In [164…

```python
#alle packages die ich verwenden werde, hat chatgpt gesagt

from uncertainties import ufloat
from uncertainties.umath import sqrt
from uncertainties import unumpy as unp
import matplotlib.pyplot as plt
import numpy as np
from uncertainties.umath import *
import pandas as pd
from uncertainties.umath import sin, radians #achtung in bogenmaß eingeben
from IPython.display import display, Latex
```

PW5 Wellenoptik

Beugungsbild hinter einem Einzelspalt

Aufgrund von Beugung tritt beim Durchtritt von monochromatischem Licht durch einen dünnen Einzelspalt eine Intensitätsverteilung auf einem inter dem Spalt liegenden Schirm auf. Hervorzuheben ist, dass der Abstand R des Schirms zum Hindernis (Spalt mit Spaltbreite a) folgende Relation erfüllt:

$$R = \frac{a^2}{\lambda}$$

Unter diesen Voraussetzungen spricht man von einer Fraunhofer- oder Fernfeldbeugung.

In vorgegebenem Experiment war die Spaltbreite a aus dem entstehenden Beugungsbild wie folgt zu ermitteln:

$$a\sin(\alpha_{\mathrm{min},n}) = n\lambda$$

wobei λ die konstante Wellenlänge 650 nm, $\alpha_{\mathrm{min},n}$ den Winkelabstand der Minima und n die Ordnung der jeweiligen Minima beschreiben. Mit der Kleinwinkelnäherung kann der Sinus von $\alpha_{\mathrm{min},n}$ mit $\alpha_{\mathrm{min},n}$ abgeschätzt werden. Das führt zur benutzten Gleichung

$$n\lambda = a\alpha_n$$

Durch Auftragen der von $n\lambda$ auf die Ordinate in Abhängigkeit vom Winkel α_n auf die Abzisse kann mithilfe eines linearen Fits die Spaltbreite a als Steigung der Fit-Funktion mit den dazugehörigen, automatisch generierten Unsicherheiten abgelesen werden. Aufgrund von Beugung tritt beim Durchtritt von monochromatischem Licht durch einen dünnen Einzelspalt eine Intensitätsverteilung auf einem hinter dem Spalt liegenden Schirm auf.

Hervorzuheben ist, dass der Abstand R des Schirms zum Hindernis (Spalt mit Spaltbreite a) folgende Relation erfüllt:

$$R = \frac{a^2}{\lambda}$$

Unter diesen Voraussetzungen spricht man von einer **Fraunhofer-** oder **Fernfeldbeugung**.

Im vorgegebenen Experiment war die Spaltbreite a aus dem entstehenden Beugungsbild wie folgt zu ermitteln:

$$a \sin(\alpha_{\min,n}) = n\lambda$$

wobei λ die konstante Wellenlänge von **650 nm**, $\alpha_{\min,n}$ den Winkelabstand der Minima und n die Ordnung der jeweiligen Minima beschreibt.

Mit der **Kleinwinkelnäherung** kann der Sinus von $\alpha_{\min,n}$ durch den Winkel selbst ersetzt werden.
Das führt zur verwendeten Gleichung:

$$n\lambda = a\alpha_n$$

Durch Auftragen von $n\lambda$ auf die **Ordinate** in Abhängigkeit vom Winkel α_n auf der **Abszisse** kann mithilfe eines linearen Fits die Spaltbreite a als **Steigung der Fit-Funktion** mit den dazugehörigen, automatisch generierten Unsicherheiten abgelesen werden.

