern** beruht die Aussendung von Licht auf der Anregung von Atomzuständen in Gasen oder Festkörpern oder der Rekombination von Elektron-Loch-Paaren in Halbleitern. Diese Art von Lichtquellen zeichnet sich durch diskrete Spektren, statt der kontinuierlichen bei Temperaturstrahlern, aus. Leuchtstoffröhren und Energiesparlampen sind Beispiele für Gasentladungslampen, die Quecksilberdampf als Leuchtgas verwenden. Da die meiste emittierte Strahlung von Quecksilber im nicht sichtbaren UV-Bereich liegt, werden derartige Lampen mit einem Leuchtstoff umzogen, welcher bei Bestrahlung mit UV-Licht im sichtbaren Bereich fluoresziert. In Leuchtdioden (LEDs) werden in pn-Übergängen bei der Rekombination von Elektron-Loch-Paaren Photonen emittiert. Die Farbe des Lichts hängt hierbei von den Halbleitermaterialien ab.

Neben den Quecksilberdampflampen sind Natriumdampflampen ein Beispiel für Gasentladungslampen, welche wir in diesem Versuch auch näher betrachten werden. Natrium ist ein Alkalimetall, dessen Atome ein Leuchtelektron besitzen. Das Potential, welches dieses eine Elektron „sieht“, kann in Kernnähe durch das Coulombpotential

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

und in weiterer Entfernung über das Wasserstoffpotential

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

angenähert werden. Dieses ist kein Coulombpotential mehr, somit ist die Entartung des Bahndrehimpuls  $l$  aufgehoben und die Energieniveaus

$$E_{n,l} = -13.6 \text{ eV} \frac{1}{(n - \Delta_{n,l})^2} \quad (5)$$

hängen neben der Hauptquantenzahl  $n$  auch von der Bahndrehimpulsquantenzahl ab. Die Korrektur  $\Delta_{l,n}$  kann aufgrund der geringen  $n$ -Abhängigkeit zu  $\Delta_l$  umgeschrieben beziehungsweise angenähert werden. Halten wir die Hauptquantenzahl  $n$  fest, so liegt das  $s$ -Niveau am tiefsten und das  $d$ -Niveau am höchsten. Die Übergänge  $np \rightarrow 3s$  werden als Hauptserie bezeichnet. Die Übergänge  $nd \rightarrow 3p$  und  $ns \rightarrow 3p$  bezeichnen wir als erste und zweite Nebenserie. Die wichtigsten Übergänge sind in Abbildung (2) dargestellt.

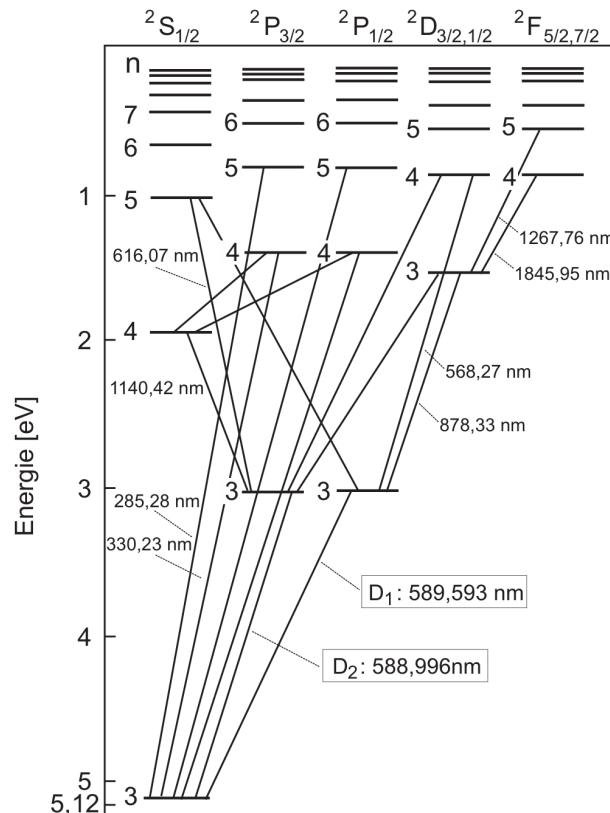


Abbildung 2: Energiespektrum und wichtige Übergänge im Natriumatom (Quelle: Praktikumsanleitung)

## 1.2 Versuchsdurchführung

Die Durchführung des Versuchs setzt sich zusammen aus allgemeineren Untersuchungen der Spektren der Sonne und verschiedenen künstlichen Lichtquellen im ersten Teil und einer detaillierteren Untersuchung des Natriumspektrums im zweiten Teil. Zur Aufzeichnung der Spektren verwenden wir ein computergesteuertes Gitterspektroskop, welches einen Spektralbereich von 180nm bis 950nm registrieren kann. Das Programm `OceanView`, welches wir zur Erfassung der Daten verwenden, bietet Funktionen zur Aufzeichnung und Subtraktion von Dunkelstrom, sowie eine automatisierte Mittelwertbildung über mehrere Scans zur Rauschreduzierung.

**Aufnahme des Sonnenspektrums.** Wir richten das Spektroskop zunächst durch das geöffnete Fenster auf den Himmel und zeichnen einmal das Spektrum auf. Dann schließen wir das Fenster und nehmen das Spektrum noch einmal durch die Scheibe auf.

**Qualitativer Vergleich der Spektren verschiedener Leuchtmittel.** Qualitativ zeichnen wir die Spektren verschiedener Lichtquellen auf. Dazu gehörten

- eine Glühlampe,
- eine Energiesparlampe,
- ein Laser,
- LEDs in Blau, Gelb, Orange, Rot und verschiedenen Weißtönen.

**Aufnahme der Spektrallinien kleiner Intensität einer Natriumlampe.** Wir stellen die Natriumdampflampe auf, warten bis diese auf Betriebstemperatur ist und richten das Spektroskop aus. Ziel dieses ersten Aufgabenteils ist die Aufzeichnung der schwächeren Spektrallinien im Bereich von 400 bis 540nm. Durch das Justieren der Irisblende am Spektroskop, sowie der Integrationszeit und Mittelwertbildung in `OceanView` versuchen wir die bestmögliche Auflösung der Spektrallinien zu erzielen.

**Aufnahme der Spektrallinien um die D-Linie einer Natriumlampe.** Wir justieren die Einstellungen so, dass im Spektrum bei 570 und 615nm jeweils eine Spektrallinie beobachtet werden kann, welche nicht von der D-Linie verschluckt werden.

**Aufnahme der Spektrallinien im Bereich von 650 bis 850nm.** Wir zeichnen abschließend die Spektrallinien des Natriumspektrums im Bereich von 650 bis 850nm auf.

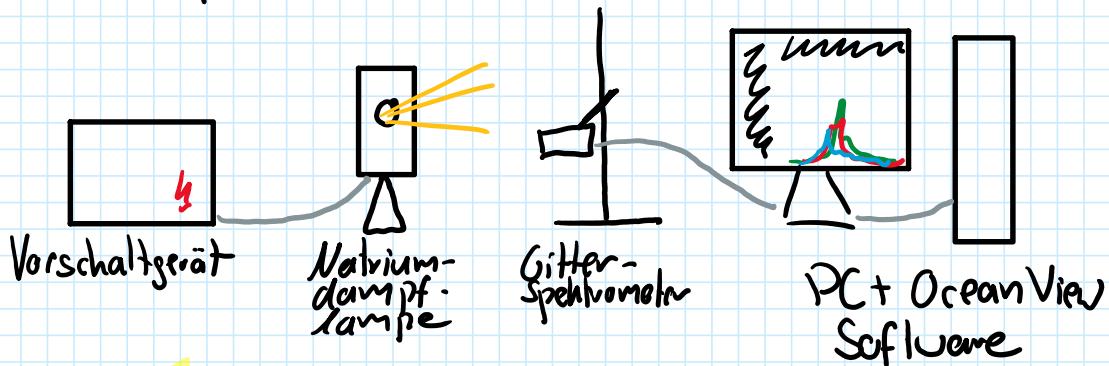
## 2 Messprotokoll

Versuchsprotokoll 234

Marius Pfeiffer  
Robert Grosch

12.02.2025

### Versuchsaufbau



### Aufgabe 1, Sonnenlichtspektrum

Integrationszeit 100ms, 1 Scan to Average

Die Dunkelstrommessung zeigt, wie erwartet, nur bei sehr hoher Zoomstufe Ausschläge im Spektrogramm.

Intensität bei Sonnenlichtspektrum ohne Fenster allgemein etwas höher als durch das Fenster.  
Vor allem im IR- und UV-Bereich des Spektrums sichtbar.

### Aufgabe 2, Vergleich der Spektren verschiedener Lichtquellen

(a) LEDs in rot, orange, gelb, blau zeigen um die jeweilige Farbe zentriertes Spektrum

LEDs in weiß zeigen ein kontinuierliches Spektrum auf. Je "wärmer" das Weiß, umso weniger ist der blaue Peak

- (b) Spektrum der Glühlampe ist kontinuierlich
- (c) Spektrum der Energiesparlampe ist diskret
- (d) Spektrum des Lasers konzentriert, diskret um 532 nm

Spektren wurden alle mit 100ms Integrationszeit aufgenommen.

### Aufgabe 3

Integrationszeit 100ms, 40 Scans für Mittelwert

### Aufgabe 4

Bei Sättigung im D-Peak

Integration Time: 10ms

Scans für Mittelwert: 1

Ohne Sättigung im D-Peak

Integration Time: 10ms

Scans für Mittelwert: 5

### Aufgabe 5

H. Lutz

Integration Time: 50ms

Scans für Mittelwert: 5

Alle Messungen wurden mit DunkelstromKorrektur durchgeführt.

### 3 Auswertung

#### Vorbemerkungen

Bevor wir mit der Aufzeichnung der eigentlichen Spektren begonnen haben, haben wir eine Dunkelstrommessung durchgeführt. Das Spektrum dieser ist in Abbildung (3) zu sehen.

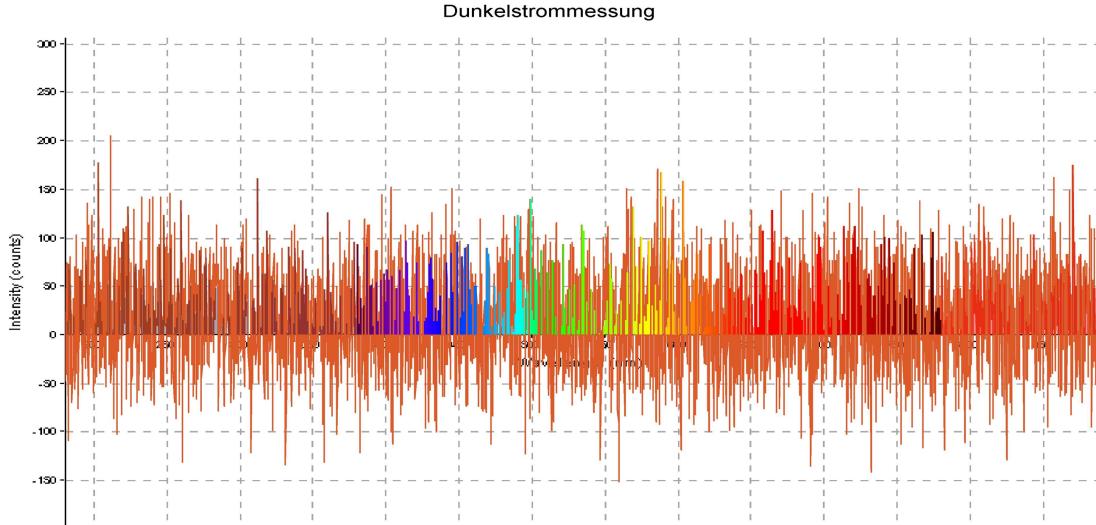


Abbildung 3: Spektrum Dunkelstrommessung

Es handelt sich hierbei um ein Untergrundrauschen, welches wir mit der **OceanView** Software automatisch von allen weiteren aufgezeichneten Spektren abziehen.

Sofern nicht anders angegeben berechnen wir die Fehler zusammengesetzter Werte anhand der standardmäßigen Gauß'schen Fehlerfortpflanzung. Die  $\sigma$ -Abweichung zweier fehlerbehafteter Werte  $x \pm \Delta x$  und  $y \pm \Delta y$  berechnen wir anhand der Formel

$$\sigma = \frac{|x - y|}{\sqrt{\Delta x^2 + \Delta y^2}}. \quad (6)$$

#### 3.1 Untersuchung des Sonnenlichtspektrums

Am Tag der Versuchsdurchführung war das Wetter leider stark bewölkt, weshalb wir mit dem Spektroskop nicht auf den blauen Himmel zielen konnten. Es ist also zu beachten, dass die folgenden Betrachtungen durch die Auswirkungen der Wolkendecke gestört sind. Die untenstehende Abbildung (4) zeigt aufgezeichnete Spektrum des Tageslichts durch das geöffnete Fenster (blau) und durch die Fensterscheibe (orange) im Vergleich. Es ist bereits hier zu sehen, dass die Intensität durch das Glas über das gesamte Spektrum hinweg abgeschwächt wird. Die stärkste Abschwächung verzeichnen wir bei niedrigen Wellenlängen, also im UV-Bereich. Dies ist auch am Verlauf der Absorption in Abbildung (5) zu sehen. Die geringste Abschwächung tritt im Bereich der Wellenlänge zwischen 400 bis 600nm, also dem sichtbaren Bereich auf. In Richtung des Rot- bis Infrarotbereichs steigt die Absorption wieder etwas an.

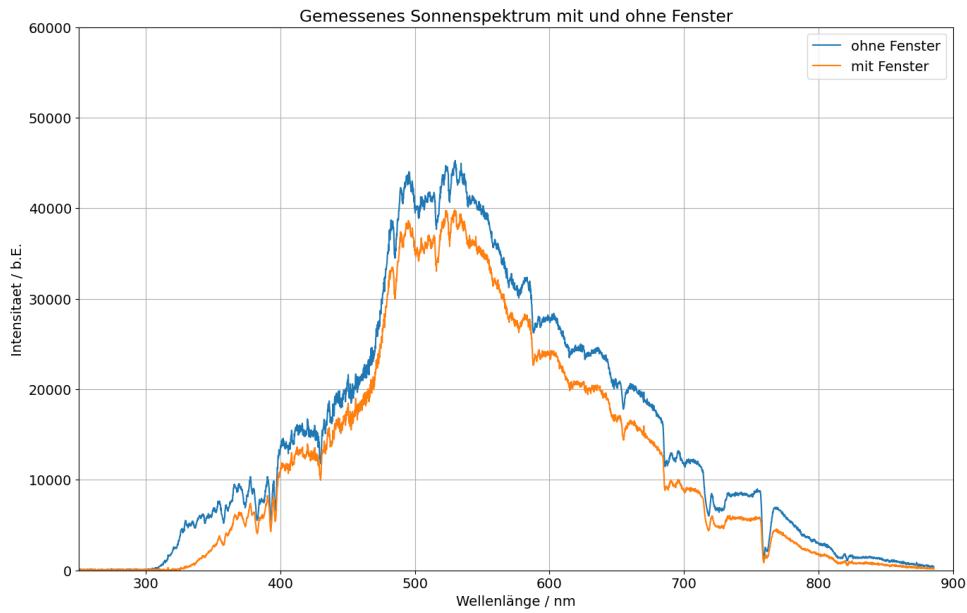


Abbildung 4: Tageslichtspektrum direkt (blau) und durch Fensterglas (orange).

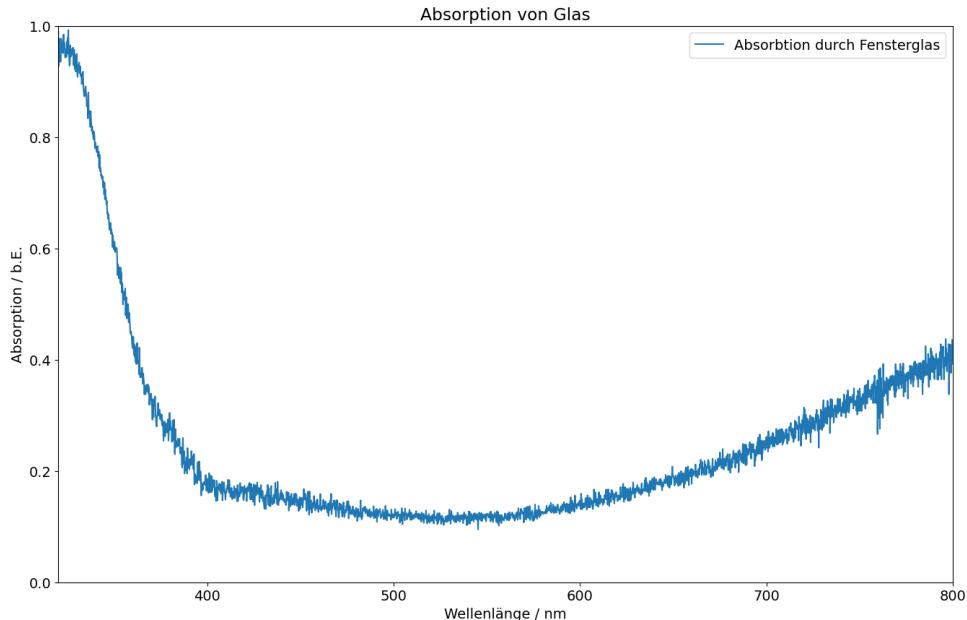


Abbildung 5: Absorption des Tageslichts durch das Fensterglas, abhängig von der Wellenlänge.

Die vielen im Spektrum sichtbaren lokalen Minima sind gerade die Wellenlängen der Fraunhoferlinien, welche durch Absorption von Licht bestimmter Wellenlängen in der Sonnen- und Erdatmosphäre entstehen. Abbildung (6) zeigt erneut das Spektrum des Sonnenlichts, ohne Fensterscheibe. Markiert sind hier in Orange nun die Minima im Spektrum, welche jeweils am nächsten an der erwarteten Wellenlänge einer Fraunhoferlinie liegen. Zusätzlich sind in Grün die Literaturwerte der Wellenlänge der Balmer Serie von Wasserstoff eingezeichnet.

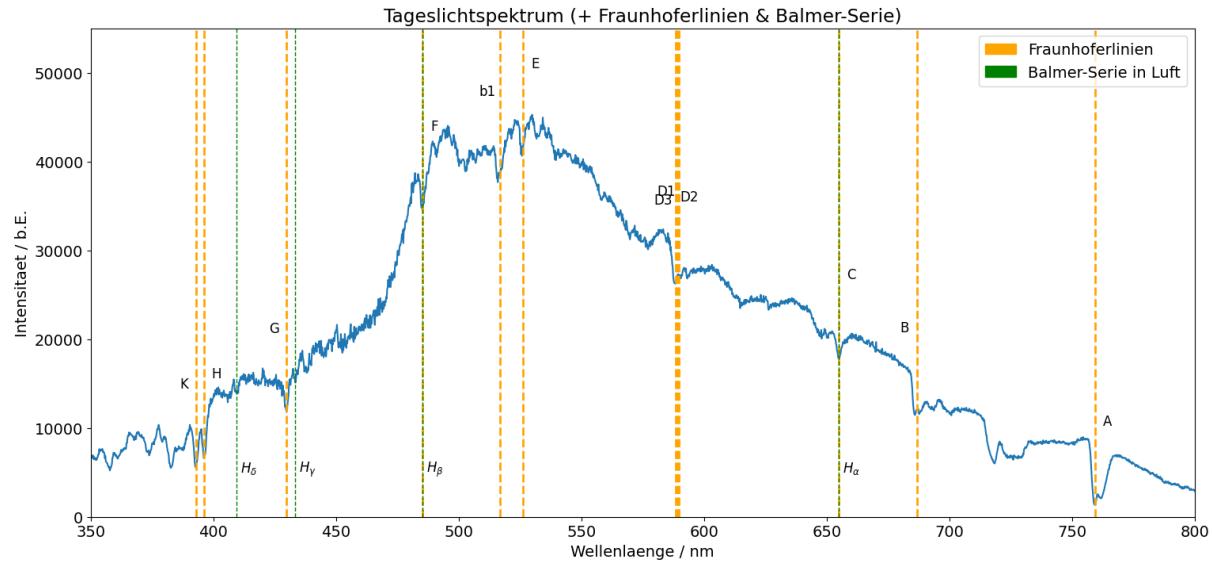


Abbildung 6: Fraunhoferlinien und Balmerserie im Tageslichtspektrum.

Tabelle (1) zeigt eine Aufschlüsselung der erwarteten Wellenlänge der Fraunhoferlinien und der Linien der Balmerserie, die von uns aus dem Spektrum abgelesenen Werte, sowie die Abweichung zwischen den Werten. Als Fehler für die abgelesenen Wellenlängen verwenden wir hier jeweils einen Wert von 1.0nm. Es ist zu sehen, dass sich die Abweichung, bis auf wenige ausnahmen auf unter einem  $\sigma$  beläuft.

Tabelle 1: Vergleich der erwarteten und gemessenen Wellenlängen der Fraunhofer- und Balmerlinien

Linie	Literaturwert [nm]	Abgelesener Wert [nm]	Abweichung [ $\sigma$ ]
K	393.4	393.0	0.4
H	396.8	396.1	0.7
G	430.8	429.8	1.0
F	486.1	485.2	0.91
b1	518.4	516.7	1.7
E	527.0	526.2	0.8
D3	587.6	588.4	0.8
D2	589.0	589.0	0.0
D1	589.6	589.7	0.11
C	656.3	655.0	1.3
B	686.7	686.7	0.0
A	759.4	759.4	0.0
$H_\alpha$	656.3	655.0	1.3
$H_\beta$	486.1	485.2	0.91
$H_\gamma$	434.0	433.4	0.61
$H_\delta$	410.1	409.5	0.61

### 3.2 Vergleich der Spektren verschiedener Leuchtmittel

Im Folgenden betrachten wir die Spektren verschiedener Lichtquellen. Abbildung (7) zeigt hierzu zunächst die Spektren der drei untersuchten farbigen LEDs, sowie des Lasers.

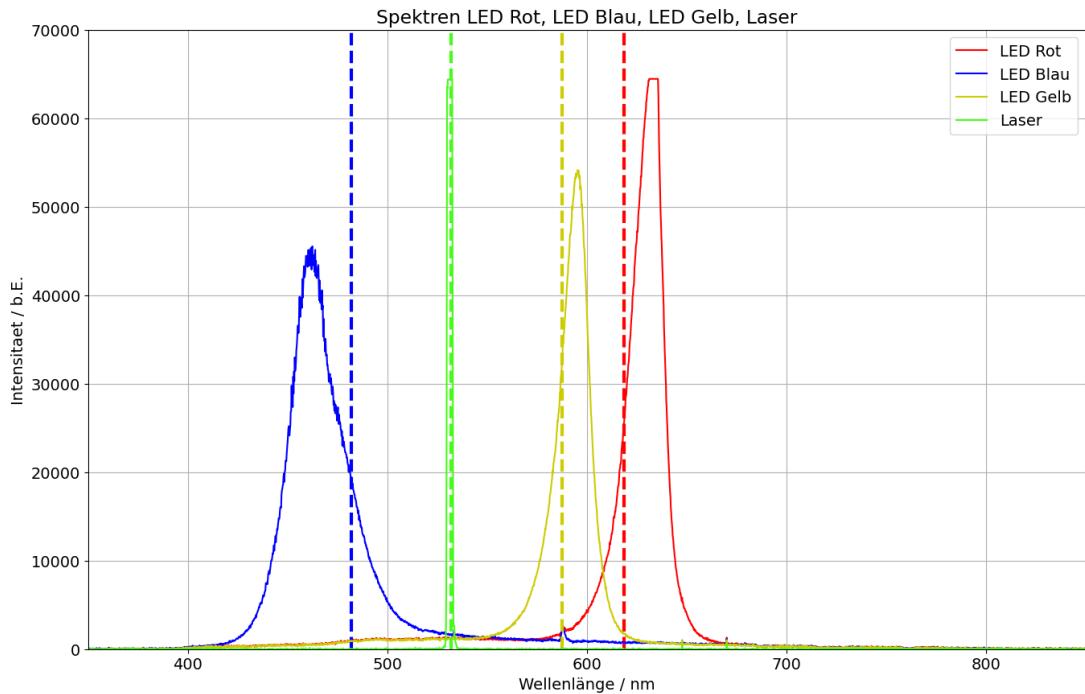


Abbildung 7: Aufgezeichnete Spektren der farbigen LEDs, sowie des Lasers.

Die vertikalen Linien im Diagramm zeigen jeweils das nach Intensität gewichtete Mittel der Wellenlängen der Lichtquellen. Es ist zu erkennen, dass es sich hierbei in allen vier Fällen um diskrete Spektren handelt. Bei den LEDs streuen dabei die Wellenlängen etwas mehr und das Maximum der Verteilung ist etwas vom Mittel verschoben. Der Laser sendet konzentriertes Licht genau einer Wellenlänge aus, was sich ebenfalls an der direkten Überlagerung des „Peaks“ mit dem Mittel zeigt.

Die Spektren der drei verschiedenen weißen LEDs sind gemeinsam mit dem der Energiesparlampe in Abbildung (8) dargestellt. Alle vier Lichtquellen erzeugen, weitestgehend, weißes Licht. Am Spektrum der Energiesparlampe ist zu erkennen, dass diese das weiße Licht durch die Überlagerung mehrerer diskreter Wellenlängen erzeugt. Bei den weißen LEDs sehen wir jeweils einen relativ deutlichen, schmalen Peak bei den „blauen“ Wellenlängen und einen sehr breiten Peak um den Bereich, in welchem Gelb zu verorten ist. Weiß wird hierbei also durch die Überlagerung von blauem und gelbem Licht erzeugt. Wir können außerdem beobachten, dass der Blau-Peak abnimmt, je „wärmer“ das Weiß der LED wird. Durch geringere Blauanteile erhalten wir also wärmer erscheinendes Licht.

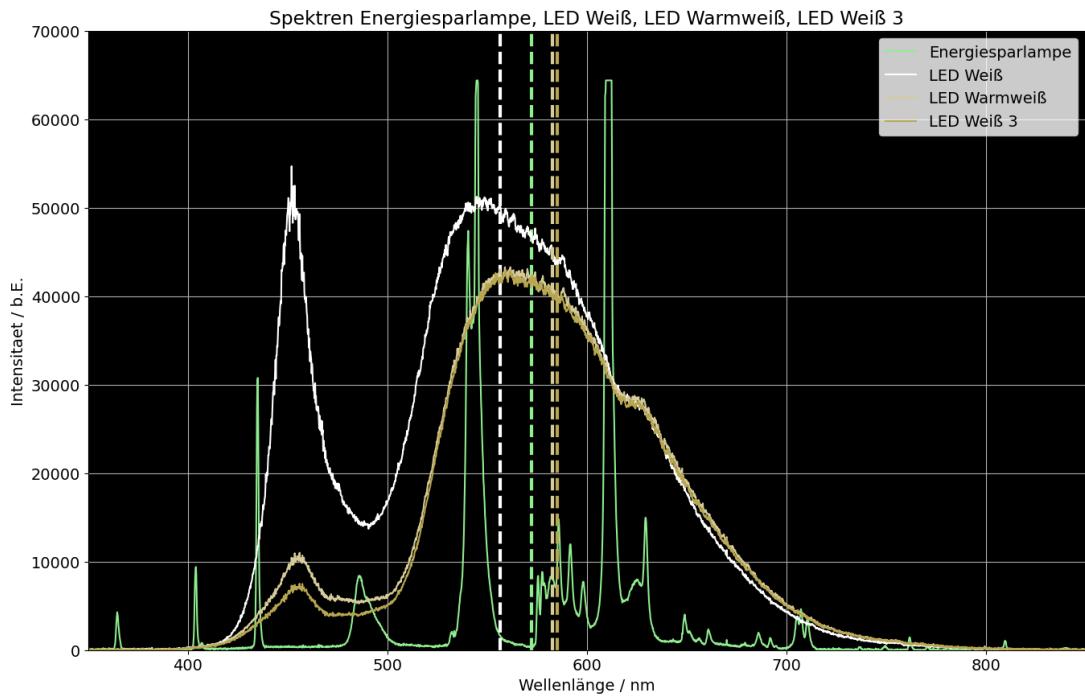


Abbildung 8: Aufgezeichnete Spektren der weißen LEDs, sowie der Energiesparlampe.

Während die bisher betrachteten Lichtquellen Nichttemperaturstrahler sind, ist die Glühlampe ein Temperaturstrahler. Sie weist also ein kontinuierliches Spektrum auf, wie es auch in Abbildung (9) zu sehen ist.

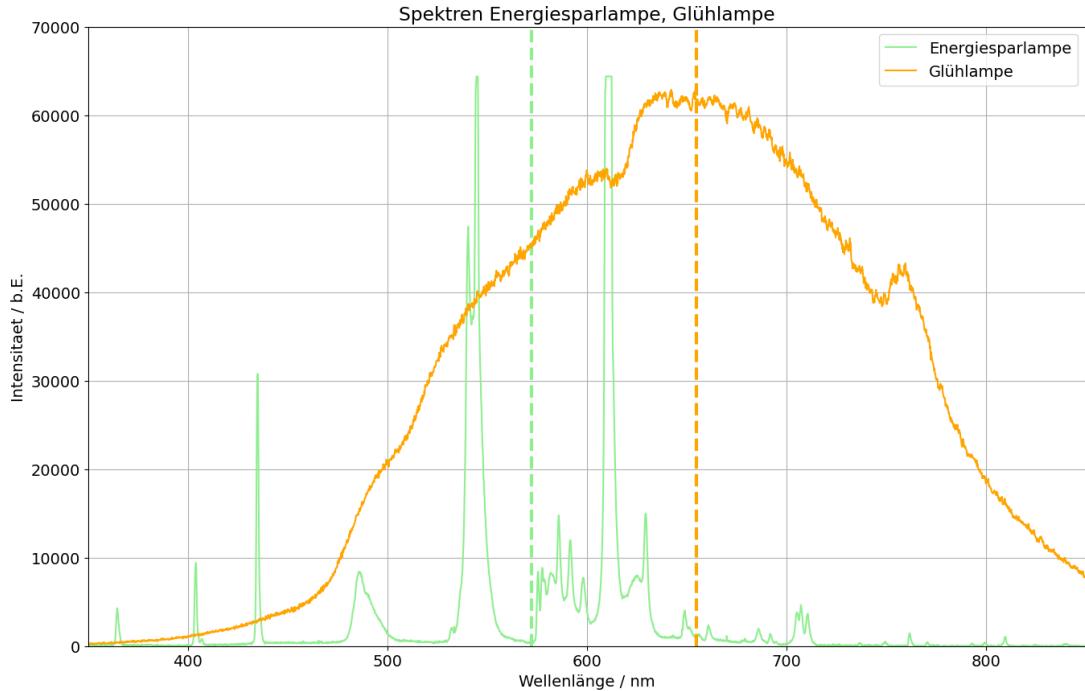


Abbildung 9: Aufgezeichnete Spektren der Glühlampe, sowie der Energiesparlampe.

Am Spektrum der Glühlampe ist zu erkennen, dass sich ein nicht ungewisser Teil des Spektrums im oberen bis nicht sichtbaren Wellenlängenbereich befindet. Es geht hierbei also sehr viel Energie in Form von Wärme an die Umgebung verloren. Im Vergleich dazu ist die Energiesparlampe, deren Spektrum ebenfalls noch einmal in diesem Diagramm

zu sehen ist, deutlich energieeffizienter. Die Glühlampe erzeugt eher ein warmes Licht, während die Lichttemperatur der Energiesparlampe eher in Richtung des kälteren Bereichs liegt.

### 3.3 Untersuchung des Natriumspektrums

#### *Beobachtete Spektrallinien*

Im Folgenden haben wir drei verschiedene Bereiche des Spektrums einer Natriumdampflampe aufgezeichnet und möchten im weiteren Verlauf die beobachtbaren Spektrallinien mit den theoretischen Vorhersagen der Zustandsübergänge vergleichen. Abbildung (10) zeigt die Linien geringer Intensität, welche im Bereich einer Wellenlänge von 350nm bis 550nm zu finden sind.

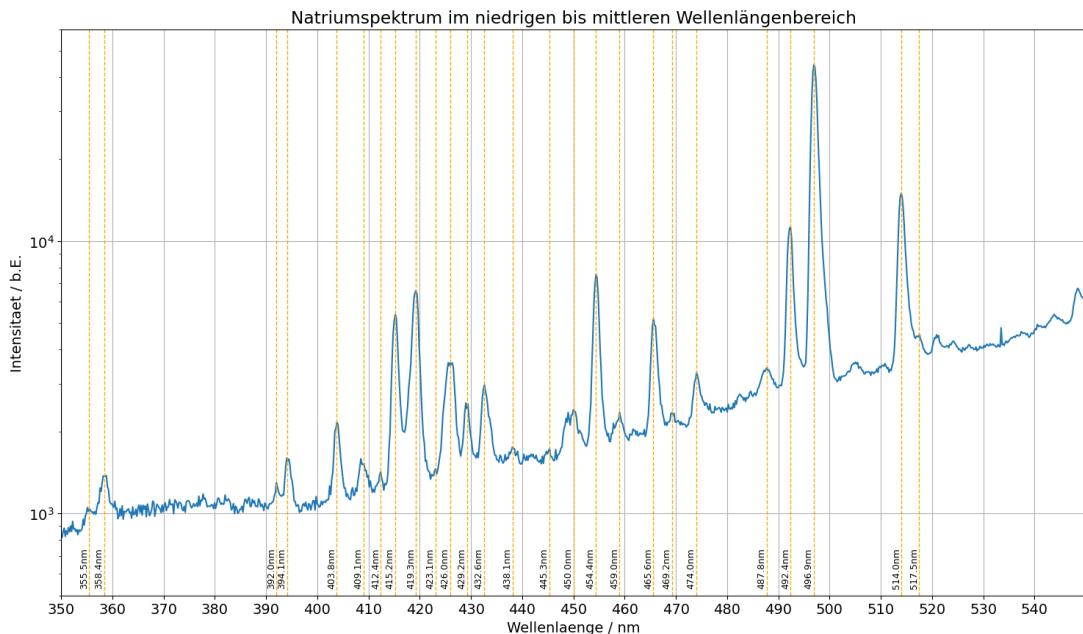


Abbildung 10: Natriuspektrum im Wellenlängenbereich von 350 bis 550nm.

Als Nächstes beobachten wir den Wellenlängenbereich zwischen 585 und 595nm. In diesem ist die markante D-Linie des Natriumspektrums zu finden. Die Intensität der D-Linie ist im Vergleich zu den anderen Linien so hoch, dass keine weiteren Linien im aufgezeichneten Spektrum sichtbar sind, wenn die D-Linie nicht in Sättigung ist. Diesen Fall zeigt die Abbildung (11). Ebenfalls relativ gut zu erkennen ist in dieser Abbildung die Doppelstruktur der D-Linie aus zwei dicht beieinander liegenden Linien.

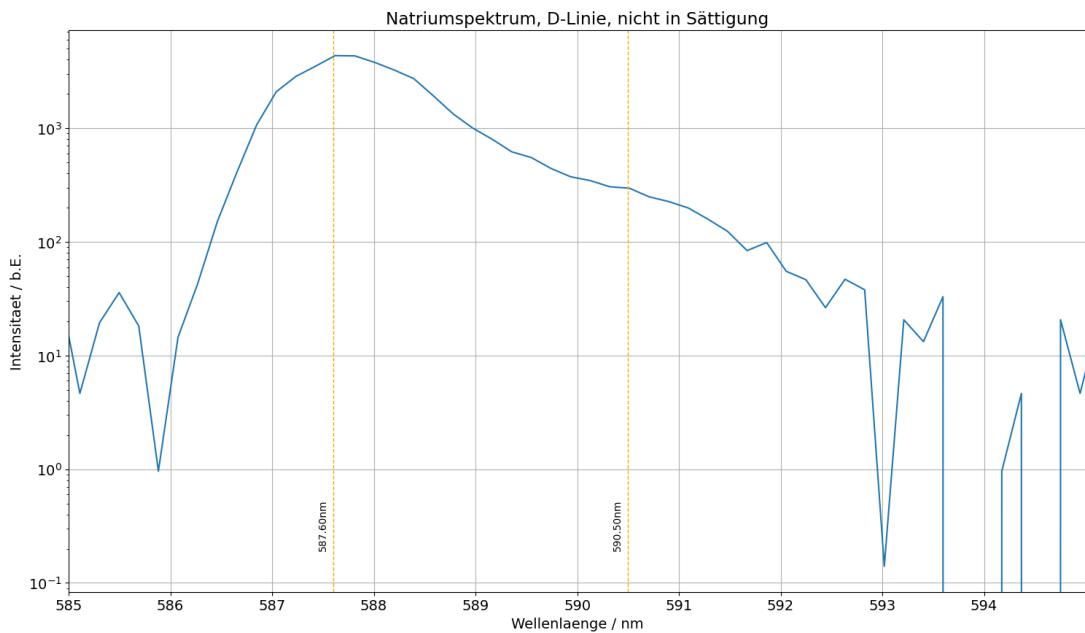


Abbildung 11: D-Linie im Natriumspektrum, nicht in Sättigung.

Betrachten wir einen etwas weiteren Wellenlängenbereich um die D-Linie, erlauben dieser allerdings in Sättigung zu gehen, so können wir in deren Umgebung weitere Spektrallinien geringer Intensität beobachtbaren. Diese sind in Abbildung (12) gekennzeichnet.

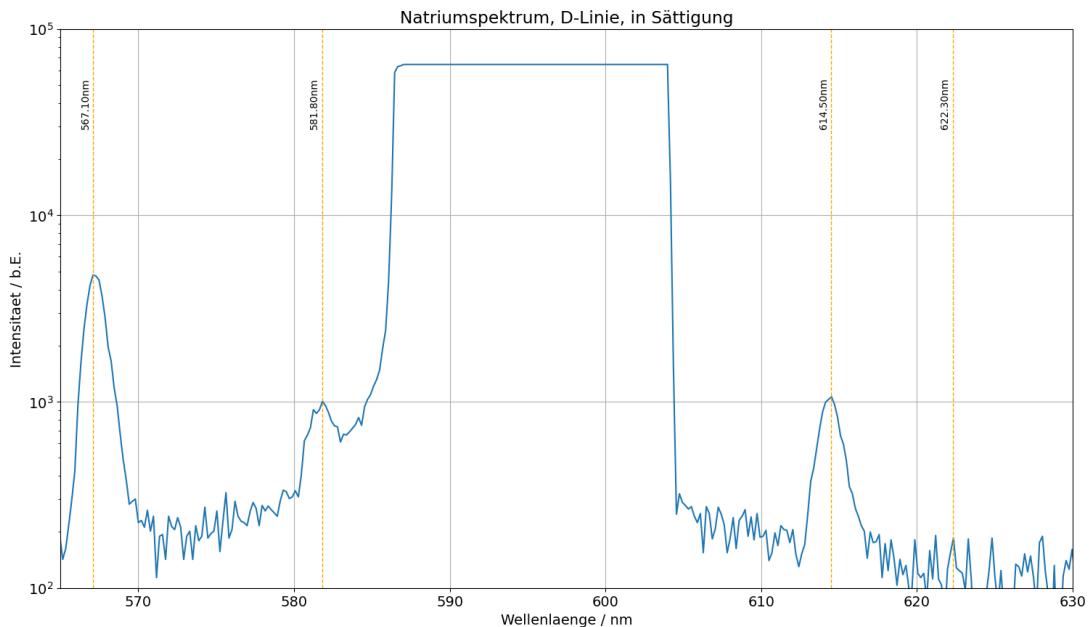


Abbildung 12: Weitere Linien um die D-Linie im Natriumspektrum, während diese in Sättigung ist.

Als letztes Betrachten wir den oberen Bereich des Natriumspektrums von 650 bis 850nm. Wie in Abbildung (13) zu sehen sind hier wieder sehr viele, teils auch sehr markante, Spektrallinien aufzufinden.

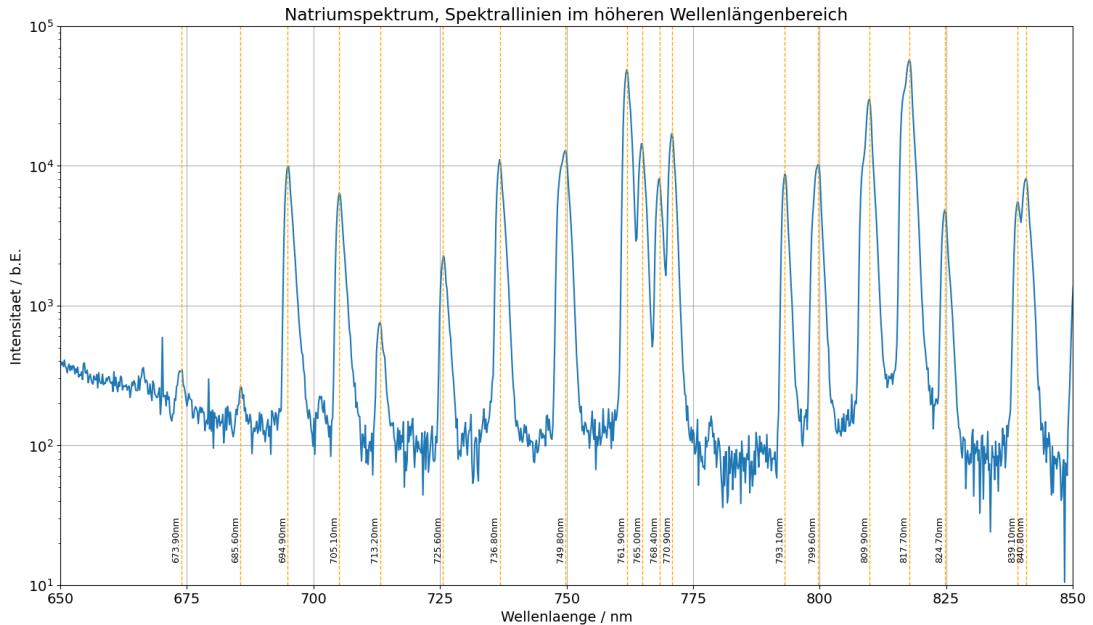


Abbildung 13: Natriumspektrum im Wellenlängenbereich von 350 bis 550nm.

Nun möchten wir die beobachteten Spektrallinien mit den Vorhersagen der Theorie vergleichen.

*Erwartete Linien für die 1. Nebenserie  $md \rightarrow 3p$*

*Vorbemerkung: In den folgenden Berechnungen gehen wir für die Konstanten von den Werten  $E_{Ry} = -13.605\text{eV}$  und  $hc = 1.2398 \times 10^3\text{nm eV}$  aus.*

Den Korrekturterm  $\Delta_d$  können wir für diese Serie vernachlässigen. Für die Wellenlänge  $\lambda_m$  für die Übergänge  $md \rightarrow 3p$  gilt

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

Hierbei sind die Rydbergenergie  $E_{Ry} = -13.605\text{eV}$ , sowie  $hc = 1.2398 \cdot 10^3\text{nm eV}$  als Konstanten gesetzt.  $E_{3p}$  können wir mit dieser Formel berechnen, indem wir  $m = 3$  und entsprechend  $\lambda_3$  gleich der gemessenen Linie im Bereich von 819nm setzen. Die von uns gemessene Linie, welche am nächsten an dieser ist, liegt bei 817.7nm. Damit erhalten wir einen Wert von

$$E_{3p} = (-3.0279 \pm 0.0019)\text{eV}. \quad (8)$$

Setzen wir nun diese Energie und die Quantenzahl  $m$  im Bereich von 3 bis 12 in die obige Formel ein, so erhalten wir die erwarteten Wellenlängen für die jeweiligen Übergänge. Diese sind in Tabelle (2) zu sehen, gemeinsam mit der zugeordneten Spektrallinie aus unseren Beobachtungen, sowie der jeweiligen Abweichung.

Tabelle 2: Vergleich der berechneten und gemessenen Wellenlängen der 1. Nebenserie  $md \rightarrow 3p$ .

$m$	$\lambda_{\text{theo.}} [\text{nm}]$	$\lambda_{\text{beob.}} [\text{nm}]$	Abweichung
3	$817.7 \pm 1.000$	$817.7 \pm 1.0$	$0.01\sigma$
4	$569.4 \pm 0.5$	$567.1 \pm 1.0$	$2.03\sigma$
5	$499.2 \pm 0.4$	$496.9 \pm 2.0$	$1.13\sigma$
6	$467.9 \pm 0.4$	$469.2 \pm 1.0$	$1.28\sigma$
7	$450.8 \pm 0.4$	$450.0 \pm 1.0$	$0.77\sigma$
8	$440.4 \pm 0.3$	$438.1 \pm 1.0$	$2.20\sigma$
9	$433.5 \pm 0.3$	$432.6 \pm 1.0$	$0.88\sigma$
10	$428.7 \pm 0.3$	$429.2 \pm 1.0$	$0.46\sigma$
11	$425.3 \pm 0.3$	$426.0 \pm 1.0$	$0.72\sigma$
12	$422.7 \pm 0.3$	$423.1 \pm 1.0$	$0.44\sigma$

### *Erwartete Linien für die 2. Nebenserie $ms \rightarrow 3p$*

Bei der Betrachtung der 2. Nebenserie können wir den Korrekturterm  $\Delta_s$  nicht vernachlässigen. Die Bindungsenergie des Grundzustandes  $E_{3s}$  berechnen wir nach der Formel

$$E_{3s} = E_{3p} - \frac{hc}{\lambda}. \quad (9)$$

Für die Wellenlänge  $\lambda$  setzen wir die beobachtete Wellenlänge der D-Linie ein, da diese dem Übergang  $3p \rightarrow 3s$  entspricht. In unseren Aufzeichnungen können wir diese Linie bei in etwa  $\lambda = (590.5 \pm 1.0)\text{nm}$  beobachten. Damit erhalten wir eine Bindungsenergie von

$$E_{3s} = (-5.127 \pm 0.005)\text{eV}. \quad (10)$$

Durch Umstellen der Gleichung

$$E_{3s} = \frac{E_{Ry}}{(3 - \Delta_s)^2} \quad (11)$$

berechnen wir den Wert des Korrekturterms zu

$$\Delta_s = 1.3711 \pm 0.0007. \quad (12)$$

Die erwarteten Werte für die Wellenlängen der Linien können wir dann pro Quantenzahl  $m$  mit der Formel

$$\lambda_m \approx \frac{hc}{\frac{E_{Ry}}{(m - \Delta_s)^2} - E_{3p}} \quad (13)$$

berechnen. Die Werte für  $m$  im Bereich von 4 bis 9 sind in der Tabelle (3) wiederzufinden. Zur Berechnung der Fehler der  $\lambda_m$  haben wir hierbei, aufgrund der Komplexität der Formel, auf eine relative Fehlerbetrachtung zurückgegriffen.

Tabelle 3: Vergleich der berechneten und gemessenen Wellenlängen der 2. Nebenserie  $ms \rightarrow 3p$ .

$m$	$\lambda_{\text{theo.}} [\text{nm}]$	$\lambda_{\text{beob.}} [\text{nm}]$	Abweichung
4	$1170.4 \pm 1.4$	-	-
5	$621.5 \pm 0.7$	$622.3 \pm 1.0$	$0.67\sigma$
6	$518.1 \pm 0.6$	$517.5 \pm 1.0$	$0.53\sigma$
7	$477.1 \pm 0.6$	$474.0 \pm 1.0$	$2.76\sigma$
8	$456.1 \pm 0.6$	$454.4 \pm 1.0$	$1.52\sigma$
9	$443.7 \pm 0.5$	$445.3 \pm 1.0$	$1.42\sigma$

### Erwartete Linien für die Hauptserie $mp \rightarrow 3s$

Zu guter Letzt betrachten wir noch die erwarteten Linien der Hauptserie  $mp \rightarrow 3s$ . Hierzu benötigen wir den Korrekturterm  $\Delta_p$ , welchen wir ähnlich zur vorherigen Betrachtung durch Umstellen der Gleichung

$$E_{3p} = \frac{E_{Ry}}{(3 - \Delta_p)^2} \quad (14)$$

zu

$$\Delta_p = 0.8803 \pm 0.0007 \quad (15)$$

berechnen.

Mit diesem Wert berechnen wir nach der Gleichung

$$\lambda_m \approx \frac{hc}{\frac{E_{Ry}}{(m - \Delta_p)^2} - E_{3s}} \quad (16)$$

die erwarteten Wellenlängen der Linien mit den Quantenzahlen  $m = 4$  und  $5$ . Diese sind in der Tabelle (4) zu finden. Da wir den Bereich des Spektrums, in dem sich diese Linien befinden, nicht aufgezeichnet haben, können wir diese nicht mit beobachteten Werten vergleichen.

Tabelle 4: Berechneten Wellenlängen der Hauptserie  $mp \rightarrow 3s$ .

$m$	$\lambda_{\text{theo.}} [\text{nm}]$	$\lambda_{\text{beob.}} [\text{nm}]$	Abweichung
4	$332.4 \pm 0.6$	-	-
5	$286.6 \pm 0.5$	-	-

### 3.4 Bestimmung der Serienenergien und der l-abhängigen Korrekturfaktoren

Die Funktionen, welche wir bisher genutzt hatten, um die Wellenlängen der Spektrallinie berechnen, können wir umgekehrt auch verwenden um die Rydbergenergie, sowie die Korrekturterme zu berechnen. Dazu tragen wir die gemessenen Wellenlängen der ersten Nebenserie in ein Diagramm auf und Fitten an diese Daten die Funktion

$$\lambda_m = \frac{hc}{\frac{E_{Ry}}{(m - \Delta_d)^2} - E_{3p}}. \quad (17)$$

Hierbei lassen wir  $E_{Ry}$ ,  $E_{3p}$  und  $\Delta_d$  freie Parameter, welche wir mit den Startwerten

$$E_{Ry} = -13.6 \text{ [eV]}, \quad E_{3p} = -3 \text{ [eV]}, \quad \Delta_d = -0.02 \quad (18)$$

vorbelegen. Abbildung (14) zeigt die aufgetragenen Messdaten sowie das grafische Resultat des Fits.

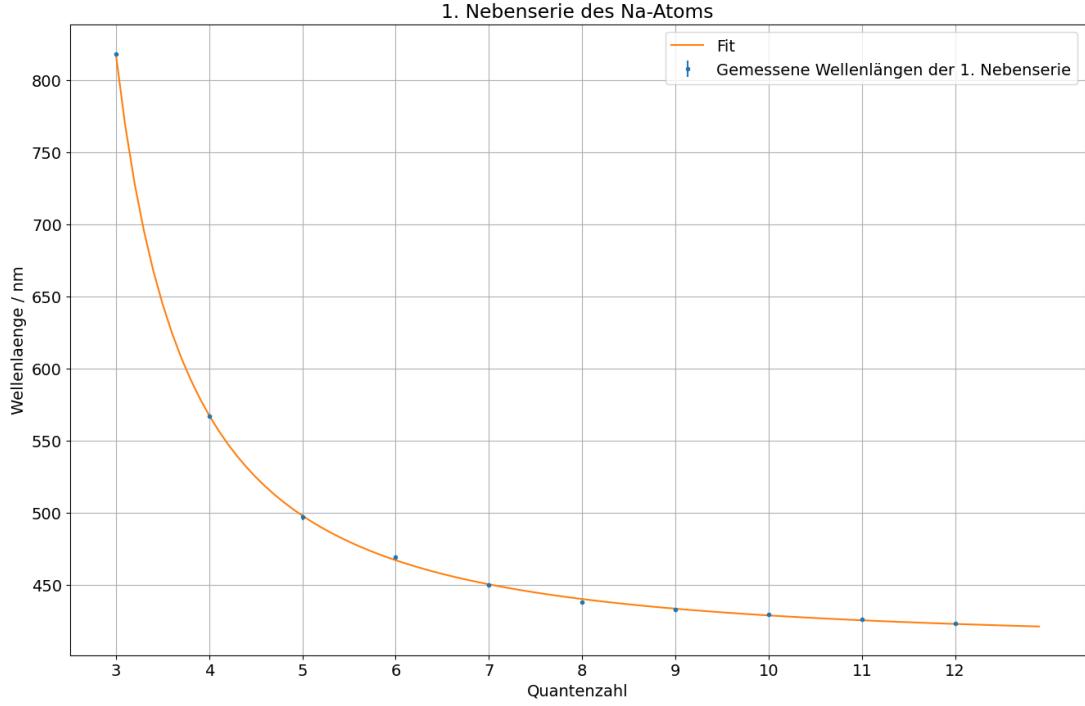


Abbildung 14: Resultat für den Fit an die Wellenlängen der 1. Nebenserie.

Die durch den Fit optimierten Werte der Parameter entsprechen

$$E_{Ry} = -13.0 \pm 0.5 \text{ [eV]}, \quad E_{3p} = -3.023 \pm 0.006 \text{ [eV]}, \quad \Delta_d = 0.07 \pm 0.05. \quad (19)$$

Die Güte des Fits bestimmen wir durch das Berechnen der  $\chi^2$ -Summe nach

$$\chi^2 = \sum_i^N \left( \frac{\text{Funktionswert}_i - \text{Messwert}_i}{\text{Fehler}_i} \right)^2, \quad (20)$$

sowie dem reduzierten  $\chi^2$ -Wert nach  $\chi^2_{\text{red}} = \chi^2 / \# \text{Freiheitsgrade}$ .

Aus dem gezeigten Fit erhalten wir für diese Werte von

$$\chi^2 = 10.53, \quad \chi^2_{\text{red}} = 1.50. \quad (21)$$

Für die Fitwahrscheinlichkeit erhalten wir einen Wert von 16.0%.

Den gleichen Prozess führen wir nun noch einmal für die 2. Nebenserie durch. Dafür fitten wir die Funktion

$$\lambda_m = \frac{hc}{\frac{E_{Ry}}{(m-\Delta_s)^2} - E_{3p}} \quad (22)$$

an die gemessenen Wellenlängen. Die Variablen initialisieren wir erneut mit den Startwerten

$$E_{Ry} = -13.6 \text{ [eV]}, \quad E_{3p} = -3 \text{ [eV]}, \quad \Delta_s = 1.5. \quad (23)$$

Das grafische Resultat ist in Abbildung (15) zu sehen.

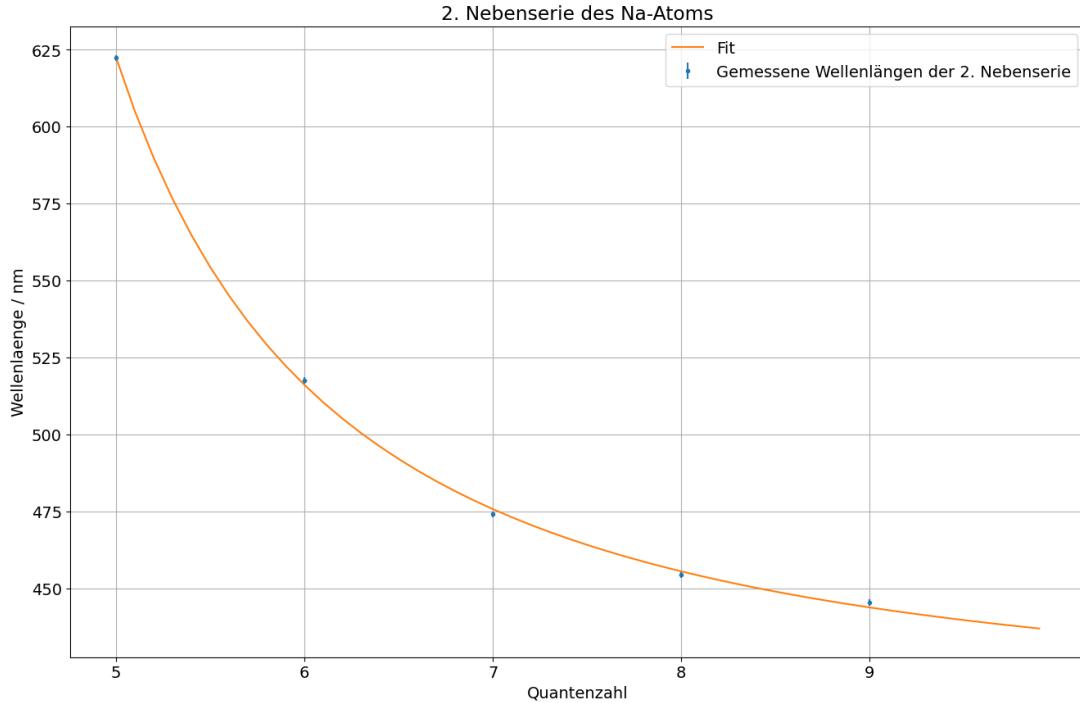


Abbildung 15: Resultat für den Fit an die Wellenlängen der 2. Nebenserie.

Wir erhalten die folgenden optimierten Werte für die Variablen:

$$E_{Ry} = -11.6 \pm 2.1 \text{ [eV]}, \quad E_{3p} = -3.01 \pm 0.04 \text{ [eV]}, \quad \Delta_s = 1.62 \pm 0.25. \quad (24)$$

Die Güte des Fits wird mit

$$\chi^2 = 8.55, \quad \chi^2_{\text{red}} = 4.28. \quad (25)$$

bewertet und die Fitwahrscheinlichkeit beläuft sich auf einen Wert von 1.0%.

## 4 Zusammenfassung und Diskussion

In Versuch 234 beschäftigen wir uns mit der Untersuchung der Spektren verschiedener Lichtquellen. Hier unterscheiden wir zwischen Temperaturstrahlern und Nichttemperaturstrahlern. Temperaturstrahler, wie beispielsweise eine Glühlampe, basieren auf dem Phänomen, dass jeder Körper, dessen Temperatur größer als 0K ist, elektromagnetische Strahlung abstrahlt, deren Intensitätsverteilung abhängig von der Wellenlänge dem Planck'schen Strahlungsgesetz folgt. Bei den Nichttemperaturstrahlern ist die Erzeugung von Licht entweder auf Anregung von Atomzuständen, zum Beispiel bei einer Natriumdampflampe, oder der Rekombination von Elektron-Loch-Paaren in Halbleitern, also LEDs, zurückzuführen.

Im ersten Versuchsteil untersuchten wir das Spektrum des wohl bekanntesten Temperaturstrahlers, der Sonne. Mit einem Gitterspektroskop nahmen wir das Spektrum des Tageslichts einmal direkt und einmal durch ein Fenster auf. Wir konnten beobachten, dass es sich um ein kontinuierliches Spektrum handelt, wie es von Temperaturstrahlern zu erwarten ist.

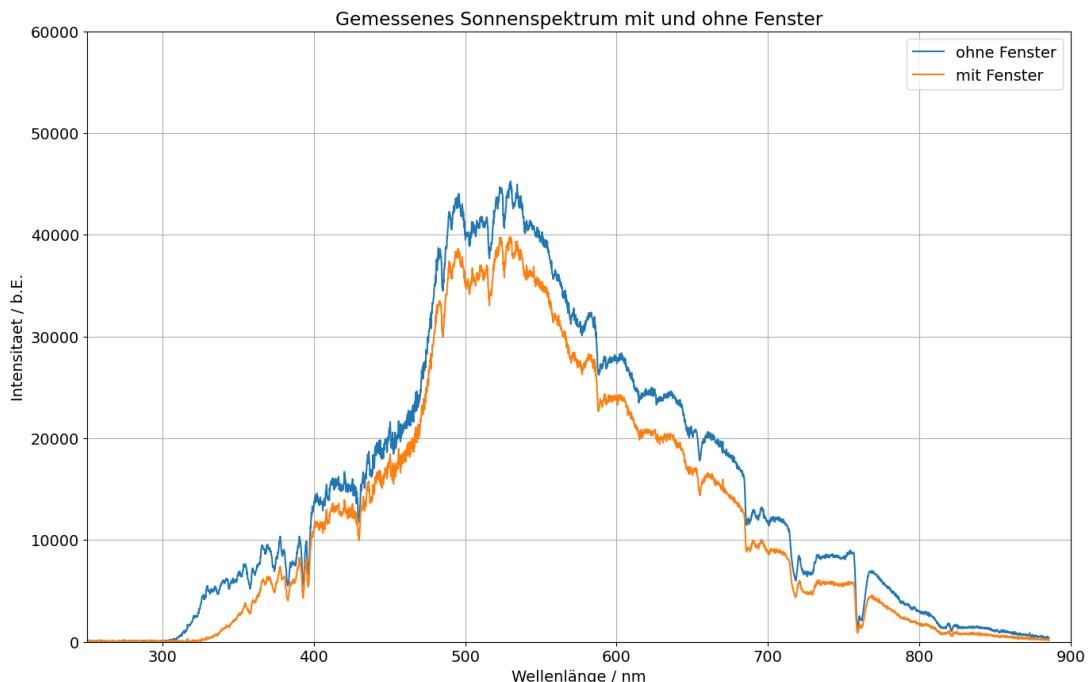


Abbildung 16: Tageslichtspektrum direkt (blau) und durch Fensterglas (orange).

Im Vergleich des Spektrums durch das Fenster mit dem, das direkt aufgezeichnet wurde, konnten wir beobachten, dass durch die Scheibe über den gesamten Wellenlängenbereich hinweg die Intensität des Lichts durch die Absorption der Glasscheibe abgeschwächt wird. Bei genauerer Betrachtung der Absorption konnten wir sehen, dass diese im Bereich von Wellenlängen unter 400nm, also im nicht-sichtbaren UV-Bereich, am höchsten und im Bereich des sichtbaren Lichts am niedrigsten ist.

Die vielen im Spektrum deutlich sichtbaren lokalen Minima sind durch die Absorption von Licht bestimmter Wellenlängen in den Atmosphärenschichten der Sonne und der Erde zu erklären. Diese Absorptionslinien, genannt „Fraunhoferlinien“ sind speziellen Wellenlängen zugeordnet, welche wir mit den Positionen der Linien in unseren aufgezeichneten Spektrum verglichen. Die Werte sind, mit der jeweiligen Abweichung vom Literaturwert

in Tabelle (5) zusammengefasst. Als Fehler für die beobachteten Wellenlängen sind wir jeweils von  $\pm 1\text{nm}$  ausgegangen.

Tabelle 5: Vergleich der erwarteten und gemessenen Wellenlängen der Fraunhofer- und Balmerlinien

Linie	Literaturwert [nm]	Abgelesener Wert [nm]	Abweichung [ $\sigma$ ]
K	393.4	393.0	0.4
H	396.8	396.1	0.7
G	430.8	429.8	1.0
F	486.1	485.2	0.91
b1	518.4	516.7	1.7
E	527.0	526.2	0.8
D3	587.6	588.4	0.8
D2	589.0	589.0	0.0
D1	589.6	589.7	0.11
C	656.3	655.0	1.3
B	686.7	686.7	0.0
A	759.4	759.4	0.0
$H_\alpha$	656.3	655.0	1.3
$H_\beta$	486.1	485.2	0.91
$H_\gamma$	434.0	433.4	0.61
$H_\delta$	410.1	409.5	0.61

Unter anderem als lokale Minima im Sonnenlichtspektrum zu beobachten sind die Linien der Balmerserie, welche auf Anregungen in Wasserstoffatom zurückzuführen sind. Die beobachteten Positionen der  $H_\alpha$ –,  $H_\beta$ –,  $H_\gamma$ –,  $H_\delta$ –Linien der Balmerserie verglichen wir ebenfalls mit den Literaturwerten, zusammengefasst in derselben Tabelle.

Insgesamt ist die Abweichung der beobachteten Absorptionslinien von den Literaturwerten sehr gering. Die dennoch sichtbaren Unterschieden sind vermutlich zu Großteilen auf das wolkige Wetter am Versuchstag, sowie Störungen und Reflexionen durch die umliegenden Gebäude zurückzuführen.

Im darauf folgenden Versuchsteil betrachteten wir qualitativ die Spektren verschiedener Lichtquellen. Hierunter untersuchten wir das Licht verschiedenfarbiger LEDs, eines Lasers und einer Energiesparlampe als Beispiele für Nichttemperaturstrahler, sowie das einer Glühlampe als ein klassisches Beispiel für einen Temperaturstrahler. Wie in der Theorie beschrieben, konnten wir beobachten, dass die LEDs, der Laser und die Energiesparlampe diskrete Spektren aufweisen. Dabei basiert die Erzeugung von weißem Licht bei den weißen LEDs und der Energiesparlampe auf verschiedenen Methoden. Die Glühlampe wies ein kontinuierliches Spektrum auf, von welchem große Teile außerhalb des sichtbaren Wellenlängenbereichs lagen. Dies zeigte auch, dass diese im Vergleich zu den anderen Lichtquellen viel weniger energieeffizient ist.

Im abschließenden großen Versuchsblock setzen wir uns mit dem Spektrum einer Natriumdampflampe auseinander. Als Beispiel für eine Gasentladungslampe weist diese ein diskretes Spektrum auf, welches durch eine Vielzahl an Spektrallinien über den gesamten beobachteten Wellenlängen charakterisiert ist. Wir betrachteten zunächst das Spektrum

in einem Wellenlängenbereich von 350 bis 550nm. Hier sind Spektrallinien mit geringer Intensität zu beobachten. Wir notierten die Wellenlängen aller gut sichtbarer Linien für den späteren Vergleich. Im Bereich zwischen 590 und 600nm befindet sich im Natriumspektrum die markante D-Linie. Auch die Wellenlänge dieser Linie selbst, sowie die Wellenlängen einiger umliegender Linien notierten wir uns. Zuletzt zeichneten wir noch die Wellenlängen einiger markanter Linien im oberen Wellenlängenbereich von 650 bis 850nm auf.

Zum Vergleich der beobachteten mit den theoretisch vorhersagten Positionen der Spektrallinien betrachteten wir dann die Übergänge  $md \rightarrow 3p$  der 1. Nebenserie, die Übergänge  $ms \rightarrow 3p$  der 2. Nebenserie, sowie die  $mp \rightarrow 3s$  der Hauptserie. Die Energie  $E_{3p}$ , welche wir für die Berechnungen der Wellenlängen der Übergänge zum  $3p$ -Zustand benötigen, bestimmten wir auf einen Wert von

$$E_{3p} = -3.0279 \pm 0.0019\text{eV}. \quad (26)$$

Mit diesem bestimmten wir anhand der Formel  $\lambda_m \approx hc / \left( \frac{E_{Ry}}{m^2} - E_{3p} \right)$  für die Quantenzahlen  $m = 3, \dots, 12$  die erwarteten Wellenlängen der Spektrallinien. Diese sind in Tabelle (6) zusammen mit den von uns beobachteten Wellenlängen, welche wir diesen zuordnen konnten, aufgelistet.

Tabelle 6: Vergleich der berechneten und gemessenen Wellenlängen der 1. Nebenserie  $md \rightarrow 3p$ .

$m$	$\lambda_{\text{theo.}} [\text{nm}]$	$\lambda_{\text{beob.}} [\text{nm}]$	Abweichung
3	$817.7 \pm 1.000$	$817.7 \pm 1$	$0.01\sigma$
4	$569.4 \pm 0.5$	$567.1 \pm 1$	$2.03\sigma$
5	$499.2 \pm 0.4$	$496.9 \pm 2$	$1.13\sigma$
6	$467.9 \pm 0.4$	$469.2 \pm 1$	$1.28\sigma$
7	$450.8 \pm 0.4$	$450.0 \pm 1$	$0.77\sigma$
8	$440.4 \pm 0.3$	$438.1 \pm 1$	$2.20\sigma$
9	$433.5 \pm 0.3$	$432.6 \pm 1$	$0.88\sigma$
10	$428.7 \pm 0.3$	$429.2 \pm 1$	$0.46\sigma$
11	$425.3 \pm 0.3$	$426.0 \pm 1$	$0.72\sigma$
12	$422.7 \pm 0.3$	$423.1 \pm 1$	$0.44\sigma$

Wir können sehen, dass die Abweichung bei den meisten Linien unter einem  $\sigma$  liegt. Bei den Vergleichen mit einer signifikanteren Abweichung kann es durchaus sein, dass den theoretisch vorhersagten Linien falsche beobachtete Linien zugeordnet wurden.

Analog gingen wir für die 2. Nebenserie vor. Für die Berechnung der Wellenlängen benötigen wir hierbei zusätzlich den Korrekturterm  $\Delta_s$ . Um diesen zu bestimmen, berechneten wir zunächst die Energie des Zustandes  $3s$  zu

$$E_{3s} = (-5.127 \pm 0.005)\text{eV}, \quad (27)$$

und daraus wiederum den Korrekturterm

$$\Delta_s = 1.3711 \pm 0.0007. \quad (28)$$

Dieser ging dann mit in die Formel  $\lambda_m \approx hc / \left( \frac{E_{Ry}}{(m - \Delta_s)^2} - E_{3p} \right)$  mit ein, um die Wellenlängen für die Quantenzahlen  $m = 4, \dots, 9$  zu bestimmen. Die Resultate sind in Tabelle (7) noch einmal wiedergegeben. Da diese außerhalb des untersuchten Bereichs liegt, konnten wir der Spektrallinie für den Übergang  $3s \rightarrow 3p$  keine beobachtete Wellenlänge zuordnen. Bei allen weiteren sehen wir erneut, dass die Abweichung zwischen den jeweils zugeordneten Linien weitestgehend gering ausfällt. Lediglich die Linie der Quantenzahl  $m = 7$  weicht etwas stärker ab, was möglicherweise auf eine falsche Zuordnung zurückzuführen ist.

Tabelle 7: Vergleich der berechneten und gemessenen Wellenlängen der 2. Nebenserie  $ms \rightarrow 3p$ .

$m$	$\lambda_{\text{theo.}} [\text{nm}]$	$\lambda_{\text{beob.}} [\text{nm}]$	Abweichung
4	$1170.366 \pm 1.302$	-	-
5	$621.527 \pm 0.692$	$622.34 \pm 1$	$0.67\sigma$
6	$518.112 \pm 0.577$	$517.50 \pm 1$	$0.53\sigma$
7	$477.124 \pm 0.531$	$474.00 \pm 1$	$2.76\sigma$
8	$456.100 \pm 0.508$	$454.40 \pm 1$	$1.52\sigma$
9	$443.719 \pm 0.494$	$445.30 \pm 1$	$1.42\sigma$

Noch einmal nach dem gleichen Schema berechneten wir abschließend die erwarteten Wellenlängen der Hauptserie. Hierzu bestimmten wir zunächst den Korrekturterm

$$\Delta_p = 0.8803 \pm 0.0007. \quad (29)$$

Die berechneten Wellenlängen für die Quantenzahlen  $m = 4, 5$  sind in Tabelle (8) zu finden. Da beide unter 350nm liegen und damit außerhalb des von uns beobachteten Bereichs, können wir diese nicht vergleichen.

Tabelle 8: Berechnete Wellenlängen der Hauptserie  $mp \rightarrow 3s$ .

$m$	$\lambda_{\text{theo.}} [\text{nm}]$	$\lambda_{\text{beob.}} [\text{nm}]$	Abweichung
4	$332.4 \pm 0.6$	-	-
5	$286.6 \pm 0.5$	-	-

Im letzten Teil der Auswertung passten wir die Funktion

$$f(m; E_{Ry}, E_{3p}, \Delta_{d(s)}) = \frac{hc}{\frac{E_{Ry}}{(m - \Delta_{d(s)})^2} - E_{3p}} = \lambda_m \quad (30)$$

abhängig von der Quantenzahl  $m$  an die Werte der beobachteten Wellenlängen der 1. und 2. Nebenserie an, um die Parameter  $E_{Ry}$ ,  $E_{3p}$  und  $\Delta_d$  bzw.  $\Delta_s$  zu bestimmen. Die optimierten Parameter aus der Anpassung an die 1. Nebenserie sind in Tabelle (9) zusammengefasst. Zusätzlich ist hier auch die Abweichung zu den Werten, welche wir im vorherigen Aufgabenteil berechnet hatten, angegeben.

Tabelle 9: Werte der Parameter, bestimmt durch Fit an die Wellenlängen der 1. Nebenserie, und Vergleich.

Parameter	Wert	Fehler	Abw. vom zuvor berechneten Wert
$E_{Ry}$	-13.0eV	0.5eV	$1.53\sigma$
$E_{3p}$	-3.023eV	0.006eV	$0.75\sigma$
$\Delta_d$	0.07	0.05	-

Die reduzierte  $\chi^2$ -Summe liegt mit einem Wert von  $\chi^2_{\text{red}} = 1.50$  nahe am optimalen Wert 1. Somit können wir bei diesen Fit die Beschreibung der Daten durch das gegebene Modell bzw. die gegebene Funktion allgemein als gut bewerten. Dies ist auch durch relativ geringen Unsicherheiten der optimierten Parameter zu erkennen. Die Fitwahrscheinlichkeit liegt allerdings nur bei 16.0%. Wie aber auch in der Praktikumsanleitung angeführt, liegt dies daran, dass  $\Delta_d$  nur eine empirische Näherungsformel ist und in Wirklichkeit auch schwach von der Hauptquantenzahl  $n$  abhängt. Die Abweichungen von den zuvor berechneten Werten sind mit Werten von um einem  $\sigma$  nicht ausschlaggebend.

Im Anschluss führten wir noch einen Fit an die Wellenlängen der 2. Nebenserie durch. Hierbei optimierten wir anstatt dem Korrekturterm  $\Delta_d$ , den Term  $\Delta_s$ , welchen wir auch bereits im vorherigen Aufgabenteil verwendet hatten. Die optimierten Werte der Parameter sind, in gleicher Form wie zuvor, in Tabelle (10) zusammengefasst.

Tabelle 10: Werte der Parameter, bestimmt durch Fit an die Wellenlängen der 2. Nebenserie, und Vergleich.

Parameter	Wert	Fehler	Abw. vom zuvor berechneten Wert
$E_{Ry}$	-11.6eV	2.1eV	$1.01\sigma$
$E_{3p}$	-3.01eV	0.04eV	$0.68\sigma$
$\Delta_s$	1.62	0.25	$1.03\sigma$

Für diesen Fit hatten wir nur halb so viele Datenpunkte wie im vorherigen Fit zur Verfügung. Dies spiegelt sich auch in den nun sehr hohen Ungenauigkeiten der optimierten Parameter, sowie der Bewertung der Güte des Fits wider. Die reduzierte  $\chi^2$ -Summe beträgt hier  $\chi^2_{\text{red}} = 4.28$ , was für eine schlechte Beschreibung der Daten durch das verwendete Modell steht. Mit gerade einmal 1% ist auch hier die Fitwahrscheinlichkeit sehr schlecht, was erneut auf das zum vorherigen Fit beschriebene Problem zurückzuführen sein könnte. Die Abweichungen zu den zuvor berechneten Werten der optimierten Parameter sind in diesem Fit geringer als im vorherigen, was allerdings daran liegt, dass die Unsicherheiten hier deutlich größer sind.

# Python Code, Hauptprogramm

## 234auswertung

April 1, 2025

```
[25]: import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import numpy as np
from scipy.signal import argrelextrema
from scipy.optimize import curve_fit
from scipy.stats import chi2

plt.rcParams.update({'font.size': 14})

%run ../lib.ipynb

class Consts:
    hc = 1.2398 * 10**3 #nm eV
    E_Ry = -13.605 # eV
```

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

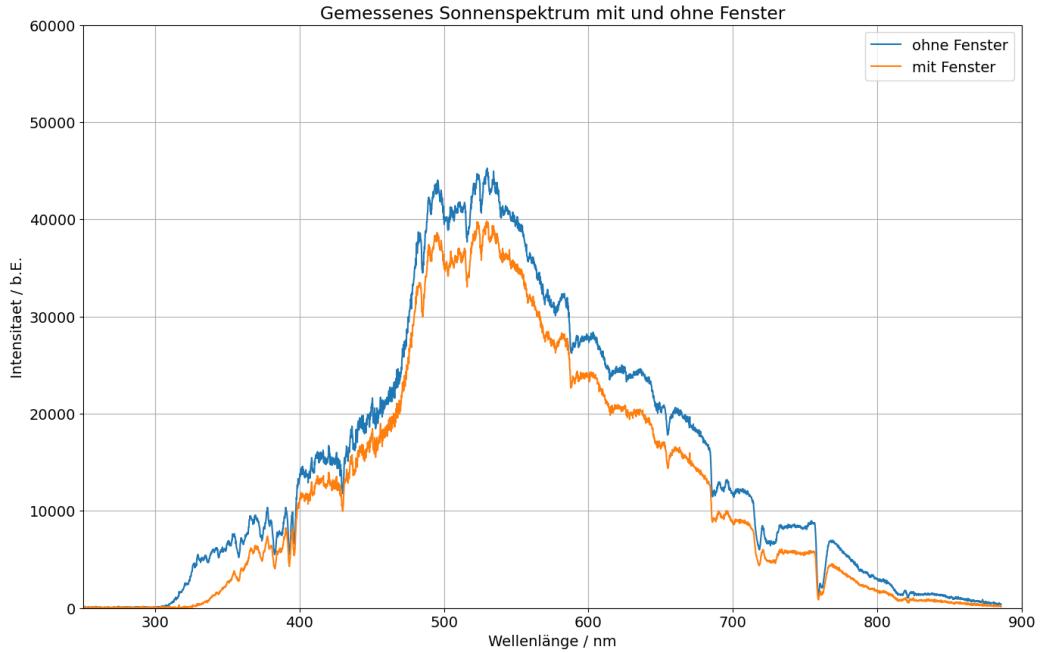
### 0.0.1 Sonnenspektrum und Fraunhoferlinien

```
[27]: lamb_og, inten_og = np.loadtxt('sonnenlichtspektrum_ohne_glas.txt', skiprows=17,
                                   converters= {0:comma_to_float, 1:comma_to_float},
                                   comments='>', unpack=True)

lamb_mg, inten_mg=np.loadtxt('sonnenlichtspektrum_durch_glas.txt', skiprows=17,
                           converters= {0:comma_to_float, 1:comma_to_float},
                           comments='>', unpack=True)
```

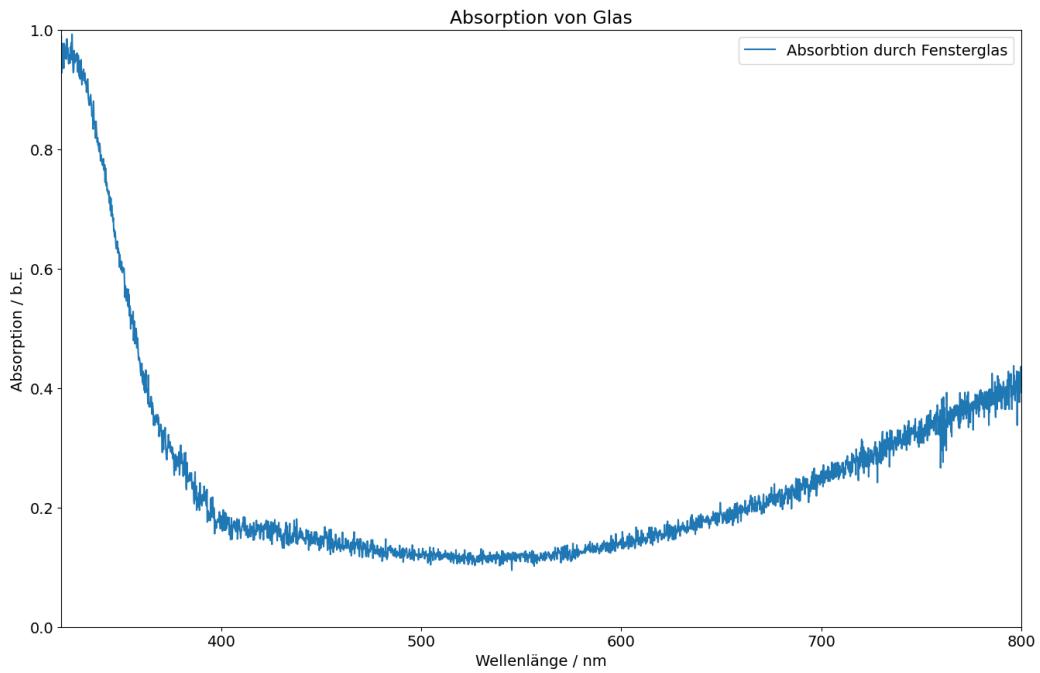
```
[28]: plt.figure(figsize=(16,10))
plt.plot(lamb_og, inten_og, label='ohne Fenster')
plt.plot(lamb_mg, inten_mg, label='mit Fenster')
plt.title('Gemessenes Sonnenspektrum mit und ohne Fenster')
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Intensität / b.E.')
plt.legend()
plt.grid()
plt.ylim((0,60000))
plt.xlim((250,900))
```

```
plt.savefig('out/himmel_m_o_g.png', format='png', bbox_inches='tight')
```



```
[29]: A = 1 - inten_mg[600:] / inten_og[600:]

plt.figure(figsize=(16,10))
plt.plot(lamb_mg[600:], A, label='Absorption durch Fensterglas')
plt.title('Absorption von Glas')
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Absorption / b.E.')
plt.ylim((0,1))
plt.xlim((320,800))
plt.legend()
plt.savefig("out/absorption_glas.png", format='png', bbox_inches='tight')
```



```
[30]: # lokale minima, abgelesen aus textdatei:
mins = [
    ('K', 393.0, 1, 5533.61, 393.4),
    ('H', 396.1, 1, 6619.03, 396.8),
    ('G', 429.8, 1, 11786.06, 430.8),
    ('F', 485.2, 1, 34493.94, 486.1),
    ('b1', 516.7, 1, 38450.61, 518.4),
    ('E', 526.2, 1, 41561.48, 527.0),
    ('D3', 588.4, 1, 26230.38, 587.6),
    ('D2', 589.0, 1, 26549.87, 589.0),
    ('D1', 589.7, 1, 27166.31, 589.6),
    ('C', 655.0, 1, 17813.23, 656.3),
    ('B', 686.7, 1, 11865.93, 686.7),
    ('A', 759.4, 1, 1339.37, 759.4)
]

balmer_air_nist = [
    (r'$H_{\alpha}$', 655.0, 1, 656.3),
    (r'$H_{\beta}$', 485.2, 1, 486.1),
    (r'$H_{\gamma}$', 433.4, 1, 434.0),
    (r'$H_{\delta}$', 409.5, 1, 410.1),
]
```

```

plt.figure(figsize=(18,8))
plt.plot(lamb_og, inten_og)
plt.title('Tageslichtspektrum (+ Fraunhoferlinien & Balmer-Serie)')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.ylim((0,55000))
plt.xlim((350,800))
i = 0
for m in mins:
    print(f'{m[0]}\t :: expected = {m[4]}nm, measured = {m[1]}nm, diff = {ValErr(m[1], m[2]).sigmadiff_fmt(ValErr(m[4], 0))}')
    plt.axvline(x=m[1], linewidth=2, linestyle='--', color='orange')
    plt.text(x=m[1] + (5 * (-1 if i % 2 == 0 else 1)), y=(m[3]+9000), s=m[0], ha='center', fontsize=12)
    i = i + 1
# plt.xticks(np.arange(350, 801, 5))

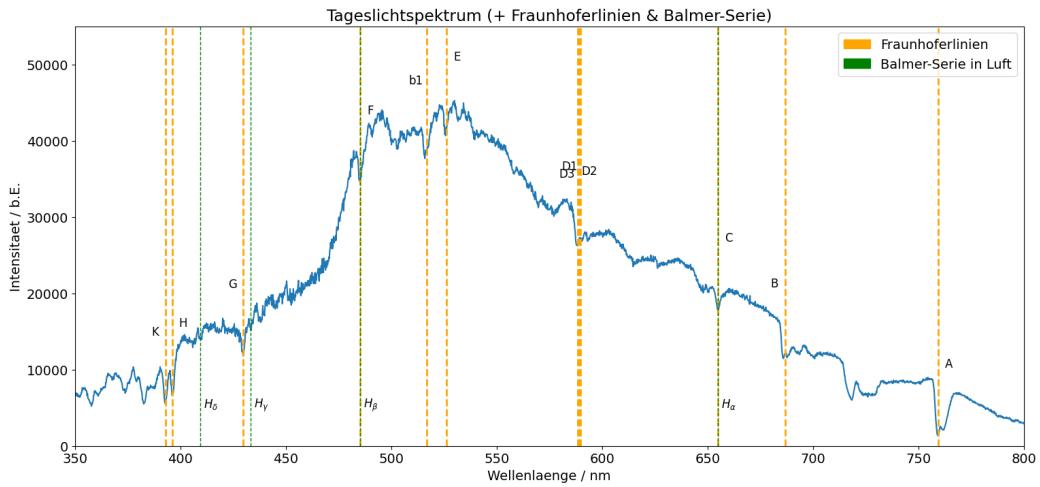
for b in balmer_air_nist:
    print(f'{b[0]}\t :: expected = {b[3]}nm, measured = {b[1]}nm, diff = {ValErr(b[1], b[2]).sigmadiff_fmt(ValErr(b[3], 0))}')
    plt.axvline(x=b[1], linewidth=1, linestyle='--', color='green')
    plt.text(x=b[1] + 5, y=5000, s=b[0], ha='center', fontsize=12)

fraunhofer_patch = mpatches.Patch(color='orange', label='Fraunhoferlinien')
balmer_air_patch = mpatches.Patch(color='green', label='Balmer-Serie in Luft')

plt.legend(handles=[fraunhofer_patch, balmer_air_patch])
plt.savefig("out/spektrum_fraunhofer_balmer.png", format='png', bbox_inches='tight')

```

K	:: expected = 393.4nm, measured = 393.0nm, diff = 0.4
H	:: expected = 396.8nm, measured = 396.1nm, diff = 0.7
G	:: expected = 430.8nm, measured = 429.8nm, diff = 1.0
F	:: expected = 486.1nm, measured = 485.2nm, diff = 0.91
b1	:: expected = 518.4nm, measured = 516.7nm, diff = 1.7
E	:: expected = 527.0nm, measured = 526.2nm, diff = 0.8
D3	:: expected = 587.6nm, measured = 588.4nm, diff = 0.8
D2	:: expected = 589.0nm, measured = 589.0nm, diff = 0.0
D1	:: expected = 589.6nm, measured = 589.7nm, diff = 0.11
C	:: expected = 656.3nm, measured = 655.0nm, diff = 1.3
B	:: expected = 686.7nm, measured = 686.7nm, diff = 0.0
A	:: expected = 759.4nm, measured = 759.4nm, diff = 0.0
\$H_\alpha\$	:: expected = 656.3nm, measured = 655.0nm, diff = 1.3
\$H_\beta\$	:: expected = 486.1nm, measured = 485.2nm, diff = 0.91
\$H_\gamma\$	:: expected = 434.0nm, measured = 433.4nm, diff = 0.61
\$H_\delta\$	:: expected = 410.1nm, measured = 409.5nm, diff = 0.61



### 0.0.2 Auswertung des Natriumspektrums

[31]: # Aufgabe 3

```

lamb_na_a3_500, inten_na_a3_500=np.loadtxt('na_a3_500nm_saett.txt', skiprows=17,
                                             converters= {0:comma_to_float, 1:comma_to_float},
                                             comments='>', unpack=True)

plt.figure(figsize=(18,10))
plt.plot(lamb_na_a3_500[586:], inten_na_a3_500[586:])
plt.title('Natriumspektrum im niedrigen bis mittleren Wellenlängenbereich')
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Intensität / b.E.')
plt.yscale('log')
plt.ylim((500,60000))
plt.xlim((350,550))
plt.xticks(np.arange(350, 550, 10))
plt.grid()

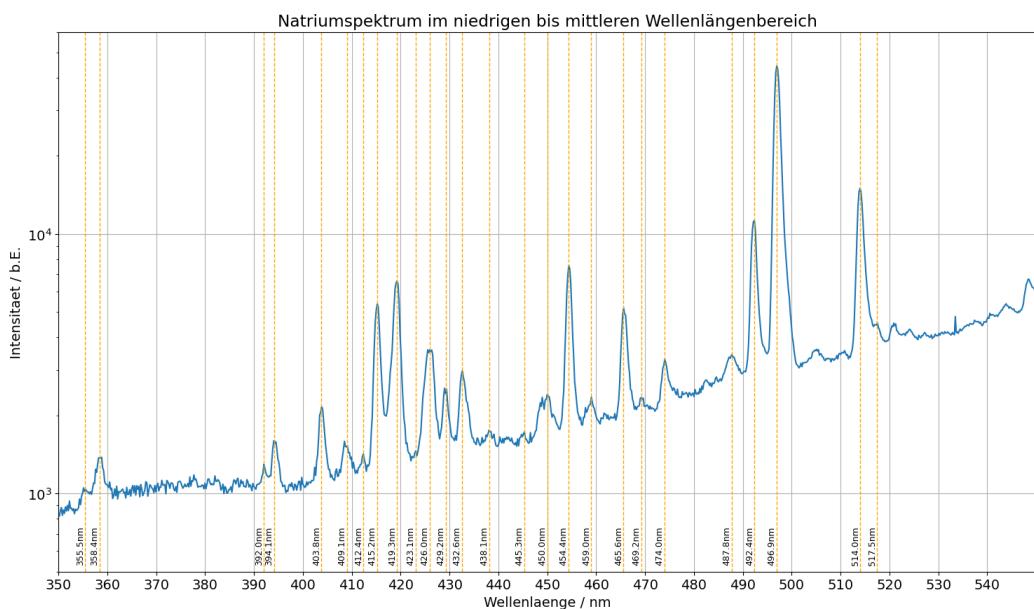
13 = [(355.5, 1, 1039.16),
       (358.4, 1, 1367.92),
       (392.0, 1, 1295.52),
       (394.1, 1, 1598.57),
       (403.8, 1, 2159.1),
       (409.1, 1, 1526.74),
       (412.4, 1, 1418.91),
       (415.2, 1, 5372.36),
       (419.3, 1, 6594.08),
       (423.1, 1, 1462.68),

```

```
(426.0, 1, 3580.95),
(429.2, 1, 2424.05),
(432.6, 1, 2966.71),
(438.1, 1, 1753.96),
(445.3, 1, 1729.02),
(450.0, 1, 2404.7),
(454.4, 1, 7544.74),
(459.0, 1, 2348.89),
(465.6, 1, 5187.63),
(469.2, 1, 2338.7),
(474.0, 1, 3294.8),
(487.8, 1, 3435.13),
(492.4, 1, 11274.4),
(496.9, 2, 44422.62),
(514.0, 1, 14992.8),
(517.5, 1, 4546.21)]
```

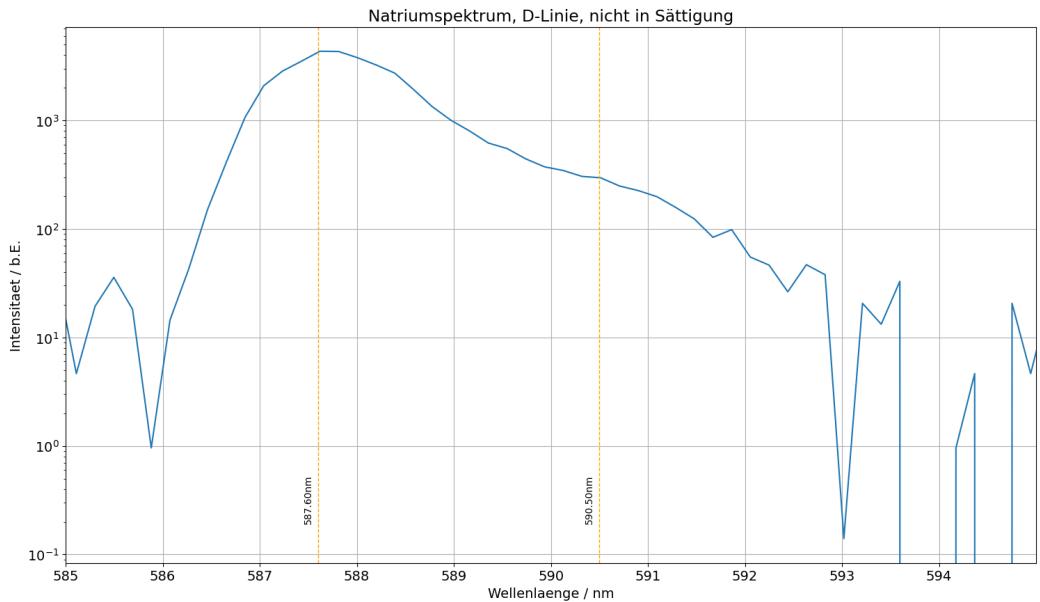
```
for l in l3:
    plt.axvline(x=l[0], linewidth=1, linestyle='--', color='orange')
    #plt.axvline(x=l[0]+l[1], linewidth=0.5, linestyle=':', color='gray')
    #plt.axvline(x=l[0]-l[1], linewidth=0.5, linestyle=':', color='gray')
    plt.text(x=l[0]-1, y=540, rotation=90, s=f'{l[0]:0.1f}nm', ha='center', fontstyle='italic', color='black')

plt.savefig("out/na_spek_350_550.png", format='png', bbox_inches='tight')
```



```
[32]: # Aufgabe 4, D-Linie nicht in Sättigung
```

```
lamb_na_a4_589_ks, inten_na_a4_589_ks=np.loadtxt('na_a4_589nm_keine_saett.txt',  
    skiprows=17,  
    converters= {0:comma_to_float, 1:comma_to_float},  
    comments='>', unpack=True)  
  
plt.figure(figsize=(18,10))  
plt.plot(lamb_na_a4_589_ks, inten_na_a4_589_ks)  
plt.title('Natriumspektrum, D-Linie, nicht in Sättigung')  
plt.xlabel('Wellenlänge / nm')  
plt.ylabel('Intensität / b.E.')  
plt.yscale('log')  
#plt.ylim((0,60000))  
plt.xlim((585,595))  
plt.xticks(np.arange(585, 595, 1))  
plt.grid()  
  
l4_ks = [  
    (587.6, 1, 4340.19),  
    (590.5, 1, 295.86)  
]  
  
for l in l4_ks:  
    plt.axvline(x=l[0], linewidth=1, linestyle='--', color='orange')  
    #plt.axvline(x=l[0]+l[1], linewidth=1, linestyle=':', color='gray')  
    #plt.axvline(x=l[0]-l[1], linewidth=1, linestyle=':', color='gray')  
    plt.text(x=l[0]-0.1, y=0.2, rotation=90, s=f'{l[0]:0.2f}nm', ha='center',  
        fontsize=10, color='black')  
  
plt.savefig("out/na_spek_dlinie_nichtsaett.png", format='png',  
    bbox_inches='tight')
```



[33]: # Aufgabe 4, D-Linie in Sättigung

```

lamb_na_a4_589_s, inten_na_a4_589_s=np.loadtxt('na_a4_589nm_saett.txt',□
    skiprows=17,
    converters= {0:comma_to_float, 1:comma_to_float},
    comments='>', unpack=True)

plt.figure(figsize=(18,10))
plt.plot(lamb_na_a4_589_s, inten_na_a4_589_s)
plt.title('Natriumspektrum, D-Linie, in Sättigung')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.yscale('log')
plt.ylim((100,100000))
plt.xlim((565,630))
plt.grid()

l4_s = [
    (567.1, 1, 4787.3),
    (581.8, 1, 1004.7),
    (614.5, 1, 1062.04),
    (622.3, 1, 185.51)
]

for l in l4_s:
    plt.axvline(x=l[0], linewidth=1, linestyle='--', color='orange')

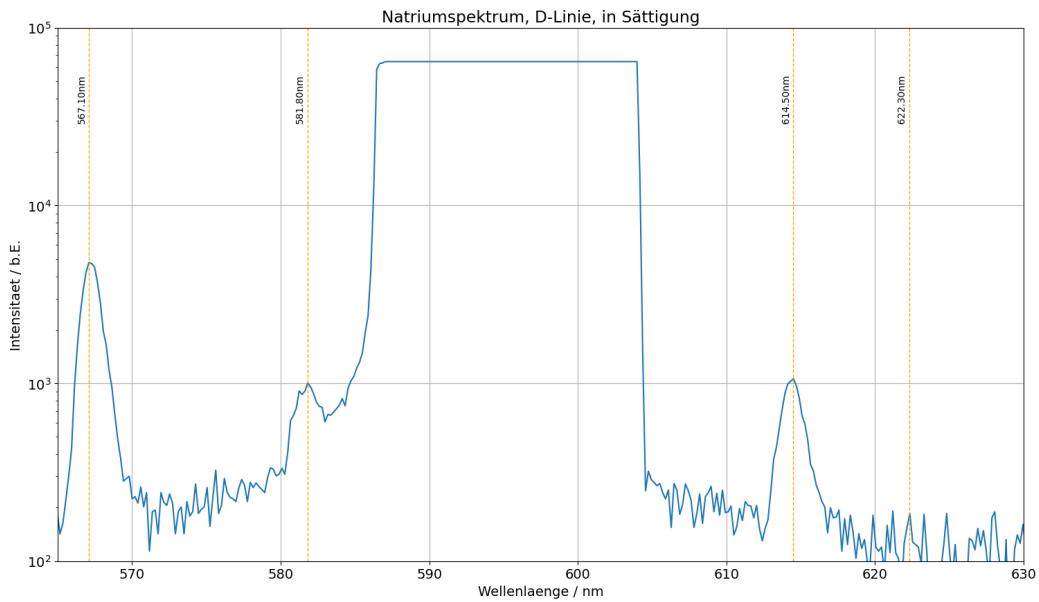
```

```

# plt.axvline(x=l[0]+l[1], linewidth=1, linestyle=':', color='gray')
# plt.axvline(x=l[0]-l[1], linewidth=1, linestyle=':', color='gray')
plt.text(x=l[0]-0.5, y=30000, rotation=90, s=f'{l[0]:0.2f}nm', ha='center', u
↪fontsize=10, color='black')

plt.savefig("out/na_spek_dlinie_saett.png", format='png', bbox_inches='tight')

```



[34]: # Aufgabe 5, Linien Hoher Wellenlänge

```

lamb_na_a5_650, inten_na_a5_650=np.loadtxt('na_a5_650nm_bis_850nm.txt', u
↪skiprows=17,
      converters= {0:comma_to_float, 1:comma_to_float},
      comments='>', unpack=True)

plt.figure(figsize=(18,10))
plt.plot(lamb_na_a5_650, inten_na_a5_650)
plt.title('Natriumspektrum, Spektrallinien im höheren Wellenlängenbereich')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.yscale('log')
plt.ylim((10,100000))
plt.xlim((650,850))
plt.grid()

l_a5 = [

```

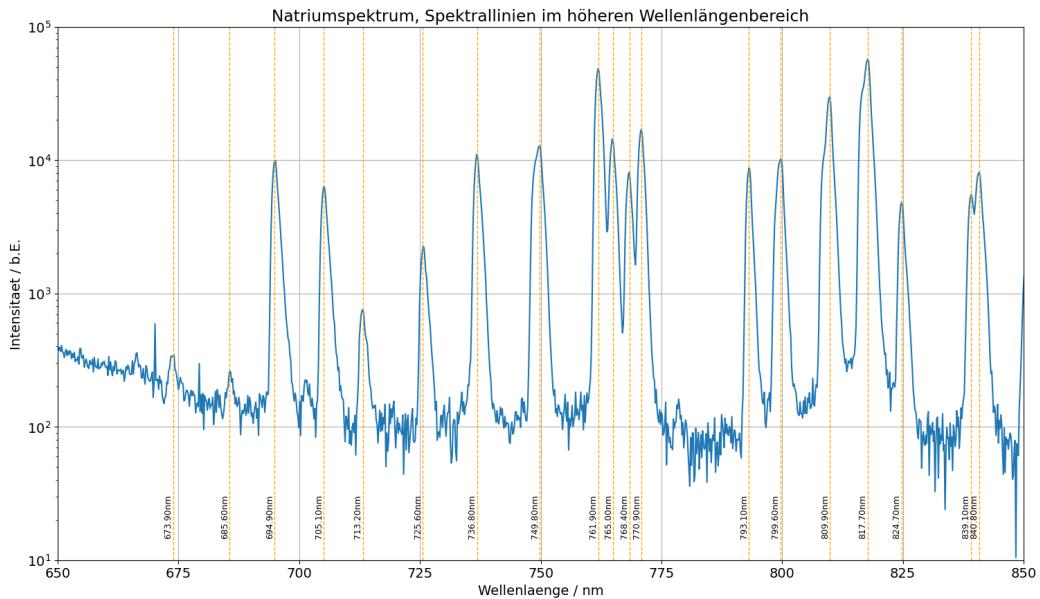
```

(673.9, 1, 338.15),
(685.6, 1, 264.02),
(694.9, 1, 9924.29),
(705.1, 1, 6364.51),
(713.2, 1, 755.53),
(725.6, 1, 2266.11),
(736.8, 1, 11056.81),
(749.8, 1, 12873.36),
(761.9, 1, 48696.02),
(765.0, 1, 14367.15),
(768.4, 1, 8108.97),
(770.9, 1, 17108.15),
(793.1, 1, 8694.69),
(799.6, 1, 10187.65),
(809.9, 1, 29784.67),
(817.7, 1, 57133.65),
(824.7, 1, 4831.81),
(839.1, 1, 5503.13),
(840.8, 1, 8105.69)
]

for l in l_a5:
    plt.axvline(x=l[0], linewidth=1, linestyle='--', color='orange')
    #plt.axvline(x=l[0]+l[1], linewidth=0.5, linestyle=':', color='gray')
    #plt.axvline(x=l[0]-l[1], linewidth=0.5, linestyle=':', color='gray')
    plt.text(x=l[0]-1, y=15, rotation=90, s=f'{l[0]:0.2f}nm', ha='center', fontsize=9, color='black')

plt.savefig("out/na_spek_650_850.png", format='png', bbox_inches='tight')

```



```
[35]: # Erwartete Linien für die 1. Nebenserie: md -> 3p
```

```
lambda_3 = ValErr(817.7, 1) #nm
E_3p = (Consts.E_Ry / 3**2) - (Consts.hc / lambda_3)
print(E_3p.strfmf(5,0,'E_3p'))

meas_1ns_vals = [
    817.7,
    567.1,
    496.9,
    469.2,
    450.0,
    438.1,
    432.6,
    429.2,
    426.0,
    423.1
]
meas_1ns_errs = np.array([1,1,2,1,1,1,1,1,1,1])
for m in range(3,13):
    l= Consts.hc / ((Consts.E_Ry / m**2) - E_3p)
```

```

print(f'm = {m:2d}, lambda = {l.strfmtf(3,0)}, lambda_measured =_
↪{meas_1ns_vals[m-3]):6.2f}, diff = {ValErr(meas_1ns_vals[m-3],_
↪meas_1ns_errs[m-3]).sigmadiff_fmt(1)}')

```

```

E_3p = -3.02787 ± 0.00185
m = 3, lambda = 817.700 ± 1.000, lambda_measured = 817.70, diff = 0.01
m = 4, lambda = 569.353 ± 0.485, lambda_measured = 567.10, diff = 2.03
m = 5, lambda = 499.181 ± 0.373, lambda_measured = 496.90, diff = 1.13
m = 6, lambda = 467.857 ± 0.327, lambda_measured = 469.20, diff = 1.28
m = 7, lambda = 450.801 ± 0.304, lambda_measured = 450.00, diff = 0.77
m = 8, lambda = 440.381 ± 0.290, lambda_measured = 438.10, diff = 2.2
m = 9, lambda = 433.510 ± 0.281, lambda_measured = 432.60, diff = 0.88
m = 10, lambda = 428.726 ± 0.275, lambda_measured = 429.20, diff = 0.46
m = 11, lambda = 425.254 ± 0.270, lambda_measured = 426.00, diff = 0.72
m = 12, lambda = 422.651 ± 0.267, lambda_measured = 423.10, diff = 0.44

```

[36]: *## Erwartete Linien für die 2. Nebensetie: ms → 3p*

```

E_3s = E_3p - (Consts.hc / ValErr(590.5, 1))
Delta_s_val = 3 - np.sqrt(Consts.E_Ry / E_3s.val)
Delta_s_err = (np.sqrt(np.abs(Consts.E_Ry)) / (2 * np.abs(E_3s.val) ** (3/2)))_
↪* E_3s.err
Delta_s = ValErr(Delta_s_val, Delta_s_err)

print_all(E_3s.strfmtf(5,0, 'E_3s'), Delta_s.strfmtf(5,0, 'Delta s'))

meas_2ns_vals = [
    622.34,
    517.5,
    474.0,
    454.4,
    445.3,
]
meas_2ns_errs = np.ones(5)

for m in range(4,10):
    l_val = Consts.hc / ((Consts.E_Ry / (m - Delta_s_val)**2) - E_3p.val)
    l_err = l_val * np.sqrt(E_3p.relerr()**2 + (2*Delta_s.relerr())**2)
    l = ValErr(l_val, l_err)
    if m == 4:
        print(f'm = {m:2d}, lambda = {l.strfmtf(3,0)}')
    else:
        print(f'm = {m:2d}, lambda = {l.strfmtf(3,0)}, lambda_measured =_
↪{meas_2ns_vals[m-5]):6.2f}, diff = {ValErr(meas_2ns_vals[m-5],_
↪meas_2ns_errs[m-5]).sigmadiff_fmt(1)}')

```

E\_3s = -5.12745 ± 0.00401

```

Delta s = 1.37108 ± 0.00064
m = 4, lambda = 1170.366 ± 1.302
m = 5, lambda = 621.527 ± 0.692, lambda_measured = 622.34, diff = 0.67
m = 6, lambda = 518.112 ± 0.577, lambda_measured = 517.50, diff = 0.53
m = 7, lambda = 477.124 ± 0.531, lambda_measured = 474.00, diff = 2.76
m = 8, lambda = 456.100 ± 0.508, lambda_measured = 454.40, diff = 1.52
m = 9, lambda = 443.719 ± 0.494, lambda_measured = 445.30, diff = 1.42

```

[37]: # Erwartete Linien für die Hauptserie: mp -> 3s

```

Delta_p_val = 3 - np.sqrt(Consts.E_Ry / E_3p.val)
Delta_p_err = (np.sqrt(np.abs(Consts.E_Ry)) / (2 * np.abs(E_3p.val) ** (3/2))) ↴
    ↪* E_3p.err
Delta_p = ValErr(Delta_p_val, Delta_p_err)

print_all(E_3s.strfmtf(5,0, 'E_3s'), Delta_p.strfmtf(5,0, 'Delta s'))

for m in range(4,6):
    l_val = Consts.hc / ((Consts.E_Ry / (m - Delta_p_val)**2) - E_3s.val)
    l_err = l_val * np.sqrt(E_3s.relerr()**2 + (2*Delta_p.relerr())**2)
    l = ValErr(l_val, l_err)

    print(f'm = {m:2d}, lambda = {l.strfmtf(3,0)}')

```

```

E_3s = -5.12745 ± 0.00401
Delta s = 0.88027 ± 0.00065
m = 4, lambda = 332.423 ± 0.555
m = 5, lambda = 286.603 ± 0.478

```

[38]: # Bestimmung der Rydbergenergie, E\_3p und Delta\_d für die 1. Nebenserie

```

qz = np.arange(3,13)

plt.figure(figsize=(16,10))
plt.errorbar(qz, meas_1ns_vals, meas_1ns_errs, fmt='.', label='Gemessene ↴
    ↪Wellenlängen der 1. Nebenserie')
plt.xlabel('Quantenzahl')
plt.ylabel('Wellenlaenge / nm')
plt.title('1. Nebenserie des Na-Atoms')
plt.xticks(qz)
plt.grid()

def fit_func_1ns(m, E_Ry, E_3p, D_d):
    return Consts.hc / (E_Ry / (m-D_d)**2 - E_3p)

start_params = [-13.6,-3,-0.02]

```

```

popt_1ns, pcov_1ns = curve_fit(fit_func_1ns, qz, meas_1ns_vals, sigma =_
    ↪meas_1ns_errs, p0 = start_params)

chi2_1ns = np.sum((fit_func_1ns(qz, *popt_1ns) - meas_1ns_vals)**2 /_
    ↪meas_1ns_errs**2)
dof_1ns = len(qz) - 3 #dof: degrees of freedom, Freiheitsgrad
chi2_red_1ns = chi2_1ns / dof_1ns
print("chi2 = ", chi2_1ns)
print("chi2_red = ", chi2_red_1ns)
prob_1ns = round(1 - chi2.cdf(chi2_1ns, dof_1ns), 2) * 100
print("Wahrscheinlichkeit:", prob_1ns, "%")

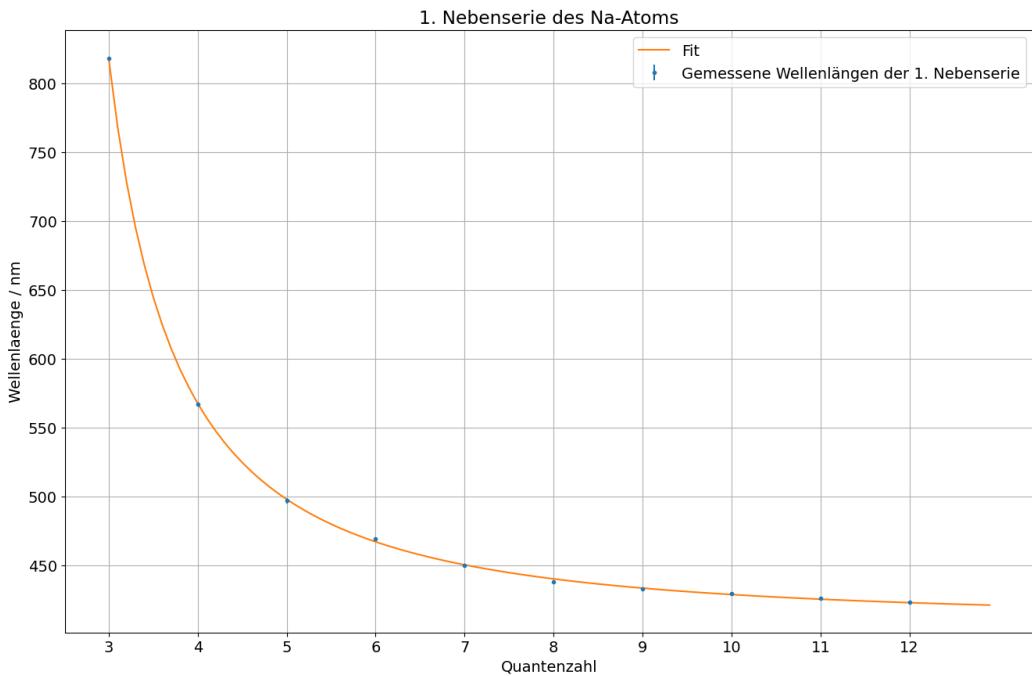
plt.plot(np.arange(3,13, 0.1), fit_func_1ns(np.arange(3,13, 0.1), *popt_1ns),_
    ↪label='Fit')
plt.legend()

plt.savefig("out/na_1ns_fit.png", format='png', bbox_inches='tight')

E_Ry_1ns = ValErr.fromFit(popt_1ns, pcov_1ns, 0)
E_3p_1ns = ValErr.fromFit(popt_1ns, pcov_1ns, 1)
D_d_1ns = ValErr.fromFit(popt_1ns, pcov_1ns, 2)

print_all(
    E_Ry_1ns.strfmrf(4, 0, 'E_Ry'),
    E_Ry_1ns.sigmadiff_fmt(ValErr(Consts.E_Ry, 0)),
    E_3p_1ns.strfmrf(4, 0, 'E_3p'),
    E_3p_1ns.sigmadiff_fmt(E_3p),
    D_d_1ns.strfmrf(4, 0, 'Δ_d'))
```

chi2 = 10.526902292499985  
chi2\_red = 1.503843184642855  
Wahrscheinlichkeit: 16.0 %  
E\_Ry = -12.9778 ± 0.4102  
1.53  
E\_3p = -3.0233 ± 0.0059  
0.75  
Δ\_d = 0.0655 ± 0.0419



```
[39]: # Bestimmung der Rydbergenergie, E_3p und Delta_d für die 2. Nebenserie
```

```
qz = np.arange(5,10)

plt.figure(figsize=(16,10))
plt.errorbar(qz, meas_2ns_vals, meas_2ns_errs, fmt='.', label='Gemessene Wellenlängen der 2. Nebenserie')
plt.xlabel('Quantenzahl')
plt.ylabel('Wellenlänge / nm')
plt.title('2. Nebenserie des Na-Atoms')
plt.xticks(qz)
plt.grid()

def fit_func_2ns(m, E_Ry, E_3p, D_s):
    return Consts.hc / (E_Ry / (m-D_s)**2 - E_3p)

start_params = [-13.6,-3,1.5]
popt_2ns, pcov_2ns = curve_fit(fit_func_2ns, qz, meas_2ns_vals, sigma = meas_2ns_errs, p0 = start_params)

chi2_2ns = np.sum((fit_func_2ns(qz, *popt_2ns) - meas_2ns_vals)**2 / meas_2ns_errs**2)
```

```

dof_2ns = len(qz) - 3 #dof:degrees of freedom, Freiheitsgrad
chi2_red_2ns = chi2_2ns / dof_2ns
print("chi2 = ", chi2_2ns)
print("chi2_red = ", chi2_red_2ns)
prob_2ns = round(1 - chi2.cdf(chi2_2ns, dof_2ns), 2) * 100
print("Wahrscheinlichkeit:", prob_2ns, "%")

plt.plot(np.arange(5,10, 0.1), fit_func_2ns(np.arange(5,10, 0.1), *popt_2ns), color='red',
         label='Fit')
plt.legend()

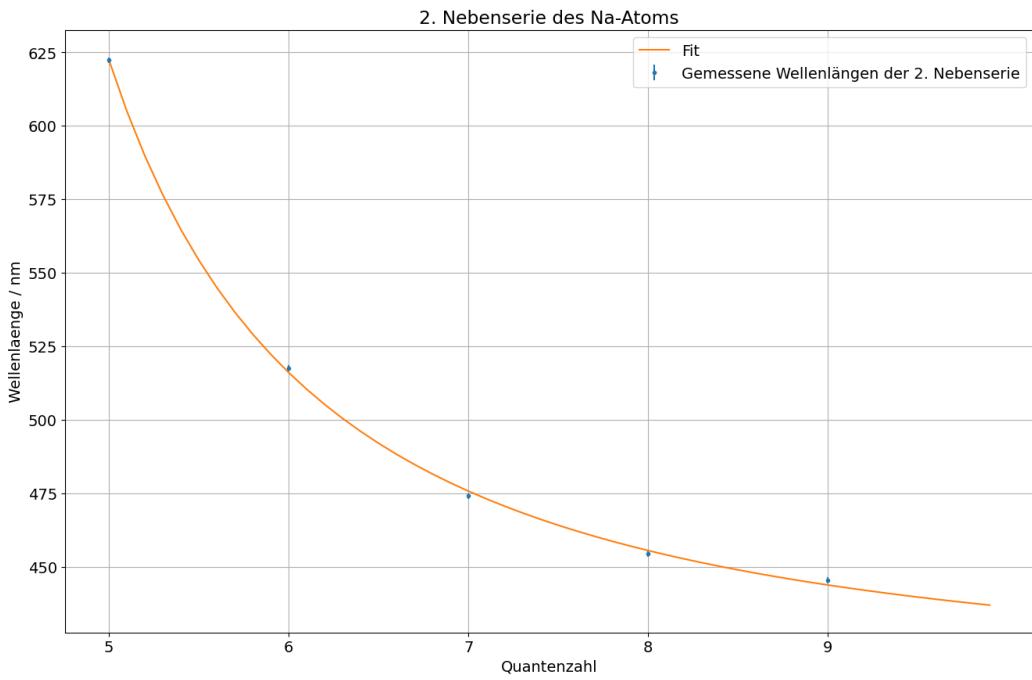
plt.savefig("out/na_2ns_fit.png", format='png', bbox_inches='tight')

E_Ry_2ns = ValErr.fromFit(popt_2ns, pcov_2ns, 0)
E_3p_2ns = ValErr.fromFit(popt_2ns, pcov_2ns, 1)
D_s_2ns = ValErr.fromFit(popt_2ns, pcov_2ns, 2)

print_all(
    E_Ry_2ns.strfmtf(4, 0, 'E_Ry'),
    E_Ry_2ns.sigmadiff_fmt(ValErr(Consts.E_Ry, 0)),
    E_3p_2ns.strfmtf(4, 0, 'E_3p'),
    E_3p_2ns.sigmadiff_fmt(E_3p),
    D_s_2ns.strfmtf(4, 0, 'Δ_s'),
    D_s_2ns.sigmadiff_fmt(Delta_s))

chi2 = 8.553390527826295
chi2_red = 4.2766952639131475
Wahrscheinlichkeit: 1.0 %
E_Ry = -11.5866 ± 2.0081
1.01
E_3p = -3.0068 ± 0.0313
0.68
Δ_s = 1.6217 ± 0.2454
1.03

```



### 0.0.3 LED Spektren

```
[40]: all_spectra = [
    ('LED Weiß', 'spektrum_led_weiss.txt', 'white'),
    ('LED Warmweiß', 'spektrum_led_warmweiss.txt', '#d6cb9a'),
    ('LED Weiß 3', 'spektrum_led_weiss3.txt', '#b8a651'),
    ('LED Rot', 'spektrum_led_rot.txt', 'red'),
    ('LED Blau', 'spektrum_led_blau.txt', 'blue'),
    ('LED Gelb', 'spektrum_led_gelb.txt', '#cccc00'),
    ('LED Orange', 'spektrum_led_orange.txt', 'orange'),
    ('Energiesparlampe', 'energiesparlampe.txt', 'lightgreen'),
    ('Glühlampe', 'gluehlampe.txt', 'orange'),
    ('Laser', 'laser.txt', '#47fb18'),
]

def draw_spectra(spectra):
    plt.figure(figsize=(16,10))
    for spec in spectra:

        spec_lam, spec_int = np.loadtxt(spec[1], skiprows=17,
                                        converters={0:comma_to_float, 1:
                                         ↪comma_to_float},
                                        comments='>', unpack=True)
```

```

plt.plot(spec_lam, spec_int, color=spec[2], label=spec[0])

wavelen_sum = 0
inten_sum = 0
for si in zip(spec_lam, spec_int):
    if si[1] > 0:
        wavelen_sum = wavelen_sum + si[0] * si[1]
        inten_sum = inten_sum + si[1]

wavelen_mean = wavelen_sum / inten_sum

plt.axvline(x=wavelen_mean, linewidth=3, linestyle='--', color=spec[2])

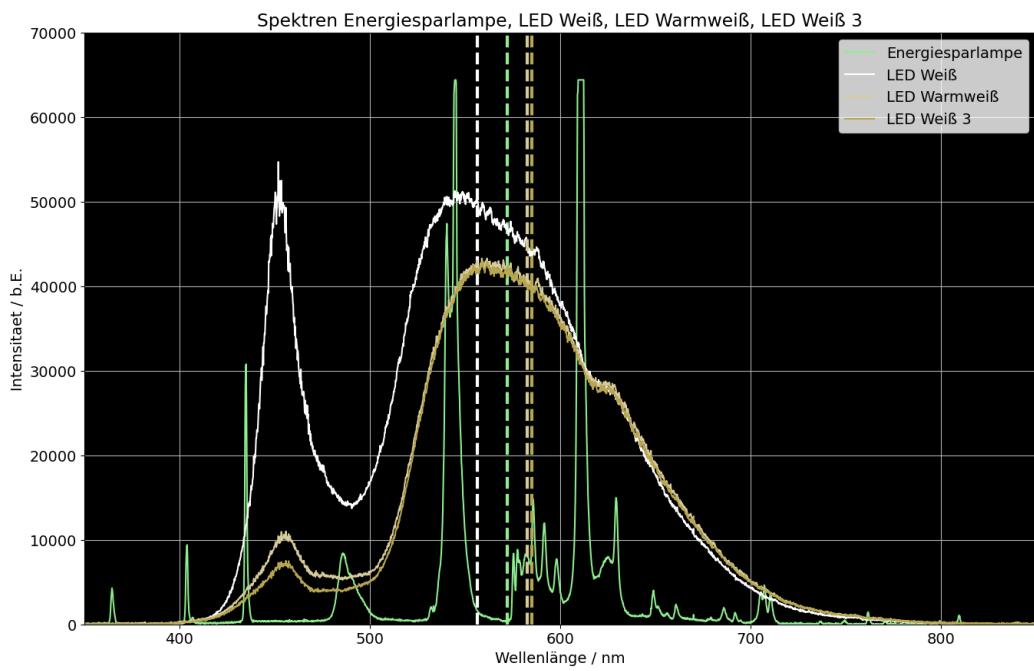
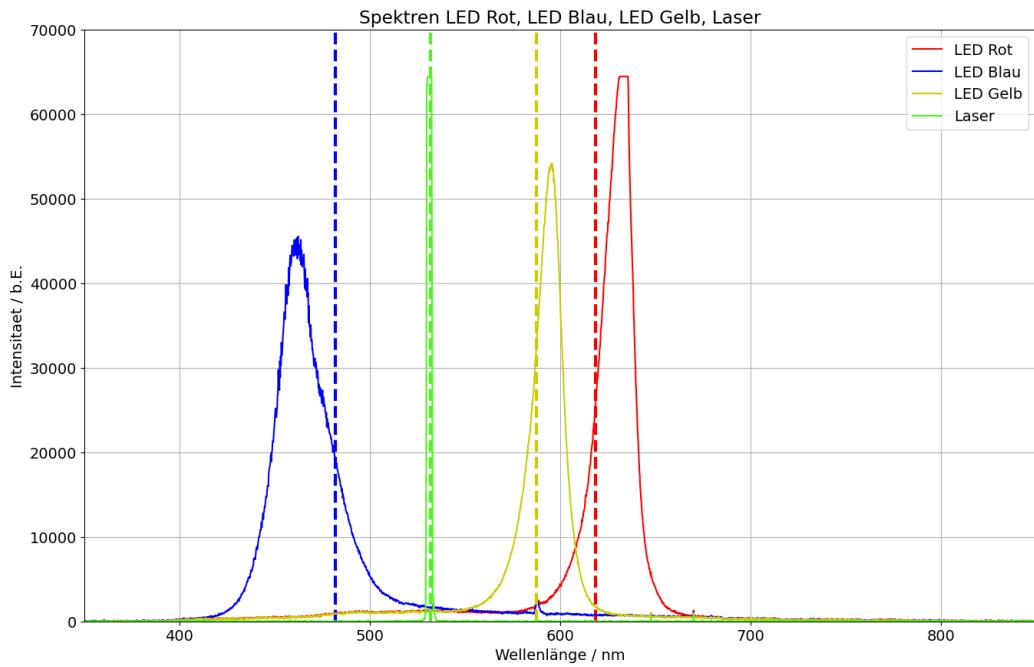
plt.title('Spektren ' + ', '.join([s[0] for s in spectra]))
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Intensität / b.E.')
plt.legend()
plt.grid()
plt.ylim((0,70000))
plt.xlim((350,850))

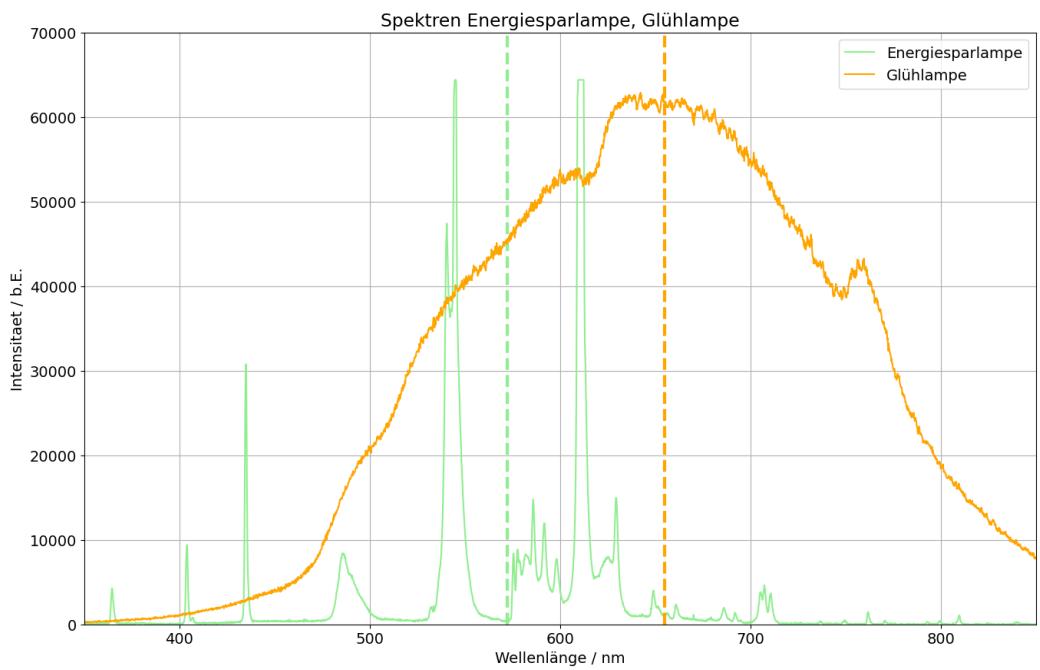
draw_spectra(all_spectra[3:6] + all_spectra[9:10])
plt.savefig('out/led_vergleich.png', format='png', bbox_inches='tight')

draw_spectra(all_spectra[7:8] + all_spectra[0:3])
ax = plt.gca()
ax.set_facecolor('black')
plt.savefig('out/led_und_energiespar.png', format='png', bbox_inches='tight')

draw_spectra(all_spectra[7:9])
plt.savefig('out/energiespar_und_glueh.png', format='png', bbox_inches='tight')

```





[ ]:

# Python Code, Bibliothek

lib

April 1, 2025

```
[51]: def floatfmt(v, prec, exp):
    return f"{v/10**exp}:0={prec}f}{LibFormatter.exp10(exp) if exp != 0 else"
        ↵' ''}""

def prec_ceil(v, prec=0):
    return np.true_divide(np.ceil(v * 10**prec), 10**prec)

def prec_floor(v, prec=0):
    return np.true_divide(np.floor(v * 10**prec), 10**prec)

[57]: class LibFormatter:
    OutputType = 'text'

    @classmethod
    def exp10(self, exp):
        if LibFormatter.OutputType == 'latex':
            return f' \cdot 10^{{{{exp}}}}'
        elif LibFormatter.OutputType == 'text':
            return f'e{exp}'
        else:
            raise ValueError(f"Unsupported OutputType: '{LibFormatter.
        ↵OutputType}'")

    @classmethod
    def pm(self):
        if LibFormatter.OutputType == 'latex':
            return f'\pm'
        elif LibFormatter.OutputType == 'text':
            return f'±'
        else:
            raise ValueError(f"Unsupported OutputType: '{LibFormatter.
        ↵OutputType}'")

[53]: import math
import numpy as np

class ValErr:
```

```

val: float = 0
err: float = 0
err_set = False

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

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

def setErr(self, err_value):
    self.err_set = True
    self.err = err_value

@classmethod
def fromMeasurements(self, measurements):
    return ValErr(np.mean(measurements), (1 / math.sqrt(len(measurements))) ↴
    ↪ * np.std(measurements, ddof=1))

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

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

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

@classmethod
def fromValPerc(self, v, perc):
    return ValErr(v, v * perc/100)

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

def strfmtf(self, prec, exp, name = ""):

```

```

prefix = ""
if name != "":
    prefix = f"{name} = "

if self.err != 0:
    return prefix + fr"{{floatfmt(self.val, prec, exp)} {LibFormatter.
→pm()} {floatfmt(self.err, prec, exp)}}"
else:
    return prefix + f"{{floatfmt(self.val, prec, exp)}"

def strfmtf2(self, prec, exp, name = ""):
    prefix = ""
    if name != "":
        prefix = f"{name} = "

    if self.err != 0:
        return prefix + fr"{{f'(' if exp != 0 else '')}{self.val/10**exp}:
→0=1.{prec}f} {LibFormatter.pm()} {self.err/10**exp}:0=1.
→{prec}f}{f'{LibFormatter.exp10(exp)}' if exp != 0 else ''}"
    else:
        return prefix + f"{{floatfmt(self.val, prec, exp)}"

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

def relerr(self):
    return self.err / self.val

def sigmadiff(self, other):
    return np.abs(self.val - other.val) / np.sqrt(self.err**2 + other.
→err**2)

def sigmadiff_fmt(self, other, prec=2):
    return f"{{prec_ceil(self.sigmadiff(other), prec)}}"

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

def __radd__(self, other):
    return self.__add__(other)

def __add__(self, other):
    if isinstance(other, self.__class__):

```

```

        return ValErr(self.val + other.val, math.sqrt(self.err**2 + other.
→err**2))
    elif isinstance(other, float) or isinstance(other, int):
        return ValErr(self.val + other, self.err)
    else:
        raise TypeError(f"unsupported operand type(s) for +: '{self.
↪__class__}' and '{type(other)}'")

def __rsub__(self, other):
    if isinstance(other, self.__class__):
        return ValErr(other.val - self.val, math.sqrt(other.err**2 + self.
→err**2))
    elif isinstance(other, float) or isinstance(other, int):
        return ValErr(other - self.val, self.err)
    else:
        raise TypeError(f"unsupported operand type(s) for +: '{self.
↪__class__}' and '{type(other)}'")

def __sub__(self, other):
    if isinstance(other, self.__class__):
        return ValErr(self.val - other.val, math.sqrt(self.err**2 + other.
→err**2))
    elif isinstance(other, float) or isinstance(other, int):
        return ValErr(self.val - other, self.err)
    else:
        raise TypeError(f"unsupported operand type(s) for +: '{self.
↪__class__}' and '{type(other)}'")

def __rmul__(self, other):
    return self.__mul__(other)

def __mul__(self, other):
    if isinstance(other, self.__class__):
        return ValErr(self.val * other.val, math.sqrt((other.val * self.
→err)**2 + (self.val * other.err)**2))
    elif isinstance(other, float) or isinstance(other, int):
        return ValErr(self.val * other, self.err * np.abs(other))
    else:
        raise TypeError(f"unsupported operand type(s) for +: '{self.
↪__class__}' and '{type(other)}'")

def __rtruediv__(self, other):
    if isinstance(other, self.__class__):
        return ValErr(other.val / self.val, math.sqrt((other.err / self.
→val)**2 + (other.val * self.err / self.val**2)**2))
    elif isinstance(other, float) or isinstance(other, int):

```

```

        return ValErr(other / self.val, np.abs(other / self.val)**2) * self.
err)
    else:
        raise TypeError(f"unsupported operand type(s) for +: '{self.
__class__}' and '{type(other)}'")

def __truediv__(self, other):
    if isinstance(other, self.__class__):
        return ValErr(self.val / other.val, math.sqrt((self.err / other.
val)**2 + (self.val * other.err / other.val**2)**2))
    elif isinstance(other, float) or isinstance(other, int):
        return ValErr(self.val / other, self.err / other)
    else:
        raise TypeError(f"unsupported operand type(s) for +: '{self.
__class__}' and '{type(other)}'")

```

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

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

```
[56]: def print_all(*args):
    for e in args:
        print(e)
```

```
[ ]:
```