
Versuch 234

15. Dezember 2021

Lichtquellen

Physikalisches Anfängerpraktikum 2.1

Jan A. Kesting

Betreuer/in: Johannes Becker

Inhaltsverzeichnis

1 Einleitung	2
1.1 Ziel/Motivation	2
1.2 Aufgaben	2
1.3 Versuchsaufbau	2
1.4 (physikalische) Grundlagen	2
3 Auswertung	10
3.1 Das Sonnenspektrum und absorption von Fensterglas	10
3.2 Auswertung der verschiedenen Lichtquellen	12
3.3 Analyse des Natriumspektrums	13
3.4 Bestimmung der Serienenergien	19
3.5 Vergleich der bestimmten Werte	21
4 Zusammenfassung und Diskussion	22

1 Einleitung

1.1 Ziel/Motivation

Man fragt sich nie, wie viele Lampen man täglich nutzt. Auch macht man sich selten darüber Gedanken, welche Art von Lampe - und wenn ja, wie - einem momentan Licht spendet. Doch über diese unheimlich alltägliche Gegenstände beschäftigt man sich jetzt etwas genauer. 

1.2 Aufgaben

1. Aufnahme des Sonnenspektrums; einmal mit Glas und einmal ohne Glas im Strahlengang und somit auch bestimmen der Absorption von Fensterglas.
2. Aufnahme und Vergleich der Lichtspektren verschiedener Lichtquellen.
3. Aufnahme des Spektrums einer Natriumlampe und folgender Zuordnung der Maxima zu Haupt- und Nebenserie.
4. Bestimmung der vom Drehimpuls abhängigen Korrekturen der Energieniveaus des Natriumatoms.

1.3 Versuchsaufbau

Der Versuchsaufbau ist im Messprotokoll als Abb. 1 zu erkennen.

1.4 (physikalische) Grundlagen

Sichtbares Licht ist nichts anderes als ein sehr geringer Anteil des elektromagnetischen Wellenpektrums. D.h. unser Auge verarbeitet gewisse Energiepakete (Lichtquanten (Photonen)), welche wir dann wahrnehmen können.

Die Technologie, Licht produzieren zu können, ist jahrtausende alt. Die Menschheit hat erstmals Licht mittels Feuer freisetzen können. Heutzutage sind viele diverse Methoden bekannt Lichtquellen zu konstruieren.

Zum Einen gibt es sogenannte **Temperaturstrahler**. Jedes Material ist ein Temperaturstrahler; je heißer ein Objekt, desto kurzwelliger wird das ausgesandte Licht. Beim absoluten Nullpunkt wirkt also jegliches Objekt komplett schwarz; je heißer der Körper, desto kurzwelligeres Licht wird ausgesandt. Bei einer Temperatur von ca. 5500 wird ein Spektrum wahrnehmbarer Wellenlängen so ausgestrahlt, dass die ausgesandte Strahlung weiß erscheint.

Wird der Körper noch weiter erhitzt überwiegen in der Strahlenkomposition kürzere Wellenlängen und das Licht wird immer mehr blau, dann violett.

Die Sonne ist der wohl bekannteste Temperaturstrahler; ihr Spektrum ist ein kontinuierliches Weißlichtspektrum, allerdings mit zahlreichen Unterbrechungen. Diese sind die sogenannten **Fraunhoferlinien**. Sie entstehen dadurch, dass gewisse Wellenlängenanteile des Lichts von bestimmten Atomen und Molekülen in der Sonnen- bzw. Erdatmosphäre absorbiert werden und somit aus dem Spektrum teilweise verloren gehen. Jene Wellenlängen sind einzigartig für jedes Atom bzw. Element, wodurch durch fehlen jener Wellenlängen genau die Elemente der Atmosphären o.ä. ermittelt werden können.

Besagte Absorption kann allerdings auf rückwärts ablaufen. Elektronen in den Atomen und Molekülen können auf ihren Energiegrundzustand (aus dem sie vorher durch Energiezufuhr gelangt sind) zurückfallen und dabei eine bestimmte Wellenlänge an Licht emittieren. Da es mehrere verschiedene Kombinationen an Energieverlusten in den Energieniveaus kommen kann haben Lichtquellen, die auf ein solches Prinzip beruhen - **Nichttemperaturstrahler** - ein sehr diskretes Lichtspektrum.

Solch ein Beispiel sind LEDs. Hier werden Photonen bei einem pn-Übergang emittiert, wobei die Wellenlänge abhängig von dem Halbleitermaterial der Diode ist. Typisch ist es, durch verschiedene Kombination von rotem, grünem und blauem Licht (RGB), verschiedene wahrnehmbare Farben hervorzurufen. Eine weitere Möglichkeit ist es eine Floureszenzschicht auf die LED aufzutragen, welche die Wellenlänge des Lichts wieder durch andere, eigene, Lichtspektren abändern.

Diesen Trick nutzt man auch bei Gasentladungslampen. Das klassische Beispiel sind Quecksilber- und Natriumdampflampen (auch bekannt: Argondampf). Da das Quecksilberspektrum aber im violetten und ultravioletten Bereich liegen ist hier eine Floureszenzschicht essentiell.

Ohne Floureszenzschicht kann auch die Natriumdampflampe verwendet werden. Natrium hat ein sehr eigenes Spektrum; die Hauptemissionswellenlänge ähnelt der bekannten Wasserstofflinie und ist mit dem Auge Wahrnehmbar. Das Licht der Hauptserie entsteht bei einem Übergang von $pn \rightarrow 3s$. Es gibt noch zwei weitere Nebenserien, wessen Spektrum bei einem Übergang von jeweils $nd \rightarrow 3p$ und $ns \rightarrow 3p$.

Die Energieniveaus hängen nicht nur von der Hauptquantenzahl ab, weswegen noch ein Korrekturfaktor $\Delta_{l,n}$ eingeführt werden muss:

$$E_{l,n} = E_{Ry} \frac{1}{(n - \Delta_{l,n})^2} \quad (1)$$

Wobei E_{Ry} die Rydbergenergie ($E_{Ry} = 13,605 \text{ eV}$) ist. Für die messbaren Wellenlängen λ gilt dann mit $E = h\frac{c}{\lambda}$:

$$\lambda = \frac{hc}{\frac{E_{Ry}}{(n - \Delta_{l,n})^2} - E_{l,n}} \quad (2)$$

Wobei der Korrekturfaktor immer passend gewählt werden muss. (h, c entsprechen den Naturkonstanten des Planckschen Wirkungsquantum und der Lichtgeschwindigkeit).

Messprotokoll

236

15.12.2021

Victoria Mohr

Jan A. Westring

- Geräteliste:
- Gitterspektrometer, Ocean Optics USB4000
 - Verschiedene Lichtquellen: LED, Laser, Energiesparlampe, Halogenlampe, Glühlampe, Natriumdampflampe
 - PC

Versuchsaufbau:

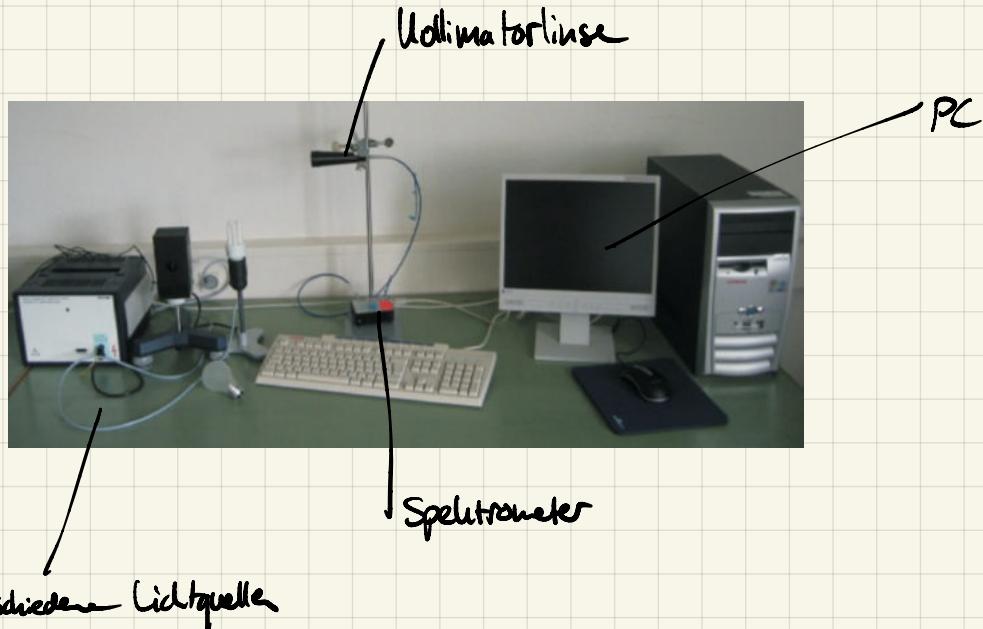


Abb. 1: Versuchsaufbau

Durchführung:

Zu Beginn führt man eine Dunkelmessung durch, damit man bei den kameradaten Messungen der Untergrund sofort herausberechnet werden kann

1) Sonnenspektrum

Zur Messung des Sonnenspektrums bzw. des Himmels, wird die Ullmstorlinsen am Fenster platziert und einmal mit offenem und einmal mit geschlossenem Fenster gemessen.

2) Qualitativer Vergleich verschiedener Lichtquellen

In diesem Versuchsteil werden die Spektren der Lichtquellen mittels des Spektrometers aufgenommen.

(Diagramm 1-11)

3) Natriumspektrum

Die Natriumlampe wird an das Vorschaltgerät angeschlossen und anschließend wird der Messaufbau so justiert, sodass die gelbe Natriumlinie bei Integrationszeit $t = 5\text{ ms}$ gerade in Sättigung geht. Scans to average: 700 $\Rightarrow 3,5\text{ s} \approx \bar{t}$

3.1 Aufnahme der Linsen bei einer Intensität

In scope-Modus werden möglichst viele schwache Linien im Bereich zwischen 300nm und 560nm registriert. Dann wird die Intensität so angepasst, sodass die Linien gut erkennbar sind und anschließend bei einer Gesamtintervallzeit von 4s über alle scans gemittelt. (Diagramme 12 und 13)

3.2 Aufnahme des Spektrums der intensiven Linien

Schließlich betrachtet man das gesamte Spektrum bei reduzierter Intensität und schließen die Irisblende,

Sodass die Na-D Linie nicht mehr in Sättigung ist.

In dieser Einstellung kann man in der Umgebung
der Na-D Linie zwei Linien beobachten. (Diagramm 14
und 15)

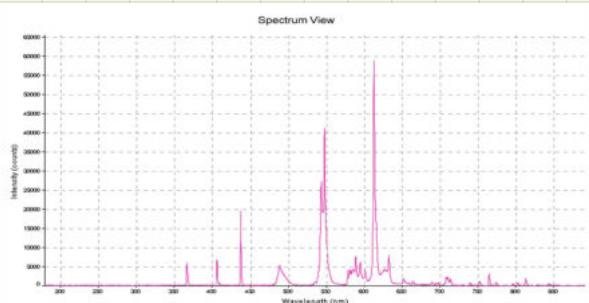


Diagramm 1 :
Spektrum der Energiesparlampe

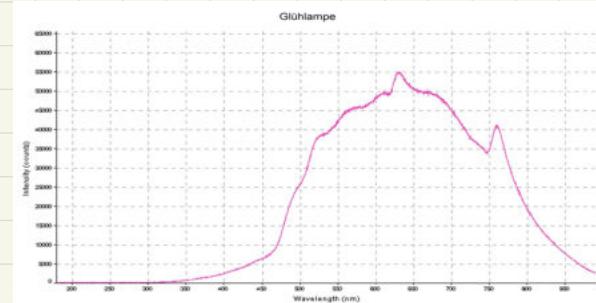


Diagramm 2 :
Spektrum der Glühlampe

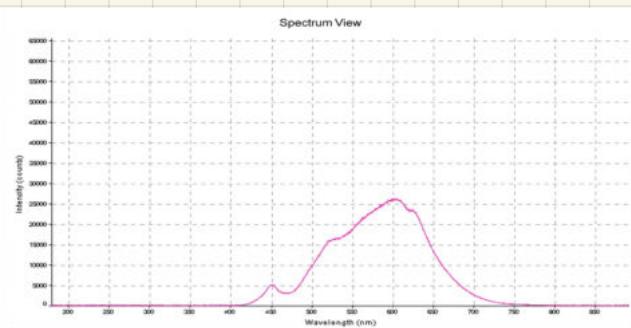


Diagramm 3 :
Spektrum der LED-Birne

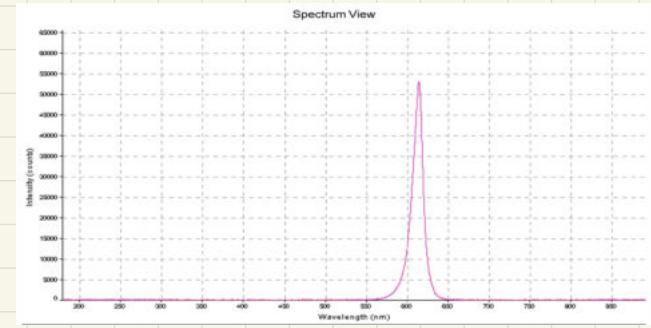


Diagramm 4 :
Spektrum der LED (Orange)

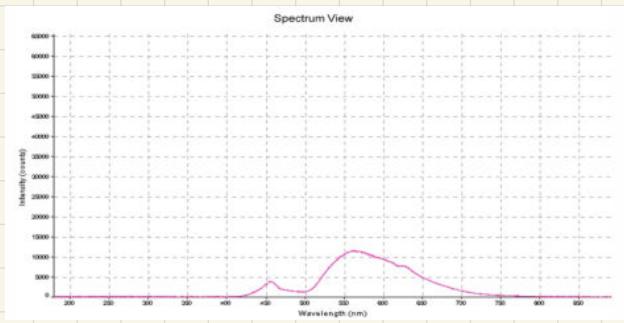


Diagramm 5 :
Spektrum der LED
(Warmweiß 1 und 2
→ identisch)

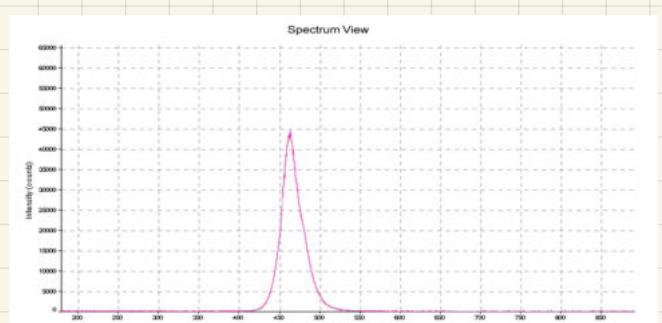


Diagramm 6 :
Spektrum der LED (Blau)

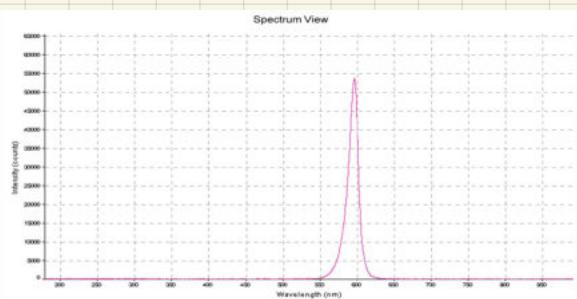


Diagramm 7 :
Spektrum der LED (Gelb)

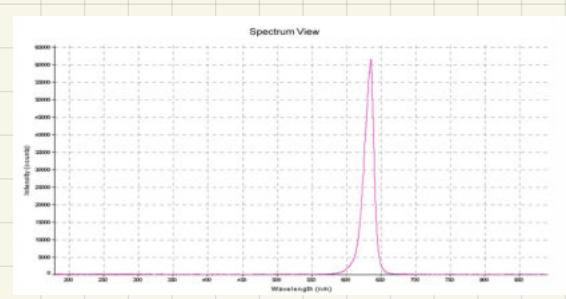


Diagramm 8 :
Spektrum der LED (Rot)

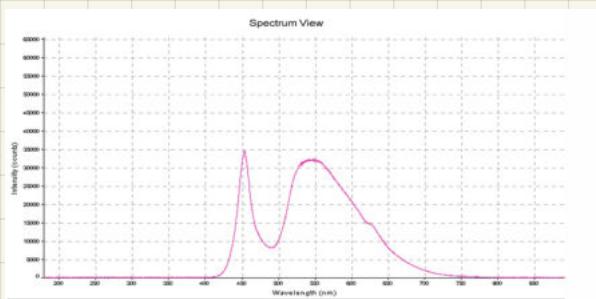


Diagramm 9 :
Spektrum der LED (Weiß)

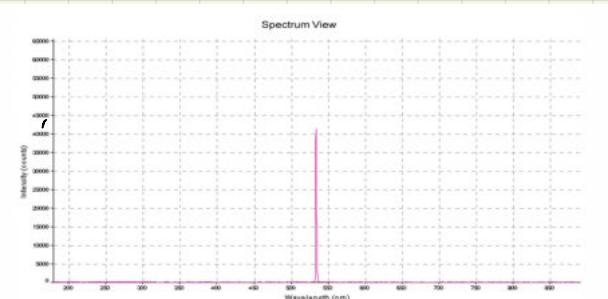


Diagramm 10 :
Spektrum des Lasers (Grün)

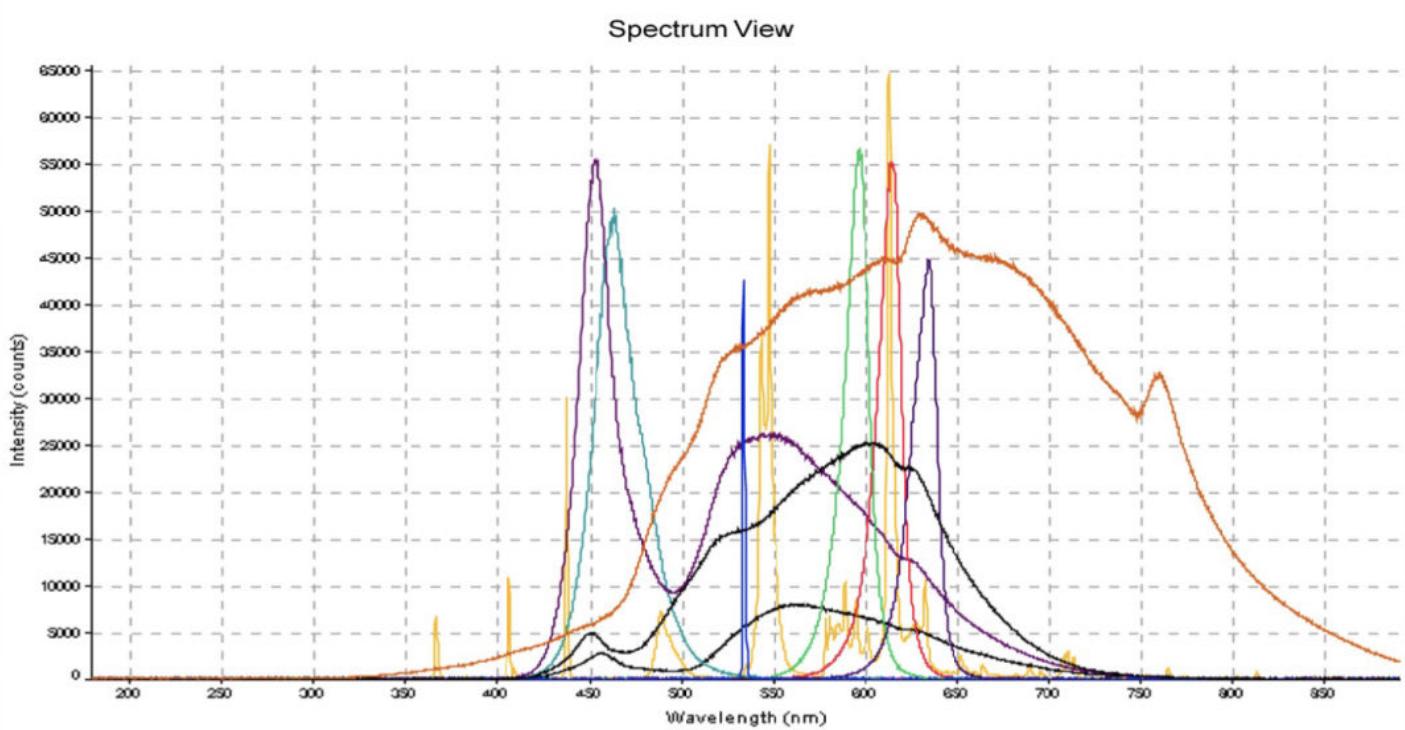


Diagramm 11 : Überlagerte Spektren der Lichtquellen

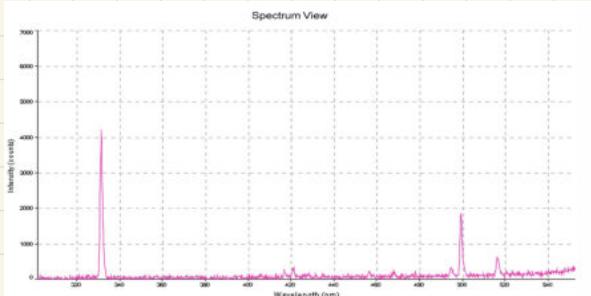


Diagramm 12:
Lage der linken Nebenlinien

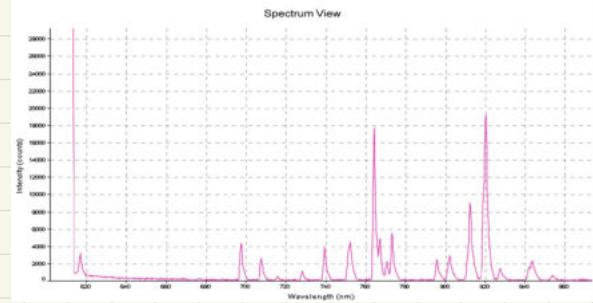


Diagramm 13:
Lage der rechten Nebenlinien

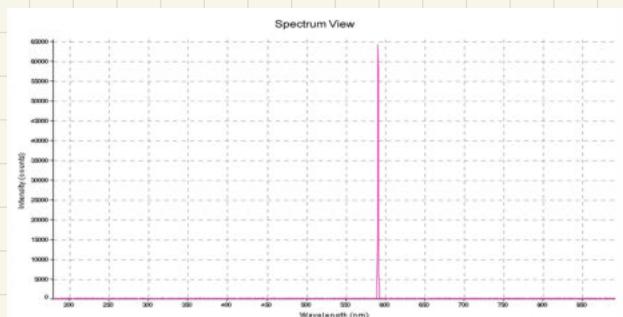


Diagramm 13:
Lage der Na-D-Linie

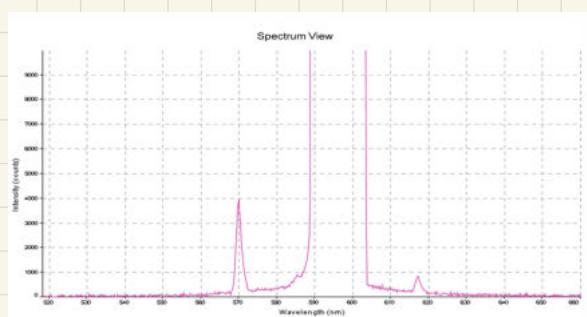


Diagramm 14:
Lage der Nebenlinien

15.12.21
J. Becker

3 Auswertung

Die Berechnungen der Auswertung finden in einem Python Skript und dementsprechendem Programm statt. Jenes ist im Anhang verzeichnet; in der Auswertung werden somit nur wesentliche Rechenwege und Ergebnisse angegeben. Außerdem werden alle Fehler nach der Gauß'schen Fehlerfortpflanzung bestimmt und in trivialen oder unwesentlichen Fällen deren Rechnung nicht angegeben; das Ergebnis nur endgültig präsentiert.

3.1 Das Sonnenspektrum und absorption von Fensterglas

Die beiden aufgenommenen Spektren des Sonnenlichts (bzw. in unserem Fall das Spektrum des bewölkten Himmels)- einmal mit Fensterglas und einmal ohne - werden zunächst übereinandergelegt (Diagramm 15) und im Anschluss wird über den folgenden Zusammenhang das Absorptionsspektrum von Fensterglas bestimmt:

$$A_G = 1 - \frac{I_{mG}(\lambda)}{I_{oG}(\lambda)} \quad (3)$$

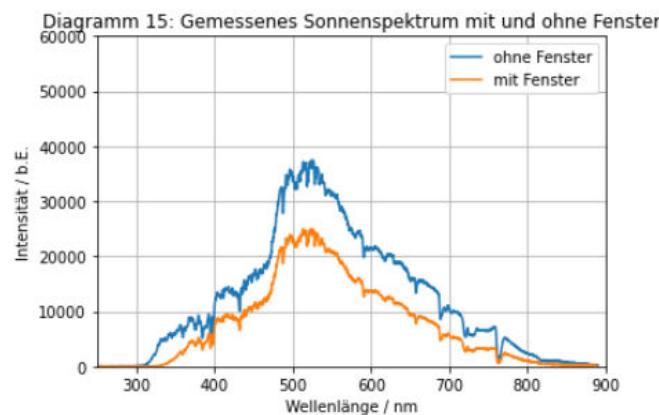


Diagramm 15: Überlagerte Himmelsspektren (einmal mit und einmal ohne Fensterglas)

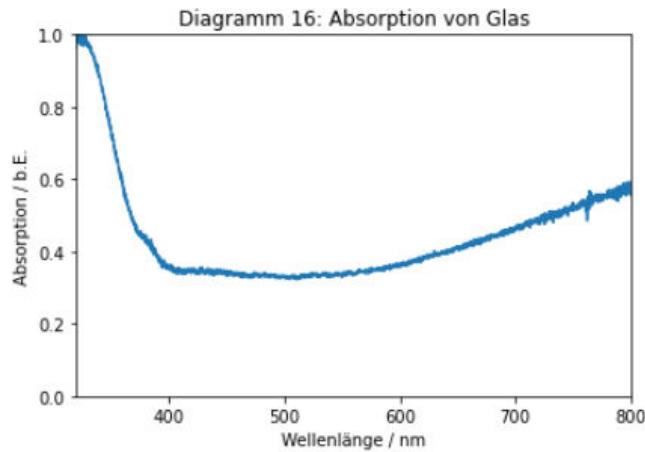


Diagramm 16: Absorptionsspektrum des Fensterglasses

Wie zu erwarten, wird nur ein geringer Anteil des sichtbaren Lichts absorbiert, geringere Wellenlängen, wie zum Beispiel UV-Strahlung, werden allerdings sehr stark absorbiert (kleinere Wellenlängen als 330 nm werden so gut wie vollständig abgeschirmt). Ab einer Wellenlänge von ca. 550 nm steigt die Absorptionsrate nur langsam an, sodass auch ein Infraroter Anteil (ca. die Hälfte) das Fensterglas durchdringen kann.

Als nächstes werden die Fraunhoferlinien des Sonnenspektrums bestimmt. Hierzu bestimmt man mittels Diagramm 17 die einzelnen Wellenlängen, bei denen die Intensität des Sonnenspektrums kurz abnehmen und vergleicht diese anschließend mit den Werten, welche im Skript geliefert werden. Die σ -Abweichungen werden folgendermaßen berechnet:

$$\sigma = \frac{|\lambda_i - \lambda_{lit_i}|}{\Delta\lambda_i} \quad (4)$$

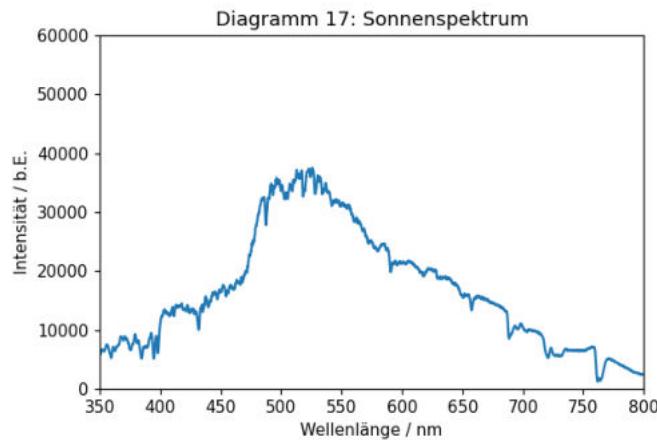


Diagramm 17: Himmelsspektrum (ohne Fensterglas) mit eingezeichneten Fraunhoferlinien

Linie	λ_{lit_i} [nm]	λ_i [nm]	σ
A	759,4	761 ± 2	0,8
B	686,7	688 ± 2	0,65
C	656,3	656 ± 2	0,15
D ₂	589,0	589 ± 2	0
E	527,0	527 ± 2	0
b ₁	518,4	517 ± 2	0,7
F	486,1	487 ± 2	0,45
G	430,8	430 ± 2	0,4
H	396,8	397 ± 2	0,1
K	393,4	393 ± 2	0,2

Tabelle 1: Die in Diagramm 17 gefundenen Fraunhoferlinien verglichen mit den gegebenen Literaturwerten

Leider lässt sich bei der Messung die D₁-Linie nicht klar von der D₂ Linie unterscheiden und die D₃-Linie ist nicht zu erkennen. Die restlichen Kennlinien sind allerdings klar zu unterscheiden.

3.2 Auswertung der verschiedenen Lichtquellen

Für diesen Versuchsteil sind die Diagramme 1-11 aus dem Messprotokoll relevant. Es fällt auf, dass die kalten Lichter (Energiesparlampe, Kaltweiße LED) einen relativ großen Anteil ihrer Intensität im kurzweligen Spektrum (Blau)

besitzen. Die Energiesparlampe (Quecksilberdampflampe) ist noch eigenartiger, da sich ihr kaltweißes Licht nicht aus einem kontinuierlichem Spektrum zusammensetzt, sondern aus einzelnen Peaks.

Die wärmeren Lichter wiederum haben ein Spektrum weiter im Langwelligen Bereich, also mehr Rot- und Gelanteile. Interessant ist vor allen Dingen, dass die LEDs ganz unterschiedlich sind. Die Warmweißen haben ein relativ kontinuierliches Spektrum mit nur leichten Grün-Blauanteilen, während die farbigen LEDs relativ breite Peaks haben (± 50 nm). Jene liegen dann auch offensichtlich im Farbbereich, welches der Farbe der LED entspricht.

Das Spektrum der Glühlampe (sowohl vom Licht als auch von der Temperatur) ist von allen Lampen am wärmsten. Dies liegt auch tatsächlich daran, dass die Glühlampe der einzige Thermische Strahler ist; das Spektrum ist auch dementsprechend sehr kontinuierlich.

Gänzlich das Gegenteil ist der Grüne Laser. Sein Licht besteht aus einem einzelnen Peak im grünwelligen Lichtbereich, ähnlich den Peaks der Quecksilberdampflampe.

Betrachtet man nun Diagramm 11, so lässt sich die mittlere Wellenlänge schwer abschätzen, dennoch erscheint der Wellenlängenmittelwert von ca. 570 nm am wahrscheinlichsten; dank fehlender Technik und Integrale ist dies aber nur eine sehr ungenaue Schätzung. Auch der Fehler der Schätzung ist nur zu erraten; lieber groß als klein, also ca. ± 50 nm.

3.3 Analyse des Natriumspektrums

Zuerst wird das Spektrum der Natriumlampe so aufgeführt, sodass die Natrium-D-Linie (Linie hoher Intensität) gut zu erkennen ist:

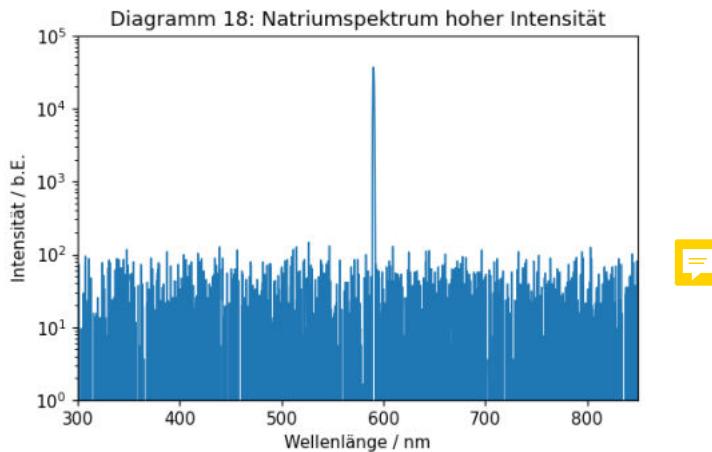


Diagramm 18: Natriumspektrum hoher Intensität; gut sichtbar ist die Natrium-D-Kennlinie (logarithmisch)

Die Hauptkennlinie liegt bei $\lambda_{NaD} = (589 \pm 1) \text{ nm}$. Allerdings sind weitere Maxima in dieser Einstellung nicht zu erkennen, da ihre Intensität deutlich geringer ist als die der Hauptmaxima, weswegen man sich nun das Natriumspektrum bei deutlich größerer Intensität I anschaut, wodurch die Maxima geringerer Intensität betont werden:

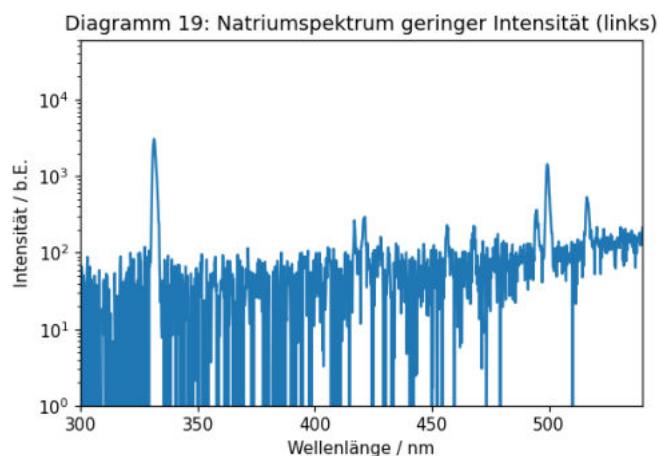


Diagramm 19: Natriumspektrum kleinerer Wellenlängen bzgl. der Na-D-Linie (logarithmisch)

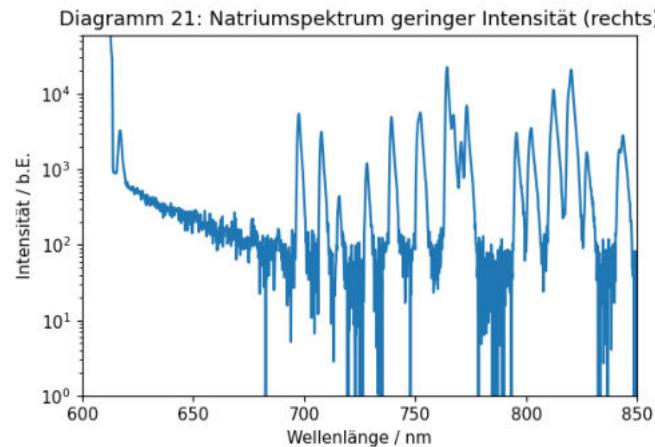


Diagramm 20: Natriumspektrum größerer Wellenlängen bzgl. der Na-D-Linie (logarithmisch)

Da man mit der logarithmischen Skala die Maxima nur schwer erkennen kann, werden die Spektren links und rechts der Natrium-D-Linie nochmal mit linearer Skala aufgeführt:

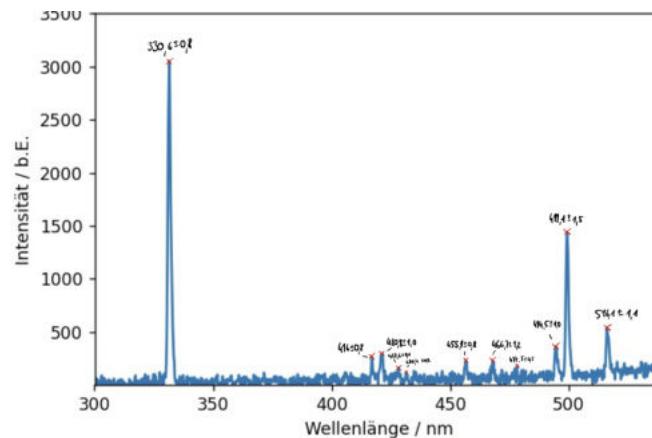


Diagramm 21: Natriumspektrum kleinerer Wellenlängen bzgl. der Na-D-Linie (linear)

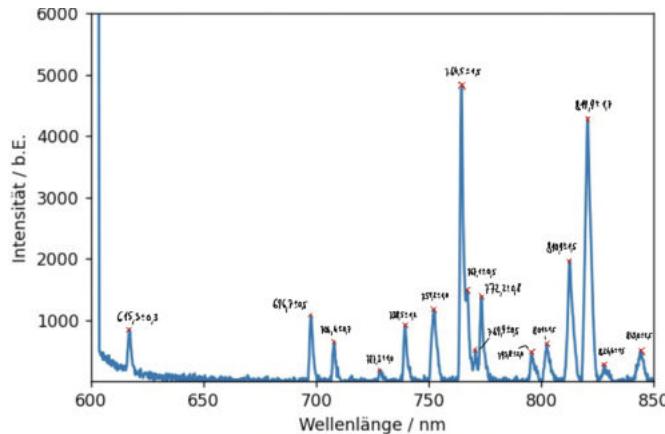


Diagramm 22: Natriumspektrum größerer Wellenlängen bzgl. der Na-D-Linie (linear)

Letzteres Diagramm ist eine Aufnahme bei wieder geringerer Intensität, da man hier die einzelnen Peaks besser bestimmen kann; an der Verteilung und Lage der Peaks hat sich nichts verändert.

3.3.1 Zuordnung der gefundenen Linien zu Serien

Als nächstes ordnet man nun der 1. Nebenserie die gefundenen Maxima zu. Man geht davon aus, dass der Korrekturterm der d-Energieniveaus gleich 0 ist. Aus (2) folgt (der Fehler folgt daraus, dass nur E_{3p} mit einem Fehler behaftet ist, wie gleich zu sehen ist):

$$\lambda_{m_b} = \frac{hc}{\frac{E_{Ry}}{m^2} - E_{3p}} \quad \Delta\lambda_{m_b} = \lambda_{m_b} \frac{\Delta E_{3p}}{E_{3p}} \quad (5)$$

Zur Berechnung der Serie fehlt allerdings noch E_{3p} . Zur Bestimmung dessen wird dem gemessenen Maxima bei 819,9 nm der Wert $m = 3$ zugeordnet und dann mithilfe obiger Gleichung bestimmt:

$$\Rightarrow E_{3p} = \frac{E_{Ry}}{m^2} - \frac{hc}{\lambda_{m_g}} \quad \Delta E_{3p} = E_{3p} \frac{\Delta\lambda_{m_g}}{\lambda_{m_g}} \quad (6)$$

$$\Rightarrow E_{3p} = \underline{\underline{(-3,024 \pm 0,006) \text{ eV}}} \quad (7)$$

Als nächstes berechnet man nun die Wellenlängen λ_{m_b} der 1. Nebenserie und vergleicht diese mit den Messwerten λ_{m_g} mittels ((8) wird auch für weitere Vergleiche genutzt):

$$\sigma = \frac{|\lambda_{m_b} - \lambda_{m_g}|}{\sqrt{\Delta\lambda_{m_b}^2 + \Delta\lambda_{m_g}^2}} \quad (8)$$

m	λ_{m_b} [nm]	λ_{m_g} [nm]	σ
3	$819,9 \pm 1,7$	$819,9 \pm 1,7$	0
4	$570,4 \pm 1,2$	-	-
5	$500,0 \pm 1,0$	$499,1 \pm 1,5$	0,5
6	$468,6 \pm 1,0$	$466,7 \pm 1,2$	1,2
7	$451,5 \pm 0,9$	-	-
8	$441,0 \pm 0,9$	-	-
9	$434,1 \pm 0,9$	-	-
10	$429,3 \pm 0,9$	$430,4 \pm 0,2$	1,2
11	$425,8 \pm 0,9$	$427,6 \pm 0,1$	1,9
12	$423,2 \pm 0,9$	$420,8 \pm 1,0$	1,8



Tabelle 2: Die berechneten und gefundenen Maxima der 1. Nebenserie und deren Vergleich

Nun widmet man sich der 2. Nebenserie des Natriumspektrums. Man weiß, dass die Natrium-D-Linie dem Übergang von $3s \rightarrow 3p$ entspricht. Mit diesem Wissen lässt sich dann folgendermaßen die Energie E_{3s} bestimmen.

$$E_{3s} = E_{3p} - \frac{hc}{\lambda_{NaD_{lit}}} \quad (9)$$

$$\Rightarrow E_{3s} = \underline{\underline{(-5,129 \pm 0,006) \text{ eV}}} \quad (10)$$

Mittels diesem gewonnenem Wert und (1) lässt sich nun der Korrekturfaktor Δ_s bestimmen:

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

$$\Rightarrow \Delta_s = \underline{\underline{1,371 \pm 0,001}} \quad (12)$$

Nun kann man die Wellenlängen der 2. Nebenserien berechnen und gegen die gefundenen Peaks eintragen:

$$\lambda_{mb} = \frac{hc}{\frac{E_{Ry}}{(m-\Delta_s)^2} - E_{3p}} \quad (13)$$

Mit dem nicht-trivialen Fehler:

$$\Delta\lambda_{mb} = \sqrt{\left(\frac{\Delta(\Delta_s)2hcE_{Ry}(m-\Delta_s)^{-3}}{(\frac{E_{Ry}}{(m-\Delta_s)^2} - E_{3p})^2}\right)^2 + \left(\frac{\Delta E_{3p}hc}{(\frac{E_{Ry}}{(m-\Delta_s)^2} - E_{3p})^2}\right)^2} \quad (14)$$

m	λ_{mb} [nm]	λ_{mg} [nm]	σ
4	1175 ± 7	-	-
5	$622,8 \pm 2,0$	$615,3 \pm 0,3$	$3,8$
6	$519 \pm 1,4$	$516,1 \pm 1,1$	$1,7$
7	$477,9 \pm 1,2$	$477,5 \pm 0,5$	$0,3$
8	$456,8 \pm 1,1$	$455,9 \pm 0,8$	$0,7$
9	$444,4 \pm 1,0$	-	-

Tabelle 3: Die berechneten und gefundenen Maxima der 2. Nebenserien und deren Vergleich

Als nächstes kann man nun das gleiche Verfahren bezüglich der Hauptserie durchlaufen. Nur wird ein unterschiedlicher Korrekturfaktor benötigt:

$$E_{3p} = \frac{E_{Ry}}{(3 - \Delta_p)^2} \quad \Rightarrow \Delta_p = 3 - \sqrt{\frac{E_{Ry}}{E_{3p}}} \quad \Delta(\Delta_p) = \frac{\Delta E_{3p}}{2} \sqrt{\frac{E_{Ry}}{E_{3p}^2}} \quad (15)$$

$$\Rightarrow \Delta_p = \underline{\underline{0,879 \pm 0,0022}} \quad \text{Feedback icon} \quad (16)$$

Nun kann man wie gewohnt die Wellenlängen der einzelnen Maxima berechnen und vergleichen:

$$\lambda_{m_b} = \frac{hc}{\frac{E_{Ry}}{(m-\Delta_p)^2} - E_{3s}} \quad (17)$$

$$\Delta\lambda_{m_b} = \sqrt{\left(\frac{\Delta(\Delta_p)2hcE_{Ry}(m-\Delta_p)^{-3}}{\left(\frac{E_{Ry}}{(m-\Delta_p)^2} - E_{3s}\right)^2}\right)^2 + \left(\frac{\Delta E_{3s}hc}{\left(\frac{E_{Ry}}{(m-\Delta_p)^2} - E_{3s}\right)^2}\right)^2} \quad (18)$$

m	λ_{m_b} [nm]	λ_{m_g} [nm]	σ
4	$332,2 \pm 0,6$	$330,6 \pm 0,8$	1,6
5	$286,5 \pm 0,4$	-	-

Tabelle 4: Die berechneten und gefundenen Maxima der Hauptserie und deren Vergleich

3.4 Bestimmung der Serienenergien

Mithilfe den gefundenen Wellenlängen lassen sich E_{Ry} , E_{3p} , Δ_d und Δ_s mithilfe einer Fitfunktion bestimmen. Mit dieser Methode werden die gemessenen Wellenlängen λ gegen die entsprechenden Quantenzahlen m aufgetragen. Dies geschieht mittels einer Funktion, deren Parameter mittels des Python-skripts so optimiert werden, sodass der Fit möglichst nah an 50% liegt. Die Fitparameter sind die variablen Größen; genau die gesuchten Werte. (Werte, die mit einer σ -Abweichung $> 3\sigma$ behaftet waren wurden in den folgenden Rechnungen nicht berücksichtigt)

Für die 1. Nebenserie gilt:

$$\lambda = \frac{hc}{\frac{E_{Ry}}{(m-\Delta_d)} E_{3p}} \quad (19)$$

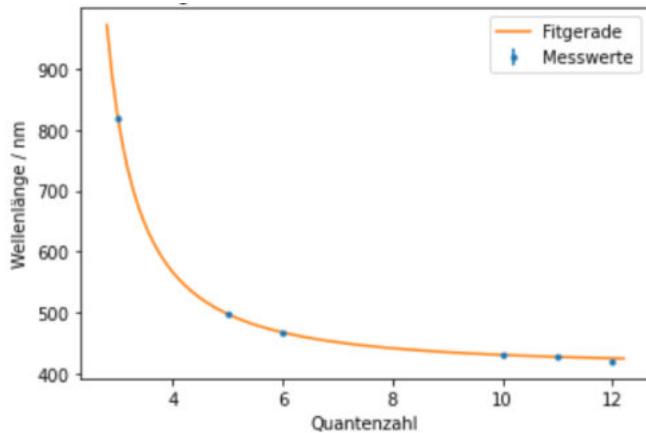


Diagramm 23: Fitfunktion der 1. Nebenserie

Die folgenden optimalen Fitparameter ergeben einen Fit von 0%, mit einer reduzierten Summe $\chi^2_{Red} = 7,8$:

$$E_{Ry_1} = \underline{(-11,8 \pm 1,2) \text{ eV}} \quad (20)$$

$$E_{3p_1} = \underline{(-3,001 \pm 0,009) \text{ eV}} \quad (21)$$

$$\Delta_d = \underline{0,18 \pm 0,14} \quad (22)$$

Das gleiche Prozedere führt man nun mit der 2. Nebenserie durch. Hier wird die Wellenlänge λ folgendermaßen bestimmt:

$$\lambda = \frac{hc}{\frac{E_{Ry}}{(m-\Delta_s)} E_{3p}} \quad (23)$$

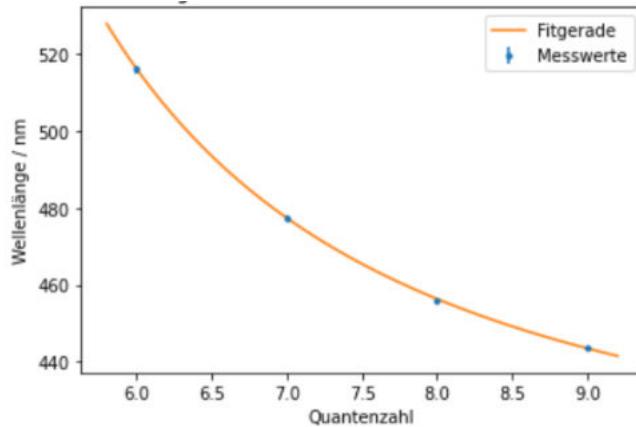


Diagramm 24: Fitfunktion der 2. Nebenserie

Die folgenden optimalen Fitparameter ergeben einen Fit von 55%, mit einer reduzierten Summe $\chi^2_{Red} = 0,36$:

$$E_{Ry_2} = \underline{\underline{(-18,0 \pm 1,8) \text{ eV}}} \quad (24)$$

$$E_{3p_2} = \underline{\underline{(-3,062 \pm 0,013) \text{ eV}}} \quad (25)$$

$$\Delta_s = \underline{\underline{0,78 \pm 0,22}} \quad (26)$$

3.5 Vergleich der bestimmten Werte

Zum Schluss der Auswertung der Daten werden die ermittelten Werte mit den entsprechenden Literaturwerten und/oder den entsprechenden, auf andere Art und Weise ermittelten, Größen verglichen. Zuerst werden die in Kapitel 3.4 ermittelten Rydbergenergien mit dem im Skript geliefertem Wert ($E_{Ry_{lit}} = -13,605 \text{ eV}$) verglichen.

$$\sigma_{E_{Ry_1}} = \frac{|E_{Ry_{lit}} - E_{Ry_1}|}{\Delta E_{Ry_1}} = \underline{\underline{1,5}} \quad (27)$$

$$\sigma_{E_{Ry_2}} = \frac{|E_{Ry_{lit}} - E_{Ry_2}|}{\Delta E_{Ry_2}} = \underline{\underline{2,4}} \quad (28)$$

Daraufhin wurden in der Auswertung 3 verschiedene Werte für die Bindungsenergie E_{3p} berechnet. Jene werden im folgenden jeweils miteinander verglichen:

$$\sigma_{E_{3p_1}} = \frac{|E_{3p_1} - E_{3p_2}|}{\sqrt{\Delta E_{3p_1}^2 + \Delta E_{3p_2}^2}} = \underline{\underline{6,7}} \quad (29)$$

$$\sigma_{E_{3p_2}} = \frac{|E_{3p_1} - E_{3p_\lambda}|}{\sqrt{\Delta E_{3p_1}^2 + \Delta E_{3p_\lambda}^2}} = \underline{\underline{2,1}} \quad (30)$$

$$\sigma_{E_{3p_3}} = \frac{|E_{3p_2} - E_{3p_\lambda}|}{\sqrt{\Delta E_{3p_2}^2 + \Delta E_{3p_\lambda}^2}} = \underline{\underline{6,0}} \quad (31)$$

Es fällt auf, dass der ermittelte Wert aus Versuchsteil 3.3.1 (E_{3p_λ}) deutlich näher an dem Wert des Kapitels 3.4 der 1. Nebenserie (E_{3p_1}) liegt. Der berechnete Fitwert der 2. Nebenserie (E_{3p_2}) liegt weiter im Abseits relativ zu den anderen Ergebnissen.

Zuletzt werden noch die zwei verschiedenen (bzw. gleichen) Korrekturfaktoren Δ_s miteinander verglichen.

$$\sigma_{\Delta_s} = \frac{|\Delta_s - \Delta_{s_\lambda}|}{\sqrt{\Delta(\Delta_s)^2 + \Delta(\Delta_{s_\lambda})^2}} = \underline{\underline{2,7}} \quad (32)$$

4 Zusammenfassung und Diskussion

Zuerst wurde das Himmelsspektrum (einmal durch ein Fensterglas und einmal ohne) aufgenommen anhand dessen zum Einen das Absorptionsspektrum für Fensterglas bestimmt wurde und zum Zweiten die Fraunhoferlinien des Spektrums bestimmt.

Das Absorptionsspektrum des Glases ist genau so, wie man es erwarten würde; die Absorptionsrate im Wellenlängenbereich des sichtbaren Spektrums ist relativ gering und weiter Richtung UV-Strahlung wird fast alles absorbiert.

Auch die Fraunhoferlinien wurden sehr gut vermessen und decken sich mit den gelieferten Werten im Skript, wobei keine σ -Abweichung > 1 gewesen ist. Allerdings lassen sich im Sonnenspektrum noch weitere, nicht-zugeordnete Linien erkennen. Hierbei handelt es sich vermutlich um weitere Fraunhoferlinien, die in der Literatur nicht weiter aufgeführt worden sind.

Als nächstes wurden dann die Lichtspektren verschiedener bekannten Lichtquellen aufgenommen und qualitativ betrachtet. Viel zu diskutieren gibt es

hier eigentlich nicht wirklich; die Spektren entsprachen den Erwartungen, wenn man denn welche hätte; kaltes Licht hat vermehrt Blauanteile, warmes Licht hingegen mehr Rotanteile. Je farbiger das Licht war, desto geringer wurde die Bandbreite und bei der Quecksilberdampflampe war das diskrete Spektrum der Floureszenzschicht gut zu erkennen.

Zuletzt erfolgte die Aufnahme des Natriumspektrums, womit anschließend Übergangsenergien, die Rydbergenergie, Korrekturfaktoren und die Wellenlängen der Haupt- und Nebenserien bestimmt und mit denen der Aufnahme, verglichen worden sind.

Die Abweichungen der einzelnen bestimmten Peaks der Serien waren alle, bis auf eines, insignifikant. Vermutlich liegt die Abweichung darin, dass beim Verstellen der Intensitäten leichte Ungenauigkeiten ins Spiel gekommen sind; andere Erklärungen könnte sonst nur noch eine defekte Apparatur gewesen sein, was sich aber auch bei anderen Ergebnissen bemerkbar gemacht hätte. Leider ließen sich nicht alle gefundenen Wellenlängen der Serien im aufgenommenen Spektrum (besonders im geringeren Wellenlängenbereich) nicht auffinden. Dies liegt sehr vermutlich daran, dass das Rauschen sich nicht gut herausfiltern ließ, wodurch die gesuchten Peaks im Rauschspektrum verloren gegangen sein müssen. Andere möglichen Fehlerquellen könnten gewesen sein, dass die Lampe nicht lange genug vorgeheizt wurde (eher unwahrscheinlich, da sie min. eine Stunde gebrannt hat), oder dass das Spektrometer nicht genau auf den richtigen Bereich der Lampe gezeigt hat, wodurch Peaks geringerer Intensität nicht mehr auffielen.

Außerdem wurden noch der Korrekturfaktor für die Hauptserie Δ_p und die 2. Nebenserie Δ_s bestimmt, sowie die Übergangsenergie E_{3p} und E_{3s} ermittelt. Jene (E_{Ry} , E_{3p} und Δ_s) wurden auch anhand der im Spektrum gefundenen Peaks über eine Fitfunktion bestimmt.

Bei der Fitfunktion der 1. Nebenserie indizieren die Fitwahrscheinlichkeit von 0% und einer reduzierten Summe $\chi^2_{Red} = 7,8$ einen nicht so erfolgreichen fit bzw. keine so genauen Fitparametern. Dennoch sind die Ergebnisse der 1. Nebenserie mit den vorherig bestimmten großen gut vergleichbar, und zeigen allesamt eine geringere Abweichung von jenen Werten als die ermittelten Größen mithilfe den Messungen der 2. Nebenserie.

Interessant ist hierbei, dass die Fitergebnisse hier (Fitwahrscheinlichkeit = 55% und $\chi^2_{Red} = 0,36$) auf deutlich bessere Werte deuten, dennoch die ermittelte Rydbergenergie weiter vom - im Teil 3.3 errechneten - Wert abweicht als mittels der 1. Nebenserie. Auch der Korrekturfaktor Δ_s liegt knapp unter einer signifikanten Abweichung.

Eng wird es dann bei den mittels der 2. Nebenserie ermittelten Werten der Bindungsenergie E_{3p} . Bei beiden Vergleichen mit sowohl dem vorherigen Auf-

gabenteil, als auch beim Vergleich mit der ermittelten Energie mittels der 1. Nebenserie (zwischen jenen zwei liegt keine signifikante Abweichung vor) treten signifikante σ -Abweichungen auf ($\sigma \geq 6$).

Ob die Erstermittelten Werte Fehlerbehaftet sind - und somit auch die Messreihe der 1. Nebenserie - oder die Messreihe der 2. Nebenserie Fehlerhaft lässt sich schwer sagen; für das eine sprechen insignifikante Abweichung; für das Andere gut Fitwerte. Die zugrundeliegende Ursache ist sehr vermutlich dieselbe: das Rauschen, welches sich nicht vernünftig herausfiltern ließ und die womöglich falsch ausgerichtete Kollimatorlinse des Spektrometers.

Trotz letzterer Ungereimtheiten, kann man dank sonstig guter Ergebnissen von einem relativ gelungenem Experiment mit unschönen Kleinigkeiten sprechen.



PAP2.1 Versuch 234 Python

January 10, 2022

1 Versuch 234 Lichtquellen

Jan A. Kesting -

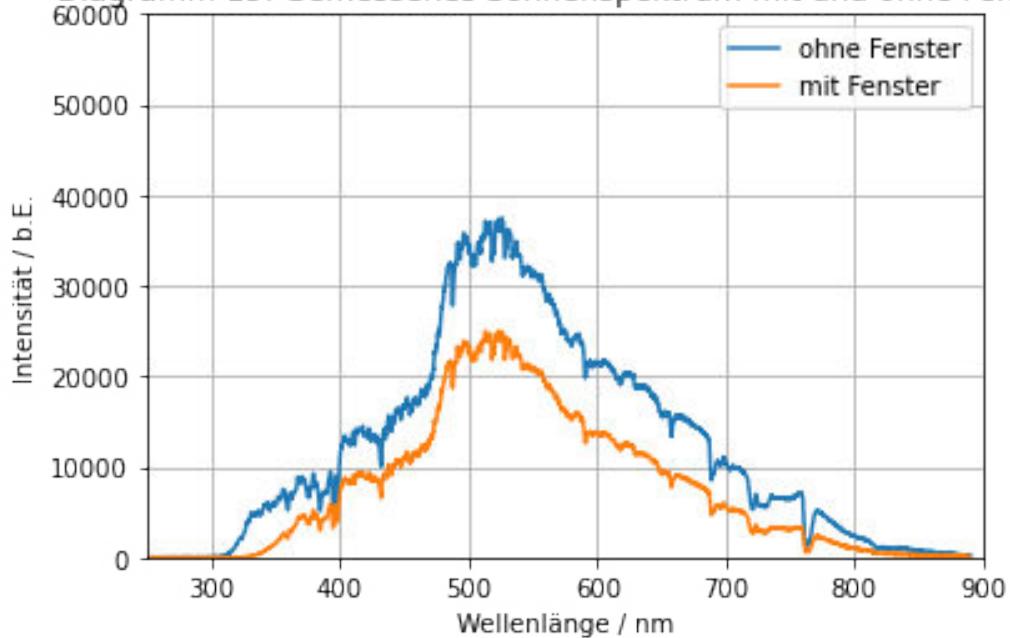
```
[2]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
def comma_to_float(valstr):
    return float(valstr.decode("utf-8").replace(',', '.'))
```

2 Auswertung des Sonnenspektrums

```
[29]: # Importieren und Header- und Kommentarzeilen ausblenden
lamb_og, inten_og = np.loadtxt('Himmel_o_g.txt', skiprows = 17, converters = {0:
    comma_to_float, 1:comma_to_float}, comments = '>', unpack = True)
lamb_mg, inten_mg = np.loadtxt('Himmel_m_g.txt', skiprows = 17, converters = {0:
    comma_to_float, 1:comma_to_float}, comments = '>', unpack = True)

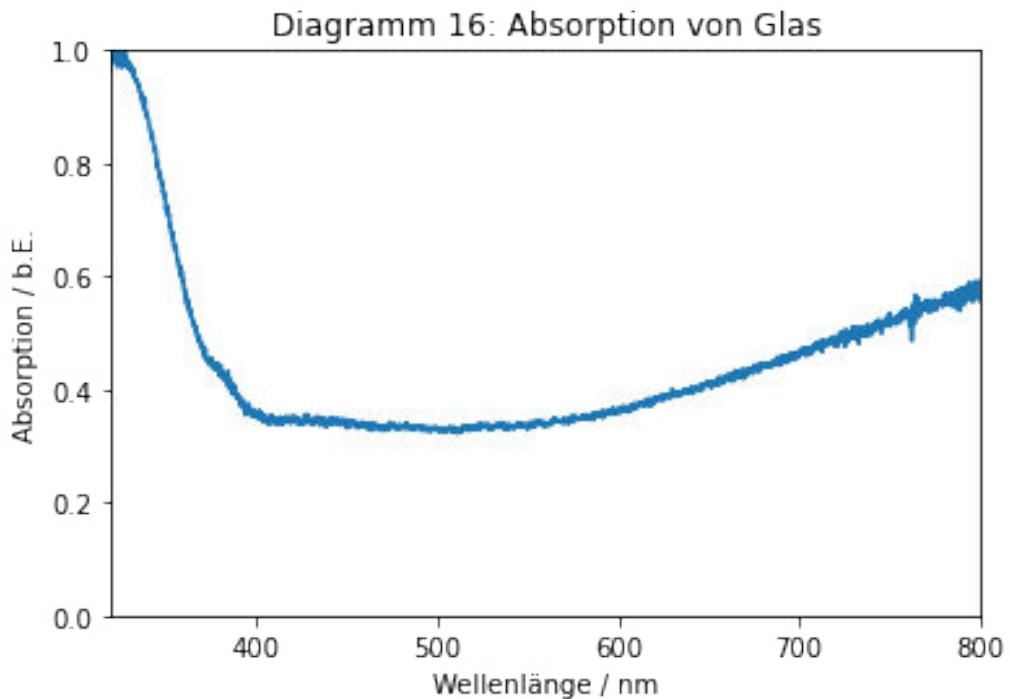
# Spektren in gemeinsames Diagramm
plt.plot(lamb_og, inten_og, label = 'ohne Fenster')
plt.plot(lamb_mg, inten_mg, label = 'mit Fenster')
plt.title('Diagramm 15: Gemessenes Sonnenspektrum mit und ohne Fenster')
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Intensität / b.E.')
plt.legend()
plt.grid()
plt.ylim((0, 60000))
plt.xlim((250, 900))
plt.savefig('Himmel_m_o_g.pdf', format = 'pdf')
```

Diagramm 15: Gemessenes Sonnenspektrum mit und ohne Fenster



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

plt.plot(lamb_mg, A)
plt.title('Diagramm 16: Absorption von Glas')
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Absorption / b.E.')
plt.ylim((0, 1))
plt.xlim((320, 800))
plt.savefig('Absorption_Glas.pdf', format = 'pdf')
```



```
[30]: # Fraunhoferlinien
%matplotlib notebook
plt.plot(lamb_og, inten_og)
plt.title('Diagramm 17: Sonnenspektrum')
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Intensität / b.E.')
plt.ylim((0, 60000))
plt.xlim((350, 800))
plt.savefig('Fraunhofer.pdf', format= 'pdf')
```

<IPython.core.display.Javascript object>

<IPython.core.display.HTML object>

```
[6]: #berechnung sigma-Abweichungen der Fraunhoferlinien

lmbda_gemessen = np.array([761,688,656,589,527,517,487,430,397,393])
lmbda_err = np.ones(10)*2
lmbda_lit = np.array([759.4,686.7,656.3,589,527,518.4,486.1,430.8,396.8,393.4])

sgma_Fl=np.abs(lmbda_gemessen-lmbda_lit)/lmbda_err

print("sigma-Abweichungen:",sgma_Fl)
```

sigma-Abweichungen: [0.8 0.65 0.15 0. 0. 0.7 0.45 0.4 0.1 0.2]

3 Auswertung des Natriumspektrums

```
[31]: # Importieren und Header- und Kommentarzeilen ausblenden
lamb_na_h, inten_na_h = np.loadtxt('Na_hohe_intensität.txt', skiprows = 17,
                                   converters = {0:comma_to_float, 1:comma_to_float}, comments = '>', unpack = True)

# Starke Linien
%matplotlib notebook
plt.plot(lamb_na_h, inten_na_h, linewidth = 1)
plt.title('Diagramm 18: Natriumspektrum hoher Intensität')
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Intensität / b.E.')
plt.yscale('log')
plt.xlim((300, 850))
plt.ylim((1, 100000))
plt.savefig('Natrium_hohe_Intensität.pdf', format = 'pdf')

<IPython.core.display.Javascript object>
<IPython.core.display.HTML object>
```

```
[32]: # Schwache Linien links (Logarithmisch)
lamb_na_g, inten_na_g = np.loadtxt('Na_geringe_intensität.txt', skiprows = 17,
                                   converters = {0:comma_to_float, 1:comma_to_float}, comments = '>', unpack = True)

%matplotlib notebook
plt.plot(lamb_na_g, inten_na_g)
plt.title('Diagramm 19: Natriumspektrum geringer Intensität (links)')
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Intensität / b.E.')
plt.yscale('log')
plt.xlim((300, 540))
plt.ylim((1, 60000))
plt.savefig('Natrium_geringe_Intensität_links.pdf', format = 'pdf')

<IPython.core.display.Javascript object>
<IPython.core.display.HTML object>
```

```
[33]: # Schwache Linien links
lamb_na_g, inten_na_g = np.loadtxt('Na_geringe_intensität.txt', skiprows = 17,
                                   converters = {0:comma_to_float, 1:comma_to_float}, comments = '>', unpack = True)

%matplotlib notebook
plt.plot(lamb_na_g, inten_na_g)
plt.title('Diagramm 20: Natriumspektrum geringer Intensität (links)')
```

```
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Intensität / b.E.')
plt.xlim((300, 540))
plt.ylim((1, 3500))
plt.savefig('Natrium_geringe_Intensität_links.pdf', format = 'pdf')
```

<IPython.core.display.Javascript object>

<IPython.core.display.HTML object>

[34]: # Schwache Linien rechts (Logarithmisch)

```
%matplotlib notebook
plt.title('Diagramm 21: Natriumspektrum geringer Intensität (rechts)')
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Intensität / b.E.')
plt.yscale('log')
plt.xlim((600, 850))
plt.ylim((1, 60000))
plt.savefig('Natrium_geringe_Intensität_rechts.pdf', format = 'pdf')
plt.plot(lamb_na_g, inten_na_g)
```

<IPython.core.display.Javascript object>

<IPython.core.display.HTML object>

[34]: [<matplotlib.lines.Line2D at 0x279deaa3490>]

[35]: # Schwache Linien rechts

```
%matplotlib notebook
plt.title('Diagramm 22: Natriumspektrum geringer Intensität (rechts)')
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Intensität / b.E.')
plt.xlim((600, 850))
plt.ylim((1, 60000))
plt.savefig('Natrium_geringe_Intensität_rechts.pdf', format = 'pdf')
plt.plot(lamb_na_g, inten_na_g)
```

<IPython.core.display.Javascript object>

<IPython.core.display.HTML object>

[35]: [<matplotlib.lines.Line2D at 0x279de409640>]

[36]: # Nebenmaxima

```
lamb_na_nm, inten_na_nm = np.loadtxt('Na_hohe_intensität_nebenmaxima.txt', skiprows = 17, converters = {0:comma_to_float, 1:comma_to_float}, comments = '>', unpack = True)
%matplotlib notebook
```

```

plt.title('Diagramm 23: Natriumspektrum geringer Intensität (rechts; geringere  

    ↪Messintensität)')
plt.xlabel('Wellenlänge / nm')
plt.ylabel('Intensität / b.E.')
plt.xlim((600, 850))
plt.ylim((1, 6000))
plt.savefig('Natrium_hohe_Intensität_nebenmaxima.pdf', format = 'pdf')
plt.plot(lamb_na_nm, inten_na_nm)

```

<IPython.core.display.Javascript object>

<IPython.core.display.HTML object>

[36]: [<matplotlib.lines.Line2D at 0x279de3a8400>]

4 Zuordnung zu Serien

```

[14]: # 1. Nebenserie
E_ry = -13.605
hc = 1.2398 * 10**3
lmbda_3m = 819.9                                     #zu finden
lmbda_3m_err = 1.7

E_3p = E_ry / 3**2 - hc / lmbda_3m
E_3p_err = (E_3p * lmbda_3m_err/lmbda_3m)

print('E_3p =', E_3p, '+/-', E_3p_err, '[eV]')

lmbda_m_b_Ns_1_list=np.array([])
lmbda_m_b_Ns_1_err_list=np.array([])

print('1. Nebenserie:')
for m in range(3, 13): #
    lmbda_m_b_Ns_1 = hc / (E_ry / m**2 - E_3p)
    lmbda_m_b_Ns_1_err = np.abs(lmbda_m_b_Ns_1 * E_3p_err / E_3p)
    print('m =', m, ', lambda =', lmbda_m_b_Ns_1, '+/-', lmbda_m_b_Ns_1_err, ↪'nm')
    lmbda_m_b_Ns_1_list = np.append(lmbda_m_b_Ns_1_list,lmbda_m_b_Ns_1)
    lmbda_m_b_Ns_1_err_list=np.append(lmbda_m_b_Ns_1_err_list,lmbda_m_b_Ns_1_err)

```

E_3p = -3.0238022929625563 +/- -0.006269623000410228 [eV]

1. Nebenserie:

```

m = 3 , lambda = 819.9000000000001 +/- 1.7000000000000002 nm
m = 4 , lambda = 570.4190578737899 +/- 1.182720329778562 nm
m = 5 , lambda = 499.9995376350145 +/- 1.036710835442767 nm
m = 6 , lambda = 468.57656569821546 +/- 0.9715577042163268 nm
m = 7 , lambda = 451.46854572623414 +/- 0.9360855320582974 nm

```

```

m = 8 , lambda = 441.0178363323295 +/- 0.9144167846871084 nm
m = 9 , lambda = 434.12806420036816 +/- 0.9001313686310841 nm
m = 10 , lambda = 429.3304529690406 +/- 0.8901838883368326 nm
m = 11 , lambda = 425.84846219200466 +/- 0.8829642465256835 nm
m = 12 , lambda = 423.23770596373885 +/- 0.8775510429788462 nm

```

[15]: #1. Nebenserie Sigma-Abweichungen

```

lmbda_m_g_Ns_1 = np.array([819.9, 0, 499.1, 466.7, 0, 0, 0, 430.4, 427.6, 420.8])  

    ↪#gefundene Werte; 0 heißt kein Wert
lmbda_m_g_Ns_1_err = np.array([1.7, 0, 1.7, 1.2, 0, 0, 0, 0.1, 0.2, 1])

sgma_Ns_1 = np.abs(lmbda_m_g_Ns_1 - lmbda_m_b_Ns_1_list)/np.sqrt(  

    ↪lmbda_m_g_Ns_1_err**2 + lmbda_m_b_Ns_1_err_list**2)
print("sigma-Abweichungen Nebenserie 1:", sgma_Ns_1)

```

```

sigma-Abweichungen Nebenserie 1: [4.72874905e-14 4.82294118e+02 4.51762704e-01  

1.21539526e+00  

4.82294118e+02 4.82294118e+02 4.82294118e+02 1.19397992e+00  

1.93469135e+00 1.83224240e+00]

```

[16]: # 2. Nebenserie

```

lmbda_3s = 589

E_3s = E_3p - hc / lmbda_3s
E_3s_err = E_3p_err
print('E_3s =', E_3s, '+/-', E_3s_err, '[eV]')

Delta_s = 3 - np.sqrt(E_ry / E_3s)
Delta_s_err = 0.5 * np.sqrt(E_ry / E_3s**3) * E_3s_err
print('Korrekturfaktor: Delta_s =', Delta_s, '+/-', Delta_s_err)

lmbda_m_b_Ns_2_list=np.array([])
lmbda_m_b_Ns_2_err_list=np.array([])
print('2. Nebenserie:')
for m in range(4, 10):
    lmbda_m_b_Ns_2 = hc / (E_ry / (m - Delta_s)**2 - E_3p)
    A = -2 * hc * E_ry * (m - Delta_s)**(-3) / ((E_ry / (m - Delta_s)**2 -  

        ↪E_3p)**2) * Delta_s_err
    B = hc / ((E_ry / (m - Delta_s)**2 - E_3p)**2) * E_3p_err
    lmbda_m_b_Ns_2_err = np.sqrt(A**2 + B**2)
    print('m =', m, ', lambda =', lmbda_m_b_Ns_2, '+/-', lmbda_m_b_Ns_2_err,  

        ↪'nm')
    lmbda_m_b_Ns_2_list = np.append(lmbda_m_b_Ns_2_list, lmbda_m_b_Ns_2)
    lmbda_m_b_Ns_2_err_list=np.  

        ↪append(lmbda_m_b_Ns_2_err_list, lmbda_m_b_Ns_2_err)

```

```

E_3s = -5.128725892283439 +/- -0.006269623000410228 [eV]
Korrekturfaktor: Delta_s = 1.371287127695769 +/- -0.000995511936076236
2. Nebenserie:
m = 4 , lambda = 1175.217063310681 +/- 7.179194190464951 nm
m = 5 , lambda = 622.833224034972 +/- 1.9697066812663886 nm
m = 6 , lambda = 519.0062381637965 +/- 1.363473540723797 nm
m = 7 , lambda = 477.87833730588085 +/- 1.155185964212201 nm
m = 8 , lambda = 456.78716055020055 +/- 1.0552742625428804 nm
m = 9 , lambda = 444.36823640006577 +/- 0.9986110652543083 nm

```

[17]: #2. Nebenserie Sigma-Abweichungen

```

lmbda_m_g_Ns_2 = np.array([0,615.3,516.1,477.5,455.9,0]) #gefundene Werte; 0
    ↪heißt kein Wert
lmbda_m_g_Ns_2_err = np.array([0,0.3,1.1,0.5,0.8,0])

sgma_Ns_2 = np.abs(lmbda_m_g_Ns_2 - lmbda_m_b_Ns_2_list)/np.sqrt(
    ↪lmbda_m_g_Ns_2_err**2 + lmbda_m_b_Ns_2_err_list**2)
print("sigma-Abweichungen Nebenserie 2:",sgma_Ns_2)

```

```

sigma-Abweichungen Nebenserie 2: [1.63697628e+02 3.78093848e+00 1.65893170e+00
3.00565594e-01
6.69940891e-01 4.44986293e+02]

```

[18]: # Hauptserie

```

Delta_p = 3 - np.sqrt(E_ry / E_3p)
Delta_p_err = 0.5 * np.sqrt(E_ry / E_3p**3) * E_3p_err
print('Korrekturfaktor: Delta_p =', Delta_p, '+/-', Delta_p_err)

lmbda_m_b_Hs_list=np.array([])
lmbda_m_b_Hs_err_list=np.array([])
print('Hauptserie:')
for m in range (4, 6):
    lmbda_m_b_Hs = hc / (E_ry / (m - Delta_p)**2 - E_3s)
    A = -2 * hc * E_ry * (m - Delta_p)**(-3) / ((E_ry / (m - Delta_p)**2 - E_3s)**2) * Delta_p_err
    B = hc / ((E_ry / (m - Delta_p)**2 - E_3s)**2) * E_3s_err
    lmbda_m_b_Hs_err = np.sqrt(A**2 + B**2)
    print('m =', m, ', lambda =', lmbda_m_b_Hs, '+/-', lmbda_m_b_Hs_err, 'nm')
    lmbda_m_b_Hs_list = np.append(lmbda_m_b_Hs_list,lmbda_m_b_Hs)
    lmbda_m_b_Hs_err_list=np.append(lmbda_m_b_Hs_err_list,lmbda_m_b_Hs_err)

```

```
Korrekturfaktor: Delta_p = 0.8788441599494563 +/- -0.002199027276549533
```

Hauptserie:

```

m = 4 , lambda = 332.1954662402193 +/- 0.5849005144079039 nm
m = 5 , lambda = 286.48181565730283 +/- 0.4188745766441933 nm

```

```
[19]: #Hauptserie Sigma-Abweichungen
```

```
lmbda_m_g_Hs = np.array([330.6,0]) #gefundene Werte; 0 heißt kein Wert
lmbda_m_g_Hs_err = np.array([0.8,0])

sgma_Hs = np.abs(lmbda_m_g_Hs - lmbda_m_b_Hs_list)/np.sqrt( lmbda_m_g_Hs_err**2
    ↪+ lmbda_m_b_Hs_err_list**2)
print("sigma-Abweichungen Hauptserie:",sgma_Hs)
```

```
sigma-Abweichungen Hauptserie: [ 1.60993321 683.93221177]
```

5 Bestimmung der Serienenergien

```
[23]: %matplotlib inline
```

```
# 1. Nebenserie (ohne Nullen)
lam1 = np.array([819.9, 499.1, 466.7, 430.4, 427.6, 420.8])
lam1_err = np.array([1.7, 1.5, 1.2, 0.2, 0.1, 1.0])
quantenzahl1 = np.array([3, 5, 6, 10, 11, 12])

# Diagramm
plt.errorbar(quantenzahl1, lam1, lam1_err, fmt = '. ', label = 'Messwerte')
plt.xlabel('Quantenzahl')
plt.ylabel('Wellenlänge / nm')
plt.title('Diagramm 24: 1. Nebenserie des Na-Atoms')

# Fit
from scipy.optimize import curve_fit
def fit_func1(m, E_Ry_1, E_3p_1, Delta_d):
    return hc/ (E_Ry_1 / (m - Delta_d)**2 - E_3p_1)

para = [E_ry, E_3p, E_3p_err]
popt1, pcov1 = curve_fit(fit_func1, quantenzahl1, lam1, sigma = lam1_err, p0 =
    ↪para)
print('E_Ry_1 =', popt1[0], ' +- ', np.sqrt(pcov1[0][0]))
print('E_3p_1 =', popt1[1], ' +- ', np.sqrt(pcov1[1][1]))
print('Delta_d =', popt1[2], ' +- ', np.sqrt(pcov1[2][2]))

x = np.linspace(2.8, 12.2, 100)
plt.plot(x, fit_func1(x, *popt1), label = 'Fitgerade')
plt.legend()
plt.savefig('Diagramm24: 1. Nebenserie.pdf', format = 'pdf')

# Chi Quadrat
chi2_1 = np.sum(((fit_func1(quantenzahl1, *popt1) - lam1) / lam1_err)**2)
dof1 = len(quantenzahl1) - 3 # Freiheitsgrade
chi2_1_red = chi2_1 / dof1
```

```

print('chi2_1 =', chi2_1)
print('chi2_1_red =', chi2_1_red)

# Wahrscheinlichkeit
from scipy.stats import chi2
prob1 = round(1 - chi2.cdf(chi2_1, dof1), 2) * 100
print('Wahrscheinlichkeit:', prob1, '%')

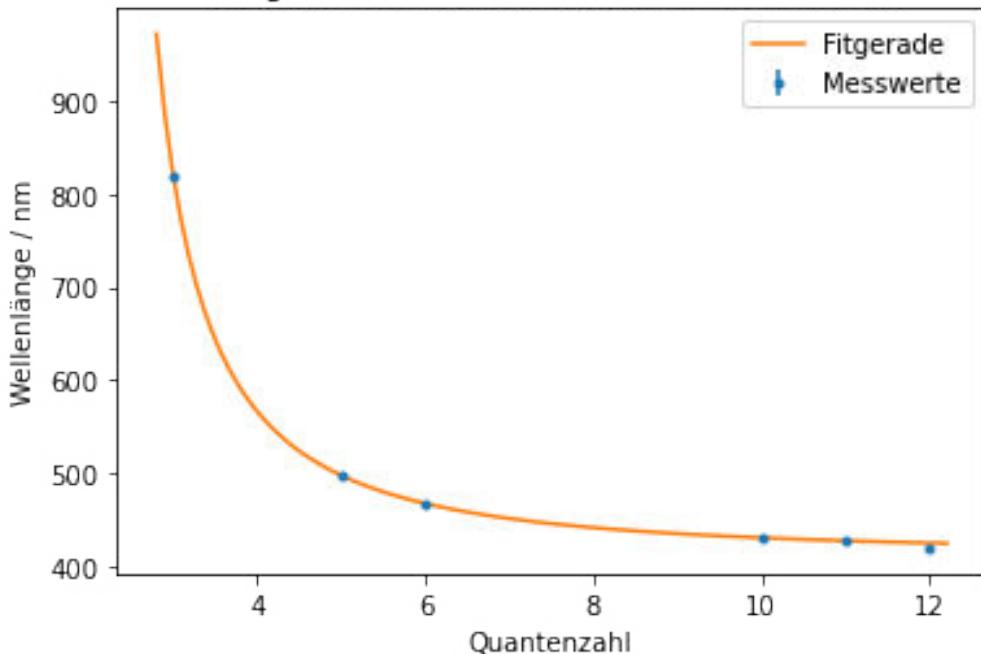
```

```

E_Ry_1 = -11.849421138920391 +- 1.2306742355683657
E_3p_1 = -3.0014360253480774 +- 0.00850802415269505
Delta_d = 0.17939192122845946 +- 0.1395644363683188
chi2_1 = 23.42253482232737
chi2_1_red = 7.807511607442457
Wahrscheinlichkeit: 0.0 %

```

Diagramm 24: 1. Nebenserie des Na-Atoms



```

[22]: # 2. Nebenserie
lam2 = np.array([516.1, 477.5, 455.9, 443.5])
lam2_err = np.array([1.1, 0.5, 0.8, 0.3])
quantenzahl2 = np.array([6, 7, 8, 9])

# Diagramm
plt.errorbar(quantenzahl2, lam2, lam2_err, fmt = 'o', label = 'Messwerte')
plt.xlabel('Quantenzahl')
plt.ylabel('Wellenlänge / nm')

```

```

plt.title('Diagramm 24: 2. Nebenserie des Na-Atoms')
plt.savefig('D24: 2. Nebenserie.pdf', format = 'pdf')

# Fit
def fit_func2(m, E_Ry_2, E_3p_2, Delta_s):
    return hc / (E_Ry_2 / (m - Delta_s)**2 - E_3p_2)

para2 = [E_ry, E_3p, E_3p_err]
popt2, pcov2 = curve_fit(fit_func2, quantenzahl2, lam2, sigma = lam2_err, p0 =
    ↪para2)
print('E_Ry_2 =', popt2[0], ' +- ', np.sqrt(pcov2[0][0]))
print('E_3p_2 =', popt2[1], ' +- ', np.sqrt(pcov2[1][1]))
print('Delta_s =', popt2[2], ' +- ', np.sqrt(pcov2[2][2]))

x = np.linspace(5.8, 9.2, 100)
plt.plot(x, fit_func2(x, *popt2), label = 'Fitgerade')
plt.legend()
plt.savefig('D24: 2. Nebenserie.pdf', format = 'pdf')

# Chi Quadrat
chi2_2 = np.sum((fit_func2(quantenzahl2, *popt2) - lam2)**2 / (lam2_err**2))
dof2 = len(quantenzahl2) - 3 # Freiheitsgrade
chi2_2_red = chi2_2 / dof2
print('chi2_2 =', chi2_2)
print('chi2_2_red =', chi2_2_red)

# Wahrscheinlichkeit
from scipy.stats import chi2
prob2 = round(1 - chi2.cdf(chi2_2, dof2), 2) * 100
print('Wahrscheinlichkeit:', prob2, '%')

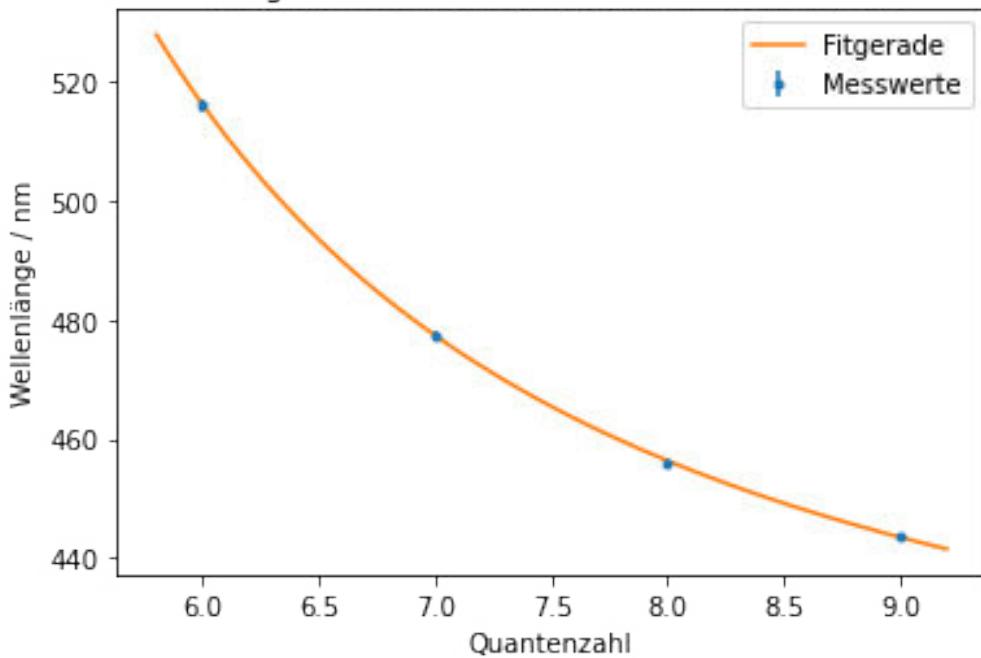
```

```

E_Ry_2 = -17.97273121414298 +- 1.7898580583180022
E_3p_2 = -3.0617626880303463 +- 0.013010702038889132
Delta_s = 0.7813568750137239 +- 0.21679819886404156
chi2_2 = 0.3583358256562406
chi2_2_red = 0.3583358256562406
Wahrscheinlichkeit: 55.000000000000001 %

```

Diagramm 24: 2. Nebenserie des Na-Atoms



[28]: #Alle sonstigen sigma-Abweichungen (obige Werte aus vorherigen Berechnungen
 ↵ manuell (leider :(!) entnehmen)

```
E_Ry_1= -11.8
E_Ry_1_err= 1.2
E_Ry_2= -18
E_Ry_2_err= 1.8
E_3p_1= -3.001
E_3p_1_err= 0.009
E_3p_2= -3.062
E_3p_2_err= 0.0013
Delta_s_1 = 0.78
Delta_s_1_err = 0.22

sigma_E_ry_1 = np.abs(E_Ry_1-E_ry)/E_Ry_1_err
print("sigma_E_ry_1:",sigma_E_ry_1)

sigma_E_ry_2 = np.abs(E_Ry_2-E_ry)/E_Ry_2_err
print("sigma_E_ry_2:",sigma_E_ry_2)

sigma_E_3p_1 = np.abs(E_3p_1-E_3p_2)/(np.sqrt(E_3p_1_err**2+E_3p_2_err**2))
print("sigma_E_3p_1:",sigma_E_3p_1)
```

```
sigma_E_3p_2 = np.abs(E_3p-E_3p_1)/(np.sqrt(E_3p_err**2+E_3p_1_err**2))
print("sigma_E_3p_2:",sigma_E_3p_2)

sigma_E_3p_3 = np.abs(E_3p-E_3p_2)/(np.sqrt(E_3p_err**2+E_3p_2_err**2))
print("sigma_E_3p_3:",sigma_E_3p_3)

sigma_Delta_s = np.abs(Delta_s-Delta_s_1)/(np.
    ↴sqrt(Delta_s_err**2+Delta_s_1_err**2))
print("sigma_Delta_s:",sigma_Delta_s)
```

```
sigma_E_ry_1: 1.5041666666666664
sigma_E_ry_2: 2.4416666666666664
sigma_E_3p_1: 6.708158863053459
sigma_E_3p_2: 2.078887347820583
sigma_E_3p_3: 5.965612553277125
sigma_Delta_s: 2.687641246167376
```

[]: