

Physikalisches Anfängerpraktikum II

Sommersemester 2023

Versuch 233

Tutor: Felix Exner

Fourieroptik

1 Einleiung¹

1.1 Ziel des Versuchs

Das Ziel dieses Experiments ist es, das Beugungsverhalten verschiedener Beugungsobjekte (Fouriersynthese) bei Manipulationen in der Fourierebene im Zusammenhang mit der Fourier-Optik zu untersuchen und sowohl qualitativ als auch quantitativ die Beugungsmuster und Objektbilder eines Einzelpaltes sowie eines Doppelpaltes zu analysieren, wobei verschiedene Beugungsordnungen berücksichtigt werden sollen. Dabei soll auch ein Vergleich der gemessenen Strukturen mit den theoretisch zu erwartenden Intensitätsprofilen durchgeführt werden.

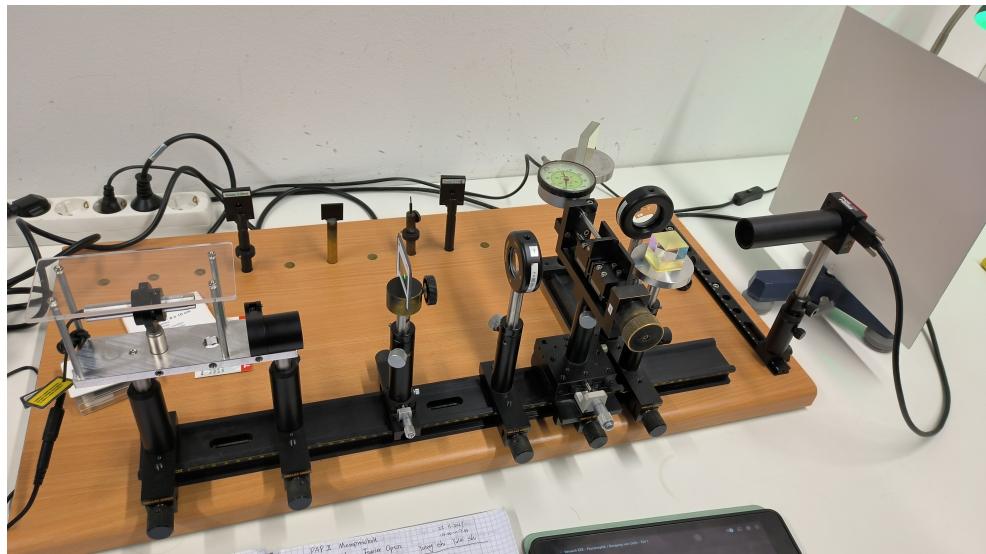


Abbildung 1: Versuchsaufbau

1.2 Fourierreihen und Fourierintegrale

Ähnlich der Approximation einer Taylorreihe um einen spezifischen x-Wert mittels eines Basissystems x^n kann auch eine periodische Funktion durch eine Linearkombination trigonometrischer Basisfunktionen in einer Fourierreihe entwickelt werden.

¹Dr. J.Wagner - Physikalisches Anfängerpraktikum - V. 2.0β Stand 04/2023 - Python Edition

Für eine periodische Funktion $f(x)$ mit der Periode L lässt sich die Fourierdarstellung schreiben als:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{2\pi n}{L}x\right) + b_n \sin\left(\frac{2\pi n}{L}x\right) \quad (1)$$

$$a_n = \frac{2}{L} \int_{-L/2}^{L/2} f(x) \cos\left(\frac{2\pi n}{L}x\right) dx \quad (2)$$

$$b_n = \frac{2}{L} \int_{-L/2}^{L/2} f(x) \sin\left(\frac{2\pi n}{L}x\right) dx \quad (3)$$

(4)

a_n , b_n bezeichnen sich als die Fourierkoeffizienten. Für zeitabhängige Funktionenersetzen wir in Gleichung (1) die Variable x durch die Zeit t und wählen für die Periode L , die Periodendauer $T = 2\pi/\omega$, dadurch bekommen wir die Überlagerung von Sinus- und Kosinusfunktionen mit unterschiedlichen Frequenzen und Amplituden:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + b_n \sin(n\omega t) \quad (5)$$

Die Fourieranalyse liefert Informationen über das Frequenzspektrum, welches die Bestandteile eines zeitperiodischen Signals darstellt:

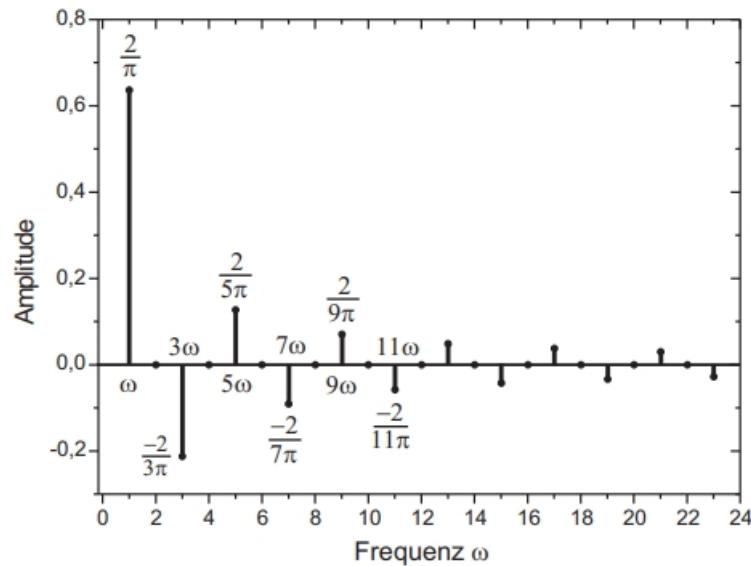


Abbildung 2: Frequenzspektrum eines Rechtecksignals

Außerdem ist die Fourierdarstellung für eine nicht periodische Funktion ($T \rightarrow \infty$) möglich. Wir werden in diesem Experiment speziell auf die Rechteckfunktion mit der Periode L eingehen:

$$f(x) = \begin{cases} 1 & -l/2 < x < l/2 \\ 0 & l/2 > |x| > L/2 \end{cases} \quad (6)$$

Hier dürfen wir nur die Koeffizienten a_n berechnen, da die Rechteckfunktion gerade ist, fallen somit alle Sinus Terme in der Fourierentwicklung aus:

$$a_0 = \frac{2l}{L} \quad a_n = \frac{2}{L} \int_{-l/2}^{l/2} \cos\left(\frac{2\pi n}{L}x\right) dx = \frac{2}{\pi n} \sin\left(\pi n \frac{l}{L}\right) \quad (7)$$

$$f(x) = \frac{1}{2} + \frac{2}{\pi} \cos\left(\frac{2\pi}{L}x\right) - \frac{2}{3\pi} \cos\left(\frac{6\pi}{L}x\right) + \frac{2}{5\pi} \cos\left(\frac{10\pi}{L}x\right) \mp \dots \quad (8)$$

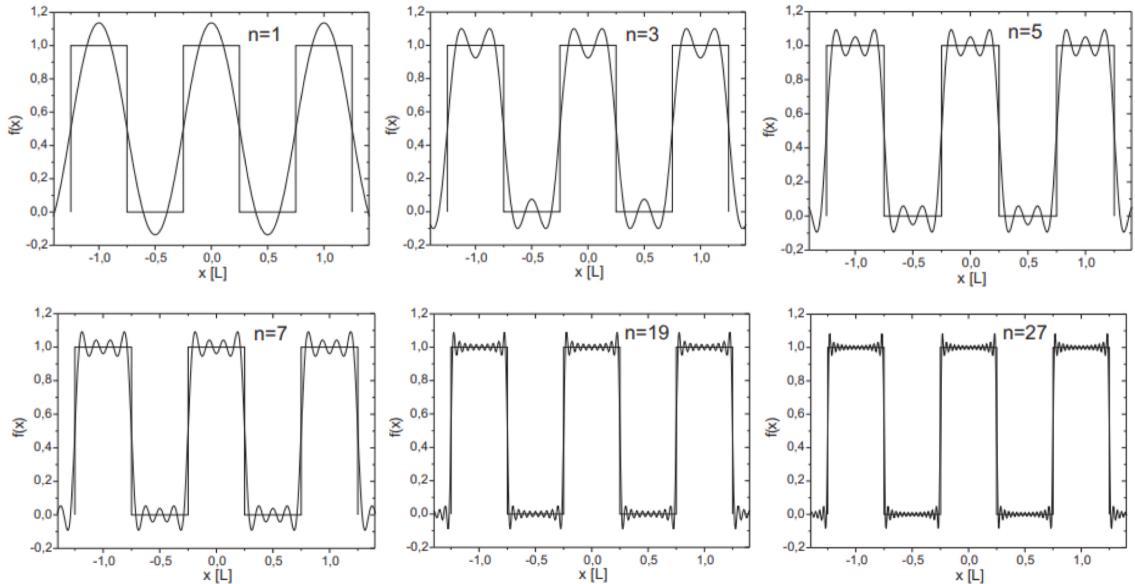


Abbildung 3: Fourierentwicklung eines Rechtecksignals

Bei $L \rightarrow \infty$ steigt die Modenzahl in Abbildung (2) gegen unendlich und wird dicht gepackt. Die Fourierreihe wird zu einem Integral, wodurch die Koeffizienten kontinuierlich und nicht mehr diskret sind (Fouriertransformation):

$$f(x) = \int_{-\infty}^{+\infty} F(k) e^{ikx} dx \quad F(k) = \int_{-\infty}^{+\infty} f(x) e^{-ikx} dx \quad (9)$$

(10)

$F(k)$ bezeichnet sich als die **Fouriertransformierte** von $f(x)$, für den Fall dass f zeitabhängig ist stellt die Fouriertransformierte das kontinuierliche Frequenzspektrum dar.

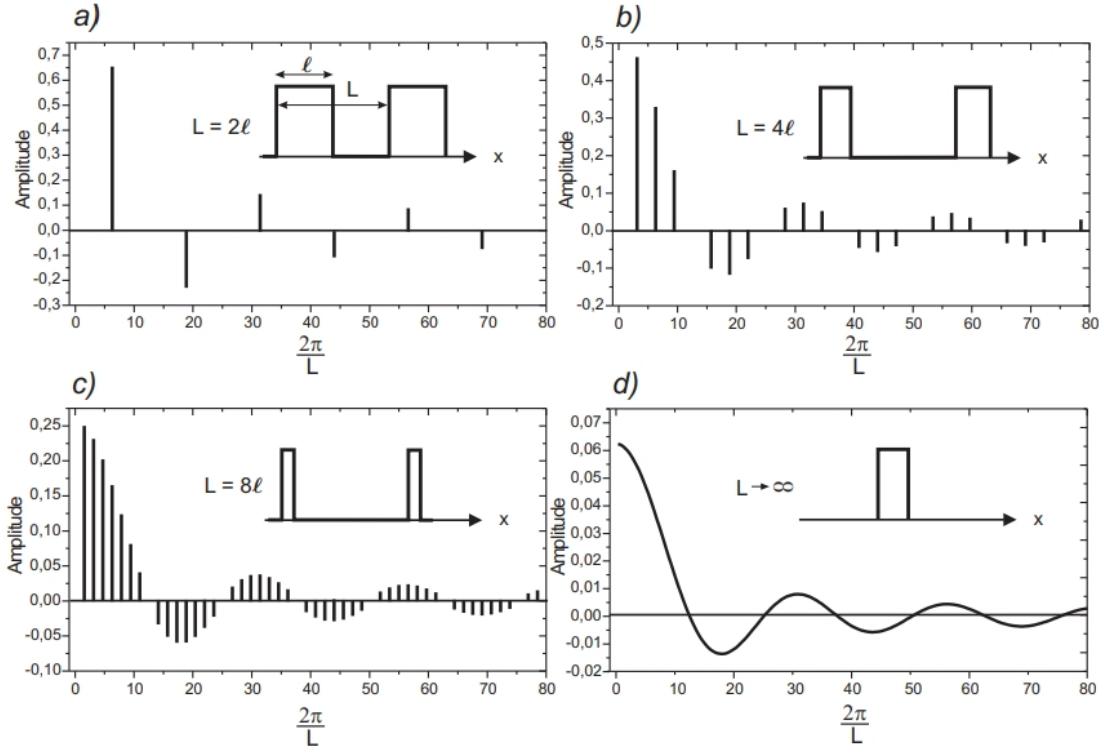


Abbildung 4: Übergang zur Fouriertransformation einer nichtperiodischen Funktion

1.3 Die Fresnelsche und Fraunhofersche Beugung

Die **Fresnel'sche Beugung** beschreibt die Lichtbeugung unter verschiedenen Winkeln an einem Objekt mit endlichem Abstand zur Lichtquelle und Beobachtungsebene, die schließlich auf der Beobachtungsebene interferieren.

Bei der **Fraunhoferschen Beugung** befindet sich die Lichtquelle im Unendlichen daher treffen die Strahlen annähernd parallel auf das beugende Objekt und interferieren auch im Unendlichen. Um die Interferenzerscheinungen in einem endlichen Abstand beobachten zu können, wird eine Sammellinse hinter dem beugenden Objekt montiert. Die Beugungsstrukturen befinden sich dann in der Brennebene.

Wird nun ein Spalt mit Breite d und sehr großer Länge (1D Problem) von einem parallelen und monochromatischen Lichtstrahl der Wellenlänge λ beleuchtet, sollen alle Punkte des Spaltes mit gleicher Amplitude E_0 und Phase $\varphi = \omega t$ angeregt werden. Nach dem Huygens-Fermat'schen Prinzip geht von jedem Punkt eines Spalts Elementarwellen aus, deren Überlagerung zu einer Intensitätsverteilung führt. Es erfordert deswegen die Analyse aller parallelen Teilbündel in einer bestimmten Richtung α :

$$E(\text{Spalt}) = E(y) = E_0 e^{i\omega t} \quad (11)$$

$$(12)$$

$$E_\infty(\alpha) = \int_{-\frac{d}{2}}^{\frac{d}{2}} E_0 e^{i(\omega t - kl)} dy \quad (13)$$

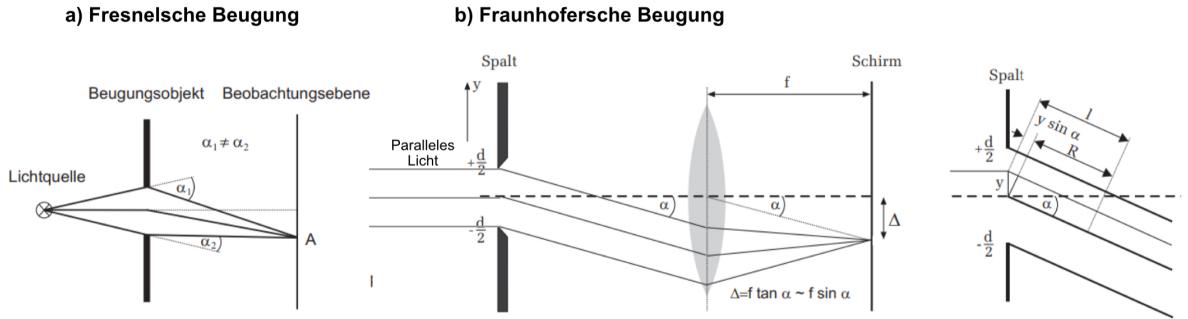


Abbildung 5: Fresnelsche Beugung und Fraunhofersche Beugung am Spalt

wobei $k = 2\pi/\lambda$ der Betrag des Wellenvektors und l die Weglänge sind. Der Nullpunkt liege in der Mitte des Spaltes. Somit ergibt sich ein Gangunterschied bei y ausgehendes Lichtbündel gegenüber einem vom Mittelpunkt ausgehenden Lichtbündel $y \sin \alpha$. Die Weglänge gilt dann:

$$l = R + y \sin \alpha \quad (14)$$

Durch Einsetzen in Gleichung (13) und Integration erhält man mithilfe von Eulerscher Formel $e^{\pm i\delta} = \cos \delta \pm i \sin \delta$:

$$E_\infty(x) = E_0 e^{i(\omega t - kR)} \frac{\sin x}{x} d \quad x := \frac{d}{\lambda} \pi \sin \alpha \quad (15)$$

Für die Intensität gilt damit:

$$I_\infty(x) = E_\infty(x)^2 \propto \frac{\sin^2 x}{x^2} d^2 \propto I_0 \frac{\sin^2 x}{x^2} \quad (16)$$

1.4 Herleitung der Fourierdarstellung der Fraunhoferschen Beugung

Um allgemein die elektrische Feldstärke in einem mit monochromatischem Licht bestrahlten Punkt P zu bestimmen wird eine willkürliche Öffnung S in der yz -Ebene betrachtet. Dabei wird eine Kugelwelle (Elementarwelle) $\frac{e^{ikr}}{r}$ mit konstanter Quellstärke pro Einheitsfläche ϵ , die von einem kleinen Flächenelement $dS(x = 0, y, z)$ ausgeht, analysiert. Für die elektrische Feldstärke im Punkt P gilt also:

$$dE = \epsilon \frac{e^{ikr}}{r} dS \quad (17)$$

Für den Abstand r zwischen Flächenelement dS und Punkt $P(X, Y, Z)$ gilt: $r = \sqrt{X^2 + (Y - y)^2 + (Z - z)^2}$ und den Abstand R zwischen Ursprung und Punkt P : $R = \sqrt{X^2 + Y^2 + Z^2}$, durch Einsetzen bekommen wir:

$$r = R\sqrt{1 + (y^2 + z^2)/R^2 - 2(Yy + Zz)/R^2} \quad (18)$$

Im Fernfeld ist R groß genug um die Näherung $(y^2 + z^2)/R^2 \ll 1$ anzunehmen, so dass wir zusätzlich mit einer Taylornäherung $\sqrt{1 - \xi} \approx 1 - \xi/2$ den Abstand r vereinfachen können:

$$r = R[1 - (Yy + Zz)/R^2] \quad (19)$$

Durch Umformen und Einsetzen in Gleichung (17) und Integrieren über die gesamte Öffnung erhält man für die elektrische Feldstärke am Punkt P :

$$E(R) = \epsilon \frac{e^{ikr}}{R} \iint e^{-\frac{ik}{R}(Yy+Zz)} dy dz \quad (20)$$

Betrachten wir den Vorfaktor des Integrals, so stellen wir fest, dass er in der Tat keine Konstante dargestellt hat, sondern wegen der Inhomogenität der Transmission eine Orts- und Phasenabhängigkeit aufweist:

$$\epsilon(y, z) = A(y, z) = A_0(y, z)e^{i\phi(y, z)} \quad (21)$$

Die Öffnungsfunktion $A(y, z)$ ist proportional zum elektrischen Feld einer Welle, die von einem Flächenelement $dy dz$ ausgeht. Nun wird das elektrische Feld bei (X, Y, Z) ohne Berücksichtigung der Konstante $\frac{e^{ikr}}{R}$ wie folgt dargestellt:

$$E(Y, Z) = \iint_S A(y, z) e^{-\frac{ik}{R}(Yy+Zz)} dy dz \quad (22)$$

Das Differential dE stellt den Beitrag der ebenen Welle, die sich in Richtung des Wellenvektors \vec{k} ausbreitet und vom Flächenelement $dy dz$ ausgeht, zum elektrischen Feld am Ort P dar:

$$dE(Y, Z) = A(y, z) e^{-\frac{ik}{R}(Yy+Zz)} dy dz \quad (23)$$

Aus Abbildung 6 definieren wir die Raumfrequenzen k_x, k_y , welche sich im Photonenbild nur um einen Faktor \hbar von den Transversalimpulsen p_i der Photonen unterscheiden:

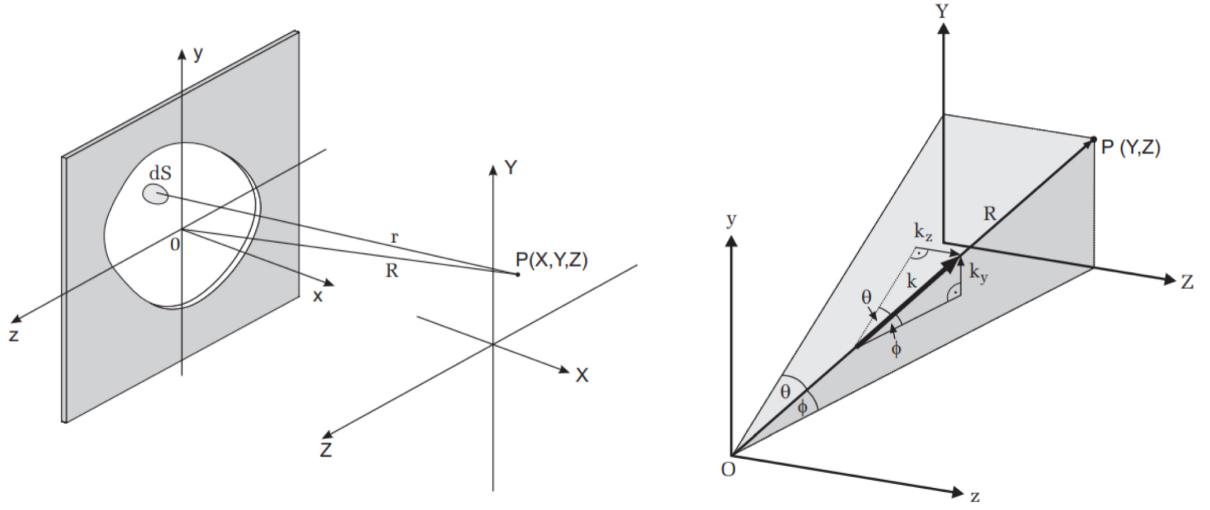


Abbildung 6: Links: Beugung an einer beliebigen Öffnung, rechts: Definition Raumfrequenz

$$p_y = \hbar k_y = \hbar k \frac{Y}{R} = \hbar k \sin \phi \quad (24)$$

$$p_z = \hbar k_z = \hbar k \frac{Z}{R} = \hbar k \sin \theta \quad (25)$$

Wenn die Photonen also nur einen ursprünglichen Impuls in x-Richtung haben, so haben sie nach Durchlaufen der Öffnung auch eine transversale Komponente in y- und z-Richtung. Wenn man jeden Punkt (Y, Z) in der Bildebene eine Raumfrequenz zuordnet, dann folgt für Gleichung (22):

$$E(k_y, k_z) = \int A(y, z) e^{-i(k_y y + k_z z)} dy dz \quad (26)$$

Dies ist genau die zweidimensionale Fouriertransformation der Öffnungsfunktion $A(y, z)$, woraus man schließen kann, dass die Feldverteilung der Beugungsstruktur bei der Fraunhoferschen Beugung die Fouriertransformierte der Feldverteilung über die beugende Öffnung ist.

1.5 Beugungsbild des Spalts als Fouriertransformierte der Spaltöffnung

Um die Beugung am Spalt im Rahmen der Fouriertheorie zu behandeln betrachten wir die folgende Spaltfunktion:

$$f(y) = \begin{cases} 1 & |y| \leq d/2 \\ 0 & |y| > d/2 \end{cases} \quad (27)$$

wobei d die Breite der Spaltöffnung ist. Durch die Berechnung der Fouriertransformierten, Einsetzen der Spaltfunktion und Verwendung der Eulerschen Formel erhält man:

$$E(k_y) = F(k_y) = \int_{-\infty}^{+\infty} f(y) e^{-ik_y y} dy = d \frac{\sin(k_y d/2)}{k_y d/2} = \text{sinc}\left(\frac{k_y d}{2}\right) d \quad (28)$$

mit Nullstellen $k_y = \frac{2\pi n}{d}$. Durch eine Rücktransformation der Fouriertransformierten (Fouriersynthese) erhält man:

$$f(y) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(k_y) e^{ik_y y} dk_y = \frac{d}{\pi} \int_{-\infty}^0 \text{sinc}\left(\frac{k_y d}{2}\right) \cos(k_y y) dk_y \quad (29)$$

Durch numerische Integration erhält man wieder das rechteckige Spaltbild.

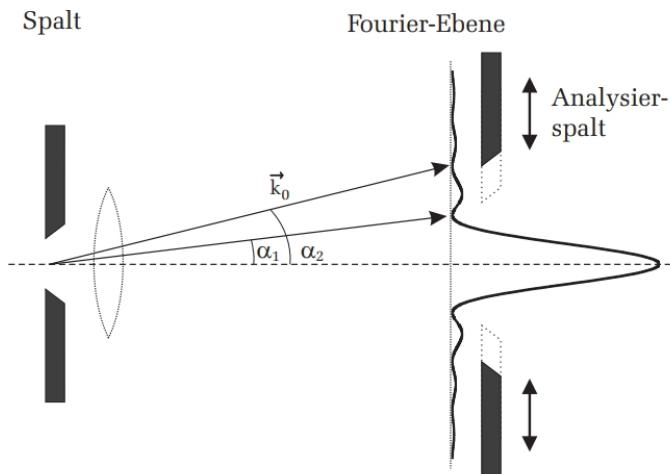


Abbildung 7: Manipulation der Beugungsstruktur durch Ausblenden von Beugungsordnungen mit Hilfe eines Analysierspaltes

Die Intensität in Richtung α kann aus dem Quadrat von dem Ergebnis in Gleichung (28) durch einsetzen der Beziehung $k_y = k_0 \sin \alpha$ berechnet werden. Durch den Analysierspalt können in der Fourierebene eine gewisse Anzahl an gebeugten Teilstrahlen ausgeblendet werden. Um daraus die rudimentäre Spaltfunktion zu bestimmen, wird als obere Integrationsgrenze der $k_{y,n}$ -Wert gewählt, der gerade noch so zur Abbildung zugelassen wird. Diese Integrationsgrenze ist die n -te Nullstelle von $F(k_y)$:

$$k_{y,n} = k_0 \sin \alpha_n = \frac{k_0 n \lambda}{d} = \frac{2\pi n}{d} \quad (30)$$

Daraus folgt für Gleichung (29):

$$f_{mod}(y) = \frac{d}{\pi} \int_0^{k_{y,n}} \text{sinc}\left(\frac{k_y d}{2}\right) \cos(k_y y) dk_y \quad (31)$$

Um die Berechnungen mit den beobachteten Spaltbildern vergleichen zu können, werden die Ergebnisse quadriert, da man mit dem Integral die Feldstärke berechnet und diese quadriert verwenden muss, um die Intensität zu erhalten.

1.6 Fouriertransformierte des Doppelspalts

Analoge Berechnungen lassen sich für den Doppelspalt machen indem man einen nach rechts verschobenen und einen nach links verschobenen Einzelpunkt betrachtet und die beiden Integrale später addiert. Daraus ergibt sich:

$$F(k_y, rechts) = \int_3^4 f(y) e^{ik_y y} dy = -\frac{1}{ik_y} e^{-ik_y} \Big|_3^4 = e^{-ik_y g/2} \text{sinc}\left(\frac{k_y d}{2}\right) d \quad (32)$$

$$F(k_y, links) = \int_1^2 f(y) e^{ik_y y} dy = -\frac{1}{ik_y} e^{-ik_y} \Big|_1^2 = e^{ik_y g/2} \text{sinc}\left(\frac{k_y d}{2}\right) d \quad (33)$$

$$F = F(k_y, rechts) + F(k_y, links) = 2 \cos\left(\frac{k_y g}{2}\right) \text{sinc}\left(\frac{k_y d}{2}\right) d \quad (34)$$

wobei g den Spaltabstand beschreibt. Nun setzt sich diese Fouriertransformierte aus dem Term der Fouriertransformierte des Einzelpaltes sowie dem Interferenzvorfaktor zusammen, mit der wir durch Einsetzen $k_y = k_0 \sin \alpha = \frac{2\pi}{\lambda} \sin \alpha$ und Quadrieren die Beugungsfigur des Doppelpaltes bekommen als:

$$I = 4 \cos^2\left(\frac{\pi g}{\lambda} \sin \alpha\right) d^2 \text{sinc}^2\left(\frac{\pi d}{\lambda} \sin \alpha\right) \quad (35)$$

Dies entspricht dem Produkt der Gitterfunktion $\cos^2\left(\frac{\pi g}{\lambda} \sin \alpha\right)$ und der Spaltfunktion des Einzelpaltes (siehe Gl.(28)).

Mit dem Analysierspalt können ähnlich wie das Einzelpunkt das "modifizierte" Bild F_{mod} des Doppelpaltes ermittelt werden:

$$F_{mod}(y) \propto [f_{mod}(y)]^2 = \left[\frac{2d}{\pi} \int_0^{k_{y,n}} \cos\left(\frac{k_y g}{2}\right) \text{sinc}\left(\frac{k_y d}{2}\right) \cos(k_y y) dk_y \right]^2 \quad (36)$$

Mit $k_{y,n} = 2\pi n/d$ kann man die Spaltbilder für eine bestimmte Anzahl an Maxima und Minima bestimmen. Allerdings verschwindet die Doppelspaltstruktur im modifizierten Bild ab einer bestimmten Spaltweite des Analysierspalts. Für die Auflösung der Doppelstruktur ist daher der Spaltabstand g maßgebend, um die Doppelstruktur noch zu erkennen muss das Maximum 1. Ordnung bei $\sin \alpha = \lambda/g$ also mindestens noch zur Abbildung zugelassen werden, damit die charakteristische Struktur zu erkennen ist.

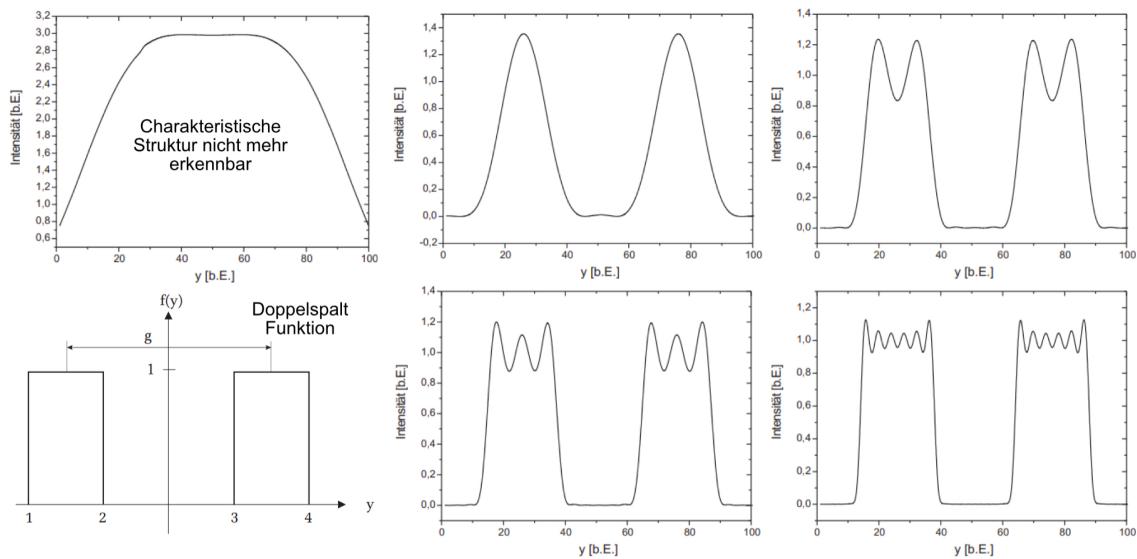


Abbildung 8: Rechts: Spaltbild verschiedener Beugungsordnungen, links: Spaltfunktion

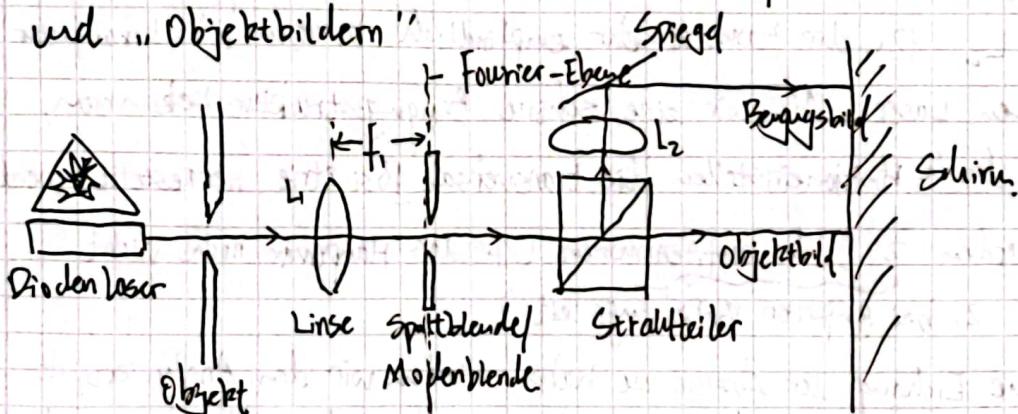
2 Versuchsdurchführung

2.1 Versuchsaufbau, Versuchsdurchführung und Messprotokoll

Siehe folgende Seiten.

Messaufbau:

- Diodenlaser 532 nm
- Ein Satz Beugungsobjekte: Spalte, Doppelspalt, Gitter, Kreuzgitter, Spalt mit Gauß-förmigem Transmissionsprofil
- Verschiedene sphärische Linsen
- Strahlteiler • Graufilter • Spiegel
- verschiedene Fert-Reiter • Feinjustierbare Verschiebereiter
- Symmetrisch öffnender Analysierspalt
- Verschiedene schmale Metallstreifen als Modenblenden
- CMOS-Kamera mit PC und Monitor zur Aufnahme von Beugungsbildern und „Objektbildern“



Lichtquelle Diodenlaser $\lambda = 532 \text{ nm}$, Leistung $P \leq 1 \text{ mW}$
 Linse L_1 mit Brennweite $f_1 = 80 \text{ mm}$

Wir halten ein Papierzettelchen hinein und überzeugen uns die Beugungsstrukturen entstehen zuerst in der Brennebene von L_1 . L_2 muss so positioniert werden, s.d. sie über den nachfolgenden Spiegel, der weiterlaufenden Strahl wieder parallel zu erstem Strahl ist, die Beugungsstruktur stark vergrößert neben dem Bild des Objektes, auf den Schirm entwirft. Wir stellen das Bild der Beugungsstruktur scharf durch feine Verschiebungen der L_2 .

Wir montieren und justieren den symm. offenen Spalt und beobachten, wie sich das Objektbild ändert während wir die hohen Beugungsordnungen ausblenden. (Tiefpass & Hochpass)



Einfachspalt: kleine Spalt: Beugungsbild.

unterscheiden sich Spalt: Beugungsbild:
von Spaltbreite
großes Spalt: Beugungsbild

Falls wir mit Metallplättchen die 0. Ordnung entfernen, geht die Informationen der Füllung weg, wir können nur die Kante sehen.

1. Quantitative Beobachtungen am Einfachspalt

Wir bilden nun das Beugungsbild auf den Kamerasensor ab, dabei benutzen wir PC-Programm "ThorCam". Wir schieben zunächst die Kamera bisschen zur Seite und schauen uns das Beugungsbild auf einem daneben gehaltenen Blatt Papier an. Zum Schärfstellen verschieben wir L_2 . Da die Kamera sehr empfindlich ist, benutzen wir Geraufilter vor dem Laser. Als wir eine schöne Beugungsstruktur bekommen, sollen durch Positioneinstellen die optische Symmetrie hergestellt werden.

Es werden 2 Bilder aufgenommen (1. das Hauptmax noch nicht sichtig, 2. bei größerer Belichtungszeit.)

Um die Eichung der Abzisse zu haben stellen wir den Analysierspalt in Fourierebene ein und die zugehörigen Abstände in Pixel auf den Monitor ablesen.

Tabelle 1: Messung der Minimum bei Einfachspalt \Rightarrow Abstand zwischen Minimum / Maxima

Aus Programm	Pixel	769.6	998.0	549.8	1148.7	374.8			
$\Delta P = 5.5$									
	Intensität	62.786	3.361	3.241	1.104	0.637			

#Pixel	358	527	701	854	1027	(Eichung der Abzisse)
Messuhr [mm]	1.783	1.375	1.048	0.700	0.352	

$a_0 = 2 \cdot 215$
 $\times 2$ für den Spalt abseitig



扫描全能王 创建

2. Beugungsstruktur des Doppelspaltes

Wir ersetzen den Einzelspalt durch Doppelspalt ohne den Reiter zu verschieben und verschaffen wieder ein scharfes Bild

der Beugungsstruktur

Skizze der Struktur:

unterscheiden sich von

Spaltabstand



⇒ Je breiter der Spalte,
desto unger wird das
Beugungsmuster.

- Wir führen weiter die quantitative Messung mit Spalt = 8'' analog wie 1. Einzelspalt und nehmen auch Bilder auf.
Bei Eichung der Abszisse wird lediglich das Rahmen mit dem Einzelspalt gegen den Doppelspalt ausgetauscht. Gegenstands- & Bildweite unverändert.

Tabelle 2: Messung Max und Min an Doppelspalt

Pixel.							
Messuhr [nm].							

+-----+-----+-----+

3. Objektbild als Fouriersynthese des Beugungsbildes

Wir untersuchen nun die Auswirkung der Manipulationen in der Fourierebene auf die Struktur des Bildes quantitativ:

Wir öffnen den Analysierspalt in der FE und stellen die Laserintensität auf Max ein und das Beugungsbild möglichst groß gebildet (Schirm entfernt und L₂ schaufenstern)

Wir bestimmen die Intensitätsverteilung wieder. Um "Wiggles" der Rechteckfunktion zu entfernen justieren wir nach, dazu auf keinen Fall L₁ verschieben. Wir verschieben den Spalt feinfühlig um auf



größte Steilheit der Kurven einzustellen. Danach wird die Belichtungszeit eingestellt.

Wir drehen den Analysierspalt langsam zu und beobachten simultan die Veränderung der Baugangsstruktur auf dem Schirm und des Spaltbildes auf dem Monitor und beginnend nach der Reihe nach bei größerer werdender Öffnung des Spaltes die charakteristischen Bildstrukturen auf, die sich einstellen, wenn wir der Reihe nach genau die 0-te, ± 1 -te, ± 2 -te Ordnung etc. zur Abbildung zulassen. Es werden von den 1-5 Min. Bilder gemacht, und bei den ersten 3 die Intensität I und L gemessen. 4 Bilder insgesamt aufzunehmen.

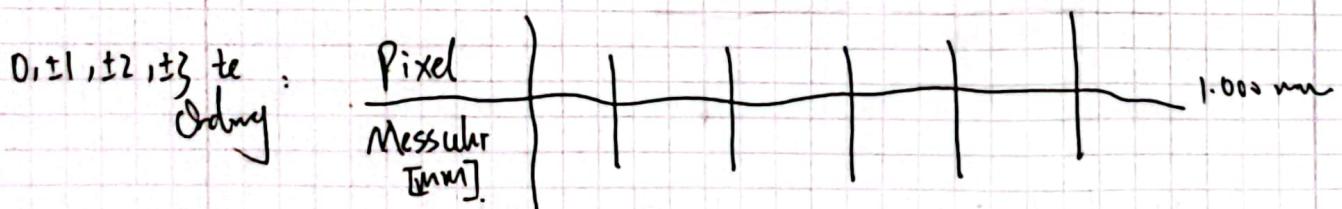
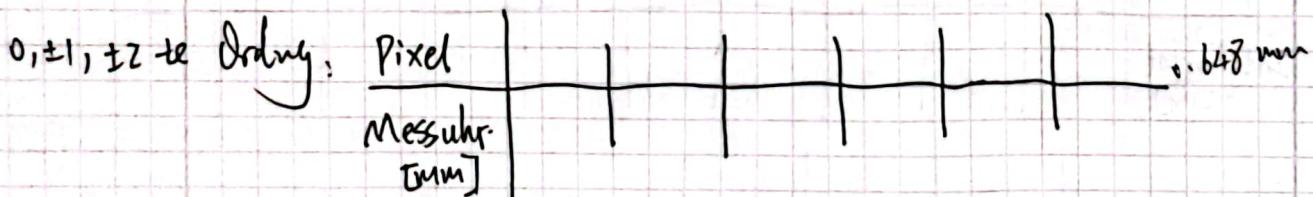
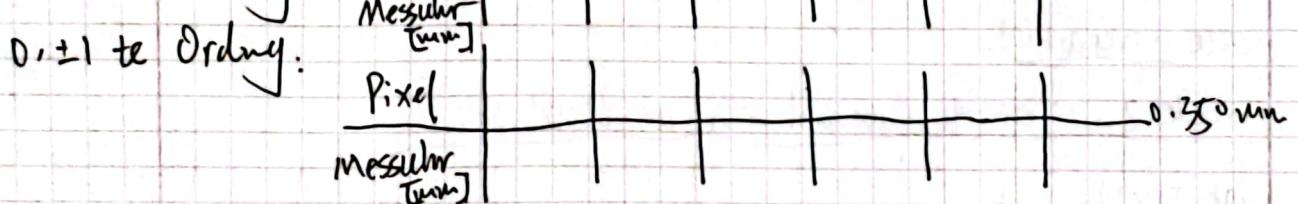
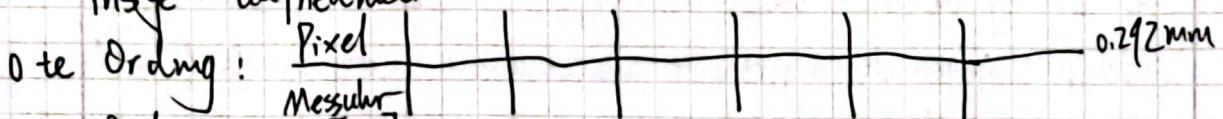


Tabelle 3: charakteristische Bildstruktur Einzelspalt

Spaltbreite des Einzelspalts: (226 ± 20) Pixel.

wir nehmen zusätzlich ein Bild bei weit geöffnetem Analysierspalt auf und vermessen das Intensitätsprofil in Gwyddion und bestimmen die Spaltbreite in Pixel. (Ein Pixel = 3.45 μm)

Vergleich zu bestimmt. (Gegenstandsweite $f_0 = (10.0 \pm 0.8)$ cm Bildweite $b = (8.0 \pm 0.5)$ cm $f_1 = 85$ nm.



扫描全能王 创建

4. Fourierbild des Doppelspaltes

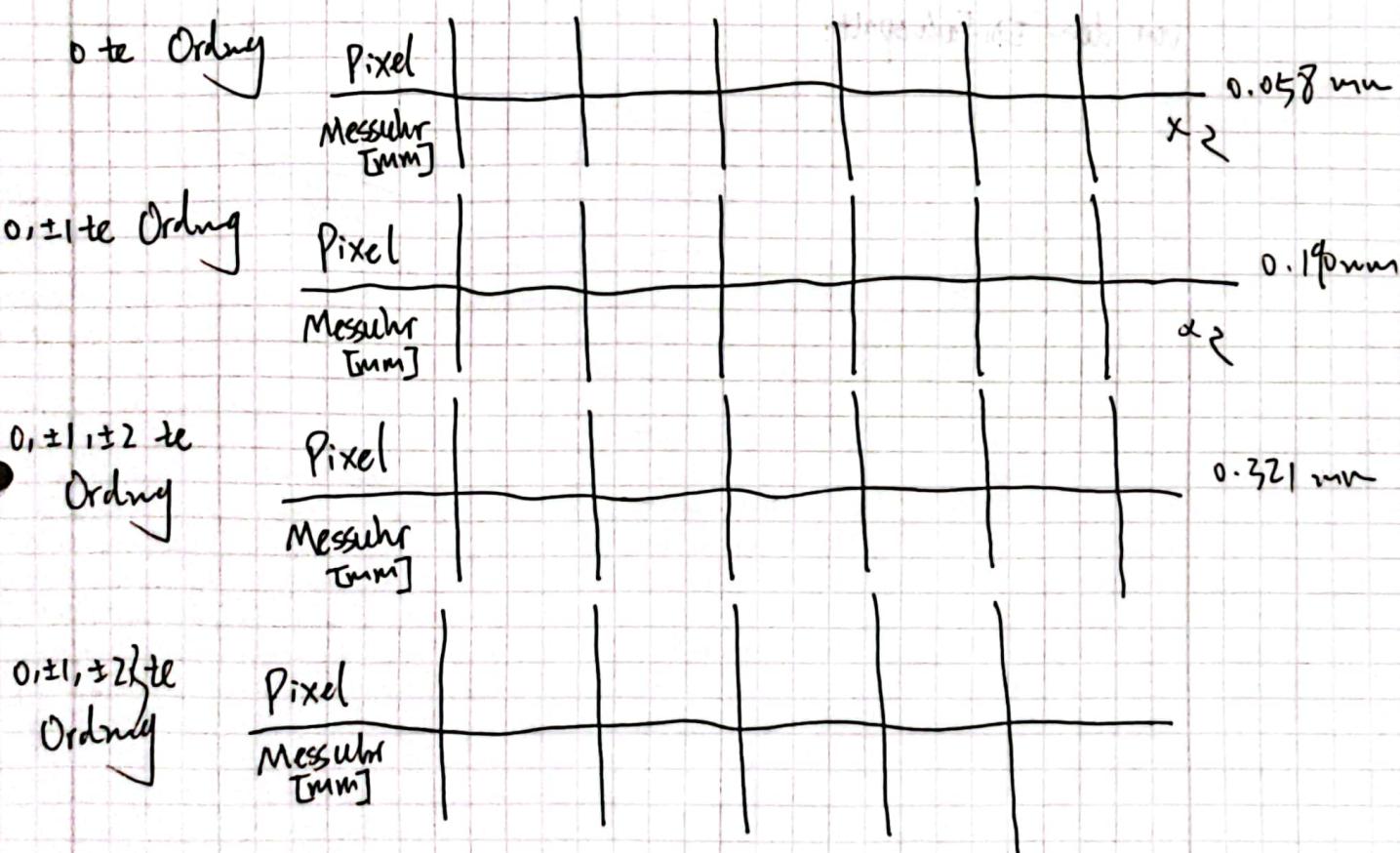
Wir schauen uns das Bild des Doppelspaltes mit der Kamera an. Die Breite und Abstände der beiden Einzelspalte gemessen: (Abb. 14)

$$\text{Spaltabstand } g = (626 \pm 20) \text{ Pixel. Spaltbreite. } d = (258.2 \pm 20) \text{ Pixel}$$

Wie beim 3. Messung am Einzelspalt bestimmen wir die Vergrößerung aus aktueller Bildweite und Brennweite. Bildweite $b' = (38.0 \pm 0.5) \text{ cm}$. $g' = (11.0 \pm 0.5) \text{ cm}$.

wir beobachten im Folgenden die Beiträge der einzelnen Beugungsordnungen der Spalte, indem wir den Analysierspalt in der Fourierebene auf und zudrehen:

Tabelle 4: Charakteristische Bildstruktur Doppelspalt



wir drehen den Analysierspalt langsam zu und beobachten, wie die beiden rechteckigen Spaltprofile zu 2 gaußförmigen Profilen verschwinden., wie sich zunehmend auch der Zwischenraum aufhellt und schließlich die Doppelstruktur



扫描全能王 创建

verschwindet und nur ein flaches Plateau

Spaltbreite des Analysierspalts $d_s = \underline{(0.052 \pm 0.008) \text{ mm}}$

Auflösung eines Kreuzgitters:

schließe Analysierspalt bis die 1. Ordnung abduicken, dann verschwindet die senkrechte Struktur. Wir brauchen also mindestens 1. Ordnung um die Struktur auflösen zu können.

Bei Aufgabe 3 (weiter):

Für weit geöffnete Analysierspalt bestimmen wir die Spaltbreite von den Einfallsspalt:





扫描全能王 创建

3 Auswertung

Für die benötigte optische Anordnung Aufbau haben wir ein Diodenlaser mit einer Wellenlänge von $\lambda = 532 \text{ nm}$ benutzt um das Objektbild das Fourierbild zu generieren. Durch feine Verschiebungen der Linse L2 würden wir nun auch das Bild der Beugungsstruktur scharf stellen, allerdings wegen des begrenzt verschiebbaren Abstands schaffen wir nur wie folgendes Bild, was später bei Diskussion auch berücksichtigt für die Fehlerquelle werden soll.

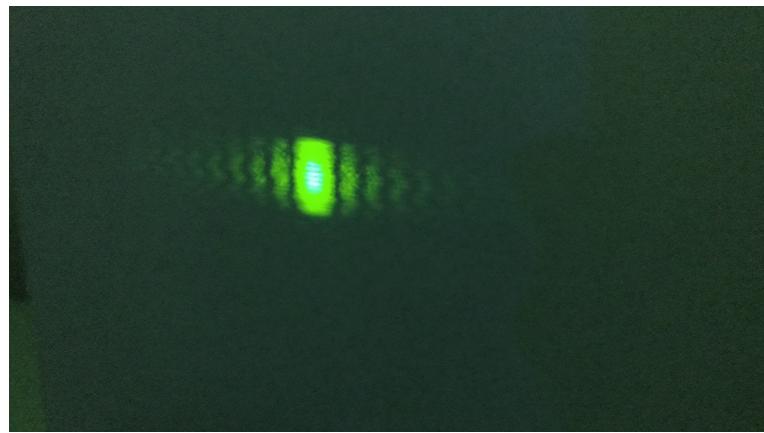


Abbildung 9: Scharfeinstellung

3.1 Eichung der Abzisse

In diesem Versuchsteil werden beim Einzelpunkt zuerst die Lage der Maxima und Minima und deren Intensitätsverhältnisse mit den theoretischen Werten verglichen. Dabei wurden insgesamt zwei Bilder aufgenommen, einmal das Hauptmaximum noch nicht in Sättigung ist und einmal das zentrale Maximum ruhig in Sättigung geht:

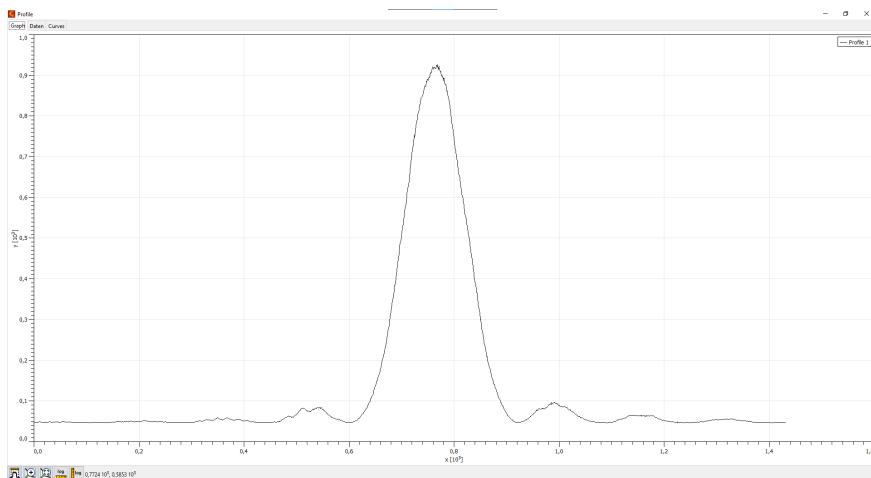


Abbildung 10: Das Hauptmaximum nicht in Sättigung

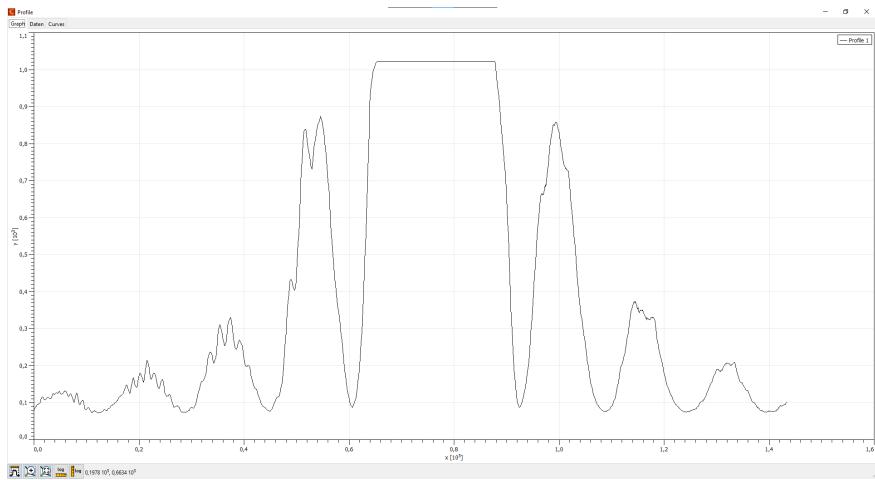


Abbildung 11: Das Hauptmaximum in Sättigung

Wir bestimmen zuerst die Eichung der Abszisse - also wie viele Pixel auf der Kamera einem Millimeter in der Fourierebene entsprechen, indem wir den Analysierspalt in der Fourierebene auf verschiedene Breiten einstellen und die entsprechenden Abstände in Pixel auf dem Monitor ablesen. Wenn man diese Daten in ein Diagramm einträgt, ist es möglich, den Umrechnungsfaktor zwischen Pixeln und dem Abstand in der Fourier-Ebene durch den Betrag der Steigung zu ermitteln. Die Daten wurden im Messprotokoll aufgefassen und als lineare Regression in die folgende Abbildung eingetragen.²

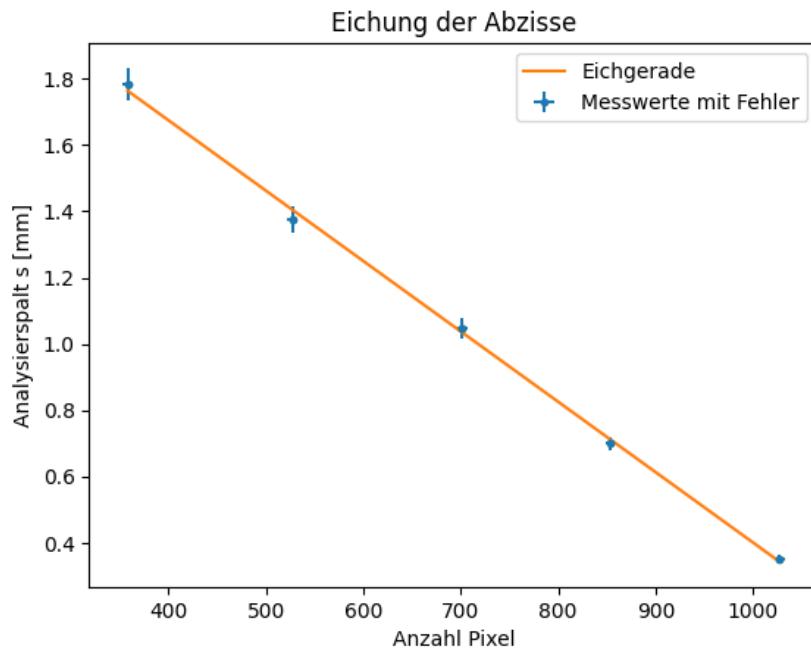


Abbildung 12: Eichung der Abszisse

²Python Code 1

Daher bekommen wir die Steigung soweit durch Umrechnung die Pixel für das Beugungsbild:

$$a = (-0.002124 \pm 0.000044) \text{ mm/pixel} \quad (37)$$

$$1 \text{ pixel} = (2.124 \pm 0.044) \mu\text{m} \quad (38)$$

Außerdem werden wir mithilfe von den Linsengleichungen die Eichung der Abzisse für das Objektbild ermitteln. Laut Versuchsanleitung entspricht ein Pixel einer Breite von $3.45 \mu\text{m}$:

$$g = \left(\frac{1}{f} - \frac{1}{b}\right)^{-1} \quad \frac{B}{G} = \frac{b}{g} \quad (39)$$

$$B = 3.45 \mu\text{m} \quad f = 80 \text{ mm} \quad b = (38.0 \pm 0.5) \text{ cm} \quad (40)$$

$$\Rightarrow 1 \text{ Pixel} = G = \frac{g}{b} B = (0.920 \pm 0.012) \mu\text{m} \quad (41)$$

$$\Delta G = \frac{gB\Delta b}{b^2} \quad (42)$$

Im Folgenden werden diese Pixels häufig implizit verwendet, um Einheiten umzurechnen. Hierbei werden jeweils die Fehler vernachlässigt, da er nur 0,2% beträgt des eigenen Wertes, was deutlich geringer ist als andere Fehlerquellen.

3.2 Quantitative Beobachtungen am Einfachspalt

Nun sollen insgesamt zwei Bilder aufgenommen werden, einmal mit der Belichtungszeit, sodass das Hauptmaximum noch nicht gesättigt ist, während es aufgrund des starken Intensitätsabfalls zu höheren Beugungsordnungen notwendig ist, das zweite Bild mit einer längeren Belichtungszeit aufzunehmen, um die höheren Beugungsordnungen angemessen zu erfassen. Mit der Text-Dateien können wir die beiden Intensitätsprofile plotten und die dazugehörende Maxima sowie Minima ablesen:³

Neben der experimentell gemessenen Daten führen wir noch zusätzlich die Code für die Auswertung der theoretischen Situation durch, indem wir die theoretische Intensitätsverteilung:⁴

$$I = \frac{\sin(x)^2}{x^2} \quad (43)$$

benutzen, mit $x = d/\lambda\pi \sin(\alpha)$.

Hier unten kann man die experimentell sowie die theoretisch ausgewertete Abbildung sehen:

³Python Code 2

⁴Python Code 3

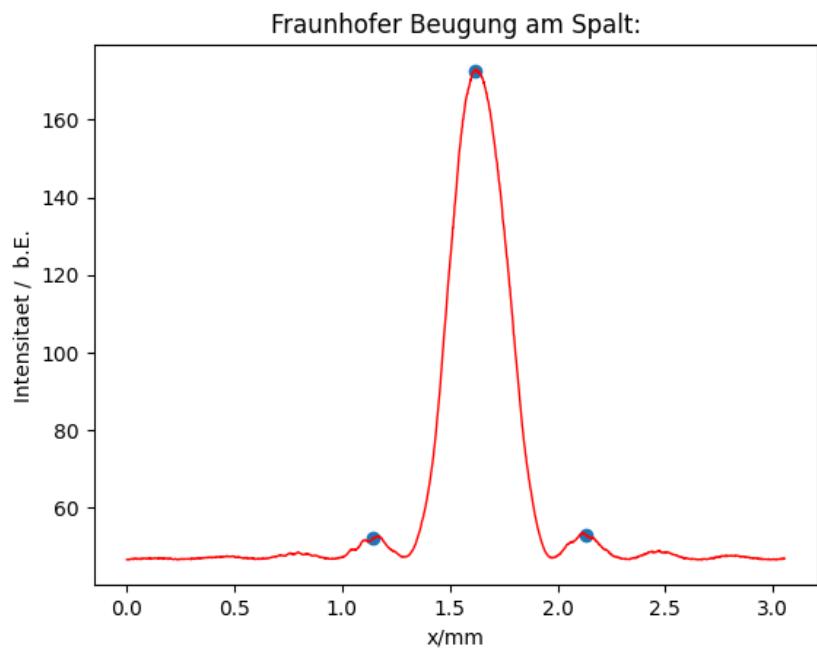


Abbildung 13: Intensitätsprofil nicht in Sättigung

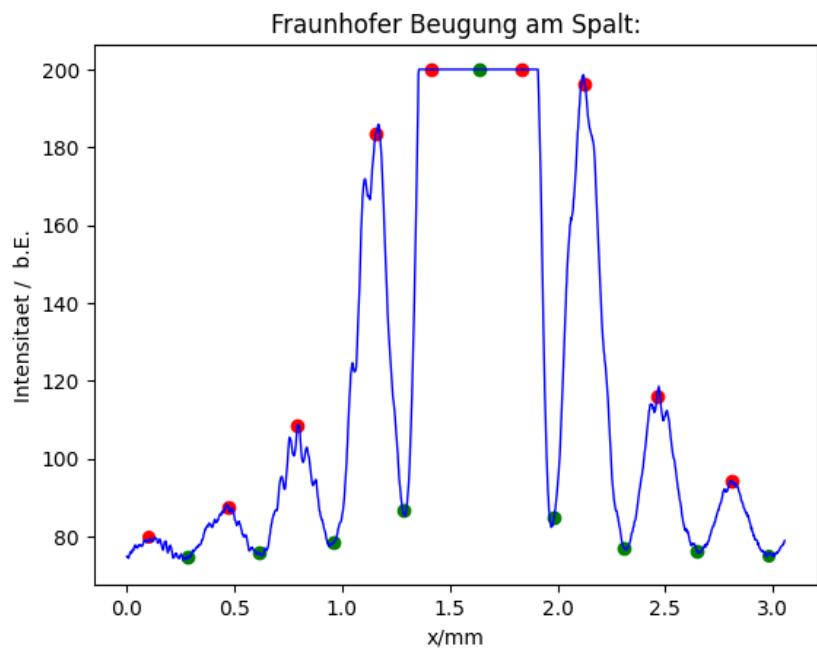


Abbildung 14: Intensitätsprofil in Sättigung

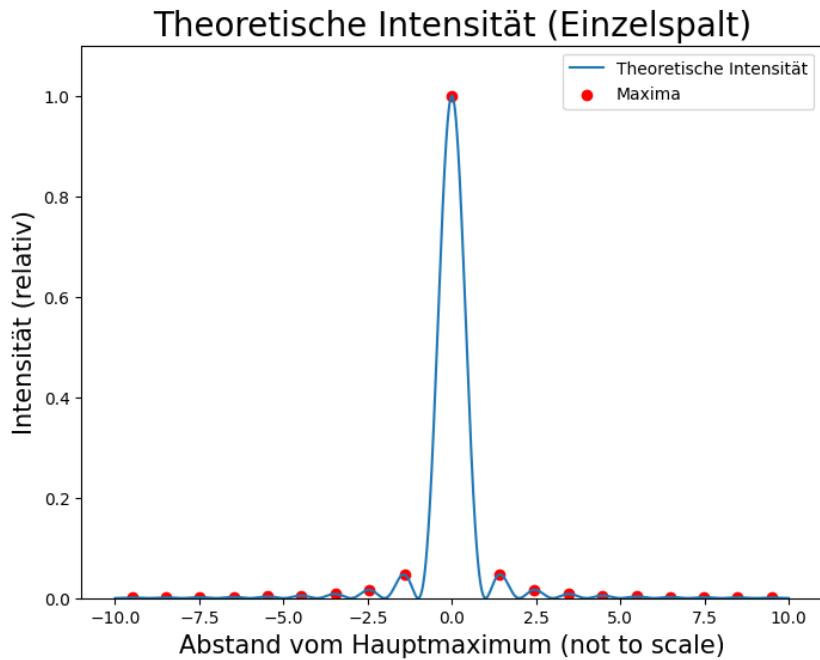


Abbildung 15: Theoretische Intensitätsverteilung für Einzelspalt

Um eine Vergleichbarkeit der Messwerte mit der Theorie zu gewährleisten, erfolgt die Berechnung der relativen Intensität im Vergleich zum Hauptmaximum anstelle der absoluten Intensität. Hierbei wird für das erste Hauptmaximum die Intensität der ersten Messung herangezogen, während für alle weiteren Maxima die zweite Messung verwendet wird:⁵

$$I_{rel} = \frac{I_1}{I_0} \quad I'_{rel} = \frac{I'_n}{I'_1} \frac{I_1}{I_0} \quad (44)$$

Wir berücksichtigen zudem, dass aufgrund des Untergrunds alle Messdaten um einen bestimmten Wert nach oben verschoben sind. Diesen Verschiebungswert müssen wir bei der Berechnung der relativen Intensitäten entsprechend abziehen.

⁵Python Code 4

Tabelle 1: Vergleich zwischen theoretischen und experimentell bestimmten Intensitätsprofilen

Ordnung	$I_{rel,exp}$	$\Delta I_{rel,exp}$	$I_{rel,theo}$	σ Abweichung
0	1	0	1	0
1	0.04312	0.00693	0.04665	0.51
2	0.01333	0.00327	0.01643	0.95
3	0.00492	0.00073	0.00833	4.67
4	0.00285	0.00034	0.00502	6.38
5	0.00197	0.00045	0.00336	3.09
6	0.00135	0.00055	0.00240	1.91

Wobei die Fehler sich ergeben als:

$$\Delta I_{rel} = \sqrt{\left(\frac{\Delta I_1}{I_0}\right)^2 + \left(\frac{I_1 \Delta I_0}{I_0^2}\right)} \quad (45)$$

$$\Delta I'_{rel} = I'_{rel} \sqrt{\left(\frac{\Delta I}{I_0}\right)^2 + \left(\frac{\Delta I}{I_1}\right)^2 + \left(\frac{\Delta I}{I'_1}\right)^2 + \left(\frac{\Delta I}{I'_n}\right)^2} \quad (46)$$

Die Intensitäten der Maxima wurden mithilfe des Programms Gwyddion und manuell mit Python bestimmt, daher wird ein Fehler von $\Delta I = 20\text{ Counts}$ für die Intensitätsmessungen angenommen.

Um einen Vergleich mit den theoretischen Werten zu ermöglichen machen wir eine Fehlerabschätzung:

$$\frac{|I_{rel,exp} - I_{rel,theo}|}{\Delta I_{rel,exp}} \quad (47)$$

Es ist festzustellen, dass die Abweichungen in den Fehlern bei der 4. und 5. Ordnung bereits über 3σ liegen. Zudem wird deutlich, dass die Fehler zunächst an Signifikanz gewinnen und anschließend wieder abnehmen. Besonders auffällig ist dabei die signifikante Abweichung von 6.38σ bei der 4. Ordnung. Alle gemessenen Daten zeigen systematisch kleinere Werte als die erwarteten. Dies könnte wahrscheinlich auf die teilweise Absorption durch Objekte oder die Dispersionseigenschaften des verwendeten Prismas zurückgeführt werden.

Es sollten nun der Abstand der Minima n-ter Ordnung (jeweils in Pixel angegeben) gegen die Ordnungszahl n aufgetragen werden. Es sollte sich eine Gerade ergeben, deren Steigung die Spaltweite des verwendeten Beugungsspaltes zu bestimmen gestattet.⁶

⁶Python Code 5

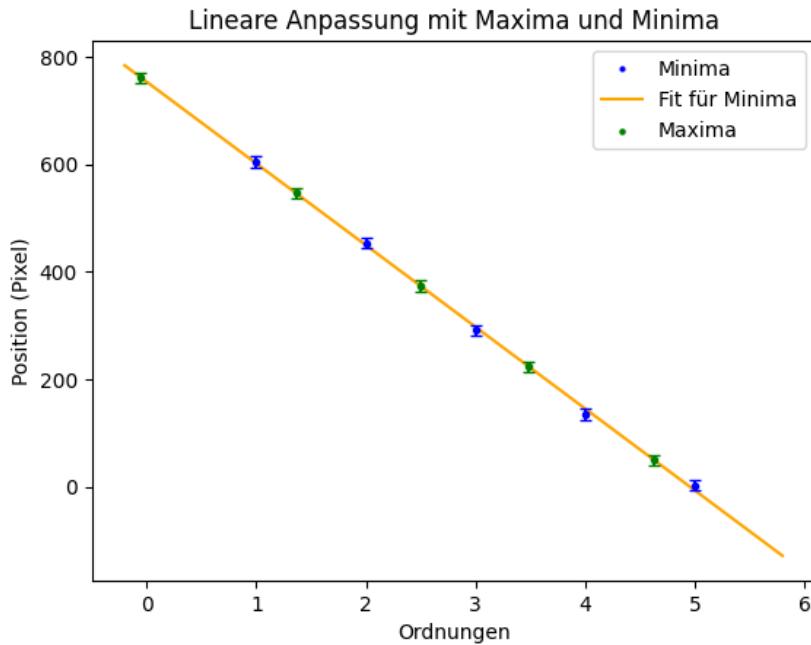


Abbildung 16: Abstand der Minima in Pixel aufgetragen gegen Ordnungen

Die Maxima müssen dann bei den Ordnungen $n = 0.5, 1.5, 2.5, \dots$ auftauchen, da die Differenz zwischen den Gangunterschieden von Maxima und Minima immer genau $\lambda/2$ beträgt, während die Differenz zwischen den Gangunterschieden von zwei Maxima λ beträgt. Für den Fehler haben wir für alle Extrema 10 Pixel abgeschätzt, da es während des Experiments eine besonders starke Rauschung gibt. Aus der Eichgerade ermittelten Ordnungen von den Maxima bekommen wir ein Array:

$$x = -0.05, 1.36, 2.50, 3.49, 4.63 \quad (48)$$

$$\pi x = -0.157, 4.270, 7.85, 10.96, 14.55 \quad (49)$$

was innerhalb des Toleranzbereichs der mathematischen Tatsache sowie unserer Erwartung entspricht, da die Position der Sinc Funktion⁷ bei jeden Extrema bisschen von den Nullstellen der Cosine Funktion abweicht (siehe Abbildung 17):

⁷Wiki: Sinc Funktion

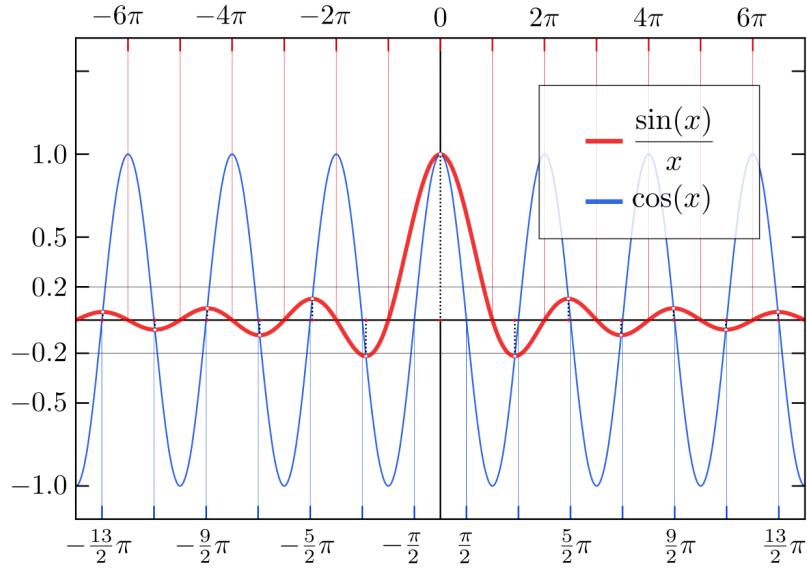


Abbildung 17: Sinc Funktion

Nun werden wir die Spaltbreite durch Auswertung der Eichgerade ermitteln. Aus der Abbildung 5 kann das Beugungsmuster (Minimum) eines Einzelspaltes für monochromatisches Licht durch die Formel beschrieben werden (Gangunterschied):

$$\sin(\alpha) = \frac{n\lambda}{d} \quad (50)$$

wobei: d die Spaltbreite, α der Beugungswinkel, n die Ordnung des Minimums, $\lambda = 532 \text{ nm}$ die Wellenlänge des Lichts. Mit der Winkelnährung $\frac{X}{f} = \tan(\alpha) \approx \sin(\alpha)$ für kleine Winkel ist:

$$d = \frac{n\lambda f}{X} \quad (51)$$

X bezeichnet sich als der Abstand der Minima und f die Brennweite der Linse. Mit der Steigung ist es also möglich, die Spaltbreite zu ermitteln:

$$\frac{X}{n} = k = (151.98 \pm 2.95) \frac{\text{Pixel}}{\text{Ordnung}} \quad (52)$$

$$\frac{n}{X} = (3.10 \pm 0.06) \text{ mm}^{-1} \quad (53)$$

Bei dem letzten Schritt haben wir die Umrechnung der Pixel aus Gl.(38) benutzt. Mit Vernachlässigung der Fehler von Wellenlänge und Brennweite ergibt sich der Fehler der Spaltbreite als:

$$\Delta d = \lambda f \Delta \left(\frac{n}{X} \right) \quad (54)$$

Zusammen mit Gl.(51) lässt sich die Spaltbreite ausrechnen als:

$$\underline{d_{\text{Einzel}} = (131.94 \pm 2.56) \mu m} \quad (55)$$

3.3 Beugungsstruktur des Doppelspaltes

Zuerst wird die theoretische Beugungsstruktur sowohl des Doppel- als auch des Einzelspalts betrachtet.⁸ Beim Doppelspalt entsteht ein Beugungsmuster, das eine Überlagerung zweier Einzelspaltmuster ist, jedoch mit zusätzlichen Interferenzstreifen. Während das Einzelspaltmuster relativ glatt und gleichmäßig abfällt, zeigt das Doppelspaltmuster eine Reihe von zusätzlichen hellen und dunklen Streifen, die auf Interferenzeffekte zurückzuführen sind. Die theoretische Intensitätsverteilung wird durch

$$I(x) = \frac{\sin(x)^2}{x^2} \cos(vx)^2 \quad (56)$$

beschrieben, wobei $v = g/d$ das Verhältnis von Spaltabstand zu der Spaltbreite des Doppelspalts aus Aufgabe 4 darstellt.

Hierbei lässt sich beobachten, dass die Intensitätsverteilung des Einzelspalts als Einhüllende für die des Doppelspalts dient. Das Beugungsmuster eines Einzelspalts zeichnet sich durch ein zentrales Maximum aus, das wesentlich intensiver und breiter ist als die seitlichen Maxima.⁹

⁸Python Code 3

⁹Python Code 6

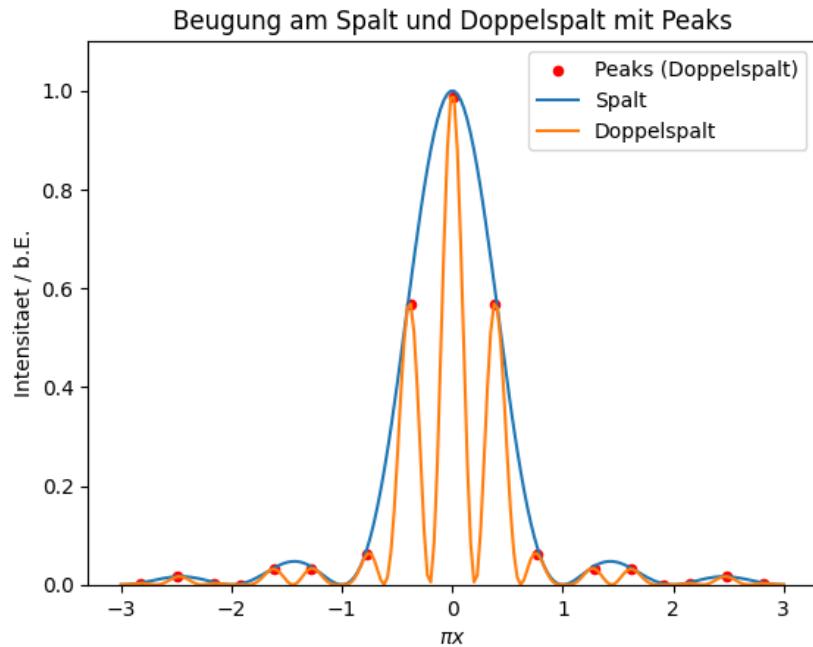


Abbildung 18: Theoretische Intensitätsverteilung des Doppelspalts

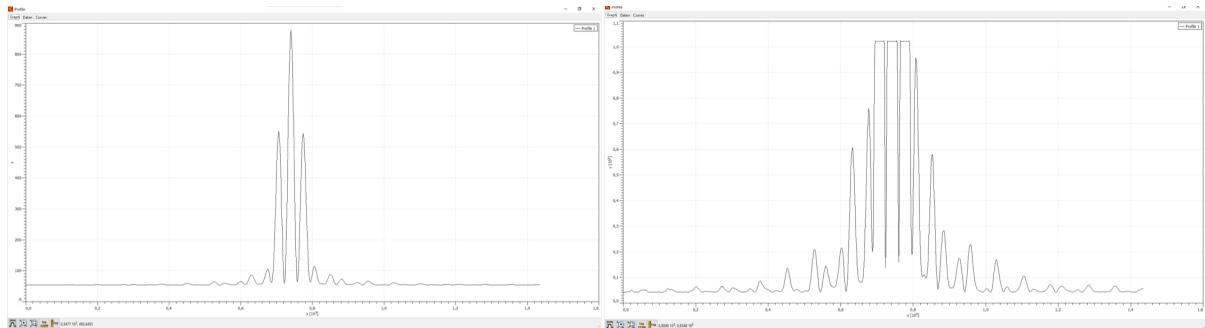


Abbildung 19: Experiment Doppelspalt

Wir ergänzen die Darstellung mit den experimentell ermittelten Intensitätsverteilungen für beide Zustände, sowohl in Sättigung als auch nicht in Sättigung, analog zum Einzelspaltexperiment.

Allerdings sollen wir beachten, dass aufgrund der unbewerkstelligten Scharfeinstellung können wir nur das Bild bekommen, das besonders gerauscht wird bzw. sehr viele "cluttered Signals" beinhaltet. Die schaffen wir leider nicht zu beseitigen, weshalb werden es manuell Peaks gefunden.

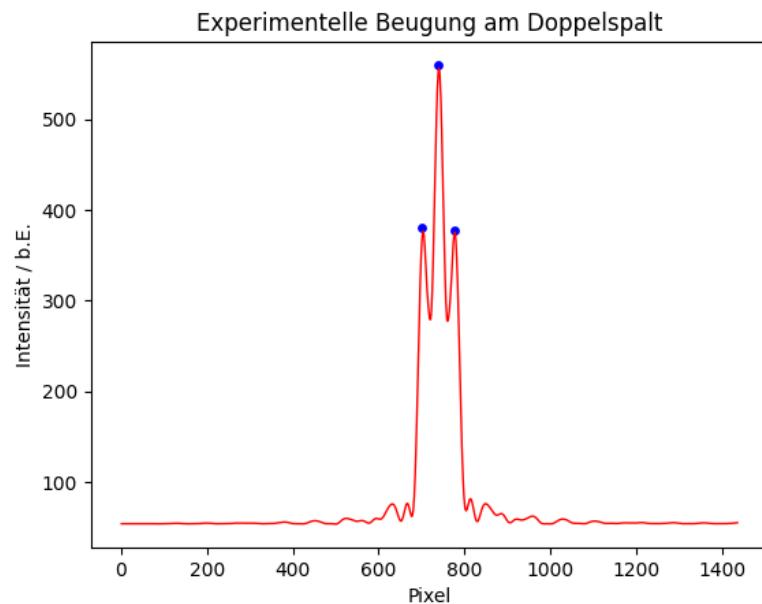


Abbildung 20: Experimentelle Intensitätsverteilung des Doppelspalts nicht in Sättigung

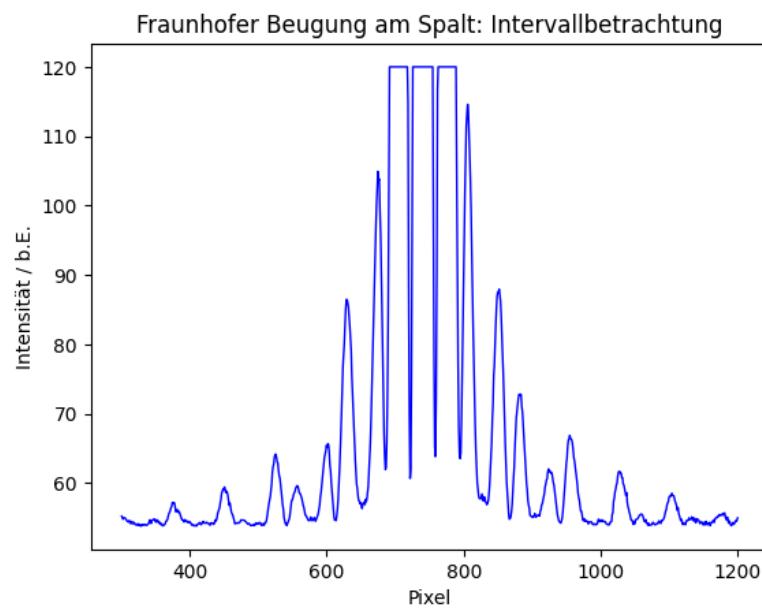


Abbildung 21: Experimentelle Intensitätsverteilung des Doppelspalts in Sättigung

Da die Breiten der beiden Spalte voneinander abwichen, erfolgte die Berechnung der mittleren Spaltbreite durch das Bilden des Durchschnitts der beiden Werte. Die halbe Differenz zwischen den Spaltbreiten wurde als Maß für den Fehler herangezogen.

Demzufolge ergab sich folgendes Ergebnis:¹⁰

$$g = (626 \pm 20) \text{ Pixel} \quad b = (258 \pm 20) \text{ Pixel} \quad (57)$$

für den Spaltabstand. Der Fehler für den Spaltabstand ist lediglich eine Schätzung, die auf dem Fehler der Spaltbreite basiert, während der schlussendliche Fehler aus quadratischer Addition der relativen Fehler der Messwerte und des Umrechnungsfaktors stammt.

Wenn wir uns zunächst die Darstellung der Intensitätsverteilung im nicht gesättigten Zustand ansehen, stellen wir fest, dass die Gestaltung beim ersten Maximum und beim Hauptmaximum eine starke Ähnlichkeit mit der theoretischen Intensitätsverteilung aufweist. Im Gegensatz dazu sind bei der Darstellung im gesättigten Zustand, insbesondere bei höheren Ordnungen, die entsprechenden Maxima und Minima aufgrund starker Rauschung weniger deutlich erkennbar.

Die Spaltfunktion formt als Einhüllende das Beugungsmuster des Doppelspalts und ihre Nullstellen sind in den größeren Minima des Musters erkennbar. Die Gitterfunktion, resultierend aus der Interferenz der Wellen beider Spalte, erzeugt zusätzliche, kleinere Minima innerhalb dieser Einhüllenden. Daraus sehen wir, dass die Minima der ersten und zweiten Ordnung wahrscheinlich durch die Gitterfunktion verursacht werden, während das dritte von der Spaltfunktion stammen könnte, weil das 3. Minimum genau dem der Spaltfunktion entspricht. Obwohl dies nicht klar ist, da das Minimum 3. und 4. Ordnung fast zusammenfällt.

Nun werden wir für die zwischen den 1. Minima der Spaltfunktion gemessenen Nebenmaxima die relativen Intensitäten bezogen auf das zentrale Maximum mit den entsprechenden theoretischen Werten vergleichen. Die Berechnung von Fehlern und Abweichungen erfolgt in ähnlicher Weise wie bei der Analyse des Einzelspalts. Dabei wird erneut ein Fehler von $\Delta I = 10$ counts berücksichtigt, der auf die Ungenauigkeiten bei der Bestimmung der Maxima zurückzuführen ist:

Tabelle 2: Vergleich zwischen theoretischen und experimentell bestimmten Intensitätsprofilen Doppelspalt

Ordnung	$I_{rel,exp}$	$\Delta I_{rel,exp}$	$I_{rel,theo}$	σ Abweichung
0	1	0	1	0
1	0.64032	0.02347	0.57598	2.74
2	0.07509	0.00498	0.06349	2.33

Alle Fehlerabweichungen liegen innerhalb von 3σ und sind damit nicht signifikant.

¹⁰siehe Messprotokoll

3.4 Quantitative Auswertung der modizierten Objektbilder am Einzelspalt

Am Beispiel des Spaltes soll nun die Auswirkung der Manipulationen in der Fourierebene auf die Struktur des Bildes quantitativ untersucht werden. Am Anfang kann man eine recht deutlich erkennbare Rechteckfunktion mit einer Vielzahl von Schwingungen¹¹ beobachten:¹¹

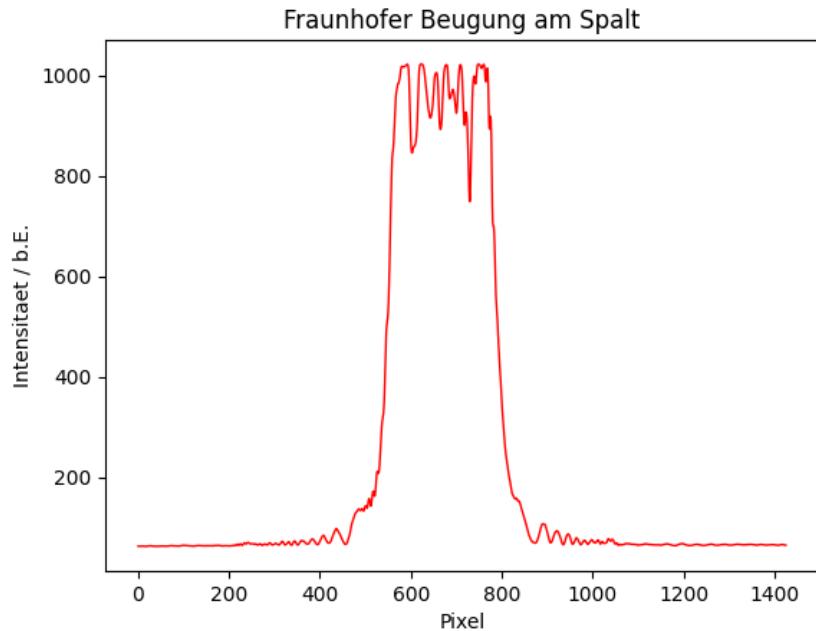


Abbildung 22: "Wiggles" vom Einzelspalt

Mit der Spaltfunktion können wir ebenso die Spaltbreite bestimmen und die nach Gl.(41) umrechnen, die beträgt:

$$d_{\text{Einzel},\text{Obj}} = (226 \pm 20) \text{ Pixel} \approx (207.92 \pm 18.40) \mu\text{m} \quad (58)$$

Um die Zuverlässigkeit der ermittelten Spaltbreite zu sehen, da das Objektbild nicht ganz stabil ist, machen wir einen Vergleich mit der in **3.2** bestimmten Spaltbreite d_{Einzel} :

$$\frac{|d_{\text{Einzel}} - d_{\text{Einzel},\text{Obj}}|}{\sqrt{(\Delta d_{\text{Einzel}})^2 + (\Delta d_{\text{Einzel},\text{Obj}})^2}} \approx 4.09\sigma \quad (59)$$

Die Fehlerabweichung befindet sich nicht innerhalb von 3σ und ist signifikant. Man beachtet, dass die Methode zur Messung der Spaltbreite mit dem Objektbild deutlich ungenauer ist als die Methode mit den Positionen der Maxima und Minima, daher

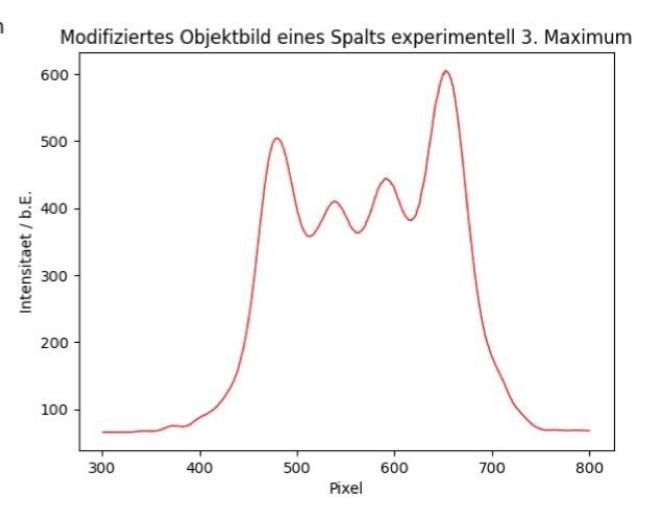
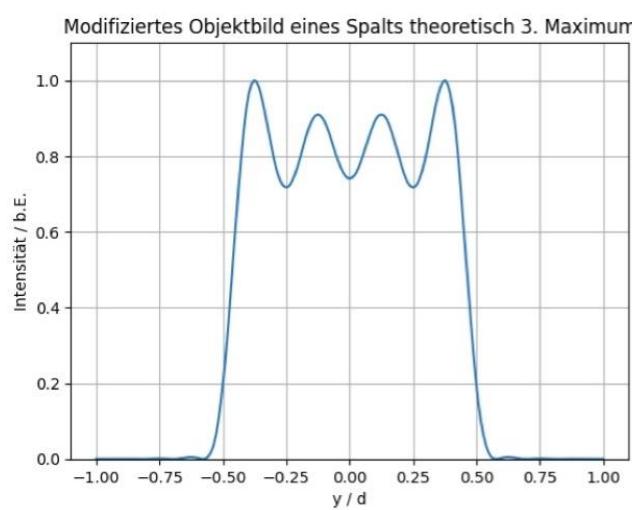
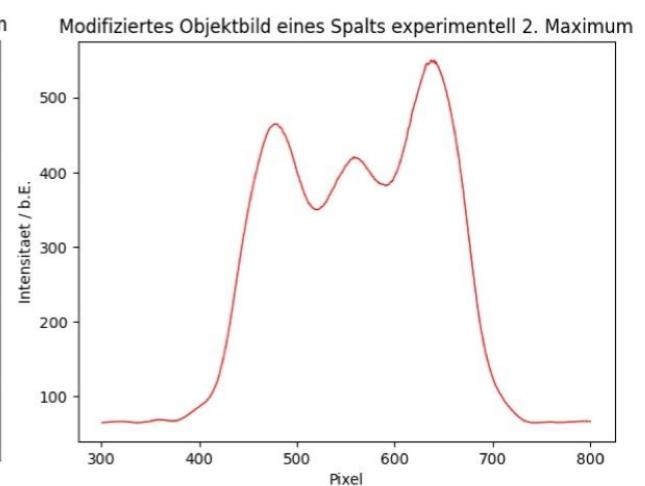
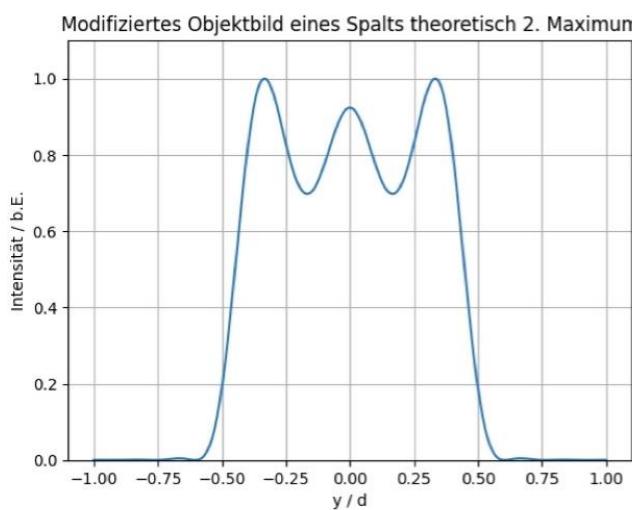
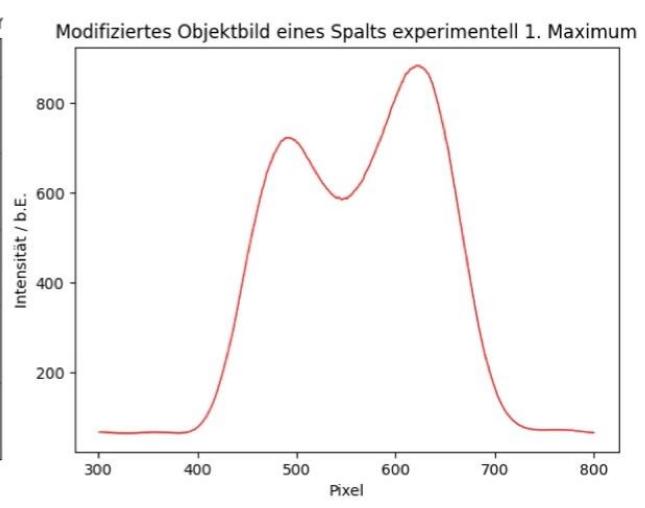
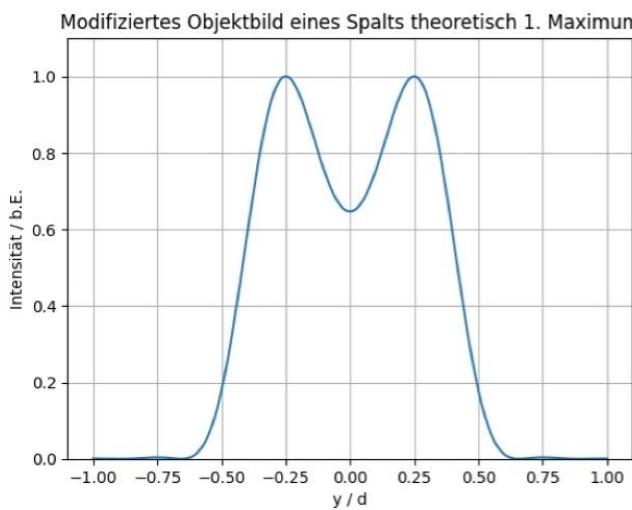
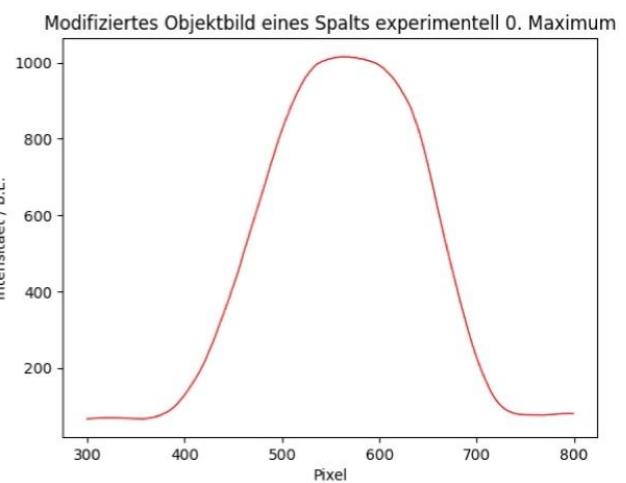
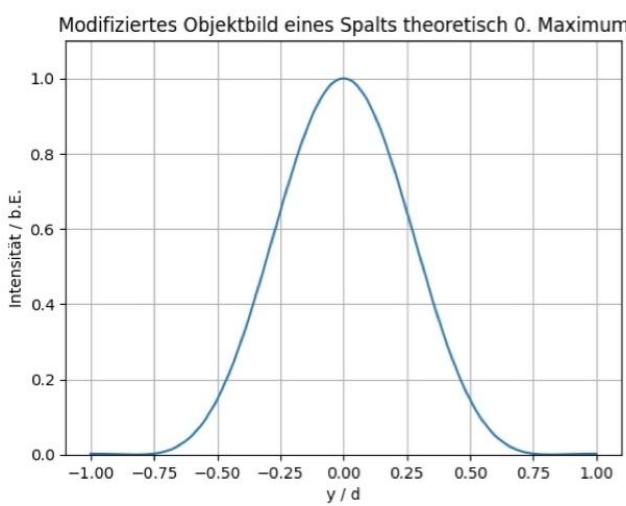
¹¹Python Code 7

haben wir tatsächlich die Fehler schon gezielt höher abgeschätzt, jedoch weichen die beiden berechneten Spaltbreiten stark voneinander ab, die genauen Gründen werden wir uns dann in Diskussion weiter beschäftigen.

Als nächster Schritt erfolgt der Vergleich der gemessenen modifizierten Objektbilder mit den theoretischen modifizierten Objektbildern. Dazu wird Gl.(31) benutzt um die Theorie zu erstellen, wobei die Spaltbreite d auf 1 eingestellt wird, während die experimentell ermittelten Messdaten in Python importiert und geplottet werden.

Die Maxima sind symmetrisch zum Spaltmittelpunkt angeordnet: Bei ungerader Maximaanzahl ist das mittlere Maximum die Symmetrieachse, bei gerader das mittlere Minimum. Im Vergleich zeigt sich auch deutlich, dass die Anzahl der Maxima in den Messdaten exakt den theoretischen Vorhersagen entspricht. Wenn lediglich das Hauptmaximum durchgelassen wird, weist das Objektbild lediglich ein Maximum auf. Mit jeder weiteren durchgelassenen Ordnung erhält das Objektbild jeweils ein zusätzliches Maximum.

Bei mehr als zwei Maxima ist auffällig, dass die Intensität an den Rändern deutlich dichter ist als in der Mitte. Diese Beobachtung erklärt die visuell wahrnehmbaren Veränderungen im Bild: Die Ränder erscheinen schärfer, wenn man die Beugungsordnungen höherer Ordnung mit zur Abbildung zulässt.



Außerdem machen wir einen Vergleich von den theoretischen sowie experimentellen Intensitäten der Maxima und Minima der ersten drei Bildkurven, normiert auf das Maximum des zur 0-ten Beugungsordnung gehörenden modifizierten Spaltbildes. Die Ergebnisse werden in die folgende Tabelle eingetragen:

Tabelle 3: Normierte Intensitäten Vergleich 0.-1. Ordnung

Ordnung	theoretisch	experimentell	Abweichung
Maximum links	1	0.69 ± 0.15	2.07σ
Minimum mittel	0.65	0.55 ± 0.15	0.67σ
Maximum rechts	1	0.86 ± 0.15	0.93σ

Tabelle 4: Normierte Intensitäten Vergleich 0.-2. Ordnung

Ordnung	theoretisch	experimentell	Abweichung
Maximum links	1	0.42 ± 0.15	3.87σ
Minimum links	0.7	0.30 ± 0.15	2.67σ
Maximum mittel	0.92	0.37 ± 0.15	3.67σ
Minimum rechts	0.7	0.34 ± 0.15	2.40σ
Maximum rechts	1	0.51 ± 0.15	3.27σ

Tabelle 5: Normierte Intensitäten Vergleich 0.-3. Ordnung

Ordnung	theoretisch	experimentell	Abweichung
Maximum links 1	1	0.45 ± 0.15	3.67σ
Minimum links	0.72	0.31 ± 0.15	2.73σ
Maximum links 2	0.91	0.35 ± 0.15	3.73σ
Minimum mittel	0.74	0.32 ± 0.15	2.8σ
Maximum rechts 2	0.91	0.39 ± 0.15	3.47σ
Minimum rechts	0.72	0.34 ± 0.15	2.53σ
Maximum rechts 1	1	0.55 ± 0.15	3σ

Als Fehler der experimentell ausgerechneten Daten wählen wir willkürlich $\Delta = 0.15$.

Der Vergleich zeigt, dass die meisten gemessenen Intensitätswerte ungefähr mit den theoretisch berechneten übereinstimmen, eine Beobachtung, die bereits an den Diagrammen sichtbar war. Allerdings gibt es, besonders bei den Objektbildern höherer Ordnungen, deutliche Abweichungen. Manche befinden sich zwar noch innerhalb des 3σ -Bereichs, aber aufgrund der willkürlichen Fehlerwahl wenig aussagekräftig sind. Trotz der beobachteten Ergebnisse sind solche Ausgänge zu erwarten, wie zuvor erwähnt. Die mechanische Scharfeinstellung zu Beginn ist aufgrund der begrenzten Schienenlänge nicht möglich, was bedeutet, dass bei jeder Analysierspaltsbreite jeweils verschiedene Belichtungszeiten erforderlich sind, um das Objektbild so scharf wie möglich darzustellen. Dies kann jedoch zu Problemen führen, da die Intensitäten

unter stets wechselnden Bedingungen gemessen werden, was die Konsistenz der Ergebnisse beeinträchtigt.

3.5 Qualitative Auswertung der modifizierten Objektbilder am Doppelspalt

Ähnlich wie beim Einzelpaltexperiment soll der Analysierspalt zuerst ganz weit aufgemacht werden, damit wir aus dem unmodifizierten Objektbild die Spaltbreite sowie den Spaltabstand bestimmen können.¹²

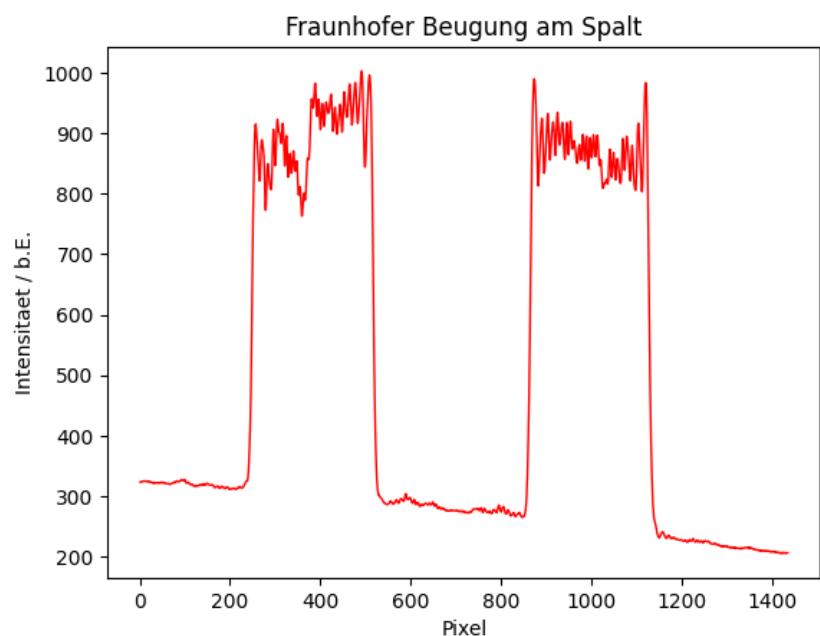


Abbildung 23: "Wiggles" vom Doppelspalt

Betrachtet man das Objektbild, so stellt man fest dass viele störende Strukturen überlagert sind. Der Laserstrahl selbst hat kein homogenes Intensitätsprofil, sondern weist eine überlagerte Beugungsstruktur auf. Diese Störungen können teilweise durch FFT-Filter in Gwyddion kompensiert werden. Allerdings wegen des nicht schaff eingestellten Lichtbündels treten schon so viele zusätzliche Beugungen und Streuungen auf, dass wir nicht einmal mit Filtern oder Pixelmitteln Extrahiere Profile schaffen können.

Durch grobe Abschätzung sowie statistische Methoden bekommen wir die Spaltbreite d und den Spaltabstand g :

$$d = (258 \pm 20) \text{ Pixel} = (237.36 \pm 18.40) \mu\text{m} \quad (60)$$

$$g = (626 \pm 20) \text{ Pixel} = (613.48 \pm 18.40) \mu\text{m} \quad (61)$$

¹²Python Code 8

Um sich einen Überblick über die sukzessiv auftretenden Strukturen zu verschaffen wurden der Analysierspalt langsam zgedreht und simultan die Veränderungen der charakteristischen Beugungsstruktur auf dem Schirm und des Spaltbildes auf dem Monitor beobachtet. Der Analysierspalt wurde so gedreht, dass immer wenige Beugungsordnungen zugelassen wurden, bis schließlich die Doppelstruktur nicht mehr identifiziert werden kann. Analog wie vorher werden wir auch einen Vergleich zwischen theoretischen Objektbildern mithilfe von Gl.(36) und den experimentell modifizierten Objektbildern durchführen.¹³

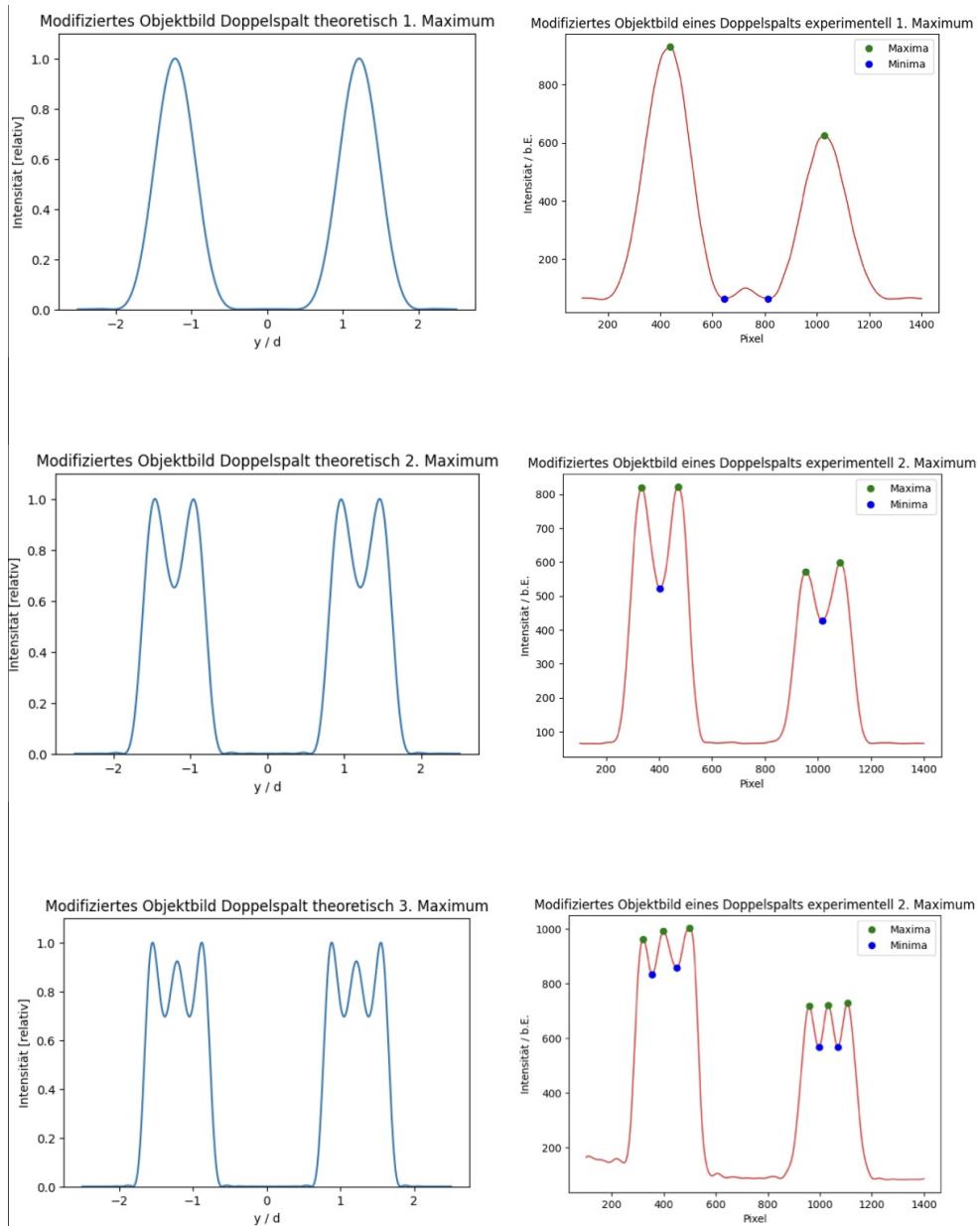


Abbildung 24: Vergleich theoretisch sowie experimentell Modifizierter Objektbilder

¹³Python Code 9

Wir können tatsächlich bei der Messung der Objektbilder zwei unterschiedlich große Peaks beobachten, obwohl theoretisch die zwei identisch sein sollten. Der Grund liegt zuerst daran, dass der Doppelspalt eine unvollkommene Spaltgeometrie besitzt, insbesondere die Spaltbreite, die wir aus Abbildung (23) herausfinden können. Außerdem kann eine leichte Fehlausrichtung (Winkelabweichung) des experimentellen Aufbaus, einschließlich des Doppelspalts, der Lichtquelle oder des Detektors, zu unterschiedlichen Intensitäten führen, weil optische Apparatur besonders sensitiv sind. Dies erklärt warum die zwei Peaks unterschiedlich hoch sind.

Bei einem quantitativen Vergleich der Intensitätswerte von Maxima und Minima, normiert auf das Maximum der 0-ten Ordnung, zwischen Theorie und Experiment, wie durch die Berechnung von Verhältnissen gemäß Abschnitt 3.4, zeigen sich auch erhebliche Abweichungen. Die experimentell ermittelten Werte sind deutlich niedriger als die theoretischen Vorhersagen.

Zum Schluss wurde der Analysierspalt weiter zgedreht. Die beiden rechteckigen Spaltprofile verschwimmen zu zwei gaussähnlichen Profilen bis die Doppelstruktur endgültig verschwindet und nur noch ein flaches Plateau sichtbar bleibt (Abbildung (25)):¹⁴

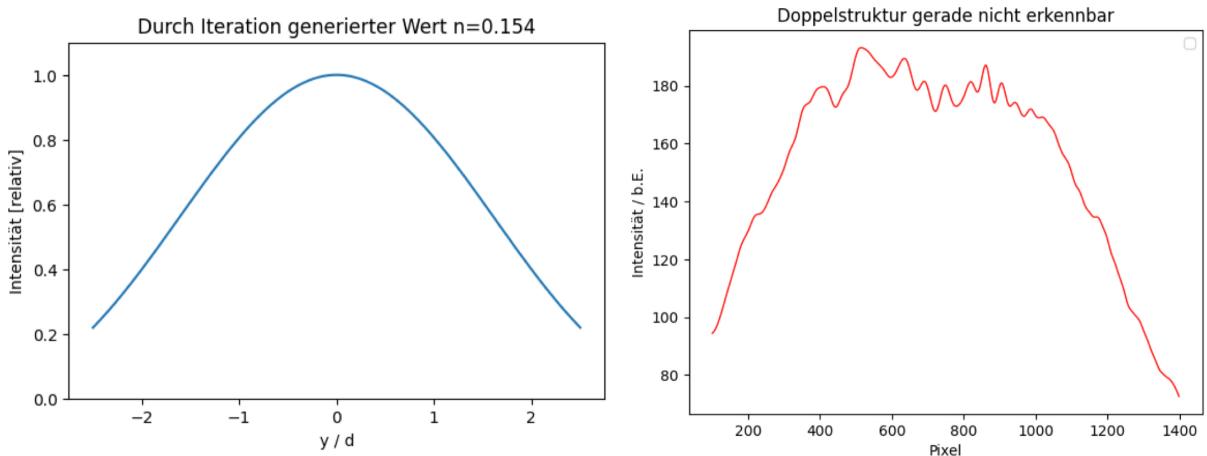


Abbildung 25: Doppelstruktur nicht mehr erkennbar, links: theoretisch mit $n=0.145$, rechts: experimentell

Selbst wenn wir die Rauschunterdrückung von Python verwendet haben lassen sich noch viele Störungen im modifizierten Objektbild beobachten, das gesamte Bild sieht viel unstabiler als die 3 anderen aus.

Um den entsprechend experimentellen Fall zu simulieren, sollen für unterschiedliche Werte von k_y nahe dem Wert $k_y = k_0 \sin \alpha = \frac{k_0 \lambda}{g}$ durch iterative Anpassung einen Wert finden, der die experimentell bestimmte Struktur genau wiedergibt. Dieser Wert

¹⁴Python Code 10

ergibt sich als $n = 0.154$.

Experimentell beträgt die Spaltbreite des Analysierspaltes für den Fall, wenn die Doppelstruktur gerade verschwindet, $X_b = (0.052 \pm 0.008) \text{ mm}$, mit umgestellter Gl.(51) können wir also theoretisch die dazugehörige n_b berechnen (Zur Herleitung beachten wir beim Doppelsaplt $k_{y,n} = k_0 \sin \alpha_n = 2\pi n/d$ nach Gl.(30)):

$$n_b = \frac{X_b d}{2 f \lambda} \quad (62)$$

$$\Delta n_b = n_b \sqrt{\left(\frac{\Delta X_b}{X_b}\right)^2 + \left(\frac{\Delta d}{d}\right)^2} \quad (63)$$

$$\Rightarrow n_b = 0.145 \pm 0.025 \quad (64)$$

Wir berechnen die Fehlerabweichung zwischen den beiden n um die Zuverlässigkeit zu überprüfen:

$$\frac{|n_b - n|}{\Delta n_b} \approx 0.36\sigma \quad (65)$$

Die Fehlerabweichung befindet sich innerhalb von 3σ und ist nicht signifikant.

Wenn wir das Objektbild für verschiedene Werte von n'_b plotten, stellen wir fest, dass bei einem Wert von $n'_b = 0.27$ die Doppelstruktur gerade nicht mehr erkennbar ist. Dies entspricht dem theoretischen Fall, bei dem die höchste Ordnung erreicht ist, um die Doppelstruktur noch nicht zu erkennen.

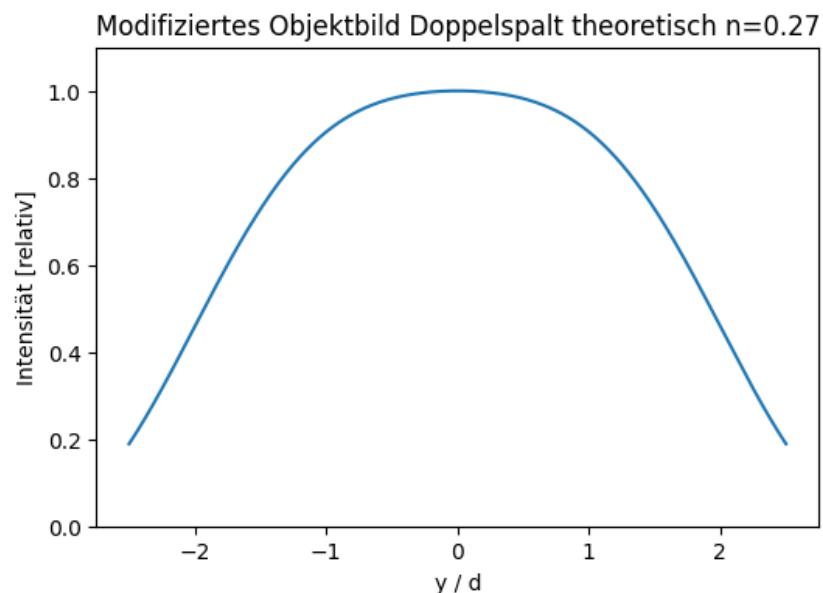


Abbildung 26: Bei Analysierspaltsbreite d_s verschwindet die Doppelstruktur

4 Zusammenfassung und Diskussion

In diesem Experiment werden quantitative und qualitative Untersuchungen der Beugungsstrukturen sowie der Eigenschaften der Beugung an Einzel- und Doppelspalten durchgeführt. Dabei werden Beugungsbilder und Objektbilder mit einer CCD-Kamera aufgenommen und mit theoretischen Vorhersagen verglichen, die auf dem Prinzip der Fraunhofer'schen Beugung basieren. Bei der Beugung am Spalt liegt der Fokus vor allem auf der Untersuchung der Position und Intensität der Maxima und Minima. Beim Objektbild wird hingegen die Wirkung eines Analysierspalts untersucht. Zusätzlich sollen qualitative Beobachtungen verschiedener Beugungsobjekte vorgenommen werden, die durch Manipulationen in der Fourier-Ebene entstehen.

Es wurden zuerst zwei Umrechnungsfaktoren durch Eichung der Abzisse an zwei verschiedenen Kamerapositionen bestimmt, um vergleichbare Ergebnisse in späteren Berechnungen zu erzielen. Für Beugungsbilder ergibt sich ein Umrechnungsfaktor von 1 Pixel gleich $(2.124 \pm 0.044) \mu m$, während dieser Faktor für Objektbilder bei $(0.920 \pm 0.012) \mu m$ liegt. Dieser Unterschied ist nicht überraschend, da das Lichtbündel einen längeren Weg zurücklegen muss, um das Beugungsbild zu erzeugen, was zu einer stärkeren Streuung und somit zu einer stärkeren Vergrößerung des Objektbildes im Vergleich zum Beugungsbild führt.

Bei der Datenaufnahme hätten wir nur die Hälfte der Intensitätsverteilung erfasst, was zu einer größeren Anzahl von Peaks führte. Dies ermöglichte es, die Eichgerade mit höherer Genauigkeit zu bestimmen. Bei unserer Messung gibt es wenige zu auswertende lokale Extrema, was wenige Datenmenge für weitere Auswertungen liefert, die die Zuverlässigkeit der Ergebnisse beeinflussen kann, insbesondere bei der Anpassung der linearen Regression von Abstand der Minima gegen Ordnungszahl in Abbildung 16. Interessanterweise war die Eichung mit dem Analysierspalt ungenauer als die Berechnung des Abbildungsmaßstabs über die Linsengleichung. Dies war unerwartet, da die Messuhr aufgrund ihres geringeren Skalenfehlers präzisere Ergebnisse liefern würde als ein Zollstock zur Messung der Bildweite. Eine mögliche Erklärung könnte sein: Verglichen mit der optischen Anordnung weist die Längenmessung eine geringere Genauigkeit auf, weshalb nur $0.5 cm$ Fehler für die gesamte Länge zu konservativ erscheint.

Bei der Untersuchung der Intensitätsverteilung der Beugungsfigur am Einzelspalt wurde festgestellt, dass die relativen Intensitäten der Maxima ab der dritten Ordnung deutlich von den theoretisch bestimmten Werten abweichen. Diese Diskrepanz könnte auf einen systematischen Fehler zurückzuführen sein, da die Bestimmung der relativen Intensitäten auf zwei separaten Messungen basiert, wobei die Ergebnisse ab der zweiten Ordnung aus der zweiten Messung stammen.

Ein wesentlicher Faktor für die Abweichungen könnte die Sättigung des Hauptmaximums bei der Messung sein, was zu einer verfälschten Skalierung der übrigen Werte führt. Darüber hinaus könnte die asymmetrische Geometrie des Spalts zu leicht

erhöhten Intensitäten auf der rechten Seite des Hauptmaximums beitragen. Jedoch sollten, wenn dies der Fall ist, alle Nebenmaxima erhöhte Intensitäten aufweisen, wodurch die relative Intensität gleich bleiben würde. Eine weitere mögliche Ursache für die Abweichungen ist eine leichte Fehlausrichtung des experimentellen Aufbaus, einschließlich des Spalts, der Lichtquelle oder des Detektors. Selbst kleine Winkelabweichungen können zu signifikanten Unterschieden in der Beugungsstruktur führen und somit die Messergebnisse beeinflussen.

Wie bereits zuvor erwähnt, stellte auch die unvollständige Scharfeinstellung ein Problem dar, insbesondere da während des Experiments ständige Schwankungen auftraten, die zu erheblichem Rauschen in den Textdateien führten. Trotz dieser Herausforderungen wurde bei der Auswertung gezielt der Untergrund subtrahiert, was jedoch paradoxerweise zu größeren Abweichungen führte. Ein positiver Aspekt ist jedoch hervorzuheben: Die aus der Eichgeraden ermittelten Positionen der Maxima stimmten fast genau mit den Vorhersagen der Sinc-Funktion überein, was die mathematische Beschreibung der Intensitätsverteilung des Einzelspalts bestätigt.

Bei der anschließenden Untersuchung wurden die Positionen der Maxima und Minima genutzt, um die Spaltbreite zu bestimmen. Interessanterweise ergab sich ein Wert, der im Vergleich zu dem Wert, der durch die Analyse der modifizierten Objektbilder generiert wurde, zu niedrig erschien. Eine mögliche Ursache für diese Diskrepanz könnte darin liegen, dass einige Ordnungen der Minima während der Messung nicht aufgenommen wurden. Dies hätte zur Folge, dass die Berechnung der „vermeintlichen“ Minima eine höhere Steigung der Eichgerade verursacht. Da die Spaltbreite umgekehrt proportional zur Steigung der Eichgerade ist, würde eine höhere Steigung somit zu einer fälschlicherweise niedrigeren Spaltbreiten führen.

Bei der Untersuchung der Doppelspalt-Beugungsfigur zeigte sich eine hohe Übereinstimmung mit der Theorie, besonders in der Identifikation der Minima und Einhüllenden. Die ersten Intensitätsmaxima stimmten weitgehend mit den Berechnungen überein. Leichte Abweichungen von 2.7σ in der ersten und zweiten Ordnung deuten jedoch auf mögliche Ungenauigkeiten bei höheren Ordnungen hin, die nähere Betrachtung erfordern.

Nachdem die Position der Kamera angepasst wurde, um die Objektbilder zu analysieren, ergab sich eine Spaltbreite von $(207.92 \pm 18.40) \mu m$. Diese Messung weist eine signifikante Abweichung von 4.09σ im Vergleich zur zuvor ermittelten Spaltbreiten auf. Zusätzlich wurden die Objektbilder durch das Verstellen des Analysierspalts modifiziert, was zu Bildern führte, die grob den theoretischen Erwartungen entsprachen.

Obwohl die Positionen der Maxima und Minima des modifizierten Einzelspaltbildes ungefähr übereinstimmten, waren die relativen Intensitäten, insbesondere in höheren Ordnungen, stark abweichend. Besonders auffällig waren die Abweichungen bei den lokalen Extrema in der Bildmitte. Eine mögliche Erklärung für diese Abweichungen haben wir vorher schon erwähnt, dass wir bei jeder Messung die Belichtungszeit an-

passen müssen, um ein scharfes Bild zu erhalten, da eine Schafstellung nicht möglich ist, was die Konsistenz der Ergebnisse beeinträchtigen könnte.

Des Weiteren könnte die Art und Weise, wie verschiedene Fourier-Ordnungen in der Bildmitte überlagert sind und dadurch ein flaches Signal erzeugen, zu einer erhöhten Anfälligkeit für Störungen geführt haben. Darüber hinaus haben wir die Fehler bisschen überschätzt: Ein angesetzter Fehler von 0.15 Counts, der fast die Hälfte des gemessenen Wertes ausmacht, könnte dazu beigetragen haben, dass die experimentellen Ergebnisse weniger aussagekräftig sind.

Bei der Untersuchung des Doppelspalts wurde der Analysierspalt so geschlossen, dass zwei Gauß kurven auftauchten bis schließlich die Doppelspaltstruktur verschwand. Im Vergleich mit der theoretischen Analysierspalt $n = 0.27$, mit der die Doppelstruktur theoretisch gerade nicht mehr erkennbar ist, ist die ermittelte n_b zu niedrig. Und das errechnete n_b übereinstimmte außerdem nicht mit dem theoretischen Wert n_b' übereinstimmte, der durch iteratives Annähern bestimmt wurde: Wir haben den Analysierspalt zu weit geschlossen. Die beobachteten Gauß-Kurven waren zudem weniger ausgeprägt als die theoretisch berechneten modifizierten Objektbilder. Auffällig war auch, dass bei der Messung ein Peak kleiner als der andere ausfiel, was durch die intrinsischen Unregelmäßigkeiten des Doppelspalts erklärt werden könnte wie Einzelpunkt. Dennoch schienen die Messungen eine gute Näherung an das theoretische Modell zu bieten.

Der Versuchsaufbau ermöglichte es, die Beziehung zwischen Objektbild und Beugungsfigur effektiv darzustellen, trotzdem die Datenerfassung dementsprechend sehr unkompliziert ist und viel Aufwand erfordert. Vorschlag dafür ist, dass man bevor dem Experiment einen zusätzlichen Versuch oder Videoanleitung bietet, sodass wir einen besseren Überblick über dem Programm bekommen, statt verzweifelt 50 Seiten lang Anleitungen durchzulesen. Außerdem würden wir vom Herzen wünschen, dass man die Schiene für optische Anordnung bisschen verlängern, sodass wir bevor dem Experiment wirklich eine Scharfeinstellung durchführen können. Insgesamt können wir sagen, dass die experimentellen Ergebnisse, wenn auch nicht immer quantitativ, so doch qualitativ gut durch die theoretischen Berechnungen beschrieben werden können.

5 Anhang

Versuch 233 Python Code

December 28, 2023

0.0.1 Versuch 233 Python Code

1. Eichung der Abzisse

```
[521]: import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
#%matplotlib notebook
import pandas as pd
from scipy.signal import find_peaks_cwt, find_peaks
from scipy.integrate import quad
from math import log10, floor

[522]: from math import log10, floor

def round_significant_digit(x):
    """Input error
    Return tuple in form of (error rounded to significant digit, decimal places
    of significant digit)"""
    first_dig = -int(floor(log10(abs(x))))
    if x // (10**floor(log10(x))) <= 2:
        return (round(x, first_dig+1),first_dig+1)
    else:
        return (round(x, first_dig),first_dig)

def latex_table(data):
    """Input 2d-array-like
    Return latex table in string format
    Entries are centered, border for all cells
    Does not escape special characters"""
    latex_string = "\\begin{tabular}{|"
    for i in range(len(data[0])):
        latex_string += "c|"
    latex_string += "}\n\\hline\n"
    latex_string += " \\\\".join([" & ".join([str(a) for a in row]) for row in data])
    latex_string += " \\\\".join("\\hline\n") + "\\end{tabular}"
    return latex_string
```

Python Code 1

```
[523]: # Messwerte plotten
#Messwerte, s:Position Analysierspalt
pixel = np.array([358, 527, 701, 854, 1027])
s = np.array([1.783, 1.375, 1.048, 0.700, 0.352])
s_err = np.array([0.05, 0.04, 0.03, 0.02, 0.01])
pixel_err=np.array([5.5,5.5,5.5,5.5,5.5])

#Gerade anfitten
def linear (x,a,b):
    return a*x+b

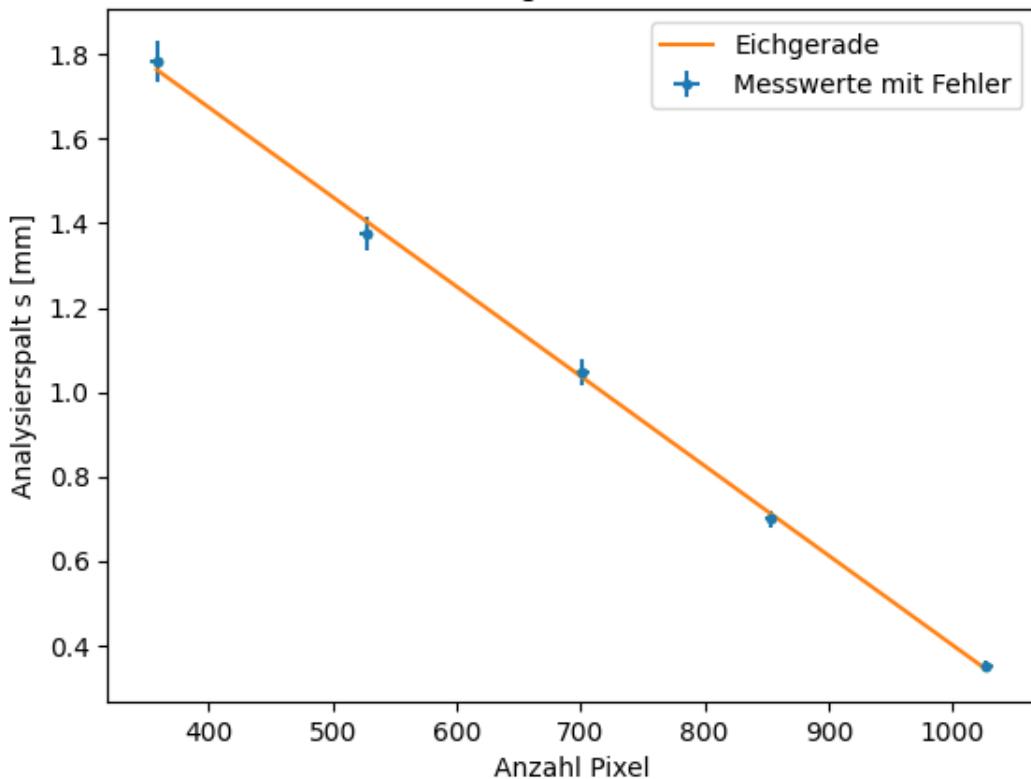
popt, pcov, = curve_fit(linear, pixel, s)
# Fehler der Steigung berechnen
steigungsfehler = np.sqrt(pcov[0, 0])

# Fehlerbalken plotten
plt.errorbar(pixel, s, xerr=pixel_err, yerr=s_err, fmt='.', label='Messwerte mit Fehler')
plt.plot(pixel, linear(pixel, *popt), label='Eichgerade')

#Plotten
plt.plot(pixel,s,linestyle='None', marker='.')
plt.xlabel('Anzahl Pixel')
plt.ylabel('Analysierspalt s [mm]')
plt.title('Eichung der Abzisse')
plt.legend()
plt.show()

print("Steigung:", popt[0], "mm/pixel")
print("Fehler der Steigung:", steigungsfehler, "mm/pixel")
```

Eichung der Abzisse



Steigung: -0.002123913353573366 mm/pixel

Fehler der Steigung: 4.400525540205145e-05 mm/pixel

Python Code 2

Daten importieren (nicht in Sättigung)

```
[524]: # Lade Datei mit vollem Pfad
pix, intensity = np.loadtxt("text nicht sättigung.txt", unpack=True, skiprows=4)

# Aus Eichung: Abzisse von Pixel in mm umrechnen
x=pix*(-popt[0])

# Manuelle Korrektur
peakind_single_0=find_peaks_cwt(intensity, np.arange(30,80),noise_perc=2)[3:-2]

# Minima finden, indem man Intensität negiert
peakind_minima = find_peaks_cwt(-intensity, np.arange(20, 60), noise_perc=20)[1:-1]
```

```

#Daten plotten
plt.plot(x,intensity, color='red', linewidth=1)
plt.scatter(x[peakind_single_0], intensity[peakind_single_0])
plt.xlabel('x/mm')
plt.ylabel('Intensitaet / +' b.E.')
plt.title('Fraunhofer Beugung am Spalt:')

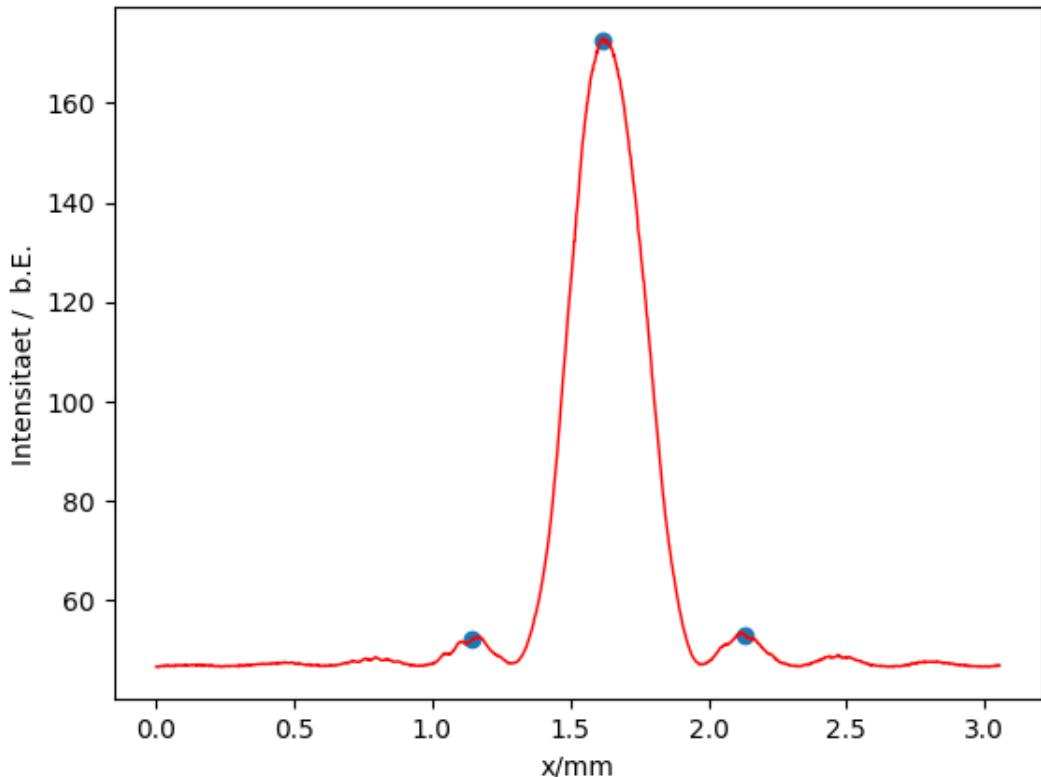
# Ausgabe der Peak-Positionen
print("Positionen der Maxima (x/mm):")
for peak in peakind_single_0:
    print(x[peak])
    print("Position (x/mm):", x[peak], ", Intensität (b.E.):", intensity[peak])

```

Positionen der Maxima (x/mm):

1.1425010782854386
 Position (x/mm): 1.1425010782854386 , Intensität (b.E.): 52.133159
 1.6160656634701838
 Position (x/mm): 1.6160656634701838 , Intensität (b.E.): 172.69952
 2.1321023138994035
 Position (x/mm): 2.1321023138994035 , Intensität (b.E.): 53.064809

Fraunhofer Beugung am Spalt:



```

[525]: # Lade Datei mit vollem Pfad
pix, intensity = np.loadtxt("andere Ordnung text.txt", unpack=True, skiprows=4)

#Aus Eichung: Abzisse von Pixel in mm umrechnen
x=pix*(-popt[0])

# Manuelle Korrektur
peakind_single_0=find_peaks_cwt(intensity, np.arange(20,60),noise_perc=20)[1:-1]

intensity=np.min([intensity,np.ones(len(intensity))*200],axis=0) # :3

# Minima finden, indem man Intensität negiert
peakind_minima = find_peaks_cwt(-intensity, np.arange(40, 100), noise_perc=100)

#Das rechteste Minimum manuell plotten, da Python nicht in der Lage ist
x_punkt = 2.98
y_punkt = 75
plt.scatter(x_punkt, y_punkt, color='green', s=30)

#Das linkeste Maximum manuell plotten, da Python nicht in der Lage ist
x_punkt = 0.1
y_punkt = 80
plt.scatter(x_punkt, y_punkt, color='red', s=30)

#Daten plotten
plt.scatter(x[peakind_minima], intensity[peakind_minima], label='Minima', color='green')
plt.plot(x,intensity, color='blue', linewidth=1)
plt.scatter(x[peakind_single_0], intensity[peakind_single_0], color='red')
plt.xlabel('x/mm')
plt.ylabel('Intensität / +' b.E.')
plt.title('Fraunhofer Beugung am Spalt:')

# Ausgabe der Peak-Positionen
print("Positionen der Maxima (x/mm):")
for peak in peakind_single_0:
    print(x[peak])
    print("Position (x/mm):", x[peak], ", Intensität (b.E.):", intensity[peak])

print("\nPositionen der Minima:")
for peak in peakind_minima:
    print("Position (x/mm):", x[peak], ", Intensität (b.E.):", intensity[peak])

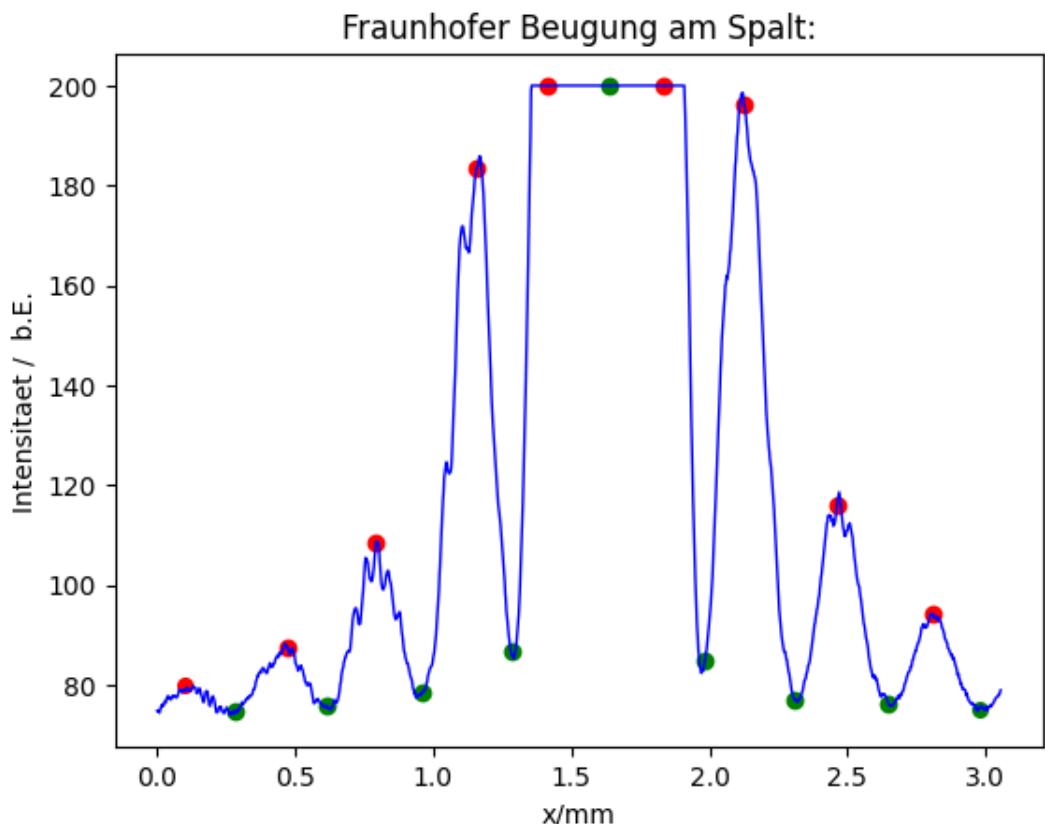
```

Positionen der Maxima (x/mm):
0.4735594877926965
Position (x/mm): 0.4735594877926965 , Intensität (b.E.): 87.444237
0.7942208320446025

Position (x/mm): 0.7942208320446025 , Intensität (b.E.): 108.62391
1.1594774752422834
Position (x/mm): 1.1594774752422834 , Intensität (b.E.): 183.65559
1.416431273844949
Position (x/mm): 1.416431273844949 , Intensität (b.E.): 200.0
1.8369011473429895
Position (x/mm): 1.8369011473429895 , Intensität (b.E.): 200.0
2.1257086975311417
Position (x/mm): 2.1257086975311417 , Intensität (b.E.): 196.23948
2.4612351839099063
Position (x/mm): 2.4612351839099063 , Intensität (b.E.): 115.90625
2.8095030264967575
Position (x/mm): 2.8095030264967575 , Intensität (b.E.): 94.233971

Positionen der Minima:

Position (x/mm): 0.2866839978998489 , Intensität (b.E.): 74.802532
Position (x/mm): 0.6179632735063394 , Intensität (b.E.): 75.823862
Position (x/mm): 0.9619840752339848 , Intensität (b.E.): 78.44966
Position (x/mm): 1.2826454194858905 , Intensität (b.E.): 86.645615
Position (x/mm): 1.6351605578015502 , Intensität (b.E.): 200.0
Position (x/mm): 1.983428506584069 , Intensität (b.E.): 84.898204
Position (x/mm): 2.308336997890849 , Intensität (b.E.): 77.072821
Position (x/mm): 2.648110673802754 , Intensität (b.E.): 76.181059



Python Code 3

```
[526]: #Beugungsfunktion Einzel-Spalt
def beugung_spalt(x):
    return np.sinc(x)**2

#Beugungsfunktion Doppelspalt
def beugung_doppelspalt(x):
    return np.sinc(x)**2*np.cos(np.pi*v*x)**2

#Anzahl a der Nebenmaxima die dargestellt werden sollen:
a=2
#Verhaeltnis Spaltabstand zu Spaltbreite:

g=626
d=258.2
v = g/d

x=np.linspace(-(a+1),a+1,200)
```

```

# Finden der Peaks für die Doppelspalt-Beugungsfunktion
peaks, _ = find_peaks(beugung_doppelspalt(x))
# Hervorheben der Peaks für den Doppelspalt
plt.scatter(x[peaks], beugung_doppelspalt(x)[peaks], color='red', label='Peaks_{\rightarrow(Doppelspalt)}', s=20)

plt.plot(x,beugung_spalt(x),label='Spalt')
plt.plot(x,beugung_doppelspalt(x),label='Doppelspalt')
plt.xlabel('$\pi x$')
plt.ylabel('Intensität / b.E.')
plt.title('Beugung am Spalt und Doppelspalt mit Peaks')
plt.ylim((0,1.1))
plt.legend()
plt.savefig("Beugung_spalt_doppelspalt.pdf", format="pdf")

# Ausgabe der Peaks
print("Peaks des Doppelspalts: ", x[peaks])

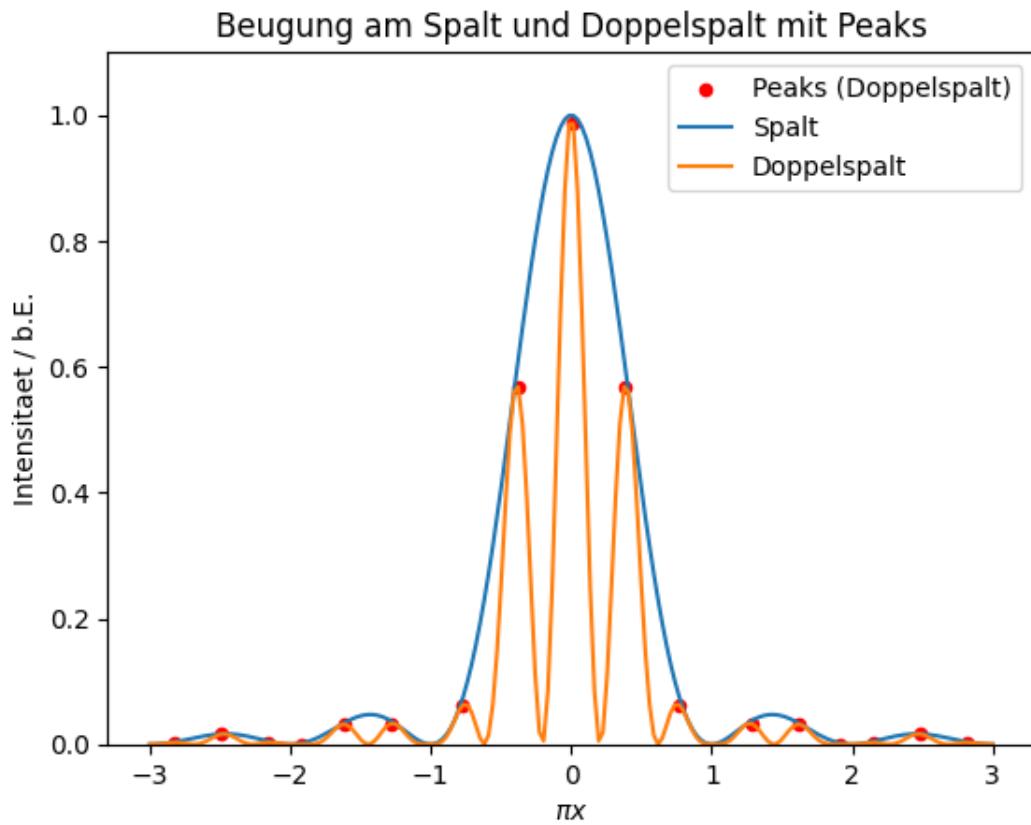
```

```

<>:26: SyntaxWarning: invalid escape sequence '\p'
<>:26: SyntaxWarning: invalid escape sequence '\p'
C:\Users\shiy0\AppData\Local\Temp\ipykernel_13600\3218391075.py:26:
SyntaxWarning: invalid escape sequence '\p'
    plt.xlabel('$\pi x$')

Peaks des Doppelspalts: [-2.81909548 -2.48743719 -2.15577889 -1.91457286
-1.61306533 -1.28140704
-0.76884422 -0.37688442  0.01507538  0.37688442  0.76884422  1.28140704
 1.61306533  1.91457286  2.15577889  2.48743719  2.81909548]

```



```
[527]: single_int_theo=lambda x : np.sinc(x)**2
x=np.linspace(-10,10,100000)
#Dieser Befehl erstellt ein Array x mit 100.000 linear verteilten Punkten
#zwischen -10 und 10. Diese Punkte dienen als Basis für die Berechnung der
#theoretischen Intensitätsverteilung.
peakind=find_peaks_cwt(single_int_theo(x), np.arange(1,30), noise_perc=10)
#np.arange(1,30) gibt die Breiten der Wavelets an, die zur Erkennung der Peaks
#verwendet werden. In diesem Fall werden Wavelets verschiedener Breiten von 1
#bis 29 verwendet.

peaks=single_int_theo(x[peakind])[9:16]
I_rel_theo=peaks/peaks[0]
print(I_rel_theo)
plt.figure(figsize=(8,6))
plt.plot(x, single_int_theo(x), label="Theoretische Intensität")

plt.scatter(x[peakind], single_int_theo(x[peakind]), color='red',label="Maxima")
#markiert die gefundenen Maxima auf der Intensitätskurve

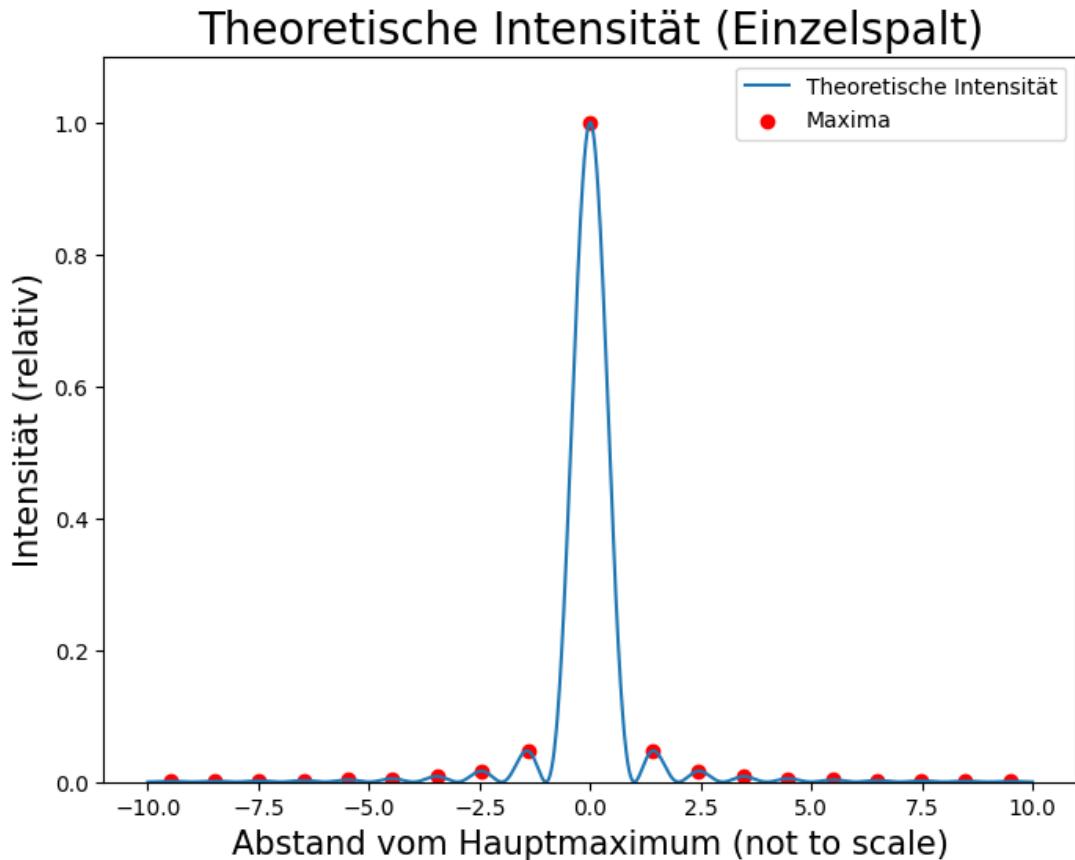
plt.title("Theoretische Intensität (Einzelspalt)", fontsize=20)
```

```

plt.xlabel("Abstand vom Hauptmaximum (not to scale)", fontsize=15)
plt.ylabel("Intensität (relativ)", fontsize=15)
plt.legend()
plt.ylim((0,1.1))
plt.savefig('Fourier_single_theo.png')
plt.show()

```

[1. 0.04664596 0.01642654 0.00832688 0.00502388 0.00335872
0.0024028]



Python Code 4

[528]:

```

I_0 = 172.69952-46.7
I_1 = 52.133159-46.7
d_I_1 = 20
d_I_0= 20

k = I_1 / I_0

```

```

d_k = k*np.sqrt((d_I_1/I_0)**2+(d_I_1*d_I_0/I_0**2)**2)

print('relative Verhältnis', k)
print('relative Verhältnis Fehler', d_k)

```

relative Verhältnis 0.04312047379228108
relative Verhältnis Fehler 0.006930235058960626

[529]:

```

I_n=87.44237-75.051674
I_1_a=183.65559-75.051674
d_I=10

ka=I_n/I_1_a*k
d_ka=ka*np.sqrt((d_I/I_0)**2+(d_I/I_1_a)**2+(d_I/I_n)**2+(d_I/I_1)**2)

print('Verhältnis anderer Ordnung', ka)
print('Verhältnis anderer Ordnung Fehler', d_ka)

```

Verhältnis anderer Ordnung 0.004919644721983338
Verhältnis anderer Ordnung Fehler 0.003473609352716678

Python Code 5

[530]:

```

ordnungen = np.array([1, 2, 3, 4, 5])
minima = np.array([603.90666, 452.93, 290.955, 134.97914, 2.98])-75.051674
maxima = np.array([760.89058, 545.91562, 373.9422, 222.96554, 50])-75.051674

# Lineare Funktion definieren
def linear(x, a, b):
    return a * x + b

# Lineare Anpassung (Fit) für Minima
popt, pcov = curve_fit(linear, ordnungen, minima)

# Maxima in Bezug auf die lineare Anpassung berechnen
ordnungen_max = (maxima - popt[1]) / popt[0]

# Plotten der Messdaten und des Fits
plt.plot(ordnungen, minima, linestyle='None', marker='o', color='blue', □
          ↴markerSize=2, label="Minima")
plt.plot(np.arange(-0.2, max(ordnungen)+1), linear(np.arange(-0.2, □
          ↴max(ordnungen)+1), *popt), color='orange', label="Fit für Minima") # □
          ↴Eichgerade verlängert

# Die Maxima und berechneten Ordnungen hervorheben
plt.scatter(ordnungen_max, maxima, color='green', label="Maxima", zorder=5, □
          ↴s=5) # Verkleinerte Punkte für Maxima

```

```

# Fehlerbalken hinzufügen (angenommener Fehler)
error = 10 # Aangenommener Fehler
plt.errorbar(ordnungen, minima, yerr=error, fmt='o', color='blue', capsize=3,
             markersize=3)
plt.errorbar(ordnungen_max, maxima, yerr=error, fmt='o', color='green', capsize=3,
             markersize=3)

# Plot beschriften
plt.xlabel('Ordnungen')
plt.ylabel('Position (Pixel)')
plt.title('Lineare Anpassung mit Maxima und Minima')
plt.legend()

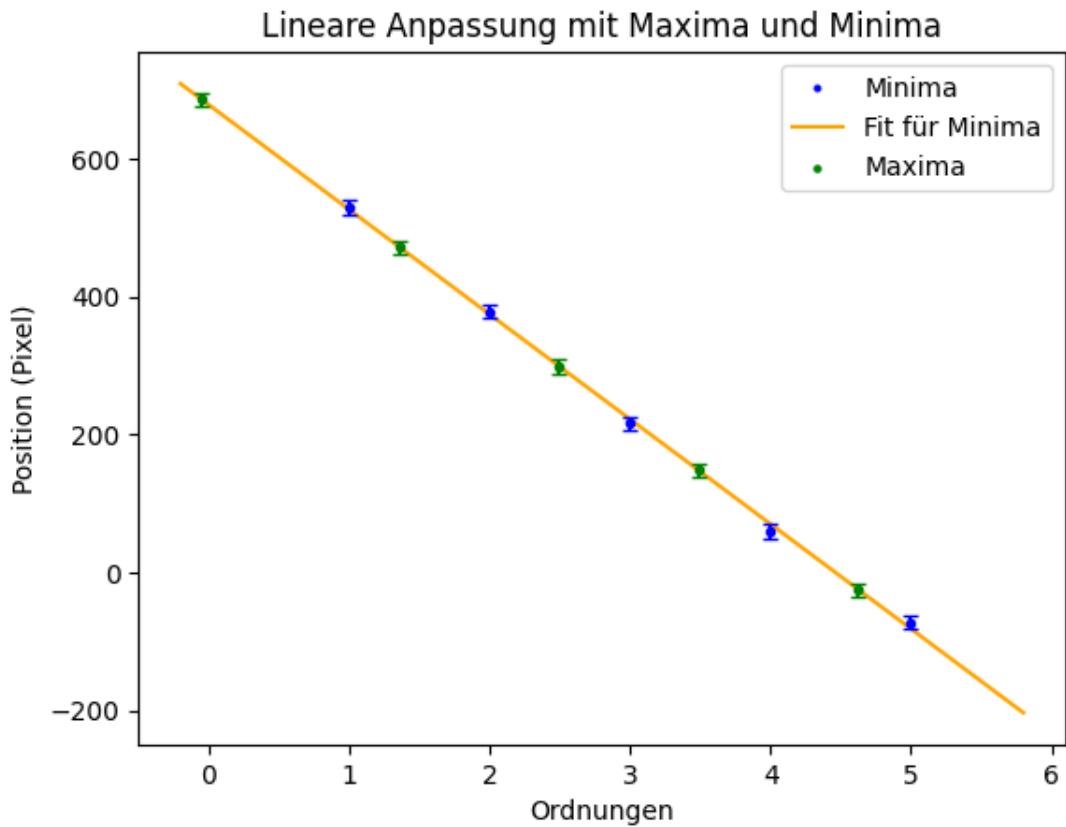
# Plot anzeigen
plt.show()

# Ausgabe
print(ordnungen_max)

# Berechnung der Steigung und ihres Fehlers
steigung = popt[0]
steigungsfehler = np.sqrt(pcov[0, 0])

print(steigung, steigungsfehler)

```



```
[ -0.05131691  1.36317426  2.49472412  3.48811959  4.6261974 ]
-151.9804180235707 2.9458278933329103
```

```
[531]: # Datei laden
pix, intensity = np.loadtxt("andere Ordnung text.txt", unpack=True, skiprows=4)

# Peaks (Maxima) finden
peakind_single_0 = find_peaks_cwt(intensity, np.arange(20,60), noise_perc=20)[1:-1]

# Intensität begrenzen
intensity = np.minimum(intensity, 200)

# Minima finden, indem man Intensität negiert
peakind_minima = find_peaks_cwt(-intensity, np.arange(40, 100), noise_perc=100)

# Manuelle Punktpplot
plt.scatter(2.98, 75, color='green', s=30) # Rechtestes Minimum
plt.scatter(50, 80, color='red', s=30)      # Linkstes Maximum
```

```

# Daten plotten
plt.scatter(pix[peakind_minima], intensity[peakind_minima], label='Minima', ↴
            color='green')
plt.plot(pix, intensity, color='blue', linewidth=1)
plt.scatter(pix[peakind_single_0], intensity[peakind_single_0], color='red')

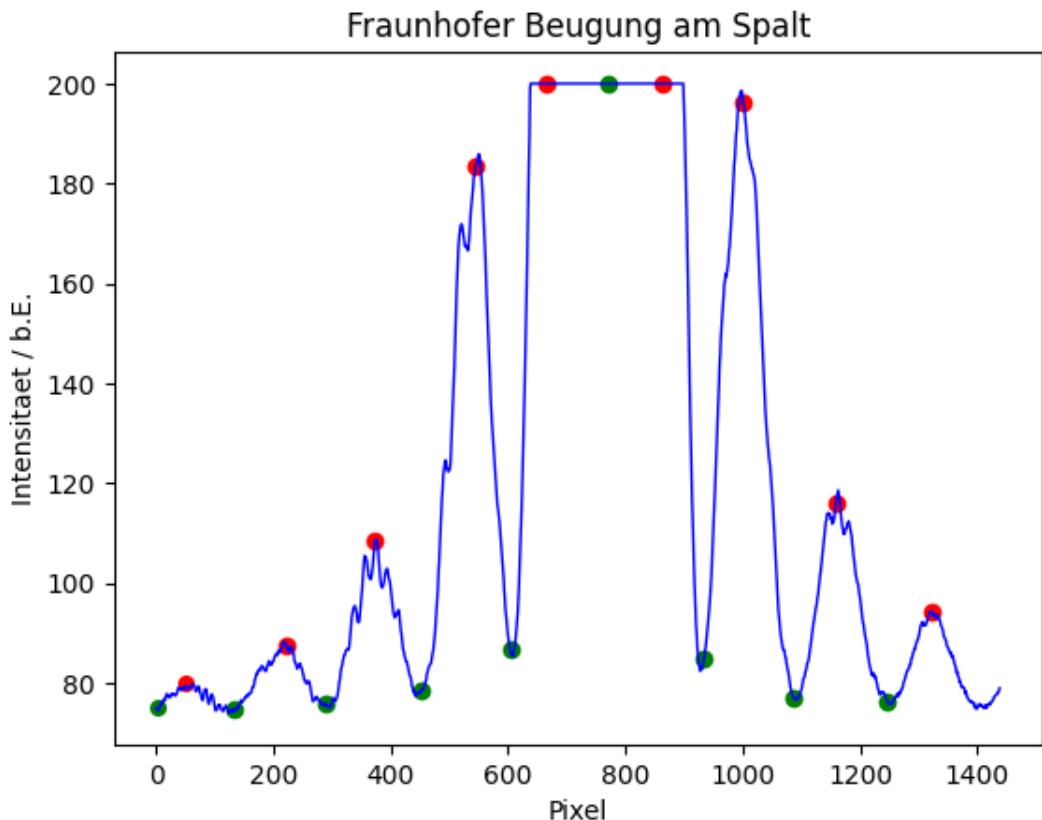
# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensitaet / b.E.')
plt.title('Fraunhofer Beugung am Spalt')

# Plot anzeigen
plt.show()

# Ausgabe der Peak-Positionen
print("Positionen der Maxima (Pixel):")
for peak in peakind_single_0:
    print("Position (Pixel):", pix[peak], ", Intensität (b.E.):", ↴
          intensity[peak])

print("\nPositionen der Minima:")
for peak in peakind_minima:
    print("Position (Pixel):", pix[peak], ", Intensität (b.E.):", ↴
          intensity[peak])

```



Positionen der Maxima (Pixel):

Position (Pixel): 222.96554 , Intensität (b.E.): 87.444237
 Position (Pixel): 373.9422 , Intensität (b.E.): 108.62391
 Position (Pixel): 545.91562 , Intensität (b.E.): 183.65559
 Position (Pixel): 666.89692 , Intensität (b.E.): 200.0
 Position (Pixel): 864.86633 , Intensität (b.E.): 200.0
 Position (Pixel): 1000.8453 , Intensität (b.E.): 196.23948
 Position (Pixel): 1158.8209 , Intensität (b.E.): 115.90625
 Position (Pixel): 1322.7955 , Intensität (b.E.): 94.233971

Positionen der Minima:

Position (Pixel): 134.97914 , Intensität (b.E.): 74.802532
 Position (Pixel): 290.95503 , Intensität (b.E.): 75.823862
 Position (Pixel): 452.93 , Intensität (b.E.): 78.44966
 Position (Pixel): 603.90666 , Intensität (b.E.): 86.645615
 Position (Pixel): 769.88101 , Intensität (b.E.): 200.0
 Position (Pixel): 933.85566 , Intensität (b.E.): 84.898204
 Position (Pixel): 1086.832 , Intensität (b.E.): 77.072821
 Position (Pixel): 1246.8073 , Intensität (b.E.): 76.181059

```
[532]: import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks_cwt

# Datei laden
pix, intensity = np.loadtxt("text nicht sättigung.txt", unpack=True, skiprows=4)

# Peaks (Maxima) finden
peakind_single_0 = find_peaks_cwt(intensity, np.arange(30, 80), noise_perc=2)[3:-2]

# Minima finden, indem man Intensität negiert
peakind_minima = find_peaks_cwt(-intensity, np.arange(20, 60), noise_perc=20)[1:-1]

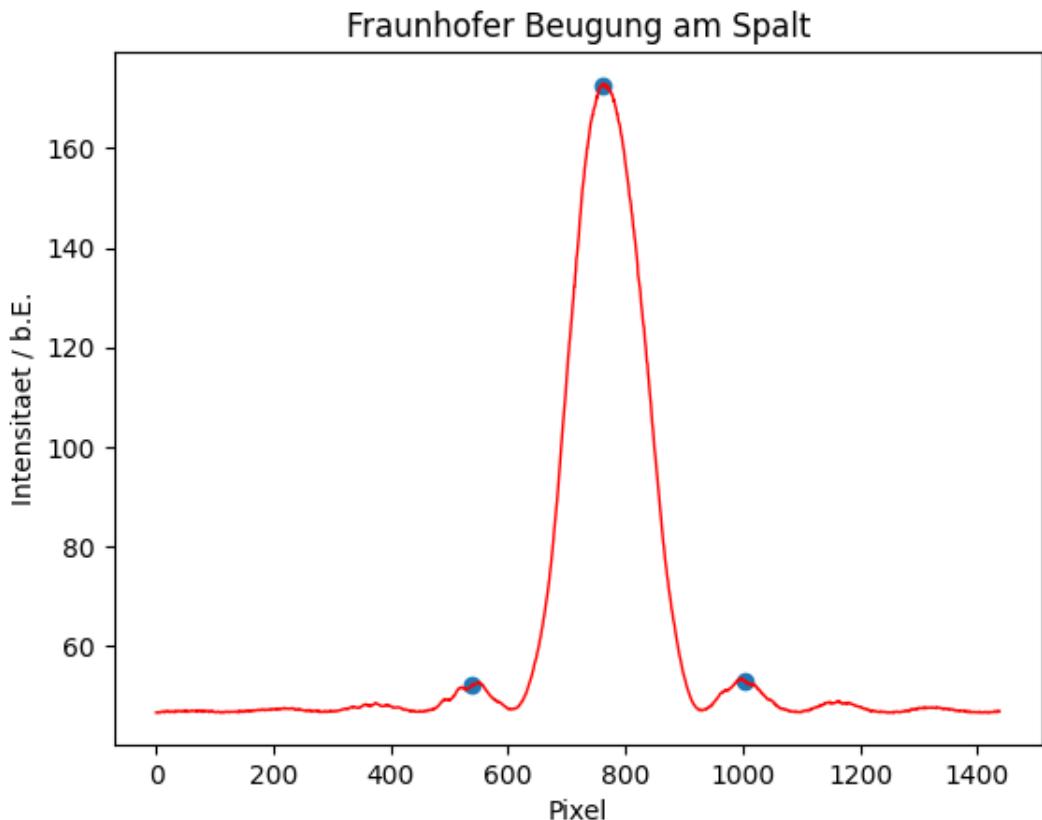
# Daten plotten
plt.plot(pix, intensity, color='red', linewidth=1)
plt.scatter(pix[peakind_single_0], intensity[peakind_single_0])

# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensitaet / b.E.')
plt.title('Fraunhofer Beugung am Spalt')

# Plot anzeigen
plt.show()

# Ausgabe der Peak-Positionen
print("Positionen der Maxima (Pixel):")
for peak in peakind_single_0:
    print("Position (Pixel):", pix[peak], ", Intensität (b.E.):", intensity[peak])

print("\nPositionen der Minima:")
for peak in peakind_minima:
    print("Position (Pixel):", pix[peak], ", Intensität (b.E.):", intensity[peak])
```



Positionen der Maxima (Pixel):

Position (Pixel): 537.92264 , Intensität (b.E.): 52.133159

Position (Pixel): 760.89058 , Intensität (b.E.): 172.69952

Position (Pixel): 1003.8556 , Intensität (b.E.): 53.064809

Positionen der Minima:

Position (Pixel): 294.95758 , Intensität (b.E.): 46.825681

Position (Pixel): 461.93357 , Intensität (b.E.): 47.063422

Position (Pixel): 663.90453 , Intensität (b.E.): 67.61361

Position (Pixel): 887.87232 , Intensität (b.E.): 61.290919

Position (Pixel): 1077.845 , Intensität (b.E.): 47.149251

Python Code 6

```
[533]: def beugung_spalt(x):
         return np.sinc(x)**2

# Beugungsfunktion Doppelspalt
def beugung_doppelspalt(x, v):
    return np.sinc(x)**2 * np.cos(np.pi * v * x)**2
```

```

# Anzahl a der Nebenmaxima, die dargestellt werden sollen
a = 2
# Verhältnis Spaltabstand zu Spaltbreite
g = 626
d = 258.2
v = g / d

x = np.linspace(-(a+1), a+1, 200)

# Finden der Peaks für die Doppelspalt-Beugungsfunktion
peaks, _ = find_peaks(beugung_doppelspalt(x, v))

# Hervorheben der Peaks für den Doppelspalt
plt.scatter(x[peaks], beugung_doppelspalt(x[peaks], v), color='red', □
             ↳label='Peaks (Doppelspalt)', s=20)

# Plotten der Beugungsfunktionen
plt.plot(x, beugung_spalt(x), label='Einzelspalt')
plt.plot(x, beugung_doppelspalt(x, v), label='Doppelspalt')
plt.xlabel('$\pi x$')
plt.ylabel('Intensität / b.E.')
plt.title('Beugung am Spalt und Doppelspalt mit Peaks')
plt.ylim((0, 1.1))
plt.legend()

# Ausgabe der Peaks
print("Peaks des Doppelspalts: ", x[peaks])
for peak in peaks:
    print(f"Position (x/pi): {x[peak]}, Intensität: □
          ↳{beugung_doppelspalt(x[peak], v)}")

plt.show()

```

```

Peaks des Doppelspalts: [-2.81909548 -2.48743719 -2.15577889 -1.91457286
-1.61306533 -1.28140704
-0.76884422 -0.37688442  0.01507538  0.37688442  0.76884422  1.28140704
 1.61306533  1.91457286  2.15577889  2.48743719  2.81909548]
Position (x/pi): -2.819095477386935, Intensität: 0.0027849477704494285
Position (x/pi): -2.4874371859296485, Intensität: 0.016198101557266358
Position (x/pi): -2.1557788944723617, Intensität: 0.0027612921284987948
Position (x/pi): -1.914572864321608, Intensität: 0.0003610314110172694
Position (x/pi): -1.6130653266331658, Intensität: 0.0316139915406444
Position (x/pi): -1.2814070351758795, Intensität: 0.03290537247037854
Position (x/pi): -0.7688442211055277, Intensität: 0.06260954859665814
Position (x/pi): -0.37688442211055273, Intensität: 0.5679901163157558
Position (x/pi): 0.015075376884421843, Intensität: 0.9861354392048531
Position (x/pi): 0.37688442211055273, Intensität: 0.5679901163157558

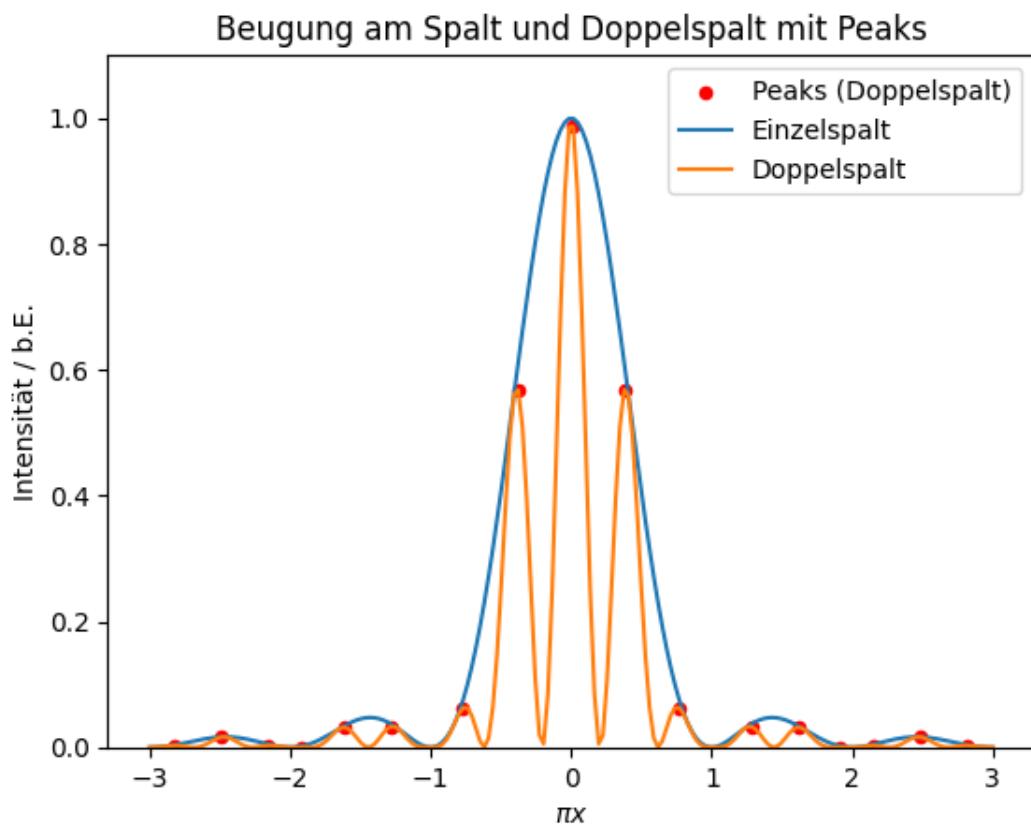
```

```

Position (x/pi): 0.7688442211055273, Intensität: 0.06260954859665827
Position (x/pi): 1.2814070351758788, Intensität: 0.03290537247037859
Position (x/pi): 1.6130653266331656, Intensität: 0.03161399154064439
Position (x/pi): 1.9145728643216078, Intensität: 0.00036103141101726915
Position (x/pi): 2.1557788944723617, Intensität: 0.0027612921284987948
Position (x/pi): 2.4874371859296476, Intensität: 0.016198101557266385
Position (x/pi): 2.8190954773869343, Intensität: 0.0027849477704494333

<>:26: SyntaxWarning: invalid escape sequence '\p'
<>:26: SyntaxWarning: invalid escape sequence '\p'
C:\Users\shiy0\AppData\Local\Temp\ipykernel_13600\3380039446.py:26:
SyntaxWarning: invalid escape sequence '\p'
    plt.xlabel('$\pi x$')

```



```
[534]: import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import savgol_filter

# Datei laden
pix, intensity = np.loadtxt("Aufgabe 2 anderer Ordnung (2).txt", unpack=True,
                           skiprows=4)
```

```

# Intensität auf einen maximalen Schwellenwert begrenzen
intensity_limited = np.minimum(intensity, 700)

# Glättung der Intensitätskurve
intensity_smooth = savgol_filter(intensity_limited, 51, 2) # Fenstergröße 51, ↴Polynomgrad 2

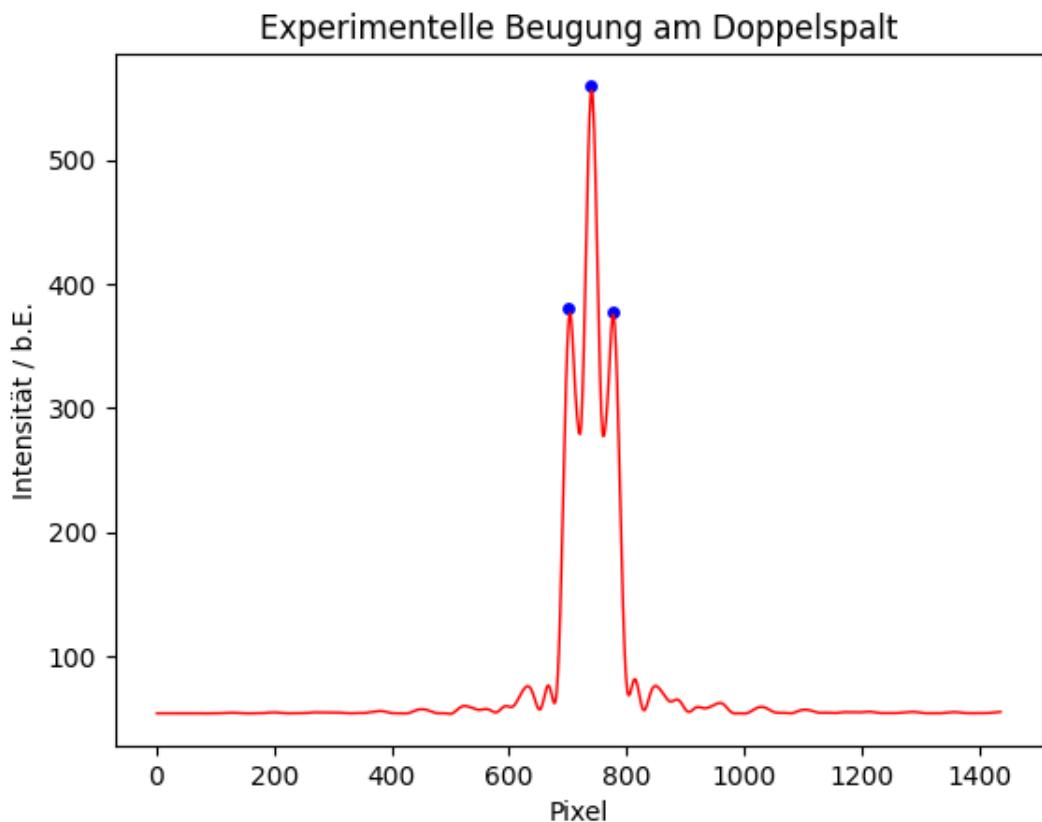
# Manuell hinzugefügte Peaks plotten
peaks = [(700, 380), (778, 378), (740, 560)]
for x_punkt, y_punkt in peaks:
    plt.scatter(x_punkt, y_punkt, color='blue', s=15)

# Daten plotten
plt.plot(pix, intensity_smooth, color='red', linewidth=1)

# Beschriftung und Titel
plt.xlabel('Pixel')
plt.ylabel('Intensität / b.E.')
plt.title('Experimentelle Beugung am Doppelspalt')

plt.show()

```



```
[535]: pix, intensity = np.loadtxt("Messung 2 symmetrisch.txt", unpack=True, u
↪skiprows=4)

# Bestimmen des zu betrachtenden Intervalls (z.B. Pixel 100 bis Pixel 300)
start_pixel = 300
end_pixel = 1200

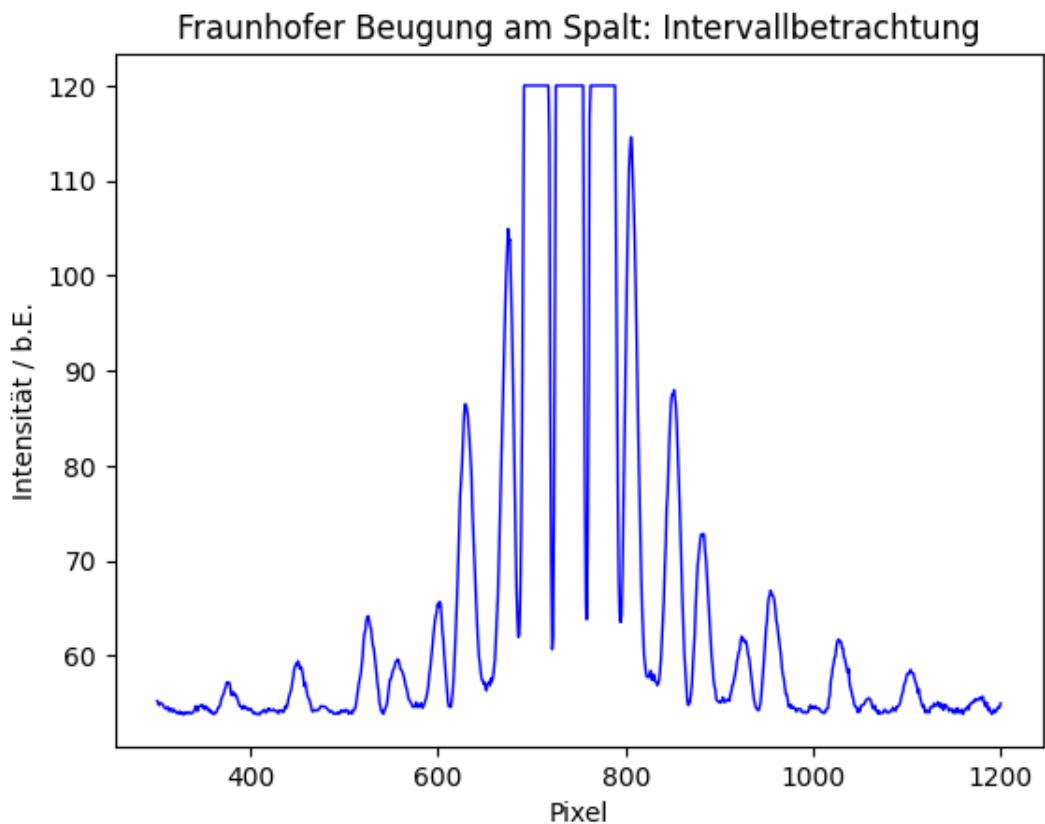
# Einschränkung des Intervalls
intervall_index = (pix >= start_pixel) & (pix <= end_pixel)
pix_intervall = pix[intervall_index]
intensity_intervall = intensity[intervall_index]

# Intensität auf einen maximalen Schwellenwert begrenzen
intensity_limited = np.minimum(intensity_intervall, 120)

# Daten plotten
plt.plot(pix_intervall, intensity_limited, color='blue', linewidth=1)

# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensität / b.E.')
plt.title('Fraunhofer Beugung am Spalt: Intervallbetrachtung')

# Plot anzeigen
plt.show()
```



```
[536]: d_j=10
j_0=560-53.9904
j_n=378-53.9904
m=j_n/j_0
d_m=m*np.sqrt((d_j/j_n)**2+(d_j/j_0)**2)
print('Verhältnis', m)
print('Fehler Verhältnis', d_m)
```

Verhältnis 0.6403230294445006
 Fehler Verhältnis 0.023466747029291298

```
[537]: j_n_a=50-42.693489
j_1_a=105-42.693489
j_1=378-53.9904

ma=j_n_a/j_1_a*m
d_ma=ka*np.sqrt((d_j/j_0)**2+(d_j/j_1_a)**2+(d_j/j_n)**2+(d_j/j_1)**2)

print('Verhältnis anderer Ordnung', ma)
print('Verhältnis anderer Ordnung Fehler', d_ma)
```

Verhältnis anderer Ordnung 0.07508889814404096
Verhältnis anderer Ordnung Fehler 0.004975515133643335

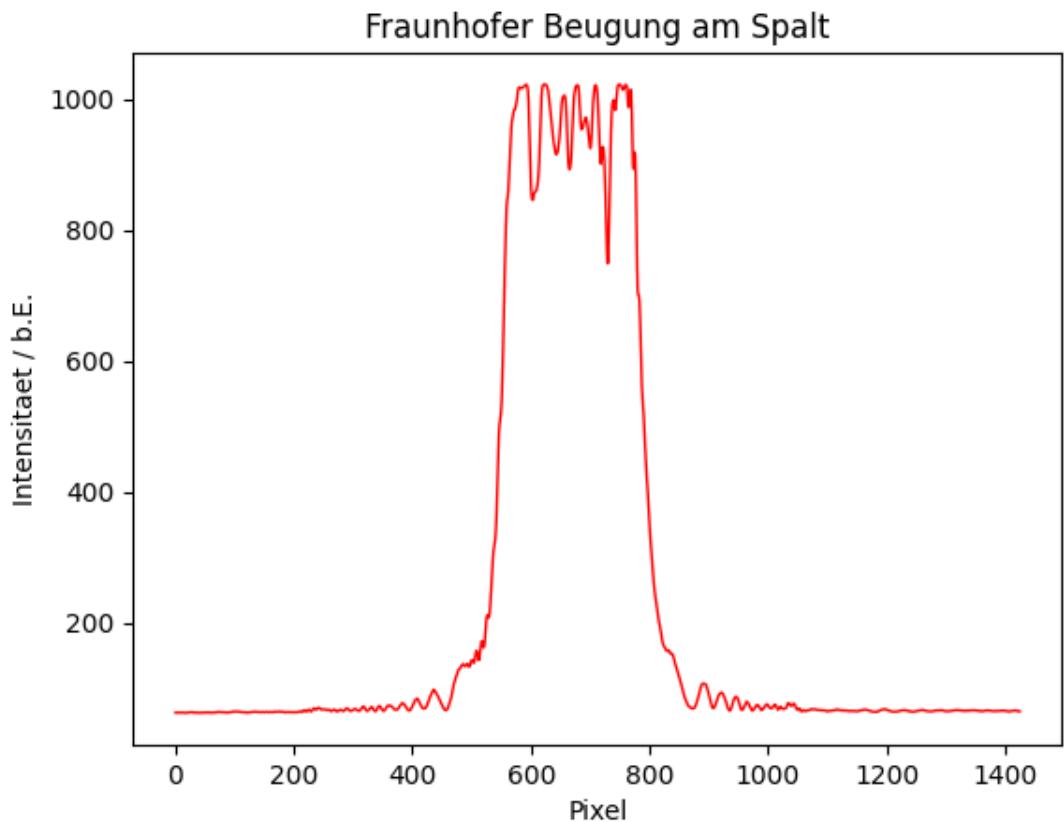
Python Code 7

```
[538]: # Datei laden
pix, intensity = np.loadtxt("Aufgabe 3 Einzelspalt.txt", unpack=True,
                           skiprows=4)

# Daten plotten
plt.plot(pix, intensity, color='red', linewidth=1)

# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensitaet / b.E.')
plt.title('Fraunhofer Beugung am Spalt')

# Plot anzeigen
plt.show()
```



[539]: # Gesamter Code für die Berechnung des modifizierten Objektbildes eines Spalts
→ und die Bestimmung der Intensitäten der Peaks

```
from scipy.integrate import quad
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks

# Anpassung der spalt-Funktion
def spalt(k, d, y):
    return d / np.pi * np.sin(k * d / 2) / (k * d / 2) * np.cos(y * k)

d = 1 # Spaltbreite auf 1 setzen
n = 4 # Integrationsgrenze

# y-Werte:
Y = np.linspace(-1, 1, 200) * d

# Berechnung des resultierenden Objektbilds
f_modifiziert = []
for y in Y:
    result, error = quad(spalt, 0, 2 * np.pi * n / d, args=(d, y))
    f_modifiziert.append(result**2)

# Kurve auf 1 normieren:
f_modifiziert = f_modifiziert / np.max(f_modifiziert)

# Finden der Peaks (Extrema)
peaks, _ = find_peaks(f_modifiziert)

# Berechnung der Intensitäten der Peaks
intensities_of_peaks = f_modifiziert[peaks]

# Berechnung der Intensitäten der lokalen Minima
minima, _ = find_peaks(-np.array(f_modifiziert)) # Minima finden durch
→ Invertieren der Werte
intensities_of_minima = f_modifiziert[minima]

# Plot der Funktion mit markierten Peaks
plt.plot(Y, f_modifiziert, label='Modifiziertes Objektbild')
plt.scatter(Y[peaks], intensities_of_peaks, color='red', label='Maxima')
plt.xlabel('y / d')
plt.ylabel('Intensität / b.E.')
plt.title('Modifiziertes Objektbild eines Spalts theoretisch 3. Maximum')
plt.ylim((0, 1.1))
plt.legend()
```

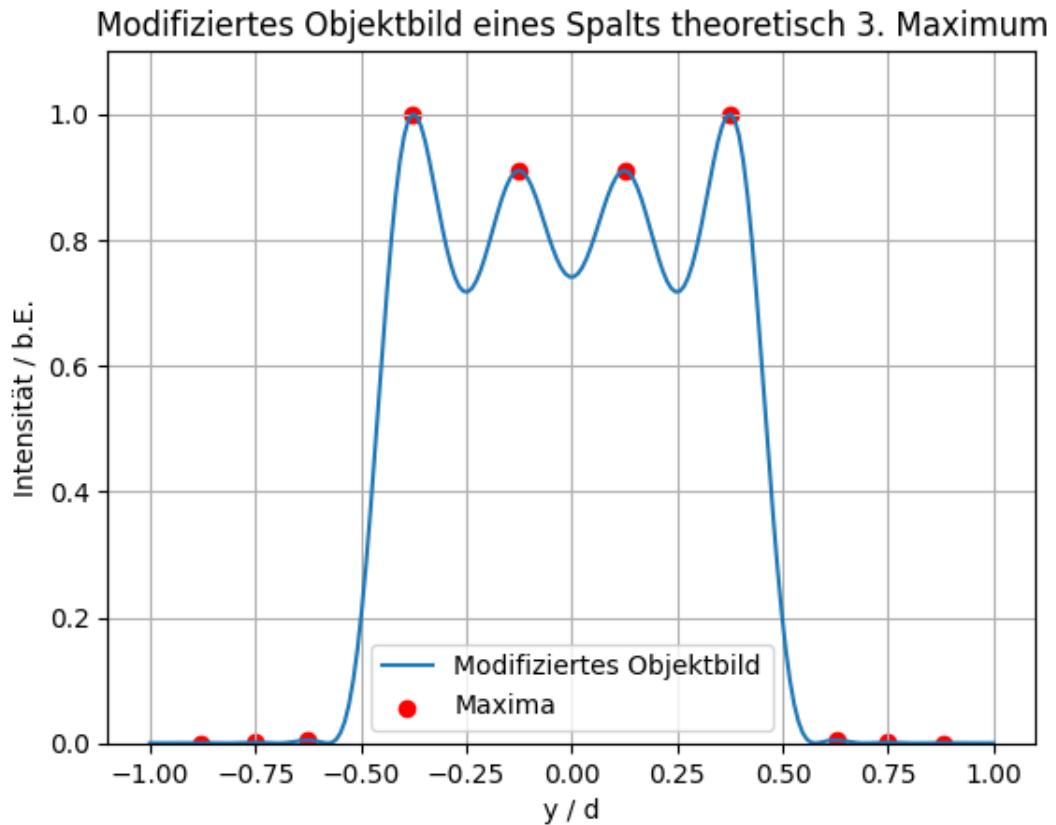
```

plt.grid(True)
plt.show()

intensities_of_peaks

print(intensities_of_minima)
print(intensities_of_peaks)

```



```

[1.11872103e-06 2.54669095e-07 9.52446360e-06 4.40692720e-07
 7.17440220e-01 7.41309563e-01 7.17440220e-01 4.40692720e-07
 9.52446360e-06 2.54669095e-07 1.11872103e-06]
[4.62950856e-04 1.21387015e-03 4.99176989e-03 1.00000000e+00
 9.10448642e-01 9.10448642e-01 1.00000000e+00 4.99176989e-03
 1.21387015e-03 4.62950856e-04]

```

```

[540]: d = 1 # Spaltbreite auf 1 setzen
n = 1 # Integrationsgrenze

# y-Werte:
Y = np.linspace(-1, 1, 200) * d

```

```

# Berechnung des resultierenden Objektbilds
f_modifiziert = []
for y in Y:
    result, error = quad(spalt, 0, 2 * np.pi * n / d, args=(d, y))
    f_modifiziert.append(result**2)

# Kurve auf 1 normieren:
f_modifiziert = f_modifiziert / np.max(f_modifiziert)

# Finden der Peaks (Extrema)
peaks, _ = find_peaks(f_modifiziert)

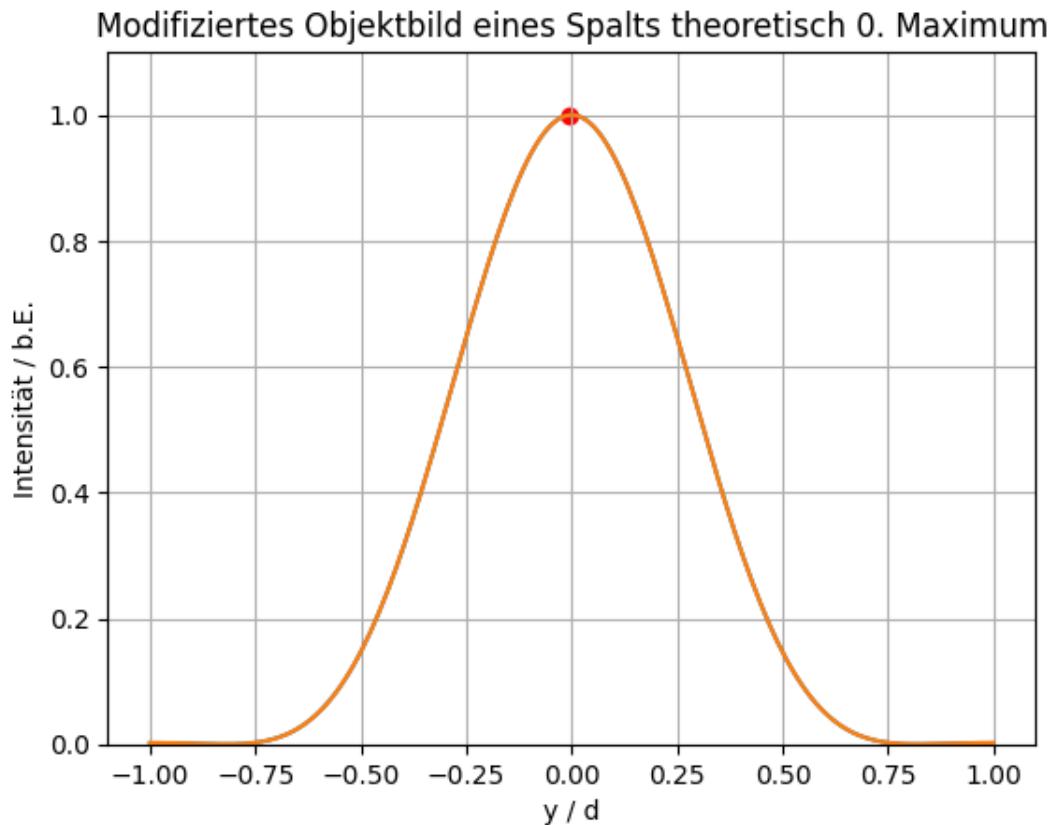
# Berechnung der Intensitäten der Peaks
intensities_of_peaks = f_modifiziert[peaks]
plt.plot(Y, f_modifiziert, label='Modifiziertes Objektbild')
plt.scatter(Y[peaks], intensities_of_peaks, color='red', label='Maxima')

# Berechnung der Intensitäten der lokalen Minima
minima, _ = find_peaks(-np.array(f_modifiziert)) # Minima finden durch ↵
    → Invertieren der Werte
intensities_of_minima = f_modifiziert[minima]

# Plot der Funktion
plt.plot(Y, f_modifiziert)
plt.xlabel('y / d')
plt.ylabel('Intensität / b.E.')
plt.title('Modifiziertes Objektbild eines Spalts theoretisch O. Maximum')
plt.ylim((0, 1.1))
plt.grid(True)
plt.show()

print(intensities_of_minima)
print(intensities_of_peaks)

```



```
[7.93916854e-09 7.93916854e-09]
```

```
[1.]
```

```
[541]: d = 1 # Spaltbreite auf 1 setzen
n = 2 # Integrationsgrenze

# y-Werte:
Y = np.linspace(-1, 1, 200) * d

# Berechnung des resultierenden Objektbilds
f_modifiziert = []
for y in Y:
    result, error = quad(spalt, 0, 2 * np.pi * n / d, args=(d, y))
    f_modifiziert.append(result**2)

# Kurve auf 1 normieren:
f_modifiziert = f_modifiziert / np.max(f_modifiziert)

# Finden der Peaks (Extrema)
peaks, _ = find_peaks(f_modifiziert)
```

```

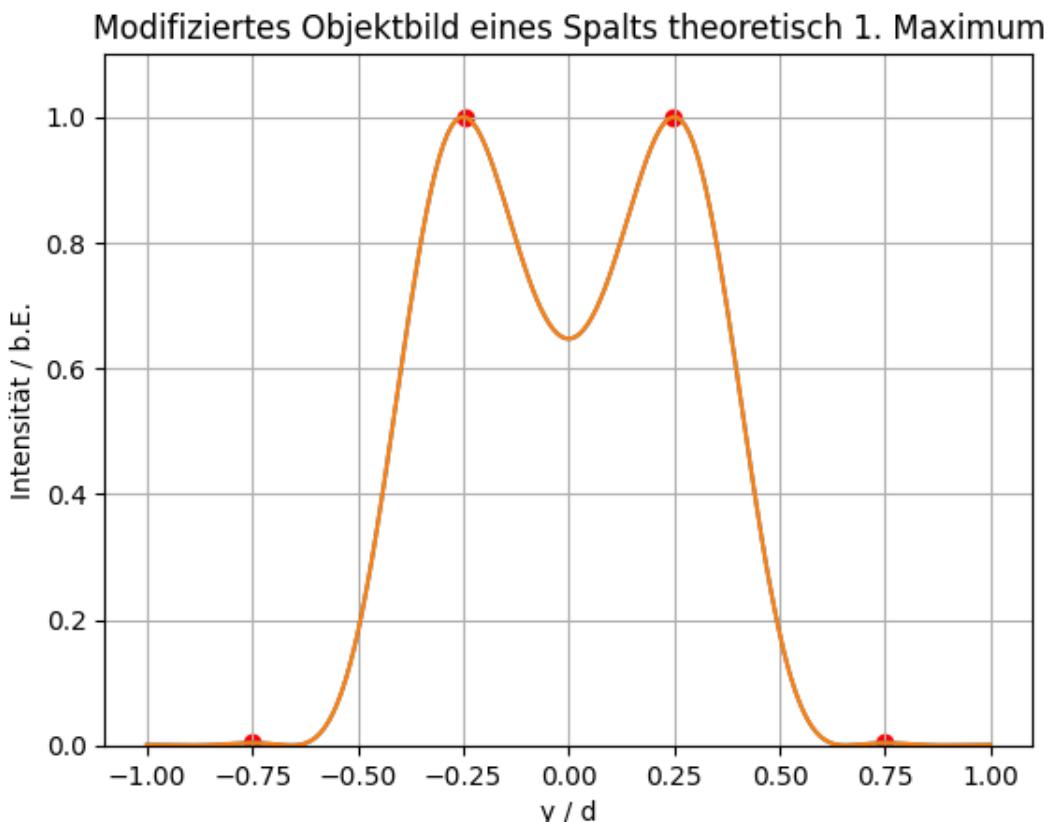
# Berechnung der Intensitäten der Peaks
intensities_of_peaks = f_modifiziert[peaks]
plt.plot(Y, f_modifiziert, label='Modifiziertes Objektbild')
plt.scatter(Y[peaks], intensities_of_peaks, color='red', label='Maxima')

# Berechnung der Intensitäten der lokalen Minima
minima, _ = find_peaks(-np.array(f_modifiziert)) # Minima finden durch ↵
    ↵ Invertieren der Werte
intensities_of_minima = f_modifiziert[minima]

# Plot der Funktion
plt.plot(Y, f_modifiziert)
plt.xlabel('y / d')
plt.ylabel('Intensität / b.E.')
plt.title('Modifiziertes Objektbild eines Spalts theoretisch 1. Maximum')
plt.ylim((0, 1.1))
plt.grid(True)
plt.show()

print(intensities_of_minima)
print(intensities_of_peaks)

```



```
[4.04086291e-06 3.08923459e-06 6.47260736e-01 3.08923459e-06
 4.04086291e-06]
[0.00381993 1.          0.00381993]
```

```
[542]: d = 1 # Spaltbreite auf 1 setzen
n = 3 # Integrationsgrenze

# y-Werte:
Y = np.linspace(-1, 1, 200) * d

# Berechnung des resultierenden Objektbilds
f_modifiziert = []
for y in Y:
    result, error = quad(spalt, 0, 2 * np.pi * n / d, args=(d, y))
    f_modifiziert.append(result**2)

# Kurve auf 1 normieren:
f_modifiziert = f_modifiziert / np.max(f_modifiziert)

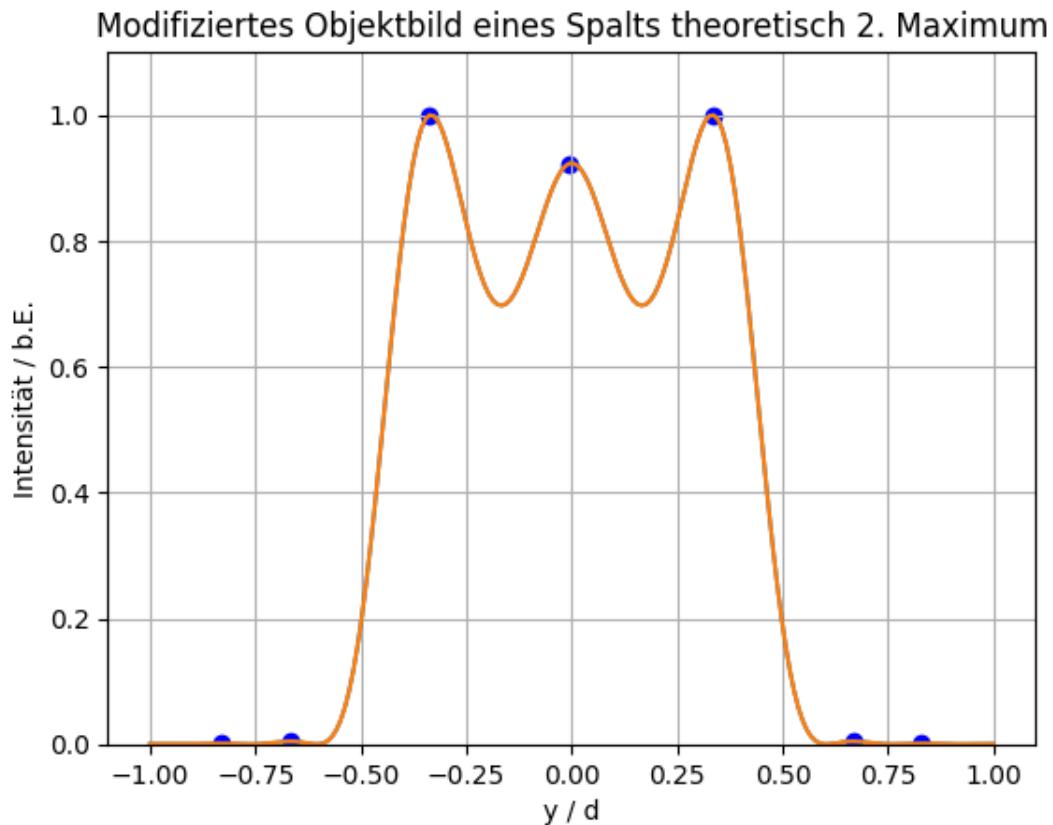
# Finden der Peaks (Extrema)
peaks, _ = find_peaks(f_modifiziert)

# Berechnung der Intensitäten der Peaks
intensities_of_peaks = f_modifiziert[peaks]
plt.plot(Y, f_modifiziert, label='Modifiziertes Objektbild')
plt.scatter(Y[peaks], intensities_of_peaks, color='blue', label='Maxima')

# Berechnung der Intensitäten der lokalen Minima
minima, _ = find_peaks(-np.array(f_modifiziert)) # Minima finden durch ↵
intensities_of_minima = f_modifiziert[minima]

# Plot der Funktion
plt.plot(Y, f_modifiziert)
plt.xlabel('y / d')
plt.ylabel('Intensität / b.E.')
plt.title('Modifiziertes Objektbild eines Spalts theoretisch 2. Maximum')
plt.ylim((0, 1.1))
plt.grid(True)
plt.show()

print(intensities_of_minima)
print(intensities_of_peaks)
```



```
[4.97274741e-06 5.75117999e-06 8.19119778e-05 6.97348189e-01
 6.97348189e-01 8.19119778e-05 5.75117999e-06 4.97274741e-06]
[0.00104393 0.00457448 1.          0.92316817 1.          0.00457448
 0.00104393]
```

[543]: #Experimentell bestimmten modifizierte Bilder

```
# Datei laden
pix, intensity = np.loadtxt("Aufgabe 3 Ote.txt", unpack=True, skiprows=4)

# Bestimmen des zu betrachtenden Intervalls (z.B. Pixel 100 bis Pixel 300)
start_pixel = 300
end_pixel = 800

# Einschränkung des Intervalls
intervall_index = (pix >= start_pixel) & (pix <= end_pixel)
pix_intervall = pix[intervall_index]
intensity_intervall = intensity[intervall_index]

# Intensität auf einen maximalen Schwellenwert begrenzen
```

```

intensity_limited = np.minimum(intensity_intervall, 1100)

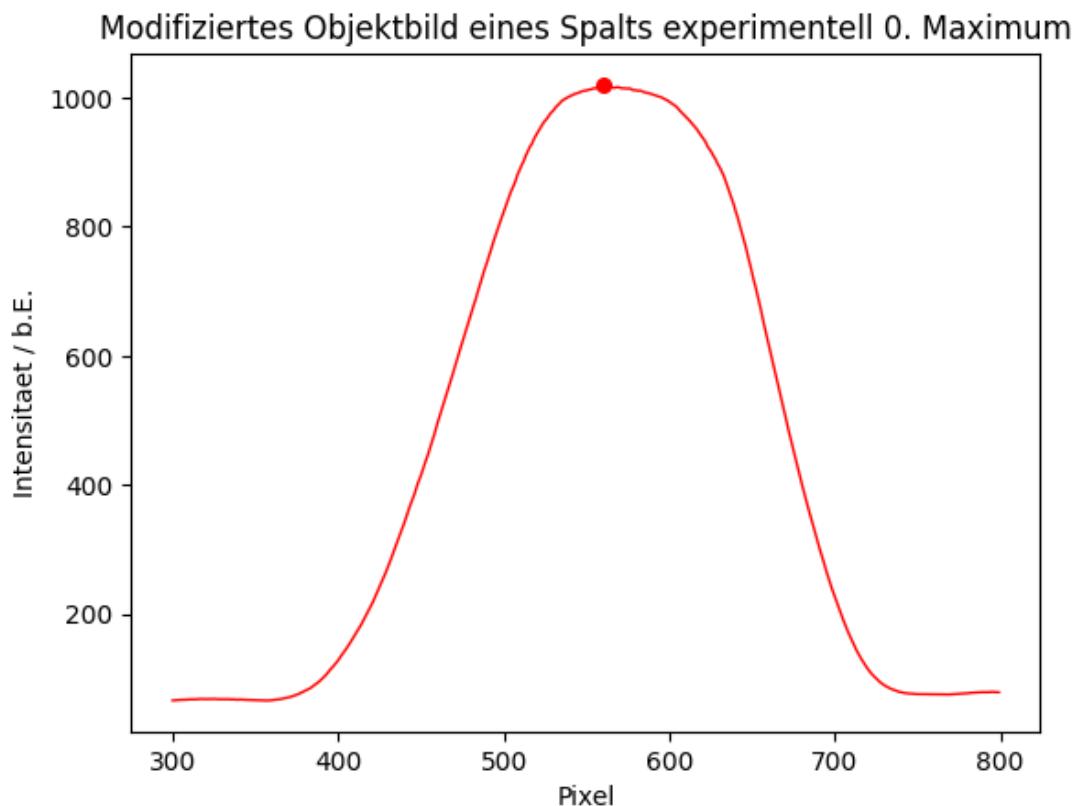
x_punkt = 560
y_punkt = 1020
plt.scatter(x_punkt, y_punkt, color='red', s=30)

# Daten plotten
plt.plot(pix_intervall, intensity_limited, color='red', linewidth=1)

# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensitaet / b.E.')
plt.title('Modifiziertes Objektbild eines Spalts experimentell 0. Maximum')

# Plot anzeigen
plt.show()

```



[544]: # Anpassung des Codes zum Laden der Datei, um mit fehlenden Werten umzugehen
file_path = r'C:\Users\shiy0\OneDrive\Dokumente\PAP2.1\Versuch 233\Python\u233\Aufgabe 3 0,1te.txt'

```

# Laden der Daten, wobei fehlende Werte als 'nan' behandelt werden
data = np.genfromtxt(file_path, delimiter='\t', skip_header=3,
                     filling_values=np.nan)

# Extrahieren der Pixel- und Intensitätswerte aus der ersten Profilspalte
pix = data[:, 0]
intensity = data[:, 1]

# Bestimmen des zu betrachtenden Intervalls (z.B. Pixel 100 bis Pixel 300)
start_pixel = 300
end_pixel = 800

# Einschränkung des Intervalls
intervall_index = (pix >= start_pixel) & (pix <= end_pixel)
pix_intervall = pix[intervall_index]
intensity_intervall = intensity[intervall_index]

# Intensität auf einen maximalen Schwellenwert begrenzen
intensity_limited = np.minimum(intensity_intervall, 1100)

# Entfernen von NaN-Werten für das Plotten
valid_indices = ~np.isnan(intensity)
pix = pix[valid_indices]
intensity = intensity[valid_indices]

x_punkt = 622
y_punkt = 883
plt.scatter(x_punkt, y_punkt, color='red', s=30)

x_punkt = 545
y_punkt = 590
plt.scatter(x_punkt, y_punkt, color='red', s=30)

x_punkt = 490
y_punkt = 725
plt.scatter(x_punkt, y_punkt, color='red', s=30)

# Daten plotten
plt.plot(pix_intervall, intensity_limited, color='red', linewidth=1)

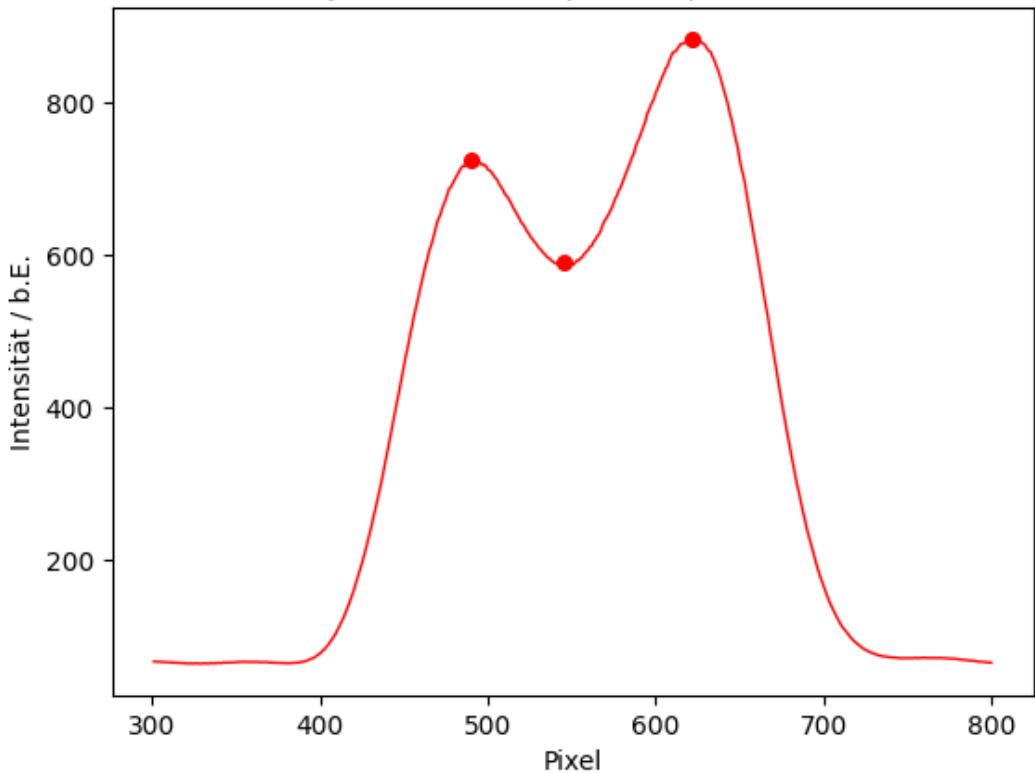
# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensität / b.E.')
plt.title('Modifiziertes Objektbild eines Spalts experimentell 1. Maximum')

# Plot anzeigen

```

```
plt.show()
```

Modifiziertes Objektbild eines Spalts experimentell 1. Maximum



```
[545]: from scipy.signal import argrelextrema, savgol_filter

# Vorhandenen Code zum Laden der Daten
pix, intensity = np.loadtxt("Aufgabe 3 0,1,2te.txt", unpack=True, skiprows=4)

# Bestimmen des zu betrachtenden Intervalls (z.B. Pixel 300 bis Pixel 800)
start_pixel = 300
end_pixel = 800

# Einschränkung des Intervalls
intervall_index = (pix >= start_pixel) & (pix <= end_pixel)
pix_intervall = pix[intervall_index]
intensity_intervall = intensity[intervall_index]

# Intensität auf einen maximalen Schwellenwert begrenzen
intensity_limited = np.minimum(intensity_intervall, 1100)

# Glättung der Daten mit Savitzky-Golay-Filter
```

```

window_size = 51 # muss ungerade sein
polynom_order = 3
smoothed_intensity = savgol_filter(intensity_limited, window_size, ↴
                                     polynom_order)

# Bestimmung der lokalen Maxima und Minima
maxima_indices = argrelextrema(smoothed_intensity, np.greater)[0]
minima_indices = argrelextrema(smoothed_intensity, np.less)[0]

# Daten plotten
plt.plot(pix_intervall, smoothed_intensity, color='red', linewidth=1)

# Maxima und Minima im Plot markieren
plt.plot(pix_intervall[maxima_indices], smoothed_intensity[maxima_indices], ↴
          "o", label="Maxima", color='green')
plt.plot(pix_intervall[minima_indices], smoothed_intensity[minima_indices], ↴
          "o", label="Minima", color='blue')

# Drucken der Maxima und Minima
print("Maxima:")
for index in maxima_indices:
    print(f"Pixel: {pix_intervall[index]}, Intensität: ↴
          {smoothed_intensity[index]}")

print("\nMinima:")
for index in minima_indices:
    print(f"Pixel: {pix_intervall[index]}, Intensität: ↴
          {smoothed_intensity[index]}")

# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensitaet / b.E.')
plt.title('Modifiziertes Objektbild eines Spalts experimentell 2. Maximum')
plt.legend()

# Plot anzeigen
plt.show()

```

Maxima:

Pixel: 314.97551, Intensität: 66.08293708483721
 Pixel: 360.97193, Intensität: 67.30339315350163
 Pixel: 477.96283, Intensität: 462.3282170927905
 Pixel: 558.95654, Intensität: 416.57730503793175
 Pixel: 638.95031, Intensität: 546.3256594466362

Minima:

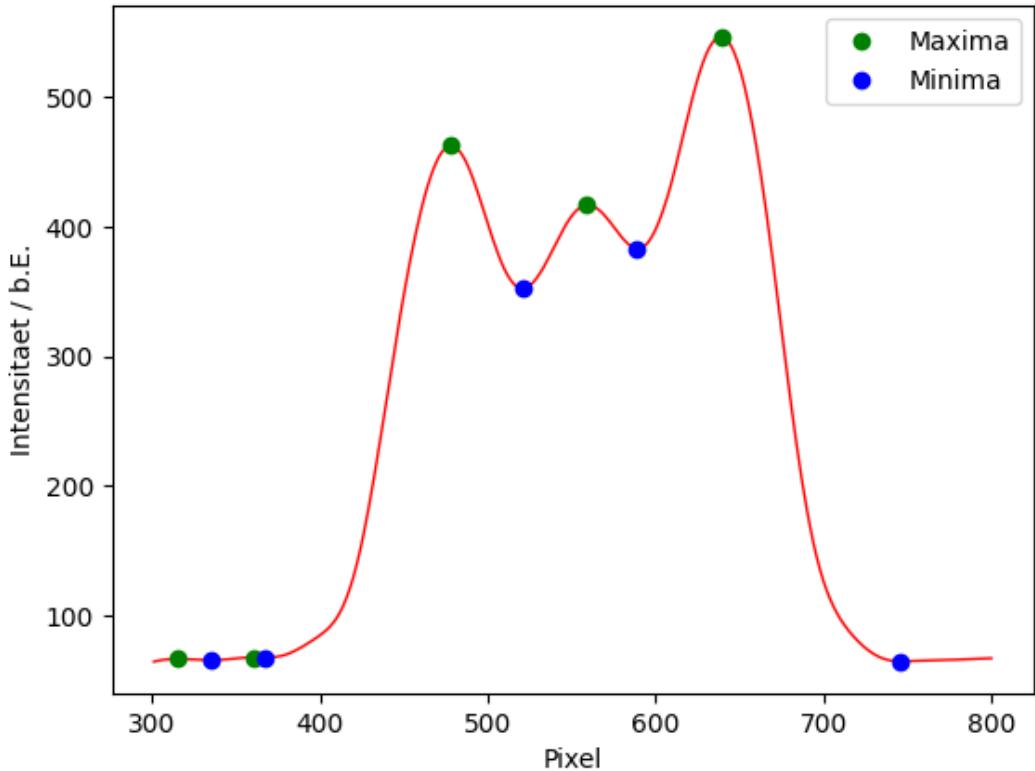
Pixel: 334.97395, Intensität: 65.08715123540613

```

Pixel: 366.97146, Intensität: 67.20511130843153
Pixel: 520.95949, Intensität: 352.2542346188973
Pixel: 588.9542, Intensität: 382.98616427053105
Pixel: 745.942, Intensität: 64.22539774517986

```

Modifiziertes Objektbild eines Spalts experimentell 2. Maximum



```

[546]: pix, intensity = np.loadtxt("Aufgabe 3 0,1,2,3te.txt", unpack=True, skiprows=4)

# Bestimmen des zu betrachtenden Intervalls (z.B. Pixel 300 bis Pixel 800)
start_pixel = 300
end_pixel = 800

# Einschränkung des Intervalls
intervall_index = (pix >= start_pixel) & (pix <= end_pixel)
pix_intervall = pix[intervall_index]
intensity_intervall = intensity[intervall_index]

# Intensität auf einen maximalen Schwellenwert begrenzen
intensity_limited = np.minimum(intensity_intervall, 1100)

# Glättung der Daten mit Savitzky-Golay-Filter

```

```

window_size = 51 # muss ungerade sein
polynom_order = 3
smoothed_intensity = savgol_filter(intensity_limited, window_size, □
    ↪polynom_order)

# Bestimmung der lokalen Maxima und Minima
maxima_indices = argrelextrema(smoothed_intensity, np.greater)[0]
minima_indices = argrelextrema(smoothed_intensity, np.less)[0]

# Daten plotten
plt.plot(pix_intervall, smoothed_intensity, color='red', linewidth=1)

# Maxima und Minima im Plot markieren
plt.plot(pix_intervall[maxima_indices], smoothed_intensity[maxima_indices], □
    ↪"o", label="Maxima", color='green')
plt.plot(pix_intervall[minima_indices], smoothed_intensity[minima_indices], □
    ↪"o", label="Minima", color='blue')

# Drucken der Maxima und Minima
print("Maxima:")
for index in maxima_indices:
    print(f"Pixel: {pix_intervall[index]}, Intensität: □
        ↪{smoothed_intensity[index]}")

print("\nMinima:")
for index in minima_indices:
    print(f"Pixel: {pix_intervall[index]}, Intensität: □
        ↪{smoothed_intensity[index]}")

# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensitaet / b.E.')
plt.title('Modifiziertes Objektbild eines Spalts experimentell 2. Maximum')
plt.legend()

# Plot anzeigen
plt.show()

```

Maxima:

Pixel: 479.96423, Intensität: 496.6511421438668
 Pixel: 537.9599, Intensität: 402.60131882941084
 Pixel: 591.95588, Intensität: 436.4234827135298
 Pixel: 652.95133, Intensität: 595.8720091739224
 Pixel: 773.94231, Intensität: 68.71539790799207

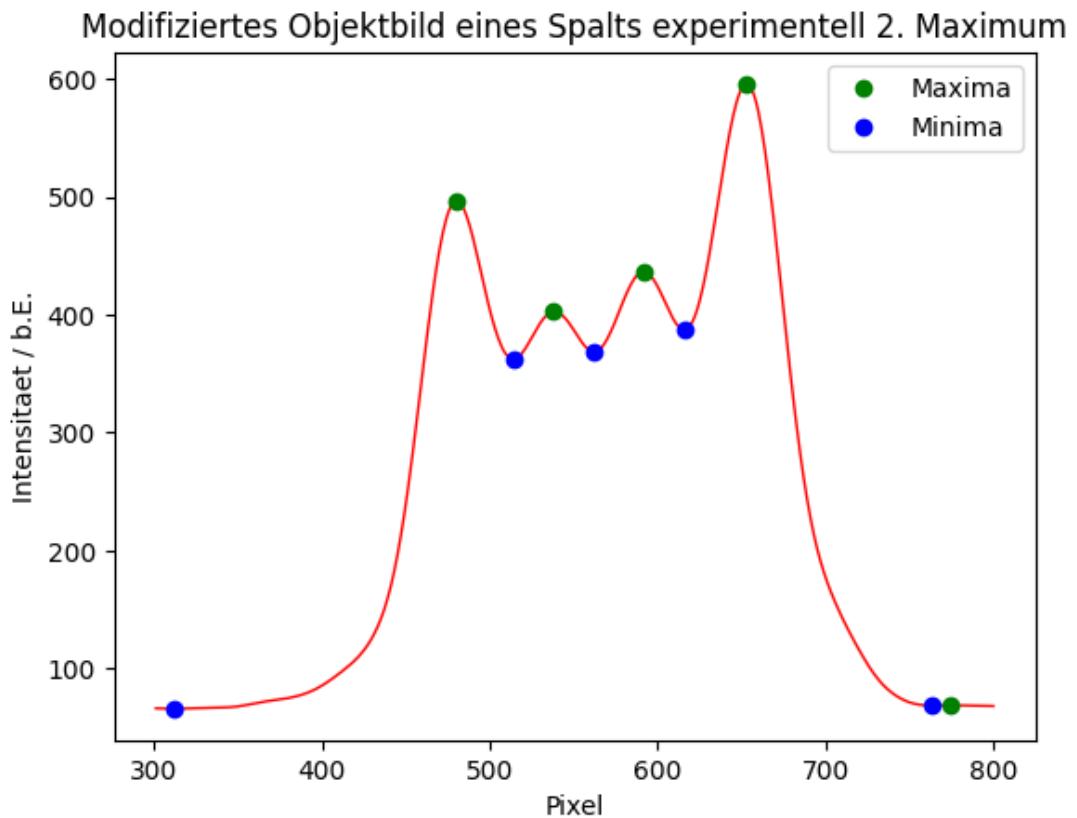
Minima:

Pixel: 311.97675, Intensität: 65.78585129399521

```

Pixel: 513.96169, Intensität: 362.4377322550838
Pixel: 561.95811, Intensität: 368.7144773061607
Pixel: 615.95409, Intensität: 387.34849744206383
Pixel: 762.94313, Intensität: 68.49389656766725

```



Python Code 8

```

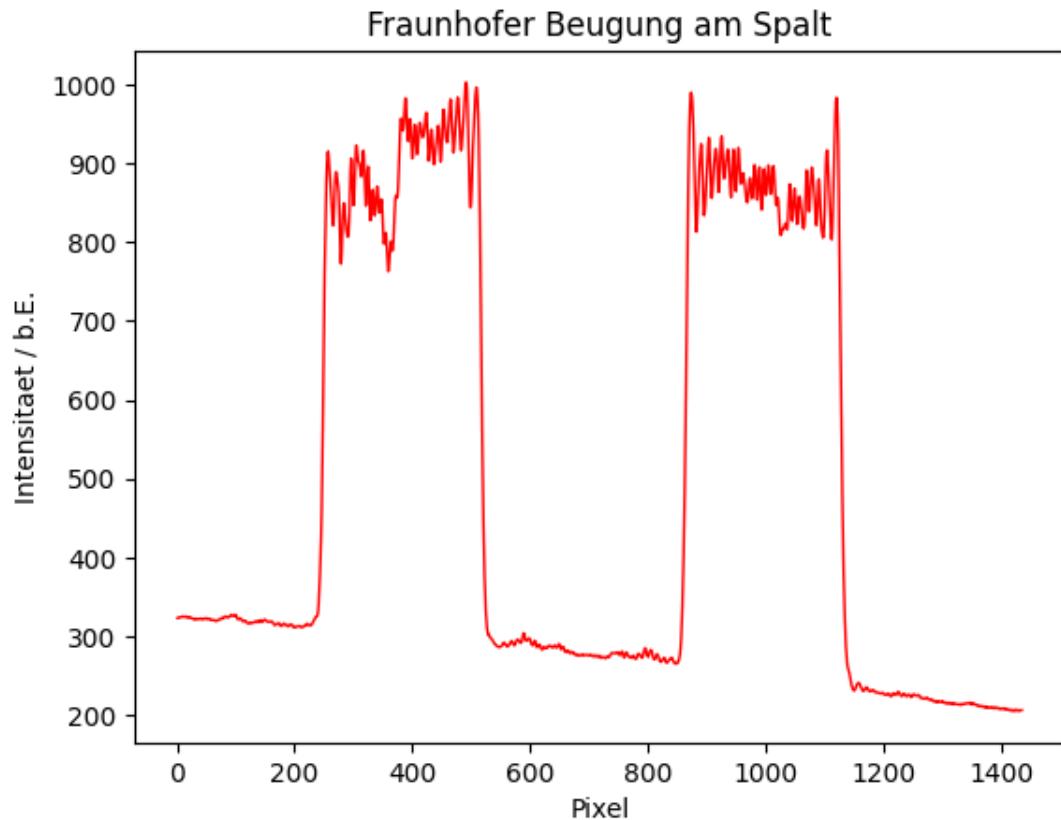
[547]: # Datei laden
pix, intensity = np.loadtxt("Aufgabe 4 doppelspalt.txt", unpack=True,
                           skiprows=4)

# Daten plotten
plt.plot(pix, intensity, color='red', linewidth=1)

# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensität / b.E.')
plt.title('Fraunhofer Beugung am Spalt')

```

```
# Plot anzeigen
plt.show()
```



Python Code 9

```
[548]: # Modifiziertes Objektbild (Doppelspalt)
def doppelspalt(k,d,g,y):
    return d/np.pi*np.cos(k*g/2)*np.sin(k*d/2)/(k*d/2)*np.cos(y*k)
```

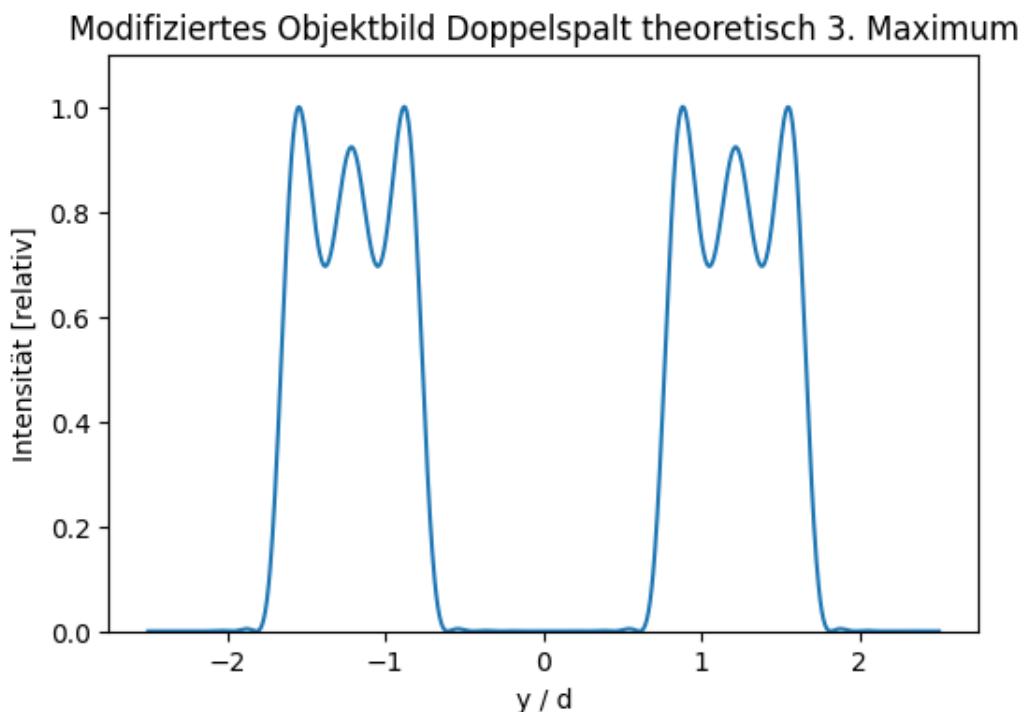
```
[549]: d=1
g=v*d
n=3
Y=np.linspace(-2.5,2.5,1000)*d
f_modified_double=[]
for y in Y:
    result, error = quad(doppelspalt, 0, 2*np.pi*n/d, args =(d,g,y,))
    f_modified_double.append(result**2)

f_modified_double = f_modified_double/np.max(f_modified_double)
plt.figure(figsize=(6,4))
```

```

plt.plot(Y,f_modified_double)
plt.xlabel("y / d")
plt.ylabel("Intensität [relativ]")
plt.title("Modifiziertes Objektbild Doppelspalt theoretisch 3. Maximum")
plt.ylim((0,1.1))
plt.savefig('DoppelModatheo.png')
plt.show()

```



```

[550]: d=1
g=v*d
n=2
Y=np.linspace(-2.5,2.5,1000)*d
f_modified_double=[]
for y in Y:
    result, error = quad(doppelspalt, 0, 2*np.pi*n/d, args =(d,g,y,))
    f_modified_double.append(result**2)

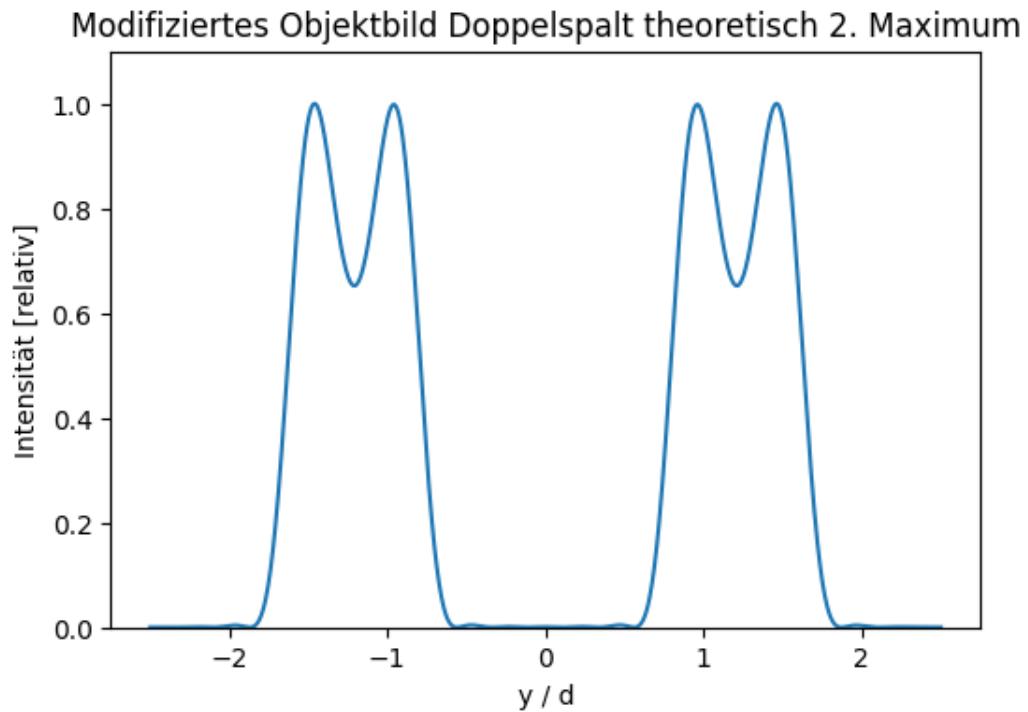
f_modified_double = f_modified_double/np.max(f_modified_double)
plt.figure(figsize=(6,4))
plt.plot(Y,f_modified_double)
plt.xlabel("y / d")
plt.ylabel("Intensität [relativ]")
plt.title("Modifiziertes Objektbild Doppelspalt theoretisch 2. Maximum")

```

```

plt.ylim((0,1.1))
plt.savefig('DoppelModatheo.png')
plt.show()

```



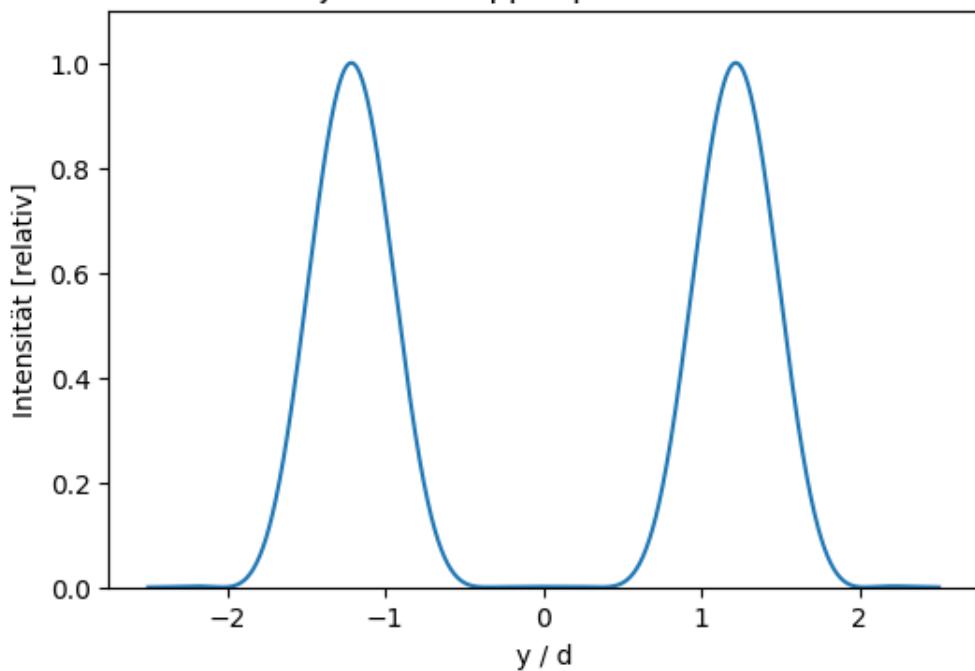
```

[551]: d=1
g=v*d
n=1
Y=np.linspace(-2.5,2.5,1000)*d
f_modified_double=[]
for y in Y:
    result, error = quad(doppelspalt, 0, 2*np.pi*n/d, args =(d,g,y,))
    f_modified_double.append(result**2)

f_modified_double = f_modified_double/np.max(f_modified_double)
plt.figure(figsize=(6,4))
plt.plot(Y,f_modified_double)
plt.xlabel("y / d")
plt.ylabel("Intensität [relativ]")
plt.title("Modifiziertes Objektbild Doppelspalt theoretisch 1. Maximum")
plt.ylim((0,1.1))
plt.savefig('DoppelModatheo.png')
plt.show()

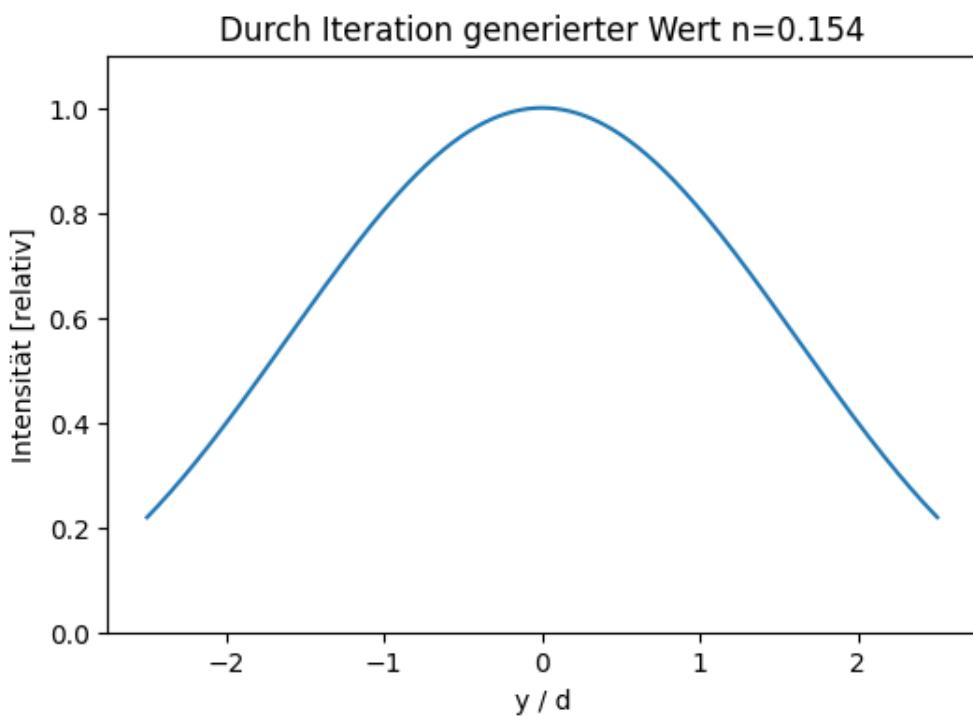
```

Modifiziertes Objektbild Doppelspalt theoretisch 1. Maximum



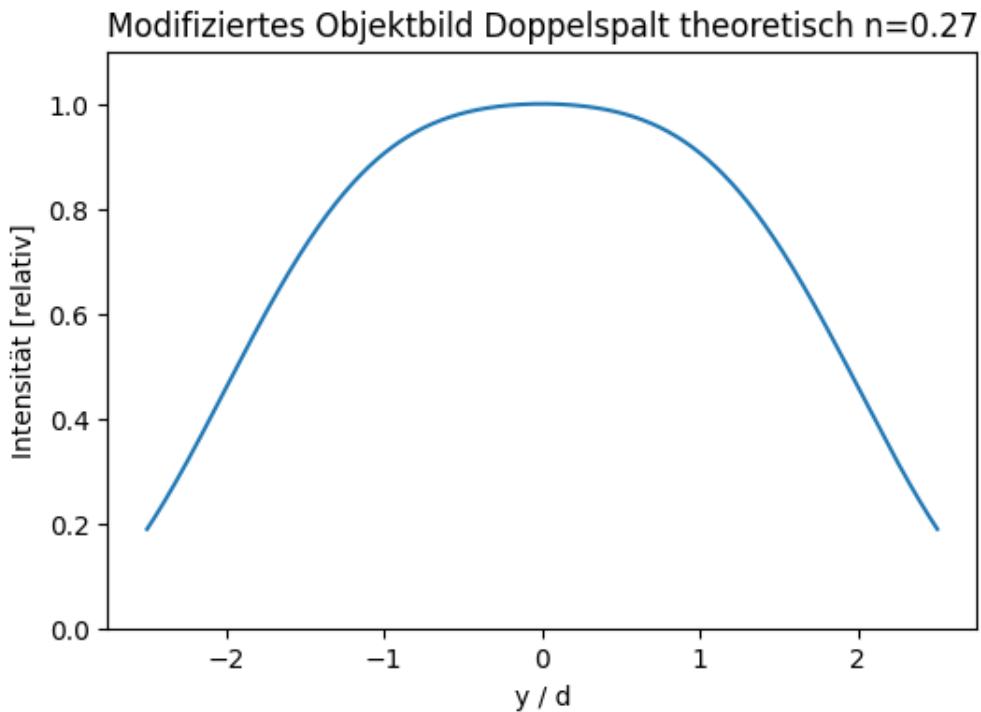
```
[564]: d=1
g=v*d
n=0.14
Y=np.linspace(-2.5,2.5,1000)*d
f_modified_double=[]
for y in Y:
    result, error = quad(doppelspalt, 0, 2*np.pi*n/d, args =(d,g,y,))
    f_modified_double.append(result**2)

f_modified_double = f_modified_double/np.max(f_modified_double)
plt.figure(figsize=(6,4))
plt.plot(Y,f_modified_double)
plt.xlabel("y / d")
plt.ylabel("Intensität [relativ]")
plt.title("Durch Iteration generierter Wert n=0.154")
plt.ylim((0,1.1))
plt.savefig('DoppelModatheo.png')
plt.show()
```



```
[560]: n=0.27
Y=np.linspace(-2.5,2.5,1000)*d
f_modified_double=[]
for y in Y:
    result, error = quad(doppelspalt, 0, 2*np.pi*n/d, args =(d,g,y,))
    f_modified_double.append(result**2)

f_modified_double = f_modified_double/np.max(f_modified_double)
plt.figure(figsize=(6,4))
plt.plot(Y,f_modified_double)
plt.xlabel("y / d")
plt.ylabel("Intensität [relativ]")
plt.title("Modifiziertes Objektbild Doppelspalt theoretisch n=0.27")
plt.ylim((0,1.1))
plt.savefig('DoppelModatheo.png')
plt.show()
```



```
[554]: pix, intensity = np.loadtxt("Aufgabe 4 doppelspalt o,1,2,3te.txt", unpack=True, skiprows=4)

# Zu betrachtendes Intervall festlegen
start_pixel = 100
end_pixel = 1400

# Intervall einschränken
intervall_index = (pix >= start_pixel) & (pix <= end_pixel)
pix_intervall = pix[intervall_index]
intensity_intervall = intensity[intervall_index]

# Intensität begrenzen
intensity_limited = np.minimum(intensity_intervall, 1100)

# Daten glätten
window_size = 51 # muss ungerade sein
polynom_order = 3
smoothed_intensity = savgol_filter(intensity_limited, window_size, polynom_order)

# Lokale Maxima und Minima bestimmen
maxima_indices = argrelextrema(smoothed_intensity, np.greater)[0][3:-5]
```

```

minima_indices = argrelextrema(smoothed_intensity, np.less)[0][3:-6]

# Zentralen Bereich ausschließen
zentral_start = 560 # Anfang des auszuschließenden zentralen Bereichs
zentral_end = 900 # Ende des auszuschließenden zentralen Bereichs

# Filter für Maxima und Minima
maxima_indices = [index for index in maxima_indices if not (zentral_start <= pix_intervall[index] <= zentral_end)]
minima_indices = [index for index in minima_indices if not (zentral_start <= pix_intervall[index] <= zentral_end)]

# Daten plotten
plt.plot(pix_intervall, smoothed_intensity, color='red', linewidth=1)
plt.plot(pix_intervall[maxima_indices], smoothed_intensity[maxima_indices], "o", label="Maxima", color='green')
plt.plot(pix_intervall[minima_indices], smoothed_intensity[minima_indices], "o", label="Minima", color='blue')

# Maxima und Minima ausgeben
print("Maxima:")
for index in maxima_indices:
    print(f"Pixel: {pix_intervall[index]}, Intensität:{smoothed_intensity[index]}")

print("\nMinima:")
for index in minima_indices:
    print(f"Pixel: {pix_intervall[index]}, Intensität:{smoothed_intensity[index]}")

# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensität / b.E.')
plt.title('Modifiziertes Objektbild eines Doppelspalts experimentell 3. Maximum')
plt.legend()

# Plot anzeigen
plt.show()

```

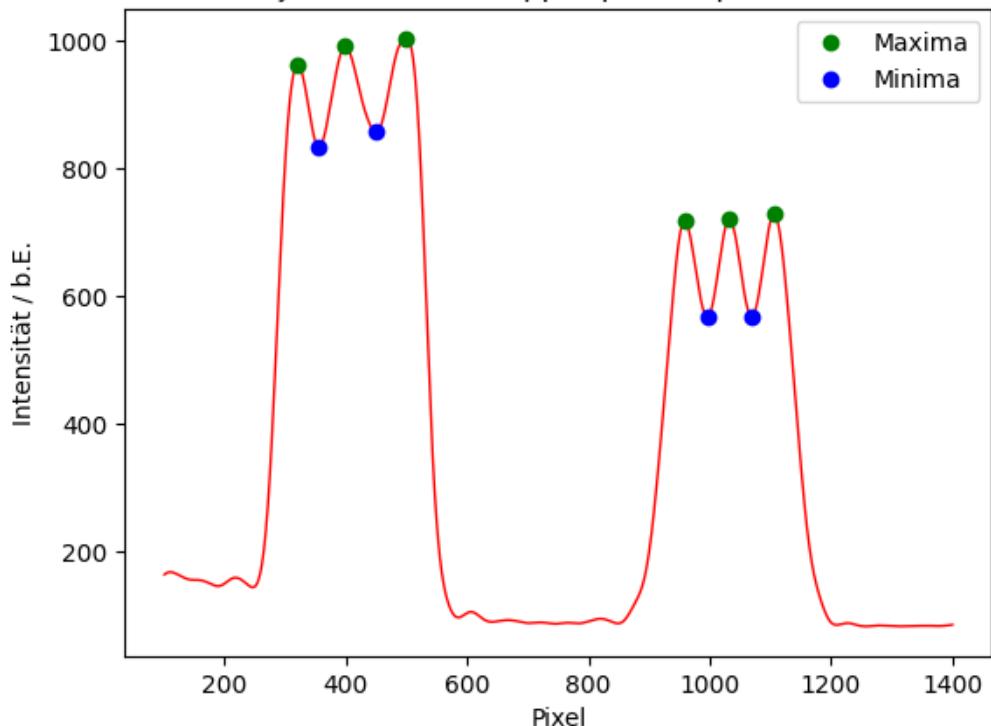
Maxima:

Pixel: 320.95044, Intensität: 963.2741412487073
Pixel: 398.93839, Intensität: 991.4428440018837
Pixel: 499.9228, Intensität: 1003.7706598466364
Pixel: 958.85192, Intensität: 718.5427115678586
Pixel: 1031.8407, Intensität: 720.6212805481299
Pixel: 1105.8292, Intensität: 729.3310217146342

Minima:

Pixel: 355.94503, Intensität: 833.5082789220432
Pixel: 449.93052, Intensität: 857.2563816222173
Pixel: 995.84621, Intensität: 568.1953226188479
Pixel: 1068.8349, Intensität: 569.2178268714923

Modifiziertes Objektbild eines Doppelspalts experimentell 3. Maximum



```
[555]: # Anpassung des Codes zum Laden der Datei, um mit fehlenden Werten umzugehen
file_path = r'C:\Users\shiy0\OneDrive\Dokumente\PAP2.1\Versuch 233\Python_233\Aufgabe 4 doppelspalt_o,1,2te.txt'

# Laden der Daten, wobei fehlende Werte als 'nan' behandelt werden
data = np.genfromtxt(file_path, delimiter='\t', skip_header=3,
                     filling_values=np.nan)

# Extrahieren der Pixel- und Intensitätswerte aus der ersten Profilspalte
pix = data[:, 0]
intensity = data[:, 1]

# Zu betrachtendes Intervall festlegen
start_pixel = 100
```

```

end_pixel = 1400

# Intervall einschränken
intervall_index = (pix >= start_pixel) & (pix <= end_pixel)
pix_intervall = pix[intervall_index]
intensity_intervall = intensity[intervall_index]

# Intensität begrenzen
intensity_limited = np.minimum(intensity_intervall, 1100)

# Daten glätten
window_size = 51 # muss ungerade sein
polynom_order = 3
smoothed_intensity = savgol_filter(intensity_limited, window_size,
                                   polynom_order)

# Lokale Maxima und Minima bestimmen
maxima_indices = argrelextrema(smoothed_intensity, np.greater)[0][1:-9]
minima_indices = argrelextrema(smoothed_intensity, np.less)[0][2:-9]

# Zentralen Bereich ausschließen
zentral_start = 500 # Anfang des auszuschließenden zentralen Bereichs
zentral_end = 800 # Ende des auszuschließenden zentralen Bereichs

# Filter für Maxima und Minima
maxima_indices = [index for index in maxima_indices if not (zentral_start <=
                                                               pix_intervall[index] <= zentral_end)]
minima_indices = [index for index in minima_indices if not (zentral_start <=
                                                               pix_intervall[index] <= zentral_end)]

# Daten plotten
plt.plot(pix_intervall, smoothed_intensity, color='red', linewidth=1)
plt.plot(pix_intervall[maxima_indices], smoothed_intensity[maxima_indices], "o",
          label="Maxima", color='green')
plt.plot(pix_intervall[minima_indices], smoothed_intensity[minima_indices], "o",
          label="Minima", color='blue')

# Maxima und Minima ausgeben
print("Maxima:")
for index in maxima_indices:
    print(f"Pixel: {pix_intervall[index]}, Intensität:{smoothed_intensity[index]}")

print("\nMinima:")
for index in minima_indices:

```

```

    print(f"Pixel: {pix_intervall[index]}, Intensität:{smoothed_intensity[index]}")

# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensität / b.E.')
plt.title('Modifiziertes Objektbild eines Doppelspalts experimentell 2. Maximum')
plt.legend()

# Plot anzeigen
plt.show()

```

Maxima:

Pixel: 333.97886, Intensität: 819.2102858603663

Pixel: 471.97013, Intensität: 822.037068878328

Pixel: 953.93963, Intensität: 571.486500072924

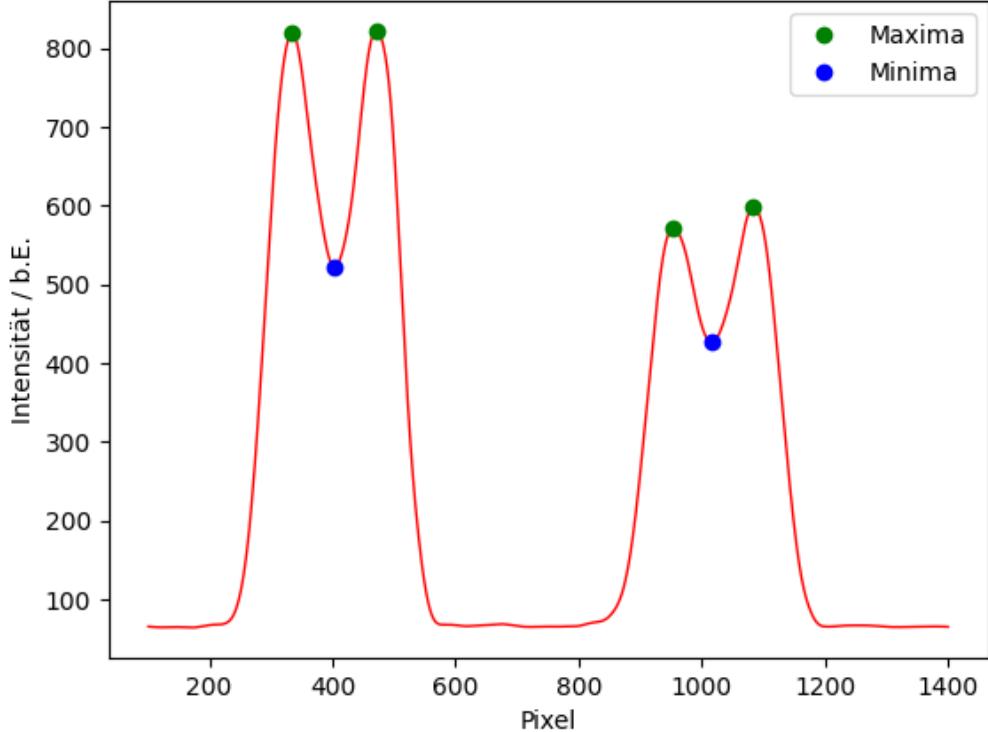
Pixel: 1083.9314, Intensität: 599.0759166352464

Minima:

Pixel: 403.97443, Intensität: 522.2843255958131

Pixel: 1015.9357, Intensität: 426.58647356405896

Modifiziertes Objektbild eines Doppelspalts experimentell 2. Maximum



```
[556]: pix, intensity = np.loadtxt("Aufgabe 4 doppelspalt 0,1te.txt", unpack=True,
                                skiprows=4)

# Zu betrachtendes Intervall festlegen
start_pixel = 100
end_pixel = 1400

# Intervall einschränken
intervall_index = (pix >= start_pixel) & (pix <= end_pixel)
pix_intervall = pix[intervall_index]
intensity_intervall = intensity[intervall_index]

# Intensität begrenzen
intensity_limited = np.minimum(intensity_intervall, 1100)

# Daten glätten
window_size = 51 # muss ungerade sein
polynom_order = 3
smoothed_intensity = savgol_filter(intensity_limited, window_size,
                                    polynom_order)

# Lokale Maxima und Minima bestimmen
maxima_indices = argrelextrema(smoothed_intensity, np.greater)[0][1:-3]
minima_indices = argrelextrema(smoothed_intensity, np.less)[0][1:-4]

# Zentralen Bereich ausschließen
zentral_start = 700 # Anfang des auszuschließenden zentralen Bereichs
zentral_end = 800 # Ende des auszuschließenden zentralen Bereichs

# Filter für Maxima und Minima
maxima_indices = [index for index in maxima_indices if not (zentral_start <=
    pix_intervall[index] <= zentral_end)]
minima_indices = [index for index in minima_indices if not (zentral_start <=
    pix_intervall[index] <= zentral_end)]

# Daten plotten
plt.plot(pix_intervall, smoothed_intensity, color='red', linewidth=1)
plt.plot(pix_intervall[maxima_indices], smoothed_intensity[maxima_indices], "o",
         label="Maxima", color='green')
plt.plot(pix_intervall[minima_indices], smoothed_intensity[minima_indices], "o",
         label="Minima", color='blue')

# Maxima und Minima ausgeben
print("Maxima:")
for index in maxima_indices:
```

```

    print(f"Pixel: {pix_intervall[index]}, Intensität: {smoothed_intensity[index]}")

print("\nMinima:")
for index in minima_indices:
    print(f"Pixel: {pix_intervall[index]}, Intensität: {smoothed_intensity[index]}")

# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensität / b.E.')
plt.title('Modifiziertes Objektbild eines Doppelspalts experimentell 1. Maximum')
plt.legend()

# Plot anzeigen
plt.show()

```

Maxima:

Pixel: 437.89892, Intensität: 930.1215960599156

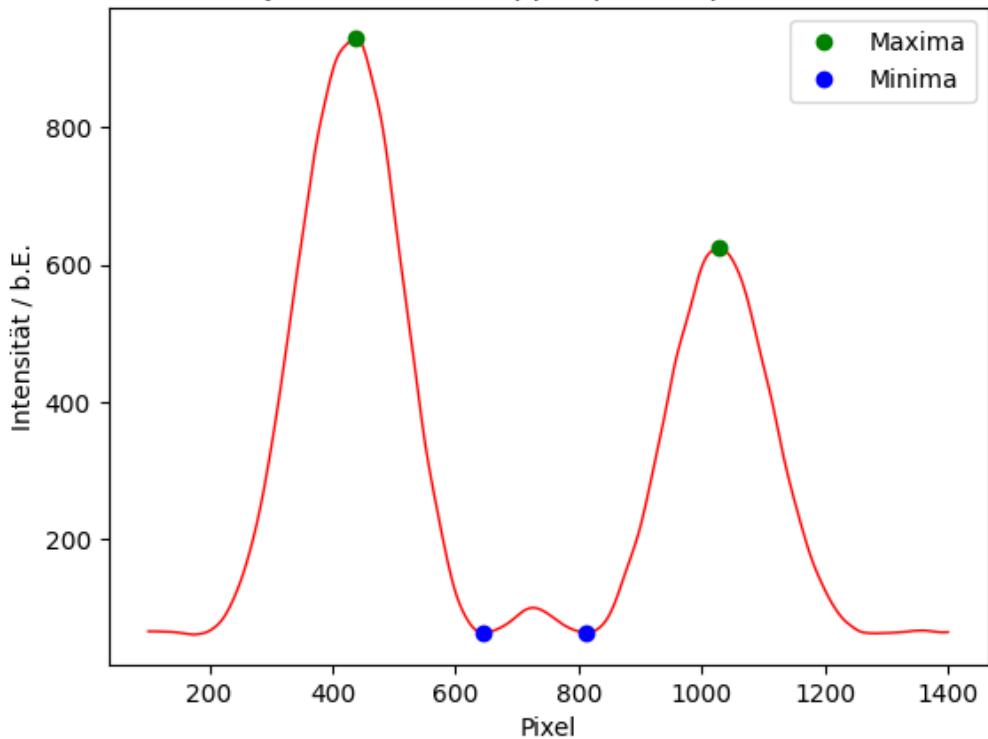
Pixel: 1027.7628, Intensität: 624.0557616224484

Minima:

Pixel: 645.85092, Intensität: 62.68226599044027

Pixel: 812.81239, Intensität: 63.54603455983026

Modifiziertes Objektbild eines Doppelspalts experimentell 1. Maximum



Python Code 10

```
[565]: pix, intensity = np.loadtxt("Aufgabe 4 Ote.txt", unpack=True, skiprows=4)

# Zu betrachtendes Intervall festlegen
start_pixel = 100
end_pixel = 1400

# Intervall einschränken
intervall_index = (pix >= start_pixel) & (pix <= end_pixel)
pix_intervall = pix[intervall_index]
intensity_intervall = intensity[intervall_index]

# Intensität begrenzen
intensity_limited = np.minimum(intensity_intervall, 1100)

# Daten glätten
window_size = 51 # muss ungerade sein
polynom_order = 3
smoothed_intensity = savgol_filter(intensity_limited, window_size, polynom_order)
```

```

# Lokale Maxima und Minima bestimmen
maxima_indices = argrelextrema(smoothed_intensity, np.greater)[0][1:-3]
minima_indices = argrelextrema(smoothed_intensity, np.less)[0][1:-4]

# Zentralen Bereich ausschließen
zentral_start = 200 # Anfang des auszuschließenden zentralen Bereichs
zentral_end = 1400 # Ende des auszuschließenden zentralen Bereichs

# Filter für Maxima und Minima
maxima_indices = [index for index in maxima_indices if not (zentral_start <= pix_intervall[index] <= zentral_end)]
minima_indices = [index for index in minima_indices if not (zentral_start <= pix_intervall[index] <= zentral_end)]

# Daten plotten
plt.plot(pix_intervall, smoothed_intensity, color='red', linewidth=1)

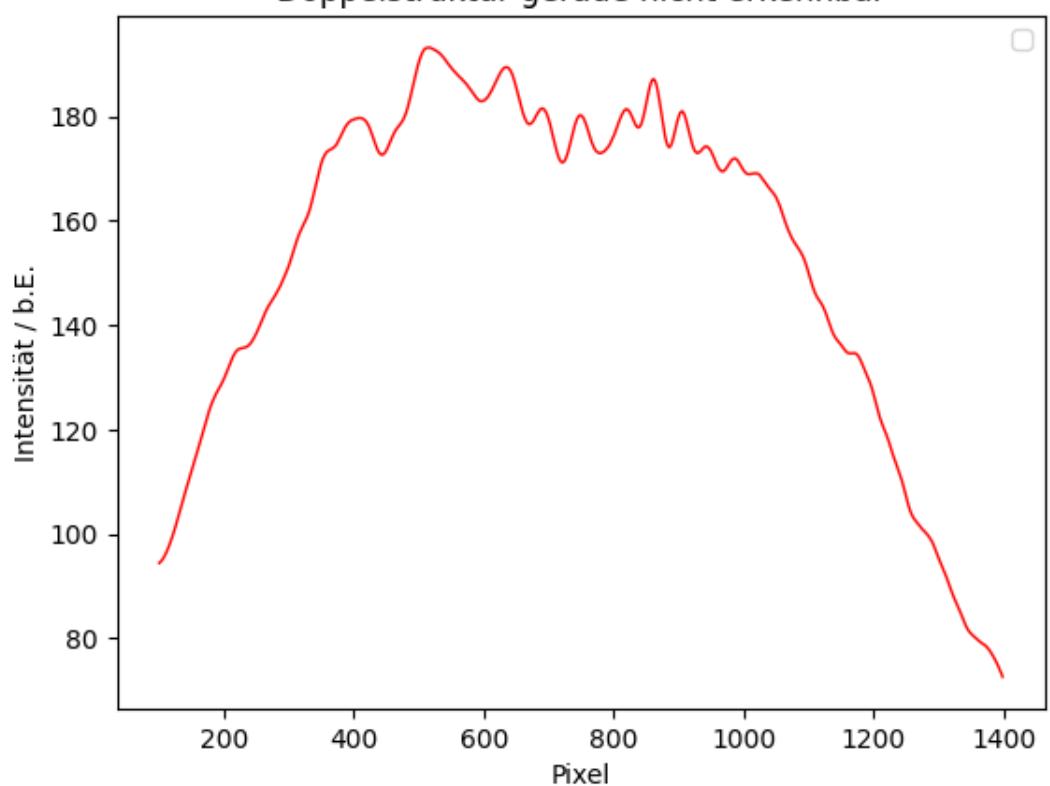
# Plot beschriften
plt.xlabel('Pixel')
plt.ylabel('Intensität / b.E.')
plt.title('Doppelstruktur gerade nicht erkennbar')
plt.legend()

# Plot anzeigen
plt.show()

```

No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.

Doppelstruktur gerade nicht erkennbar



[]:

6 Quelle

- Wagner, J. (April 2022). Physikalisches Praktikum PAP 2.1 für Studierende der Physik [Praktikumsanleitung]. Ruprecht-Karls-Universität Heidelberg. Abgerufen am 29. Oktober 2023, von https://www.physi.uni-heidelberg.de/Einrichtungen/AP/info/Corona /PAP2_1_2023.pdf