

Versuch 255 - Röntgenspektrometer

Marius Mangold

January 3, 2022

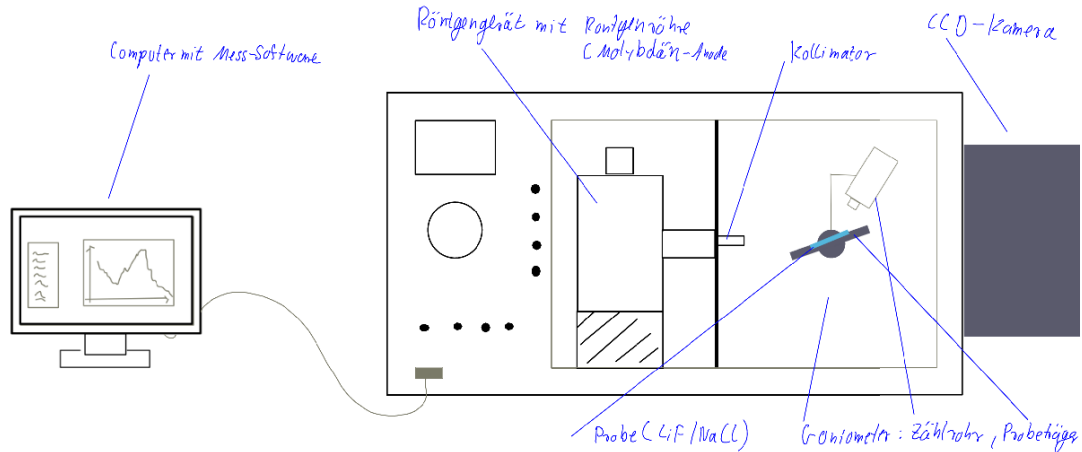
Contents

1	Einleitung	2
1.1	Ziel des Versuchs	2
1.2	Grundlagen	2
2	Durchführung des Versuchs	5
2.1	Sicherheitsüberprüfung	5
2.2	Messung des Röntgenspektrums am LiF-Kristall	5
2.3	Messung des Röntgenspektrums am NaCl-Kristall	5
3	Messprotokoll	6
4	Auswertung	7
4.1	Messung des Röntgenspektrums am LiF-Kristall	7
4.1.1	Berechnung von h aus Grenzwellenlänge	7
4.1.2	Bestimmung der K_α und K_β Linien	10
4.1.3	Berechnung von h durch Variation der Einsatzspannung	11
4.2	Messung des Röntgenspektrums	12
4.2.1	Bestimmung der Gitterkonstante von NaCl	13
4.2.2	Bestimmung der Avogadrozahl N_a	13
5	Zusammenfassung und Diskussion	14
5.1	Messung des Röntgenspektrums am LiF-Kristall	14
5.1.1	Berechnung von h aus Grenzwellenlänge	14
5.1.2	Bestimmung der K_α und K_β Linien	14
5.1.3	Berechnung von h durch Variation der Einsatzspannung	15
5.2	Messung des Röntgenspektrums am NaCl-Kristall	15
5.2.1	Bestimmung der Gitterkonstante von NaC	16
5.2.2	Bestimmung der Avogadrozahl N_a	16
6	Literaturverzeichnis	16

1 Einleitung

1.1 Ziel des Versuchs

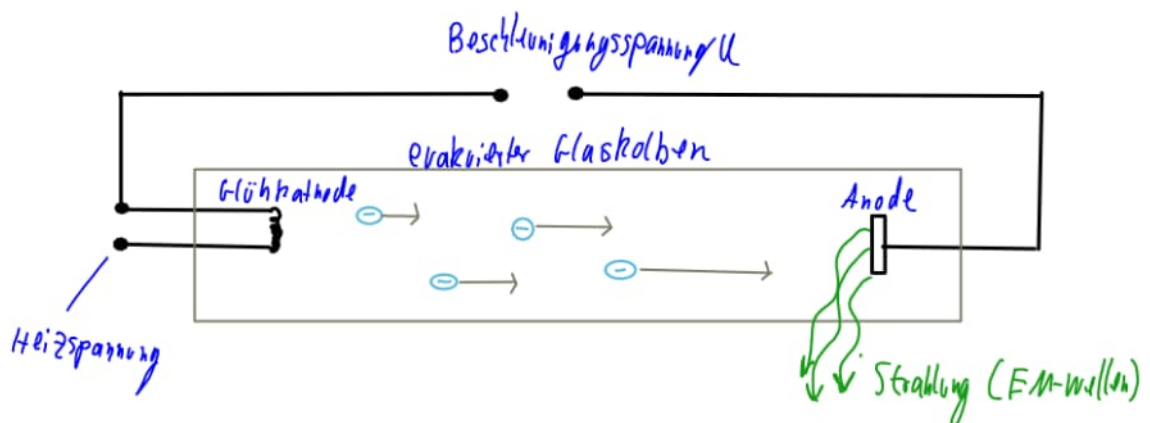
In diesem Versuch wollen wir durch Aufnahme des Röntgenspektrums einer Molybdän-Anode einmal bei fester Spannung und variablen Winkel und einmal bei festem Winkel und variabler Spannung die plancksche Konstante h bestimmen. Weiterhin wollen wir aus allen Röntgenspektren die Wellenlängen der K_α - und K_β -Linie des Anodenmaterials messen. Schließlich wollen wir die Gitterkonstante von Natriumchlorid bestimmen, um daraus die Avogadrozahl N_A zu ermitteln.



Skizze 1: Der Versuchsaufbau

1.2 Grundlagen

Eine Röntgenröhre besteht aus einem evakuierten Glaskolben mit zwei Elektroden, zwischen denen eine Beschleunigungsspannung U anliegt. An der Kathode werden durch Erhitzen (Heizspannung an Kathode) freie Elektronen erzeugt, die durch U beschleunigt werden und dann auf die Anode prallen. Dort werden sie abgebremst. Die dabei abgegebene Energie wird als elektromagnetische Strahlung emittiert. Da der Energieverlust beim Bremsvorgang unterschiedlich sein kann (verschieden starke Bremsung an einzelnen Atomen) ist dieses sogenannte Bremsspektrum kontinuierlich.



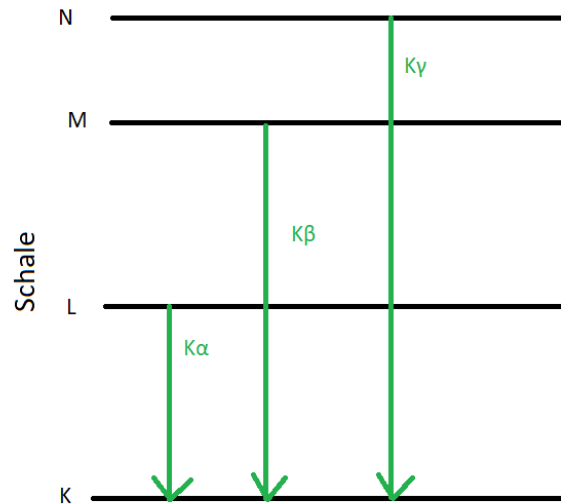
Skizze 2: Aufbau der Röntgenröhre

Das Bremsspektrum entsteht auf der kurzwelligen Seite erst ab einer Grenzwellenlänge λ , bei welcher die gesamte Energie des Elektrons in ein Photon übergeht.

$$E = eU = h\lambda_{gr} \Rightarrow \lambda_{gr} = \frac{hc}{eu} \quad (1)$$

(Lichtgeschwindigkeit = c und Elementarladung = e)

Ist die Beschleunigungsspannung groß genug, können die Elektronen die Anodenatome auch ionisieren. Es werden Hüllenelektronen aus tiefen Schalen herausgeschlagen, beim Nachrücken äußerer Elektronen wird Energie in Form von Photonen frei. Da dieses Spektrum vom Anodenmaterial abhängt, heißt es charakteristisches Spektrum. Je nachdem zwischen welchen Schalen der Übergang stattfindet, werden bestimmte Wellenlängen ausgestrahlt. Dieses Spektrum ist also diskret (Linienspektrum). Bei Übergängen auf die innerste Schale (K-Schale) spricht man von K-Übergängen.



Skizze 3: Übergänge in Molybdän

Die Energie der Übergänge von n nach m lässt sich mit dem Moseleyschen Gesetz abschätzen:

$$E_{n \rightarrow m} = hcR_{\text{inf}}(z - A)^2 \left(\frac{1}{m^2} - \frac{1}{n^2} \right) \quad (2)$$

(Rydberg-konstante = R_{inf} , Kernladungszahl z , Hauptquantenzahl n und m , Abschirmungskonstante A).

Für den Alphaübergang ist $A \approx 1$ ungefähr gleich eins. Damit erhält man ($n = 2$, $n = 1$) $E_\alpha = 17,2 \text{ keV}$. Der Literaturwert beträgt $E_{\alpha L} = 17,4 \text{ keV}$. Weitere Effekte der Feinstruktur (Drehimpuls und Spin) werden wir in diesem Versuch nicht betrachten.

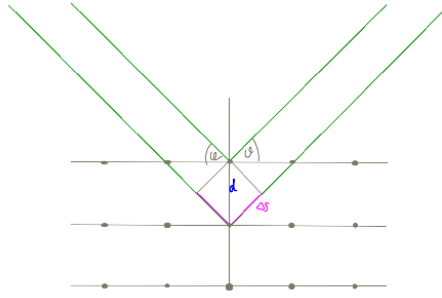
Bragg-reflexion:

Um das Spektrum der Röntgenröhre aufzunehmen, nutzen wir die Reflexion der Röntgenstrahlung an Kristallen aus, da die Kristallstrukturen im gleichen Größenbereich wie die Röntgenwellenlänge liegen.

Eintreffende Strahlung wird an der Oberfläche gemäß dem Reflexionsgesetz reflektiert. Die Strahlung dringt jedoch auch tiefer ein und wird an weiteren Netzebenen reflektiert. Die Intensität der ausfallenden Strahlung hängt dann vom Gangunterschied Δs ab. Ist dieser für zwei benachbarte Ebenen gerade ein Vielfaches der Wellenlänge λ , so ist die Intensität maximal:

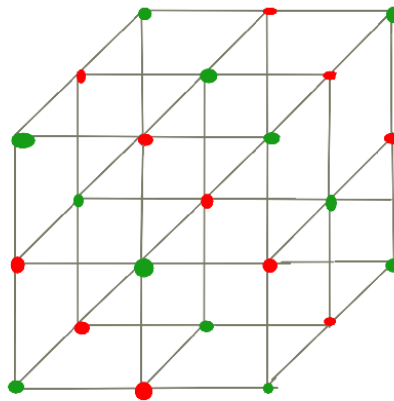
$$\Delta s = 2d \sin(\nu), n \in \mathbb{N} \quad (3)$$

Diesen Zusammenhängen nennen wir das Braggsche Gesetz.



Skizze 4: Bragg-Reflexion am Kristall

Erfüllt θ diese Bedingungen nicht, so kommt es zu destruktiver Interferenz und man wird weniger bis nichts detektieren. Nach dieser Methode kann man nun über die Variation des Einfallswinkels leicht ein Röntgenspektrum aufnehmen. Bei der Drehkristallmethode wird der Kristall gedreht, um den Einfallswinkel zu variieren. Die reflektierte Röntgenstrahlung besitzt eine bestimmte Wellenlänge, deren Intensität gemessen wird. Kristalle sind streng periodisch aufgebaut, wobei die kleinste sich wiederholende Einheit als Elementarzelle bezeichnet wird. NaCl und LiF besitzen kubische Elementarzellen mit konstanter Kantenlänge a . Der Netzebenenabstand d entspricht somit $\frac{a}{2}$.



Skizze 5: Die Elementarzelle

Kennt man das Volumen einer Elementarzelle und die Zeichenanzahl pro Zelle, so kann man die Avogadrozahl N_A bestimmen. Hierzu wird die Röntgenbeugung genutzt.

Beachtet man, dass die Teilchen an den Ecken in 8 Zellen, die Teilchen an den Kanten in 4 Zellen, die Teilchen an der Stirnfläche in zwei Zellen und das zentrale Zeichen in einer Zelle liegt, erhält man für die gesamte Atomzahl pro Elementarzelle für NaCl und LiF jeweils 4.

$$N_A = 4 \frac{V_{mol}}{V} = 4 \frac{V_{mol}}{(2d)^3} = 4 \frac{M_{mol}}{\varphi(d)^3} = \frac{1}{2} \frac{M_{mol}}{\varphi d^3} \quad (4)$$

Das Röntgengerät besitzt einen schwenkbaren Probenträger und Zählrohr. Zählrohr und Probenträger können hierbei in einem 2:1 Verhältnis fest gekoppelt werden. Das Gerät kann weiterhin über einen Computer Messreihen über Winkelabschnitte automatisch erstellen.

2 Durchführung des Versuchs

2.1 Sicherheitsüberprüfung

Zuerst überprüfen wir die Sicherungssysteme, indem wir am Gerät $I = 1mA$ und $U = 5kV$ wählen, und das Bleiglasfenster schließen. Wir schalten die Spannung ein und prüfen, ob die Kontrollleuchte blinkt und die Kathode aufleuchtet. Wenn wir den Verriegelungsschalter der Scheibe nach unten drücken, sollte die Vorrichtung abschalten.

2.2 Messung des Röntgenspektrums am LiF-Kristall

Wir montieren den LiF-Kristall, wählen $U = 53kV$, $I = 1mA$, $\Delta t = 5s$, $\Delta\beta = 0,2$ und Scannen im Bereich 3 bis 22.

Die Messung wird mit der Computer Software durchgeführt. Das generierte Messprotokoll wird sowohl in graphischer Auswertung, als auch als Textdatei gespeichert. Aus der vorherigen Messung wird durch grobes ablesen die Lage der Alpha und Beta ermittelt. In diesen Bereichen wird die Messung aus a) mit $\Delta\beta = 0.1$ und $t = 20s$ wiederholt.

Wir stellen einen festen Winkel von $= 7,5$ und messen die Zählrate für verschiedene Spannungen von 20 bis 35 kV in ein Kilovoltschritten über 20 Sekunden. Aus technischen Gründen müssen wir hierfür einen Scan zwischen $7,5^\circ$ und $7,6^\circ$ einstellen, welchen wir nach Erreichen der $7,6^\circ$ abbrechen und über die Replay-Funktion den ermittelten Wert bei $7,5^\circ$ ablesen und notieren.

2.3 Messung des Röntgenspektrums am NaCl-Kristall

Analog zur Messung 1a wird der NaCl-Kristall überall Winkelbereich von 3° bis 18° vermessen.

17.11.21

Darin Korte
Marius Mangold

Messprotokoll 255

Charakteristisches Spektrum:

1. Ordnung:

K_{α} : (8,4 - 9,0)

K_{β} : (9,6 - 10,2)

2. Ordnung:

K_{α} : (17,6 - 18,4)

K_{β} : (20 - 21)

Messung der Counts abhängig von der Spannung

kv	t	rate
20	20	1.40
21	20	1.60
22	20	4.15
23	20	30.65
24	20	86.65
25	20	106.90
26	20	151.20
27	20	191.70
28	20	242.60
29	20	277.80
30	20	318.20
31	20	354.00
32	20	396.80
33	20	430.90
34	20	462.90
35	20	494.20

17.11.2021

[Signature]

4 Auswertung

4.1 Messung des Röntgenspektrums am LiF-Kristall

4.1.1 Berechnung von h aus Grenzwellenlänge

Wir maßen die Zählrate in Abhängigkeit vom Winkel bei $U = (35,0 \pm 0,1)kV$ in einem Winkelbereich von 3° bis 22° . Der Fehler der Zählrate entspricht der Wurzel des Wertes. In Diagramm 1 stellen wir unsere Messwerte graphisch dar.

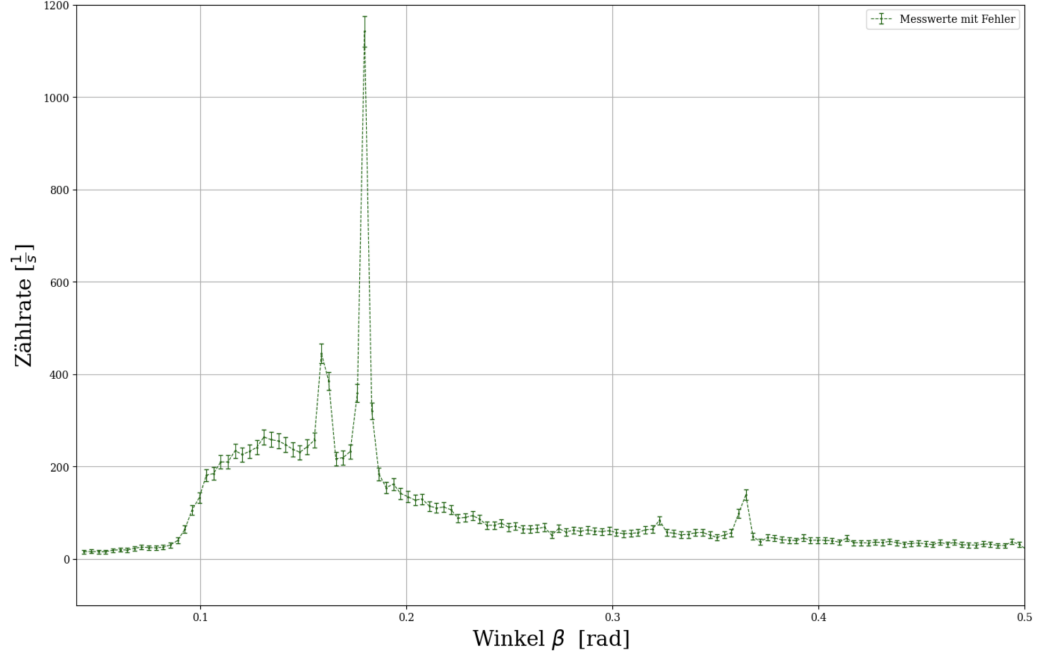


Diagramm 1: Spektrum vom LiF Kristall mit Bragg Reflexion

Nun betrachten wir den Bereich, in dem das Bremsspektrum einsetzt. In Diagramm zwei legen wir mit Python eine Ausgleichsgerade durch den Untergrund (die ersten 13 Werte) und eine Ausgleichsgerade durch den annähernd linearen Anstieg (die Werte 14 bis 18). Die Geraden sind also von der Form $y = ax + b$ wobei wir die Faktoren der Untergrundfunktion mit a_{ug} bzw. b_{ug} und die der Anstiegsfunktion entsprechend mit a_{anst} bzw. b_{anst} bezeichnen. Nach der fit-Funktion aus Python ergibt sich der Schnittpunkt der beiden Geraden und somit der Grenzwinkel:

$$v_{gr} = \frac{b_{ug} - b_{anst}}{a_{anst} - a_{ug}} \quad (5)$$

mit dem Fehler:

$$\Delta v_{gr} = \sqrt{\left(\frac{\Delta b_{ug}}{a_{anst} - a_{ug}}\right)^2 + \left(\frac{\Delta b_{anst}}{a_{anst} - a_{ug}}\right)^2 + \left(\frac{\Delta a_{ug}(b_{ug} - b_{anst})}{a_{anst} - a_{ug}}\right)^2 + \left(\frac{\Delta a_{anst}(b_{ug} - b_{anst})}{a_{anst} - a_{ug}}\right)^2} \quad (6)$$

$$v_{gr} = (0,088 \pm 0,008)rad \quad (7)$$

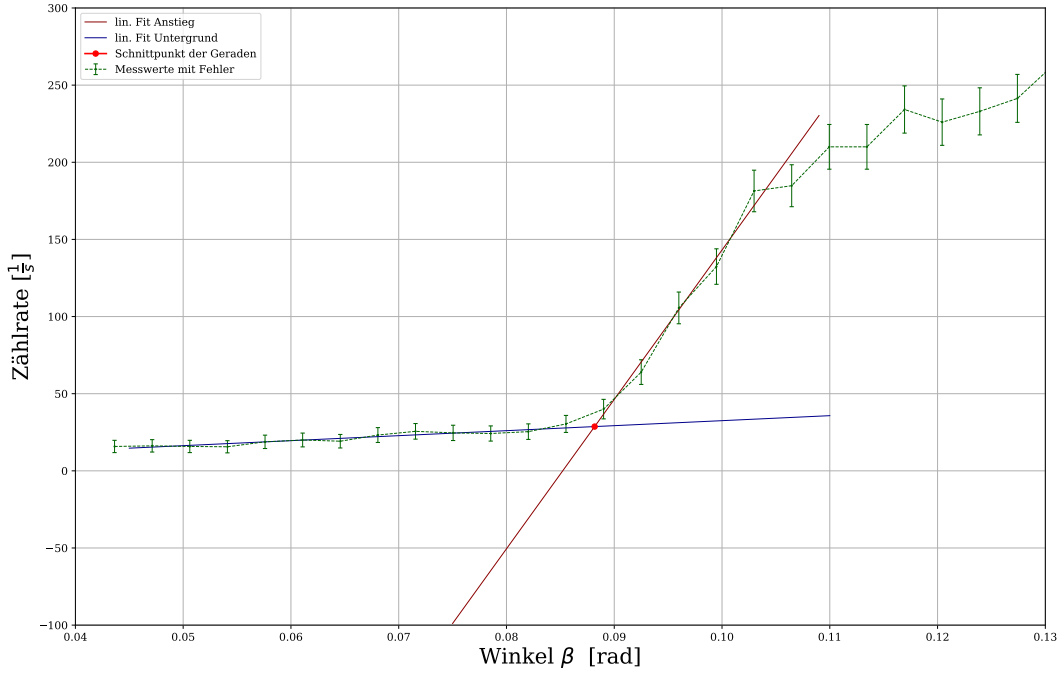


Diagramm 2: Bestimmung der Grenzwellenlänge

Nach dem Bragg-gesetz (4) erhalten wir somit unter der Annahme, dass es sich um das erste Maximum handelt ($\Delta s = n\lambda = \lambda$) eine Grenzwellenlänge von:

$$\lambda_{gr} = 2d \sin(v_{gr}) \quad (8)$$

$$\lambda_{gr} = 2 \sqrt[3]{\frac{\rho}{M}} \sin(v_{gr}) \quad (9)$$

Wobei wir den Gitterabstand d aus dem Verhältnis von Dichte ρ und Molekularmasse M abschätzen können. Da wir diese gegebenen Größen als gegeben und exakt annehmen erhalten wir über die einfache Gaußsche Fehlerfortpflanzung den Fehler

$$\Delta \lambda_{gr} = 2 \sqrt[3]{\frac{\rho}{M}} \cos(v_{gr}) \Delta v_{gr} \quad (10)$$

Es ergibt sich schlussendlich:

$$\lambda = (3,5 \pm 0,3) 10^{-11} m \quad (11)$$

Nach Gleichung (1) gilt:

$$\lambda_{gr} = \frac{hc}{U_e} \leftrightarrow h = \frac{\lambda_{gr} U_e}{c} \quad (12)$$

Damit ist der Fehler:

$$\Delta h = \frac{e}{c} \sqrt{(\Delta \lambda_{gr} U)^2 + (\Delta U \lambda_{gr})^2} \quad (13)$$

Und unser berechnetes Wirkungsquantum:

$$h = (6,6 \pm 0,6) 10^{-34} Js \quad (14)$$

Nach dem Bragg'schen Gesetz fängt das Spektrum 2.Ordnung (also $n = 2$) bei v_2 an, für welches gilt:

$$2d\sin(v_2) = \frac{\lambda_{gr}}{d} \Rightarrow v_2 = \arcsin\left(\frac{\lambda_{gr}}{d}\right) \quad (15)$$

Für den Fehler gilt nach Fehlerfortpflanzung dann:

$$\Delta v_2 = \sqrt{\left(\frac{1}{\sqrt{1 - \frac{\lambda_{gr}^2}{d^2}}} * \frac{\Delta \lambda_{gr}}{d}\right)^2} \quad (16)$$

Damit ist unser Ergebnis:

$$v_2 = (0,177 \pm 0,018)rad \quad (17)$$

4.1.2 Bestimmung der K_α und K_β Linien

Hier maßen wir die Zählrate um die Peaks aus 4.1.1 in kleineren Winkelschritten (von $0,1^\circ$). In Diagramm drei bis sechs stellen wir diese genaueren Messungen zu den α und β Linien 1. und 2. Ordnung graphisch dar, wobei der Fehler der Zerfallsrate wieder die Wurzel ist. Mithilfe von Python legen wir nun durch die Peaks jeweils eine Gaußfunktion als fit-Funktion und erhalten als Parameter jeweils Mittelwert und Standardabweichung, also Winkel mit Fehler der Linien (eine Standardabweichung nehmen wir also als Fehler). Die Indizes 1 und 2 bezeichnen die Ordnung und Indizes α und β bezeichnen die Linien.

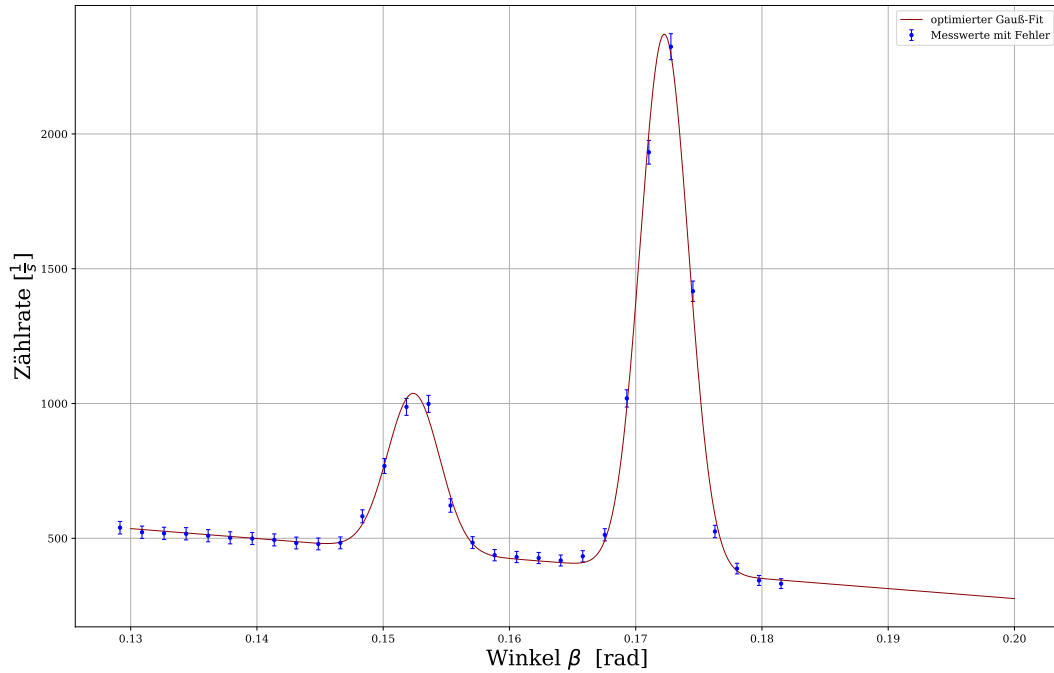


Diagramm 3: Fit der K_α und K_β Linien erster Ordnung

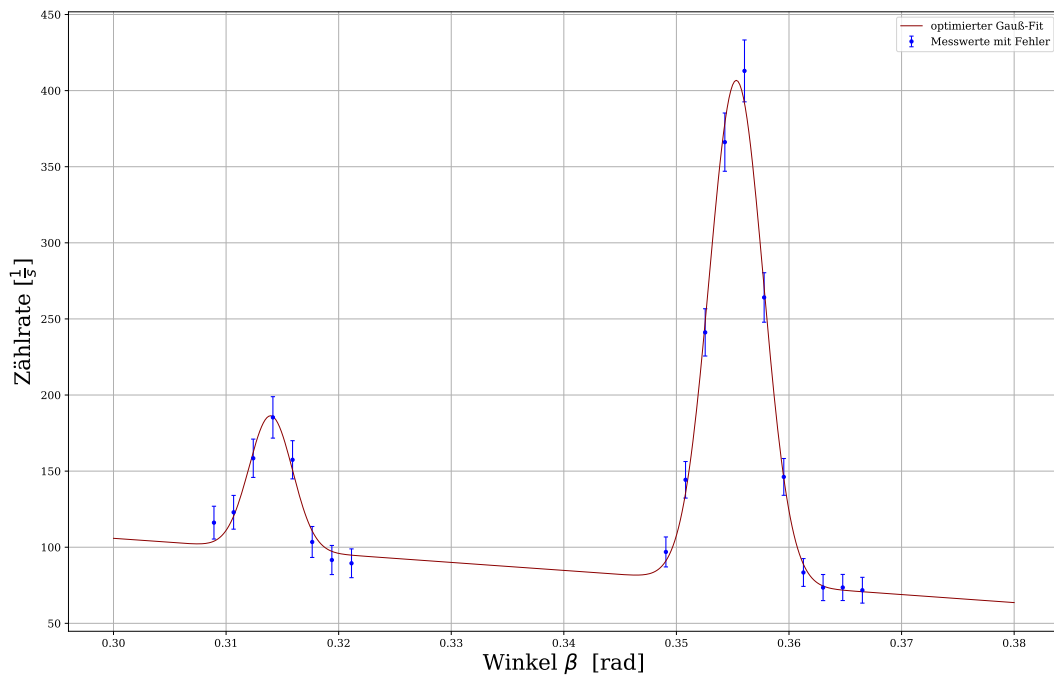


Diagramm 4: Fit der K_α und K_β Linien zweiter Ordnung

$$v_{\alpha 1} = (0,1245 \pm 0,0020)rad \quad (18)$$

$$v_{\beta 1} = (0,1104 \pm 0,0018)rad \quad (19)$$

$$v_{\alpha 2} = (0,2531 \pm 0,0020)rad \quad (20)$$

$$v_{\beta 2} = (0,2238 \pm 0,0024)rad \quad (21)$$

Hiermit ergeben sich die zugehörigen Wellenlängen mit Fehler erneut aus dem Bragg'schen Gesetz, wobei wir die α und β Werte mitteln und den Fehler nach $\lambda = \frac{1}{2}\sqrt{(\Delta\lambda_1)^2 + (\Delta\lambda_2)^2}$ auswerten.

$$\lambda_{\alpha 1} = (69,0 \pm 0,7)pm \quad (22)$$

$$\lambda_{\beta 1} = (61,2 \pm 0,8)pm \quad (23)$$

$$\lambda_{\alpha 2} = (70,1 \pm 0,5)pm \quad (24)$$

$$\lambda_{\beta 2} = (62,2 \pm 0,4)pm \quad (25)$$

Da die verwendeten Winkel alle sehr klein sind, insbesondere $v_{\alpha 1} < \frac{\pi}{2}$, nähern wir $\sin(v) \approx v$. Damit sind λ und v proportional zueinander und für die Halbwertsbreite $Hb_{\alpha 1}$ von $\lambda_{\alpha 1}$ gilt dann:

$$Hb_{\alpha 1} = 2\sqrt{2\ln(2)}\sigma_{\alpha 1} = 1,9pm \quad (26)$$

(Da für die Halbwertsbreite des Winkels auch $2\sqrt{2\ln(2)}\sigma$ gilt). Da die Halbwertsbreite in gewissem Sinne den Fehler der Wellenlänge darstellt, berechnen wir hier keinen Fehler, da dies der Fehler des Fehlers wäre.

4.1.3 Berechnung von h durch Variation der Einsatzspannung

Hier maßen wir die Zerfallsraten bei $v = 7,5$ für verschiedene Spannungen U (von 20kV bis 35kV). Die Messwerte sind in Tabelle 2 zu sehen. Der Fehler der Zerfallsrate ist wieder die Wurzel der Werte. Wir stellen die Messwerte in Diagramm 5 dar und fitten zwei lineare Funktionen einmal als Untergrund durch die ersten drei und anschließend durch die letzten 14 als Steigungsfunktion.

Der Schnittpunkt der beiden Geraden, der die Einsatzspannung U_E darstellt errechnet sich symmetrisch zu a) aus

$$U_E = \frac{b_{ug} - b_{Anst}}{a_{anst} - a_{ug}} \quad (27)$$

mit dem Fehler:

$$\Delta U_E = \sqrt{\left(\frac{\Delta b_{ug}}{a_{anst} - a_{ug}}\right)^2 + \left(\frac{\Delta b_{anst}}{a_{anst} - a_{ug}}\right)^2 + \left(\frac{\Delta b_{ug}(b_{ug} - b_{anst})}{a_{anst} - a_{ug}}\right)^2 + \left(\frac{\Delta b_{anst}(b_{ug} - b_{anst})}{a_{anst} - a_{ug}}\right)^2} \quad (28)$$

$$U_E = (22,0 \pm 2,7)kV \quad (29)$$

Ein erneuter Einsatz der Gleichung 2 und 3 ergibt

$$h = \frac{eU_g\lambda}{c} \quad (30)$$

$$= \frac{eU_g}{c} 2d\sin(v) \quad (31)$$

$$\Delta h = \frac{2ed}{c} \sqrt{(\Delta U_E \sin(v))^2 + (U_E \Delta v \cos(v))^2} \quad (32)$$

mit $\Delta v = 0,1 = \frac{0,2\pi}{360}$ erhalten wir schließlich

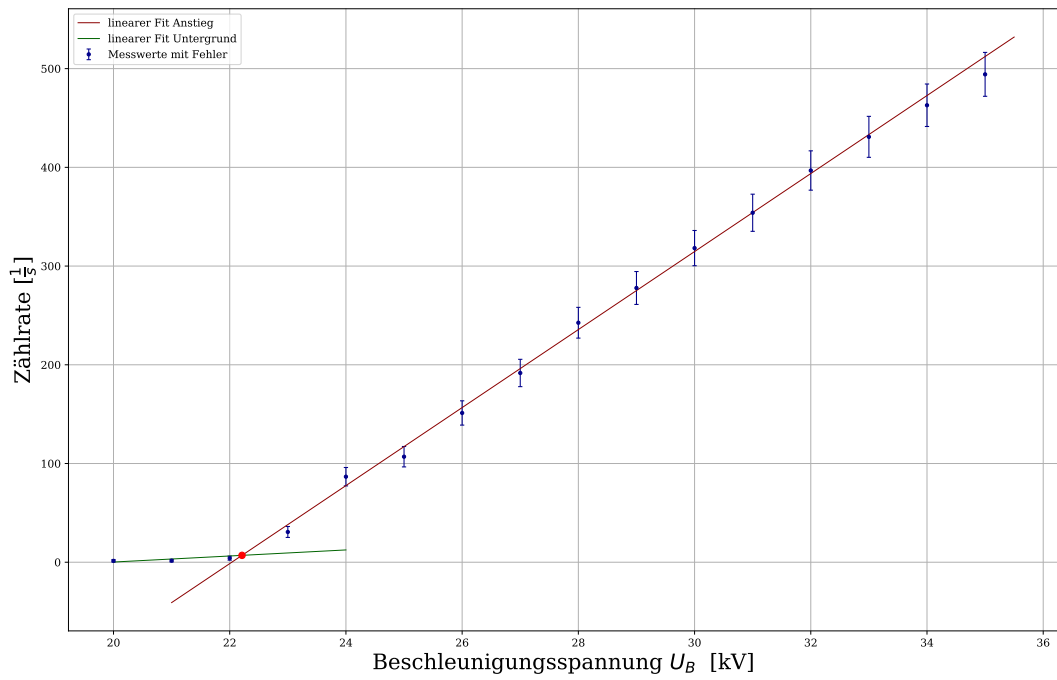


Diagramm 5: Fit mit zwei Linearen Funktionen und Schnittpunkt in rot

4.2 Messung des Röntgenspektrums

In diesem Teil führten wir die Gleiche Messung wie in 4.1.1 durch, wobei wir den NaCl-Kristall verwendeten. Das Spektrum ist im Diagramm 8 dargestellt. Aus diesem Spektrum bestimmen wir die Lagen der Linien und schätzen die Fehler über die Breite der Peaks ab und erhalten:

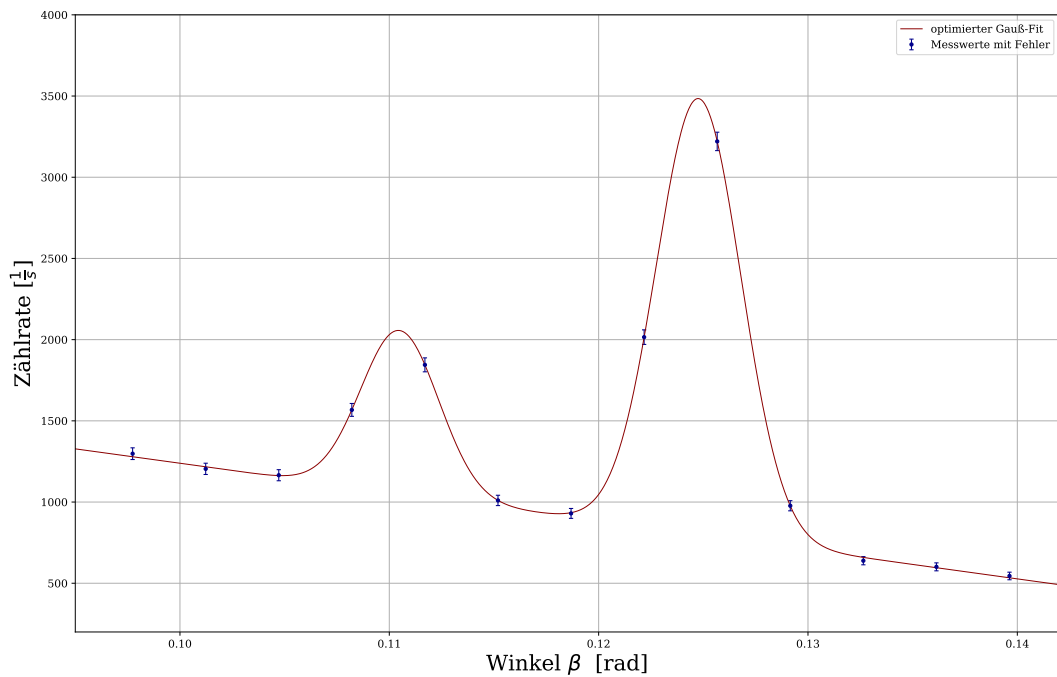


Diagramm 6: Fit der K_α und K_β Linien erster Ordnung bei NaCl

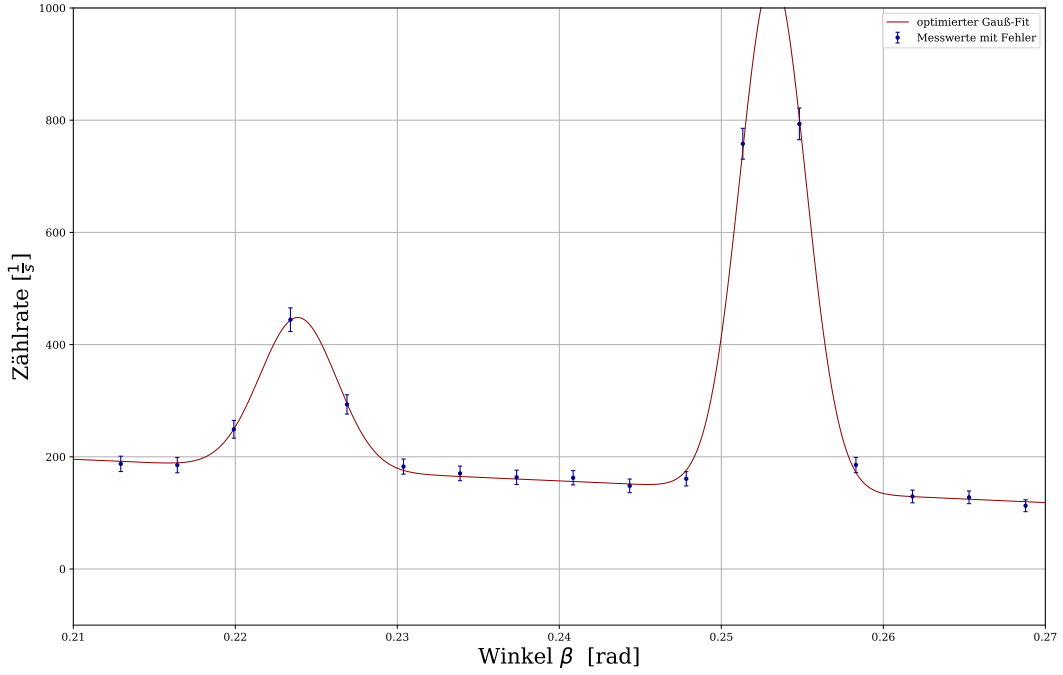


Diagramm 7: Fit der K_α und K_β Linien zweiter Ordnung bei NaCl

$$\lambda_{\alpha 1} = (0,1245 \pm 0,0020)pm \quad (33)$$

$$\lambda_{\beta 1} = (0,1104 \pm 0,0018)pm \quad (34)$$

$$\lambda_{\alpha 2} = (0,2531 \pm 0,0020)pm \quad (35)$$

$$\lambda_{\beta 2} = (0,2238 \pm 0,0024)pm \quad (36)$$

4.2.1 Bestimmung der Gitterkonstante von NaCl

Da beide Messungen das selbe Anodenmaterial verwendet haben, ist zu erwarten, dass sie in der Auswertung ungefähr die gleichen Wellenlängen aufweisen. Es gilt weiterhin:

$$d * \sin(v_{\alpha, LiF}) = n * \lambda_\alpha = g * \sin(v_{\alpha, NaCl}) \quad (37)$$

$$g = \frac{n * \lambda_\alpha}{\sin(v_{\alpha, NaCl})} \quad (38)$$

mit dem Fehler aus der Gaußschen Fehlerfortpflanzung:

$$\Delta g = \sqrt{\left(\frac{n * \Delta \lambda_\alpha}{\sin(v_{\alpha, NaCl})}\right)^2 + \left(\frac{n * \lambda_\alpha * \Delta v_{\alpha, NaCl}}{\tan(v_{\alpha, NaCl})}\right)^2} \quad (39)$$

Wodurch sich ein systematischer Fehler ergibt:

$$g_{m, err} = \frac{1}{4} * \sqrt{g_1^2 + g_2^2 + \dots} \quad (40)$$

Es folgt als Ergebnis

$$g = (558,33 \pm 4,3 \pm 0,99)pm = (558,33 \pm 5,3)pm \quad (41)$$

4.2.2 Bestimmung der Avogadrozahl N_A

Zuletzt möchten wir die Avogadrozahl N_A bestimmen. Dies können wir über die Dichte und die Molekularmasse über den einfachen Zusammenhang

$$N_A = 4 * \frac{M}{\rho * g_m^3} \quad (42)$$

und den resultierenden Fehler

$$\Delta N_A = 4 * 3 * \frac{g_{mean,err}}{\rho * g_m^4} + 4 * 3 * \frac{g_{mean,std}}{\rho * g_m^4} \quad (43)$$

Wir erhalten:

$$N_A = (6,21 \pm 0,14 \pm 0,03)10^{23} \quad (44)$$

5 Zusammenfassung und Diskussion

5.1 Messung des Röntgenspektrums am LiF-Kristall

Die Sicherheitsüberprüfung verlief nach Plan, und stellte sicher, dass das Sicherungssystem intakt ist.

5.1.1 Berechnung von h aus Grenzwellenlänge

Die erste zu bestimmende Größe war das Wirkungsquantum h , welches wir aus der Grenzwellenlänge ermitteln wollten. Eine erste grafische Analyse der Spektren entspricht unseren Voraussetzungen, der relativ scharfen Kante bei kleinem Winkel und dem asymptotischen Verhalten bei größeren Winkel, als auch den verschiedenen Peaks. Über die lineare Annäherung und den Achsenabschnitt erhielten wir einen Grenzwinkel von:

$$v_{gr} = (0,088 \pm 0,008)rad \quad (45)$$

und hiermit weiterhin das Wirkungsquantum von:

$$h = (6,6 \pm 0,6)10^{-34}Js \quad (46)$$

Da das Wirkungsquantum eine Definitionsgröße ist, können wir die Sigtabweichung sehr einfach durch die Differenz zum Literaturwert dividiert durch unseren Fehler berechnen und kommen auf eine Abweichung von:

$$\sigma = \frac{h_{literatur} - h}{\Delta h} \quad (47)$$

$$\sigma_{h_1} = 0.05 \quad (48)$$

Dieses sehr gute Ergebnis bestätigt uns in der Auswertung als auch in der Durchführung des Versuches.

Weiterhin werden wir nach dem praktischen Gesetz den Beginn des Spektrums 2. Ordnung auf, und als Winkel:

$$v_2 = (0,177 \pm 0,018)rad \quad (49)$$

5.1.2 Bestimmung der K_α und K_β Linien

Eine nähere Betrachtung der Peaks aus 4.1.1 ermöglichte es uns, die Alpha und Beta Linie erster und zweiter Ordnung zu betrachten. Indem wir in Python für jeden Peak eine Gaußfunktion fitteten, konnten wir über dessen Maximum den jeweiligen Winkel, und somit die Wellenlängen bestimmen.

Wir fanden folgende Wellenlängen vor:

$$\lambda_{\alpha 1} = (69, 0 \pm 0, 7) pm \quad (50)$$

$$\lambda_{\beta 1} = (61, 2 \pm 0, 8) pm \quad (51)$$

$$\lambda_{\alpha 2} = (70, 1 \pm 0, 5) pm \quad (52)$$

$$\lambda_{\beta 2} = (62, 2 \pm 0, 4) pm \quad (53)$$

Aus dem Praktikumsskript stehen uns die Literaturwerte für die K-Linien von Molybdän zur Verfügung, wobei der Fehler dieser Werte als halbe Einheit der letzten Ziffer angegeben ist. Wir erhalten somit Abweichungen von:

$$\sigma_{\beta 2} = \frac{\lambda_{\beta 2} - \lambda_{\beta 2 Lit}}{\Delta \lambda_{\beta 2} + \Delta \lambda_{\beta 2 Lit}} = 2$$

Wir finden also für beide Linien eine Abweichung im 2-sigma Bereich vor, welche zufriedenstellend ist. Auffällig ist, dass beide Linien um etwa 1pm zu klein vermessen wurden, was Hinweis auf einen systematischen Fehler, jedoch auch reiner Zufall sein könnte. Weiterhin berechneten wir die Halbwertsbreite von $\lambda_{\alpha 1}$ und erhielten:

$$Hb_{\alpha 1} = 1, 9 pm \quad (54)$$

5.1.3 Berechnung von h durch Variation der Einsatzspannung

Unsere Aufgabe bestand nun erneut darin, das Wirkungsquantum h zu bestimmen, wobei dies durch die Variation der Einsatzspannung vollzogen wurde. Die aufgenommenen Messwerte stellten wir in Diagramm 5 dar und erhielten über die Steigungsfunktion eine Einsatzspannung von:

$$U_E = (22, 0 \pm 2, 7) kV \quad (55)$$

Hierbei fällt direkt auf, dass wir einen recht großen Fehler vorliegen haben, welcher aufgrund zu weniger Messwerte entstanden ist. Eine genauere Vermessung der linearen Umgebung würde uns ein genaueres Ergebnis liefern.

Für das plancksche Wirkungsquantum erhielten wir schließlich:

$$h = (6, 2 \pm 0, 7) 10^{-34} Js \quad (56)$$

Was symmetrisch zum ersten Aufgabenteil zu einer Sigmaumgebung von 0.61σ führte. Diese ist um ein Vielfaches größer als bei der ersten Methode, jedoch trotzdem noch sehr gut. Die größeren Abweichungen sind wiederum auf die von angesprochene Ungenauigkeit der Steigungsfunktion abzuleiten.

5.2 Messung des Röntgenspektrums am NaCl-Kristall

Der zweite Teil des Versuchs beschäftigte sich mit der Aufnahme des Röntgenspektrums. Ähnlich zu Aufgabe 4.1.1 vermaßen wir jetzt einen Natriumchloridkristall. Wir fütterten wieder Gaußkurven an die Peaks und hielten:

$$\lambda_{\alpha 1} = (0, 1245 \pm 0, 0020) pm \quad (57)$$

$$\lambda_{\beta 1} = (0, 1104 \pm 0, 0018) pm \quad (58)$$

$$\lambda_{\alpha 2} = (0, 2531 \pm 0, 0020) pm \quad (59)$$

$$\lambda_{\beta 2} = (0, 2238 \pm 0, 0024) pm \quad (60)$$

Für diese Werte waren leider keine Literaturangaben getätigt worden, deswegen will sie nicht vergleichen können. Auch eine Internetrecherche führte nicht zu verlässlichen Daten. Die Größenordnung der Messungen liegt jedoch im sinnvollen Bereich, auffällig ist der besonders kleine berechnete Fehler.

5.2.1 Bestimmung der Gitterkonstante von NaCl

Über die bereits getätigten Aufnahmen ist es uns möglich, die Gitterkonstante von Natriumchlorid zu bestimmen. Wir erhielten als Ergebnis:

$$g = (558,33 \pm 4,3 \pm 0,99)pm = (558 \pm 6)pm \quad (61)$$

Hierfür finden wir einen Literaturwert ¹ von $g = 564pm$ und somit eine sehr gute Abweichung von nur einem σ

5.2.2 Bestimmung der Avogadrozahl N_A

Zuletzt bestimmten wir die Avogadrozahl zu:

$$N_A = (6,21 \pm 0,14 \pm 0,03 = 6,21 \pm 0,17) \frac{10^{23}}{mol} \quad (62)$$

Für diese finden wir wieder die Literaturgrößen in der Praktikumsanleitung, und bestimmen den Literaturwert mit $N_{A,lit} = 6,0221 \frac{10^{23}}{mol}$ auf rund 1.1σ .

Wir sehen also, dass auch dieses Ergebnis sehr gut mit der Theorie übereinstimmt. Generell fanden wir in diesem Versuch sehr geringer Abweichungen vor, was uns in der Durchführung des Versuchs sowie in der Auswertung sehr bestätigt. Dies steht auch auf dem Fundament eines als reibungslos empfundenen Versuchs vor Ort.

Dies und die recht komplizierten Details des Versuches führen dazu, dass ich keine weiteren Verbesserungsvorschläge aufführen kann. Sicherlich wäre dies grundsätzlich möglich, doch fehlt es an Einsicht in das komplizierte Messgerät, um dies nachhaltig zu beurteilen.

6 Literaturverzeichnis

¹Holleman, Wiberg: Lehrbuch der anorganischen Chemie. 100. Auflage, Walter de Gruyter, Berlin, New York 1985

Versuch 255: Röntgenspektrometer

Auswertung der Messergebnisse

```
In [2]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.optimize import curve_fit
from scipy.stats import chi2

plt.style.use('default')
plt.rcParams["font.family"]='serif'
plt.rcParams["figure.figsize"][0] = 14
plt.rcParams["figure.figsize"][1] = 9
plt.rcParams['errorbar.capsize']=2
```

```
In [3]: def linear(x,a,b):
        return a*x+b

def gaussian(x, A, mu, sig, y0):
    return y0 + (A*np.exp(-(x-mu)**2/(2*sig**2)))

#Berücksichtigung des Untergrunds
def gauß_optimiert(x, a, c, A1, mu1, sig1, A2, mu2, sig2):
    return linear(x, a, c)+gaussian(x, A1, mu1, sig1, 0)+gaussian(x, A2, mu2, sig2,
```

Teil 1: Messung des Röntgenspektrums mit einem LiF-Kristall

a) Bestimmung λ_{gr} und h

```
In [4]: beta1, counts1=np.loadtxt('a1_korte_data-1.txt',skiprows=0, usecols=(0,1), unpack=True)
beta1=beta1*(2*np.pi/360)
counts1_err=np.sqrt(counts1)/np.sqrt(5)
```

```
In [5]: popt1,pcov1=curve_fit(linear,beta1[13:18],counts1[13:18],sigma=counts1_err[13:18])
popt2,pcov2=curve_fit(linear,beta1[:13],counts1[:13],sigma=counts1_err[:13])

#Schnittpunkt
beta0=(popt2[1]-popt1[1])/(popt1[0]-popt2[0])
d_beta0=np.sqrt(pcov2[1,1]/(popt1[0]-popt2[0])**2+pcov1[1,1]/(popt1[0]-popt2[0])**2+

print('Schnittpunkt: beta0=',beta0,'+- ',d_beta0)
print(popt1, popt2)
```

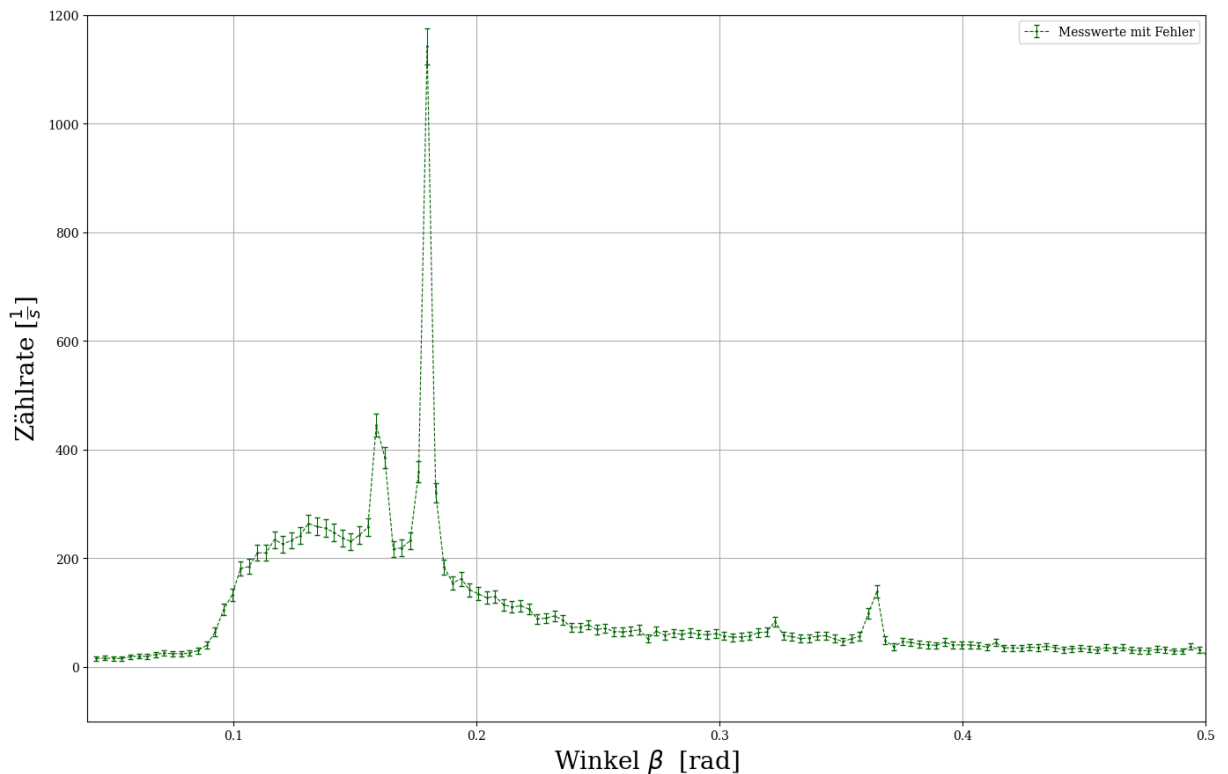
Schnittpunkt: beta0= 0.08818041157480375 +- 0.008906894824853394
[9683.41394279 -825.19925216] [3.24043900e+02 1.13850309e-01]

```
In [13]: plt.errorbar(beta1, counts1, yerr=counts1_err,linewidth=0.8,linestyle='--',marker='.'.
#plt.plot(np.linspace(0.075,0.109,2),linear(np.linspace(0.075,0.109,2),*popt1),linew
#plt.plot(np.linspace(0.045,0.11,2),linear(np.linspace(0.045,0.11,2),*popt2),linewid
plt.xlabel(r'Winkel $ \beta $ [rad]', size=20)
plt.ylabel(r'Zählrate [$\frac{1}{s}$]', size=20)
#plt.title(r'Diagramm 1: Grenzwellenlänge der Röntgenstrahlung', size=22)
plt.xlim(0.04,0.5)
plt.ylim(-100,1200)
plt.grid()
```

```
#plt.plot(beta0, linear(beta0,*popt1),color = 'red',marker='o',markersize=5,label='S')

plt.legend(frameon=True)

plt.tight_layout()
plt.savefig('V255Diagramm5.pdf', format='PDF')
```



```
In [14]: #Güte des Fits, Anstieg
chi2_1=np.sum((linear(beta1[10:15],*popt1)-counts1[10:15])**2/counts1_err[10:15]**2)
dof1=len(beta1[10:15])-2
chi2_1_red=chi2_1/dof1

print('Die Güte des Fits beträgt: chi^2=',chi2_1)
print('chi2_red=',chi2_1_red)
print()
prob1=round(1-chi2.cdf(chi2_1,dof1),3)*100

print('Die Fitwahrscheinlichkeit beträgt: ',prob1,' %')
```

Die Güte des Fits beträgt: chi^2= 476.702781639901
chi2_red= 158.90092721330032

Die Fitwahrscheinlichkeit beträgt: 0.0 %

```
In [15]: #Güte des Fits, Untergrund
chi2_2=np.sum((linear(beta1[:9],*popt2)-counts1[:9])**2/counts1_err[:9]**2)
dof2=len(beta1[:9])-2

chi2_2_red=chi2_2/dof2

print('Die Güte des Fits beträgt: chi^2=',chi2_2)
print('chi2_red=',chi2_2_red)
print()

#Fitwahrscheinlichkeit
prob2=round(1-chi2.cdf(chi2_2,dof2),3)*100

print('Die Fitwahrscheinlichkeit beträgt: ',prob2,' %')
```

Die Güte des Fits beträgt: chi^2= 0.9220403410398893

chi2_red= 0.1317200487199842

Die Fitwahrscheinlichkeit beträgt: 99.6 %

```
In [16]: #Grenzwellenlänge aus beta0
d_LiF=201.4e-12 #[m]
lambda_gr=2*d_LiF*np.sin(beta0)
dlambda_gr=2*np.sqrt((d_LiF*np.cos(beta0)*d_beta0)**2)

print('Grenzwellenlänge lambda_gr=',lambda_gr,'+- ',dlambda_gr)
```

Grenzwellenlänge lambda_gr= 3.547305623396219e-11 +- 3.573757690391258e-12

```
In [17]: #Plancksches Wirkungsquantum aus Gl. (1)

e=1.60221e-19 #C
c=2.9979e8 #m/s
U0= 35e3 #V

h1=lambda_gr*e*U0/c
dh1=h1*(dlambda_gr/lambda_gr)

print('Plancksches Wirkungsquantum h=',h1,'+- ',dh1)
print()

#Vergleich mit dem Literaturwert
h_lit=6.6261e-34 #Js

diff_h1=np.abs(h_lit-h1)
print('Differenz zu Literaturwert diff_h=',diff_h1,'+- ',dh1)
```

Plancksches Wirkungsquantum h= 6.635428099675038e-34 +- 6.684908129666174e-35

Differenz zu Literaturwert diff_h= 9.328099675037897e-37 +- 6.684908129666174e-35

```
In [19]: #mit dem Bragg'schen Gesetz folgt die Beziehung zwischen theta1 und theta2 zu sin(theta2)
theta2=np.arcsin(2*np.sin(beta0))
dtheta2=2*np.cos(beta0)*(2*np.cos(2*beta0)-1)**(-0.5)*d_beta0
print('Das Spektrum zweiter Ordnung setzt bei theta=',theta2,'+- ',dtheta2)
```

Das Spektrum zweiter Ordnung setzt bei theta= 0.1770559885663019 +- 0.01802639213466684

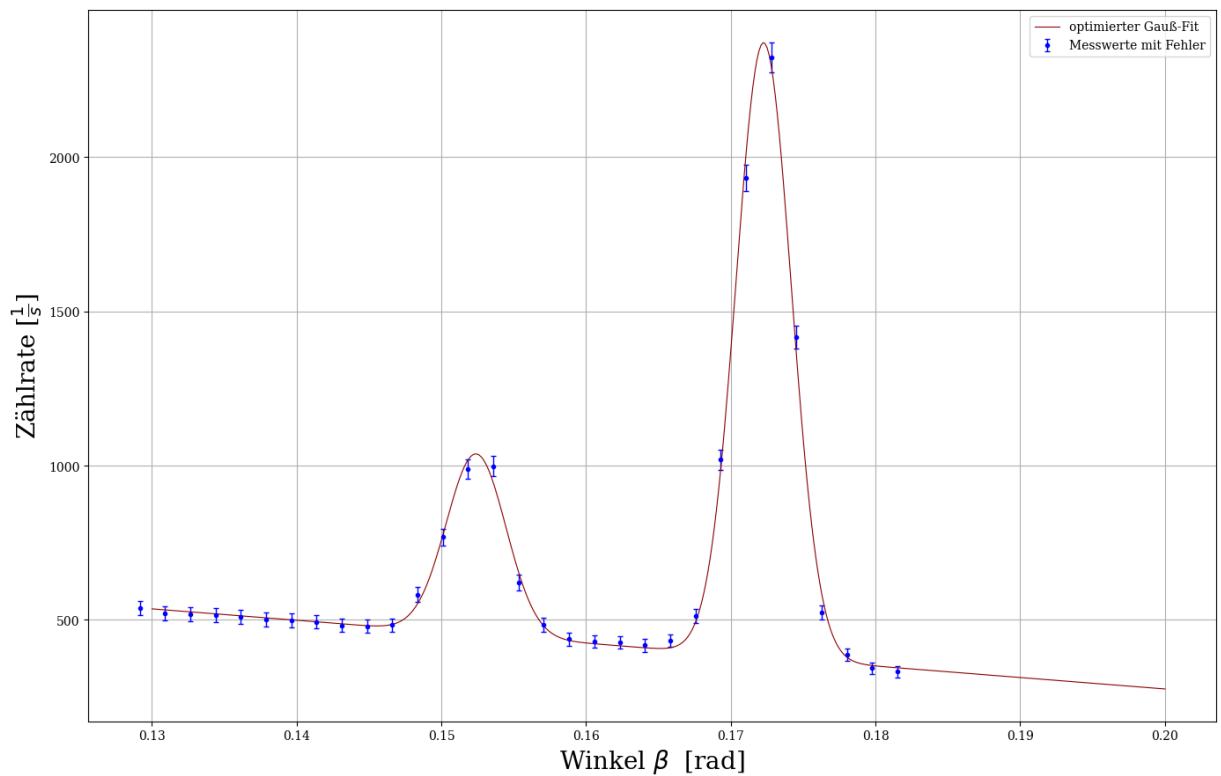
b) Analyse der K_α und K_β Linien

erste Ordnung

```
In [20]: beta2, counts2=np.loadtxt('b1_korte_data-1.txt',skiprows=1, usecols=(0,1), unpack=True)
beta2=beta2*(2*np.pi)/360
#statistischer Fehler
counts2_err=np.sqrt(counts2)
popt3,pcov3=curve_fit(gauß_optimiert,beta2,counts2,sigma=counts2_err,p0=[7000,2000,1])
print(popt3)
```

```
[-3.70864496e+03  1.01794145e+03  5.85090527e+02  1.52402959e-01
 2.06043634e-03  1.99147633e+03  1.72268691e-01  1.91266690e-03]
```

```
In [22]: plt.errorbar(beta2, counts2, yerr=counts2_err,linewidth=1,fmt='b.',linestyle='',label='counts2')
plt.plot(np.linspace(0.13,0.20,500),gauß_optimiert(np.linspace(0.13,0.20,500),*popt3))
plt.xlabel(r'Winkel $ \beta $ [rad]', size=20)
plt.ylabel(r'Zählrate [$\frac{1}{s}$]', size=20)
plt.title(r'Diagramm 6: Analyse der $K_\alpha$- & $K_\beta$-Linie (LiF, erste Ordnung)')
plt.grid()
plt.legend(frameon=True)
plt.tight_layout()
plt.savefig('V255Diagramm6.pdf', format='PDF')
```



```
In [23]: chi2_3=np.sum((gauß_optimiert(beta2,*popt3)-counts2)**2/counts2_err**2)

dof3=len(beta2)-8

chi2_3_red=chi2_3/dof3

print('Die Güte des Fits beträgt: chi^2=',chi2_3)
print('chi2_red=',chi2_3_red)
print()

prob3=round(1-chi2.cdf(chi2_3,dof3),2)*100

print('Fitwahrscheinlichkeit: ',prob3,' %')
```

Die Güte des Fits beträgt: chi^2= 24.79821626904241
chi2_red= 1.0781833160453222

Fitwahrscheinlichkeit: 36.0 %

```
In [24]: #gesuchte Winkel entsprechen Mittelwerten der Gaußkurven
print('Für die erste Ordnung folgen die berechneten Winkel:')
print('K_alpha_1: (',popt3[6], ' +/- ',popt3[7],') rad' )
print('K_beta_1: (',popt3[3], ' +/- ',popt3[4],') rad' )
```

Für die erste Ordnung folgen die berechneten Winkel:
K_alpha_1: (0.17226869066109718 +/- 0.0019126668973202018) rad
K_beta_1: (0.15240295920488126 +/- 0.0020604363372916713) rad

```
In [25]: #Wellenlänge analog wie Grenzwellenlänge
lambdaK_beta1=2*d_LiF*np.sin(popt3[3])
lambdaK_beta1_err=2*np.sqrt((d_LiF*np.cos(popt3[3])*popt3[4])**2)

lambdaK_alpha1=2*d_LiF*np.sin(popt3[6])
lambdaK_alpha1_err=2*np.sqrt((d_LiF*np.cos(popt3[6])*popt3[7])**2)

print('Daraus Wellenlängen')
print('K_alpha_1: (',lambdaK_alpha1*1e12,' +/- ',lambdaK_alpha1_err*1e12,') pm' )
print('K_beta_1: (',lambdaK_beta1*1e12,' +/- ',lambdaK_beta1_err*1e12,') pm' )
```

Daraus Wellenlängen

K_alpha_1: (69.04712960374646 +/- 0.7590187511214741) pm
 K_beta_1: (61.150548413534466 +/- 0.8203239863538175) pm

```
In [26]: #Vergleich mit Literaturwerten
lambdaK_alphalit=71.1e-12 #m
lambdaK_betalit=63.1e-12 #m

diffalpha1=np.abs(lambdaK_alpha1-lambdaK_alphalit)
diffbeta1=np.abs(lambdaK_beta1-lambdaK_betalit)

print('Der Vergleich mit den Literaturwerten liefert:')
print('diff_lambdaK_alpha1 =(',diffalpha1*1e12,' +/- ',lambdaK_alpha1_err*1e12,') pm
print('diff_lambdaK_beta1 =(',diffbeta1*1e12,' +/- ',lambdaK_beta1_err*1e12,') pm =>
```

Der Vergleich mit den Literaturwerten liefert:
 diff_lambdaK_alpha1 =(2.052870396253555 +/- 0.7590187511214741) pm => sigma = 2.7
 diff_lambdaK_beta1 =(1.949451586465526 +/- 0.8203239863538175) pm => sigma = 2.38

Halbwertsbreite für K_α in erster Ordnung

```
In [27]: #Wir wissen: FWHM = 2*sqrt(2*ln(2))*sigma (=2.35482*sigma=2.4*sigma)

FWHM_K_alpha1=2*np.sqrt(2*np.log(2))*popt3[7] #rad
FWHM_K_alpha1_err=2*np.sqrt(2*np.log(2))*pcov3[7,7]**0.5

FWHM2_K_alpha1=2*d_LiF*np.sin(FWHM_K_alpha1) #m
FWHM2_K_alpha1_err=2*np.sqrt((d_LiF*np.cos(FWHM_K_alpha1)*FWHM_K_alpha1_err)**2)

print('Aus dem Fit folgt für die Halbwertsbreite: ')
print('FWHM_K_alpha1 = (',FWHM_K_alpha1*1e3,' +/- ',FWHM_K_alpha1_err*1e3,') 10^-3
print('          = (',FWHM2_K_alpha1*1e12,' +/- ',FWHM2_K_alpha1_err*1e12,') pm'
```

Aus dem Fit folgt für die Halbwertsbreite:
 FWHM_K_alpha1 = (4.503986349276763 +/- 0.08145284040376692) 10⁻³ rad
 = (1.814199567697768 +/- 0.032808871333197265) pm

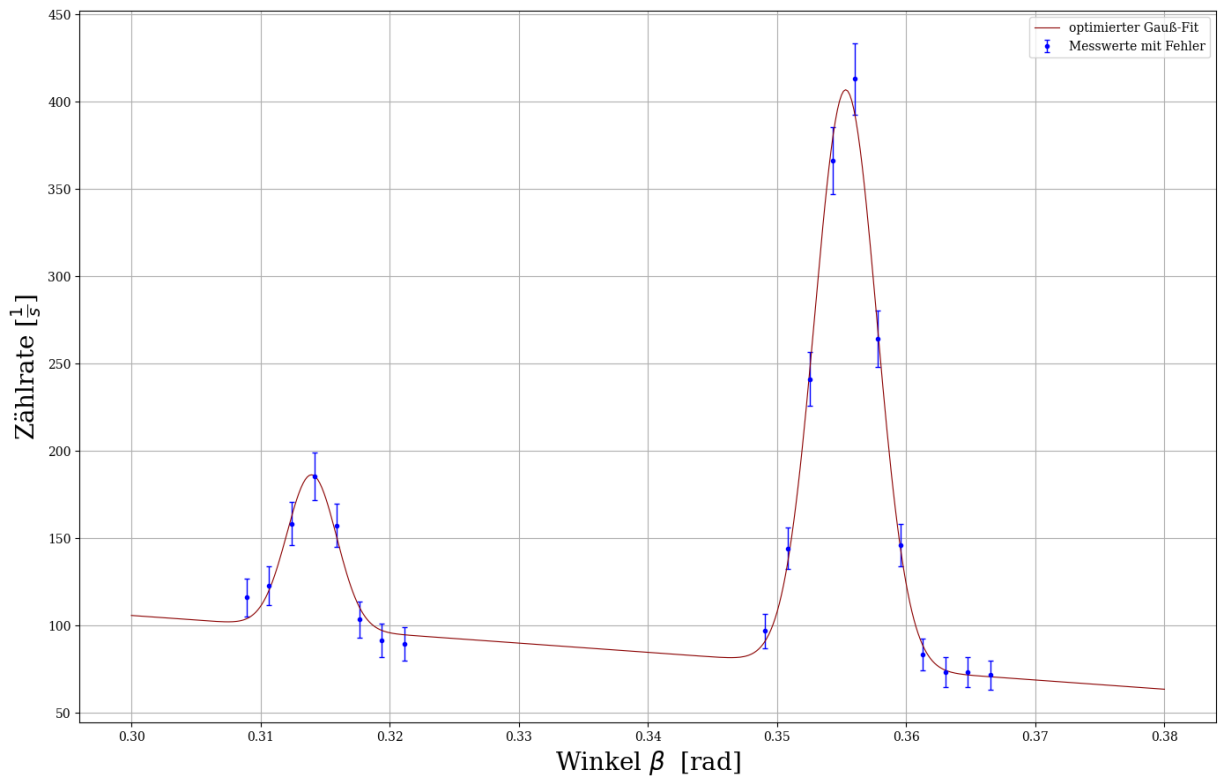
zweite Ordnung

```
In [28]: beta3, counts3=np.loadtxt('b2&3_korte_data.txt',skiprows=1, usecols=(0,1), unpack=True)
beta3=beta3*(2*np.pi)/360
counts3_err=np.sqrt(counts3)
```

```
In [29]: #Fit der Messdaten
popt4,pcov4=curve_fit(gauß_optimiert,beta3,counts3,sigma=counts3_err,p0=[100,300,40],
print(popt4)
```

[-5.27876875e+02 2.64215164e+02 8.78799861e+01 3.13970349e-01
 1.92557029e-03 3.30107051e+02 3.55326178e-01 2.40260329e-03]

```
In [31]: #Plot der Messdaten
plt.errorbar(beta3, counts3, yerr=counts3_err,linewidth=1,fmt='b.',linestyle='',label=
plt.plot(np.linspace(0.30,0.38,500),gauß_optimiert(np.linspace(0.30,0.38,500),*popt4
plt.xlabel(r'Winkel $ \beta $ [rad]', size=20)
plt.ylabel(r'Zählrate [$\frac{1}{s}$]', size=20)
#plt.title(r'Diagramm 7: Analyse der $K_\alpha$- & $K_\beta$-Linie (LiF, zweite Ordn
plt.grid()
plt.legend(frameon=True)
plt.tight_layout()
plt.savefig('V255Diagramm7.pdf', format='PDF')
```



In [32]:

```
chi2_4=np.sum((gauß_optimiert(beta3,*popt4)-counts3)**2/counts3_err**2)

dof4=len(beta3)-8

chi2_4_red=chi2_4/dof4

print('Güte des Fits:')
print('chi2= ',chi2_4)
print('chi2_red= ',chi2_4_red)
print()

#Fitwahrscheinlichkeit
prob4=round(1-chi2.cdf(chi2_4,dof4),3)*100

print('Fitwahrscheinlichkeit',prob4,' %')
```

Güte des Fits:
 chi2= 6.07618725178876
 chi2_red= 0.5523806592535236

Fitwahrscheinlichkeit 86.8 %

In [33]:

```
#gesuchte Winkel aus Gaußkurven
print('Für die zweite Ordnung folgen die berechneten Winkel:')
print('K_alpha_2: (',popt4[6], ' +/- ',popt4[7],') rad' )
print('K_beta_2: (',popt4[3], ' +/- ',popt4[4],') rad' )
```

Für die zweite Ordnung folgen die berechneten Winkel:
 K_alpha_2: (0.3553261775889331 +/- 0.002402603290456352) rad
 K_beta_2: (0.3139703494698608 +/- 0.0019255702905726765) rad

In [36]:

```
#Wellenlänge analog Grenzwellenlänge (allerdings Ordnung 2, deshalb Vorfaktor=1, da
lambdaK_beta2=d_LiF*np.sin(popt4[3])
lambdaK_beta2_err=np.sqrt((d_LiF*np.cos(popt4[3])*popt4[4])**2)

lambdaK_alpha2=d_LiF*np.sin(popt4[6])
lambdaK_alpha2_err=np.sqrt((d_LiF*np.cos(popt4[6])*popt4[7])**2)

print('Wellenlängen:')
```

```
print('K_alpha_2: (',lambdaK_alpha2*1e12,' +/- ',lambdaK_alpha2_err*1e12,') pm' )
print('K_beta_2: (',lambdaK_beta2*1e12,' +/- ',lambdaK_beta2_err*1e12,') pm' )
```

Wellenlängen:

K_alpha_2: (70.06629186260942 +/- 0.45365753232750283) pm

K_beta_2: (62.1998360817144 +/- 0.3688517241955787) pm

```
In [37]: #Vergleich mit Literaturwerten

diffalpha2=np.abs(lambdaK_alpha2-lambdaK_alphalit)
diffbeta2=np.abs(lambdaK_beta2-lambdaK_betalit)

print('Der Vergleich mit den Literaturwerten liefert:')
print('diff_lambdaK_alpha2 =( ',diffalpha2*1e12,' +/- ',lambdaK_alpha2_err*1e12,') p
print('diff_lambdaK_beta2 =( ',diffbeta2*1e12,' +/- ',lambdaK_beta2_err*1e12,') pm =>
```

Der Vergleich mit den Literaturwerten liefert:

diff_lambdaK_alpha2 =(1.033708137390593 +/- 0.45365753232750283) pm => sigma = 2.28

diff_lambdaK_beta2 =(0.9001639182855891 +/- 0.3688517241955787) pm => sigma = 2.44

c) Bestimmung der Einsatzspannung U und erneute Bestimmung von h

```
In [38]: #eingestellte Beschleunigungsspannungen
U=np.linspace(20,35,16)

#gemessene Zählraten
counts4=np.array([1.4,1.6,4.15,30.65,86.65,106.9,151.2,191.7,242.6,277.80,318.20,354
counts4_err=np.sqrt(counts4)
```

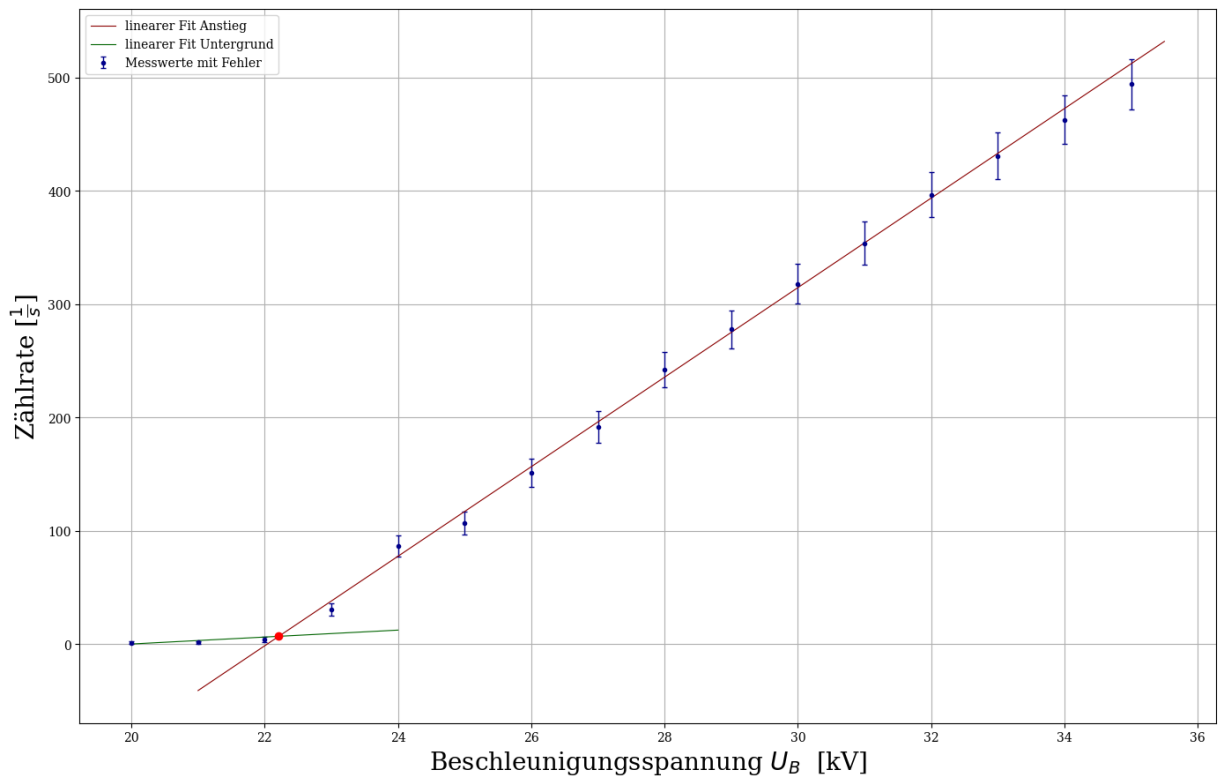
```
In [42]: popt5,pcov5=curve_fit(linear,U[4:-2],counts4[4:-2],sigma=counts4_err[4:-2])
popt6,pcov6=curve_fit(linear,U[:4],counts4[:4],sigma=counts4_err[:4])
UE=(popt6[1]-popt5[1])/(popt5[0]-popt6[0])

#Plot der Messdaten
plt.errorbar(U, counts4, yerr=counts4_err,linewidth=1,linestyle=' ',marker='.',color
plt.xlabel(r'Beschleunigungsspannung $U_B$ [kV]', size=20)
plt.ylabel(r'Zählrate [$\frac{1}{s}$]', size=20)
#plt.title(r'Diagramm 8: Spannungsabhängigkeit der Zählrate für $\beta=7.5^\circ$', size
plt.grid()
#Fit linearer Anstieg
plt.plot(np.linspace(21,35.5),linear(np.linspace(21,35.5),*popt5),linewidth=0.8,colo

#Fit Untergrund
plt.plot(np.linspace(20,24,2),linear(np.linspace(20,24,2),*popt6),linewidth=0.8,colo
plt.plot(UE, linear(UE, *popt5), 'o', color='red')

plt.legend(frameon=True)
plt.tight_layout()
plt.savefig('V255Diagramm8.pdf', format='PDF')
print(popt5, popt6)
```

```
[ 39.51057951 -870.62832714] [ 3.06564351 -61.12284251]
```



```
In [43]: #Güte des Fits des Anstiegs
chi2_5=np.sum((linear(U[4:-2],*popt5)-counts4[4:-2])**2/counts4_err[4:-2]**2)

#Freiheitsgrade
dof5=len(counts4[4:-2])-2

chi2_5_red=chi2_5/dof5

print('Güte des Fits:')
print('chi_squared= ',chi2_5)
print('chi_squared_red= ',chi2_5_red)
print()

#Fitwahrscheinlichkeit
prob5=round(1-chi2.cdf(chi2_5,dof5),3)*100

print('Fitwahrscheinlichkeit:',prob5,' %')
```

Güte des Fits:
chi_squared= 2.518111495208806
chi_squared_red= 0.31476393690110077

Fitwahrscheinlichkeit: 96.1 %

```
In [44]: #Güte des Fits des Untergrundes
chi2_6=np.sum((linear(U[:4],*popt6)-counts4[:4])**2/counts4_err[:4]**2)

#Freiheitsgrade
dof6=len(counts4[:4])-2

chi2_6_red=chi2_6/dof6

print('Güte des Fits:')
print('chi_squared= ',chi2_6)
print('chi_squared_red= ',chi2_6_red)
print()
```



```
#Fitwahrscheinlichkeit
prob6=round(1-chi2.cdf(chi2_6,dof6),3)*100

print('Fitwahrscheinlichkeit: ',prob6,' %')
```

Güte des Fits:
 chi_squared= 18.646028028327358
 chi_squared_red= 9.323014014163679

Fitwahrscheinlichkeit: 0.0 %

```
In [45]: #Spannungsgrenze aus Schnittpunkt
UE=(popt6[1]-popt5[1])/(popt5[0]-popt6[0])
UE_err=np.sqrt(pcov6[1,1]/(popt5[0]-popt6[0])**2+pcov5[1,1]/(popt5[0]-popt6[0])**2+(

print('UE = ',UE,' +/- ',UE_err,' kV')
```

UE = 22.211741152243707 +/- 2.7124539895345867 kV

Bestimmung von h aus U_E

```
In [47]: #Wellenlänge für theta=7.5°
lambda_U=2*d_LiF*np.sin(7.5*(2*np.pi)/360)
#Daraus folgt mit UE und GL.(1):
h2=lambda_U*e*(UE*1e3)/c
h2_err=h2*np.sqrt((UE_err/UE)**2)

print('h=',h2,'+- ',h2_err)
diff_h2=np.abs(h_lit-h2)

print('diff_h1 =( ',diff_h2,' +/- ',h2_err,' ) Js')
lambda_U
```

h= 6.241256483276339e-34 +- 7.621699231832361e-35
 diff_h1 =(3.8484351672366133e-35 +/- 7.621699231832361e-35) Js

Out[47]: 5.2575950226236776e-11

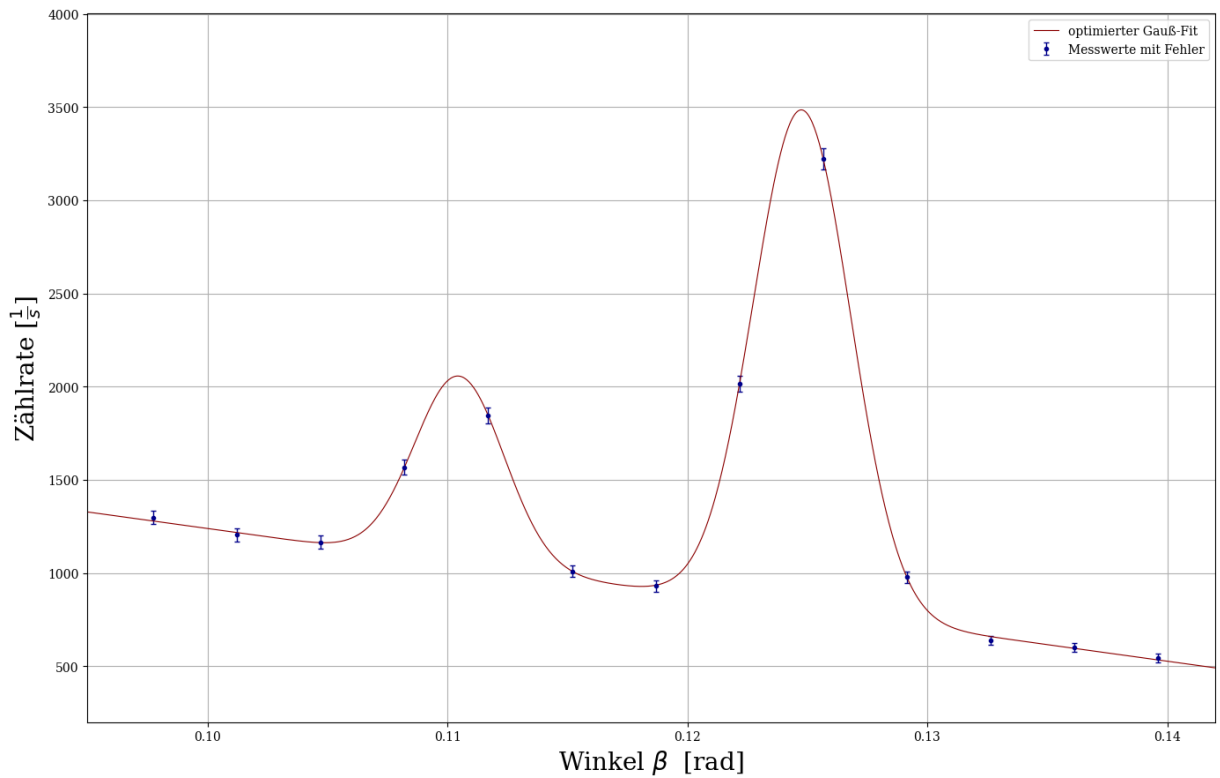
Teil 2: Messung des Röntgenspektrums mit einem NaCl-Kristall

Bestimmung der Lage der K_α und der K_β

```
In [48]: beta4, counts5=np.loadtxt('2_korte_data.txt',skiprows=1, usecols=(0,1), unpack=True)
beta4=beta4*(2*np.pi)/360
counts5_err=np.sqrt(counts5)
```

```
In [49]: mask1=np.all([beta4>0.095, beta4<0.142],axis=0)
popt7,pcov7=curve_fit(gauß_optimiert,beta4[mask1],counts5[mask1],sigma=counts5_err[m
```

```
In [57]: plt.errorbar(beta4, counts5, yerr=counts5_err,linewidth=1,linestyle=' ',marker='.',c
x=np.linspace(0.095,0.142,1000)
plt.plot(x,gauß_optimiert(x,*popt7),linewidth=0.8,color='darkred',label='optimierter
plt.xlabel(r'Winkel $\beta$ [rad]', size=20)
plt.ylabel(r'Zählrate [$\frac{1}{s}$]', size=20)
#plt.title(r'Diagramm 9: Analyse der $K_\alpha$- & $K_\beta$-Linie (NaCl, erste Ordn
plt.grid()
plt.legend(frameon=True)
plt.xlim(0.095,0.142)
plt.ylim(200,4000)
plt.tight_layout()
plt.savefig('V255Diagramm9.pdf', format='PDF')
```



```
In [58]: #Güte des Fits
chi2_7=np.sum((gauß_optimiert(beta4[mask1],*popt7)-counts5[mask1])**2/counts5_err[ma

#Freiheitsgrade
dof7=len(beta4[mask1])-8

chi2_7_red=chi2_7/dof7

print('Güte des Fits:')
print('chi_squared= ',chi2_7)
print('chi_squared_red= ',chi2_7_red)
print()

#Fitwahrscheinlichkeit
prob7=round(1-chi2.cdf(chi2_7,dof7),3)*100

print('Fitwahrscheinlichkeit: ',prob7,' %')
```

Güte des Fits:
chi_squared= 1.358215865362891
chi_squared_red= 0.27164317307257824

Fitwahrscheinlichkeit: 92.9 %

```
In [59]: #gesuchte Winkel entsprechen Mittelwerten der Gaußkurven
print('Für die erste Ordnung folgen die berechneten Winkel:')
print('K_alpha_2: ( ',popt7[6], ' +/- ',popt7[7],') rad' )
print('K_beta_2: ( ',popt7[3], ' +/- ',popt7[4],') rad' )
```

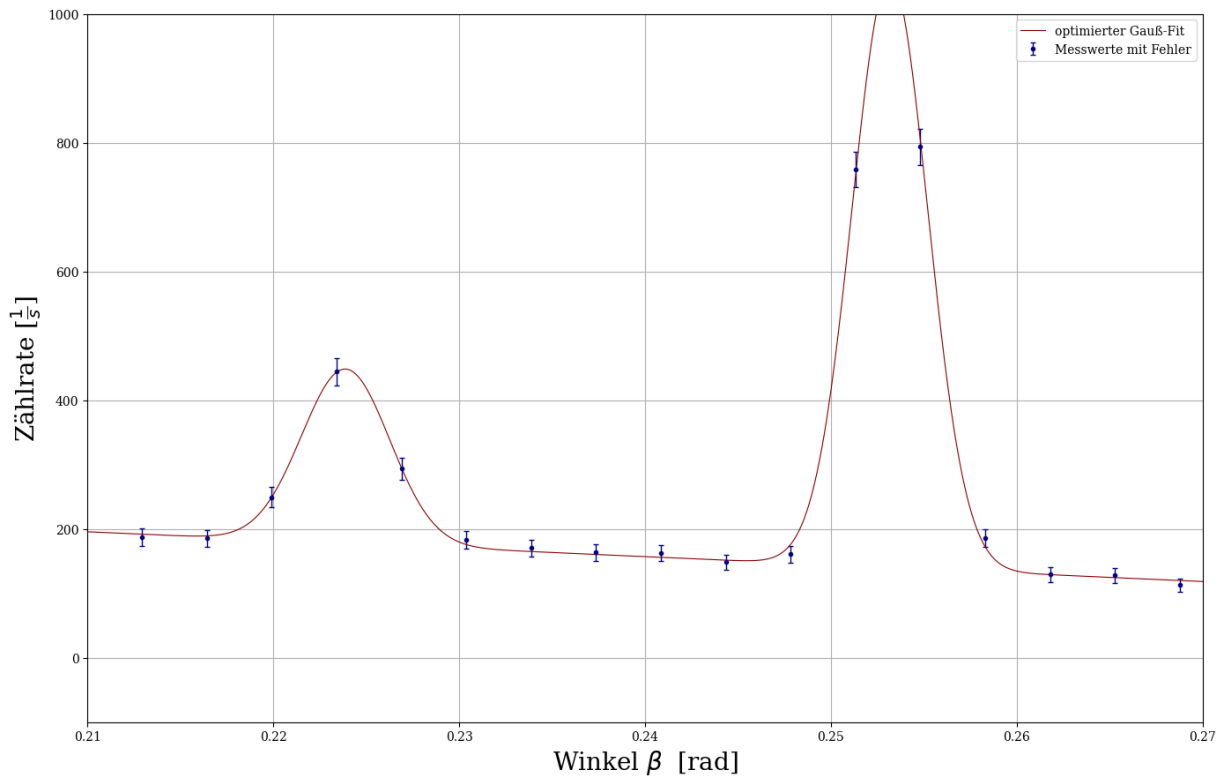
Für die erste Ordnung folgen die berechneten Winkel:
K_alpha_2: (0.12477635143977492 +/- 0.002019588342814202) rad
K_beta_2: (0.11049281818766853 +/- 0.0018652160230034247) rad

zweite Ordnung

```
In [60]: mask2=np.all([beta4>0.21, beta4<0.27],axis=0)
popt8,pcov8=curve_fit(gauß_optimiert,beta4[mask2],counts5[mask2],sigma=counts5_err[m
```

```
In [61]: plt.errorbar(beta4, counts5, yerr=counts5_err,linewidth=1,linestyle=' ',marker='.',c
x=np.linspace(0.21,0.27,1000)
```

```
plt.plot(x, gauß_optimiert(x, *popt8), linewidth=0.8, color='darkred', label='optimierter
plt.xlabel(r'Winkel  $\beta$  [rad]', size=20)
plt.ylabel(r'Zählrate  $[\frac{1}{s}]$ ', size=20)
#plt.title(r'Diagramm 10: Analyse der  $K_{\alpha}$ - &  $K_{\beta}$ -Linie (NaCl, zweite Or
plt.grid()
plt.legend(frameon=True)
plt.xlim(0.21, 0.27)
plt.ylim(-100, 1000)
plt.tight_layout()
plt.savefig('V255Diagramm10.pdf', format='PDF')
```



```
In [55]: #Güte des Fits
chi2_8=np.sum((gauß_optimiert(beta4[mask2], *popt8)-counts5[mask2])**2/counts5_err[ma

#Freiheitsgrade
dof8=len(beta4[mask2])-8

chi2_8_red=chi2_8/dof8

print('Güte des Fits:')
print('chi_squared= ', chi2_8)
print('chi_squared_red= ', chi2_8_red)
print()

#Fitwahrscheinlichkeit
prob8=round(1-chi2.cdf(chi2_8, dof8), 3)*100

print('Fitwahrscheinlichkeit: ', prob8, ' %')
```

```
Güte des Fits:
chi_squared= 4.246043987828231
chi_squared_red= 0.4717826653142479
```

```
Fitwahrscheinlichkeit: 89.4 %
```

```
In [56]: #gesuchte Winkel entsprechen Mittelwerten der Gaußkurven
print('Für die zweite Ordnung folgen die berechneten Winkel:')
print('K_alpha_2: (', popt8[6], ' +/- ', popt8[7], ') rad' )
print('K_beta_2: (', popt8[3], ' +/- ', popt8[4], ') rad' )
```

Für die zweite Ordnung folgen die berechneten Winkel:

K_alpha_2: (0.2531853616769591 +/- 0.0020362679802055383) rad

K_beta_2: (0.22388521051963686 +/- 0.0023728612999788816) rad

Gitterkonstante von NaCl

```
In [40]: #Wellenlängen von K_alpha und K_beta aus Aufgabe 1b) (sowohl erste, als auch zweite
K_a1=lambdaK_alpha1
K_b1=lambdaK_beta1
K_a2=lambdaK_alpha2
K_b2=lambdaK_beta2

lam=np.array([K_a1,K_b1,K_a2,K_b2])
lam_err=np.array([lambdaK_alpha1_err,lambdaK_beta1_err,lambdaK_alpha2_err,lambdaK_beta2_err])

#mit dem Bragg-Gesetz gilt für die Gitterkonstante: a=2*d=n*lam/sin(theta)
n=np.array([1,1,2,2])
theta=np.array([popt7[6],popt7[3],popt8[6],popt8[3]])
theta_err=np.array([popt7[7],popt7[4],popt8[7],popt8[4]])
a=n*lam/np.sin(theta)
a_err=a*np.sqrt((lam_err/lam)**2+(theta_err/np.tan(theta))**2)

a_mean=np.mean(a)
a_mean_err=1/4*np.sum(a_err**2)**0.5
a_mean_std=1/np.sqrt(len(a))*np.std(a)

print('Gitterkonstante:')
print('a_mean = (',round(a_mean*1e12,2),' +/- ',round(a_mean_err*1e12,2),'(sys.) +/- ')

Gitterkonstante:
a_mean = ( 558.33 +/- 4.3 (sys.) +/- 0.99 (stat.))pm
```

Bestimmung der Avogadro-Konstante N_A

```
In [41]: #Dichte NaCl
rho=2.164e3 #kg/(m**3)

#Molekulargewicht NaCl
M=58.44e-3 #kg

N_A=4*M/(rho*a_mean**3)
N_A_err=N_A*3*a_mean_err/a_mean
N_A_std=N_A*3*a_mean_std/a_mean

print('Avogadro-Zahl:')
print('N_A = (',round(N_A*1e-23,4),' +/- ',round(N_A_err*1e-23,4),' +/- ',round(N_A_std*1e-23,4),'(stat.)) 1/mol')
print()

#Vergleich mit dem Literaturwert
N_A_lit=6.0221e23 #1/mol

diff_N_A=np.abs(N_A_lit-N_A)
print('diff_N_A = (',diff_N_A,' +/- ',N_A_err,'(sys.) +/- ',N_A_std,'(stat.)) 1/mol')
print('sigma(sys.)=',round(diff_N_A/N_A_err,2),' ; sigma(stat.)=',round(diff_N_A/N_A_std,2))

Avogadro-Zahl:
N_A = ( 6.2065 +/- 0.1436 +/- 0.0329 ) 10^23 1/mol

diff_N_A =( 1.8439297350641365e+22 +/- 1.4356568242201878e+22 (sys.) +/- 3.292932
106011728e+21 (stat.)) 1/mol
sigma(sys.)= 1.28 ; sigma(stat.)= 5.6
```

In []: