

Versuch 255

Frauendorfer Andreas

09.01.2022

1 Einleitung

1.1 Röntgenröhre

In einer Röntgenröhre werden freie Elektronen mit einer Beschleunigungsspannung von ca. 10 bis 100kV auf ein Anodenmaterial geschossen. Dabei verlieren die Elektronen Energie in Form von elektromagnetischer Strahlung. Die größte Energie die ein Elektron verlieren kann ist, wenn es durch ein einziges Anodenatom komplett abgebremst wird. Die Grenzwellenlänge dafür ergibt sich zu:

$$E = eU = h\nu_{gr} = h \frac{c}{\lambda_{gr}} \quad (1)$$

mit dem Planckschem Wirkungsquantum h und der Lichtgeschwindigkeit c . Das charakteristische Spektrum besteht aus diskreten, materialspezifischen Wellenlängen.

1.2 K-Serie

Das charakteristische Spektrum entsteht, wenn ein beschleunigtes Elektron ein Atom eigenes Elektron aus der Hülle des Anodenatoms herausschlägt. Trifft es das aus der untersten Schale, der K-Schale, dann nennt man die resultierende Strahlung K-Strahlung. K_α -Strahlung heißt dabei, dass Elektronen aus der L-Schale nachrücken um die K-Schale aufzufüllen, K_β -Strahlung steht dabei für eine Kompensation durch Elektronen aus der M schale. Die Energie der charakteristischen Strahlung lässt sich dabei mit dem Moseley'schen Gesetz abschätzen:

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

mit der Rydbergkonstante R_∞ , der Kernladungszahl Z und der Hauptquantenzahlen n bzw. m , als der Differenz, wie viele Schalen das Elektron zur energetisch niedrigeren Schale springt. A ist eine Abschirmungskonstante, bei der K_α -Strahlung wird der Kern z.B. nur von einem Elektron abgeschirmt, also $A = 1$.

$$E_{n \rightarrow m} = \frac{3}{4} hcR_\infty (Z - A)^2 \quad (3)$$

Will man genauere Werte für die Energien bekommen, muss man die Feinstruktur der Übergänge anschauen, die zusätzlich noch von der Drehimpulsquantenzahl

und der Spinquantenzahl abhängt. Dieses feinere Auflösung ist jedoch mit unserem Spektrometer nicht möglich.

1.3 Bragg-Reflexion

Wenn Röntgenstrahlung unter einem Winkel θ auf einer Oberfläche trifft, dann wird sie wie im Reflexionsgesetz vorausgesagt, reflektiert. Da die Strahlung durch mehrere Schichten durchdringen kann, hängt die Intensität in einem bestimmten Raumwinkel vom Gangunterschied Δs ab. Ist dieser ein vielfaches der Wellenlänge, so interferieren die Intensitäten konstruktiv, sonst löschen sie sich aus (Vielstrahlinterferenz). Mit dem Netzebenenabstand zweier benachbarter Gitterschichten folgt das Bragg'sche Gesetz:

$$\delta s = 2d \sin(\theta) = n\lambda \quad (4)$$

Die λ -Abhängigkeit erlaubt eine Monochromatisierung der Röntgenstrahlung und die Spezifität des Netzebenenabstands für Kristalle lässt die Messung des Spektrums der Röntgenquelle zu.

1.4 Drehkristallmethode

Hierbei wird der Einfallswinkel der Röntgenstrahlung durch Drehung des Kristalls um eine zur Strahlung senkrechte Achse variiert. Die Intensität je nach Winkel kann dann von einem Zählrohr gemessen werden.

Ist die Wellenlänge bekannt, kann die Kristallstruktur bestimmt werden (Röntgenstrukturanalyse).

1.5 Elementarzelle

Die Elementarzelle ist die kleinste periodische Struktureinheit eines Kristall. Bei NaCl und LiF ist dieser Abstand in alle drei euklidischen Raumrichtungen gleich groß.

Die Orientierung der Netzebende zum einfallenden Röntgenstrahl (Kristallschnitt) ist parallel zur Seitenfläche der Elementarzelle.

Um die Avogadro-Konstante zu bestimmen muss man das Volumen einer Elementarzelle kennen. Hier entspricht es der kubischem doppelten Gitterkonstante $V = (2d)^3$.

Außerdem braucht man die Anzahl der Atome, die einer Elementarzelle im Schnitt innewohnen.

Es liegen 8 Chlor (Fluor) Atome an den Ecken, die jedoch nur zu einem Achtel zu einer Elementarzelle gehören, also auch noch zu 7 anderen Elementarzellen gehören.

12 Natrium (Lithium) Atome liegen auf den Kanten einer Elementarzelle und jedes Atom liegt gleichzeitig noch in 3 weiteren Elementarzellen, trägt also jeweils mit einem Viertel bei.

Außerdem gibt es 6 weitere Chlor (Fluor) Atome, die jeweils mit $\frac{1}{2}$ beitragen. Zuletzt befindet sich noch ein Natrium (Lithium) Atom genau in der Mitte einer Elementarzelle und trägt deshalb auch als ganzes bei.

Somit befinden sich im Schnitt 4 Chlor (Fluor) Atome und 4 Natrium (Lithium)

Atome in einer Elementarzelle.

Damit folgt die Avogadro-Konstante als das Vierfache des Molvolumens V_{Mol} normiert auf das Volumen einer Elementarzelle.

$$N_A = 4 \frac{V_{Mol}}{V} = 4 \frac{V_{Mol}}{(2d)^3} = \frac{1}{2} \frac{M_{Mol}}{\rho d^3} \quad (5)$$

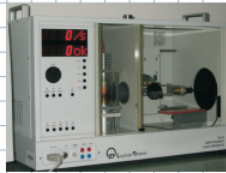
2 Auswertung

Messprotokoll V255

Andreas Frauendorf, Junis Heindl

Versuchsaufbau

Handwritten signature



Röntengerät mit Röntgenröhre (Molybdän-Anode)
Goniometer
Zählrohr
LiF-Kristall
NaCl-Kristall
Computer mit Drucker
Leuchtschirm mit CCD-Kamera (nur ein Aufbau vorhanden)

Aufgabe 1)

b) erste Ordnung: $8,2^\circ - 10,5^\circ$ ($8,2^\circ - 9,4^\circ$, $9,4^\circ - 10,5^\circ$)
zweite Ordnung: $17,5^\circ - 21^\circ$ ($17,5^\circ - 19^\circ$, $19,5^\circ - 21^\circ$)

c) Tabelle: Zählrate - Spannung Verhältnisse:

U [kV]	Zählrate n	U [V]	Zählrate n
20	1,30	28	107,0
21	2,55	29	114,7
22	3,40	30	109,3
23	10,25	31	74,5
24	29,05	32	761,7
25	52,30	33	774,3
26	27,55	34	133,0
27	86,00	35	219,6

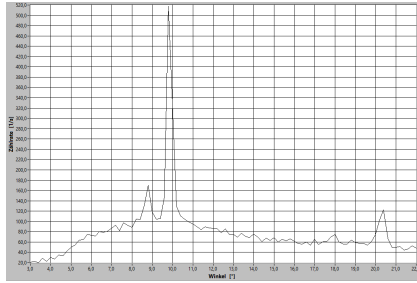


Figure 1: K-Linien von Lithium-Fluorid

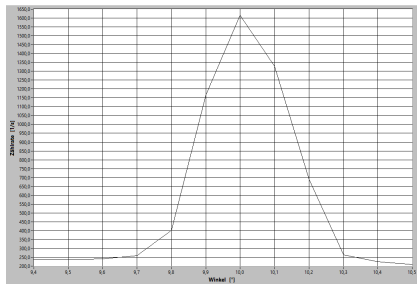


Figure 2: $K\alpha$ -Linien erste Ordnung

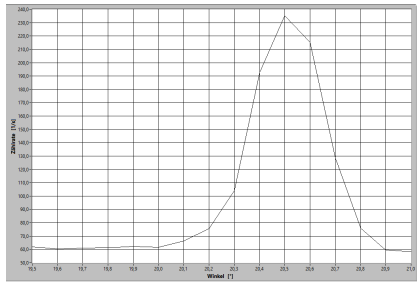


Figure 3: $K\alpha$ -Linien zweite Ordnung

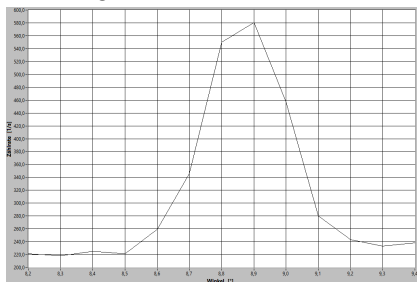


Figure 4: $K\beta$ -Linien erste Ordnung

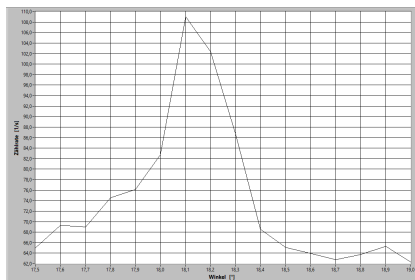


Figure 5: K β -Linien zweite Ordnung

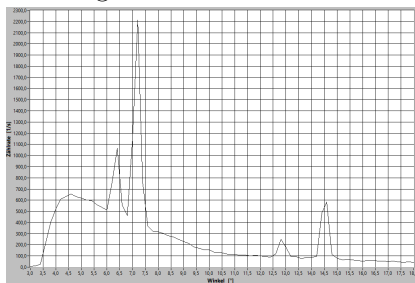


Figure 6: K-Linien von Natrium-Chlorid

2.1 LiF-Kristall

In diesem Versuchsteil bestimmten wir die Grenzwellenlänge des Bremsspektrums von 35kV und daraus die Planck-Konstante h . Dazu plotten wir das Bremsspektrum und interpolieren linear den Grenzwinkel, also den Schnittpunkt mit der x-Achse.

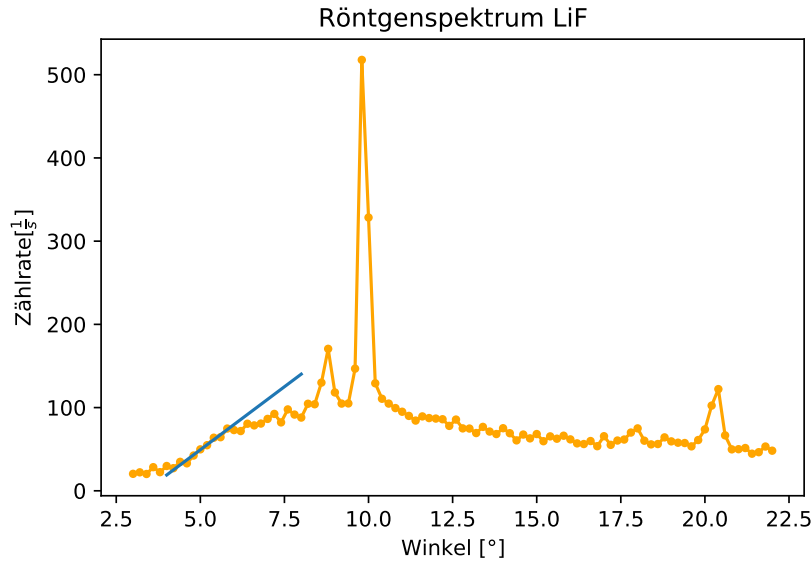


Figure 7

Wir ermitteln den Schnittpunkt indem wir unsere Fitfunktion mit den Best-Fit Parametern 0 setzen und nach x auflösen. Mit Python ergibt sich daraus ein Winkel von

$$\theta_{gr} = (3,4 \pm 1,4)^\circ \quad (6)$$

Der Fehler ergibt sich aus den quadratisch addierten Werten der Kovarianzmatrix der Bestfit parameter (siehe Python skript), hier genannt a und b.

$$\Delta\theta = \theta \cdot \sqrt{\left(\frac{\Delta a}{a}\right)^2 + \left(\frac{\Delta b}{b}\right)^2} \quad (7)$$

Um daraus die Grenzwellenlänge zu bestimmen, verwenden wir das Bragg'sche Gesetz mit $n=1$

$$\lambda_{gr} = 2d \sin(\theta_{gr}) = 2,39 \cdot 10^{-11} m \quad (8)$$

Der Fehler pflanzt sich dabei wie folgt fort:

$$\Delta\lambda = 2d \cos(\theta_{gr}) \Delta\theta_{gr} = 0,98 \cdot 10^{-11} m \quad (9)$$

Daraus folgt für die Grenzwellenlänge:

$$\lambda_{gr} = (2,39 \pm 0,98) \cdot 10^{-11} m \quad (10)$$

Das plancksche Wirkungsquantum erhalten wir durch Umstellen von Gleichung (1):

$$h = \frac{eU\lambda_{gr}}{c} \quad (11)$$

Der Fehler kann wie folgt abgeschätzt werden:

$$\Delta h = \frac{eU}{c} \Delta \lambda_{gr} \quad (12)$$

Daraus bestimmt sich der Wert der Plack-Konstante zu:

$$h = (4,5 \pm 1,9) \cdot 10^{-34} Js \quad (13)$$

Im Vergleich zum Literaturwert von $6,6261 \cdot 10^{-34}$ liegt unser Ergebnis im 2σ -Bereich.

Der Winkel, bei dem das Spektrum zweiter Ordnung einsetzt, lässt sich durch das setzen von $n=2$ berechnen. Dadurch kürzt sich die 2 in der Braggbeziehung und es ergibt sich:

$$\arcsin\left(\frac{\lambda_{gr}}{d}\right) = \theta_{gr} \quad (14)$$

der Fehler pflanzt sich wie folgt fort:

$$\Delta \theta_{gr} = \frac{\Delta \lambda_{gr}}{d \sqrt{1 - (\lambda_{gr}/d)^2}} \quad (15)$$

Daraus folgt

$$\theta_{gr} = (6,8 \pm 3,0)^\circ \quad (16)$$

Die charakteristischen Linien der ersten und zweiten Ordnung wurden extra genauer ausgemessen. Die Winkel unter denen die einzelnen Linien auftreten wurden aus den Plots abgelesen. Mit Gleichung (4) ergibt sich dann:

$$\theta_{\alpha 1} = (10,0 \pm 0,1)^\circ : \lambda_{\beta 1} = (70,0 \pm 0,7) pm \quad (17)$$

$$\theta_{\beta 1} = (8,9 \pm 0,1)^\circ : \lambda_{\alpha 1} = (62,3 \pm 0,7) pm \quad (18)$$

$$\theta_{\alpha 2} = (20,5 \pm 0,1)^\circ : \lambda_{\alpha 2} = (70,5 \pm 0,4) pm \quad (19)$$

$$\theta_{\beta 2} = (18,1 \pm 0,1)^\circ : \lambda_{\beta 2} = (62,6 \pm 0,4) pm \quad (20)$$

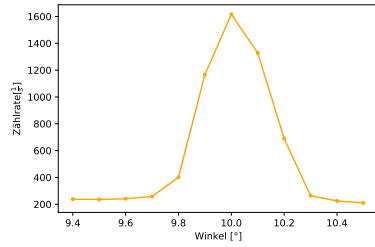


Figure 8: K α erste Ordnung

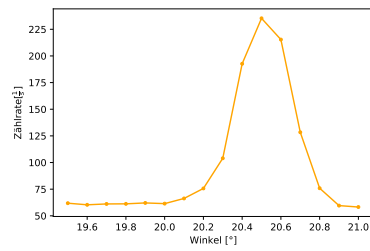


Figure 9: K α zweite Ordnung

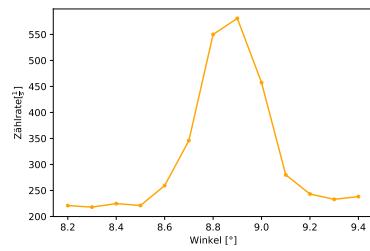


Figure 10: K β erste Ordnung

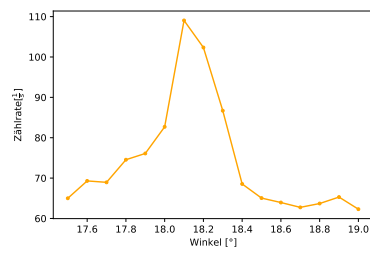


Figure 11: K β zweite Ordnung

Die Halbwertsbreite H der K_{α} -Linie kann man aus Diagramm 1 ablesen.

$$H = 0,25 \pm 0,05)^{\circ} \quad (21)$$

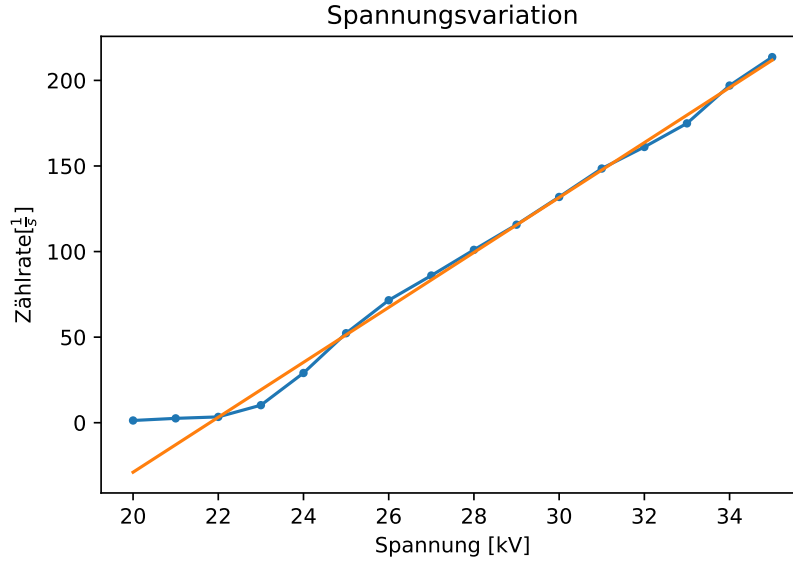


Figure 12

Jetzt wird nochmal das Plancksche Wirkungsquantum bestimmt, indem der Winkel bei $7,5^\circ$ festgehalten wird und die Einsatzspannung variiert wird. Obige Extrapolation gibt auch hier den Schnittpunkt mit der x-Achse wieder. Diese Methode ist genauer also die zuvorige, weil nur eine Wellenlänge verwendet wird und somit etwagie Fehler nicht korrigiert werden müssen. Für die Nullstelle ergibt sich ein Wert von:

$$U = (21,8 \pm 0,8)kV \quad (22)$$

Wir verwenden wieder Formel (10) für das Plancksche Wirkungsquantum und dessen Fehler (11). So erhalten wir:

$$h = (6,20 \pm 0,18) \cdot 10^{-34} Js \quad (23)$$

2.2 NaCl-Kristall Röntgenspektrumsmessung

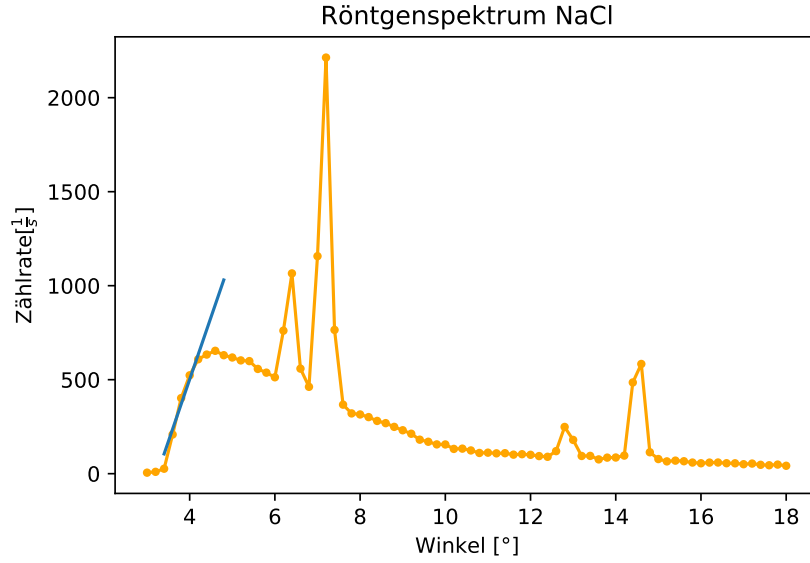


Figure 13

Wir bestimmen wieder die Winkel für die K_α und K_β -Strahlung.

$$\theta_{\alpha 1} = (7,2 \pm 0,1)^\circ \quad (24)$$

$$\theta_{\beta 1} = (6,5 \pm 0,1)^\circ \quad (25)$$

$$\theta_{\alpha 2} = (14,6 \pm 0,1)^\circ \quad (26)$$

$$\theta_{\beta 2} = (13,0 \pm 0,1)^\circ \quad (27)$$

mit $d = \frac{a}{2}$ berechnet sich die Gitterkonstante zu:

$$a = \frac{n\lambda_{gr}}{\sin(\theta)} \quad (28)$$

Der Fehler ergibt sich zu:

$$\Delta a = \sqrt{\left(\frac{n}{\sin(\theta)}\Delta\lambda\right)^2 + \left(\frac{n\lambda\cos(\theta)}{\sin^2(\theta)}\Delta\theta\right)^2} \quad (29)$$

Mit den Werten für die Wellenlängen ergibt sich die Gitterkonstante für die 4 Linien:

$$a_{\alpha 1} = (558 \pm 1, 4) pm \quad (30)$$

$$a_{\beta 1} = (559 \pm 1, 4) pm \quad (31)$$

$$a_{\alpha 2} = (550 \pm 1, 6) pm \quad (32)$$

$$a_{\beta 2} = (556, 3 \pm 1, 7) pm \quad (33)$$

Mittelt man diese Werte, erhält man:

$$a = (556 \pm 4) pm \quad (34)$$

Im Vergleich mit dem Literaturwert $a_L = 564 pm$ (Quelle: Wikipedia) weicht unser ermittelter Wert um 3sigma vom Literaturwert ab. Mit Gleichung (5) lässt sich die Avogadrozahl ermitteln:

$$N_A = \frac{1}{2} \frac{M_{Mol}}{\rho d^3} = \frac{4M_{mol}}{\rho a^3} = (6, 28 \pm 0, 12) \cdot 10^{23} \quad (35)$$

Der Fehler war hierbei:

$$\Delta N_A = 3N_A \cdot \frac{\Delta a}{a} \quad (36)$$

Der bestimmte Wert weicht um ca $2,15 \sigma$ vom Literaturwert $N_A = 6, 0221 \cdot 10^{23} \frac{1}{mol}$ (Quelle: Praktikumsskript) ab.

3 Diskussion

In diesem Versuch haben wir die Röntgenspektren von Lithium-Fluorid und Natrium-Chlorid ermittelt.

Wir haben daraus die Planckkonstante 2mal bestimmt und die Avogadrozahl ermittelt.

Zuerst wurde die Planckkonstante $h_{Li} = 6, 6261 \cdot 10^{-34} Js$ (Quelle Praktikumsskript) mithilfe des Bremspektrums des LiF-Kristalls bestimmt. Unser Ergebnis:

$$h = (4, 5 \pm 1, 9) \cdot 10^{-34} Js \quad (37)$$

weicht um weniger als 2σ vom Literaturwert ab. Diese Messmethode ist jedoch ungenau, da Eigenabsorptionen in der Anode oder im Röhrenfenster vernachlässigt werden und unterschiedliche Reflexionsvermögen des Kristalls korrigiert werden müssten. Außerdem ist die Best-Fit Gerade schwer zu bestimmen, da der lineare Anteil sehr klein ist. Auch könnte der Kristall durch Schmutz ungleichmäßig Röntgenstrahlen reflektieren, was zu weiteren Fehlern führen kann.

Auch wurde die Plank-Konstante bei der Messung mit konstantem Winkel und Variation der Spannung bestimmt.

$$h = (6, 20 \pm 0, 18) \cdot 10^{-34} Js \quad (38)$$

Die beste Schätzung des wahren Werts liegt näher am Literaturwert, jedoch ist die σ -Abweichung ca. 2,4. Dies kann an dem kleinen Fehler liegen (3Prozent). Eventuell war der lineare Fit wieder zu ungenau oder Schmutz verantwortlich für die Abweichungen.

Die Avogadrozahl wurde zu einem Wert von

$$N_A = (6,28 \pm 0,12) \cdot 10^{23} \quad (39)$$

bestimmt und weicht vom Literaturwert $N_A = 6,0221 \cdot 10^{23} \frac{1}{mol}$ (Quelle: Praktikumsskript) ca 2,15 σ ab. Etwaiger Fehlerquellen könnten Verunreinigungen in der Kristallstruktur selber sein, sodass die spezifischen Atome die pro Volumeneinheit gezählt werden nur im Mittel stimmt.

Die Gitterkonstante wurde zu

$$a = (556 \pm 4) pm \quad (40)$$

bestimmt und weicht um 3sigma vom Literaturwert ab. Hier kommen die Fehler der Wellenlänge und des Winkels ins Spiel, was nochmals mehr Unsicherheiten bringt. Allerdings konnte durch die Mittelung trotzdem ein akzeptables Ergebnis berechnet werden.

Auswertung Versuch 255

January 10, 2022

```
[3]: import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit
%matplotlib inline
```

1 Auswertung

1.1 Aufgabe 1a und 2

```
[11]: def comma_to_float(valstr):
        return float(valstr.decode('utf-8').replace(',', '.'))

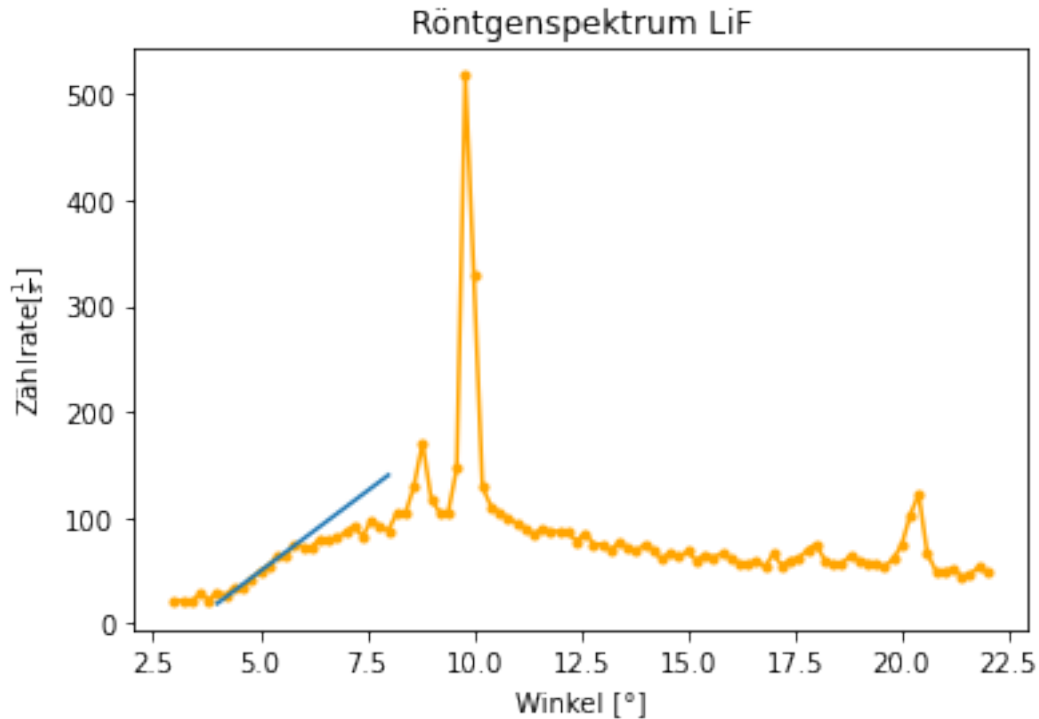
Winkel1, Rate1 = np.loadtxt('06_12_2021_14_30_07.txt', usecols = (0, 1),
    ↪ converters = {0:comma_to_float, 1: comma_to_float}, unpack = True)
plt.plot(Winkel1, Rate1, marker = '.', color = 'orange')

plt.xlabel('Winkel [°]')
plt.ylabel('Zählrate' + r'[$\frac{1}{s}$]')
plt.savefig('Röntgenspektrum_LiF_V255.pdf', format = 'pdf')

def linear(x, a, b):
    return a * x + b
popt1, pcov1 = curve_fit(linear, Winkel1[11:15], Rate1[11:15])
plt.plot(Winkel1[5:26], linear(Winkel1[5:26], *popt1))
nullstelle_LiF = -popt1[1]/popt1[0]
delta_nullstelle_LiF = nullstelle_LiF * np.sqrt((np.sqrt(pcov1[1][1])/
    ↪ popt1[1])**2 + (np.sqrt(pcov1[0][0])/popt1[0])**2)
print('Nullstelle LiF = ', nullstelle_LiF, '+/-', delta_nullstelle_LiF)
plt.title('Röntgenspektrum LiF')
plt.savefig('Röntgenspektrum_LiF_Extrapolation_V255.pdf', format = 'pdf')

plt.show()
print(popt1)
```

Nullstelle LiF = 3.3729372937294357 +/- 1.3911109490159874



[30.3 -102.2]

1.2 Aufgabe 1b

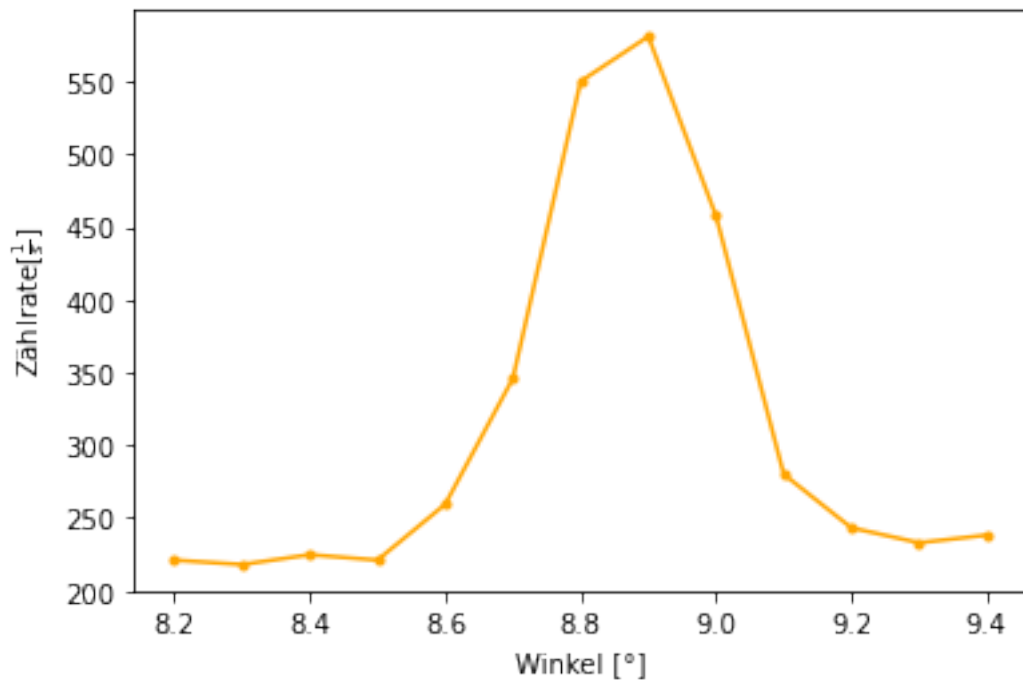
```
[12]: #zu 1b)
Winkel2, Rate2 = np.loadtxt('06_12_2021 14_43_00.txt', usecols = (0, 1),
    ↳ converters = {0:comma_to_float, 1: comma_to_float}, unpack = True)
plt.plot(Winkel2, Rate2, marker = '.', color = 'orange')
plt.xlabel('Winkel [°]')
plt.ylabel('Zählrate' + r'[$\frac{1}{s}$]')
plt.savefig('Kb-Linien_erste_Ordnung.pdf', format = 'pdf')
plt.show()

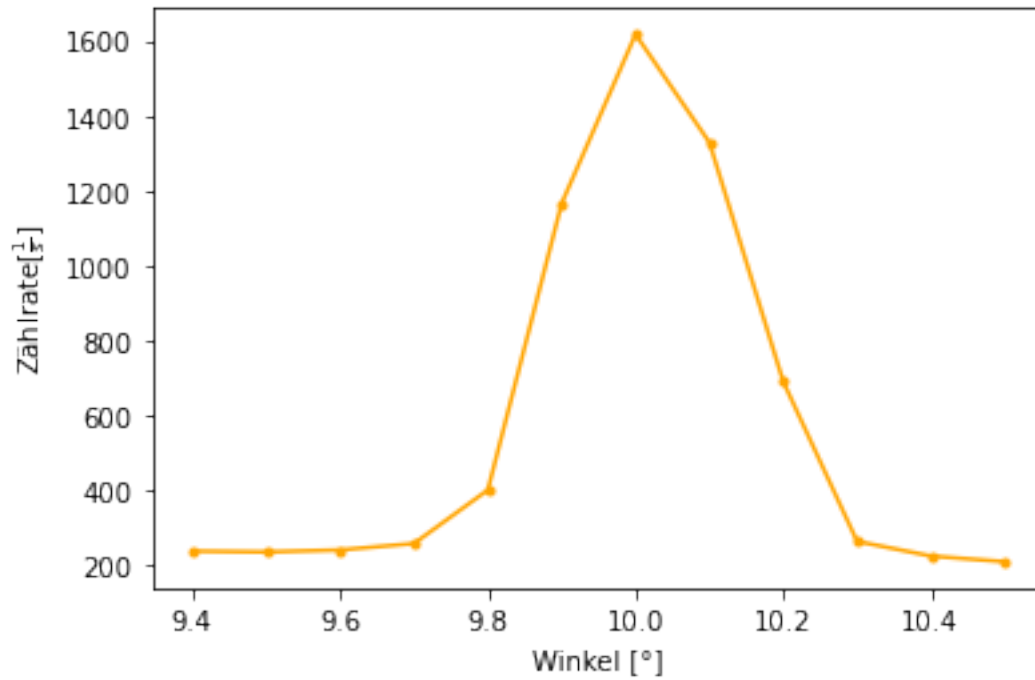
Winkel2, Rate2 = np.loadtxt('06_12_2021 14_50_08.txt', usecols = (0, 1),
    ↳ converters = {0:comma_to_float, 1: comma_to_float}, unpack = True)
plt.plot(Winkel2, Rate2, marker = '.', color = 'orange')
plt.xlabel('Winkel [°]')
plt.ylabel('Zählrate' + r'[$\frac{1}{s}$]')
plt.savefig('Ka-Linien_erste_Ordnung.pdf', format = 'pdf')
plt.show()

Winkel3, Rate3 = np.loadtxt('06_12_2021 14_58_13.txt', usecols = (0, 1),
    ↳ converters = {0:comma_to_float, 1: comma_to_float}, unpack = True)
```

```
plt.plot(Winkel3, Rate3, marker = '.', color = 'orange')
plt.xlabel('Winkel [°]')
plt.ylabel('Zählrate' + r'[$\frac{1}{s}$]')
plt.savefig('Kb-Linien_zweite_Ordnung.pdf', format = 'pdf', color = 'orange')
plt.show()
```

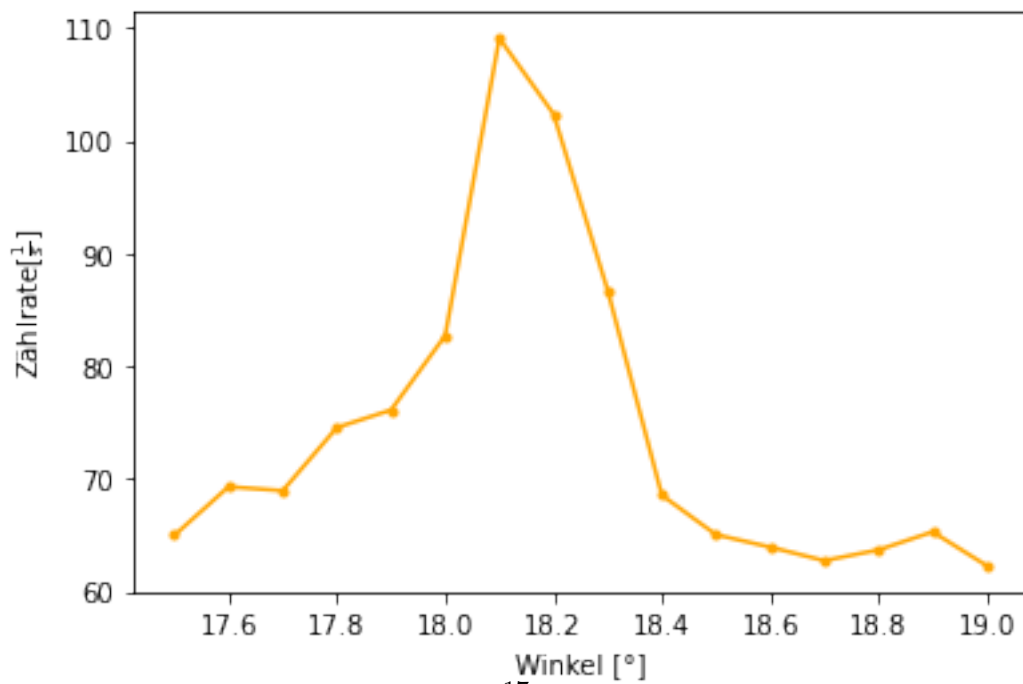
```
Winkel3, Rate3 = np.loadtxt('06_12_2021_15_04_53.txt', usecols = (0, 1),
    ↳ converters = {0: comma_to_float, 1: comma_to_float}, unpack = True)
plt.plot(Winkel3, Rate3, marker = '.', color = 'orange')
plt.xlabel('Winkel [°]')
plt.ylabel('Zählrate' + r'[$\frac{1}{s}$]')
plt.savefig('Ka-Linien_zweite_Ordnung.pdf', format = 'pdf')
plt.show()
```

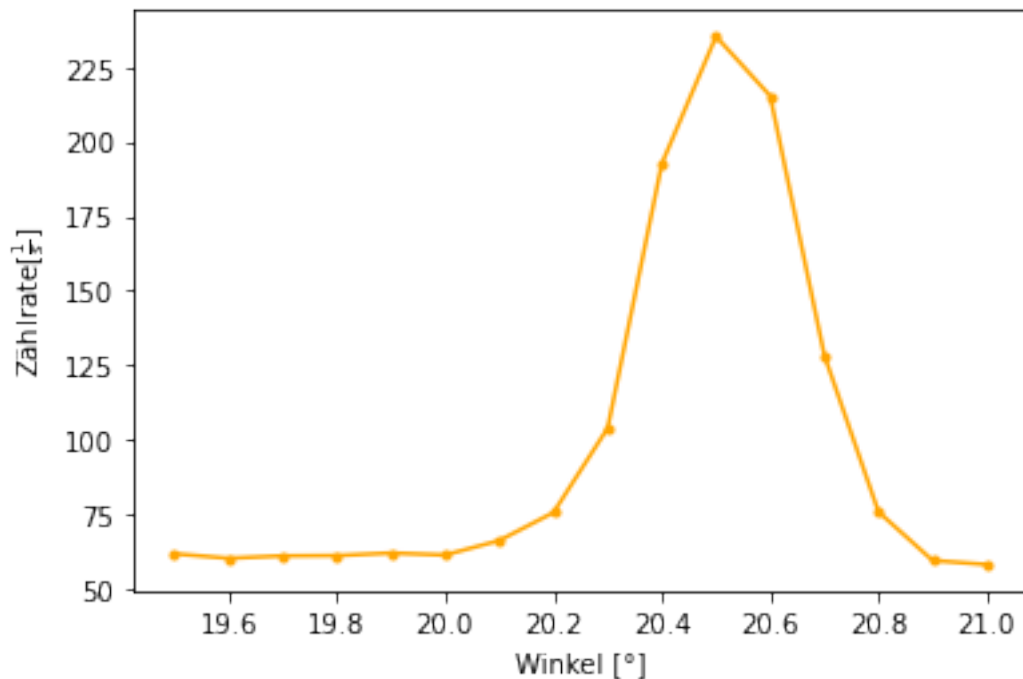




<ipython-input-12-9be02be40cbd>:20: MatplotlibDeprecationWarning: savefig() got unexpected keyword argument "color" which is no longer supported as of 3.3 and will become an error two minor releases later

plt.savefig('Kb-Linien_zweite_Ordnung.pdf', format = 'pdf', color = 'orange')





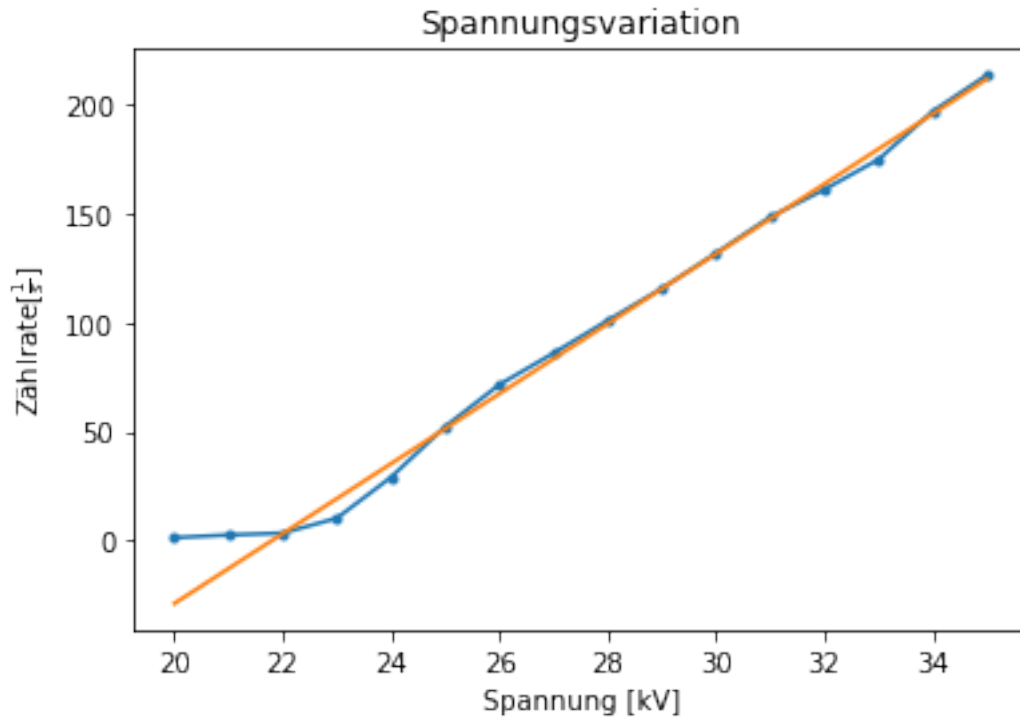
```
[13]: #entpacke Daten zu 1c
%matplotlib inline
Winkel, Rate = np.loadtxt('Versuch_1c.txt', skiprows = 1, usecols=(0,1), unpack_
    ↪ = True)
plt.plot(Winkel, Rate, marker = '.')
def fit_func(x, a, b):
    return a * x + b
popt3, pcov3 = curve_fit(fit_func, Winkel[4:], Rate[4:])
nullstelle_c = -popt3[0]/popt3[1] * 100
delta_nullstelle_c = nullstelle_c * np.sqrt((np.sqrt(pcov3[1][1])/popt3[1])**2_
    ↪ + (np.sqrt(pcov3[0][0])/popt3[0])**2)
print('Nullstelle = ', nullstelle_c, '+/-', delta_nullstelle_c)

plt.xlabel('Spannung [kV]')
plt.ylabel('Zählrate' + r'[\frac{1}{s}]')
plt.savefig('Abb_1c.pdf_V255', format = 'pdf')

plt.plot(Winkel, fit_func(Winkel, *popt3))
plt.title('Spannungsvariation')
plt.savefig('Abb_1c_Extrapolation_V255.pdf', format = 'pdf')
plt.show()
```

18

Nullstelle = 4.586802998661091 +/- 0.12871144628814202



Berechnungen in arrays

[]:

[]:

[]:

[]:

[]:

[]:

```
[14]: #Aufgabe 1a
print('Aufgabe 1a')
theta_a = 3.4 * (2*np.pi)/360
delta_theta_a = 1.4 * (2*np.pi)/360
d = 201.4 * 10**(-12)                                     #in m

lam_Gr = 2 * d * np.sin(theta_a)
delta_lam_Gr = 2 * d * np.cos(theta_a) * delta_theta_a
print('Grenzwellenlänge =', lam_Gr, '+/-', delta_lam_Gr)
#Bestimme das Planck'sche Wirkungsquantum
```

```

h = (lam_Gr * 1.6022*10**(-19) * 35 * 10**3)/(2.9979 * 10**8)
delta_h = (h*delta_lam_Gr)/lam_Gr
print('h=', h, '+/-', delta_h)
#Winkel, unter dem die zweite Beugungsordnung liegen müsste:
theta_2 = (np.arcsin(lam_Gr/d)) * 360/(2 * np.pi)
delta_theta_2 = (1 / (d * np.sqrt(1 - lam_Gr / d) ) * delta_lam_Gr) * 360 / (2*
↳ np.pi)
print('theta = ', theta_2, '+/-', delta_theta_2)

print()

#Aufgabe 1b
print('Aufgabe 1b')
d = 201.4 * 10**(-12) #in m
theta_b = np.array([10.1, 20.5, 8.9, 18.1]) * (2*np.pi)/360 #Winkel in Grad
n = np.array([1, 2, 1, 2]) ↳
↳ #Hauptquantenzahl
delta_theta_b = np.ones(4) * 0.1 * (2 * np.pi)/360 #Fehler
↳ Winkel in Grad

lam_b = (2*d*np.sin(theta_b))/n
delta_lam_b = (2*d*np.cos(theta_b)*delta_theta_b)/n

for i in range(len(theta_b)):
    print('Grenzwellenlänge = ', lam_b[i], delta_lam_b[i])

print()
#Halbwertsbreite
FWHM = 0.25 * (2*np.pi)/360
delta_FWHM = 0.05 * (2*np.pi)/360
n = 1
print('FWHM = ', FWHM, '+/-', delta_FWHM)
lam_FWHM = (2*d*np.sin(FWHM))/n
delta_lam_FWHM = (2*d*np.cos(FWHM)*delta_FWHM)/n
print('Wellenlänge, die der Halbwertsbreite entspricht =', lam_FWHM, '+/-',
↳ delta_lam_FWHM)

print()

#Aufgabe 1c)
print('Aufgabe 1c')
e = 1.6022 * 10**(-19) #Elementarladung
c = 2.9979 * 10**8 #Lichtgeschwindigkeit

#U = nullstelle_c * 1000 20 #Extrapolierte Spannung
↳ (Umrechnung in V)

```

```

#delta_U = delta_nullstelle_c * 1000
#theta_c = 7.5 * (2*np.pi)/360
#lam_neu = 2 * d * np.sin(theta_c)
#h_neu = (lam_neu * e * U) / c
#delta_h_neu = h_neu * delta_U / U
#print('Grenzwellenlänge = ', lam_neu)
#print('neu bestimmtes Planck'sches Wirkungsquantum h = ', h_neu, '+/-',
      ↪ delta_h_neu)

[a,b]=popt3
d_a=np.sqrt(pcov3[0,0])
d_b=np.sqrt(pcov3[1,1])
U_a = (np.sum(Rate[0:2])/3)
d_z = np.sqrt(Rate)
d_U = 1/3*(np.sqrt(d_z[0]**2+d_z[1]**d_z[2]**2))
U = ((U_a-b)/a)*10**3
print('U_E=', U, '+-', d_U)
theta_c = 7.5 * (2*np.pi)/360
lam_neu = 2 * d * np.sin(theta_c)
h_neu = (lam_neu * e * U) / c
delta_h_neu = h_neu * d_U / U
print('Grenzwellenlänge = ', lam_neu)
print('neu bestimmtes Planck'sches Wirkungsquantum h = ', h_neu, '+/
      ↪-',delta_h_neu)

```

Aufgabe 1a

Grenzwellenlänge = 2.388860727639734e-11 +/- 9.824936645363848e-12
 h= 4.468466027014022e-34 +/- 1.8377963649957252e-34
 theta = 6.812047041558294 +/- 2.977210501238369

Aufgabe 1b

Grenzwellenlänge = 7.06377172698524e-11 6.921240696326631e-13
 Grenzwellenlänge = 7.053176658565673e-11 3.2924899620495413e-13
 Grenzwellenlänge = 6.231734360142575e-11 6.945541841193489e-13
 Grenzwellenlänge = 6.257023292762938e-11 3.341151302533996e-13

FWHM = 0.004363323129985824 +/- 0.0008726646259971648
 Wellenlänge, die der Halbwertsbreite entspricht = 1.7575409798959188e-12 +/-
 3.515059652363588e-13

Aufgabe 1c

U_E= 21881.641247441617 +- 0.8306902238745482
 Grenzwellenlänge = 5.2575950226236776e-11
 neu bestimmtes Planck'sches Wirkungsquantum h = 6.148463642978694e-34 +/-
 2.334134164030156e-38

```
[15]: #Aufgabe 2
print('Aufgabe 2')
theta_2 = np.array([7.2, 14.7, 6.4, 13.0]) * (2*np.pi)/360
delta_theta_2 = np.ones(4) * 0.1 * (2*np.pi)/360

Gitterkonstante = (n*lam_b)/np.sin(theta_2)
delta_Gitterkonstante = np.sqrt(((n*delta_theta_2)/(np.
    ↳sin(theta_2)))**2+((n*lam_b*np.cos(theta_2)*delta_theta_2)/((np.
    ↳sin(theta_2))**2))**2)
#delta_Gitterkonstante = np.sqrt((n/(np.
    ↳sin(theta_2))*delta_theta_2)**2+((n*lam_b*np.cos(theta_2))/((np.
    ↳sin(theta_2))**2)*delta_theta_2)**2)

#for i in range(len(theta_2)):
#    print('Gitterkonstante =', Gitterkonstante[i], '+/-',
    ↳delta_Gitterkonstante[i])

# Bestimme durchschnittliche Gitterkonstante:
Gitterkonstante_mittel = (Gitterkonstante[0] + Gitterkonstante[1] +
    ↳Gitterkonstante[2] + Gitterkonstante[3])/4

delta_Gitterkonstante = np.sqrt(1/4 * ((Gitterkonstante_mittel -
    ↳Gitterkonstante[0])**2 + (Gitterkonstante_mittel - Gitterkonstante[1])**2 +
    ↳(Gitterkonstante_mittel - Gitterkonstante[2])**2 + (Gitterkonstante_mittel -
    ↳Gitterkonstante[3])**2))

print('durchschnittliche Gitterkonstante =', Gitterkonstante_mittel, '+/-',
    ↳delta_Gitterkonstante)

print()
#Bestimme die Avogadro-Zahl
rho = 2.164 * 10**3
M_mol = 58.44 * 10**(-3)
N_A = (4*M_mol)/(rho * Gitterkonstante_mittel**3)
delta_N_A = (12 * M_mol * delta_Gitterkonstante)/(rho*Gitterkonstante_mittel**4)
print('N_A =', N_A, '+/-', delta_N_A)
```

Aufgabe 2

durchschnittliche Gitterkonstante = 4.196886005587231e-10 +/-
1.416480122358451e-10

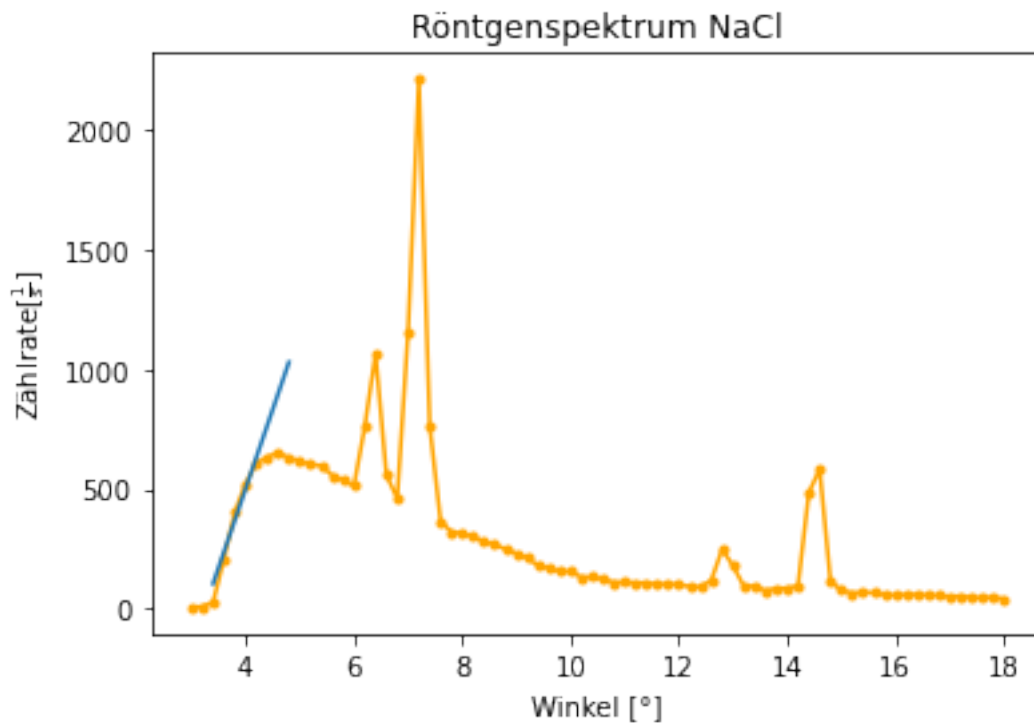
N_A = 1.4612732107506233e+24 +/- 1.4795716063821604e+24

1.3 NaCl Kristall Ausmessung

[17]: *#für NaCl*

```
Winkel4, Rate4 = np.loadtxt('06_12_2021 15_31_40.txt', usecols = (0, 1),  
    ↳ converters = {0: comma_to_float, 1: comma_to_float}, unpack = True)  
plt.plot(Winkel4, Rate4, marker = '.', color = 'orange')  
plt.xlabel('Winkel [°]')  
plt.ylabel('Zählrate' + r'[$\frac{1}{s}$]')  
plt.title('Röntgenspektrum NaCl')  
  
def linear(x, a, b):  
    return a * x + b  
popt4, pcov4 = curve_fit(linear, Winkel4[3:7], Rate4[3:7])  
plt.plot(Winkel4[2:10], linear(Winkel4[2:10], *popt4))  
nullstelle_NaCl = -popt4[1]/popt4[0]  
delta_nullstelle_NaCl = nullstelle_NaCl * np.sqrt((np.sqrt(pcov4[1][1])/  
    ↳ popt4[1])**2 + (np.sqrt(pcov4[0][0])/popt4[0])**2)  
print('Nullstelle NaCl = ', nullstelle_NaCl, '+/-', delta_nullstelle_NaCl)  
plt.savefig('Röntgenspektrum_NaCl_V255.pdf', format = 'pdf')  
plt.show()
```

Nullstelle NaCl = 3.2409744288091993 +/- 0.6644322912963466



[]: