PAP 2.1
Versuch 234: "Lichtquellen"

Jonas Schweer

19. März 2025

Inhaltsverzeichnis

1 Einleitung **2**
- 1.1 Ziel des Versuchs . . . 2
- 1.2 Versuchsaufbau . . . 2
- 1.3 Physikalische Grundlagen . . . 3
  - 1.3.1 Temperaturstrahler . . . 3
  - 1.3.2 Nichttemperaturstrahler . . . 4
  - 1.3.3 Natriumspektrum . . . 4

2 Messprotokoll **5**

3 Auswertung **8**
- 3.1 Himmelsspektrum . . . 8
  - 3.1.1 Absorption der Glasscheibe . . . 8
  - 3.1.2 Absorptionslinien . . . 9
- 3.2 Natriumspektrum . . . 12
  - 3.2.1 Gemessene Spektrallinien . . . 12
  - 3.2.2 Linien der 1. Nebenserie . . . 14
  - 3.2.3 Linien der 2. Nebenserie . . . 16
  - 3.2.4 Linien der Hauptserie . . . 18
  - 3.2.5 Qualitative Untersuchung der Spektren verschiedener Strahler . 18

4 Zusammenfassung und Diskussion **19**

5 Anhang **22**

1 Einleitung

Anmerkung: Alle Bilder, welche in diesem Abschnitt verwendet wurden, sind aus dem Praktikumsskript übernommen worden.
Quelle: 'Physikalisches Praktikum PAP2.2 für Studierende der Physik B.Sc' - Jens Wagner, Februar 2025

1.1 Ziel des Versuchs

In diesem Versuch wollen wir mit Hilfe eines Gitterspektrometers die Spektren verschiedener Lichtquellen untersuchen. So wollen wir die Fraunhoferlinien des Sonnenlichts vermessen sowie die Absorption einer Fensterscheibe analysieren. Anschließend schauen wir uns qualitativ verschiedene 'technische' Lichtquellen wie LEDs an. Zu Letzt wollen wir das Spektrum einer Natriumdampflampe aufnehmen und die erkennbaren Peaks den Haupt- und Nebenübergängen innerhalb des Atoms zuordnen.

1.2 Versuchsaufbau

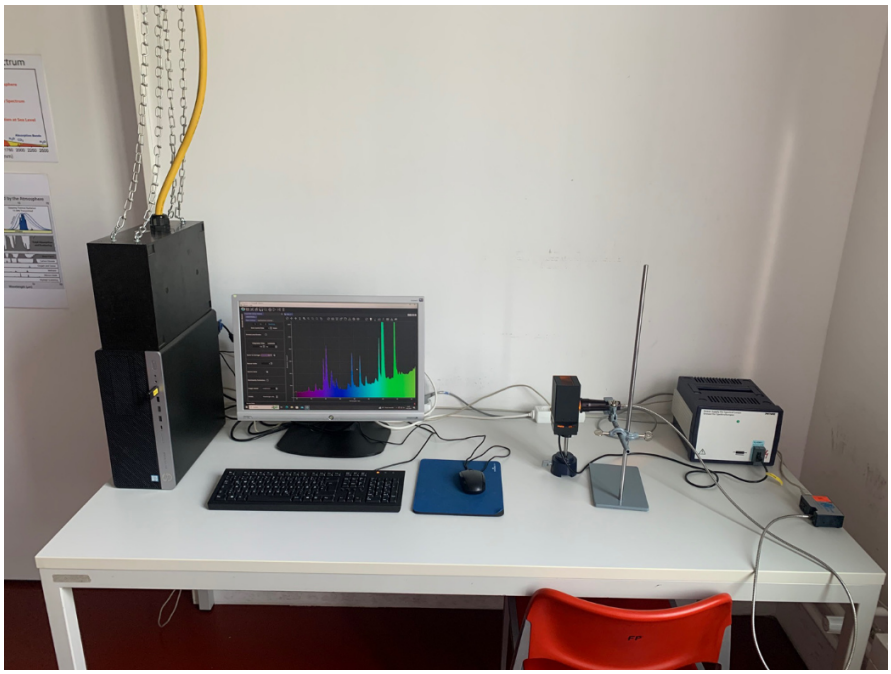


Abbildung 1: Aufbau mit Natriumlampe vor Spektrometer

Unser Aufbau ist recht simpel gehalten: Für alle Messungen nutzen wir ein Gitterspektrometer. Dieses nimmt Licht (über eine Lichtleitfaser mit Objektiv) auf und spaltet es durch ein Gitter in seine verschiedenen Wellenlängen auf. Diese können dann separat vom Computer aufgezeichnet und mit einem Programm dargestellt werden. Das Objektiv halten wir dann vor die jeweils zu untersuchenden Lichtquellen. Welche genau das sind, ist im Messprotokoll vermerkt.

1.3 Physikalische Grundlagen

1.3.1 Temperaturstrahler

Jeder Körper mit einer Temperatur über dem absoluten Nullpunkt ($T > 0\,\mathrm{K}$) strahlt Energie in Form von elektromagnetischer Strahlung ab. Die Intensität hiervon ist von der Temperatur abhängig. Ein schwarzer Strahler ist ein idealisierter Körper, welcher jegliche einfallende Strahlung absorbiert und so auch die maximale Emission aufzeigt. Das Emissionsvermögen ϵ von im ist also $\epsilon = 1$. Reelle Körper haben dann also ein $\epsilon < 1$. Die Strahlungsleistung M_λ eines schwarzen Strahlers auf Flächenelement $\mathrm{d}A$ und in Wellenlängenbereich $[\lambda, \lambda + \mathrm{d}\lambda]$ lässt sich durch das Plancksche Strahlungsgesetz berechnen:

$$M_\lambda(\lambda, T)\mathrm{d}A\mathrm{d}\lambda = \frac{2\pi hc^2}{\lambda^5} \frac{1}{\exp\left[\frac{hc}{\lambda kT}\right] - 1}\mathrm{d}A\mathrm{d}\lambda \tag{1}$$

Wir sehen also, dass die Intensitätsverteilung über alle Wellenlängen nur von der Temperatur abhängt. Plottet man die Verteilungen für einige Temperaturen, so ergibt sich folgendes Bild:

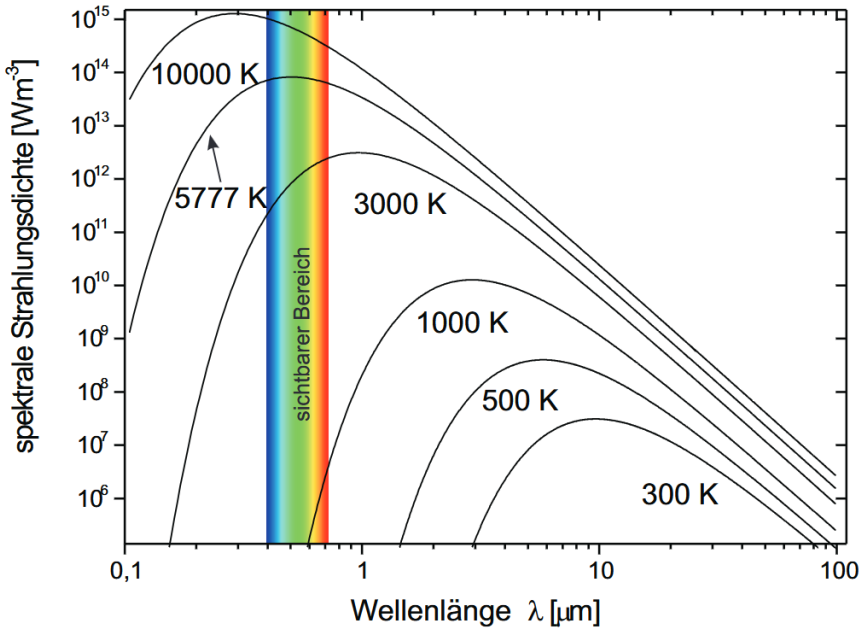


Abbildung 2: Intensitätsverteilung unterschiedlicher Temperaturen

Bei höheren Temperaturen wandert das Maximum also in den kleineren Wellenlängenbereich. Tatsächlich folgt das Intensitätsmaximum dem Wienschen Verschiebungsgesetz:

$$\lambda_{\max} = \frac{2897.8\,\mu\mathrm{m} \cdot \mathrm{K}}{T} \tag{2}$$

Ist ein Körper nun bei exakt $T = 0\,\mathrm{K}$, so ist er vollständig schwarz. Erwärmt man ihn nun, erscheint er langsam immer rötlicher. Bei weiterhin steigender Temperatur wird er orange und anschließend gelb, bis alle Wellenlängen des sichtbaren Spektrums nahezu gleichmäßig abgedeckt sind. Dann erscheint er für uns weiß. Wird er nun noch

heißer, nimmt der Rotanteil langsam ab, während der Blauanteil dominanter wird → er erscheint bläulicher. Da die Sonne circa 5777 K warm ist, erscheint direktes Sonnenlicht für uns gelb-rötlich. Das indirekte wiederum scheint bläulich, da durch Rayleigh-Streuung kurzwelliges (blaues) Licht stärker gestreut wird als das langwellige (rote) Licht. Legt das Licht also eine weite Strecke in der Atmosphäre zurück (zum Beispiel morgens und abends durch die Schräglage), kommt eher das weniger gestreute Licht an und der Himmel scheint rot.
Schaut man sich das Spektrum der Sonne an, so erkennt man einige Absorptionslinien. Diese treten auf, wenn die Elektronen der Atome in Sonnen- und Erdatmosphäre die Photonen einer bestimmten Wellenlänge aufnehmen und dann spontan in eine zufällige Richtung emittieren. Diese Wellenlängen sind diskret und charakteristisch für Elemente. So kommt es zu Tiefstellen im Spektrum, aus denen man die Zusammensetzung der Sonne (oder auch anderen astronomischen Objekten) bestimmen.

1.3.2 Nichttemperaturstrahler

Nichttemperaturstrahler strahlen wie der Name schon verrät nicht durch Wärme, sonder beruhen entweder auf Anregung bestimmter Atomzuständen in Gasen oder Festkörpern, oder Funktionieren durch Rekombination von Elektron-Loch Paaren in Halbleitern.
Bei erstem Verfahren werden Elektronen ein einem Atom auf ein höheres, energetisch ungünstigeres Niveau angehoben. Wechseln sie nun spontan wieder in den Ursprungszustand, so geben sie die Energiedifferenz als Strahlung ab. Da die Energieniveaus diskret sind, ist auch die pro Photon emittierte Energie (und mit $E_{\text{ph}} = hc/\lambda$ dann auch die Wellenlänge) diskret. Man erkennt also mit dem Spektrometer ein diskretes Spektrum. Wenn nun viel Intensität im UV-Bereich abgegeben wird, was für uns Menschen nicht sichtbar ist, so kann man den Strahler mit einer Leuchtstoffbeschichtung umhüllen, welcher unter UV-Bestrahlung im sichtbaren Bereich fluoresziert.
Das zweite Verfahren wird zum Beispiel in LEDs verwendet. Bei der Rekombination der Elektron-Loch-Paare wird ein Photon emittiert. Diese Art der Lichterzeugung ist besonders Energieeffizient, der Wirkungsgrad beträgt circa 70%.

1.3.3 Natriumspektrum

Da Alkalimetallatome nur ein Leuchtelektron besitzen, entstehen bei ihnen wasserstoffartige Spektren. Dieses Leuchtelektron, welches nach dem Bohrschen Atommodell auf der äußersten Schale sitzt, erfährt aufgrund der Elektronen der inneren Schalen näherungsweise ein von 10 Elektronen abgeschirmtes Kugelpotential $V(r)$. Für dieses gilt in Kernnähe:

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

und für große Radien:

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

Dabei ist Z die Ordnungszahl, e die Elementarladung und r der Abstand vom Kern. Es wird also bei großen Radien zum Wasserstoffpotential. Die Energieniveaus hängen nun allerdings auch von der Bahndrehimpulsquantenzahl l ab. Näherungsweise sind die

Energieterme dann:

$$E_{n,l} = E_{\mathrm{Ry}} \frac{1}{(n - \Delta_{l,n})^2} = -13.6\,\mathrm{eV} \cdot \frac{1}{(n - \Delta_{l,n})^2} \tag{5}$$

wobei E_{Ry} die Rydbergenergie ist und $\Delta_{l,n}$ der Korrekturfaktor, welcher allerdings nur sehr wenig von n abhängt, und deswegen als Δ_l genähert werden kann. Durch diese Formeln ergibt sich ein Energieübergangsschema:

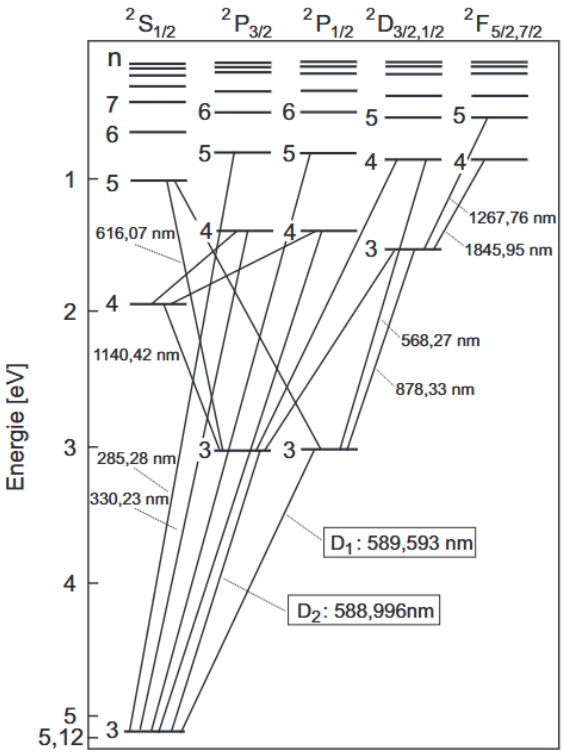


Abbildung 3: Energieübergangsschema

Der Grundzustand ist 3s, also $n = 3$ und $l = 1$. Die Hauptserie entspricht also allen Übergängen, welche im Grundzustand landen. Nach den Auswahlregeln sind das also np $\rightarrow$ 3s Übergänge. Die erste Nebenserie sind alle Übergänge der folgenden Art: nd $\rightarrow$ 3p. Analog ist die zweite Nebenserie: ns $\rightarrow$ 3p

2 Messprotokoll

Messprotokoll 234: Lichtquellen

04.03.2025
Jonas Schweer
Eleonora Bockemühl-S.

Messaufbau: - Gitterspektrometer (Ocean Optics USB4000)

- Verschiedene Lichtquellen (LED, LASER, Energiesparlampe, Glühbirne, Na-Dampflampe)
- PC mit OceanView

Skizze:

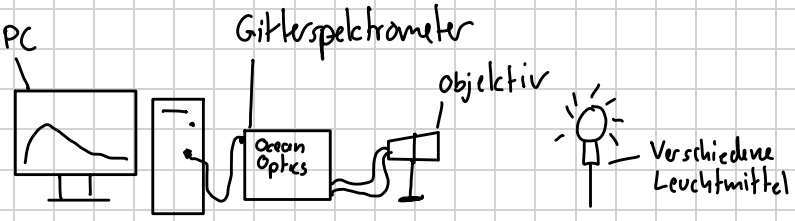


Bild 1: Messaufbau

Aufgabe 1: Aufnahme des Sonnenspektrum

Wir stellen den Messaufbau vor das Fenster und messen mit OceanView:

- das Himmelslicht eines möglichst blauen Teils des Himmels bei geschlossenem Fenster
- den selben Himmelsteil nun bei offenem Fenster
- das direkte Sonnenlicht

Aufgabe 2: Spektren verschiedener Leuchtmittel

Nun nehmen wir für folgende Lichtquellen die Spektren auf:

- Glühlampe, Halogenlampe, Energiesparlampe, verschiedene LEDs, Laser

Nachdem wir alle Daten und Bilder gespeichert haben, laden wir nochmal alle gemeinsam in das Programm und speichern auch davon ein Bild.

Aufgabe 3: Spektrallinien einer Na-Lampe

Nun nehmen wir die Natriumdampflampe in Betrieb. Mit dem Spektrometer analysieren wir wieder das Licht. Wir nehmen einmal die Messung auf, wenn der 500nm-Peak in Sättigung ist, und ein mal, wenn nicht. So wollen wir alle Spektrallinien analysieren können.

Aufgabe 4: Spektrallinien um die D-Linie

Wir reduzieren die Integrationszeit und schließen die Irisblende, sodass man die D-Linie, sowie die 2 Linien links und rechts von ihr klar getrennt erkennen kann. Diese präziseren Messungen speichern wir ebenfalls ab. Eine weitere Messung speichern wir mit D-Linie in Sättigung.

Aufgabe 5: Spektrallinien im Bereich 650-850 nm

Wir wählen den Zoombereich so, dass nur der Bereich von 650-850 nm angezeigt wird. Mit Blendenöffnung und Integrationszeit bringen wir alle Spektrallinien in Sättigung und speichern die Daten.



3 Auswertung

3.1 Himmelsspektrum

3.1.1 Absorption der Glasscheibe

Um die Absorption des Fensterglases zu bestimmen, plotten wir zunächst unsere gemessenen Spektren des Himmels mit und ohne Glas in das selbe Diagramm. Da wir im Verlaufe der Messung die Integrationszeit geändert haben, ist die Intensität mit Scheibe über der Intensität ohne Scheibe. Zur besseren Vergleichbarkeit plotten wir also außerdem beide Spektren nochmal, wobei deren Maximum auf den selben Wert normiert sind.

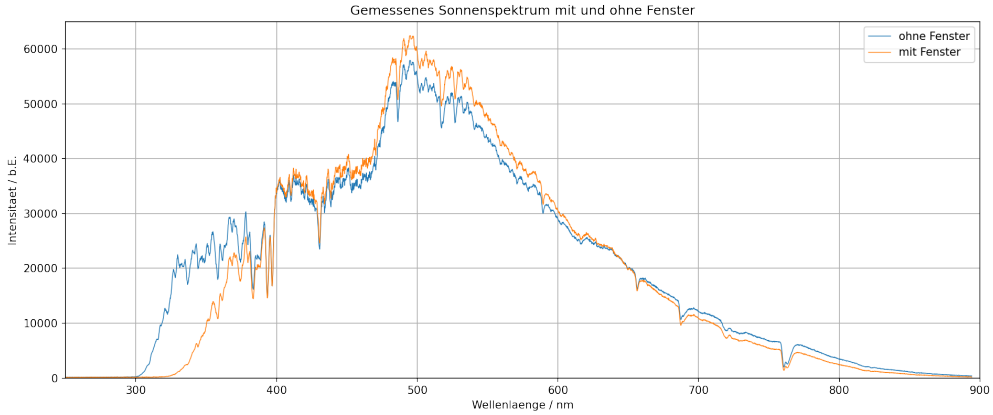


Abbildung 4: Gemessenes Sonnenspektrum mit und ohne Scheibe

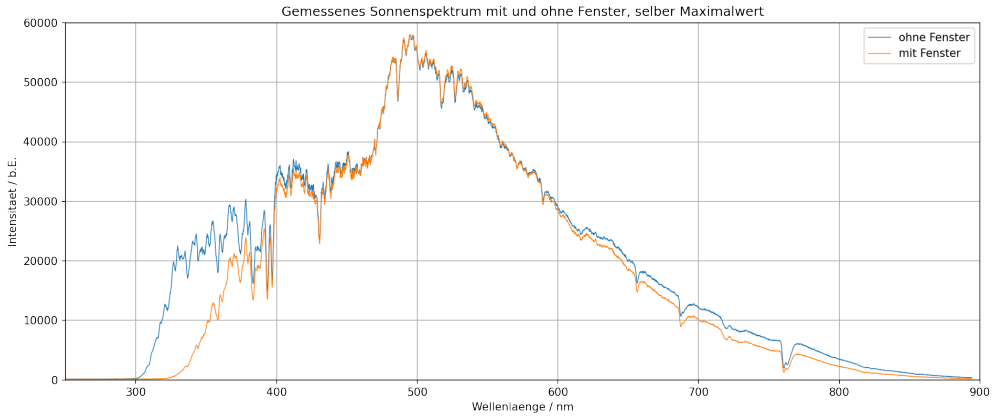


Abbildung 5: Gemessenes Sonnenspektrum mit und ohne Scheibe, normiert auf selbes Maximum

Um nun die Absorption des Glases zu bestimmen, nutzen wir folgende Formel:

$$A_{\text{Glas}} = 1 - \frac{I_{\text{mG}}(\lambda)}{I_{\text{oG}}(\lambda)} \tag{6}$$

Da bei beiden Plots der genaue Unterschied nicht bekannt ist, die unterschiedliche Skalierung der Spektren jedoch nur die y-Verschiebung und die Skalierung des Absorptionsplots ändert, ist trotzdem bei Beiden der selbe qualitative Verlauf zu erkennen:

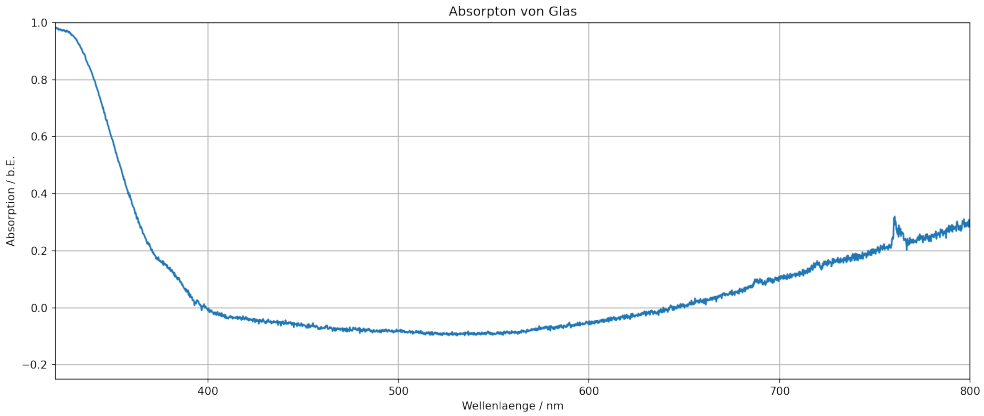


Abbildung 6: Absorptionsspektrum Glas

Man erkennt also, dass im UV-Bereich sehr viel vom Glas absorbiert wird. Diese Strahlen sind bei zu starker Aussetzung schädlich für den Menschen, weshalb es sinnvoll ist, dass Glas diese abschirmt und so UV-Strahlen innerhalb von Räumen minimiert werden. Anschließend sehen wir, dass vom sichtbaren Licht sehr wenig und gleichmäßig absorbiert wird. Aufgrund der unterschiedlichen Integrationszeiten geht unser Graph hier ins negative, allerdings können wir logisch herleiten, dass das Glas weder selbst Licht emittiert (nur so kann 'negative Absorption' entstehen), noch sehr viel des Lichtes dimmt. Hier würde sich der reale Graph also sehr nah an der null bewegen, allerdings positiv sein. Zuletzt sieht man noch, dass die Absorption bei höheren Wellenlängen wieder anfängt zuzunehmen, bis sie im Infrarotbereich nahezu linear ansteigt.

3.1.2 Absorptionslinien

Wir plotten zuerst unser Spektrum, bei welchem wir das Objektiv nahezu direkt in die Sonne gehalten haben. Anschließend zeichnen wir die theoretische Lage der wichtigsten Fraunhofer-Linien ein, welche im Praktikumsskript zu finden sind, aber auch nochmal in Tabelle 1 notiert sind.

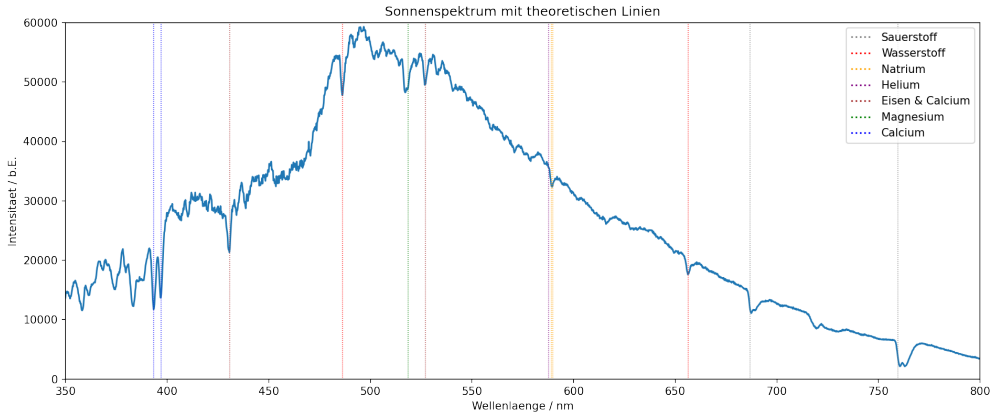


Abbildung 7: Theoretische Fraunhofer-Linien

Mit dem Wissen, welche 'Dips' vermutlich zu den Fraunhofer-Linien gehören, können wir nun manuell die Lage der Tiefpunkte festlegen. Die von Hand festgelegten Linien sehen so aus:

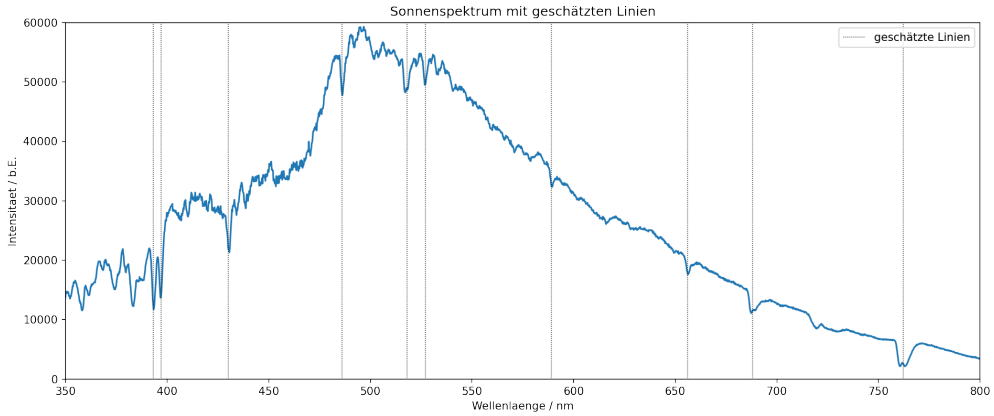


Abbildung 8: Gemessene Fraunhofer-Linien

Dabei wurden die beiden Natriumlinien und die Heliumlinie zusammengefasst, da man diese in der Messung nicht unterscheiden kann. Tabellarisch lauten die Daten:

Element	Theoretische Wellenlänge [nm]	Gemessene Wellenlänge [nm]
Sauerstoff - O_2	759.4	762 ± 1
	686.7	688 ± 1
Wasserstoff - H_2	656.3	656 ± 0.5
	486.1	486 ± 0.5
Natrium - Na	589.6	589 ± 0.5
	589.0	589 ± 0.5
Helium - He	587.6	589 ± 0.5
Eisen und Calcium - Fe & Ca	527.0	527 ± 0.5
	430.8	430 ± 0.5
Magnesium - Mg	518.4	518 ± 0.5
Calcium - Ca	396.8	397 ± 0.5
	393.4	393 ± 0.5

Tabelle 1: Gemessene und Theoretische Fraunhofer-Linien

Das selbe Vorgehen führen wir nun mit der Balmer-Serie des Wasserstoff durch, welche auch im Anhang des Skripts oder in Tabelle 2 zu finden sind.

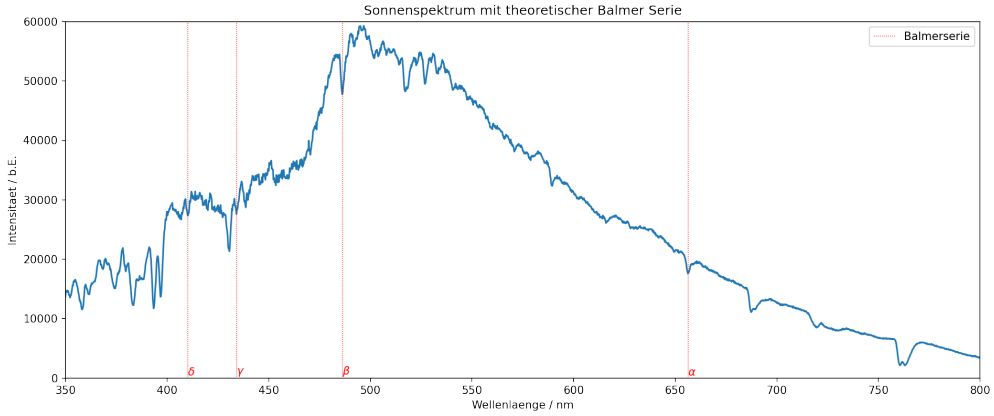


Abbildung 9: Theoretische Balmer-Serie

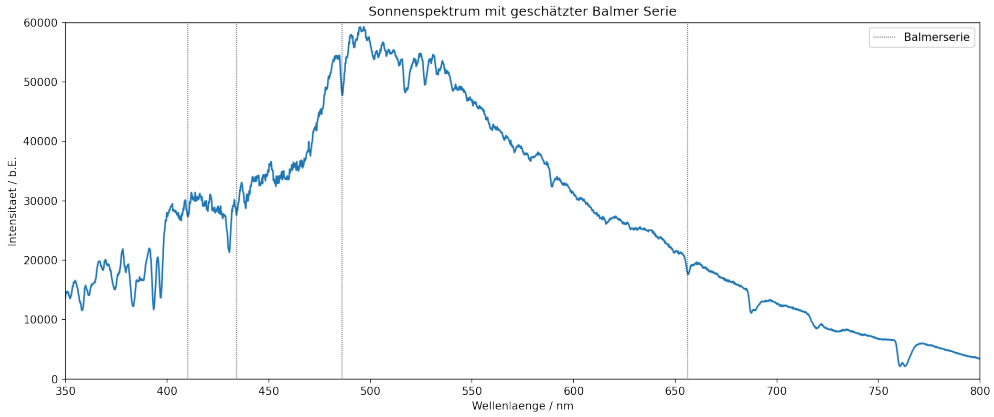


Abbildung 10: Geschätzte Balmer-Serie

Übergang	Theoretische Wellenlänge [nm]	Gemessene Wellenlänge [nm]
α	656.3	656 ± 0.5
β	486.1	486 ± 0.5
γ	434.0	434 ± 0.5
δ	410.1	410 ± 0.5

Tabelle 2: Gemessene und Theoretische Fraunhofer-Linien

3.2 Natriumspektrum

3.2.1 Gemessene Spektrallinien

Wir plotten unsere gemessenen Natriumspektren bei verschiedene Integrationszeiten, und suchen für jeden Wellenlängenbereich die aussagekräftigste Messung. Anschließend legen wir in alle Peaks, die (mehr oder weniger) erkennbar sind eine vertikale nummerierte Linie. Mit diesen Nummern identifizieren wir die Wellenlänge in Tabelle 3:

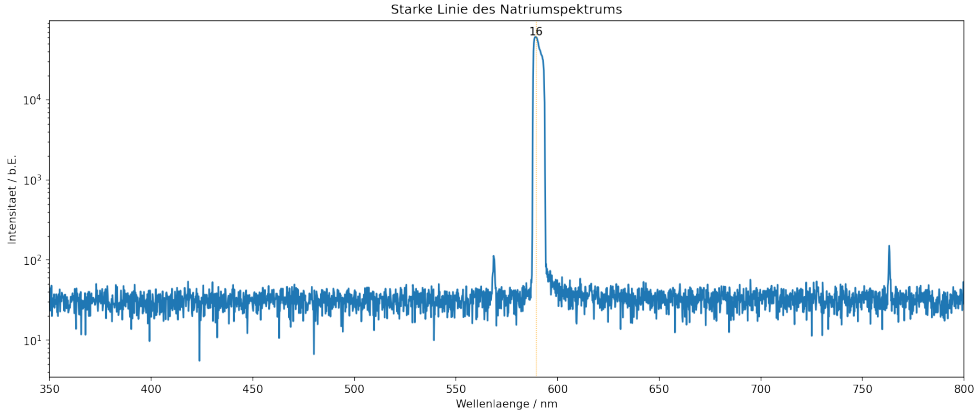


Abbildung 11: Starke Hauptlinie

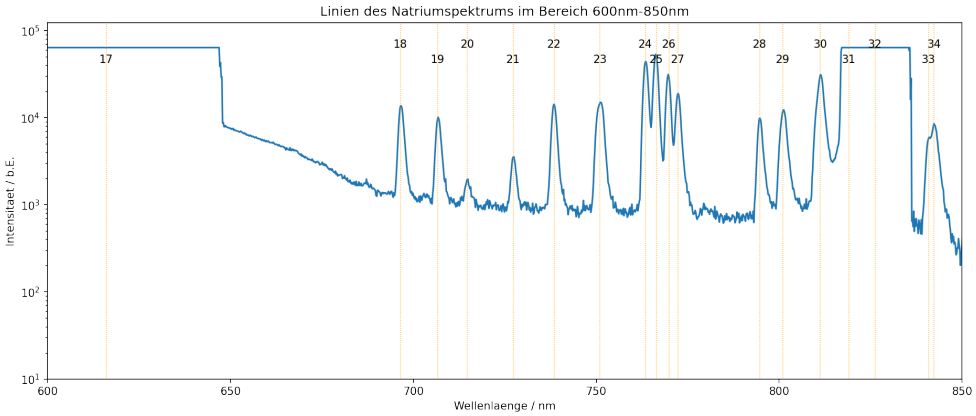


Abbildung 12: Hoher Wellenlängenbereich

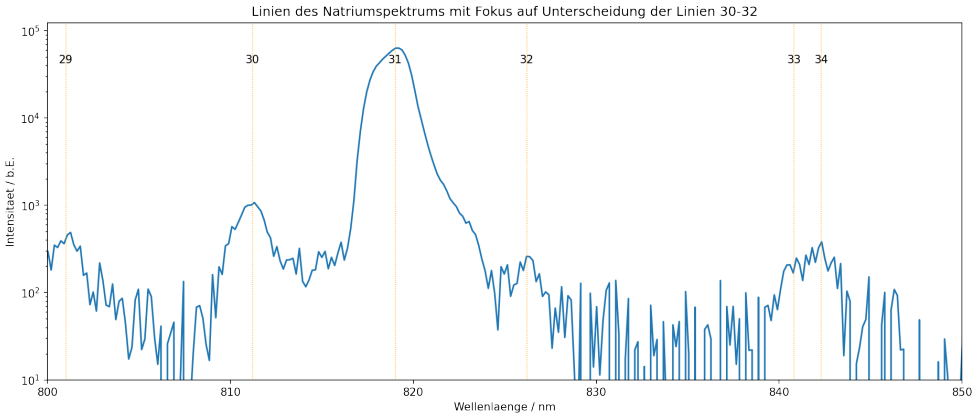


Abbildung 13: Genauere Unterteilung der 800nm-850nm Wellenlängen

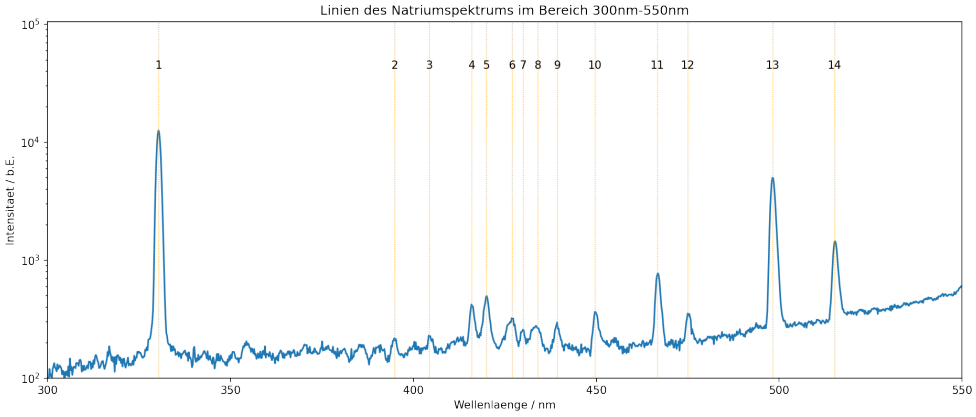


Abbildung 14: Niedrige Wellenlängen

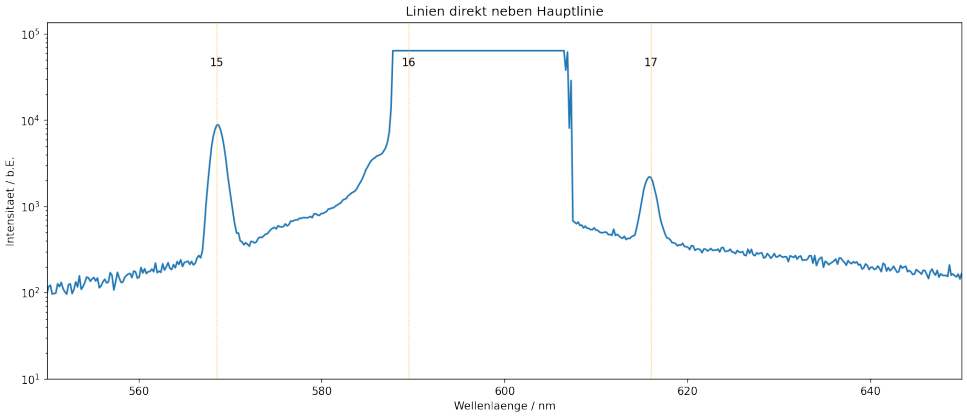


Abbildung 15: Unterteilung der D-Linien

Die Peaks lauten wie folgt:

Peak	λ [nm]	Peak	λ [nm]
1	330.4 ± 1.0	18	696.5 ± 1.0
2	394.9 ± 1.0	19	706.7 ± 1.0
3	404.3 ± 1.0	20	714.7 ± 1.0
4	416.0 ± 1.0	21	727.3 ± 1.0
5	420.0 ± 1.0	22	738.4 ± 1.0
6	427.0 ± 1.0	23	751.0 ± 1.0
7	430.0 ± 1.0	24	763.4 ± 1.0
8	434.0 ± 1.0	25	766.4 ± 1.0
9	439.3 ± 1.0	26	769.8 ± 1.0
10	449.6 ± 1.0	27	772.3 ± 1.0
11	466.7 ± 1.0	28	794.6 ± 1.0
12	475.0 ± 1.0	29	801.0 ± 1.0
13	498.3 ± 1.0	30	811.2 ± 1.2
14	515.2 ± 1.0	31	819.0 ± 1.0
15	568.5 ± 1.5	32	826.2 ± 1.0
16	589.5 ± 2.5	33	840.8 ± 1.2
17	616.5 ± 1.5	34	842.3 ± 1.2

Tabelle 3: Vermessene Peaks des Natriumspektrums

3.2.2 Linien der 1. Nebenserie

Für die Übergänge der ersten Nebenserie kann man davon ausgehen, dass der Korrekturterm nahezu Null ist. Somit gilt:

$$E_{3\mathrm{p}}\,[\mathrm{eV}] = \frac{E_{\mathrm{Ry}}\,[\mathrm{eV}]}{m^2} - \frac{hc}{\lambda_m} \tag{7}$$

$$\Delta E_{3\mathrm{p}}\,[\mathrm{eV}] = \frac{hc}{\lambda_m^2}\Delta\lambda_m \tag{8}$$

bzw. umgestellt:

$$\lambda_m[\mathrm{nm}] = \frac{(hc)\,[\mathrm{nm\,eV}]}{\frac{E_{\mathrm{Ry}}\,[\mathrm{eV}]}{m^2} - E_{3\mathrm{p}}\,[\mathrm{eV}]} \tag{9}$$

$$\Delta\lambda_m[\mathrm{nm}] = \frac{(hc)\,[\mathrm{nm\,eV}]}{\left(\frac{E_{\mathrm{Ry}}\,[\mathrm{eV}]}{m^2} - E_{3\mathrm{p}}\,[\mathrm{eV}]\right)^2}\Delta E_{3\mathrm{p}}\,[\mathrm{eV}] \tag{10}$$

Der Fehler wurde jeweils nach Gauß berechnet. Dabei ist h das Planck'sche Wirkungsquantum, c die Lichtgeschwindigkeit und E_{Ry} die Rydbergenergie mit $E_{\mathrm{Ry}} \approx -13.605\,\mathrm{eV}$. m beschreibt die Hauptquantenzahl des d-Niveaus und $E_{3\mathrm{p}}$ die Energie des 3p-Zustands. Wir ordnen der bei $\lambda = (819.0 \pm 1.0)\,\mathrm{nm}$ gemessenen Linie den Wert $m = 3$ zu und berechnen so $E_{3\mathrm{p}}$ nach Formel (7) und Formel (8) die Energie $E_{3\mathrm{p}}$:

$$\boxed{E_{3\mathrm{p}} = (-3.0256 \pm 0.0027)\,\mathrm{eV}}$$

Dieses Ergebnis können wir nun nutzen, um die Wellenlängen der theoretisch erwarteten Linien mit Formeln (9) und (10) zu berechnen und sie unseren gemessenen zuzuordnen. Die Tabelle sieht dann so aus:

m	$\lambda_{\mathrm{theo}}[\mathrm{nm}]$	Zugeordnete Nummer	$\lambda_{\mathrm{exp}}[\mathrm{nm}]$
3	819.0 ± 1.5	31	819.00 ± 1.0
4	570.0 ± 0.7	15	568.50 ± 1.5
5	499.7 ± 0.6	13	498.30 ± 1.0
6	468.3 ± 0.5	11	466.70 ± 1.0
7	451.2 ± 0.5	10	449.60 ± 1.0
8	440.8 ± 0.4	9	439.30 ± 1.0
9	433.9 ± 0.4	8	434.00 ± 1.0
10	429.1 ± 0.4	7	430.00 ± 1.0
11	425.6 ± 0.4	6	427.00 ± 1.0
12	423.0 ± 0.4	5	420.00 ± 1.0

Tabelle 4: Theoretische Werte der ersten Nebenreihe gegenüber unserer Messungen

Die letzte Zeile ist abgetrennt, da dort die Zuordnung eventuell nicht wirklich passt. Wir werden diese also auch bei dem Fit nicht beachten. Nun wollen wir sowohl Rydbergenergie E_{Ry}, die Energie $E_{3\mathrm{p}}$ und den Korrekturterm Δ_d bestimmen, indem wir an die identifizierten Linien die Funktion

$$\lambda_m[\mathrm{nm}] = \frac{(hc)\,[\mathrm{nm\,eV}]}{\frac{E_{\mathrm{Ry}}\,[\mathrm{eV}]}{(m-\Delta_d)^2} - E_{3\mathrm{p}}\,[\mathrm{eV}]} \tag{11}$$

heranfitten und uns die Parameter ausgeben lassen.

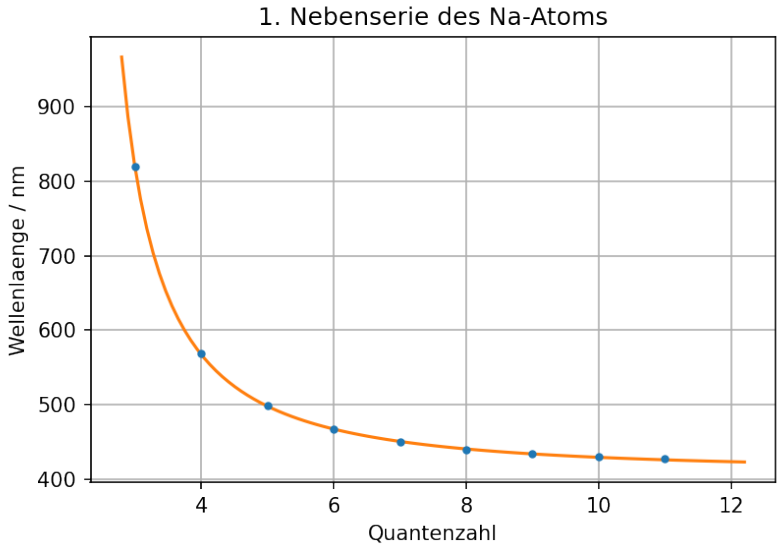


Abbildung 16: Angefittete Kurve an erste Nebenserie

$$\boxed{E_{\mathrm{Ry}} = (-12.7 \pm 0.4)\,\mathrm{eV}} \qquad \boxed{E_{3\mathrm{p}} = (-3.018 \pm 0.005)\,\mathrm{eV}} \qquad \boxed{\Delta_d = (0.09 \pm 0.04)}$$

Wir können nun mit den Formeln

$$\chi^2 = \sum_{i=1}^{N} \frac{(y_i - f(x_i))^2}{\sigma_i^2} \tag{12}$$

$$\chi^2_{\mathrm{red}} = \frac{\chi^2}{f} \tag{13}$$

Die χ^2-Summe sowie die reduzierte χ^2-Summe berechnen. Dabei ist y_i unser Messwert, $f(x_i)$ der laut Fit erwartete Wert, σ_i unser Fehler und f die Anzahl der Freiheitsgrade. Mit Python können wir schließlich auch noch die Fitwahrscheinlichkeit berechnen. Wir erhalten:

$$\boxed{\chi^2 = 5.2} \qquad \boxed{\chi^2_{\mathrm{red}} = 0.74} \qquad \boxed{p_{\mathrm{fit}} = 63.0\%}$$

3.2.3 Linien der 2. Nebenserie

Wir gehen nun ähnlich wie davor vor. Wir ordnen die Linie bei $\lambda = (589.5 \pm 2.5)\,\mathrm{nm}$ dem Übergang $3\mathrm{p} \to 3\mathrm{s}$ zu und berechnen die Energie im Grundzustand nach:

$$E_{3\mathrm{s}} = E_{3\mathrm{p}} - \frac{hc}{\lambda} \tag{14}$$

$$\Delta E_{3\mathrm{s}} = \sqrt{(\Delta E_{3\mathrm{p}})^2 + \left(\frac{hc}{\lambda^2}\Delta\lambda\right)^2} \tag{15}$$

Auch hier wurde der Fehler nach Gauß berechnet. Unser Wert für die Energie ergibt sich als:

$$\boxed{E_{3\mathrm{s}} = (-5.129 \pm 0.009)\,\mathrm{eV}}$$

Mittels folgender Formel kann man daraus den Korrekturfaktor Δ_s berechnen:

$$\Delta_s = 3 - \sqrt{\frac{E_{\mathrm{Ry}}}{E_{3\mathrm{s}}}} \tag{16}$$

$$\Delta(\Delta_s) = \sqrt{\frac{E_{\mathrm{Ry}}}{E_{3\mathrm{s}}^3}} \cdot \frac{\Delta E_{3\mathrm{s}}}{2} \tag{17}$$

Wir erhalten:

$$\boxed{\Delta_s = 1.3713 \pm 0.0015}$$

Diesen Wert können wir nun nutzen, um nach

$$\lambda_m = \frac{ch}{\frac{E_{\mathrm{Ry}}}{(m-\Delta_s)^2} - E_{3\mathrm{p}}} \tag{18}$$

$$\Delta\lambda_m = \frac{ch}{\left(\frac{E_{Ry}}{(m-\Delta_s)^2} - E_{3p}\right)^2} \sqrt{\left(\frac{2E_{\mathrm{Ry}}}{(m-\Delta_s)^3}\Delta(\Delta_s)\right)^2 + (\Delta E_{3p})^2} \tag{19}$$

Die erwarteten Wellenlängen zu berechnen. Diese sind:

m	λ_{theo}[nm]	Zugeordnete Nummer	λ_{exp}[nm]
4	1173 ± 4	-	-
5	622.3 ± 0.9	17	616.5 ± 1.5
6	518.6 ± 0.6	14	515.2 ± 1.0
7	477.6 ± 0.5	12	475.0 ± 1.0
8	456.5 ± 0.5	10	449.6 ± 1.0
9	444.1 ± 0.4	9	439.3 ± 1.0

Tabelle 5: Theoretische Werte der zweiten Nebenreihe gegenüber unserer Messungen

Für $m = 4$ haben wir nicht den richtigen Bereich gemessen, also kann hier logischerweise kein Wert zugeordnet werden. Zwar weichen $m = 8$ und $m = 9$ schon sehr von den erwarteten Werten ab, allerdings haben wir nicht genug Messdaten um die beiden rauszuschmeißen und trotzdem einen halbwegs sinnvollen fit durchzuführen. Also lassen wir sie während dem Fitvorgang drin:

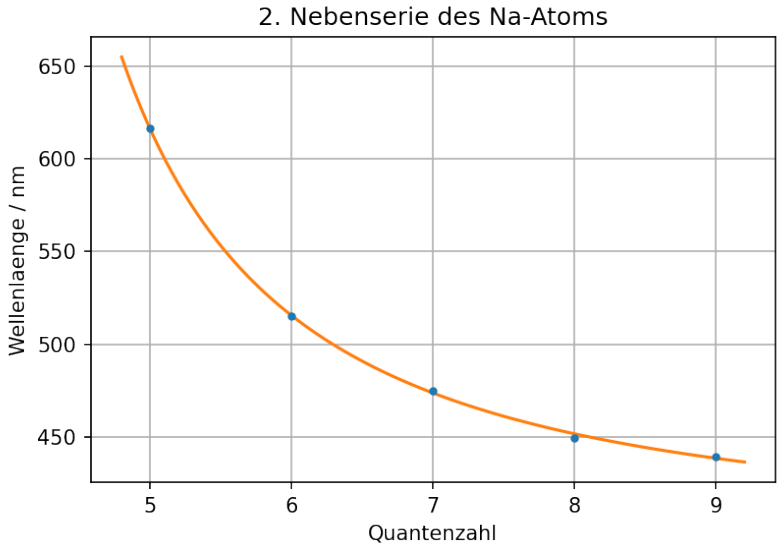


Abbildung 17: Angefittete Kurve an zweite Nebenserie

$$\boxed{E_{\mathrm{Ry}} = (-17.0 \pm 2.9)\,\mathrm{eV}} \quad \boxed{E_{3\mathrm{p}} = (-3.10 \pm 0.03)\,\mathrm{eV}} \quad \boxed{\Delta_s = (1.04 \pm 0.28)}$$

Für die χ^2-Summen und die Fitwahrscheinlichkeit erhalten wir:

$$\boxed{\chi^2 = 6.77} \quad \boxed{\chi^2_{\mathrm{red}} = 3.39} \quad \boxed{p_{\mathrm{fit}} = 3.0\%}$$

3.2.4 Linien der Hauptserie

Zuerst errechnen wir Δ_p, indem wir Formel (16) und Formel (17) nutzen, allerdings Δ_s mit Δ_p und $E_{3\mathrm{s}}$ mit $E_{3\mathrm{p}}$ ersetzen. Das Ergebnis lautet:

$$\boxed{\Delta_p = 0.8794 \pm 0.0019}$$

Anschließend können wir daraus die erwarteten Linien der Hauptreihe berechnen. Dafür verwenden wir Formel (18) und Formel (19), ersetzen jedoch wieder Δ_s mit Δ_p und $E_{3\mathrm{s}}$ mit $E_{3\mathrm{p}}$. Die Wellenlängen lauten:

m	λ_{theo}[nm]	Zugeordnete Nummer	λ_{exp}[nm]
4	332.3 ± 0.9	1	330.4 ± 1.0
5	286.5 ± 0.6	-	-

Tabelle 6: Theoretische Werte der Hauptlinie gegenüber unserer Messungen

Auch hier haben wir für $m = 5$ keinen Wert gemessen, da er außerhalb des von uns angeschauten Bereich liegt.

3.2.5 Qualitative Untersuchung der Spektren verschiedener Strahler

Wir laden alle anderen Strahler in Python und plotten sie in das selbe Diagramm. Dieses sieht dann so aus:

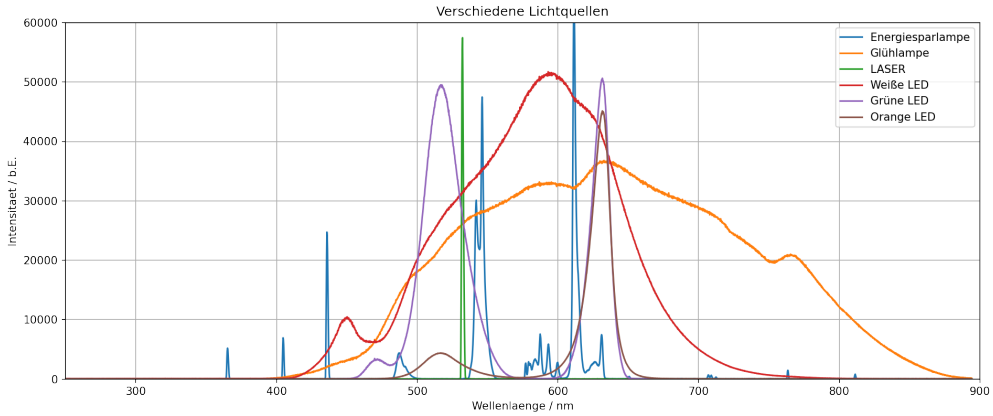


Abbildung 18: Spektren verschiedener Lichtquellen

Wir berechnen die mittlere Wellenlänge der Leuchtmittel mit Python aus den einzelnen gemessenen Daten, indem wir folgende Formel nutzen:

$$\lambda_{\text{mittel}} = \frac{\sum(\lambda_i I_i)}{\sum I_i} \tag{20}$$

Den Fehler schätzen wir aufgrund der Kurven ab. So erhalten wir folgende mittlere Wellenlängen:

Strahler	λ_{mittel} [nm]
Energiesparlampe	566 ± 10
Glühlampe	644 ± 20
Laser	534 ± 3
Weiße LED	582 ± 15
Grüne LED	555 ± 10
Orange LED	612 ± 10

Tabelle 7: Mittlere Wellenlänge verschiedener Leuchtmittel

Man erkennt vor allem, dass die weiße LED, anders als die anderen beiden LEDs nicht aus Rot, Grün und Blau kombiniert wird. Es handelt sich hier also um eine mit einem Fluoreszensmittel beschichtete Leuchtdiode. Diese strahlt dann, ähnlich wie die Glühlampe, ein breiteres Spektrum aus. Allerdings ist dieses mehr auf den sichtbaren Bereich beschränkt und deshalb effizienter. Die weiße LED hat eher kaltes Licht abgestrahlt ($\rightarrow$ mehr Blau) als die Energiesparlampe und die Glühlampe. Man erkennt im Spektrum vor allem im Vergleich zwischen LED und Glühlampe, dass das Glühlampenspektrum mehr rote (langwellige) Komponenten hat, und deswegen Wärmer erscheint. Die Energiesparlampe würden eher als neutral, aber in Richtung kalt eingeordnet werden.

4 Zusammenfassung und Diskussion

Zusammenfassend haben wir in diesem Versuch die Spektren verschiedener Lichtquellen untersucht. Zuerst haben wir das Himmelsspektrum aufgenommen und die Absorption

der Fensterscheibe bestimmt. Konkrete Vergleichswerte haben wir hier nicht, allerdings ist das erhaltene Absorptionsspektrum ungefähr das, was man erwarten würde. Hier scheint also alles funktioniert zu haben. Anschließend haben wir direktes Sonnenlicht aufgenommen und die Fraunhofer-Linien sowie die Balmerserie in unserem Spektrum vermessen. Die Abweichungen zu den im Skript angegebenen Linien lauten:

Element	O_2		H_2		Na		He	Fe & Ca		Mg	Ca	
Abweichungen	2.6	1.3	0.6	0.2	1.2	0	2.8	0	1.6	0.8	0.4	0.8

Tabelle 8: Abweichungen der Fraunhofer-Linien

Übergang	α	β	γ	δ
Abweichungen	0.6	0.2	0	0.2

Tabelle 9: Balmer-Serie

Die Abweichungen haben wir nach folgender Formel bestimmt:

$$\sigma = \frac{|x - y|}{\sqrt{\Delta x^2 + \Delta y^2}} \tag{21}$$

Wir sehen also, dass größtenteils die Abweichungen sehr klein ausgefallen sind. Lediglich die Heliumlinie und eine Sauerstofflinie weisen eine starke Abweichung auf. Bei der Heliumlinie ist das allerdings vermutlich darauf zurückzuführen, dass sie von den beiden Natriumlinien überdeckt wird und so eigentlich nicht erkennbar ist. Wir haben ja sowohl für beide Natriumlinien als auch für die Heliumlinie den selben 'Dip' im Spektrum verwendet. Dass da eine nicht ganz dazu passt war also quasi abzusehen. Auch die Sauerstofflinienabweichung lässt sich erklären: Was wir ausgemessen haben, war die Mitte des doch recht breiten Spalt. Wenn man diese Breite als zwei Linien interpretiert (wie der kleine Hügel in seiner Mitte vermuten lässt), so rutscht die gemessene Sauerstofflinie doch etwas nach links und näher an den erwarteten Wert. Alle anderen Messungen werfen aber keine großen Bedenken auf und sind vermutlich hauptsächlich auf die relative grobe Rundung zurückzuführen.
Im nächsten Versuchsteil haben wir das Spektrum von Natrium aufgenommen, und aus unseren gemessenen Wellenlängen die theoretischen und experimentell bestimmten Werte der Hauptreihe sowie beider Nebenreihen bestimmt. Wir haben zudem geschaut, wie gut ein Fit der theoretischen Formel mit unseren Messungen übereinstimmt und mit ihm sowohl Rydbergenergie als auch $E_{3\mathrm{p}}$ und den Korrekturfaktor bestimmt. Die Abweichungen der ersten Nebenreihe lauten:

m	3	4	5	6	7	8	9	10	11	12
Abweichungen	0	0.9	1.2	1.4	1.4	1.4	0.09	0.8	1.3	2.8

Tabelle 10: Abweichungen der ersten Nebenreihe

Die Abweichungen für die mit dem Fit bestimmten Größen zu Richtwerten (im Falle der Rydbergenergie, wir nehmen den Wert aus dem Skript von $E_{\mathrm{Ry}} = -13.6$) und zuvor berechneten Größen (im Falle von $E_{3\mathrm{p}}$) lauten:

$$\boxed{\sigma_{E_{\mathrm{Ry}}} = 2.3} \quad \boxed{\sigma_{E_{3\mathrm{p}}} = 1.3}$$

Die Abweichungen hier sind also ebenfalls nicht sonderlich groß, ich gehe davon aus dass die Abweichung der Rydbergenergie daher kommt, dass schon leichte Abweichungen in den Messpunkten zu großen Änderungen der Konstanten führt. Zudem ist der Richtwert ohne Fehler angegeben, aber auf eine Nachkommastelle gekürzt. Das trägt sicher auch etwas zum Fehler bei. Eine Fitwahrscheinlichkeit von $p_{\mathrm{fit}} = 63\%$ und eine reduzierte χ^2-Summe von $\chi^2_{\mathrm{red}} = 0.74$ weisen ebenfalls darauf hin, dass unser Fit eigentlich gut gelaufen ist und so nicht von systematischen Fehlern auszugehen ist.
Nun die Abweichungen der Ergebnisse der 2. Nebenserie:

m	4	5	6	7	8	9
Abweichungen	—	3.3	2.9	2.3	6.2	4.5

Tabelle 11: Abweichungen der zweiten Nebenreihe

Die Rydbergenergie sowie $E_{3\mathrm{p}}$ zeigen eine Abweichung von:

$$\boxed{\sigma_{E_{\mathrm{Ry}}} = 1.17} \quad \boxed{\sigma_{E_{3\mathrm{p}}} = 2.7}$$

Hier sind also bis auf die der Rydbergenergie alle Abweichungen signifikant oder nahezu signifikant. Auch dass die Rydbergenergie nicht signifikant abweicht ist auf den im Vergleich zu vorher ziemlich großen Fehler zurückzuführen. Die Fitfunktion scheint also nicht wirklich mit unseren Messwerten übereinzustimmen. Das bestätigt auch die Fitwahrscheinlichkeit von $p_{\mathrm{fit}} = 3.0\%$ mit einer reduzierten χ^2-Summe von $\chi^2_{\mathrm{red}} = 3.39$ (optimal wäre 1). Es liegt also offensichtlich ein systematischer Fehler vor. Denkbar wäre vor allem, dass der Wellenlängenwert, den wir zum Berechnen der theoretischen Werte benutzt haben, besonders Fehlerbehaftet war mit einem zu kleinen tatsächlich angegebenen Fehler. Das würde sich durch alle theoretischen Werte weiterziehen und so den Vergleich immer zu großen Abweichungen führen lassen. Dass der Funktionsfit dann allerdings Probleme aufzeigt, ist nicht direkt gefolgert, wir fitten ja gegen die gemessenen Werte, welche nach der Theorie trotzdem halbwegs richtig sein sollten. Um das zu erklären wäre möglich, dass die falsche Berechnung der theoretischen Werte zur folge hatte, dass wir die falschen Linien des Spektrums zu den Übergängen zugeordnet haben. So würde sie logischerweise nicht mehr durch die Funktion beschrieben werden. Das klingt besonders plausibel, da vor allem am Ende die Differenz zwischen theoretischem Wert und experimentellen Wert so weit auseinander gingen, dass die Zugehörigkeit berechtigterweise anzuzweifeln ist. Aus dem Fit konnten wir sie jedoch leider nicht nehmen, da wir mindestens 4 Variablen für einen Fit mit 3 Parametern brauchen (der nicht einfach exakt die Messwerte durchläuft) und für $m = 4$ ja schon keinen Wert hatten.
Bei der Hauptreihe können wir uns nur die Abweichung einer Wellenlänge anschauen. Diese lautet:

$$\boxed{\sigma_{m=4} = 1.4}$$

Diese Abweichung ist wieder nicht besonders groß, hier kann man also wieder davon ausgehen, dass die Abweichung auf statistische Gründe zurückzuführen sind.

In dem letzten Teil der Auswertung haben wir uns die Spektren verschiedener Strahler angeschaut, und haben dabei unsere Erwartungen bestätigt. Die Energiesparlampe wies mehrere Peaks auf, die farbigen LEDs haben sich aus 3 Farbkomponenten (RGB) zusammengesetzt, die weiße LED und die Glühlampe haben breite, kontinuierliche Spektren aufgezeigt, wobei das der LED besser auf das sichtbare Spektrum angepasst war und der Laser bestand aus einem sehr dünnen Peak.

Alles in allem bin ich recht zufrieden mit dem Ergebnis. Fast alle Messreihen lieferten Ergebnisse, welche die Erwartungen widerspiegelten oder sich mit vorhergehenden Ergebnissen deckten. Nur bei der zweiten Nebenserie schien etwas schiefgelaufen zu sein.
Würde ich den Versuch also erneut durchführen, würde ich eventuell noch mehr Messungen aufnehmen, die jeweils auf andere Bereiche fokussiert sind, um so mit der Integrationszeit eindeutigere Peaks zu erhalten. Hoffentlich würde dass dann die Fehler der theoretischen Berechnungen reduzieren, falls oben erwähnte Theorie zu den Abweichungen wahr ist. Falls möglich würde es sich auch anbieten, die Auswertung direkt vor Ort vorzunehmen, um direkt zu sehen, falls man eine bessere Auflösung eines Bereichs braucht. Die Auflösung des Spektrometers wird vermutlich keinen großen Unterschied gemacht haben, aber vielleicht würde es sich auch lohnen, ein Hochauflösenderes zu nutzen. Zu dem ersten Versuchsteil lässt sich noch sagen, dass ich darauf achten würde, zwischen Messungen, die im besten Fall quantitativ verglichen werden sollen ($\rightarrow$ Messung mit und ohne Fenster), nicht die Integrationszeit zu ändern. Dann ist das Absorptionsspektrum auch quantitativ nutzbar und zeigt nicht nur den groben Verlauf.

5 Anhang

Versuch 234

March 19, 2025

Versuch 234: Lichtquellen

[1]:
```
#imports
import numpy as np
import matplotlib.pyplot as plt
import scipy.constants as c
from scipy.optimize import curve_fit
from scipy.stats import chi2
```

[2]:
```
#Dateipfade
INPUT_DIRECTORY = "../Input_Data/Versuch 234/"
OUTPUT_DIRECTORY = "../Plots/Versuch 234/"
```

[3]:
```
#Aufgabe 1 Messdatennamen
A1_HIMMEL_KEIN_GLAS = "1_Himmel direkt.txt"
A1_HIMMEL_GLAS = "1_sonne mit glas.txt"
A1_SONNE = "1_Sonne direkt.txt"
```

[4]:
```
#Aufgabe 2 Messdatennamen
A2_ENERGIESPARLAMPE = "2_Energiesparlampe.txt"
A2_GLUEHLAMPE = "2_Glühlampe.txt"
A2_LASER_UEBERBELICHTET = "2_Laser grün überbelichtet.txt"
A2_LASER = "2_Laser grün.txt"
A2_LED_WEISS = "2_LED weiß.txt"
A2_LED_GRUEN = "2_Minzgrün LED.txt"
A2_LED_ORANGE = "2_Sonnenaufgangsorange.txt"
```

[5]:
```
#Aufgabe 3 Messdatennamen
A3_MAX1 = "3_Natriumlampe max.txt"
A3_MID = "3_Natriumlampe mid.txt"
A3_MAX2 = "3_Natriumlampe other max.txt"
A3_ALL = "3_Natriumlampe.txt"
```

[6]:
```
#Aufgabe 4 Messdatennamen
A4_570_416 = "4_Natriumlampe 570,416 Linien.txt"
```

[7]:
```
#Aufgabe 5 Messdatennamen
A5_650_850 = "5_Natriumlampe 650-850.txt"
```

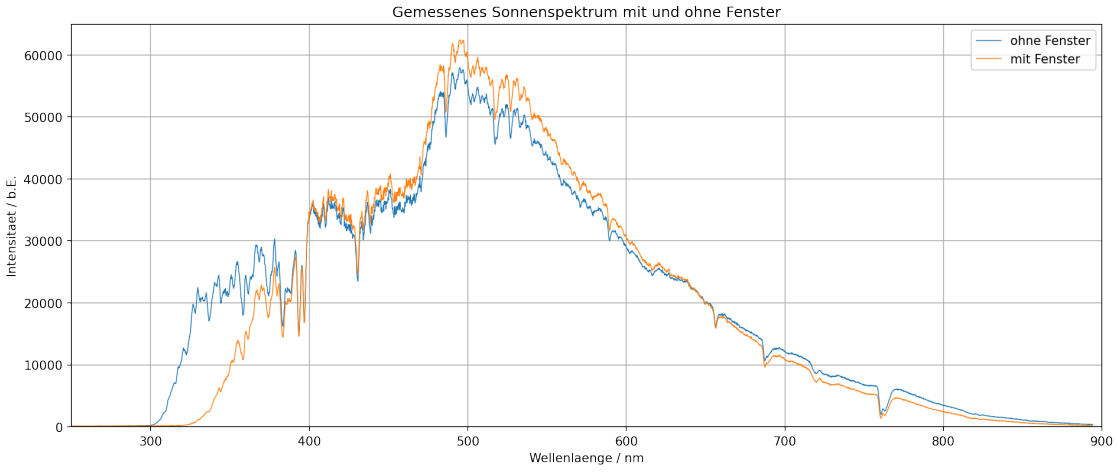
```
A5_650_850_UEBERBELICHTET = "5_Natriumlampe 650-850 überbelichtet.txt"
```

[8]:
```
#Benötigte Funktionen
def comma_to_float(valstr):
    return float(valstr.decode("utf-8").replace(',','.'))
def fit_func1(m,E_Ry,E_3p,korrd):
    return 1.2398E3/(E_Ry/(m-korrd)**2-E_3p)
def fit_func2(m,E_Ry,E_3p,korrs):
    return 1.2398E3/(E_Ry/(m-korrs)**2-E_3p)
```

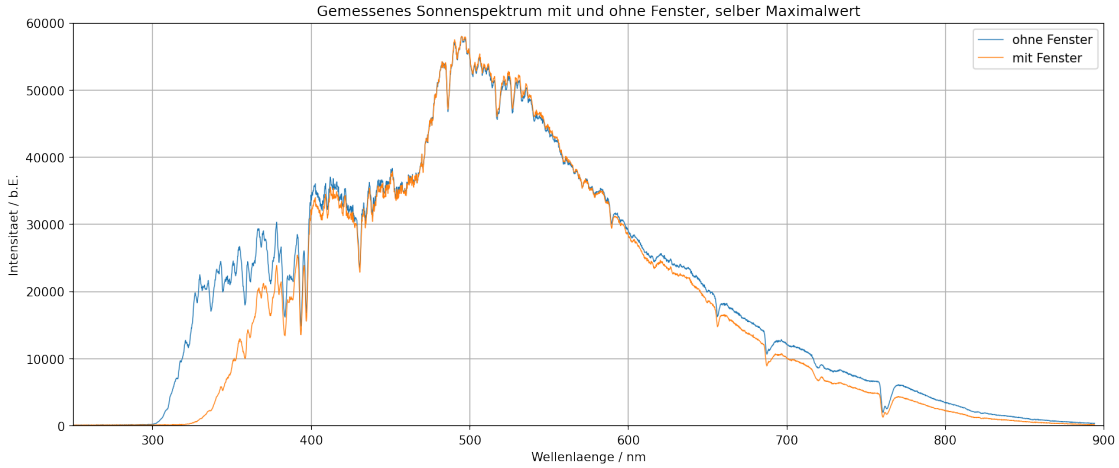
Aufgabe 1: Absorption von Glas

[9]:
```
lamb_og, inten_og=np.loadtxt(INPUT_DIRECTORY + A1_HIMMEL_KEIN_GLAS,␣
 ↪skiprows=17, converters= {0:comma_to_float, 1:comma_to_float}, comments='>',␣
 ↪unpack=True)
lamb_mg, inten_mg=np.loadtxt(INPUT_DIRECTORY + A1_HIMMEL_GLAS, skiprows=17,␣
 ↪converters= {0:comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
lamb_sonne, inten_sonne=np.loadtxt(INPUT_DIRECTORY + A1_SONNE, skiprows=17,␣
 ↪converters= {0:comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
```

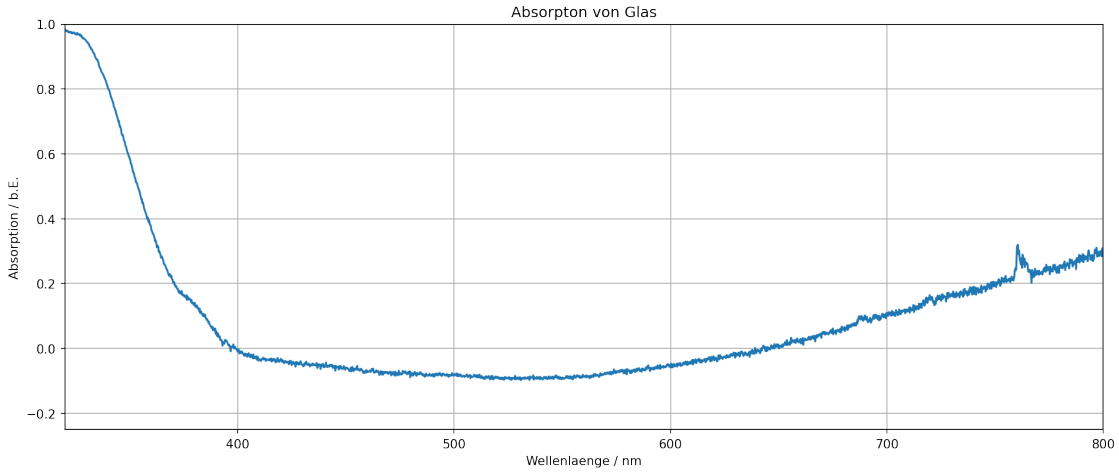
[10]:
```
plt.figure(dpi=150,figsize=(15,6))
plt.plot(lamb_og, inten_og, label='ohne Fenster', linewidth=0.7)
plt.plot(lamb_mg, inten_mg, label='mit Fenster', linewidth=0.7)
plt.title('Gemessenes Sonnenspektrum mit und ohne Fenster')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.legend()
plt.grid()
plt.ylim((0,65000))
plt.xlim((250,900))
plt.savefig(OUTPUT_DIRECTORY + "Plot1.png")
```



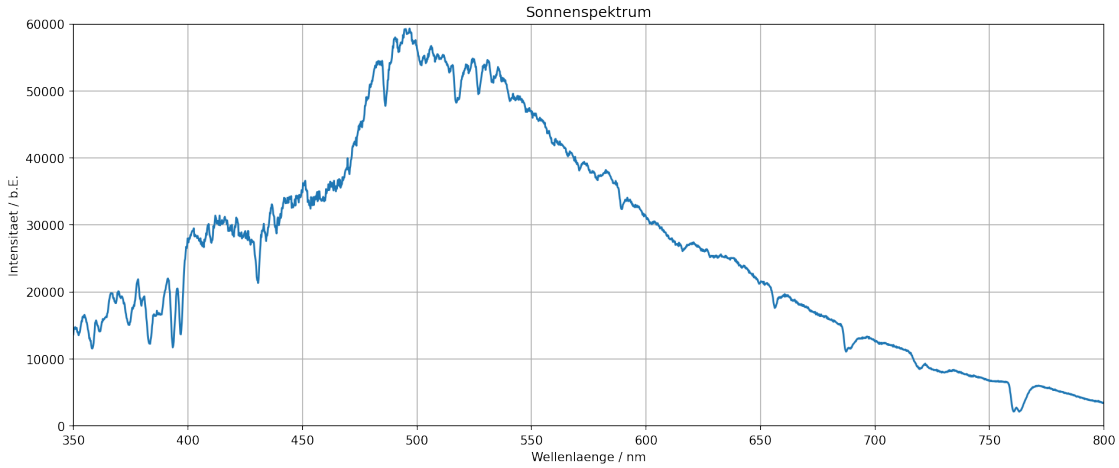
[11]:
```
#Normalisiert plotten
plt.figure(dpi=150,figsize=(15,6))
plt.plot(lamb_og, 5.8e4*inten_og/np.max(inten_og), label='ohne Fenster',␣
 ↪linewidth=0.7)
plt.plot(lamb_mg, 5.8e4*inten_mg/np.max(inten_mg), label='mit Fenster',␣
 ↪linewidth=0.7)
plt.title('Gemessenes Sonnenspektrum mit und ohne Fenster, selber Maximalwert')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.legend()
plt.grid()
plt.ylim((0,60000))
plt.xlim((250,900))
plt.savefig(OUTPUT_DIRECTORY + "Plot2.png")
```



[12]:
```
#Absorption berechnen:
A = 1 - inten_mg/inten_og
plt.figure(dpi=150,figsize=(15,6))
plt.grid()
plt.plot(lamb_mg, A)
plt.title('Absorpton von Glas')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Absorption / b.E.')
plt.ylim((-0.25,1))
plt.xlim((320,800))
plt.savefig(OUTPUT_DIRECTORY + "Plot3.png")
```



[13]:
```
plt.figure(dpi=150,figsize=(15,6))
plt.grid()
plt.plot(lamb_sonne, inten_sonne)
plt.title('Sonnenspektrum')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.ylim((0,60000))
plt.xlim((350,800))
plt.savefig(OUTPUT_DIRECTORY + "Plot4.png")
```



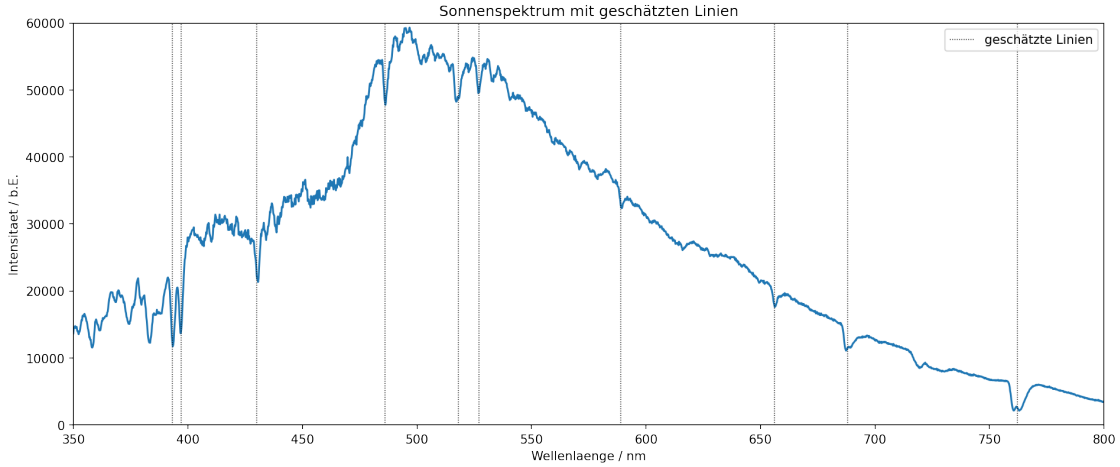
[14]:
```
plt.figure(dpi=150,figsize=(15,6))

plt.plot(lamb_sonne, inten_sonne)
```

```
frauenhofer_lines = [393, 397, 430, 486, 518, 527,589,656,688,762]

labeled = False
for i in frauenhofer_lines:
    plt.axvline(i, linestyle="dotted", color = "black", linewidth = 0.7, label␣
 ↪= "geschätzte Linien" if not labeled else None)
    labeled = True

plt.title('Sonnenspektrum mit geschätzten Linien')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.ylim((0,60000))
plt.xlim((350,800))
plt.legend()
plt.savefig(OUTPUT_DIRECTORY + "Plot5.png")
```



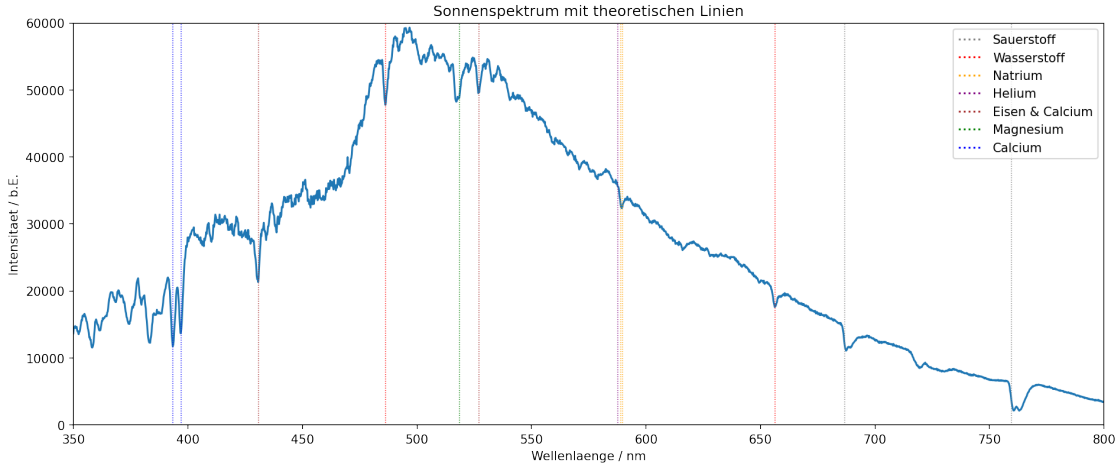
[15]:
```
plt.figure(dpi=150,figsize=(15,6))
plt.plot(lamb_sonne, inten_sonne)

fraunhofer_lines = {
    "Sauerstoff": ("gray", [759.4, 686.7]),
    "Wasserstoff": ("red", [656.3, 486.1]),
    "Natrium": ("orange", [589.6, 589.0]),
    "Helium": ("purple", [587.6]),
    "Eisen & Calcium": ("brown", [527.0, 430.8]),
    "Magnesium": ("green", [518.4]),
    "Calcium": ("blue", [396.8, 393.4]),
}
for element, (color, wavelengths) in fraunhofer_lines.items():
    for wavelength in wavelengths:
```

```
        plt.axvline(wavelength, color=color, linestyle="dotted", linewidth = 0.
↪7)

handles = [plt.Line2D([0], [0], color=color, linestyle="dotted", label=element)
           for element, (color, _) in fraunhofer_lines.items()]
plt.legend(handles=handles, loc="upper right")


plt.title('Sonnenspektrum mit theoretischen Linien')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.ylim((0,60000))
plt.xlim((350,800))
plt.savefig(OUTPUT_DIRECTORY + "Plot6.png")
```



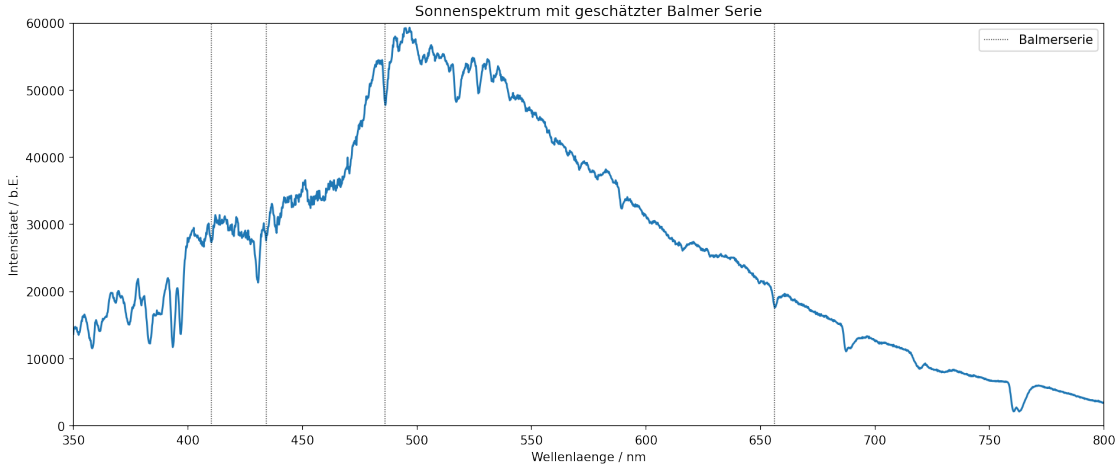
[16]:
```
plt.figure(dpi=150,figsize=(15,6))
plt.plot(lamb_sonne, inten_sonne)

balmer_serie = [656,486,434,410]

labeled = False
for i in balmer_serie:
    plt.axvline(i, linestyle="dotted", color = "black", linewidth = 0.7, label␣
↪= "Balmerserie" if not labeled else None)
    labeled = True

plt.title('Sonnenspektrum mit geschätzter Balmer Serie')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.ylim((0,60000))
plt.xlim((350,800))
```

```
plt.legend()
plt.savefig(OUTPUT_DIRECTORY + "Plot7.png")
```

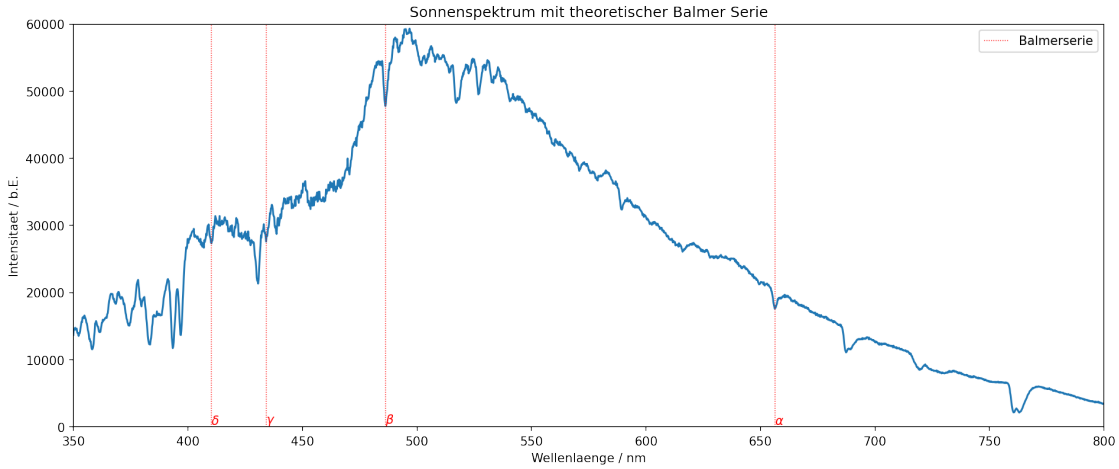


[17]:
```
plt.figure(dpi=150,figsize=(15,6))
plt.plot(lamb_sonne, inten_sonne)

balmer_serie = [656.3,486.1,434.0,410.1]
name = ['$\\alpha$', '$\\beta$', '$\\gamma$', '$\\delta$']

labeled = False
j=0
for i in balmer_serie:
    plt.axvline(i, linestyle="dotted", color = "red", linewidth = 0.7, label =␣
 ↪"Balmerserie" if not labeled else None)
    plt.text(i, 1.05, name[j], color="red", rotation=0,␣
 ↪verticalalignment="bottom", horizontalalignment="left")
    j+=1
    labeled = True

plt.title('Sonnenspektrum mit theoretischer Balmer Serie')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.ylim((0,60000))
plt.xlim((350,800))
plt.legend()
plt.savefig(OUTPUT_DIRECTORY + "Plot8.png")
```



Aufgabe 2: Natriumspektrum

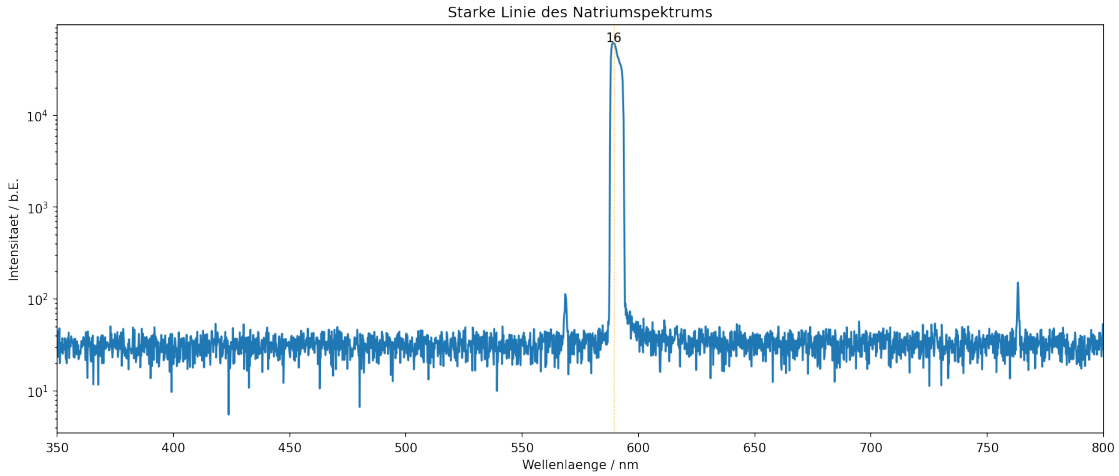
[18]:
```
visible_peaks = [589.5]

lamb, inten=np.loadtxt(INPUT_DIRECTORY + A3_ALL, skiprows=17, converters= {0:
 ↪comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
inten = inten - np.min(inten) if np.min(inten) <=0 else inten
plt.figure(dpi=150,figsize=(15,6))
plt.plot(lamb, inten)

leftlim = 350
rightlim = 800

for i in visible_peaks:
    plt.axvline(i, linestyle="dotted", color = "orange", linewidth = 0.7)
    plt.text(i, 6e4, "16", color="black", rotation=0,␣
 ↪verticalalignment="bottom", horizontalalignment="center")

plt.title('Starke Linie des Natriumspektrums')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.yscale('log')
plt.xlim(leftlim,rightlim)
plt.savefig(OUTPUT_DIRECTORY + "Plot9.png")
```



[19]:
```
#Alle peaks einspeichern
visible_peaks = [330.4,394.9,404.3,416,420,427,430,434,439.3,449.6,466.
↪7,475,498.3,515.2,568.5,589.5,616.5,696.5,706.7,714.7,727.3,738.4,751,763.
↪4,766.4,769.8,772.3,794.6,801,811.2,819,826.2,840.8,842.3]
```

[20]:
```
lamb, inten=np.loadtxt(INPUT_DIRECTORY + A3_MID, skiprows=17, converters= {0:
↪comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
plt.figure(dpi=150,figsize=(15,6))
plt.plot(lamb, inten)

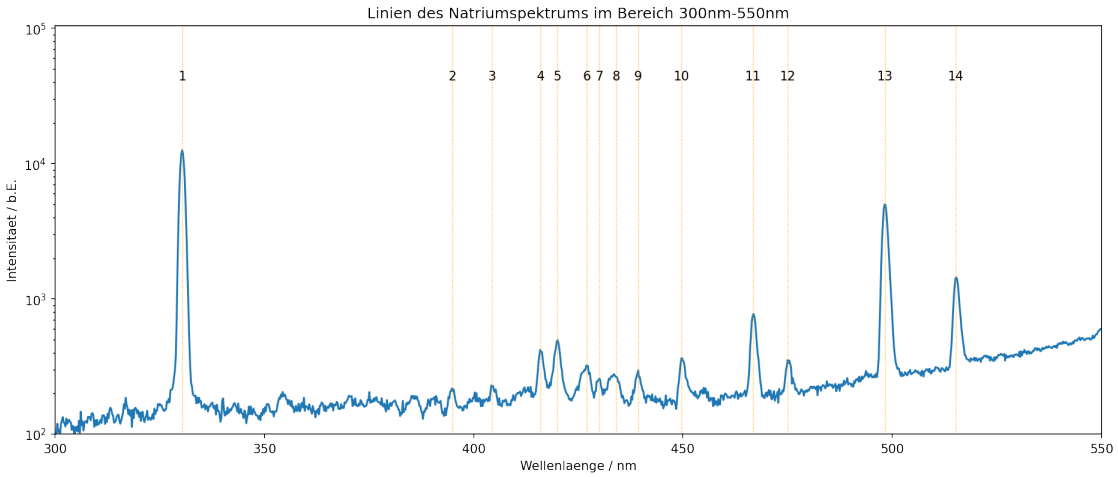
#Festlegen der Grenzen des Plots und ob Beschriftungsoverlap vermieden werden␣
↪muss
leftlim = 300
rightlim = 550
overlap = False

#Starten der Beschriftung
j = 1

for i in visible_peaks: #Alle sichtbaren Peaks durchgehen
    if(leftlim < i < rightlim): #Wenn er in den Bounds des Plots ist:
        plt.axvline(i, linestyle="dotted", color = "orange", linewidth = 0.7)␣
↪#Orangene Linie an der Stelle platzieren
        plt.text(i, 6e4 if (j%2 == 0 and overlap) else 4e4, str(j),␣
↪color="black", rotation=0, verticalalignment="bottom",␣
↪horizontalalignment="center") #Und markieren, der wievielte peak es ist␣
↪(overlap verändert nur die Anordnung)
    j+=1

plt.title('Linien des Natriumspektrums im Bereich 300nm-550nm')
```

```
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.yscale('log')
plt.ylim(100)
plt.xlim(leftlim,rightlim)
plt.savefig(OUTPUT_DIRECTORY + "Plot10.png")
```



[21]:
```
lamb, inten=np.loadtxt(INPUT_DIRECTORY + A4_570_416, skiprows=17, converters=␣
↪{0:comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
plt.figure(dpi=150,figsize=(15,6))
plt.plot(lamb, inten)

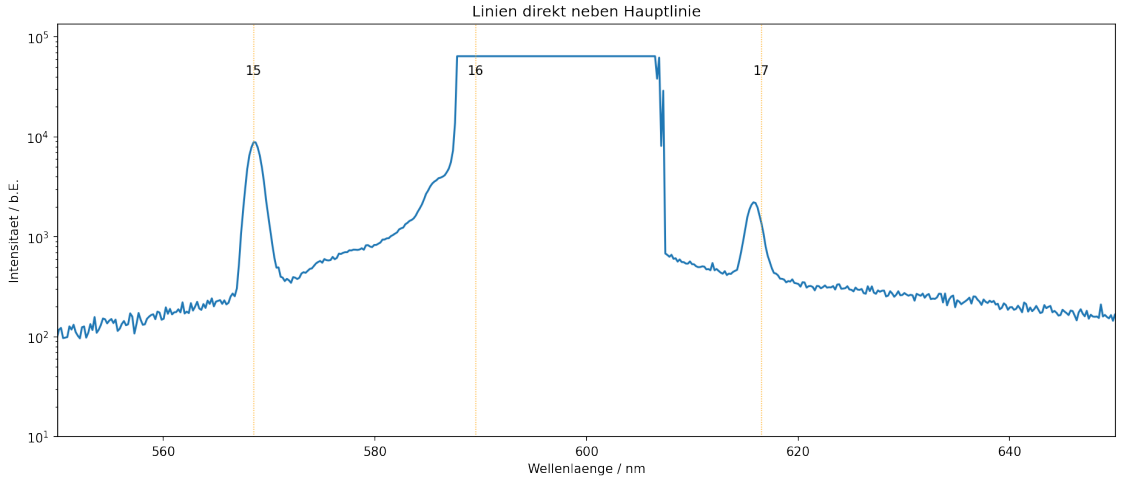
leftlim = 550
rightlim = leftlim + 100
overlap = False

j = 1

for i in visible_peaks:
    if(leftlim < i < rightlim):
        plt.axvline(i, linestyle="dotted", color = "orange", linewidth = 0.7)
        plt.text(i, 6e4 if (j%2 == 0 and overlap) else 4e4, str(j),␣
↪color="black", rotation=0, verticalalignment="bottom",␣
↪horizontalalignment="center")
    j+=1

plt.title('Linien direkt neben Hauptlinie')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.yscale('log')
```

```
plt.ylim(10)
plt.xlim(leftlim, rightlim)
plt.savefig(OUTPUT_DIRECTORY + "Plot11.png")
```



[22]:
```
lamb, inten=np.loadtxt(INPUT_DIRECTORY + A5_650_850, skiprows=17, converters=␣
 ↪{0:comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
plt.figure(dpi=150,figsize=(15,6))
plt.plot(lamb, inten)

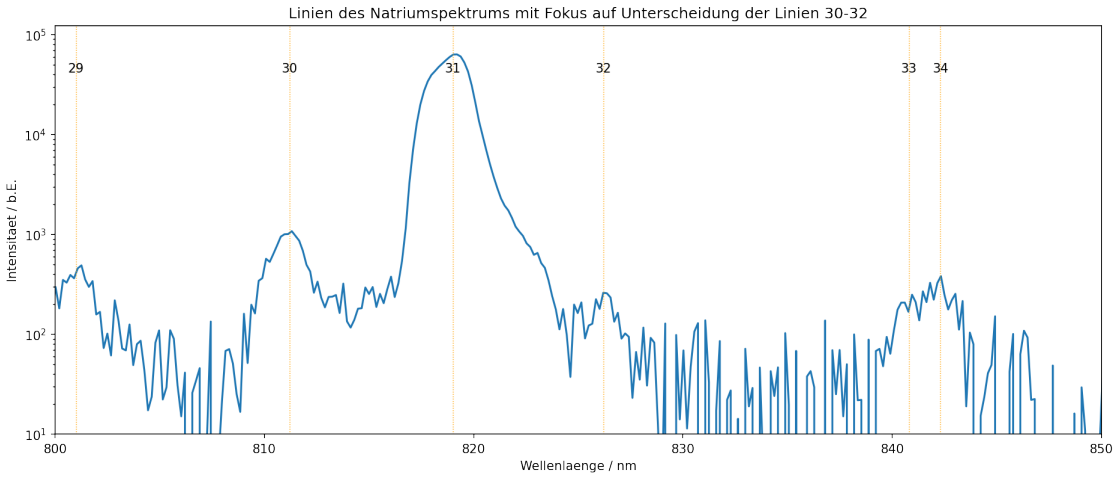
leftlim = 800
rightlim = 850
overlap = False

j = 1

for i in visible_peaks:
    if(leftlim < i < rightlim):
        plt.axvline(i, linestyle="dotted", color = "orange", linewidth = 0.7)
        plt.text(i, 6e4 if (j%2 == 0 and overlap) else 4e4, str(j),␣
 ↪color="black", rotation=0, verticalalignment="bottom",␣
 ↪horizontalalignment="center")
    j+=1

plt.title('Linien des Natriumspektrums mit Fokus auf Unterscheidung der Linien␣
 ↪30-32')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.yscale('log')
plt.ylim(10)
plt.xlim(leftlim,rightlim)
```

```
plt.savefig(OUTPUT_DIRECTORY + "Plot12.png")
```



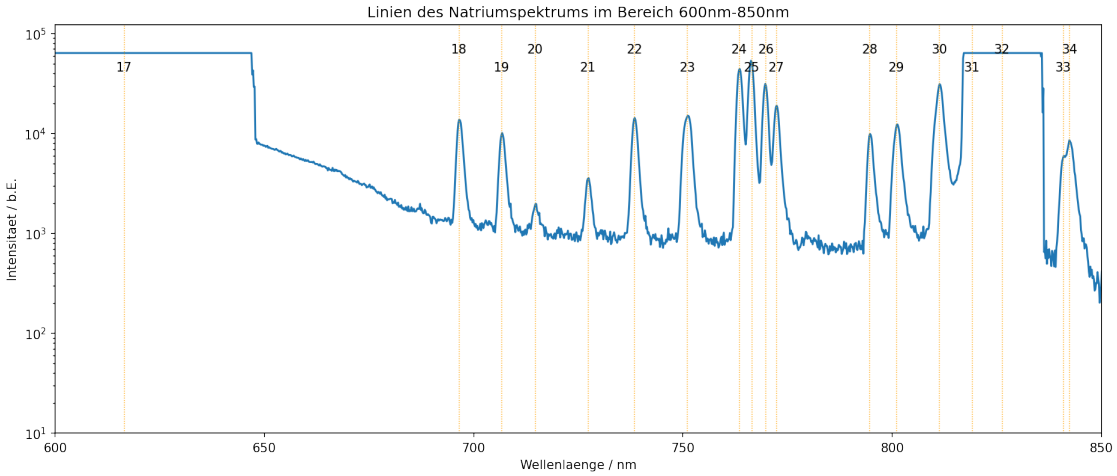
[23]:
```
lamb, inten=np.loadtxt(INPUT_DIRECTORY + A5_650_850_UEBERBELICHTET,␣
↪skiprows=17, converters= {0:comma_to_float, 1:comma_to_float}, comments='>',␣
↪unpack=True)
plt.figure(dpi=150,figsize=(15,6))
plt.plot(lamb, inten)

leftlim = 600
rightlim = 850
overlap = True

j = 1

for i in visible_peaks:
    if(leftlim < i < rightlim):
        plt.axvline(i, linestyle="dotted", color = "orange", linewidth = 0.7)
        plt.text(i, 6e4 if (j%2 == 0 and overlap) else 4e4, str(j),␣
↪color="black", rotation=0, verticalalignment="bottom",␣
↪horizontalalignment="center")
    j+=1

plt.title('Linien des Natriumspektrums im Bereich 600nm-850nm')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.yscale('log')
plt.ylim(10)
plt.xlim(leftlim,rightlim)
plt.savefig(OUTPUT_DIRECTORY + "Plot13.png")
```



1. Nebenserie berechnen

[24]:
```
#Berechnen von E_3p
RyE = -1*c.h*c.c*c.Rydberg/c.eV
m = 3
lamb_3 = 819
Delta_lamb_3 = 1.5
E_3p = (RyE*c.eV / m**2 - c.h * c.c / (lamb_3*1e-9))/c.eV
Delta_E_3p = c.h * c.c / (lamb_3*1e-9)**2 * Delta_lamb_3*1e-9 / c.eV
E_3p, Delta_E_3p
```

[24]:
```
(-3.0255922570506444, 0.00277261644087537)
```

[25]:
```
def lamb_m(m):
    lamb = c.h * c.c / (RyE * c.eV / m**2 - E_3p * c.eV)*1e9
    Delta_lamb = c.h * c.c * Delta_E_3p * c.eV / (RyE * c.eV / m**2 - E_3p * c.
 ↪eV)**2 * 1e9
    return lamb, Delta_lamb
```

[26]:
```
nebenserie1 = np.array([])
Delta_nebenserie1 = np.array([])
for m in range(3,13):
    print('m={m:2d}, lambda={l:6.2f} ± {k:6.2f}'.
 ↪format(m=m,l=lamb_m(m)[0],k=lamb_m(m)[1]))
    nebenserie1,Delta_nebenserie1 = np.append(nebenserie1,lamb_m(m)[0]), np.
 ↪append(Delta_nebenserie1,lamb_m(m)[1])
```

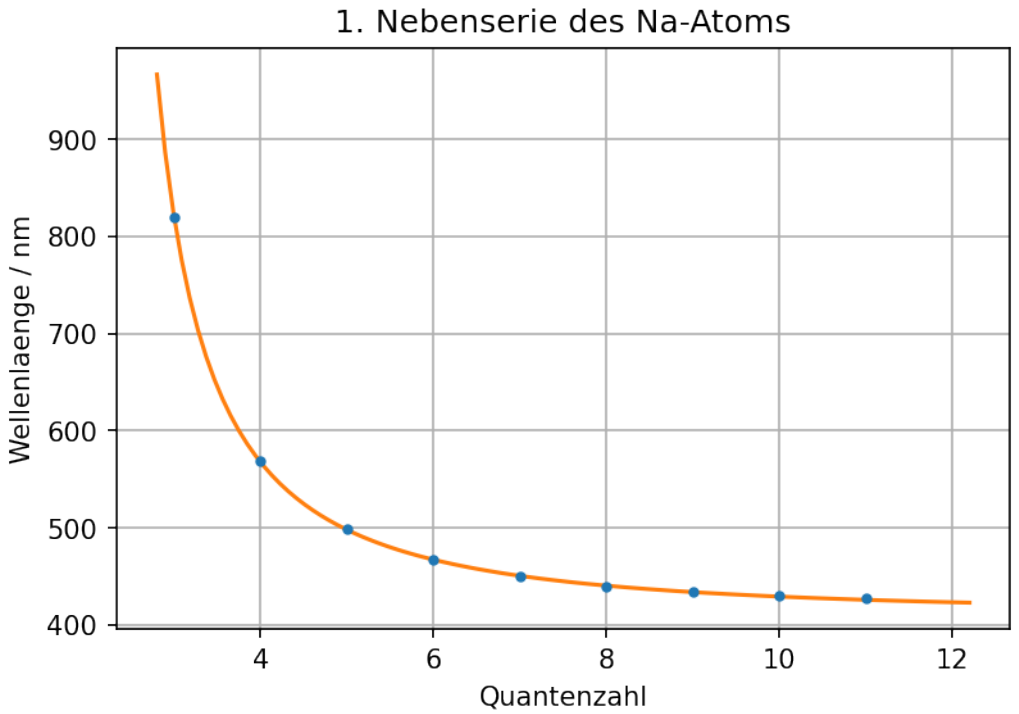
```
m= 3, lambda=819.00 ±   1.50
m= 4, lambda=569.98 ±   0.73
m= 5, lambda=499.66 ±   0.56
m= 6, lambda=468.28 ±   0.49
```

```
m= 7, lambda=451.19 ±   0.46
m= 8, lambda=440.75 ±   0.43
m= 9, lambda=433.87 ±   0.42
m=10, lambda=429.08 ±   0.41
m=11, lambda=425.60 ±   0.41
m=12, lambda=422.99 ±   0.40
```

[27]:
```
best_guess1 = np.array([819,568.5,498.3,466.7,449.6,439.3,434,430,427,420])
Delta_best_guess1 = np.array([2,1.5,1,1,1,1,1,1,1,1])
quantenzahl1 = np.arange(3,13)
```

[28]:
```
para = [-13.6,-3,-0.02]
popt, pcov = curve_fit(fit_func1, quantenzahl1[:-1], best_guess1[:-1],␣
 ↪sigma=Delta_best_guess1[:-1],p0=para)
x=np.linspace(2.8,12.2, 100)
```

[29]:
```
plt.figure(dpi=150)
plt.errorbar(quantenzahl1[:-1],best_guess1[:-1], Delta_best_guess1[:-1], fmt=".
 ↪")
plt.plot(x,fit_func1(x, *popt))
plt.xlabel('Quantenzahl')
plt.ylabel('Wellenlaenge / nm')
plt.title('1. Nebenserie des Na-Atoms')
plt.grid()
plt.savefig(OUTPUT_DIRECTORY + "Plot14.png")
```



[30]:
```
print("E_Ry=",popt[0], ", Standardfehler=", np.sqrt(pcov[0][0]))
print("E_3p=",popt[1], ", Standardfehler=", np.sqrt(pcov[1][1]))
print("D_d=",popt[2], ", Standardfehler=", np.sqrt(pcov[2][2]))
```

```
E_Ry= -12.705154597213117 , Standardfehler= 0.3709941502613164
E_3p= -3.017751979086327 , Standardfehler= 0.005381971162288037
D_d= 0.09401632072351973 , Standardfehler= 0.038705843974858864
```

[31]:
```
chi2_=np.sum((fit_func1(quantenzahl1[:-1],*popt)-best_guess1[:-1])**2/
 ↪Delta_best_guess1[:-1]**2)
dof=len(quantenzahl1)-3 #dof:degrees of freedom, Freiheitsgrad
chi2_red=chi2_/dof
print("chi2=", chi2_)
print("chi2_red=",chi2_red)
prob=round(1-chi2.cdf(chi2_,dof),2)*100
print("Wahrscheinlichkeit:", prob,"%")
```

```
chi2= 5.206571317878185
chi2_red= 0.7437959025540265
Wahrscheinlichkeit: 63.0 %
```

2. Nebenserie

[32]:
```
m = 3
lamb_3 = 589.5
Delta_lamb_3 = 2.5
E_3s = E_3p - c.h*c.c*1e9/c.eV/lamb_3
Delta_E_3s = np.sqrt((Delta_E_3p)**2 + (1.2398e3/lamb_3**2 * Delta_lamb_3)**2)
E_3s, Delta_E_3s
```

[32]:
```
(-5.128801730048105, 0.009340173531670487)
```

[33]:
```
korrs = 3-np.sqrt(RyE/E_3s)
Delta_korrs =np.sqrt(RyE/E_3s**3)*Delta_E_3s/2
korrs,Delta_korrs
```

[33]:
```
(1.3712576818115578, 0.0014830692129438811)
```

[34]:
```
def lamb_m(m):
    lamb = c.h * c.c / (RyE * c.eV / (m-korrs)**2 - E_3p * c.eV)*1e9
    Delta_lamb = np.sqrt(((-2*RyE*c.c*c.eV*c.h/((-korrs + m)**3*(-E_3p*c.eV +␣
 ↪RyE*c.eV/(-korrs + m)**2)**2))*Delta_korrs)**2 + ((c.c*c.eV*c.h/(-E_3p*c.eV␣
 ↪+ RyE*c.eV/(-korrs + m)**2)**2)*Delta_E_3p)**2)*1e9
    return lamb, Delta_lamb
```

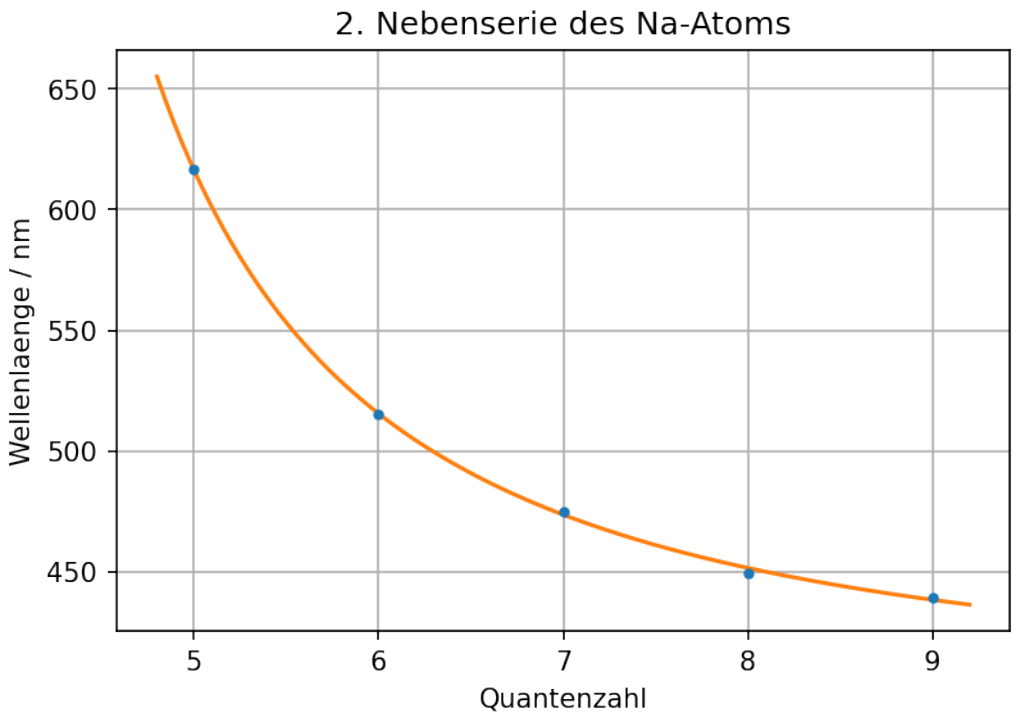
[35]:
```
nebenserie2 = np.array([])
Delta_nebenserie2 = np.array([])
for m in range(4,10):
    print('m={m:2d}, lambda={l:6.2f} ± {k:6.2f}'.
 ↪format(m=m,l=lamb_m(m)[0],k=lamb_m(m)[1]))
    nebenserie2,Delta_nebenserie2 = np.append(nebenserie2,lamb_m(m)[0]), np.
 ↪append(Delta_nebenserie2,lamb_m(m)[1])
```

```
m= 4, lambda=1173.33 ±   3.95
m= 5, lambda=622.31 ±   0.91
m= 6, lambda=518.64 ±   0.61
m= 7, lambda=477.57 ±   0.51
m= 8, lambda=456.50 ±   0.47
m= 9, lambda=444.10 ±   0.44
```

[36]:
```
best_guess2 = np.array([616.5,515.2,475,449.6,439.3])
Delta_best_guess2 = np.array([1.5,1,1,1,1])
quantenzahl2 = np.arange(5,10)
```

[37]:
```
para = [-13.6,-3,-0.02]
popt, pcov = curve_fit(fit_func2, quantenzahl2, best_guess2,␣
 ↪sigma=Delta_best_guess2,p0=para)
x=np.linspace(4.8,9.2, 100)
```

[38]:
```
plt.figure(dpi=150)
plt.errorbar(quantenzahl2,best_guess2,Delta_best_guess2, fmt=".")
plt.plot(x,fit_func2(x, *popt))
plt.xlabel('Quantenzahl')
plt.ylabel('Wellenlaenge / nm')
plt.title('2. Nebenserie des Na-Atoms')
plt.grid()
plt.savefig(OUTPUT_DIRECTORY + "Plot15.png")
```



[39]:
```
print("E_Ry=",popt[0], ", Standardfehler=", np.sqrt(pcov[0][0]))
print("E_3p=",popt[1], ", Standardfehler=", np.sqrt(pcov[1][1]))
print("D_s=",popt[2], ", Standardfehler=", np.sqrt(pcov[2][2]))
```

```
E_Ry= -16.993792033677714 , Standardfehler= 2.8927330632762995
E_3p= -3.096572323229392 , Standardfehler= 0.034494753964011506
D_s= 1.0431740130059686 , Standardfehler= 0.2823254551910686
```

[40]:
```
chi2_=np.sum((fit_func2(quantenzahl2,*popt)-best_guess2)**2/
 ↪Delta_best_guess2**2)
dof=len(quantenzahl2)-3
chi2_red=chi2_/dof
print("chi2=", chi2_)
print("chi2_red=",chi2_red)
prob=round(1-chi2.cdf(chi2_,dof),2)*100
print("Wahrscheinlichkeit:", prob,"%")
```

```
chi2= 6.773182678630892
chi2_red= 3.386591339315446
Wahrscheinlichkeit: 3.0 %
```

Hauptserie

[41]:
```
korrp = 3-np.sqrt(RyE/E_3p)
Delta_korrp =np.sqrt(RyE/E_3p**3)*Delta_E_3p
korrp,Delta_korrp
```

[41]:
```
(0.8794176832486378, 0.0019432761906212112)
```

[42]:
```
def lamb_m(m):
    lamb = c.h * c.c / (RyE * c.eV / (m-korrp)**2 - E_3s * c.eV)*1e9
    Delta_lamb = np.sqrt(((-2*RyE*c.c*c.eV*c.h/((-korrp + m)**3*(-E_3s*c.eV +␣
 ↪RyE*c.eV/(-korrp + m)**2)**2))*Delta_korrp)**2 + ((c.c*c.eV*c.h/(-E_3s*c.eV␣
 ↪+ RyE*c.eV/(-korrp + m)**2)**2)*Delta_E_3s)**2)*1e9
    return lamb, Delta_lamb
```

[43]:
```
hauptserie = np.array([])
Delta_hauptserie = np.array([])
for m in range(4,6):
    print('m={m:2d}, lambda={l:6.2f} ± {k:6.2f}'.
 ↪format(m=m,l=lamb_m(m)[0],k=lamb_m(m)[1]))
    hauptserie,Delta_hauptserie = np.append(hauptserie,lamb_m(m)[0]), np.
 ↪append(Delta_hauptserie,lamb_m(m)[1])
```

```
m= 4, lambda=332.25 ±   0.85
m= 5, lambda=286.50 ±   0.62
```

Qualitatives Plotten der Lichtquellen

[44]:
```
plt.figure(dpi=150,figsize=(15,6))
lamb, inten=np.loadtxt(INPUT_DIRECTORY + A2_ENERGIESPARLAMPE, skiprows=17,␣
 ↪converters= {0:comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
print("Energiesparlampe & $" + str(np.sum(lamb * inten) / np.sum(inten)) +␣
 ↪"$\\\\")
plt.plot(lamb,inten, label = "Energiesparlampe")
lamb, inten=np.loadtxt(INPUT_DIRECTORY + A2_GLUEHLAMPE, skiprows=17,␣
 ↪converters= {0:comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
plt.plot(lamb,inten, label = "Glühlampe")
print("Glühlampe & $" + str(np.sum(lamb * inten) / np.sum(inten)) + "$\\\\")
lamb, inten=np.loadtxt(INPUT_DIRECTORY + A2_LASER, skiprows=17, converters= {0:
 ↪comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
plt.plot(lamb,inten, label = "LASER")
print("Laser & $" + str(np.sum(lamb * inten) / np.sum(inten)) + "$\\\\")
lamb, inten=np.loadtxt(INPUT_DIRECTORY + A2_LED_WEISS, skiprows=17, converters=␣
 ↪{0:comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
plt.plot(lamb,inten, label = "Weiße LED")
print("Weiße LED & $" + str(np.sum(lamb * inten) / np.sum(inten)) + "$\\\\")
lamb, inten=np.loadtxt(INPUT_DIRECTORY + A2_LED_GRUEN, skiprows=17, converters=␣
 ↪{0:comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
plt.plot(lamb,inten, label = "Grüne LED")
```

```
print("Grüne LED & $" + str(np.sum(lamb * inten) / np.sum(inten)) + "$\\\\")
lamb, inten=np.loadtxt(INPUT_DIRECTORY + A2_LED_ORANGE, skiprows=17,␣
 ↪converters= {0:comma_to_float, 1:comma_to_float}, comments='>', unpack=True)
plt.plot(lamb,inten, label = "Orange LED")
print("Orange LED & $" + str(np.sum(lamb * inten) / np.sum(inten)) + "$\\\\")
plt.title('Verschiedene Lichtquellen')
plt.xlabel('Wellenlaenge / nm')
plt.ylabel('Intensitaet / b.E.')
plt.legend()
plt.grid()
plt.ylim((0,60000))
plt.xlim((250,900))
plt.savefig(OUTPUT_DIRECTORY + "Plot16.png")
```

```
Energiesparlampe & $566.286220742088$\\
Glühlampe & $643.7478492222596$\\
Laser & $534.3535325837951$\\
Weiße LED & $582.3052054868791$\\
Grüne LED & $555.3210361692156$\\
Orange LED & $611.6131715590665$\\
```