In [165…

```
# konstante wellenlänge lambda
lambda_ = 635 # in nm

# unsicherheiten
us = 0.5 # in mrad, strahlendivergenz
um = 1 # in mrad, messunsicherheit - eintragen!!!!!

#abstände der minima zum mittelpunkt
abstände = unp.uarray([4, 8, 12, 16, 20, 24], um) # in mm
abstand_schirm_spalt = ufloat(935, um) # in mm

# Winkelberechnung in Radiant
winkel_rad = unp.arctan(abstände / abstand_schirm_spalt)

# Umrechnung in Grad
winkel_deg = winkel_rad * 180 / np.pi


# darstellung der werte als tabelle

messwerte_winkel = winkel_rad #liste der messwerte für n = 1 bis 6
n_mal_lambda = [i* lambda_ for i in range(1, 7)]

tabelle = {
    "nλ [nm]": n_mal_lambda,
    "α(n)": messwerte_winkel # darstellung!!
}

#pandas dataframe
df = pd.DataFrame(tabelle)

#tabelle anzeigen
display(df)
```

	nλ [nm]	α(n)
0	635	0.0043+/-0.0011
1	1270	0.0086+/-0.0011
2	1905	0.0128+/-0.0011
3	2540	0.0171+/-0.0011
4	3175	0.0214+/-0.0011
5	3810	0.0257+/-0.0011

In [166…

```
a_einzelspalt = ufloat(0.148, 0.062) # abgelesen aus scidavis, wert und unsicher
print(f"Die Spaltbreite beträgt {a_einzelspalt} mm. (siehe Abbildungsverzeichnis
```

```
Die Spaltbreite beträgt 0.15+/-0.06 mm. (siehe Abbildungsverzeichnis)
```

Beugungsmuster eines Doppelspalts

Analog zum vorhergehenden Experiment wird auch beim Doppelspalt die Spaltbreite bestimmt. Aus Symmetriegründen ist die Winkelbreite des zentral liegenden Maximums des Einzelspalts $\frac{2\lambda}{a}$. Wissend, dass der Winkelabstand zweier nebeneinanderliegender Minima $\frac{\lambda}{b}$ beträgt, können wir zählen wie oft dieser Abstand im Maximum 1. Ordnung liegt und aus dieser Relation den Spaltenabstand b bestimmen.

$$\frac{k\lambda}{b} = \frac{2\lambda}{a}$$

, wobei k die Anzahl der Minima des Doppelspalts innerhalb des Maximums des Einzelspalts ist. Daraus folgt $b = \frac{ka}{2}$. Nachdem das Abzählen der Minima nicht eindeutig möglich ist, da die Minima des Einzelspaltes und des Doppelspaltes sich nicht genau überlagern müssen, unterliegt k einer Unsicherheit von plus/minus 1.

In [167…

```
#alles analog zum beispiel voher

#minima beim doppelspalt
minima_doppelspalt = unp.uarray([1,1,1,1,1,1], um) # benutzen wir gar nicht

# anzahl minima 2. ordnung im hauptmaximum
k = ufloat(12, 1) # unsicherheit mindestens 1

b = ufloat(0.0951, 0.0040)  # spaltbreite bestimmt mit scidavis, in mm

um_ = 1 # in mrad, messunsicherheit - eintragen!!!!!

#abstände der minima zum mittelpunkt
abstände_ = unp.uarray([7, 13, 18, 25, 32, 38], um) # in mm
abstand_schirm_spalt_ = ufloat(935, um) # in mm

# Winkelberechnung in Radiant
winkel_rad_ = unp.arctan(abstände_ / abstand_schirm_spalt_)

# Umrechnung in Grad
winkel_deg_ = winkel_rad_ * 180 / np.pi
```

```
# darstellung der werte als tabelle

messwerte_winkel_ = winkel_rad_ #liste der messwerte für n = 1 bis 6
n_mal_lambda = [i* lambda_ for i in range(1, 7)]

tabelle_ = {
    "nλ [nm]": n_mal_lambda,
    "α(n)": messwerte_winkel_ # darstellung!!
}

#pandas dataframe
df_ = pd.DataFrame(tabelle_)

#tabelle anzeigen
display(df_)

g = (k*b)/2

print(f"Die Spaltbreite beträgt {b} mm. Der Spaltenabstand beträgt {g} mm.(siehe
```

	nλ [nm]	α(n)
0	635	0.0075+/-0.0011
1	1270	0.0139+/-0.0011
2	1905	0.0192+/-0.0011
3	2540	0.0267+/-0.0011
4	3175	0.0342+/-0.0011
5	3810	0.0406+/-0.0011

```
Die Spaltbreite beträgt 0.095+/-0.004 mm. Der Spaltenabstand beträgt 0.57+/-0.05
mm.(siehe Abbildungsverzeichnis)
```

Beugungsgitter

Mittels eines Beugungsgitters kann das Beugungsmuster einer Spektrallampe erzeugt werden. Auch hier kann man die Beugungswinkel der einzelnen Spektrallinien messen und mithilfe der Gittergleichung

$$\sin \alpha_{\max,k} = \frac{k\lambda}{g}$$

werden die Wellenlängen berechnet. g ist hier die Gitterkonstante und k die Ordnung der Beugungsmaxima. Nach dem Vergleich mit Literaturwerten (Quelle) lässt sich sagen, dass es sich bei dem untersuchten Gas vermutlich um ... handelt.

In [169…

```
# beugungswinkel messen - achtung auf einheit!

g = 1/(140) # gitterkonstante in mm

u_alpha = unp.radians(1/3)  #was auch immer, messunsicherheit winkel
alpha_1_grün = unp.radians(ufloat(((358 + 2/3)-349+0.925)/2, u_alpha)) # beugung
```

```
alpha_2_grün = unp.radians(ufloat(((360+3+0.074)-(345+5/12))/2, u_alpha)) # beug
alpha_3_grün = unp.radians(ufloat(((360+7+5/12)-(341+4/15))/2, u_alpha)) # beugu

alpha_1_orange = unp.radians(ufloat(((358+55/60)-(359+2/3))/2, u_alpha)) # beugu
alpha_2_orange = unp.radians(ufloat(((360+3+35.5/60)-(345+12/60))/2, u_alpha)) #
alpha_3_orange = unp.radians(ufloat(((360+8+19.5/6)-(340+27/60))/2, u_alpha)) #

alpha_1_violett = unp.radians(ufloat(((357+1/3)-(347+19/60))/2, u_alpha)) # beug
alpha_2_violett = unp.radians(ufloat(((360+55/60)-(343+54/60))/2, u_alpha)) # be
alpha_3_violett = unp.radians(ufloat(((360+4+45/60)-(340+1/3))/2, u_alpha)) # be

lambda_1_grün = ((unp.sin(alpha_1_grün))*g)/1 # 1. ordnung
lambda_2_grün = ((unp.sin(alpha_2_grün))*g)/2 # 2. ordnung
lambda_3_grün = ((unp.sin(alpha_3_grün))*g)/3 # 3. ordnung
lambda_grün = ((lambda_1_grün + lambda_2_grün + lambda_3_grün)/3)

lambda_1_orange = ((unp.sin(alpha_1_orange))*g)/1 # 1. ordnung
lambda_2_orange = ((unp.sin(alpha_2_orange))*g)/2 # 2. ordnung
lambda_3_orange = ((unp.sin(alpha_3_orange))*g)/3 # 3. ordnung
lambda_orange = ((lambda_1_orange + lambda_2_orange + lambda_3_orange)/3)

lambda_1_violett = ((unp.sin(alpha_1_violett))*g)/1 # 1. ordnung
lambda_2_violett = ((unp.sin(alpha_2_violett))*g)/2 # 2. ordnung
lambda_3_violett = ((unp.sin(alpha_3_violett))*g)/3 # 3. ordnung
lambda_violett = ((lambda_1_violett + lambda_2_violett + lambda_3_violett)/3)
```

Diskussion

Beim Doppelspalt war es schwierig, die Minima 1. Klasse zu erkennen, daher müsste die Unsicherheit etwas erhöht werden. Beim Experiment mit dem Beugungsgitter dürften die Werte im Code falsch eingetragen sein, weswegen unzureichende Informationen zur eindeutigen Bestimmung eines Elements vorliegen.

Abbildungsverzeichnis

Aus Zeitgründen werden die Ergebnisse der linearen Fits zur Bestimmung der Spaltbreite bei Einzel- und Doppelspalt hier in Form eines Screenshots mit all seinen Informationen eingefügt.