

Fourieroptik

Physikalisches Anfängerpraktikum II

Juan Provencio Lameiras

Betreuer/in: Jana Felz

Inhaltsverzeichnis

1 Ziel des Versuches	2
2 Grundlagen	2
2.1 "Klassische" Theorie der Beugung	2
2.2 Beugungsbild des Spaltes als Fouriertransformation der Spaltöffnung . . .	3
3 Versuchsaufbau	6
3.1 Materialen und Geräte	6
3.2 Aufbau	7
4 Messung und Auswertung	8
4.1 Messprotokoll	8
4.2 Auswertung	12
5 Zusammenfassung und Diskussion	29
6 Quellen	34
7 Anhang	36

1 Ziel des Versuches

In diesem Versuch beobachten wir die Beugung eines Lasers am Einzel- und Doppelspalt. Wir wollen die Beobachtungen untersuchen und sie mit der Theorie vergleichen, insbesondere die Tatsache, dass das Spaltbild und die Beugung am Spalt durch Fouriertransformationen verbunden sind.

2 Grundlagen

2.1 "Klassische" Theorie der Beugung

In der "klassischen" Theorie der Beugung nach Fresnel ist eine Beschreibung der Beugung kompliziert sehr kompliziert. Nach Fraunhofer wird angenommen, dass gebeugte Lichtstrahlen aus dem Unendlichen kommen und parallel zueinander sind. Dies erlaubt wesentliche Vereinfachungen im Gegensatz zur Fresnelschen Beugung und ist auch durch Systeme von Linsen gut realisierbar. Dabei werden Lichtstrahlen als Ebene Wellen dargestellt. Bei kohärentem Licht der Wellenlänge λ , Amplitude E_0 , Phase $\phi = \omega t$, so beobachtet man am Winkel α die folgende Amplitude:

$$E_\infty(\alpha) = \int_{-\frac{d}{2}}^{\frac{d}{2}} E_0 e^{i(\omega t - kl)} dy \quad (1)$$

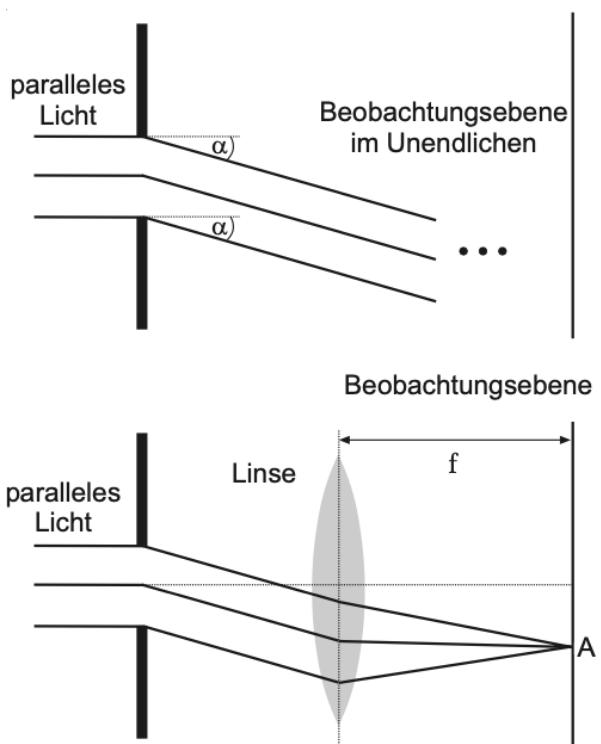


Abbildung 1: Beugung nach Fraunhofer

Dabei ist d der Spaltabstand und l wird definiert durch

$$l = R + y \sin \alpha \quad (2)$$

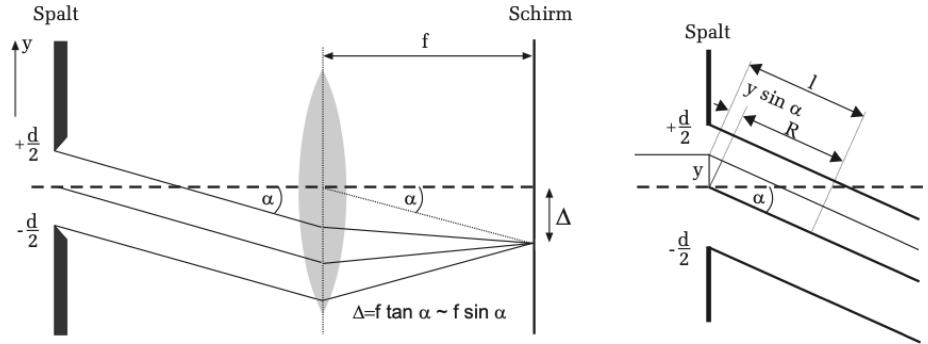


Abbildung 2: Fraunhofer Beugung am Spalt

Eine Ausführung des Integrals liefert

$$E_\infty(\alpha) = E_0 e^{i(\omega t - kl)} \frac{\sin\left(\pi d \frac{\sin \alpha}{\lambda}\right)}{\pi \frac{\sin \alpha}{\lambda}} \quad |x = \pi d \frac{\sin \alpha}{\lambda} \quad (3)$$

$$= E_0 e^{i(\omega t - kl)} \frac{\sin x}{x} d \quad (4)$$

Die sinc-Funktion wird in der Durchführung dieses Versuches eine zentrale Rolle übernehmen.

Die Intensität bestimmt man durch Quadrieren von (4):

$$I_\infty \propto \frac{\sin^2 x}{x^2} \quad (5)$$

2.2 Beugungsbild des Spaltes als Fouriertransformation der Spaltöffnung

2.2.1 Einzelspalt

Indem man den Spalt durch eine rechteckige Spaltfunktion darstellt, kann man mithilfe der Fourierdarstellung auch das Beugungsbild aufstellen. Im Fall des Einzelspalts hat man eine einzige Rechtecksfunktion $f(y)$ in Abhängigkeit der Position y :

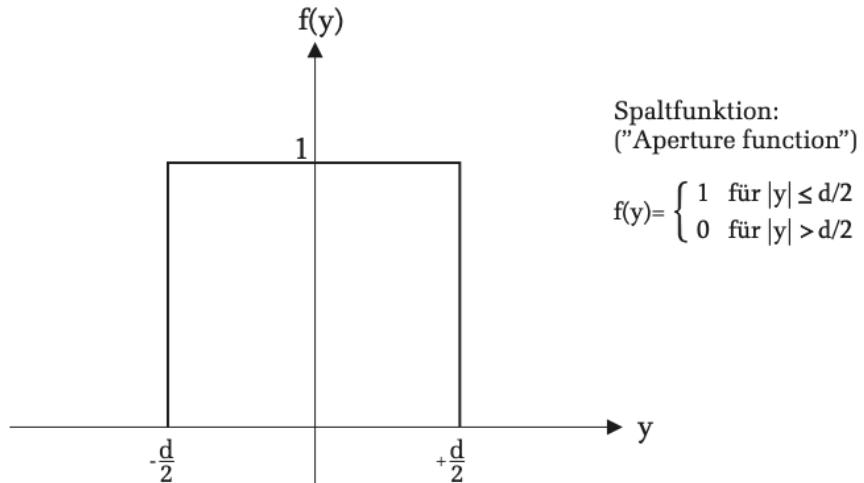


Abbildung 3: Spaltfunktion des Einfachspalts

Daraus wollen wir die Fouriertransformierte $F(k_y)$ bestimmen, woraus sich ergibt:

$$F(k_y) = \text{sinc}\left(\frac{k_y d}{2}\right) d \quad (6)$$

mit den Nullstellen

$$k_{y,n} = \frac{2\pi n}{d} \quad (7)$$

Will man die obige Funktion zurück in die Ortsabhängigkeit übersetzen, so stellt sich heraus dass das Integral gar nicht analytisch lösbar ist, aber durch numerische Integration wieder die Spaltfunktion rauskommt. Für diese Integration nutzt man die Symmetrie $F(k_y) = F(-k_y)$ und erhält

$$f(y) = \frac{d}{\pi} \int_0^{\infty} \text{sinc}\left(\frac{k_y d}{2}\right) \cos(k_y y) dk_y \quad (8)$$

Wir untersuchen wie sich das Integral verhält, wenn wir nicht den gesamten Integrationsbereich betrachten, sondern nur gezielt wenigere Teilstrahlen durch Ausblenden in der Fourierebene durchlässt, so ersetzen wir die obere Grenze mit der n -ten Nullstelle von F :

$$f_{\text{modifiziert}}(y) = \frac{d}{\pi} \int_0^{k_{y,n}} \text{sinc}\left(\frac{k_y d}{2}\right) \cos(k_y y) dk_y \quad (9)$$

Dies können wir numerisch integrieren und durch Quadrieren erhalten wir wieder ein ähnliches Bild wie die Intensitätsstruktur des Beugungsbildes.

2.2.2 Doppelspalt

Ähnlich wie wir mit dem Einfachspalt gemacht haben, wollen wir jetzt die Spaltfunktion des Doppelspaltes transformieren. Diese wird ähnlich zum Einfachspalt beschrieben, nur haben wir hier eine weitere wichtige Größe, und zwar den Spaltabstand g .

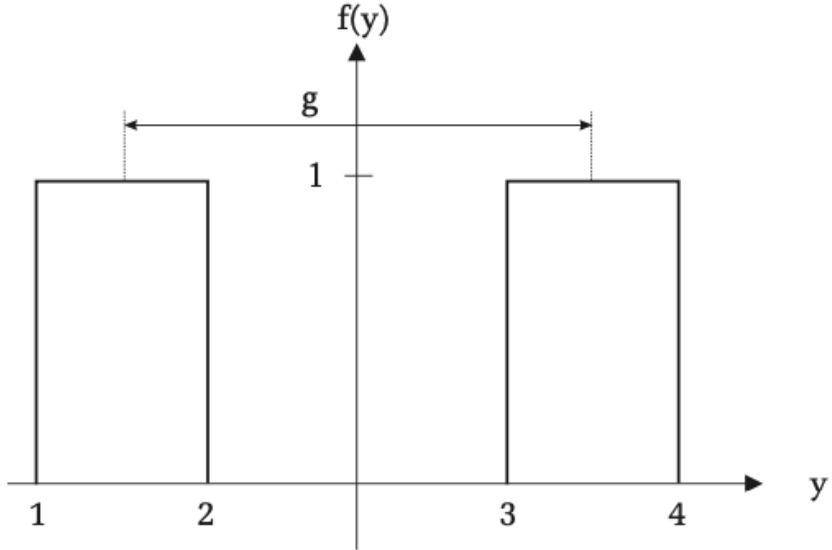


Abbildung 4: Spaltfunktion des Doppelspalts

Diese Funktion können wir als zwei verschobene Einzelspalte und analog rechnen. Wir erhalten

$$F(k_y) = 2 \cos\left(\frac{k_y g}{2}\right) d \operatorname{sinc}\left(\frac{k_y d}{2}\right) \quad (10)$$

Quadrieren dieses Ausdrucks und eine geeignete Substitution

$$k_y = \frac{2\pi \sin \alpha}{\lambda} \quad (11)$$

ergibt die uns vertraute Intensitätsstruktur des Doppelspaltes

$$I = 4d^2 \cos^2\left(\frac{\pi g \sin \alpha}{\lambda}\right) \operatorname{sinc}^2\left(\frac{\pi d \sin \alpha}{\lambda}\right) \quad (12)$$

Aus diesem Ausdruck wollen wir wieder das modifizierte Bild bestimmen

$$F_{\text{modifiziert}}(y) \propto [f_{\text{modifiziert}}(y)]^2 = \left[\frac{2d}{\pi} \int_0^{k_y n} \cos\left(\frac{k_y g}{2}\right) \operatorname{sinc}\left(\frac{k_y d}{2}\right) \cos(k_y y) dk_y \right]^2 \quad (13)$$

Die zugehörigen Beugungsbilder sehen folgendermaßen aus

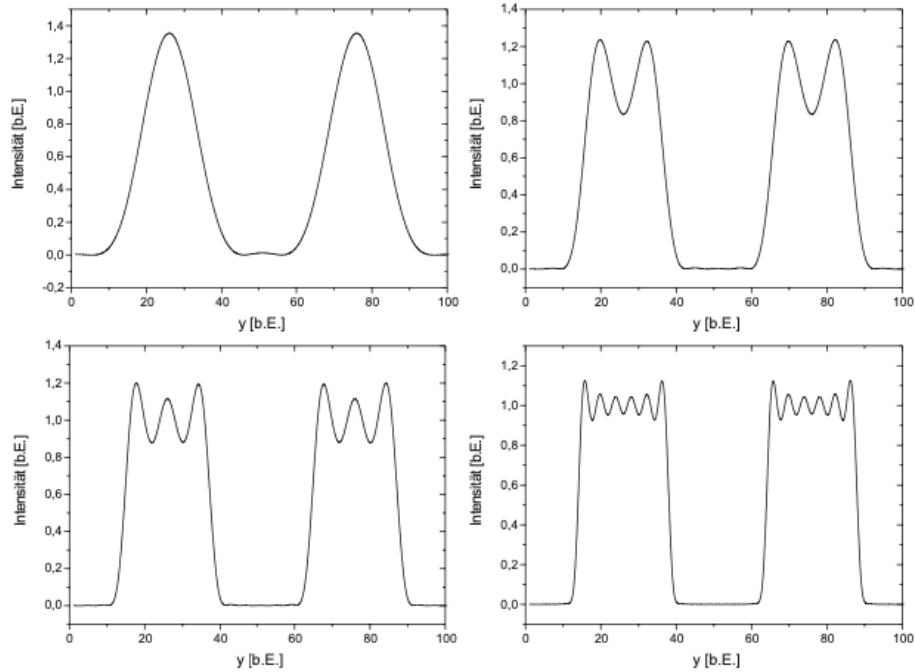


Abbildung 5: Spaltbild des Doppelspalts beim Ausblenden verschiedener Beugungsordnungen

Bei einer gewissen Spaltweite verschwindet diese Doppelspaltstruktur, dafür muss man das Maximum erster Ordnung rauschneiden und es entsteht statt der zwei Gaußförmigen Beulen ein Plateau in der Mitte.

3 Versuchsaufbau

3.1 Materialen und Geräte

- Singlemode Glasfaser-gekoppelter Diodenlaser mit regelbarer Stromversorgung und Justieroptik für einen Parallelstrahl
- Ein Satz Beugungsobjekte: Spalte, Doppelspalt, Gitter, Kreuzgitter, Spalt mit gaußförmigem Transmissionsprofil
- Verschiedene sphärische und Zylinder-Linsen
- Strahlteiler
- Graufilter
- Spiegel
- Verschiedene Fest-Reiter

- Feinjustierbare Verschiebereiter
- Symmetrisch öffnender Analysierspalt
- Verschiedene schmale Metallstreifen als Modenblende
- SCDD-Zeilenkamera mit PC und Monitor

3.2 Aufbau

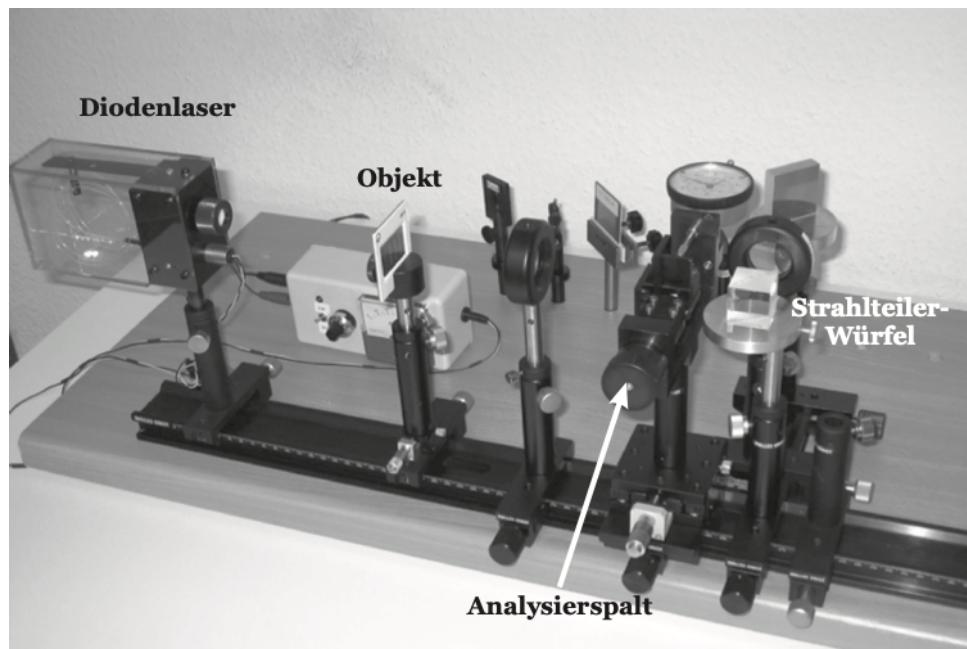


Abbildung 6: Aufbau nach Praktikumsskript

4 Messung und Auswertung

4.1 Messprotokoll

Messprotokoll V233 Fourieroptik			
1. S. 2022			
Mike Brandt Juan Provencio			
Teilaufgabe 1: Quantitative Beobachtungen am Einfachspalt			
Wir protokollieren Lage und Intensität der ersten 5 Max./Min.			
Tabelle 1: Lage/Intensität der Extrema			
Messung	Lage [Px]	Intensität 1	Intensität 2
Max. 0	897	0,9672	0,157
Min. 1	997	0,019	
Max. 1	1032	0,0867	0,8746
Min. 2	1096		0,0683
Max. 2	1142		0,299
Min. 3	1198		0,028
Max. 3	1260		0,1496
Min. 4	1295		0,0199
Max. 4	1242		0,100
Min. 5	1394		0,018
Max. 5	1436		0,0804
Untergrund			0,006
Wir eichen die Abszisse			
Tabelle 2: Eichung			
Messung	Lage [Px]	Lage [mm]	
1	384	2,05	
2	476	1,81	
3	576	1,60	
4	678	1,45	
5	786	1,20	

Abbildung 7: Messprotokoll

Teilaufgabe 2: Doppelspalt			
Tabelle 3: Lage / Intensität der Extrema			
Messung	Lage [Px]	Intensität (1)	Intensität (2)
Max. 0	895	0.9445	0.986
Min. 1	917	0.020	
Max. 1	937	0.644	0.954
Min. 2	962	0.012	
Max. 2	975	0.0697	0.1248
Min. 3	1003		0.0176
Max. 3	1027		0.062
Min. 4	1046		0.0160
Max. 4	1065		0.066
Min. 5	1103		0.008
Max. 5	1153		0.038
Untergrund			0.006

Teilaufgabe 3: Fouriersynthese			
Tabelle 4:			
Messung [n]	Art	Intensität	Absolute Lage [Px.]
0	Max.	0,726	1250
1	Max.	0,704	1209
	Min.	0,608	1255
	Max.	0,692	1302
2	Max.	0,710	1221
	Min.	0,609	1243
	Max.	0,653	1271
	Min.	0,610	1290
3	Max.	0,712	1312
	Max.	0,717	

Abbildung 8: Messprotokoll 2

3	Max	0,688	1194
	Min	0,440	1216
	Max	0,672	1240
	Min	0,632	1262
	Max	0,654	1276
	Min	0,570	1298
	Max	0,737	1319

≈ 15 Breite des "Kastens": 136
Untergrundintensität: 0.006
Bildweite: 78 ± 0.5 cm
Brennweite: $80\text{ mm} \hat{=} 8\text{ cm}$
Abstand Einzelspalt-Linse = 8.5 ± 0.5 cm

Abbildung 9: Messprotokoll 3

Messprotokoll 2 : VZ33 FourieroptikAufgabe 1 :

Mike Brandt
Juan Provencio
03.03.2022

Das Objektbild wird scharf im Messprogramm dargestellt

> Abstand der Maxima $732 - 323 = 409$ Fehler 8-10 Pix.

> Breite der Maxima $392 - 256 = 136$ (links)
 $790 - 668 = 122$ (rechts)

> Bildweite : $(84,5 \pm 0,1) \text{ cm}$

Gegenstandsweite : $(8,8 \pm 0,1) \text{ cm}$

Brennweite : 80 mm

> Spaltbreite 2 Gaußkurven : $(0,27 \pm 0,01) \text{ mm}$

> Spaltbreite Plateau : $(0,07 \pm 0,01) \text{ mm}$

Abbildung 10: Messprotokoll 4

4.2 Auswertung

Im Folgenden wird bei der Fehleranalyse wenn nicht anders explizit angegeben die Gaußsche Fehlerfortpflanzung benutzt um die Fehlern der Größen zu bestimmen. Diese wird explizit in der digitalen Auswertung durch Python und wird in trivialen Fällen nicht nochmal bei der Ausarbeitung vorkommen.

4.2.1 Quantitative Beobachtungen am Einfachspalt

Eichung der Abszisse

Um unsere Messergebnisse, welche in der Einheit von Pixels vorhanden sind zu benutzen, müssen wir ein Verhältnis der Skala zwischen Pixel und Milimeter erstellen. Dafür haben wir eine Messreihe durchgeführt, aus welcher wir die Datenpunkte für eine Eichung benötigen. Wir passen darauf eine Gerade an.

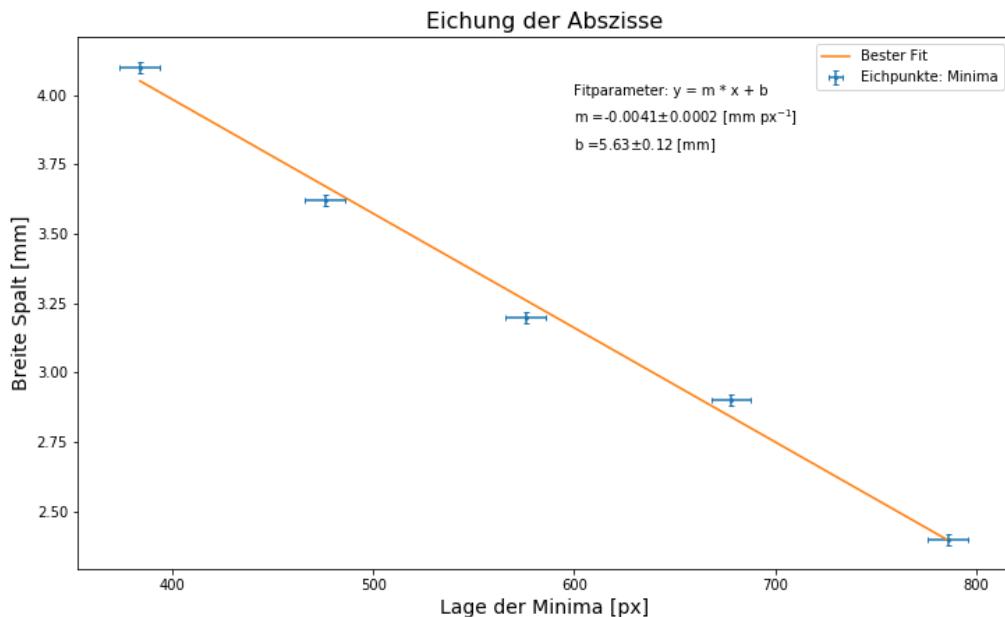


Diagramm 1: Eichungsgerade

Wir erhalten eine Steigung von

$$m = -0,0041(2) \text{ mm px}^{-1} \quad (14)$$

Beugungsstrukturen

Als nächstes wird die Lage der Minima in Pixels gegen ihre entsprechende Beugungs-

ordnung aufgetragen und eine Gerade darauf angepasst. Die Steigung dieser Gerade sollte dem mittleren Abstand zwischen Minima entsprechen.

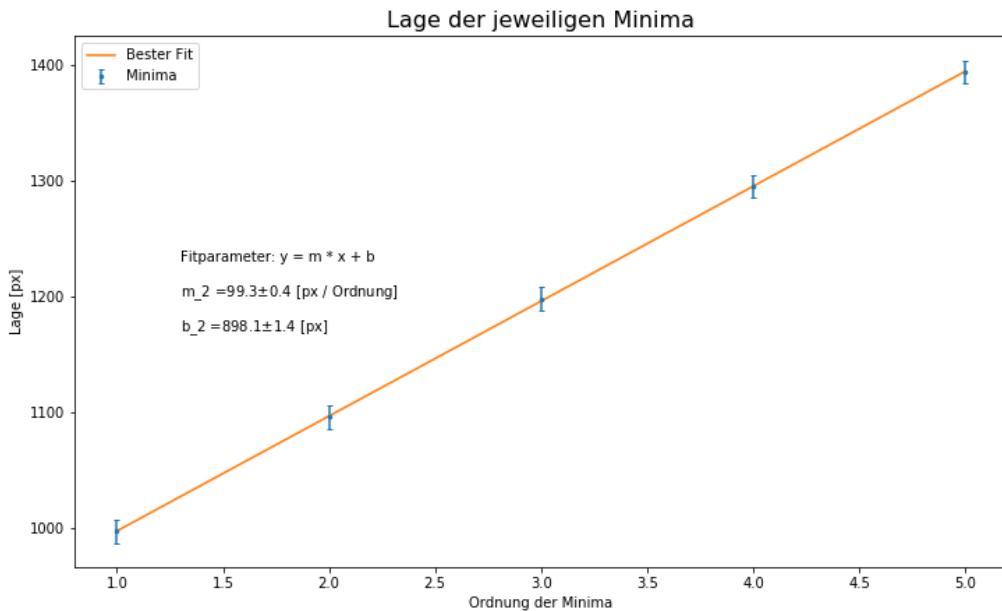


Diagramm 2: Lage der Minima und ihre Ordnung

$$m_2 = 99,3(4) \text{ px/Ordnung} \quad (15)$$

Diesen Wert vergleichen wir mit dem aus den Messwerten bestimmten mittleren Abstand

$$\overline{\Delta \text{min}} = 99(1) \text{ px} \quad (16)$$

Nun legen auf Diagramm 2 auch die Werte für die Maxima. Damit sie zwischen den Minima liegende müssen wir ihre Position auf der Ordinate berechnen durch

$$n = \frac{\max_{\text{Lage}} - b_2}{m_2} \quad (17)$$

mit dem Fehler

$$\Delta n = \sqrt{\left(\frac{\Delta \max_{\text{Lage}}}{m_2}\right)^2 + \left(\frac{\Delta b_2}{m_2}\right)^2 + \left(\frac{\max_{\text{Lage}} - b_2}{m_2^2} \cdot \Delta m_2\right)^2} \quad (18)$$

Es ergibt sich für die Lage der Maxima

Tabelle 5: Lage der Maxima

$n_{\max, \text{exp}}$	-0.01(18)	1.35(19)	2.46(20)	3.54(23)	4.47(26)	5.42(29)
------------------------	-----------	----------	----------	----------	----------	----------

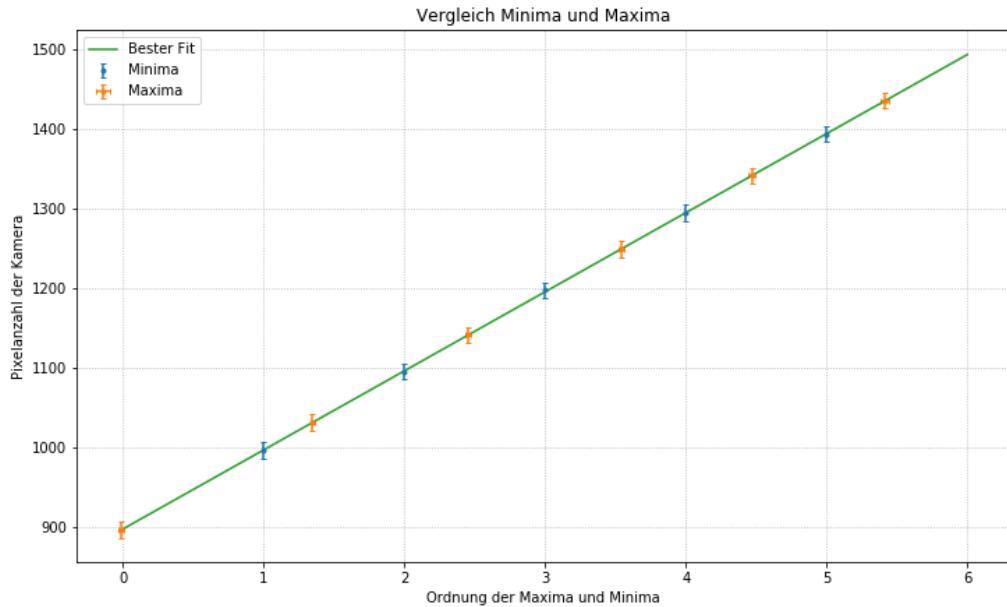


Diagramm 3: Ordnung der Maxima aufgetragen auf die Gerade der Minima

In der Theorie suchen wir die Maxima der sinc-Funktion welche sich ganz gut annähern lassen durch

$$x_{\text{theo}} \approx \left(n + \frac{1}{2}\right)\pi - \frac{1}{\left(n + \frac{1}{2}\right)\pi} \quad (19)$$

Um diese zu vergleichen werden wir die von uns gefundenen Maxima noch mit dem Faktor π anpassen sollen. Der Fehler des theoretischen Wertes ist deutlich überschätzt, es wurde angenommen, dass der Fehler höchstens 1 / 100 beträgt (Wikipedia, 2022).

Tabelle 6: Angepasste Lage der Maxima, experimentell und theoretisch

x_{exp}	-0,03(6)	4,24(6)	7,72(6)	11,13(7)	14,04(8)	17,02(9)
x_{theo}	0,000(9)	4,50(5)	7,73(8)	10,90(11)	14,07(14)	17,22(17)

Spaltbreite

Wir wollen als nächstes die Spaltbreite bestimmen. Dies lässt sich nach folgenden

Verhältnisse bestimmen.

$$\frac{n\lambda}{b} = \sin \alpha \quad (20)$$

$$\frac{d}{f_1} = \tan \alpha \quad (21)$$

Es lässt sich außerdem durch Kleinwinkelnäherung folgendes Verhältnis benutzen: $\sin \alpha \approx \tan \alpha$, woraus sich ergibt:

$$\frac{n\lambda}{b} = \frac{d}{f_1} \quad (22)$$

Hier benutzen wir die Wellenlänge $\lambda = 635 \text{ nm}$, die Brennweite $f_1 = 80 \text{ mm}$, die Position der Minima d und schließlich die Spaltbreite b . Für die Position der Minima müssen wir den Eichfaktor m und die gemessene Lage der Minima, welche in Pixel angegeben ist.

$$d = m \cdot \min_{\text{Lage}} \quad (23)$$

mit dem Fehler

$$\Delta d = d \cdot \sqrt{\left(\frac{\Delta m}{m}\right) + \left(\frac{\Delta \min_{\text{Lage}}}{\min_{\text{Lage}}}\right)} \quad (24)$$

Dann können wir die Position d gegen die Ordnungen n auftragen und daraus die Steigung

$$m_3 = \frac{d}{n} \quad (25)$$

bestimmen. Es gilt also

$$b = \lambda f_1 m_3^{-1} \quad (26)$$

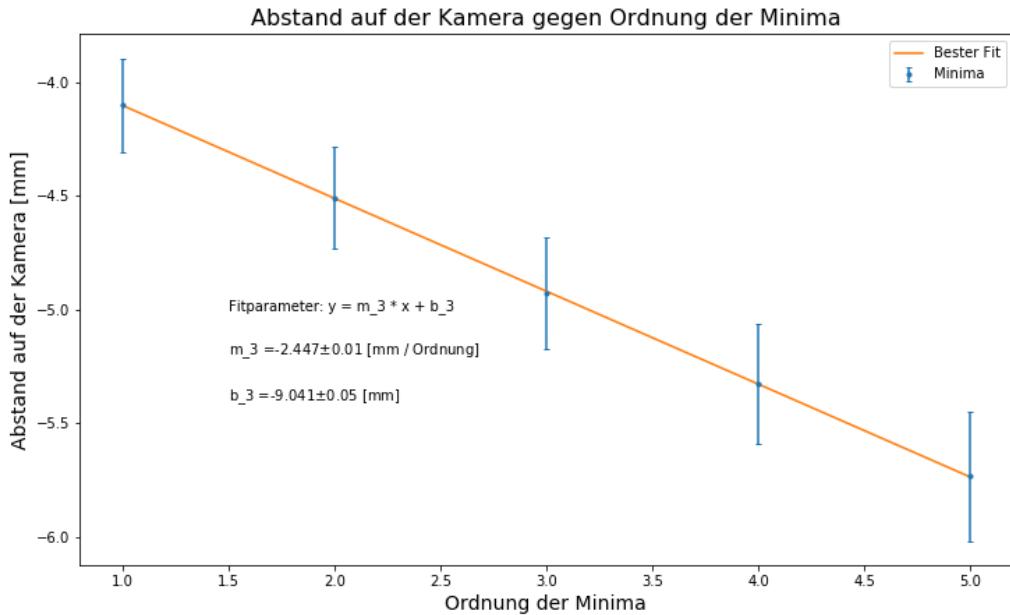


Diagramm 4: Anpassung des Abstands auf der Kamera und Ordnung der Minima

Aus der linearen Anpassung erhalten wir für die (inverse) Steigung m_3 den Wert

$$m_3 = -2,447(10) \text{ mm/Ordnung} \quad (27)$$

und eingesetzt in die Spaltbreite bekommen wir

$$b = 207,6(9) \cdot 10^{-6} \text{ m} \quad (28)$$

Intensitätsverhältnisse

Als nächstes vergleichen wir die Intensitätsverhältnisse der Maxima. Es wurde für eine Intensität des Lasers die Intensität der ersten Maxima gemessen, und damit die anderen Maxima gut erkennbar waren mit einer niedrigeren Intensität die der weiteren. Von den gemessenen Intensitäten ziehen wir noch den Untergrund ab und bestimmen die Intensitätsverhältnisse gemäß

$$v_i = \frac{I_i}{I_0} \quad (29)$$

Für die Verhältnisse mit der zweiten Intensität wurde das Verhältnis mit dem Faktor v_1 noch angepasst. Man kann an einer normierten sinc^2 Funktion die theoretischen Intensitätsverhältnisse ausrechnen, diese sind ebenfalls auf Tabelle 7 aufgetragen. Wir erhalten

Tabelle 7: Intensitätsverhältnisse des Einzelspalts

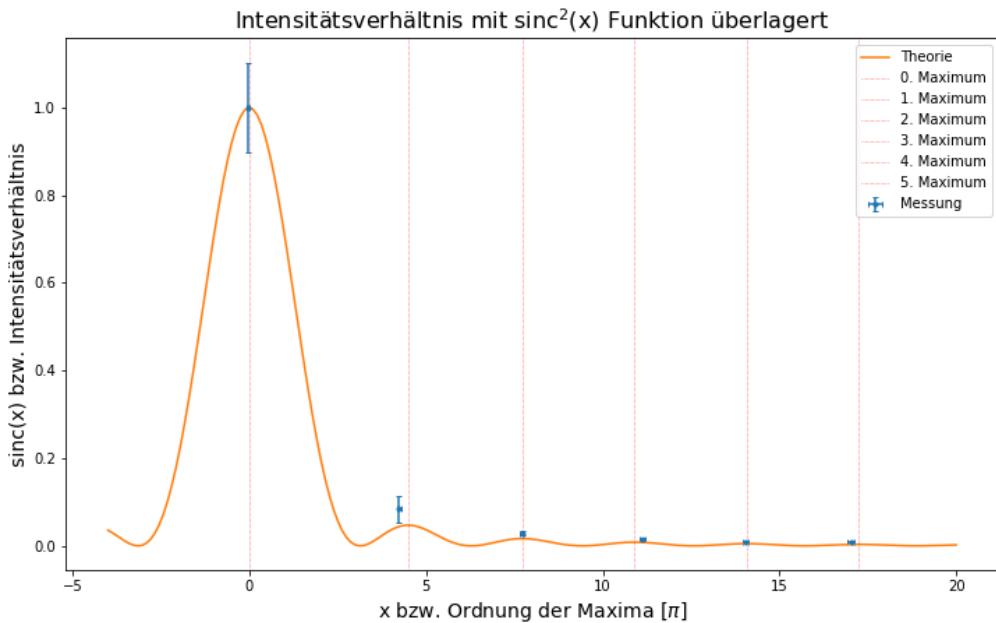
v_{exp}	1,00(10)	0,08(3)	0,028(5)	0,014(4)	0,009(3)	0,007(3)
v_{theo}	1	0,047(8)	0,016(5)	0,0083(16)	0,0050(1)	0,0034(22)

Der Fehler der theoretischen Intensitäten wurde mit der Gaußschen Fehlerfortpflanzung wie folgt ausgerechnet:

$$\Delta v_{\text{theo}} = \left| \frac{\partial \text{sinc}^2\left(\frac{x}{\pi}\right)}{\partial x} \cdot \Delta x \right| \quad (30)$$

$$= \left| \frac{2}{\pi} \cdot \left(\frac{\pi \cos \frac{x}{\pi}}{x} - \frac{\pi^2 \sin \frac{x}{\pi}}{x^2} \right) \text{sinc}\left(\frac{x}{\pi}\right) \cdot \Delta x \right| \quad (31)$$

Wir haben die von uns bestimmten Intensitäten zusätzlich auf einem Diagramm aufgetragen mit der $\text{sinc}^2 x$ -Funktion darüber überlagert. Im Optimalfall würden diese mit den Maxima zusammenfallen.

Diagramm 5: Intensitätsverhältnisse und $\text{sinc}^2 x$ -Funktion

4.2.2 Beugungsstruktur des Doppelspaltes

Theoretisches Beugungsbild

Nun kommen wir zum Beugungsbild des Doppelspaltes. In diesem Fall kann man für die Beugungsstruktur zwei Verhältnisse feststellen. Als erstes sieht man die Inten-

sitätsverteilung der $\text{sinc}^2 x \cdot \cos^2 x$ -Funktion, und darüber entspricht die einhüllende Funktion die der $\text{sinc}^2 x$, welche zu den Einzelspalten gehört.

Wir wollen als erstes das Verhältnis der Spaltbreite b und des mittleren Abstands d zwischen den zwei Spalten. Dieser Abstand d lautet nach Messprotokoll $d = 409 \text{ px}$ und die Breite des Spaltes $b = 136 \text{ px}$, also ist

$$v_{d/b} = \frac{d}{b} = 3,01(23) \quad (32)$$

Man kann $v_{d/b}$ benutzen um das richtige Verhältnis zwischen Beugungsstruktur des Doppelspaltes unter der sugehörigen Einhüllenden zu bestimmen. Dieses Verhältnis ist auf Diagramm 6 dargestellt. Auch auf dem Diagramm abgebildet ist der experimentelle Verlauf der Intensitätsstruktur was wir gemessen haben und die jeweiligen theoretischen und experimentellen Maxima. Die theoretischen Maxima wurden anhand dieses Kurvenverlaufs bestimmt, indem man mit Python's scipy Modul die Extrema der Distribution gesucht hat und damit auch die entsprechenden Lagen. Auf dem Diagramm abgebildet sind allerdings nicht die Maxima aus dem Messprotokoll, denn die Konversion von Pixels auf das Diagramm nicht zu 100% Genauigkeit stimmte. Zur Auswertung werden wir die Intensitäten aus dem Messprotokoll und nicht zusätzlich die graphisch bestimmten.

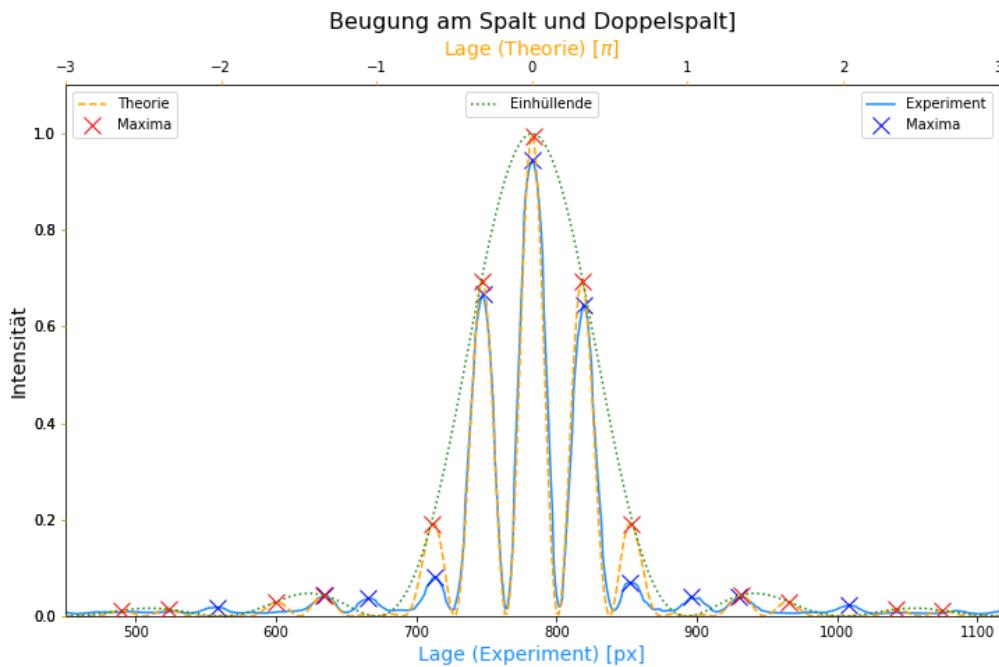


Diagramm 6: Intensitätsstruktur des Doppelspaltes mit einhüllender Intensitätsverteilung des Einzelspalts

Diese Struktur entsteht dadurch, dass jedes der Einzelspalte des Doppelspaltes die

Intensitätsstruktur des Einzelpaltes erzeugen, aber auch gleichzeitig miteinander interferieren, weshalb wir das obige Muster bekommen.

Intensitätsverhältnisse

Wie wir auch beim Einzelpalt gemacht haben, wollen wir als nächstes die Intensitätsverhältnisse vergleichen. Dadurch, dass die theoretischen Werte mithilfe graphischer Werkzeuge gefunden wurden, beschränkt sich unsere Genauigkeit bis auf die Anzahl an einzelnen x -Werten, worüber die Kurve geplottet wurde. In unserem Fall haben wir einen Array aus 400 Einträge zwischen $x \in [-3, 3]$, weshalb wir einen Fehler von 1/200 einschätzen.

Tabelle 8: Intensitätsverhältnisse des Doppelpalts

$v_{D,\text{exp}}$	1,00(10)	0,68(9)	0,085(25)	0,040(17)	0,043(18)	0,023(13)
$v_{D,\text{theo}}$	0,995(5)	0,695(5)	0,192(5)	0,043(5)	0,029(5)	0,014(5)

4.2.3 Fouriersynthese des Beugungsbildes

Spaltweite des Einzelpaltes

Im Folgenden benötigen wir die Spaltweite des verwendeten Einzelpalts bestimmen. Wir kennen dafür die Linsengleichung

$$\frac{1}{g} = \frac{1}{f} + \frac{1}{b} \quad (33)$$

und die Vergrößerungsformel

$$\frac{b}{g} = \frac{B}{G} \quad (34)$$

mit den Größen g für die Gegenstandsweite, b für die Bildweite, G für die Gegenstandsgröße und B für die Bildgröße. Wir erhalten nun folgende Formel für die Gegenstandsgröße

$$G = \frac{B}{\frac{b}{f} - 1} \quad (35)$$

Unsere Bildgröße ist allerdings in Pixels gegeben, weshalb wir den Proportionalitätsfaktor der Pixels zu mm auf der Kamera brauchen. Ein Pixel entspricht $14 \cdot 10^{-6}$ m. Wir erhalten also

$$G_m = 14 \cdot G \quad |\Delta G_m| = \sqrt{\left(\frac{\Delta B}{\left(\frac{b}{f} - 1\right)}\right)^2 + \left(\frac{B \cdot \Delta b}{\left(\left(\frac{b}{f} - 1\right)^2 \cdot f\right)}\right)^2} \quad (36)$$

Wir erhalten

$$G_m = 218(17) \cdot 10^{-6} \text{ m} \quad (37)$$

Vergleich zwischen theoretischen und experimentellen Objektbildern

Wir haben als nächstes am Analysiertspalt einzelne Beugungsordnungen ausgeblendet und mit jeder dann ein groberes Bild für die Beugungsstruktur bekommen. Wir wollen die von uns beobachteten modifizierten Beugungsbildern mit den theoretischen vergleichen. Die Theorie dahinter ist in der Einleitung unter (2.2.1) zu finden.

Wir schauen uns den Fall $n = \{0, 1, 2, 13\}$ an. Im Anhang befinden sich die Darstellung der experimentellen, der theoretischen und schließlich beide Werten zusammen. Wir schauen uns nur den Vergleich an. Der Vergleich wurde gemacht, indem man beide Werte auf einen gemeinsamen Plot aufgetragen wurden, und manuell wurden die Abstände und Definitionsbereiche so eingestellt, dass die Bilder möglichst gut übereinander liegen. Da die Skala der angegebenen Intensität auf dem Bildschirm während des Versuchs uneindeutig ist, muss man als erstes das Bild 0. Ordnung skalieren, damit er als Referenzwert dienen kann, dann werden alle weitere Bilder durch dieses Maximum geteilt um die Intensität auf diesen Wert anzupassen.

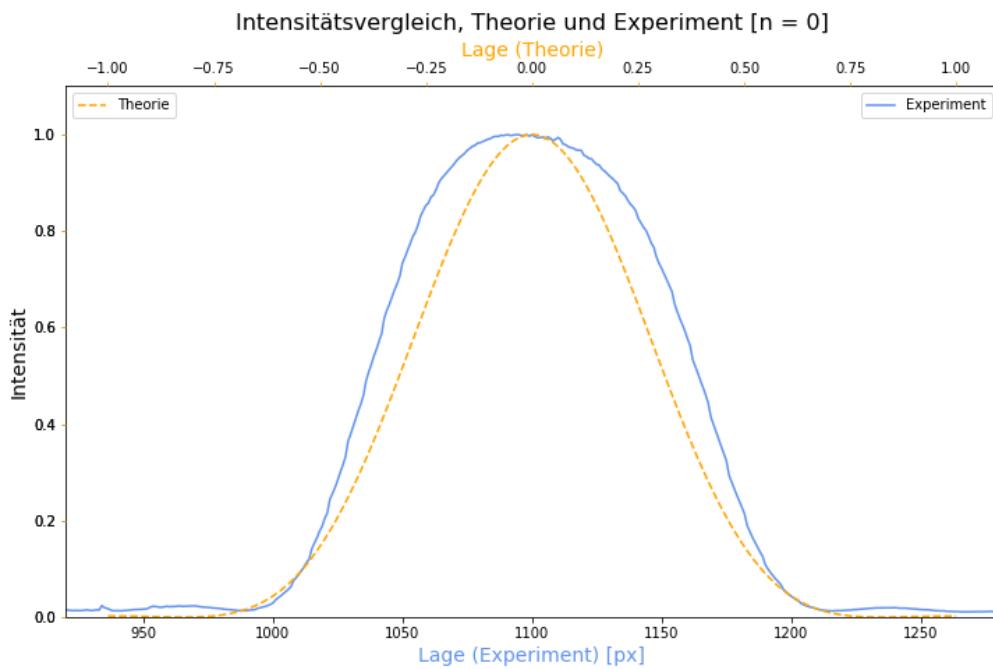
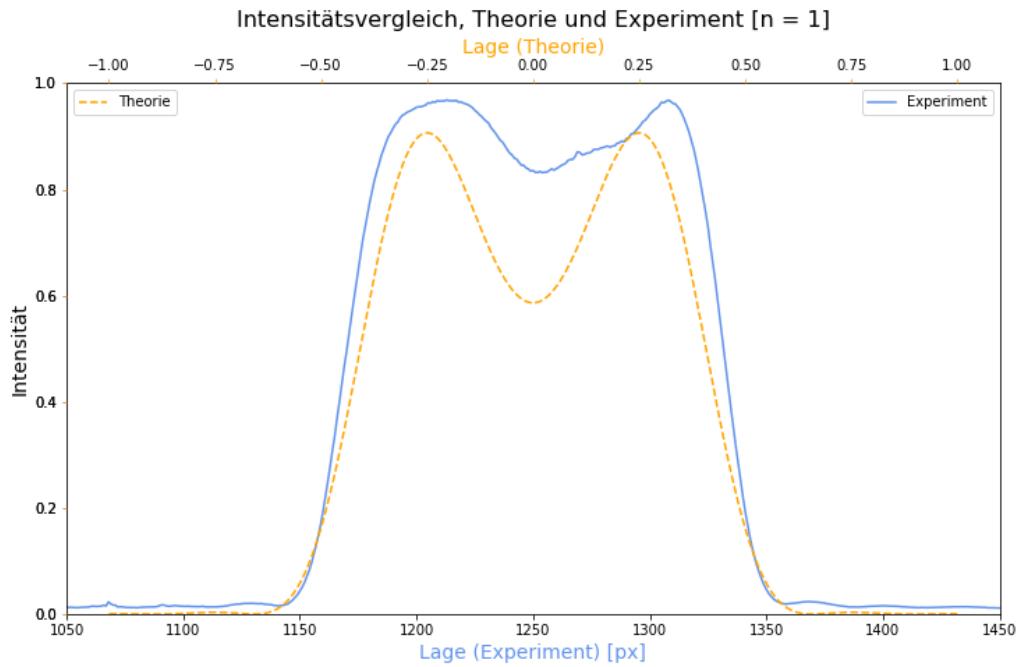
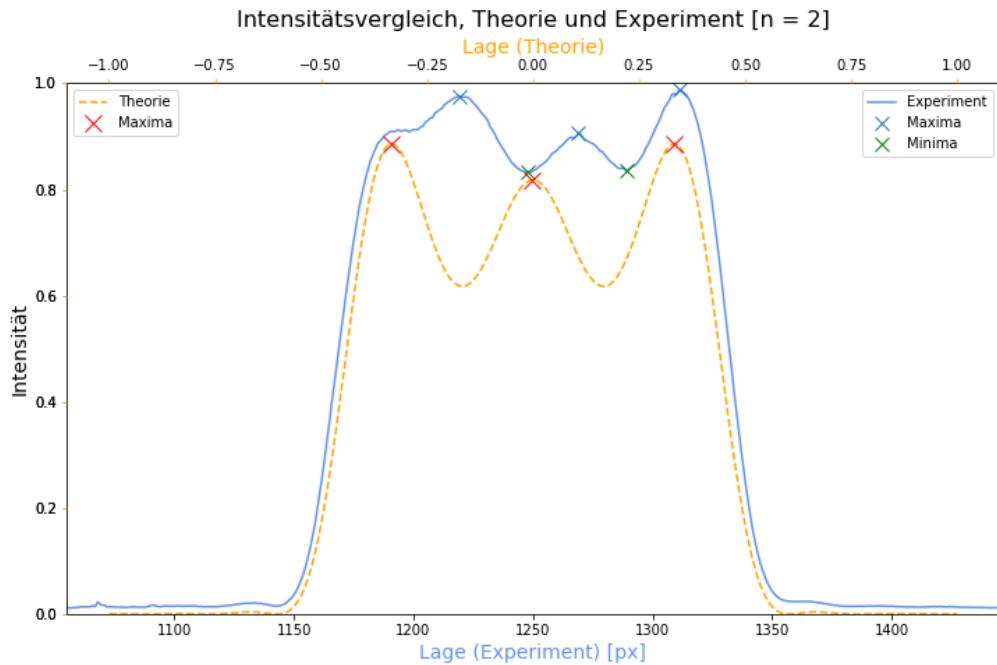


Diagramm 7: Intensitätsvergleich bei $n = 0$

Diagramm 8: Intensitätsvergleich bei $n = 1$

Auf dem Diagramm nicht abgebildet sind hier die Lagen der jeweiligen Maxima bzw. Minima. Wir stellen diese exemplarisch für die Ordnung $n = 2$ dar, aber eine gemeinsame Auftragung dieser führt zu großer Unsauberkeit auf den Diagrammen und geringe Übersichtlichkeit.

Diagramm 9: Intensitätsvergleich bei $n = 2$

Schließlich schauen wir uns die höchste Ordnung an, wir haben insgesamt 14 Maxima auf dem Bild deutlich darstellen können.

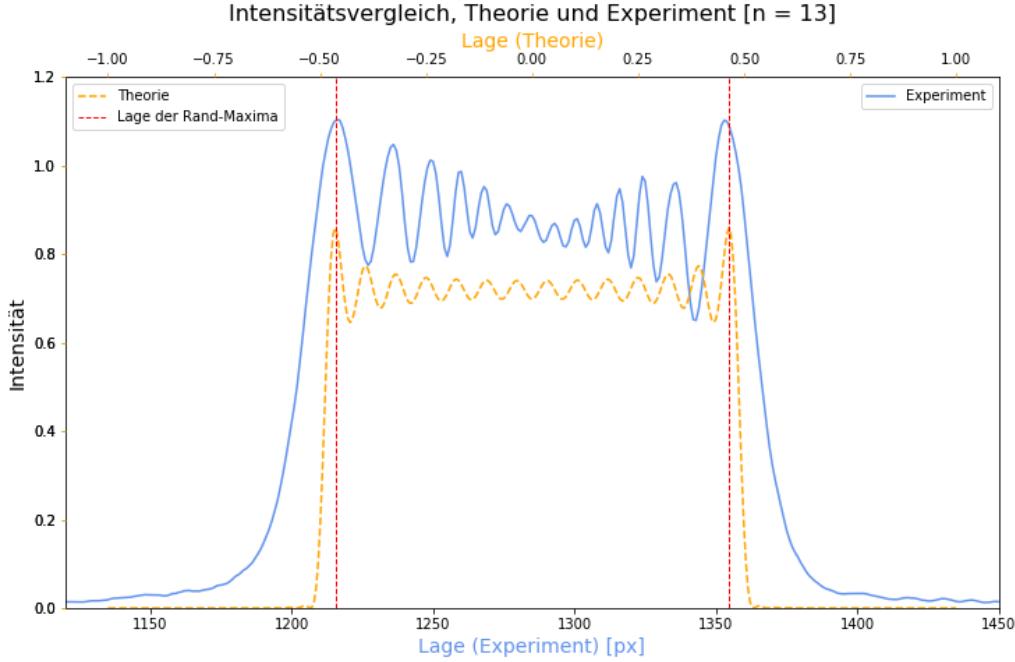


Diagramm 10: Intensitätsvergleich bei $n = 13$

Die Intensitäten der Maxima und Minima haben wir im Anhang von Python ausdrucken lassen. Wir untersuchen die theoretischen und experimentellen Intensitätsverhältnissen der ersten drei Bildern (das heißt für $n = \{1, 2, 3\}$). Die Ordnung der Maxima wird von oben nach unten in aufsteigender Reihenfolge angegeben. Bei den experimentell bestimmten Intensitäten haben wir nun zwei Möglichkeiten zur Verfügung: Wir haben auf dem Messprotokoll die Intensitäten angegeben, und wir können sie zusätzlich mithilfe von den Daten überprüfen, indem wir eine Suche der Maxima mit Python machen.

Für die rechnereisch bestimmte Intensitätsverhältnisse hilft eine normale Angabe des Fehlers durch den relativen Fehler für ein Verhältnis der Form:

$$v_i = \frac{I_1}{I_0} \quad (38)$$

Das graphisch bestimmte Verhältnis bestitzt einen Fehler, welcher den Einschränkungen der Auflösung auf der x -Achse untergeordnet ist. Dadurch, dass wir mit 2048 Pixels auf der Kamera zählen, ist die Genauigkeit ziemlich gut, wir gehen von einem Fehler von etwa $\Delta I_{\text{graphisch}} = 0,0010$ aus.

Schließlich besitzt auch der theoretische Wert einen Fehler in einer ähnlichen Größenordnung wie der graphisch, den auch hier ist die Auflösung der x -Achse entscheidend.

Tabelle 9: Intensitätsverhältnisse der Maxima der ersten drei Beugungsstrukturen am Einzelpunkt

Ordnung n	Extrema	$I_{\text{ext, exp, graphisch}}$	$I_{\text{ext, exp}}$	$I_{\text{ext, theo}}$
0	1. Max	1	1	1
1	1. Max	0,9688(10)	0,969(19)	0,9048(10)
	1. Min	0,839(1)	0,833(18)	0,5881(10)
	2. Max	0,9685(10)	0,953(19)	0,9048(10)
2	1. Max	0,9762(10)	0,978(20)	0,8859(10)
	1. Min	0,8323(10)	0,838(18)	0,6175(10)
	2. Max	0,906(1)	0,899(19)	0,8178(10)
	2. Min	0,8351	0,839(18)	0,6178(10)
	3. Max	0,9872(10)	0,987(20)	0,8859(10)
3	1. Max		0,947(19)	0,8767(10)
	2. Max		0,925(19)	0,798(1)
	3. Max		0,900(19)	0,798(1)
	4. Max		1,015(20)	0,8767(10)

Genauereres wird in der Diskussion erläutert, aber für die zu Verfügung stehenden graphisch bestimmten Intensitäten wird ein Vergleich mit den experimentell aufgeschriebenen Intensitäten gemacht, und diese werden nun auch mit den theoretischen Intensitäten verglichen. Es ist auffällig, am offensichtlichen anhand der graphischen Vergleichen an den Abbildungen Diagramm 7, 8, 9 und 10, dass die experimentellen Werten deutlich über den theoretischen Wert liegen.

Große Beugungsordnung

Schließlich schauen wir uns ein Beispiel mit einer großen Anzahl an Wiggles. Wir zählen insgesamt 13 Minima also ist die Ordnung $n = 13$.

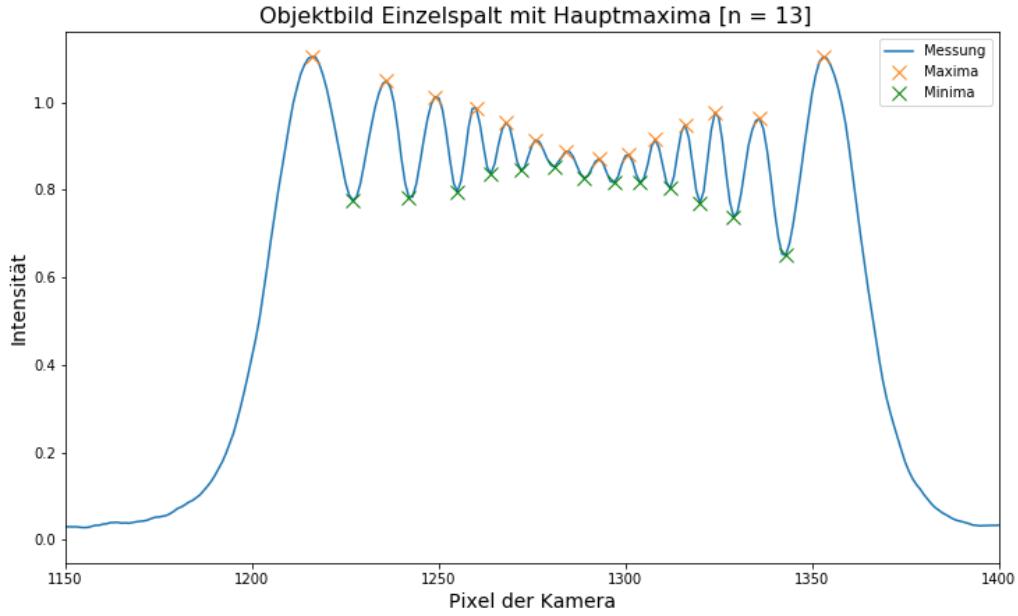


Diagramm 11: Beugungsstruktur 13. Ordnung

Uns interessiert hier den Abstand zwischen der Maxima bzw. Minima. Die Lage dieser haben wir durch Python bestimmt in einer Suche der Peaks, bzw. wenn diese gescheitert hat durch eine manuelle Suche. Hier war dies aber nur ein Mal der Fall. Für die Maxima erhalten wir einen mittleren Abstand von

$$\overline{\Delta x}_{\text{max, exp}} = 10,5(1,0) \text{ px} \quad (39)$$

und für die Minima

$$\overline{\Delta x}_{\text{min, exp}} = 9,7(7) \text{ px} \quad (40)$$

Der Fehler wurde als Fehler des Mittelwertes angegeben. Obwohl hier der relative Fehler zwischen 7-10% liegt, haben wir zwischen den Maxima bzw. Minima an den Rändern und in der Mitte große Unterschiede. Am Rand sind die Abstände über 2 Mal größer als in der Mitte. Für das theoretische Bild ist dies deutlich genauerer, wir erhalten die Ergebnisse in der normierten Einheit. Um diese in Pixels zu übersetzen haben wir die Abstände der Randmaxima experimentell und theoretisch ausdrucken lassen, und aus ihrem Verhältnis die Konversion für die normierte Einheiten auf Pixels berechnet. Wir erhalten somit

$$v_{\text{norm, px}} = 148(6) \text{ px} \quad (41)$$

$$\overline{\Delta x}_{\max, \text{ theo}} = 0,071(1) \cdot v_{\text{norm}} = 10,56(44) \text{ px} \quad (42)$$

und für die Minima

$$\overline{\Delta x}_{\min, \text{ theo}} = 10,56(44) \text{ px} \quad (43)$$

4.2.4 Fourierbild des Doppelspaltes

Spaltbreite und Spaltmittelabstand

Wir wollen nun wieder die Spaltbreite bestimmen und hier auch den Spaltmittelabstand. Mit den Messwerten aus dem Messprotokoll erhalten wir für die Spaltbreite

$$G_{m, 2} = 199(15) \cdot 10^{-6} \text{ m} \quad (44)$$

und

$$G_{S-M-\text{Abstand}} = 599(15) \cdot 10^{-6} \text{ m} \quad (45)$$

Mit diesen Werten können wir nun auch die theoretischen Beugungsbilder erstellen.

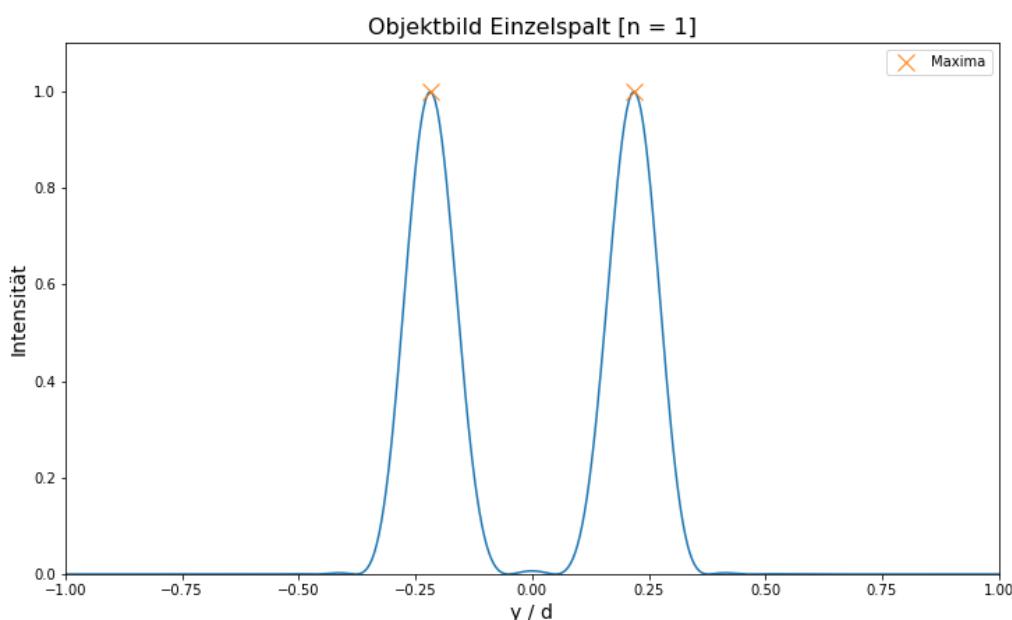


Diagramm 12: Doppelspaltstruktur

Als Vergleich zu dieser Struktur wurde ein Bild der Intensitäten gemacht:

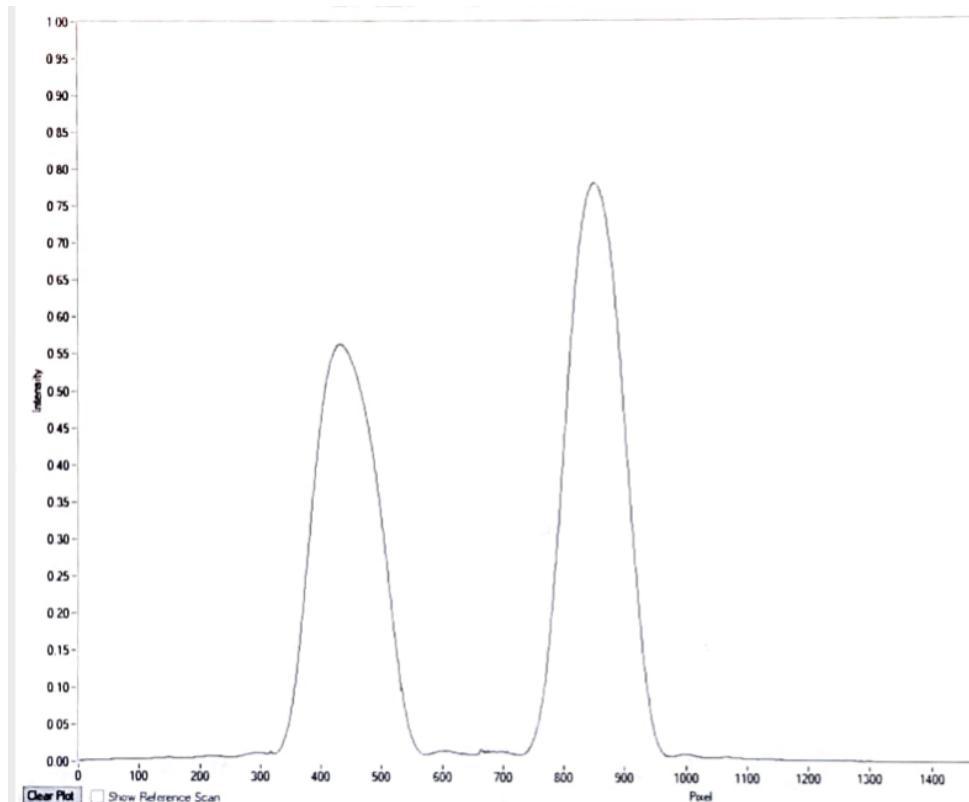
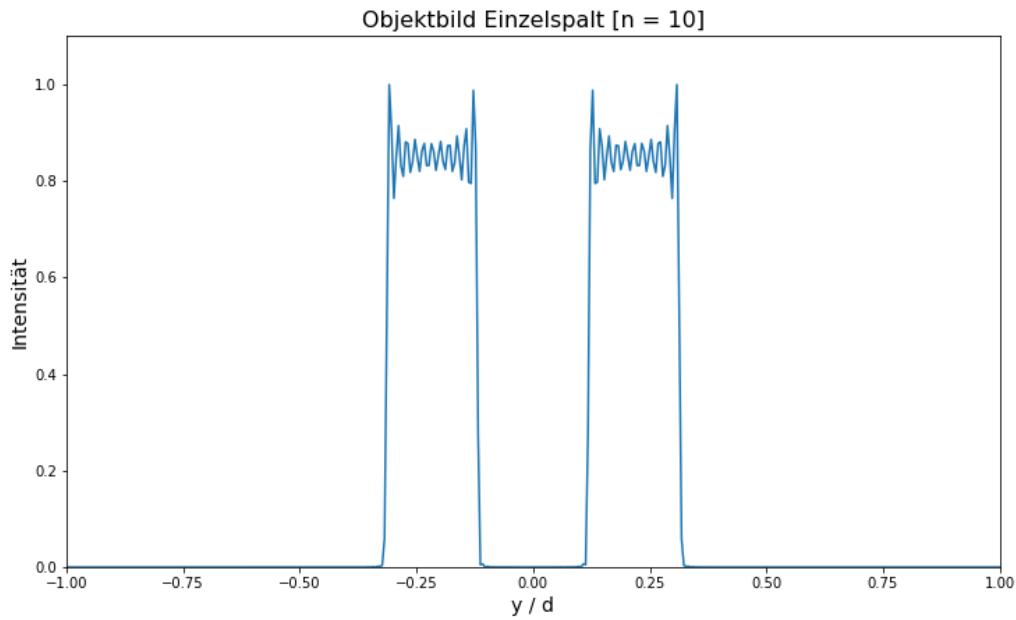


Abbildung 11: Doppelspaltstruktur, experimentell beobachtet

Man kann ein leicht asymmetrisches Verhältnis beobachten, was an einer falschen Justierung bei der Durchführung des Versuchs liegen soll. Allerdings ist qualitativ eine ähnliche Struktur zur Theorie erkennbar.

Für eine etwas höhere Ordnung, die höchste die Python ohne Probleme bei der numerischen Integrtion zulässt erhalten wir folgendes Bild

Diagramm 13: Doppelspaltstruktur der Ordnung $n = 10$

Auflösung der Doppelspaltstruktur und Grenzraumfrequenz

Wir wollen schauen, für welche Breite des Analysierspalts die Doppelspaltstruktur verschwindet, und stattdessen ein "Plateau" in der Mitte gebildet wird.

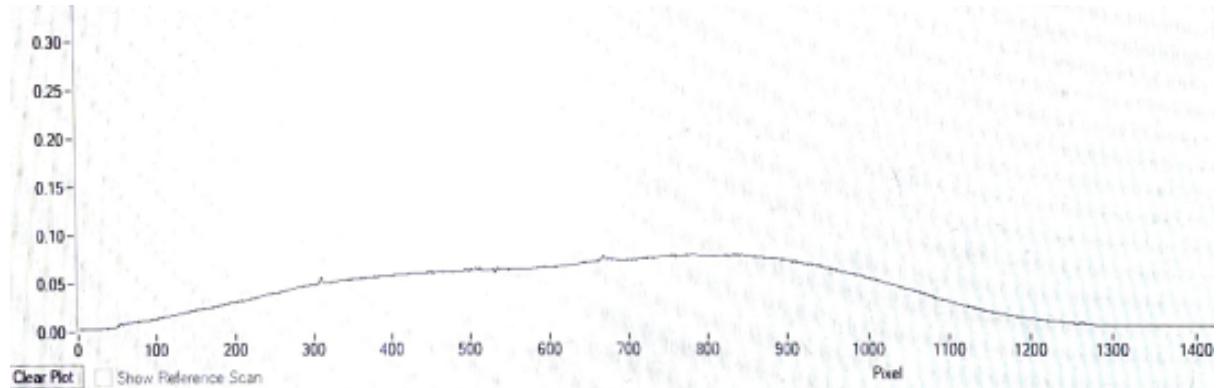


Abbildung 12: Auflösung der Doppelspaltstruktur

Dies geschieht bei der sogenannten Grenzraumfrequenz k_y . Diese ist gemäß (7) wie folgt definiert

$$k_y = \frac{2\pi n}{d} \quad (46)$$

Mit n als die Ordnung, bei der dieses Plateau auftaucht (Fall b) und $d = G_{m,2}$ die

Spaltbreite. Die Spaltbreite haben wir in angegeben. Die Plateau-Ordnung bestimmt man durch feines Variieren des n -Parameters bei der digitalen Erstellung des Beugungsbildes, um es möglichst fein zu machen haben wir uns aus dem Diagramm die Maxima ausdrucken lassen und den Wert n angegeben, ab welchem nur noch eine Maxima auftaucht. Der Fehler wurde aus der nächsten Ordnung festgestellt. Für Fall a ist die Doppelspaltstruktur wie zwei perfekte Gauss-Verteilungen zu sehen, also mit $n = 1$

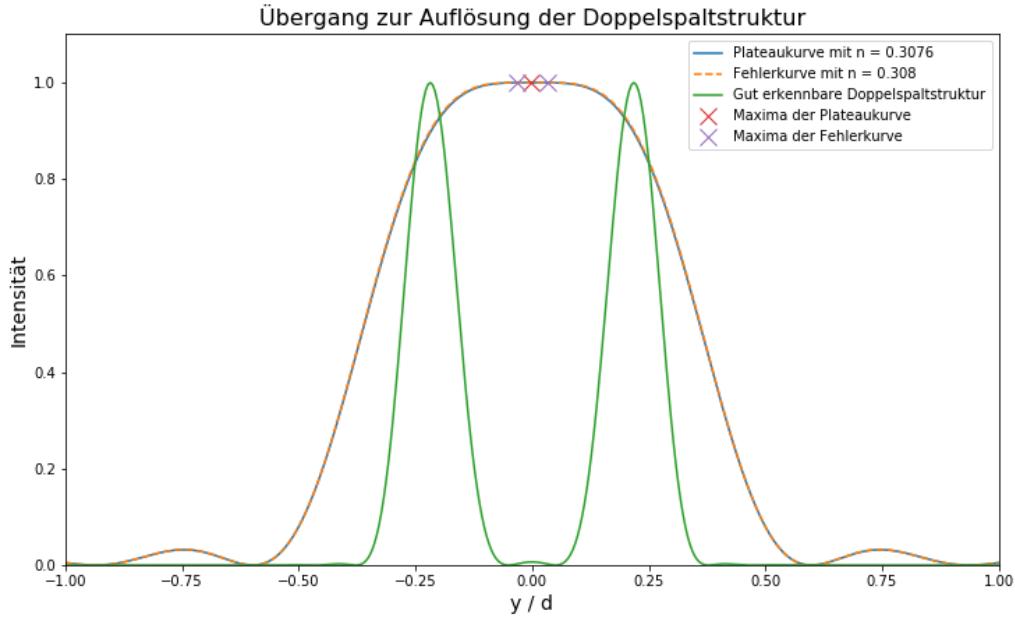


Diagramm 14: Auflösung der Doppelspaltstruktur

Wir erhalten

$$n = 0,3076(4) \quad (47)$$

und somit

$$k_{y, b, \text{theo}} = 9,7(7) \cdot 10^{-3} \text{ m}^{-1} \quad (48)$$

Für Fall a mit $n = 1$ ist es

$$k_{y, a, \text{theo}} = 31,6(2,3) \cdot 10^{-3} \text{ m}^{-1} \quad (49)$$

Experimentell lassen sich diese Grenzraumfrequenzen gemäß (11) bestimmen als

$$k_y = \frac{2\pi \sin \alpha}{\lambda} = \frac{2\pi d_a}{\lambda f} \quad (50)$$

Wir erhalten hier

$$k_{y, b, \text{exp}} = 33,4(1,7) \cdot 10^{-3} \text{ m}^{-1} \quad (51)$$

und

$$k_{y, a, \text{exp}} = 8,7(4) \cdot 10^{-3} \text{ m}^{-1} \quad (52)$$

Diese Messung schließt den Abschnitt der Auswertung. Über die Ergebnisse und möglichen Fehlerquellen werden wir uns im nächsten Teil unterhalten.

5 Zusammenfassung und Diskussion

Quantitative Beobachtungen am Einfachspalt

Im ersten Teil haben wir uns mit dem Einzelpalt beschäftigt. Hier war als erstes nötig, eine Konversion zwischen dem Abstand in Pixels auf dem Monitor und dem Abstand in Metern zwischen verschiedenen Intensitätsextrema, weshalb wir zuerst eine Eichmessung durchgeführt haben, aus welcher wir das Verhältnis von Millimeter zu Pixels erhalten haben.

Danach haben wir die Position der Minima auf ein Diagramm aufgetragen und darüber eine lineare Anpassung durchgeführt. Die Steigung dieser Gerade entspricht dem Abstand in Pixels auf die nächste Ordnung. An dieser Stelle hätten wir erwartet, dass die Steigung der Gerade dem mittleren Abstand zwischen den Minima entspricht. Die σ -Abweichung für eine Größe G wird folgendermaßen berechnet:

$$\frac{|G - G_{\text{Lit}}|}{\sqrt{(\Delta G)^2 + (\Delta G_{\text{Lit}})^2}} \quad (53)$$

Wir haben folgende Ergebnisse bekommen:

Tabelle 10: Vergleich der Steigung der Minima und Mittlerer Abstand

m_2 [px]	$\frac{\Delta m_2}{m_2}$ [%]	$\overline{\Delta \text{min}}$ px	$\frac{\Delta \overline{\Delta \text{min}}}{\overline{\Delta \text{min}}}$ [%]	σ
99,3(4)	0,4	99(1)	1	0,05

Hier erkennt man eine ausgezeichnete σ -Abweichung zwischen beiden Messungen. Dadurch, dass der relative Fehler auch ziemlich klein ist, im Bereich von 1% ist, können wir ausschließen dass die Fehlern zu groß abgeschätzt wurden und daraus folgern, dass die Messung ziemlich gut ist und der Fit erfolgreich war.

Als nächstes haben wir die Maxima ebenfalls auf die Gerade drangehängt. Hierfür mussten wir die Lage der Maxima anhand der Fitparametern anpassen. Diese Werte wollen wir mit der Theorie vergleichen, wofür wir sie zuerst mit dem Faktor π anpassen müssen. Diese sind in Tabelle 6 angegeben, aber wir schreiben sie trotz der Redundanz nochmal auf. An dieser Stelle ist ein relativer Fehler an den meisten Ordnungen nicht sinnvoll, denn die kleineren Ordnungen logischerweise einen lächerlich großen relativen Fehler besitzen werden, sinnvoller ist der absolute Fehler zwischen den Messungen

Tabelle 11: Vergleich der angepassten Lage der Maxima

x_{exp}	-0,03(6)	4,24(6)	7,72(6)	11,13(7)	14,04(8)	17,02(9)
x_{theo}	0,000(9)	4,50(5)	7,73(8)	10,90(11)	14,07(14)	17,22(17)
$x_{\text{exp}} - x_{\text{theo}}$	-0,03	-0,26	-0,01	0,23	-0,02	-0,20
σ	0,6	4,5	0,16	3,2	0,28	2,26

Hier schwanken die σ -Abweichungen zwischen sehr gut und inakzeptabel für die einzelnen Maxima. Außerdem ist es ein Kommentar wert, dass alle bis auf eine absolute Differenz negativ sind. Das heißt die experimentell bestimmten Maxima meistens etwa kleiner sind als die theoretisch bestimmten. Dies könnte ein statistischer Fehler sein, in Anbetracht dass die Differenzen meistens ziemlich klein sind, aber ein systematischer Fehler ist nicht auszuschließen.

Als Hauptteil dieser Aufgabe wurde allerdings die Spaltbreite b gemessen. Diese haben wir ebenfalls im Versuchsteil (HEY) gemessen, weshalb wir dort den Vergleich machen werden.

Abschließen wurde noch eine Messung der Intensitäten gemacht. Für den Vergleich zur Theorie wurden die theoretischen Maxima aus Tabelle 11 in die sinc^2 Funktion eingesetzt.

Tabelle 12: Vergleich der Intensitätsverhältnissen des Einzelpalts

v_{exp}	1,00(10)	0,08(3)	0,028(5)	0,014(4)	0,009(3)	0,007(3)
v_{theo}	1	0,047(8)	0,016(5)	0,0083(16)	0,0050(1)	0,0034(22)
$v_{\text{exp}} - v_{\text{theo}}$		0,037	0,012	0,006	0,004	0,004
σ		1,2	1,6	1,4	1,4	1,1

Für die Intensitätsverhältnisse erhalten wir an dieser Stelle immer akzeptable σ -Abweichungen, allerdings auch merkwürdig, dass die experimentellen Werten bei allen Messungen über den theoretischen Werten liegen. Auf dieses Phänomen werden wir bei den weiteren Vergleichen zurückkommen.

Beugungsstruktur des Doppelspaltes

In diesem Abschnitt des Versuches wurde der Doppelspalt und seine Beugungsstruktur untersucht. Um die theoretischen Beugungsbilder erstellen zu können benötigten wir das Verhältnis der Spaltbreite b und des mittleren Abstands d . Mit diesem konnten wir das von uns erhaltene Beugungsbild mit dem theoretischen und mit der Einhüllenden Funktion des Einzelpalts vergleichen, was graphisch auf Diagramm 6 dargestellt ist. Hierzu folgt ein weiterer Vergleich der Intensitätsverhältnissen aus Tabelle 8.

Tabelle 13: Vergleich der Intensitätsverhältnissen des Doppelspalts

$v_{D,\text{exp}}$	1,00(10)	0,68(9)	0,085(25)	0,040(17)	0,043(18)	0,023(13)
$v_{D,\text{theo}}$	0,995(5)	0,695(5)	0,192(5)	0,043(5)	0,029(5)	0,014(5)
$v_{D,\text{exp}} - v_{D,\text{theo}}$	0,005	-0,015	-0,11	-0,0027	0,014	0,009
σ	0,05	0,17	4,2	0,15	0,8	0,7

Bei dieser Messung sind alle bis auf das dritte Intensitätsmaximum in sehr akzeptablen σ -Bereichen. Am Diagramm 6 kann man auch erkennen, dass obwohl das theoretische und das experimentelle Maximum von der Lage her miteinander übereinstimmen, ist das Theoretische etwa zwei Mal so groß wie das Experimentelle. Eine so große Abweichung genau an dieser Stelle lässt sich möglicherweise durch eine leicht fehlerhaften Einstellung des Versuchsaufbaus oder eine falsche Abschätzung des Verhältnisses zwischen Mittelabstand und Spaltbreite, was eine leicht unterschiedliche Beugungsstruktur ergeben hätte. Dies lässt sich insbesondere anhand der Maxima höherer Ordnung unterstützen, denn ab der 3. Ordnung liegen diese gar nicht in der Nähe der zugehörigen theoretischen Maxima. Ansonsten lässt sich sehr zufrieden über die Messungen sein.

Fouriersynthese des Beugungsbildes

An dieser Stelle haben wir nun wieder die Spaltbreite des Einzelpaltes bestimmt. Dieses Mal über eine Messung der Gegenstandsweite und Bildgröße, zusätzlich zur gegebenen Brennweite. Die Gegenstandsgröße sollte die Spaltbreite in Pixels entsprechen, weshalb wir noch um einen Proportionalitätsfaktor angepasst haben. Dieser Wert ist in (37) angegeben und wir vergleichen ihn mit dem Wert aus (28) und (44) wenn wir zur Diskussion des letzten Versuchsteils kommen.

Als nächstes folgte ein Vergleich zwischen den theoretischen und experimentellen Beugungsbildern. Wir haben nun dafür beide Beugungsbilder 0. Ordnung auf eine Amplitude von 1 normiert und damit die subsequenten Bilder übereinander ausdrucken lassen in Diagramm 7, 8, 9 und 10. Hier beobachten wir nun wieder das selbe Phänomen wie beim

Intensitätsvergleich auf Tabelle 12, denn alle Intensitätsmaxima, und allgemein die Intensitätsstruktur, konsistent über den theoretischen Werten liegen. Dieses Verhältnis haben wir explizit anhand der ersten 3 nicht trivialen Ordnungen $n = \{1, 2, 3\}$. Als kleiner Sanity Check haben wir die von uns gemessenen Maxima mit den aus den Dateien gefundenen Peaks durch Python ebenfalls aufgeschrieben. Wir begnügen uns an dieser Stelle zu bemerken, dass beide Werte jeweils in ihren zugehörigen Fehlerbereichen liegen und verzichten auf eine weitere Diskussion darüber. Für unseren Vergleich mit der Theorie nehmen wir die von uns aufgeschriebenen Extrema.

Tabelle 14: Vergleich der Intensitätsverhältnissen der Maxima der ersten drei Beugungsstrukturen am Einzelpalt

Ordnung n	Extrema	$I_{\text{ext, exp}}$	$I_{\text{ext, theo}}$	$I_{\text{ext, exp}} - I_{\text{ext, theo}}$	σ
0	1. Max	1	1		
1	1. Max	0,969(19)	0,9048(10)	0,06	3,2
	1. Min	0,833(18)	0,5881(10)	0,25	14
	2. Max	0,953(19)	0,9048(10)	0,05	2,4
2	1. Max	0,978(20)	0,8859(10)	0,09	4,7
	1. Min	0,838(18)	0,6175(10)	0,22	12
	2. Max	0,899(19)	0,8178(10)	0,08	4,3
	2. Min	0,839(18)	0,6178(10)	0,22	12
	3. Max	0,987(20)	0,8859(10)	0,10	5,2
3	1. Max	0,947(19)	0,8767(10)	0,07	3,7
	1. Min	0,603(16)	0,6286(10)	-0,026	1,6
	2. Max	0,925(19)	0,798(1)	0,13	6,7
	2. Min	0,869(18)	0,6494(10)	0,22	12
	3. Max	0,900(19)	0,798(1)	0,1	5,4
	3. Min	0,783(18)	0,6288(10)	0,15	8,7
	4. Max	1,015(20)	0,8767(10)	0,14	7

An dieser Tabelle erkennen wir eine große Diskrepanz zwischen den Ergebnissen. Bis auf zwei Werte, liegen alle Vergleiche außerhalb eines akzeptablen σ -Bereiches, und bei diesen zwei Werten können wir das wahrscheinlich am besten anhand des Zufalls erklären. Außerdem sind hier wieder alle bis auf eine Intensitätsdifferenz positiv. Dies legt offensichtlich einen systematischen Fehler da, der sich an einigen Sachen vielleicht erklären lassen würde. Als wahrscheinlich äußert sich ein menschlicher Fehler als verantwortlich dafür. Am Vergleich der Intensitäten 0. Ordnung auf Diagramm 7 erkennen wir, dass die experimentell gemessene Kurve deutlich schneller abflach als das theoretisch zu erwartende Bild. Dies führt vielleicht zur Schlussfolgerung, dass der Aufbaunicht richtig eingestellt wurde und die richtige Intensität ein bisschen höher liegen sollte damit die Kurve gaussischer wird, was die nachfolgenden Intensitätsmessungen näher an die kleineren theoretischen Bildern gebracht hätte. Natürlich, dieser Effekt hätte in den nächsten

Ordnungen die gleiche Rolle spielen sollen und dadurch den Fehler ausgleichen. Außerdem ist zu bemerken, dass wir dieses Phänomen ebenfalls bei einer anderen Teilaufgabe unter verschiedener Justierung gefunden haben, ist also ein Fehler in der Justierung des Versuches eher doch unwahrscheinlich. Eine weitere Erklärung würden die Instrumente selbst liefern. Die Kamera mag Defekte haben wie Spuren von Fett auf der Linse nach unvorsichtiges Anfassen oder intrinsische Baueigenschaften wie die Linearität oder Auflösung der selben. Darüber haben wir allerdings kein genaues Wissen, wie diese eine Rolle spielen könnten und auf dem Grund lassen sich mögliche Fehlern nicht darauf zurückführen.

Bei einer höheren Beugungsordnung haben wir überprüft, dass die Maxima und Minima zueinander möglichst äquidistant sind. Hierzu haben wir die Maxima und Minima durch Python gefunden bzw. wenn Python daran gescheitert durch manuelle Eingabe der bestmöglichen Punkten. Dies war allerdings nur bei Minima der Fall, da die verwendete Funktion vom scipy Package "find_peaks" große Probleme hatte die Minima zu erkennen. An einigen Stellen wurde auf eine weitere Funktion zurückgegriffen "argrelextrema / min", diese hat aber ebenfalls an einigen Stellen gescheitert, weshalb wir nun uns auf die höhlenmenschliche Methode gestützt haben, durch ausprobieren das bestmögliche Kreuz auf dem Diagramm zu legen. Wir haben uns an dieser Stelle begnügt durch das Sinnesorgan unserer Augen zu überprüfen, dass das Kreuz möglichst gut am Minimum gelegt worden war. Wir haben die selbe Methode zur Suche der Extrema sowohl beim experimentellen als auch beim theoretischen Bild und haben aus den einzelnen Abständen einen Mittelwert inklusiver Fehler des Mittelwertes ausdrucken lassen. Die einzelnen Werten sind im Anhang im Python-Skript angegeben, wir geben als nächstes nur die Mittelwerte an, die in Gleichung 39, (42) und (40) und (43) aufgeschrieben sind. Wir haben bei den theoretischen Werten eine zusätzliche signifikante Stelle genommen, da sonst die σ -Abweichung bei den Maxima verdächtigerweise 0 wird.

Tabelle 15: Vergleich der Äquidistant der Maxima

$\overline{\Delta x}_{\max, \text{exp}} [\text{px}]$	$\frac{\Delta \overline{\Delta x}_{\max, \text{exp}}}{\overline{\Delta x}_{\max, \text{exp}}} [\%]$	$\overline{\Delta x}_{\max, \text{theo}} [\text{px}]$	$\frac{\Delta \overline{\Delta x}_{\max, \text{theo}}}{\overline{\Delta x}_{\max, \text{theo}}} [\%]$	σ
10,5(1,0)	9,8	10,56(44)	4,1	0,017

Tabelle 16: Vergleich der Äquidistant der Minima

$\overline{\Delta x}_{\min, \text{exp}} [\text{px}]$	$\frac{\Delta \overline{\Delta x}_{\min, \text{exp}}}{\overline{\Delta x}_{\min, \text{exp}}} [\%]$	$\overline{\Delta x}_{\min, \text{theo}} [\text{px}]$	$\frac{\Delta \overline{\Delta x}_{\min, \text{theo}}}{\overline{\Delta x}_{\min, \text{theo}}} [\%]$	σ
9,7(7)	7,4	10,56(44)	4,1	1,1

An den einzelnen Messwerten erkennt man zwar bei den äußeren Extrema einen

Abstandsunterschied zu ihren nächstliegenden Extrema was doppelt so groß ist wie der Unterschied der inneren Extrema zueinander, aber insgesamt sind die Mittelwerte sehr gut innerhalb akzeptabler σ -Bereichen, ungleich wie die Intensitäten.

Fourierbild des Doppelspaltes

Nun sind wir beim letzten Teil unserer Zusammenfassung und Diskussion angekommen. An dieser Stelle wollten wir die Spaltbreite und den Spaltmittelabstand bei der Doppelspalte bestimmen. Der Einzelspalt dieser sollte äquivalent zum Einzelspalt aus den vorigen Teilaufgaben sein. Wir haben somit insgesamt drei Werte für die Spaltbreite

Tabelle 17: Vergleich Spaltbreite

Größe	Wert	Rel. Fehler [%]	σ	
$G_m [10^{-6} \text{ m}]$	218(17)	7,8	$\sigma_{G_m,b}$	0,6
$b [10^{-6} \text{ m}]$	207,6(9)	0,4	$\sigma_{G_m,G_m,2}$	0,83
$G_{m,2} [10^{-6} \text{ m}]$	199(15)	7,5	$\sigma_{G_{m,2},b}$	0,58

Jeder dieser Werte liegt innerhalb eines ziemlich guten σ -Bereich unter 1. Weshalb sich auf keinen systematischen Fehler hindeuten lässt.

Schließlich wollen wir die experimentell und theoretisch ausgerechneten Grenzraumfrequenzen vergleichen.

Tabelle 18: Vergleich Grenzraumfrequenz Fall a)

Fall	$k_y, \text{exp} [10^{-3} \text{ m}^{-1}]$	$\frac{\Delta k_y, \text{exp}}{k_y, \text{exp}} [\%]$	$k_y, \text{theo} [10^{-3} \text{ m}^{-1}]$	$\frac{\Delta k_y, \text{theo}}{k_y, \text{theo}} [\%]$	σ
a)	33,4(1,7)	5	31,6(2,3)	7,4	1,3
b)	8,7(4)	5	9,7(7)	7,4	0,64

Insgesamt haben wir in diesem Versuch bei allen bis auf die Intensitäten ziemlich positive Ergebnisse bekommen, weshalb wir über die Durchführung relativ zufrieden sein können. Am Auswertung Teil des Versuches war allerdings oft sehr unklar was gewünscht war, weshalb die Auswertung deutlich langsamer und komplizierter erfolgte.

6 Quellen

Wagner, J., Universität Heidelberg (2021). Physikalisches Praktikum PAP 2.2 für Studierende der Physik B.Sc..

Anhang

Versuch Fourieroptik

20. März 2022

7 Anhang

```
[1]: import matplotlib.pyplot as plt
      import matplotlib.mlab as mlab
      %matplotlib inline
      import numpy as np
      from numpy import exp, sqrt, log, pi
      from scipy.optimize import curve_fit
      from scipy.stats import chi2
      from scipy import odr
      from scipy.integrate import quad
      from scipy.signal import find_peaks
      from scipy.signal import argrelextrema, argrelmin, argrelmax

def fehler(name, G, sig_G, G_lit, sig_G_lit):
    print(name)
    print('Relativer Fehler: ', sig_G / G * 100)
    print('Rel. Fehler (Vergleich): ', sig_G_lit / G_lit * 100)
    print('Absoluter Fehler: ', G - G_lit)
#    print('Verhältnis:', G / G_lit)
    print('Sigma-Abweichung: ', np.abs(G - G_lit) / sqrt(sig_G ** 2
                                                       + sig_G_lit ** 2), '\n')

def fehler_small(name, G, sig_G):
```

```

print(name)
print('Relativer Fehler: ', sig_G / G * 100)

def ergebnis(name, G, sig_G, komma, einheit):
    print(name + ' =', np.round(G, komma), '+/-', np.round(sig_G, komma), einheit)

def fitparameter(name, G, sig_G, komma, einheit):
    return name + ' =' + str(np.round(G, komma)) + '$\pm$' + str(np.round(sig_G, komma)) + einheit

```

VIII.1 Quantitative Beobachtungen am Einfachspalt Eichung

```
[2]: # Messdaten
px = np.array([384, 476, 576, 678, 786]) # Lage in Pixels
sig_px = 10 * np.ones(5)
mm = 2 * np.array([2.05, 1.81, 1.60, 1.45, 1.20]) # Öffnung in mm
sig_mm = 0.02 * np.ones(5)
```

Es wird der mittlere Abstand zwischen Minima bestimmt

```
[3]: # Minima
min_ord = np.array([0.5, 1.5, 2.5, 3.5, 4.5]) + 0.5
min_lage = np.array([997, 1096, 1198, 1295, 1394])
Delta_min = np.array([min_lage[i + 1] - min_lage[i] for i in range(4)])
Delta_min_mean = np.mean(Delta_min)
sig_Delta_min_mean = 1 / sqrt(len(Delta_min) - 1) * np.std(Delta_min)

# Maxima
max_ord = np.array([0, 1, 2, 3, 4, 5])
max_lage = np.array([897, 1032, 1142, 1250, 1342, 1436])
Delta_max = np.array([max_lage[i] - max_lage[i + 1] for i in range(4)])

sig_ext = 10
```

```
ergebnis('Delta_min_mean', Delta_min_mean, sig_Delta_min_mean, 0, ↵
        ' [px]')
```

Delta_min_mean = 99.0 +/- 1.0 [px]

[4]: # Fit: Lage gegen Spaltbreite

```
def line(p, x):
    return p[0] * x + p[1]

quad_model = odr.Model(line)
data = odr.RealData(px, mm, sx = sig_px, sy = sig_mm)
out = odr.ODR(data, quad_model, beta0 = [2., 1.]).run()
popt = out.beta
sig_popt = out.sd_beta
m = popt[0]
b = popt[1]
sig_m = sig_popt[0]
sig_b = sig_popt[1]
```

```
ergebnis('Steigung: m', m, sig_m, 4, '[mm px^-1]')
ergebnis('Achsenabschnitt: n', b, sig_b, 2, '[mm]')
```

Steigung: m = -0.0041 +/- 0.0002 [mm px^-1]

Achsenabschnitt: n = 5.63 +/- 0.12 [mm]

[5]: # Plot

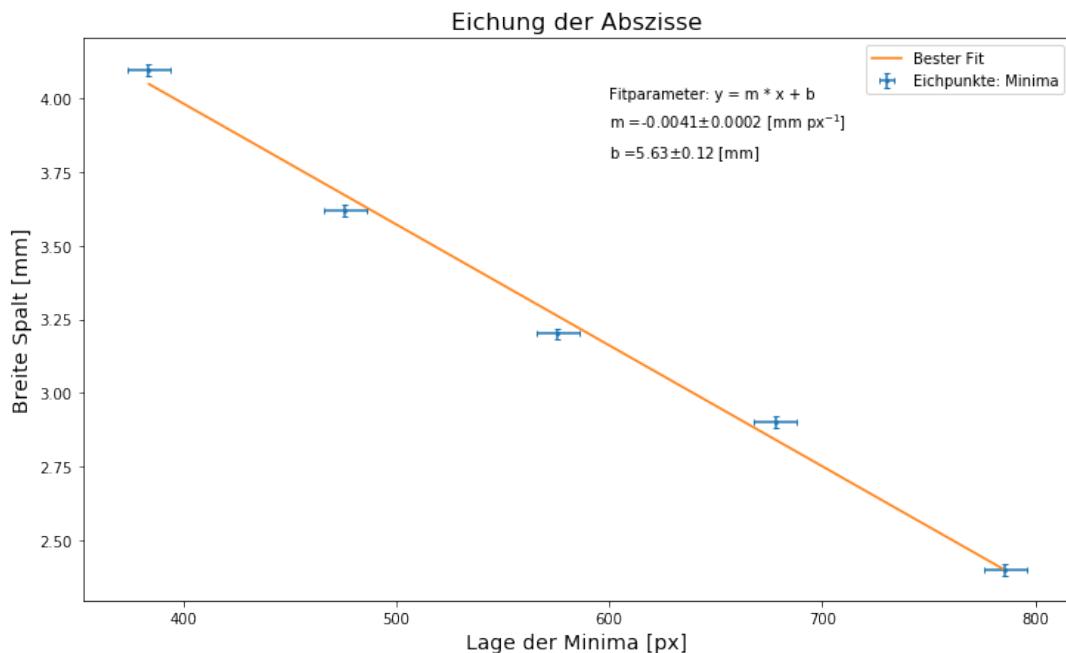
```
plt.figure(figsize = (12, 7))
plt.errorbar(px, mm, xerr = sig_px, yerr = sig_mm, fmt = '.',
             markersize = 4,
             capsize = 2, label = 'Eichpunkte: Minima')
plt.plot(px, line(popt, px), label = 'Bester Fit')
plt.title('Eichung der Abszisse', size = 16)
plt.xlabel('Lage der Minima [px]', size = 14)
plt.ylabel('Breite Spalt [mm]', size = 14)
```

```

plt.text(600, 4, 'Fitparameter:  $y = m * x + b$ ')
plt.text(600, 3.9, fitparameter('m', m, sig_m, 4, ' [mm px$^{-1}]'))
plt.text(600, 3.8, fitparameter('b', b, sig_b, 2, ' [mm]'))
plt.legend(loc = 'best')

plt.savefig('images/233/V233Diagramm1.png')

```



Auswertung der Beugungsstrukturen

Es werden die Lagen der Minima gegen ihre Beugungsordnung aufgetragen

```
[6]: # Fit: Lage gegen Minima
quad_model = odr.Model(line)
data = odr.RealData(min_ord, min_lage, sy = sig_ext)
out = odr.ODR(data, quad_model, beta0 = [2., 1.]).run()
popt_2 = out.beta
sig_popt_2 = out.sd_beta
m_2 = popt_2[0]
b_2 = popt_2[1]
sig_m_2 = sig_popt_2[0]
sig_b_2 = sig_popt_2[1]
```

```
ergebnis('Steigung: m_2', m_2, sig_m_2, 1, '[px / Ordnung]')
ergebnis('Achsenabschnitt: b_2', b_2, sig_b_2, 1, '[px]')
```

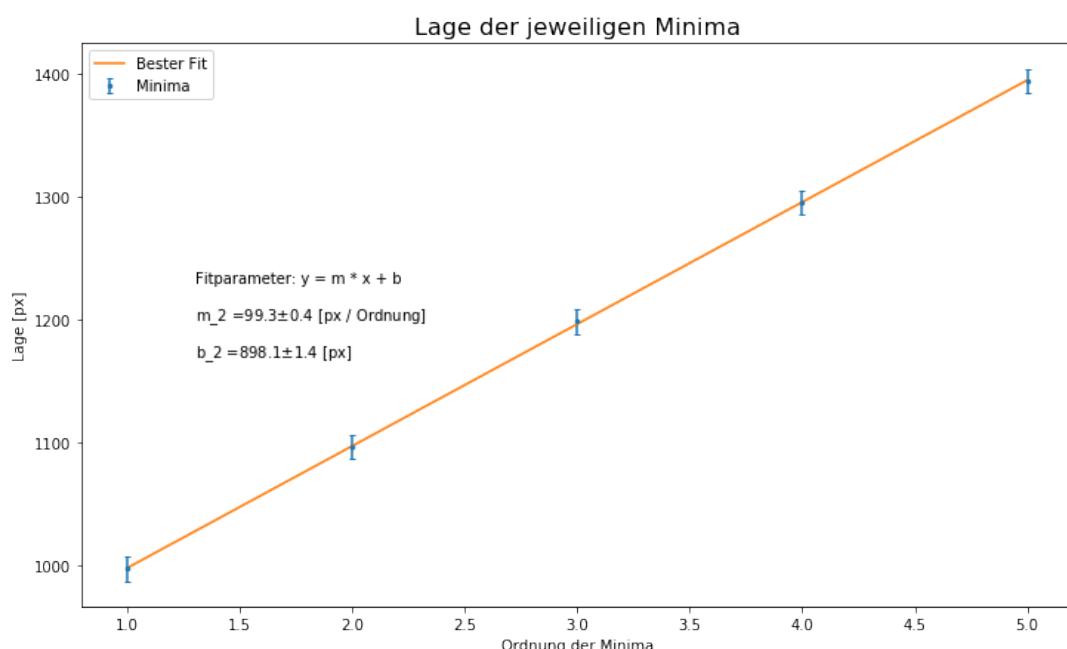
Steigung: $m_2 = 99.3 \pm 0.4 \text{ [px / Ordnung]}$

Achsenabschnitt: $b_2 = 898.1 \pm 1.4 \text{ [px]}$

[7]: # Plot

```
plt.figure(figsize = (12, 7))
plt.errorbar(min_ord, min_lage, yerr = sig_ext, fmt = '.', markersize = 5,
             capsize = 2, label = 'Minima')
plt.plot(min_ord, line(popty_2, min_ord), label = 'Bester Fit')
plt.title('Lage der jeweiligen Minima', size = 16)
plt.xlabel('Ordnung der Minima')
plt.ylabel('Lage [px]')
plt.text(1.3, 1230, 'Fitparameter:  $y = m * x + b$ ')
plt.text(1.3, 1200, fitparameter('m_2', m_2, sig_m_2, 1, '[px / Ordnung']))
plt.text(1.3, 1170, fitparameter('b_2', b_2, sig_b_2, 1, '[px']))
plt.legend()

plt.savefig('images/233/V233Diagramm2.png')
```



```
[8]: # Maxima
n = (max_lage - b_2) / m_2
sig_n = sqrt((sig_ext / b_2) ** 2 + (sig_b_2 / m_2) ** 2
              + ((n * sig_m_2) / m_2) ** 2)

npi = n * pi
sig_npi = sig_n * pi

npi_lit = np.array([(n + 0.5) * pi - 1 / ((n + 0.5) * pi) for n in
                     range(6)])
sig_npi_lit = 0.01 * npi_lit
npi_lit[0] = 0

print('Maxima n:', np.round(n, 2))
print('      +/-', np.round(sig_n, 3))

print('Maxima npi:', np.round(npi, 2))
print('      +/-', np.round(sig_npi, 3))

print('Maxima npi_lit:', np.round(npi_lit, 2))
print('      +/-', np.round(sig_npi_lit, 3))
```

```
Maxima n: [-0.01  1.35  2.46  3.54  4.47  5.42]
      +/- [0.018 0.019 0.02  0.023 0.026 0.029]
Maxima npi: [-0.03  4.24  7.72  11.13 14.04 17.02]
      +/- [0.056 0.058 0.064 0.072 0.081 0.09 ]
Maxima npi_lit: [ 0.     4.5   7.73 10.9  14.07 17.22]
      +/- [0.009 0.045 0.077 0.109 0.141 0.172]
```

```
[9]: # Plot mit Maxima und Minima
plt.figure(figsize = (12, 7))
max_ord = np.arange(0, 7, 1)
plt.errorbar(min_ord, min_lage, yerr = sig_ext, fmt = '.', capsized = 2,
             label = 'Minima')
```

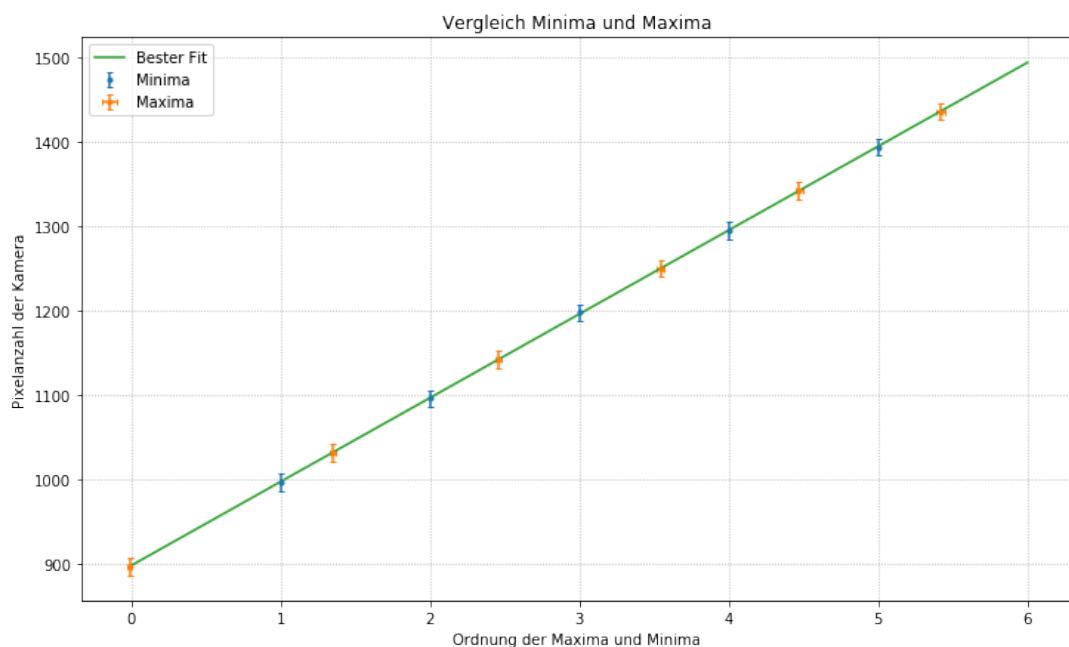
```

plt.errorbar(n, max_lage, xerr = sig_n, yerr = sig_ext, fmt = '.',
             capsize = 2,
             label = 'Maxima')

plt.plot(max_ord, line(popty_2, max_ord), label = 'Bester Fit')
plt.xlabel('Ordnung der Maxima und Minima')
plt.ylabel('Pixelanzahl der Kamera')
plt.title('Vergleich Minima und Maxima')
plt.legend()
plt.grid(linestyle = 'dotted')

plt.savefig('images/233/V233Diagramm3.png')

```



Spaltbreite

[10]: # Werte

```

lambda_k = 635e-9 # m
f = 0.08

```

[11]: # Minima gegen Abstand

```

min_lage_2 = min_lage * m

```

```

sig_ext_2 = min_lage_2 * sqrt((sig_ext / min_lage) ** 2 + (sig_m / m) ↵
                                ** 2)

# Fit

quad_model = odr.Model(line)
data = odr.RealData(min_lage_2, min_ord, sx=sig_ext_2)
out = odr.ODR(data, quad_model, beta0=[2., 1.]).run()

popt_3 = out.beta
sig_popt_3 = out.sd_beta

m_3 = popt_3[0]
b_3 = popt_3[1]
sig_m_3 = sig_popt_3[0]
sig_b_3 = sig_popt_3[1]

```

[12]: # Plot

```

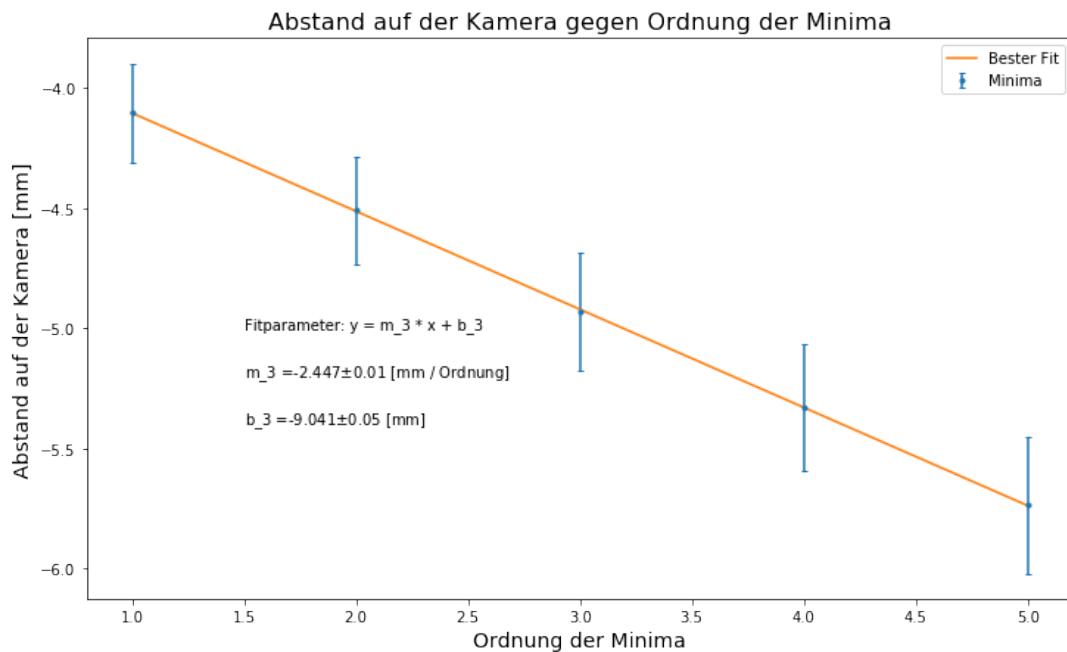
plt.figure(figsize = (12, 7))
plt.errorbar(min_ord, min_lage_2, yerr = sig_ext_2, fmt = '.', capsizedown = 2,
             label = 'Minima')
plt.plot(line(popt_3, min_lage_2), min_lage_2, label = 'Bester Fit')
plt.xlabel('Ordnung der Minima', size = 14)
plt.ylabel('Abstand auf der Kamera [mm]', size = 14)
plt.title('Abstand auf der Kamera gegen Ordnung der Minima', size = 16)
plt.text(1.5, -5, 'Fitparameter:  $y = m_3 \cdot x + b_3$ ')
plt.text(1.5, -5.2, fitparameter('m_3', m_3, sig_m_3, 3, ' [mm / ↵Ordnung] '))
plt.text(1.5, -5.4, fitparameter('b_3', b_3, sig_b_3, 3, ' [mm] '))
plt.legend()

plt.savefig('images/233/V233Diagramm4.png')

ergebnis('Steigung m_3', m_3, sig_m_3, 3, '[mm / Ordnung]')

```

Steigung $m_3 = -2.447 \pm 0.01$ [mm / Ordnung]



```
[13]: # Berechnung der Spaltbreite
```

```
b_S = lambda_k * f / np.abs(m_3)
sig_b_S = lambda_k * f * np.abs(sig_m_3) / m_3 ** 2

ergebnis('Spaltbreite b_S', 1e10 * b_S, 1e10 * sig_b_S, 1, '[10^-7 m]')
```

Spaltbreite $b_S = 207.6 \pm 0.9$ $[10^{-7} \text{ m}]$

Intensitätsverhältnisse

```
[14]: # Messdaten
```

```
#          Max 0   Max 1
I_1 = np.array([0.9672, 0.0867]) # Einheiten ?
#          Max 1   Max 2   Max 3   Max 4   Max 5
I_2 = np.array([0.8746, 0.2990, 0.1496, 0.1000, 0.0804])
sig_I = 0.010
Ug = 0.006
sig_Ug = 0.001

# Angepasst auf Untergrund
I_1_u = I_1 - Ug
```

```

I_2_u = I_2 - Ug
sig_I_u = sqrt(sig_I ** 2 + sig_Ug ** 2)

# Intensitätsverhältnis
# v_i = I_i / I_0

v = I_1_u / I_1_u[0]
sig_v = np.array([v[i] * sqrt( (sig_I_u / I_1_u[i])  

                     + (sig_I_u / I_1_u[0]) ** 2 ) for i in  

                     range(2)])  
  

w = v[1] * I_2_u / I_2_u[0]
sig_w = np.array([w[i] * sqrt( (sig_I_u / I_2_u[i])  

                     + (sig_I_u / I_2_u[0]) ** 2 ) for i in  

                     range(5)])  
  

# Experimentelle Werte
vw = np.append(v, w[1:])
sig_vw = np.append(sig_v, sig_w[1:])

# Theoretische Werte
vw_theo = np.sinc(npi_lit / pi) ** 2
sig_vw_theo = np.abs(2 / pi * (np.cos(npi_lit / pi) / (npi_lit / pi)  

                               - np.sin(npi_lit / pi) / (npi_lit / pi))  

                     ** 2 )
                     * np.sin(npi_lit / pi) / (npi_lit / pi) *  

                     sig_npi_lit)  
  

print('\nExperimentell:')
print('Verhältnis vw:', np.round(vw, 4))
print('      +/-', np.round(sig_vw, 4))

print('\nTheoretisch')
print('Verhältnis vw_theo:', np.round(vw_theo, 4))
print('      +/-', np.round(sig_vw_theo, 4))

```

Experimentell:

```
Verhältnis vw: [1.      0.084  0.0283 0.0139 0.0091 0.0072]
               +/- [0.1028 0.0296 0.0053 0.0037 0.003  0.0026]
```

Theoretisch:

```
Verhältnis vw_theo: [1.      0.0472 0.0165 0.0083 0.005  0.0034]
                     +/- [    nan 0.0077 0.0053 0.0016 0.0001 0.0022]
```

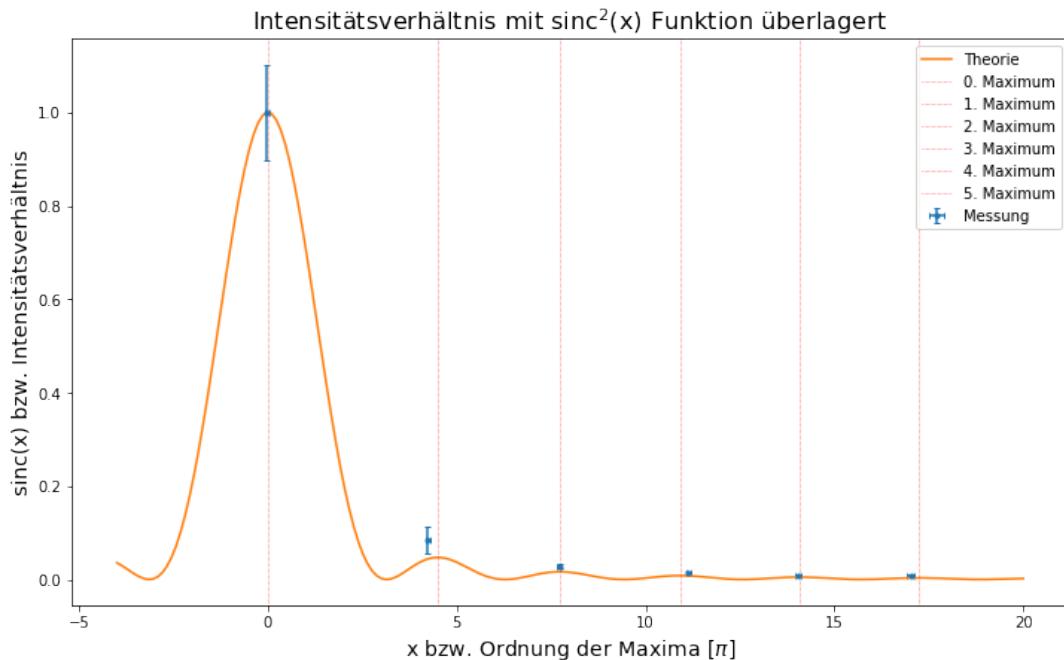
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:34: RuntimeWarning:
divide by zero encountered in true_divide
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:34: RuntimeWarning:
invalid value encountered in true_divide
```

```
[15]: # Plot
x = np.linspace(-4,20,200)

def axvline(x):
    for i in range(len(x)):
        plt.axvline(x[i], linewidth = 0.5, color = 'r', linestyle = 'dotted',
                    label = str(i) + '. Maximum')

plt.figure(figsize = (12, 7))
plt.errorbar(npi, vw, fmt = '. ', yerr = sig_vw, xerr = sig_npi, capsize = 2,
             label='Messung')
plt.plot(x, np.sinc(x / pi) ** 2, label = "Theorie")
plt.title('Intensitätsverhältnis mit sinc$^2$(x) Funktion überlagert', size = 16)
plt.xlabel('x bzw. Ordnung der Maxima [$\pi$]', size = 14)
plt.ylabel('sinc(x) bzw. Intensitätsverhältnis', size = 14)
axvline(npi_lit)
plt.legend(loc = 'upper right')
```

```
plt.savefig('images/233/V233Diagramm5.png')
```



VIII.2 Beugungstruktur des Doppelspaltes

Doppelspalt und Einhüllende

```
[16]: # Messwerte
d = 409 # Px
sig_d = 10
b = 136
sig_b = 10

vbd = d / b
sig_vbd = sqrt( (sig_d / b) ** 2 + (d * sig_b / b ** 2) ** 2 )

ergebnis('Verhältnis vbd', vbd, sig_vbd, 2, '')
```

Verhältnis vbd = 3.01 +/- 0.23

```
[17]: ## Theoretisches Beugungsbild ##
# Beugungsfunktion Spalt
```

```

def beugung_spalt(x):
    return np.sinc(x) ** 2

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

```

Intensitätsverhältnisse

```

[18]: # Messdaten
#           Max 0   Max 1
ID_1 = np.array([0.9445, 0.644])
#           Max 1   Max 2   Max 3   Max 4   Max 5
ID_2 = np.array([0.954, 0.1248, 0.062, 0.066, 0.038])

# Angepasst auf Untegrund
ID_1_u = ID_1 - Ug
ID_2_u = ID_2 - Ug

# Fehler gleich wie beim Einzelspalt

# Intensitätsverhältnisse
vD = ID_1_u / ID_1_u[0]
sig_vD = np.array([vD[i] * sqrt( (sig_I_u / ID_1_u[i]) +
                                  + (sig_I_u / ID_1_u[0]) ** 2 ) for i in
                     range(2)])]

wD = vD[1] * ID_2_u / ID_2_u[0]
sig_wD = np.array([wD[i] * sqrt( (sig_I_u / ID_2_u[i]) +
                                  + (sig_I_u / ID_2_u[0]) ** 2 ) for i in
                     range(5)])]

# Experimentelle Werte
vwD = np.append(vD, wD[1:])

```

```

sig_vwD = np.append(sig_vD, sig_wD[1:])

'''

Die Bestimmung der theoretischen Werten erfolgt durch eine Suche der_
→Maxima

aus dem Datenset des Kurvenverlaufs

'''

a = 2
x_values2 = np.linspace(- (a + 1), a + 1, 400)
y_values2 = beugung_doppelspalt(x_values2) # Wir suchen sie in diesem_
→Interval

maxima, _ = find_peaks(beugung_doppelspalt(x_values2), height = 0.01,_
→prominence = 0.01)
npiD_lit = x_values2[maxima[6:-1]]


# Theoretische Werte
vwD_theo = np.sinc(npiD_lit) ** 2 * np.cos(pi * vbd * npiD_lit) ** 2
sig_vwD_theo = 1 / 200

print('\nExperimentell:')
print('Verhältnis vwD:', np.round(vwD, 4))
print('          +/-', np.round(sig_vwD, 4))

print('\nTheoretisch')
print('Verhältnis vwD_theo:', np.round(vwD_theo, 3), '+/-',_
→sig_vwD_theo)

```

Experimentell:

Verhältnis vwD: [1. 0.6798 0.0852 0.0402 0.043 0.0229]
 +/- [0.104 0.0856 0.0248 0.017 0.0176 0.0129]

Theoretisch:

Verhältnis vwD_theo: [0.995 0.695 0.192 0.043 0.029 0.014] +/- 0.005

Graphische Darstellung

```
[19]: # Ablesen von experimentellem Bild
file = open('data/233/Messung1Doppelspalte.jdx', 'r')

for line in range(20): # Get rid of first lines
    _ = file.readline()

newintensity = np.array([])

'''

Wir entnehmen den Zeilen die einzelnen Eintragen und fügen sie in
ein einzelnes array
'''

for line in file:
    newline = np.array([])

    line = line.strip('\n')
    line = line.split()

    line = line[1:-1]

    for i in line:
        newline = np.append(newline, float(i))

    newintensity = np.append(newintensity, newline)

'''

Wir definieren die Koordinaten neu
'''

newx = np.arange(1792)
newintensity = 1e-6 * newintensity
```

```
# Plot
pixel = newx
intensity = newintensity

x_values1 = pixel
y_values1 = intensity

"""

Kleiner Exkurs: Wir bestimmen die Maxima des Theoretischen_
→Beugungsbilds
"""

maxima_exp, _ = find_peaks(intensity, prominence = 0.01, height = 0.01)
nD_exp = pixel[maxima_exp]

fig = plt.figure(figsize = (12, 7))
plt.title('Beugung am Spalt und Doppelspalt]\n\n', size = 16)
plt.plot(x_values2, beugung_spalt(x_values2), label = 'Einhüllende_'
→Einzelpunkt')
plt.axis('off')

ax = fig.add_subplot(111, label = "1")
ax2 = fig.add_subplot(111, label = "2", frame_on = False)
ax3 = fig.add_subplot(111, label = '3', frame_on = False)

ax.plot(x_values1, y_values1, label = 'Experiment', color =_
→'dodgerblue')
ax.plot(nD_exp, intensity[maxima_exp], 'x', markersize = 12, color =_
→'blue',
       label = 'Maxima')
```

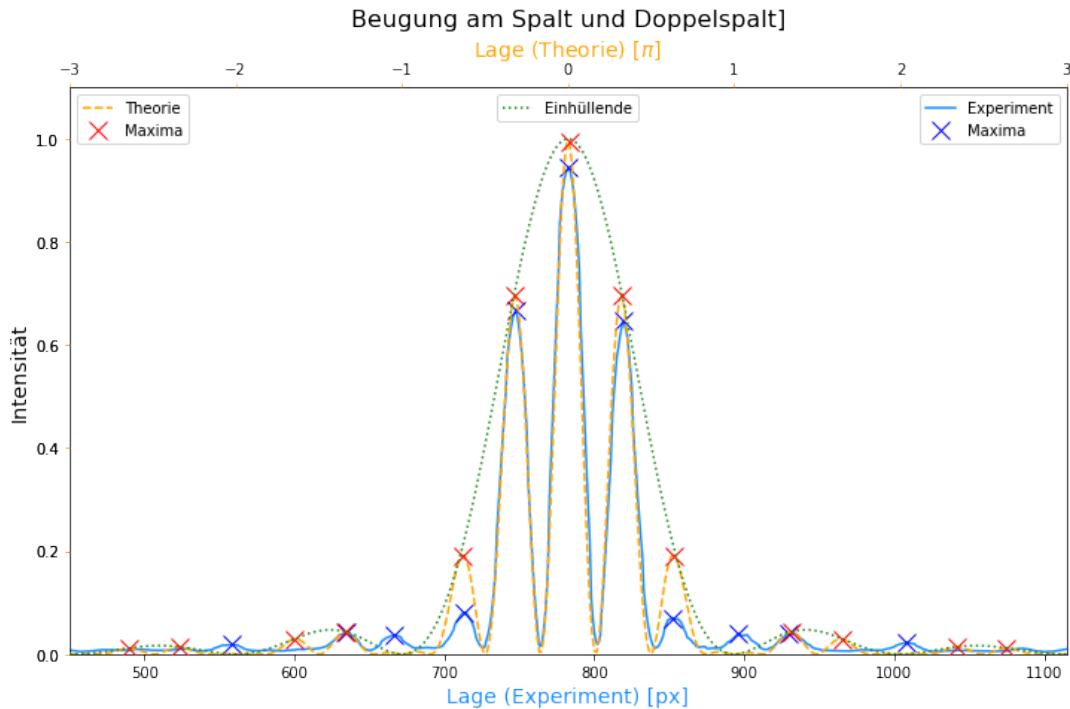
```
ax.set_xlabel('Lage (Experiment) [px]', size = 14, color = 'dodgerblue')
ax.set_ylabel('Intensität', size = 14)
ax.set_ylim([0, 1.1])
ax.set_xlim([450, 1115])
ax.xaxis.tick_bottom()
#ax.tick_params(axis = 'x', color = 'cornflowerblue')
ax.tick_params(axis = 'y', color = 'cornflowerblue')
ax.legend()

ax2.plot(x_values2, y_values2, color = 'orange', label = 'Theorie',
          linestyle = '--')
ax2.plot(x_values2[maxima], beugung_doppelspalt(x_values2[maxima]),
          'x', markersize = 12, color = 'red',
          label = 'Maxima')
ax2.set_ylim([0, 1])
ax2.xaxis.tick_top()
ax2.set_xlabel('Lage (Theorie) [ $\pi$ ]', color = 'orange', size = 14)
ax2.xaxis.set_label_position('top')
ax2.tick_params(axis = 'x', color = 'orange')
ax2.tick_params(axis = 'y', color = 'orange')
ax2.set_xlim((-3,3))
ax2.set_ylim((0, 1.1))
ax2.set_xlim((-3,3))
ax2.legend(loc = 'upper left')

ax3.plot(x_values2, beugung_spalt(x_values2), color = 'forestgreen',
          label = 'Einhüllende', linestyle = 'dotted')
ax3.legend(loc = 'upper center')
ax3.set_xlim((-3, 3))
ax3.set_ylim((0, 1.1))

#ax2.axis('off')
ax3.axis('off')

fig = plt.savefig('images/233/V233Diagramm6.png')
```



VIII.3 Fouriersynthese des Beugungsbildes

Spaltbreite

```
[20]: # Bildgrösse
B = 136 # px
sig_B = sqrt(2 * 5 ** 2)

# Bildweite
b = 0.78 # m
sig_b = 0.04

# Brennweite
f = 0.08 # m

# Gegenstandsgrösse
G = B / ( b / f - 1 )
sig_G = sqrt( (sig_B / ( b / f - 1 )) ** 2
              + (B * sig_b / ((b / f - 1) ** 2 * f)) ** 2 )
```

```

ergebnis('Spaltbreite G', G, sig_G, 2, '[px]')

# Pixel entspricht  $14 * 10^{-6} \text{ m}$ 
G_m = 14 * G
sig_G_m = 14 * sig_G
ergebnis('Spaltbreite G_m', G_m, sig_G_m, 0, '[10^{-6} \text{ m}]')

```

Spaltbreite G = 15.54 +/- 1.2 [px]
 Spaltbreite G_m = 218.0 +/- 17.0 [10^-6 m]

Intensitätsverhältnisse: Experimentell

[21]: #Als nächstes werden die Kurvenverläufe der Intensitäten aufgetragen.
 #Nicht vorhanden ist der Verlauf zu $n = 3$, daher werden die
 ↵Messwerten
 #dem Messprotokoll entnommen.

[22]: ## Befehle
 # Abstände der Wiggles
 def abstand(maxima):
 pos = np.array([])
 if len(maxima) < 2:
 True
 else:
 for i in range(len(maxima) - 1):
 pos = np.append(pos, maxima[i+1] - maxima[i])
 global dist, sig_dist
 dist = np.mean(pos)
 sig_dist = 1 / sqrt(len(maxima)) * np.std(pos)

 print('Abstand zwischen Maxima: ')
 print(pos)
 print('\nMittlerer Abstand')
 print(np.round(dist, 3), '+/-', np.round(sig_dist, 3), '[px]')

 return dist, sig_dist

```

def abstand_min(minima):
    pos = np.array([])
    if len(minima) < 2:
        True
    else:
        for i in range(len(minima) - 1):
            pos = np.append(pos, minima[i+1] - minima[i])
    global dist_min, sig_dist_min
    dist_min = np.mean(pos)
    sig_dist_min = 1 / sqrt(len(minima)) * np.std(pos)

    print('Abstand zwischen Minima: ')
    print(pos)
    print('\nMittlerer Abstand')
    print(np.round(dist_min, 3), '+/-', np.round(sig_dist_min, 3), ↴
→ '[px]')

    return dist_min, sig_dist_min


def intensitaet(intensity, maxima):
    print('')
    I_list = np.array([])
    for i in range(len(maxima)):
        print('Intensität der ' + str(i + 1) + '. Maxima: ',
              np.round(intensity[maxima[i]], 4))
        I_list = np.append(I_list, intensity[maxima[i]])
    return I_list


def intensitaet_min(intensity, minima):
    print('')
    I_list = np.array([])
    for i in range(len(minima)):
        print('Intensität der ' + str(i + 1) + '. Minima: ',
              np.round(intensity[minima[i]], 4))
        I_list = np.append(I_list, intensity[minima[i]])

```

```
    return I_list
```

```
[23]: # n = 0
# Ablesen von experimentellem Bild
file0 = open('data/233/Messung3(b).jdx', 'r')

for line in range(20): # Get rid of first lines
    _ = file0.readline()

newintensity0 = np.array([])

'''

Wir entnehmen den Zeilen die einzelnen Einträge und fügen sie in
ein einzelnes array
'''

for line in file0:
    newline = np.array([])

    line = line.strip('\n')
    line = line.split()

    line = line[1:-1]

    for i in line:
        newline = np.append(newline, float(i))

    newintensity0 = np.append(newintensity0, newline)

'''

Wir definieren die Koordinaten neu
'''

pixel0 = np.arange(1792)
```

```

intensity0 = 1e-6 * newintensity0

I_0_max = np.max(intensity0)

intensity0 = intensity0 / I_0_max

maxima_exp0, _ = find_peaks(intensity0, height = 0.6, distance = 50)
#minima_exp1, _ = find_peaks(intensity1, height = (0.5, 0.61),_
→distance = 10)

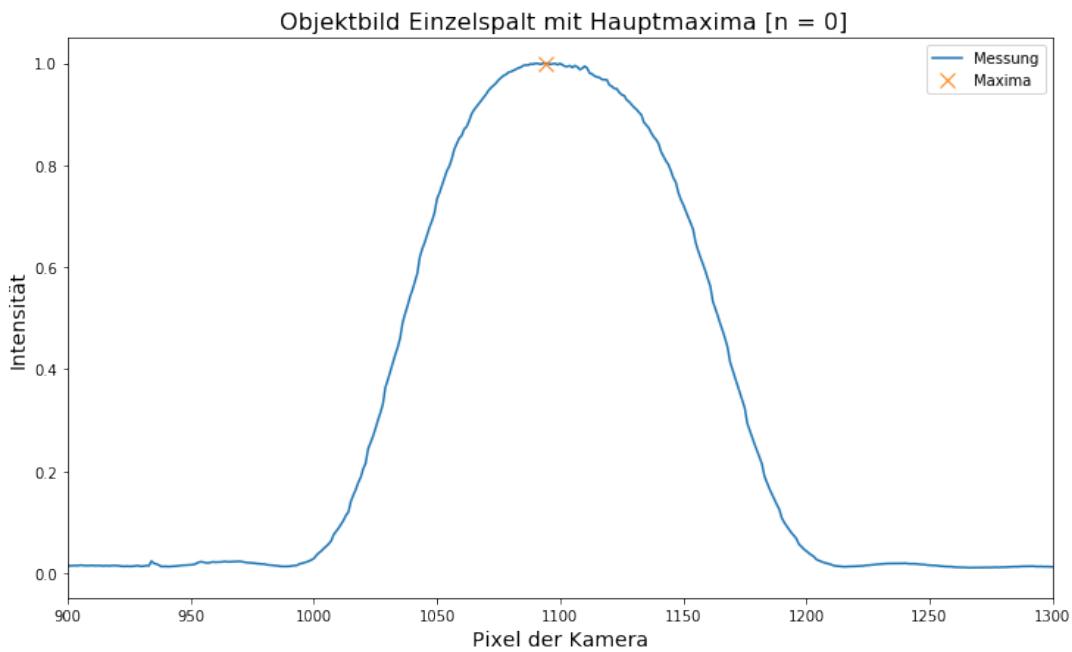
plt.figure(figsize = (12, 7))
plt.plot(pixel0[1:], intensity0[1:], label = 'Messung')
plt.title(' Objektbild Einzelspalt mit Hauptmaxima [n = 0] ', size = 16)
plt.xlabel('Pixel der Kamera', size = 14)
plt.ylabel('Intensität', size = 14)
plt.plot(maxima_exp0, intensity0[maxima_exp0], 'x', markersize = 10,_
→label = 'Maxima')
# plt.plot(minima_exp1, intensity1[minima_exp1], 'x', markersize = 10,_
→label = 'Minima')
plt.xlim(900, 1300)
plt.legend()

plt.savefig('images/233/V233DiagrammIex0.png')

# Abstand zwischen Maxima
#dist, sig_dist = abstand(maxima_exp0)
# Intensität der Maxima, aufsteigend von links nach rechts
I_0_ex = intensitaet(intensity0, maxima_exp0)

```

Intensität der 1. Maxima: 1.0



```
[24]: # n = 1
file1 = open('data/233/3n1.csv', 'r')

a1 = np.empty(2, float)

for line in file1:

    line = line.strip('\n')
    line = line.split(',')

    foo = float(line[0])
    bar = float(line[1])

    a1 = np.vstack([a1, [foo, bar]])

pixel1 = a1[:, 0]
intensity1 = a1[:, 1]

intensity1 = intensity1 / I_0_max
```

```

maxima_exp1, _ = find_peaks(intensity1, height = 0.6, distance = 50)
minima_exp1, _ = find_peaks(intensity1, height = (0.8, 0.85), distance_
    ↪= 30)

plt.figure(figsize = (12, 7))
plt.plot(pixel1[1:], intensity1[1:], label = 'Messung')
plt.title(' Objektbild Einzelspalt mit Hauptmaxima [n = 1]', size = 16)
plt.xlabel('Pixel der Kamera', size = 14)
plt.ylabel('Intensität', size = 14)
plt.plot(maxima_exp1, intensity1[maxima_exp1], 'x', markersize = 10,_
    ↪label = 'Maxima')
plt.plot(minima_exp1, intensity1[minima_exp1], 'x', markersize = 10,_
    ↪label = 'Minima')
plt.xlim(1100, 1400)
plt.legend()

plt.savefig('images/233/V233DiagrammIex1.png')

# Abstand zwischen Maxima
dist, sig_dist = abstand(maxima_exp1)
# Intensität der Maxima, aufsteigend von links nach rechts
I_1_ex_max = intensitaet(intensity1, maxima_exp1)

I_1_ex_min = intensitaet_min(intensity1, minima_exp1)

I_1_ex_gr = np.array([I_1_ex_max[0], I_1_ex_min[0], I_1_ex_max[1]])

```

Abstand zwischen Maxima:

[95.]

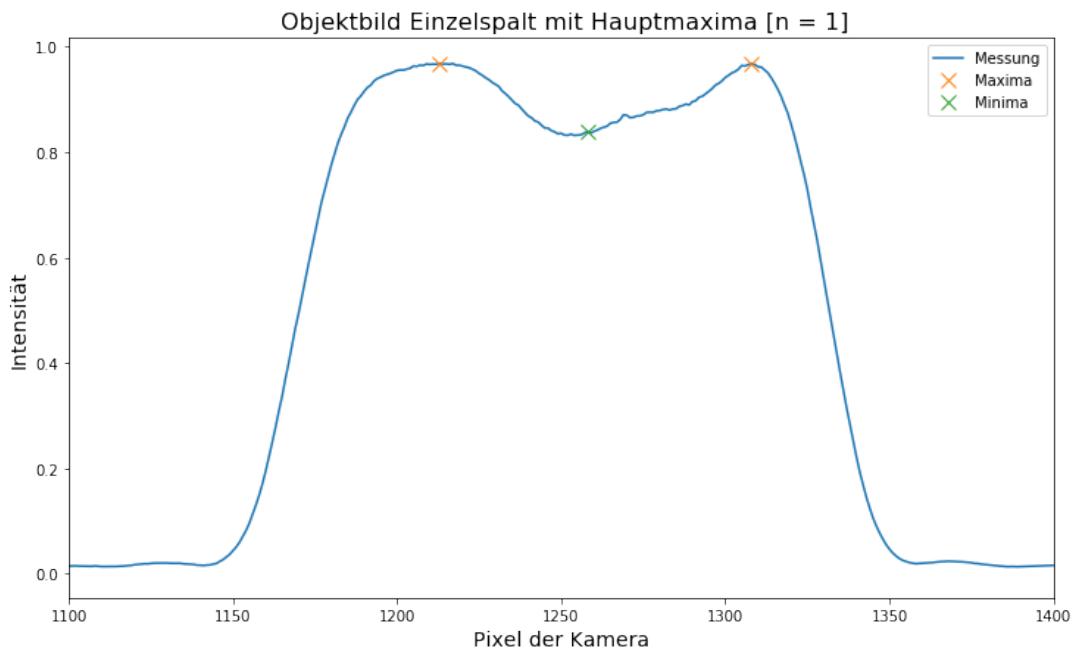
Mittlerer Abstand

95.0 +/- 0.0 [px]

Intensität der 1. Maxima: 0.9688

Intensität der 2. Maxima: 0.9685

Intensität der 1. Minima: 0.839



```
[25]: # Bild bei zwei Maxima
file2 = open('data/233/3n2.csv', 'r')

a2 = np.empty(2, float)

for line in file2:

    line = line.strip('\n')
    line = line.split(',')

    foo = float(line[0])
    bar = float(line[1])

    a2 = np.vstack([a2, [foo, bar]])

pixel2 = a2[:, 0]
intensity2 = a2[:, 1]
```

```

intensity2 = intensity2 / I_0_max

maxima_exp2, _ = find_peaks(intensity2, height = 0.6, distance = 30)
minima_exp2, _ = find_peaks(intensity2, height = (0.8, 0.88))

plt.figure(figsize = (12, 7))
plt.plot(pixel2[1:], intensity2[1:], label = 'Messung')
plt.title('Objektbild Einzelspalt mit Hauptmaxima [n = 2]', size = 16)
plt.xlabel('Pixel der Kamera', size = 14)
plt.ylabel('Intensität', size = 14)
plt.plot(maxima_exp2, intensity2[maxima_exp2], 'x', markersize = 10, ↴
         label = 'Maxima')
plt.plot(minima_exp2, intensity2[minima_exp2], 'x', markersize = 10, ↴
         label = 'Minima',
         color = 'green')
plt.plot(1289, intensity2[1289], 'x', markersize = 10, color = 'green')
plt.xlim(1100, 1400)
plt.legend()

plt.savefig('images/233/V233DiagrammIex2.png')

# Abstand zwischen Maxima
dist, sig_dist = abstand(maxima_exp2)
# Intensität der Maxima, aufsteigend von links nach rechts
I_2_ex_max = intensitaet(intensity2, maxima_exp2)

I_2_ex_min = intensitaet_min(intensity2, np.append(minima_exp2, 1289))

I_2_ex_gr = np.array([I_2_ex_max[0], I_2_ex_min[0], I_2_ex_max[1],
                      I_2_ex_min[1], I_2_ex_max[2]])

```

Abstand zwischen Maxima:

[50. 42.]

Mittlerer Abstand

46.0 +/- 2.309 [px]

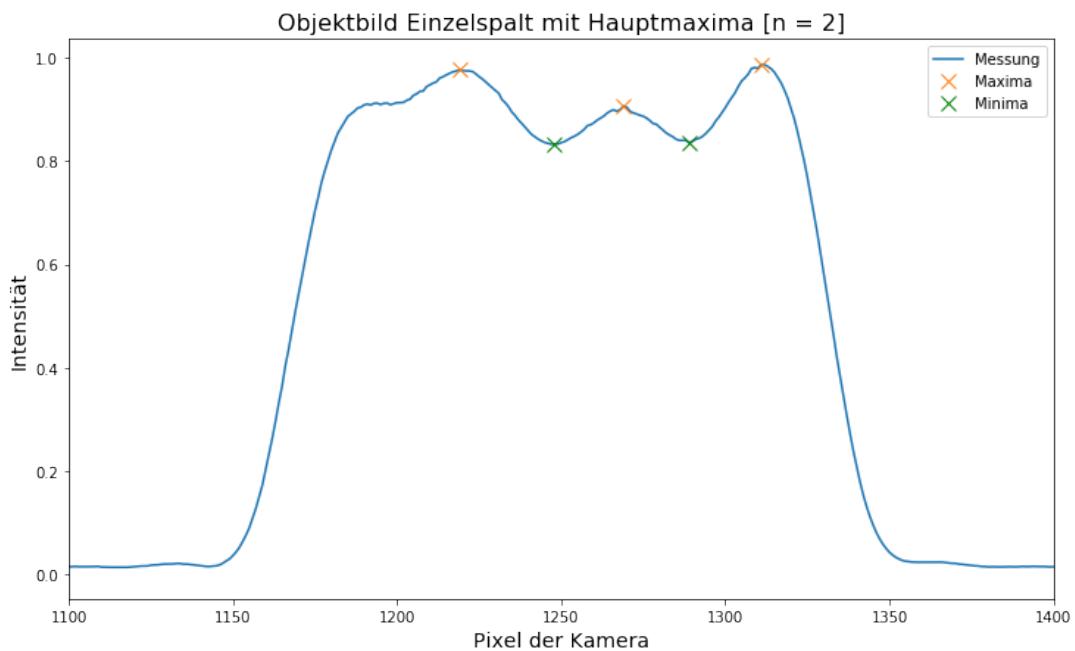
Intensität der 1. Maxima: 0.9762

Intensität der 2. Maxima: 0.906

Intensität der 3. Maxima: 0.9872

Intensität der 1. Minima: 0.8328

Intensität der 2. Minima: 0.8351



```
[26]: # Bild bei 13 Maxima
```

```
file13 = open('data/233/3n15.csv', 'r')

a13 = np.empty(2, float)

for line in file13:

    line = line.strip('\n')
    line = line.split(',')

    foo = float(line[0])
    bar = float(line[1])
```

```
a13 = np.vstack([a13, [foo, bar]])\n\npixel13 = a13[:, 0]\nintensity13 = a13[:, 1]\n\nintensity13 = intensity13 / I_0_max\n\nmaxima_exp13, _ = find_peaks(intensity13, height = 0.6, distance = 7)\nminima_exp13 = argrelextrema(intensity13, np.less)\nminima_exp13 = np.array(minima_exp13)[0]\n\na = 374\nb = 386\n\nplt.figure(figsize = (12, 7))\nplt.plot(pixel13[1:], intensity13[1:], label = 'Messung')\nplt.title(' Objektbild Einzelspalt mit Hauptmaxima [n = 13] ', size = 16)\nplt.xlabel('Pixel der Kamera', size = 14)\nplt.ylabel('Intensität', size = 14)\nplt.plot(maxima_exp13, intensity13[maxima_exp13], 'x', markersize = 10,\n         label = 'Maxima')\nplt.plot(minima_exp13[a:b], intensity13[minima_exp13[a:b]], 'x',\n         markersize = 10,\n         color = 'green')\nplt.plot(minima_exp13[373], intensity13[minima_exp13[373]], 'x',\n         markersize = 10,\n         color = 'green', label = 'Minima')\nplt.xlim(1150, 1400)\nplt.legend()\n\nplt.savefig('images/233/V233DiagrammIex13.png')\n\n# Abstand zwischen Maxima\ndist_max13, sig_dist_max13 = abstand(maxima_exp13)\nprint('')
```

```
dist_min13, sig_dist_min13 = abstand_min(np.append(minima_exp13[373],  
minima_exp13[a:b]))  
# Intensität der Maxima, aufsteigend von links nach rechts  
I_13_ex = intensitaet(intensity13, maxima_exp13)
```

Abstand zwischen Maxima:

[20. 13. 11. 8. 8. 9. 8. 7. 8. 8. 12. 17.]

Mittlerer Abstand

10.538 +/- 1.03 [px]

Abstand zwischen Minima:

[15. 13. 9. 8. 9. 8. 8. 7. 8. 8. 9. 14.]

Mittlerer Abstand

9.667 +/- 0.719 [px]

Intensität der 1. Maxima: 1.1044

Intensität der 2. Maxima: 1.0481

Intensität der 3. Maxima: 1.0125

Intensität der 4. Maxima: 0.9869

Intensität der 5. Maxima: 0.9528

Intensität der 6. Maxima: 0.9138

Intensität der 7. Maxima: 0.8875

Intensität der 8. Maxima: 0.8699

Intensität der 9. Maxima: 0.8798

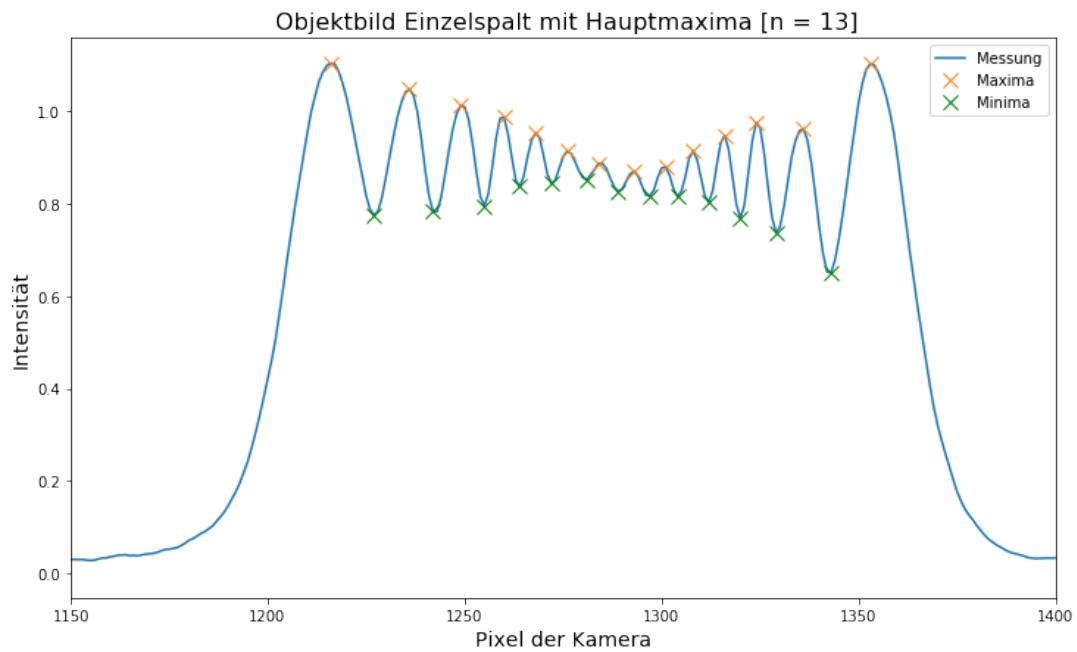
Intensität der 10. Maxima: 0.9143

Intensität der 11. Maxima: 0.948

Intensität der 12. Maxima: 0.9758

Intensität der 13. Maxima: 0.9619

Intensität der 14. Maxima: 1.1031



```
[27]: # Intensite der Maxima und Minima n.-ter Ordnung, wie abgelesen im
      ↵Messprotokoll
IF_0 = 0.726
IF_1 = np.array([0.704, 0.606, 0.692])
IF_2 = np.array([0.710, 0.609, 0.653, 0.610, 0.717])
IF_3 = np.array([0.688, 0.440, 0.672, 0.632, 0.654, 0.570, 0.737])

# Fehler und Untergrund wie bei letzten Messungen: sig_I, Ug, sig_Ug, ↵
      ↵sig_I_u
#IF_u angepasst auf Ug
IF_0_u = IF_0 - Ug
IF_1_u = IF_1 - Ug
IF_2_u = IF_2 - Ug
IF_3_u = IF_3 - Ug

# Verhaltnis zu IF_0
IF_10 = IF_1_u / IF_0_u
IF_20 = IF_2_u / IF_0_u
IF_30 = IF_3_u / IF_0_u
```

```

sig_IF_10 = IF_10 * sqrt((sig_I_u / IF_1_u) ** 2 + (sig_I_u / IF_0_u)_
    ↵** 2)
sig_IF_20 = IF_20 * sqrt((sig_I_u / IF_2_u) ** 2 + (sig_I_u / IF_0_u)_
    ↵** 2)
sig_IF_30 = IF_30 * sqrt((sig_I_u / IF_3_u) ** 2 + (sig_I_u / IF_0_u)_
    ↵** 2)

print('Verhältnis zu IF_0:')
print('IF_10 =', np.round(IF_10, 3))
print('    +/-', np.round(sig_IF_10, 3))
print('\nIF_20 =', np.round(IF_20, 3))
print('    +/-', np.round(sig_IF_20, 3))
print('\nIF_30 =', np.round(IF_30, 3))
print('    +/-', np.round(sig_IF_30, 3))

```

Verhältnis zu IF_0:

IF_10 = [0.969 0.833 0.953]

+/- [0.019 0.018 0.019]

IF_20 = [0.978 0.838 0.899 0.839 0.987]

+/- [0.02 0.018 0.019 0.018 0.02]

IF_30 = [0.947 0.603 0.925 0.869 0.9 0.783 1.015]

+/- [0.019 0.016 0.019 0.018 0.019 0.018 0.02]

Intensitätsverhältnisse: Theoretisch

```
[28]: # Spaltfunktion
def spalt(k,y):
    return d1 / pi * np.sin(k * d1 / 2) / (k * d1 / 2) * np.cos(y * k)

d1 = 1 # Spaltbreite
#n = 1 # Integrationsbereich
p = 1.38976 # maximale Intensität bei 0. Beugungsordnung
Y = np.linspace(-1,1, 400) * d1
```

```
# f modifiziert
def f_mod(Y, n):
    f_modifiziert = []
    for i in range(len(Y)):
        y = Y[i]
        result, error = quad(spalt, 0, 2 * pi * n / d1, args = (y,))
        f_modifiziert.append((result ** 2) / p)
    return f_modifiziert
```

[29]: # Plot 0. Ordnung

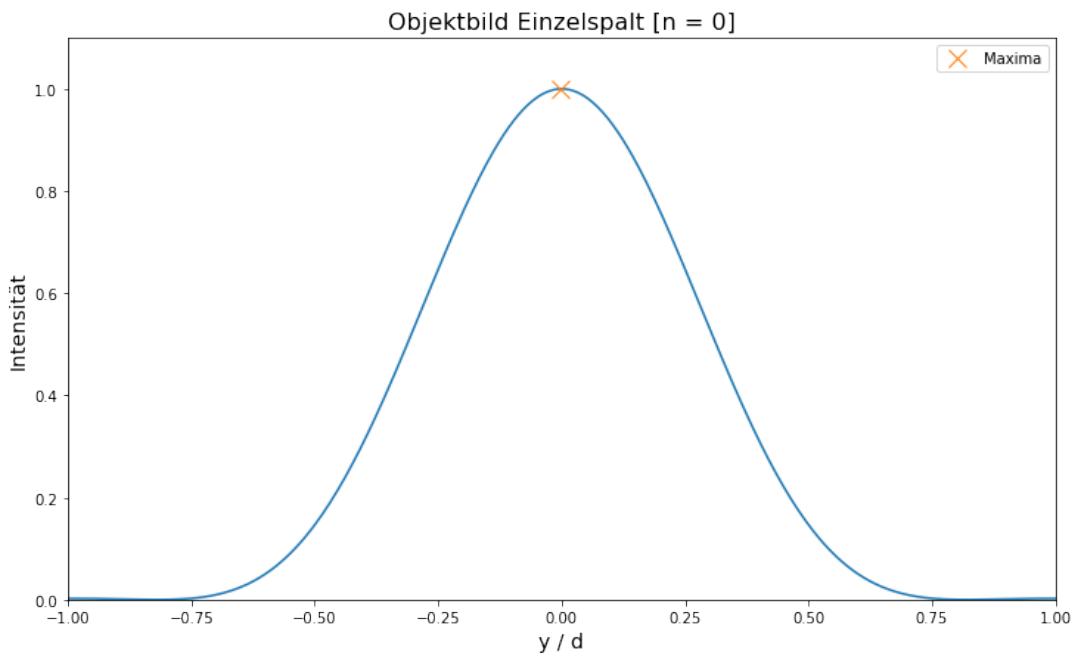
```
maxima_theo0, _ = find_peaks(f_mod(Y, 1), height = 0.5)

plt.figure(figsize = (12, 7))
plt.plot(Y, f_mod(Y, 1))
plt.plot(Y[maxima_theo0], f_mod(Y[maxima_theo0], 1), 'x', markersize = 12, label = 'Maxima')
plt.xlabel('y / d', size = 14)
plt.ylabel('Intensität', size = 14)
plt.title('Objektbild Einzelspalt [n = 0]', size = 16)
plt.xlim((-1, 1))
plt.ylim((0, 1.1))
plt.legend(loc = 'best')

plt.savefig('images/233/V233DiagrammItheo0.png')

# Intensität der Maxima, aufsteigend von links nach rechts
I_0_theo = intensitaet(f_mod(Y, 1), maxima_theo0)
```

Intensität der 1. Maxima: 1.0001



[30]: # Plot 1. Ordnung

```
maxima_theo1, _ = find_peaks(f_mod(Y, 2), height = 0.5)
minima_theo1 = np.array([199])

plt.figure(figsize = (12, 7))
plt.plot(Y, f_mod(Y, 2))
plt.xlabel('y / d', size = 14)
plt.plot(Y[maxima_theo1], f_mod(Y[maxima_theo1], 2), 'x', markersize = 12, label = 'Maxima')
plt.plot(Y[minima_theo1], f_mod(Y[minima_theo1], 2), 'x', markersize = 12, label = 'Minima')
plt.ylabel('Intensität', size = 14)
plt.title('Objektbild Einzelpalt [n = 1]', size = 16)
plt.xlim((-1, 1))
plt.ylim((0, 1.1))
plt.legend(loc = 'best')

plt.savefig('images/233/V233DiagrammItheo1.png')
```

```

# Abstand zwischen Maxima
dist, sig_dist = abstand(Y[maxima_theo1])

# Intensität der Maxima, aufsteigend von links nach rechts
I_1_theo_max = intensitaet(f_mod(Y, 2), maxima_theo1)

I_1_theo_min = intensitaet_min(f_mod(Y, 2), np.array([199]))

I_1_theo = np.array([I_1_theo_max[0], I_1_theo_min[0], I_1_theo_max[1]])

```

Abstand zwischen Maxima:

[0.4962406]

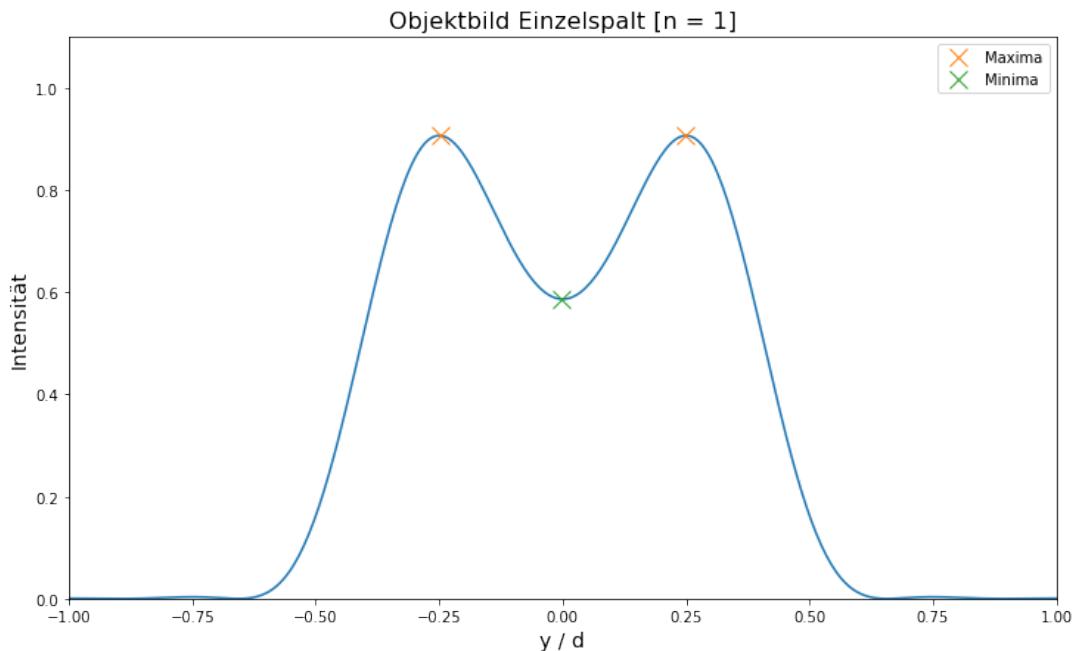
Mittlerer Abstand

0.496 +/- 0.0 [px]

Intensität der 1. Maxima: 0.9067

Intensität der 2. Maxima: 0.9067

Intensität der 1. Minima: 0.5866



```
[31]: # Plot 2. Ordnung

maxima_theo2, _ = find_peaks(f_mod(Y, 3), height = 0.5)
#min_ind_2, _ = find_peaks(f_mod(Y[argrelmin(np.array(f_mod(Y, 3)))]),_
#                           3),
#                           height = 0.4)
min_ind_2, _ = find_peaks(f_mod(Y[argrelmin(np.array(f_mod(Y, 3)))], 3),
                           height = 0.4)
minima_theo2 = np.array([argrelmin(np.array(f_mod(Y, 3)))] [0] [min_ind_2] [0])

plt.figure(figsize = (12, 7))
plt.plot(Y, f_mod(Y, 3))
plt.xlabel('y / d', size = 14)
plt.plot(Y[maxima_theo2], f_mod(Y[maxima_theo2], 3), 'x', markersize =_
          12,
          label = 'Maxima')
plt.plot(Y[minima_theo2], f_mod(Y[minima_theo2], 3), 'x', markersize =_
          12,
          label = 'Minima', color = 'green')
plt.plot(Y[[232]], f_mod(Y[[232]], 3), 'x', markersize = 12,
          color = 'green')
plt.ylabel('Intensität', size = 14)
plt.title('Objektbild Einzelspalt [n = 2]', size = 16)
plt.xlim((-1, 1))
plt.ylim((0, 1.1))
plt.legend(loc = 'best')

plt.savefig('images/233/V233DiagrammItheo2.png')

# Abstand zwischen Maxima
dist, sig_dist = abstand(Y[maxima_theo2])
# Intensität der Maxima, aufsteigend von links nach rechts
I_2_max_theo = intensitaet(f_mod(Y, 3), maxima_theo2)
```

```
I_2_min_theo = intensitaet_min(f_mod(Y, 3), np.append(minima_theo2, np.array([232])))
```

```
I_2_theo = np.array([I_2_max_theo[0], I_2_min_theo[0], I_2_max_theo[1],
                     I_2_min_theo[0], I_2_max_theo[2]])
```

Abstand zwischen Maxima:

[0.33082707 0.3358396]

Mittlerer Abstand

0.333 +/- 0.001 [px]

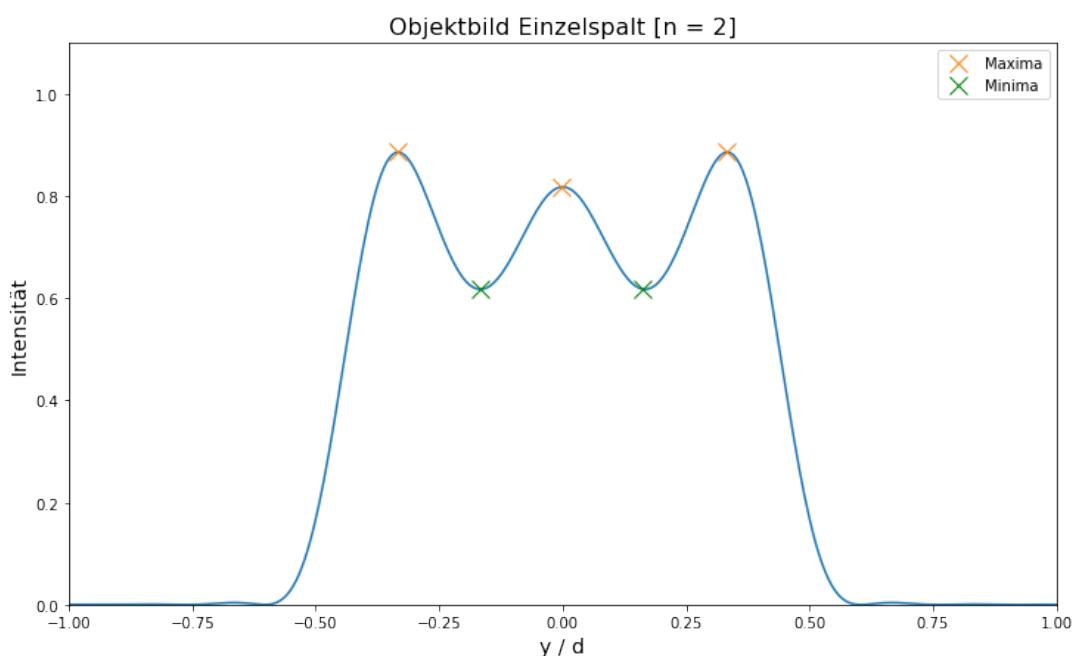
Intensität der 1. Maxima: 0.8859

Intensität der 2. Maxima: 0.8178

Intensität der 3. Maxima: 0.8859

Intensität der 1. Minima: 0.6175

Intensität der 2. Minima: 0.6178



[32]: # Plot 3. Ordnung

```
maxima_theo3, _ = find_peaks(f_mod(Y, 4), height = 0.5)
```

```

min_ind_3, _ = find_peaks(f_mod(Y[argrelmin(np.array(f_mod(Y, 4)))]), 4,
                           height = 0.4)
minima_theo3 = np.append(np.array([argrelmin(np.array(f_mod(Y,
                           ↪4)))[0][min_ind_3]]), np.array([199, 250]))


plt.figure(figsize = (12, 7))
plt.plot(Y, f_mod(Y, 4))
plt.xlabel('y / d', size = 14)
plt.plot(Y[maxima_theo3], f_mod(Y[maxima_theo3], 4), 'x', markersize = ↪
          ↪12,
          label = 'Maxima')
plt.plot(Y[minima_theo3], f_mod(Y[minima_theo3], 4), 'x', markersize = ↪
          ↪12,
          label = 'Minima')
plt.ylabel('Intensität', size = 14)
plt.title('Objektbild Einzelpunkt [n = 4]', size = 16)
plt.ylim((0, 1.1))
plt.legend(loc = 'best')

plt.savefig('images/233/V233DiagrammItheo3.png')

# Abstand zwischen Maxima
dist, sig_dist = abstand(Y[maxima_theo3])
# Intensität der Maxima, aufsteigend von links nach rechts
I_3_max_theo = intensitaet(f_mod(Y, 4), maxima_theo3)

I_3_min_theo = intensitaet_min(f_mod(Y, 4), minima_theo3)

I_3_theo = np.array([I_3_max_theo[0], I_3_min_theo[0], I_3_max_theo[1],
                     I_3_min_theo[1], I_3_max_theo[2], I_3_min_theo[2],
                     I_3_max_theo[3]])

```

Abstand zwischen Maxima:

[0.25062657 0.24561404 0.25062657]

Mittlerer Abstand

0.249 ± 0.001 [px]

Intensität der 1. Maxima: 0.8767

Intensität der 2. Maxima: 0.798

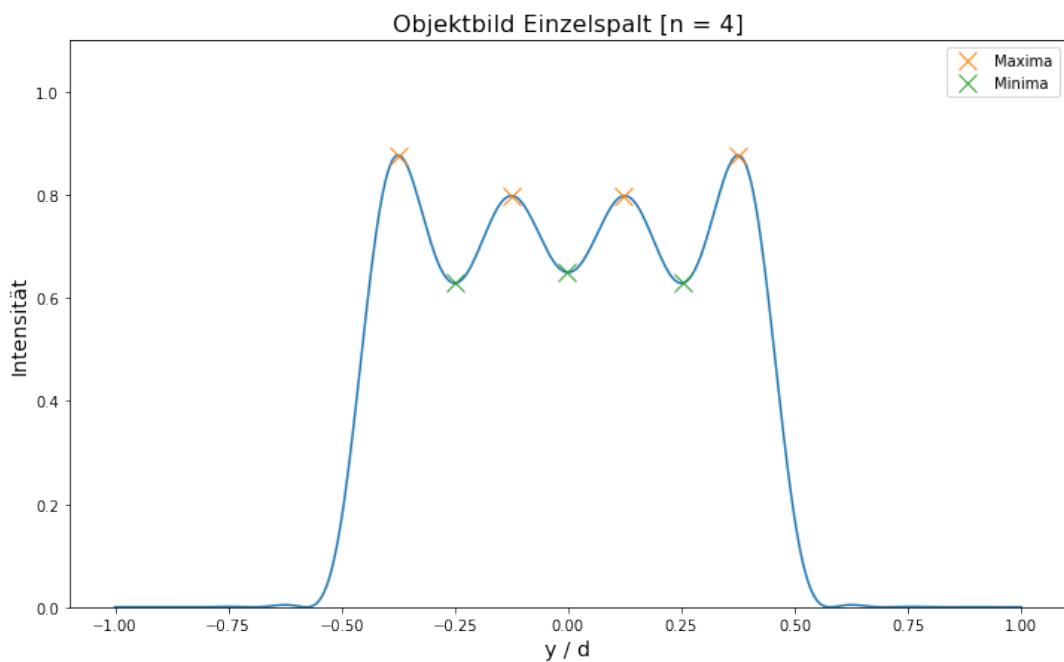
Intensität der 3. Maxima: 0.798

Intensität der 4. Maxima: 0.8767

Intensität der 1. Minima: 0.6286

Intensität der 2. Minima: 0.6494

Intensität der 3. Minima: 0.6288



[33]: # Plot $n = 13$

```
maxima_theo13, _ = find_peaks(f_mod(Y, 14), height = 0.5)
minima_theo13 = np.sort(np.append(argrelmin(np.array(f_mod(Y, ↵14)))[0][14:26], 200))

plt.figure(figsize = (12, 7))
plt.plot(Y, f_mod(Y, 14))
plt.xlabel('y / d', size = 14)
```

```

plt.plot(Y[maxima_theo13], f_mod(Y[maxima_theo13], 14), 'x', markersize=_
         ↵= 12,
         label = 'Maxima')

plt.plot(Y[minima_theo13], f_mod(Y[minima_theo13], 14), 'x', markersize=_
         ↵= 12,
         label = 'Minima')

plt.ylabel('Intensität', size = 14)
plt.title('Objektbild Einzelpalt [n = 13]',size = 16)
plt.xlim((-1, 1))
plt.ylim((0,1.1))
plt.legend(loc = 'best')

plt.savefig('images/233/V233DiagrammItheo13.png')

norm_2_px = 148
sig_norm_2_px = 6

# Abstand zwischen Maxima
dist, sig_dist = abstand(Y[maxima_theo13])
good_dist = dist * norm_2_px
sig_good_dist = good_dist * sqrt( (sig_norm_2_px / norm_2_px) ** 2
                                   + (sig_dist / dist) ** 2 )

print('')
ergebnis('Richtiger Abstand Maxima: good_dist', good_dist,_
         ↵sig_good_dist, 2,
         '[px]\n')

# Abstand zwischen Minima
dist_min, sig_dist_min = abstand_min(Y[minima_theo13])
good_dist_min = dist * norm_2_px
sig_good_dist_min = good_dist_min * sqrt( (sig_norm_2_px / norm_2_px) _ 
         ↵** 2 + (sig_dist_min / dist_min) ** 2 )

```

```
print('')

ergebnis('Richtiger Abstand Minima: good_dist_min', good_dist_min,
          sig_good_dist_min, 2, '[px]')

# Intensität der Maxima, aufsteigend von links nach rechts
I_13_theo = intensitaet(f_mod(Y, 14), maxima_theo13)
```

Abstand zwischen Maxima:

```
[0.07017544 0.07017544 0.07518797 0.07017544 0.07017544 0.07017544
 0.07518797 0.07017544 0.07017544 0.07017544 0.07518797 0.07017544
 0.07017544]
```

Mittlerer Abstand

```
0.071 +/- 0.001 [px]
```

Richtiger Abstand Maxima: good_dist = 10.56 +/- 0.44 [px]

Abstand zwischen Minima:

```
[0.07017544 0.07518797 0.07017544 0.07017544 0.07017544 0.07518797
 0.07017544 0.07017544 0.07017544 0.07017544 0.07518797 0.07017544]
```

Mittlerer Abstand

```
0.071 +/- 0.001 [px]
```

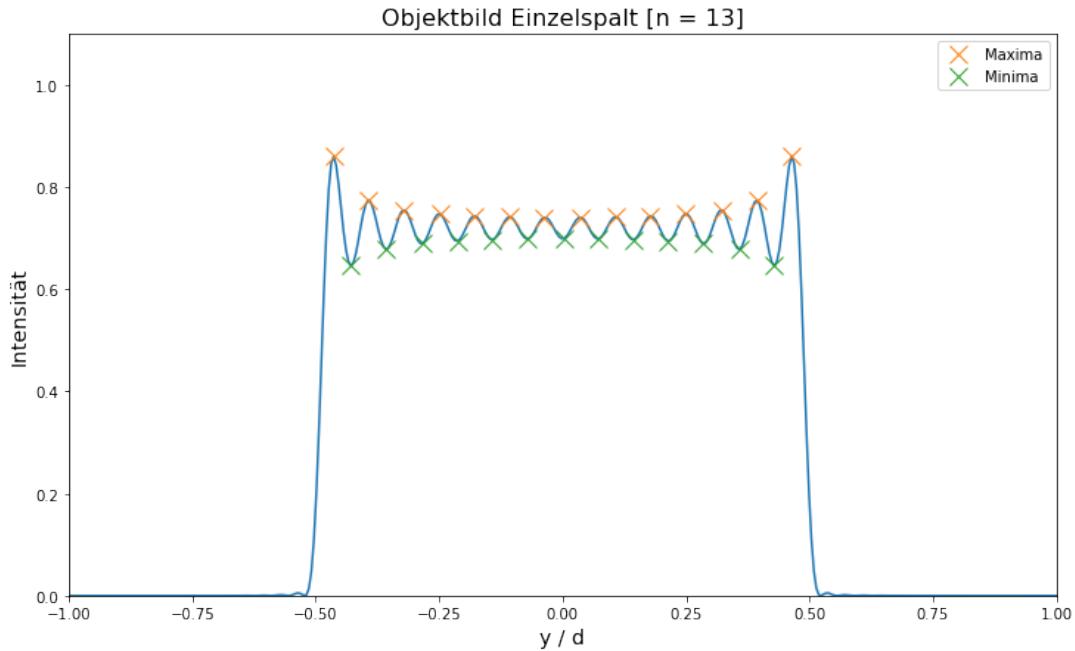
Richtiger Abstand Minima: good_dist_min = 10.56 +/- 0.44 [px]

Intensität der 1. Maxima: 0.8597
Intensität der 2. Maxima: 0.7739
Intensität der 3. Maxima: 0.7548
Intensität der 4. Maxima: 0.7471
Intensität der 5. Maxima: 0.7435
Intensität der 6. Maxima: 0.7415
Intensität der 7. Maxima: 0.7403
Intensität der 8. Maxima: 0.7403
Intensität der 9. Maxima: 0.7415
Intensität der 10. Maxima: 0.7435
Intensität der 11. Maxima: 0.7471

Intensität der 12. Maxima: 0.7548

Intensität der 13. Maxima: 0.7739

Intensität der 14. Maxima: 0.8597



Vergleich der Intensitätsbilder:

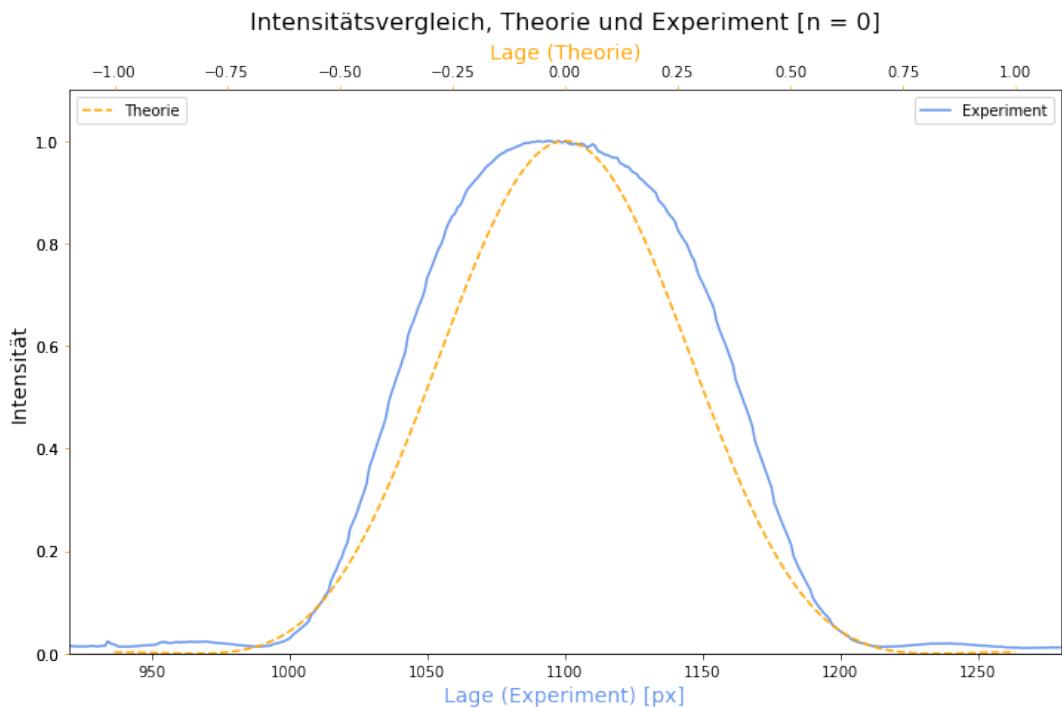
```
[34]: # Bild bei n = 0
x_values1 = pixel0
y_values1 = intensity0

x_values2 = Y
y_values2 = f_mod(Y, 1)

fig = plt.figure(figsize = (12, 7))
plt.title('Intensitätsvergleich, Theorie und Experiment [n = 0]\n\n', size = 16)
plt.axis('off')

ax = fig.add_subplot(111, label = "1")
ax2 = fig.add_subplot(111, label = "2", frame_on = False)
```

```
ax.plot(x_values1, y_values1, label = 'Experiment', color =  
        'cornflowerblue')  
ax.set_xlabel('Lage (Experiment) [px]', size = 14, color =  
        'cornflowerblue')  
ax.set_ylabel('Intensität', size = 14)  
ax.set_ylim([0, 1.1])  
ax.set_xlim([920, 1280])  
ax.xaxis.tick_bottom()  
#ax.tick_params(axis = 'x', color = 'cornflowerblue')  
ax.tick_params(axis = 'y', color = 'cornflowerblue')  
ax.legend()  
  
ax2.plot(x_values2, y_values2, color = 'orange', label = 'Theorie',  
         linestyle = '--')  
ax2.set_ylim([0, 1])  
ax2.xaxis.tick_top()  
#ax2.yaxis.tick_right()  
ax2.set_xlabel('Lage (Theorie)', color = 'orange', size = 14)  
ax2.xaxis.set_label_position('top')  
ax2.tick_params(axis = 'x', color = 'orange')  
ax2.tick_params(axis = 'y', color = 'orange')  
ax2.set_ylim((0, 1.1))  
ax2.legend(loc = 'upper left')  
  
plt.savefig('images/233/V233DiagrammIv0.png')
```



```
[35]: # Bild bei n = 1
x_values1 = pixel1
y_values1 = intensity1

x_values2 = Y
y_values2 = f_mod(Y, 2)

fig = plt.figure(figsize = (12, 7))
plt.title('Intensitätsvergleich, Theorie und Experiment [n = 1]\n\n', size = 16)
plt.axis('off')

ax = fig.add_subplot(111, label = "1")
ax2 = fig.add_subplot(111, label = "2", frame_on = False)

ax.plot(x_values1, y_values1, label = 'Experiment', color = 'cornflowerblue')

```

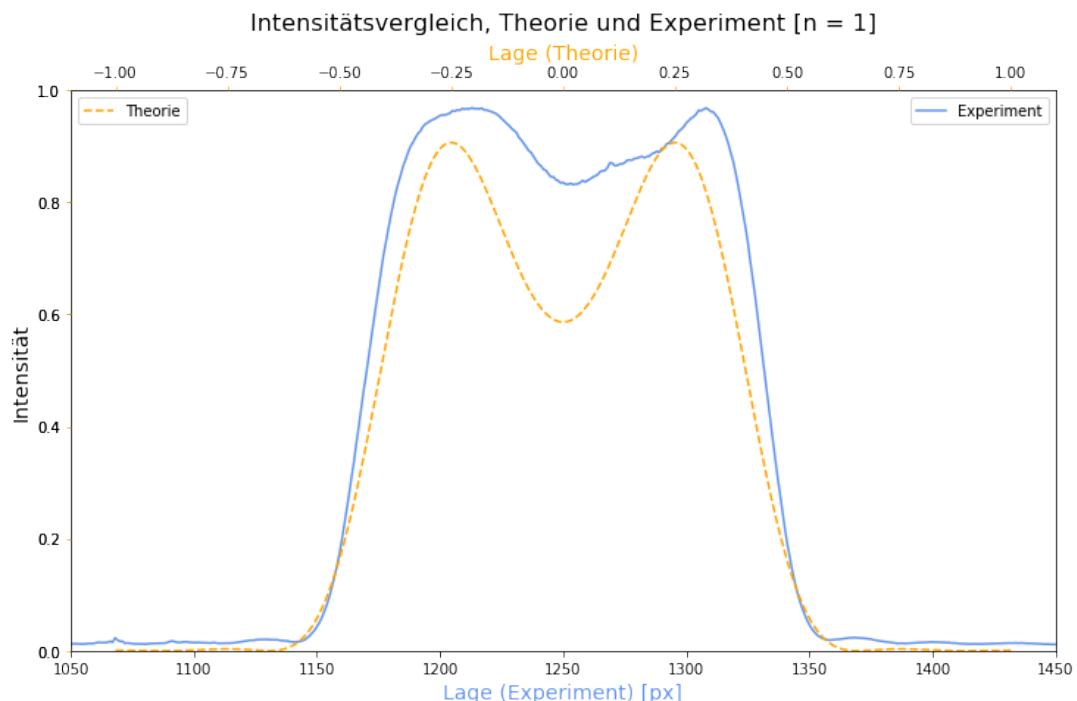
```

ax.set_xlabel('Lage (Experiment) [px]', size = 14, color = 'cornflowerblue')
ax.set_ylabel('Intensität', size = 14)
ax.set_ylim([0, 1])
ax.set_xlim([1050, 1450])
ax.xaxis.tick_bottom()
#ax.tick_params(axis = 'x', color = 'cornflowerblue')
ax.tick_params(axis = 'y', color = 'cornflowerblue')
ax.legend()

ax2.plot(x_values2, y_values2, color = 'orange', label = 'Theorie', linestyle = '--')
ax2.set_ylim([0, 1])
ax2.xaxis.tick_top()
#ax2.yaxis.tick_right()
ax2.set_xlabel('Lage (Theorie)', color = 'orange', size = 14)
ax2.xaxis.set_label_position('top')
ax2.tick_params(axis = 'x', color = 'orange')
ax2.tick_params(axis = 'y', color = 'orange')
ax2.legend(loc = 'upper left')

plt.savefig('images/233/V233DiagrammIv1.png')

```



```
[36]: # Bild bei n = 2
maxima2, _ = find_peaks(f_mod(Y, 3), height = 0.5)

x_values1 = pixel2
y_values1 = intensity2

x_values2 = Y
y_values2 = f_mod(Y, 3)

fig = plt.figure(figsize = (12, 7))
plt.title('Intensitätsvergleich, Theorie und Experiment [n = 2]\n\n', size = 16)
plt.axis('off')

ax = fig.add_subplot(111, label = "1")
ax2 = fig.add_subplot(111, label = "2", frame_on = False)

ax.plot(x_values1, y_values1, label = 'Experiment', color = 'cornflowerblue')
ax.plot(maxima_exp2, intensity2[maxima_exp2], 'x', markersize = 10,
        label = 'Maxima')
ax.plot(minima_exp2, intensity2[minima_exp2], 'x', markersize = 10,
        label = 'Minima', color = 'green')
ax.plot(1289, intensity2[1289], 'x', markersize = 10,
        color = 'green') # Python braucht ein bisschen Hilfe
ax.set_xlabel('Lage (Experiment) [px]', size = 14, color = 'cornflowerblue')
ax.set_ylabel('Intensität',
             size = 14)
ax.set_ylim([0, 1])
ax.set_xlim([1055, 1445])
ax.xaxis.tick_bottom()
#ax.tick_params(axis = 'x', color = 'cornflowerblue')
```

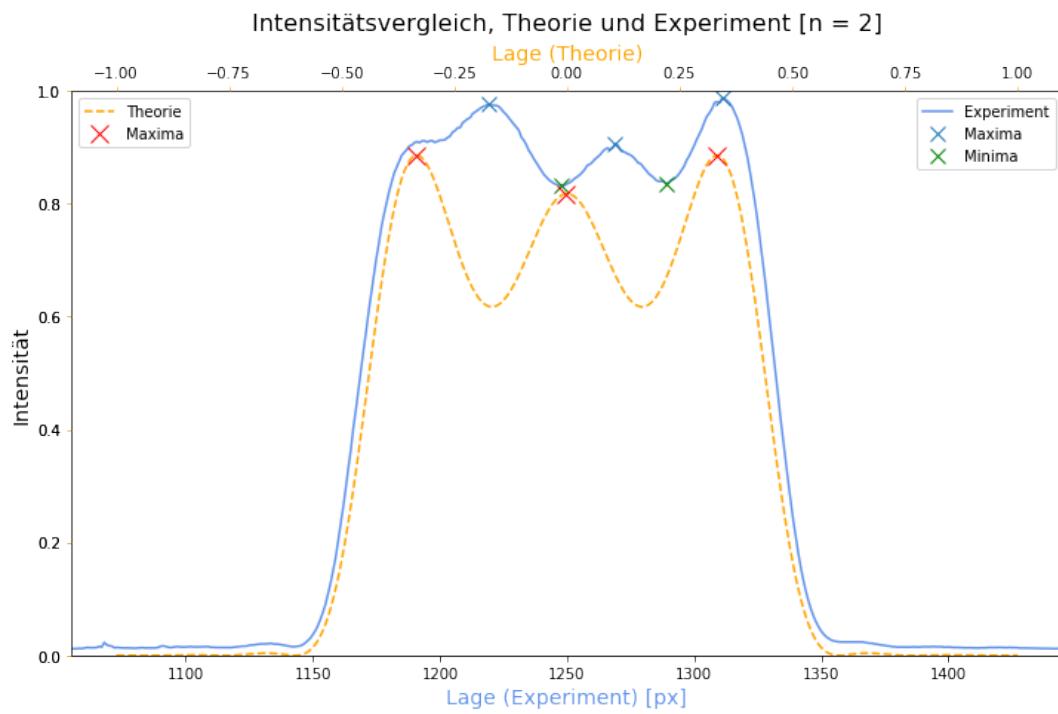
```

ax.tick_params(axis = 'y', color = 'cornflowerblue')
ax.legend()

ax2.plot(x_values2, y_values2, color = 'orange', label = 'Theorie',
          linestyle = '--')
ax2.plot(Y[maxima2], f_mod(Y[maxima2], 3), 'x', markersize = 12,
          label = 'Maxima', color = 'red')
ax2.set_ylim([0, 1])
ax2.xaxis.tick_top()
#ax2.yaxis.tick_right()
ax2.set_xlabel('Lage (Theorie)', color = 'orange', size = 14)
ax2.xaxis.set_label_position('top')
ax2.tick_params(axis = 'x', color = 'orange')
ax2.tick_params(axis = 'y', color = 'orange')
ax2.legend(loc = 'upper left')

plt.savefig('images/233/V233DiagrammIv2.png')

```



```
[37]: # Bild bei 13 Maxima
x_values1 = pixel13
y_values1 = intensity13

x_values2 = Y
y_values2 = f_mod(Y, 14)

fig = plt.figure(figsize = (12, 7))
plt.title('Intensitätsvergleich, Theorie und Experiment [n = 13]\n\n', size = 16)
plt.axis('off')

ax = fig.add_subplot(111, label = "1")
ax2 = fig.add_subplot(111, label = "2", frame_on = False)

ax.plot(x_values1, y_values1, label = 'Experiment', color = 'cornflowerblue')
ax.axvline()
ax.set_xlabel('Lage (Experiment) [px]', size = 14, color = 'cornflowerblue')
ax.set_ylabel('Intensität', size = 14)
ax.set_ylim([0, 1.2])
ax.set_xlim([1120, 1450])
ax.xaxis.tick_bottom()
#ax.tick_params(axis = 'x', color = 'cornflowerblue')
ax.tick_params(axis = 'y', color = 'cornflowerblue')
ax.legend()

ax2.plot(x_values2, y_values2, color = 'orange', label = 'Theorie',
          linestyle = '--')
ax2.axvline(Y[maxima_theo13[0]], color = 'red', linestyle = '--',
            label = 'Lage der Rand-Maxima', linewidth = 1)
ax2.axvline(Y[maxima_theo13[13]], color = 'red', linestyle = '--',
            linewidth = 1)
ax2.set_ylim([0, 1.2])
ax2.xaxis.tick_top()
```

```
#ax2.yaxis.tick_right()

ax2.set_xlabel('Lage (Theorie)', color = 'orange', size = 14)
ax2.xaxis.set_label_position('top')
ax2.tick_params(axis = 'x', color = 'orange')
ax2.tick_params(axis = 'y', color = 'orange')
ax2.legend(loc = 'upper left')

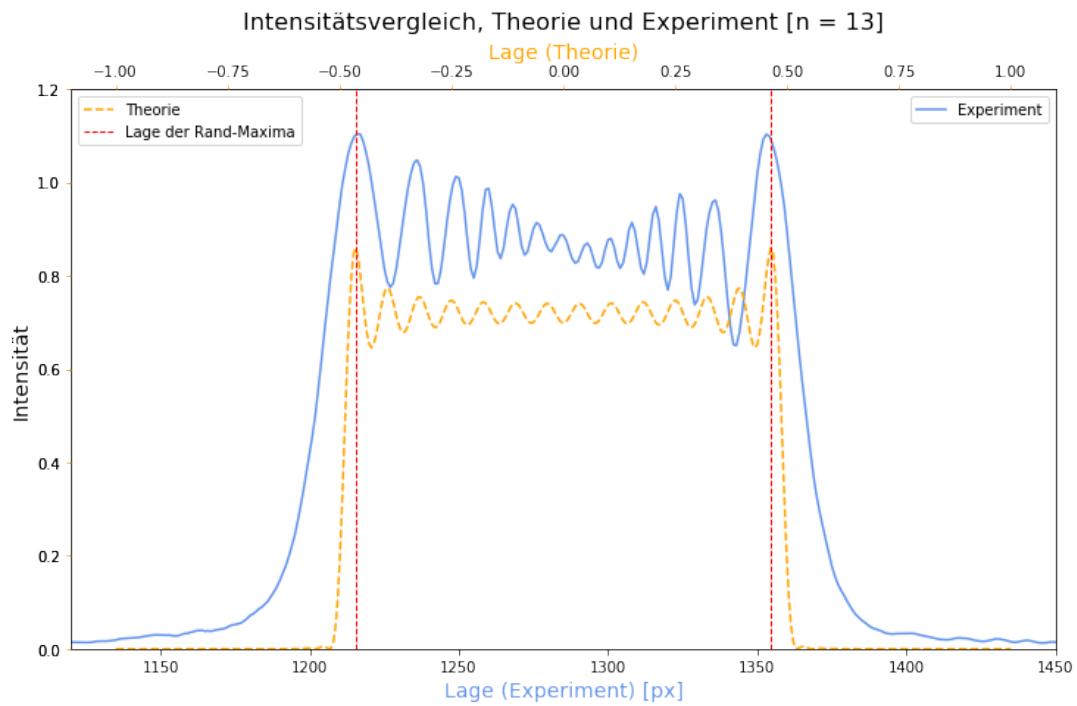
plt.savefig('images/233/V233DiagrammIv13.png')

Delta_ex = np.abs(maxima_exp13[0] - maxima_exp13[-1])
sig_Delta_ex = 5
Delta_theo = np.abs(Y[maxima_theo13[0]] - Y[maxima_theo13[-1]])
sig_Delta_theo = 0.01

norm_2_px = Delta_ex / Delta_theo
sig_norm_2_px = norm_2_px * sqrt( (sig_Delta_ex / Delta_ex) ** 2
                                  + (sig_Delta_theo / Delta_theo) ** 2 )

print('Abstand der Maxima (exp): ', Delta_ex)
print('Abstand der Maxima (theo): ', Delta_theo)
ergebnis('norm_2_px', norm_2_px, sig_norm_2_px, 0, '[px]')
```

Abstand der Maxima (exp): 137
Abstand der Maxima (theo): 0.9273182957393483
norm_2_px = 148.0 +/- 6.0 [px]



VIII.4 Fourierbild des Doppelspaltes

Spaltbreite und Spaltmittelabstand

```
[38]: # Spaltbreite
B = 136 # px
sig_B = 10
b = 0.845 # m
sig_b = 0.001
f = 0.08 #

G = B / (b / f - 1)
sig_G = sqrt((sig_B / (b / f - 1)) ** 2
             + (((B / f) * sig_b) / ((b / f - 1) ** 2)) ** 2)

scale = 14
G_m2 = scale * G
sig_G_m2 = scale * sig_G
```

```
ergebnis('Spaltbreite: G', G_m2, sig_G_m2, 0, '[10^-6 m]')
```

Spaltbreite: G = 199.0 +/- 15.0 [10^-6 m]

[47]: # Berechnung für den Spaltmittenabstand

```
B = 409
sig_B = 10
```

```
G_m3 = B / (b / f - 1)
sig_G_m3 = sqrt((sig_B / (b / f - 1)) ** 2
                 + (((B / f) * sig_b) / ((b / f - 1) ** 2)) ** 2)

G_mm_m3 = scale * G_m3
sig_G_mm_m3 = scale * sig_G_m3
```

```
ergebnis('Spaltmittelabstand: G_mittel', G_mm_m3, sig_G_mm_m3, 0,
         '[10^-6 m]')
```

Spaltmittelabstand: G_mittel = 599.0 +/- 15.0 [10^-6 m]

[40]: def doppelspalt(k, y):

```
    return d / pi * np.cos(k * g / 2) * np.sin(k * d / 2) / (k * d / 2)
    ↵* np.cos(y * k)
```

```
d = 0.199 # Spaltbreite
```

```
g = 2.177 * d
```

```
n = 1
```

```
Y = np.linspace(-1, 1, 400)
```

f modifiziert

def f_mod_d(Y, n):

```
f_modifiziert = []
```

```
for i in range(len(Y)):
```

```
    y = Y[i]
```

```
    result, error = quad(doppelspalt, 0, 2 * pi * n / d, args =
    ↵(y,))
```

```
f_modifiziert.append((result ** 2))
```

```
    return f_modifiziert / np.max(f_modifiziert)
```

```
[41]: # Plot 0. Ordnung
maxima_dop0, _ = find_peaks(f_mod_d(Y, 1), height = 0.5)

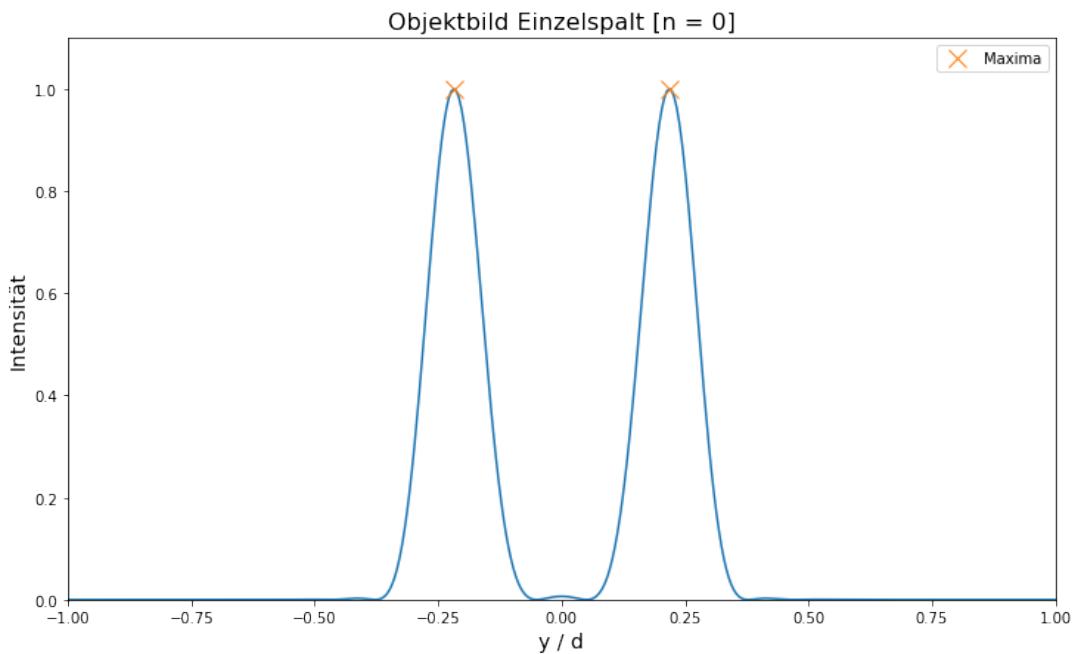
plt.figure(figsize = (12, 7))
plt.plot(Y, f_mod_d(Y, 1))
plt.plot(Y[maxima_dop0], f_mod_d(Y[maxima_dop0], 1), 'x', markersize = 12,
         label = 'Maxima')
plt.xlabel('y / d', size = 14)
plt.ylabel('Intensität', size = 14)
plt.title('Objektbild Einzelspalt [n = 0]', size = 16)
plt.xlim((-1, 1))
plt.ylim((0, 1.1))
plt.legend(loc = 'best')

plt.savefig('images/233/V233DiagrammIDop0.png')

# Intensität der Maxima, aufsteigend von links nach rechts
ID_0_theo = intensitaet(f_mod_d(Y, 1), maxima_dop0)
```

Intensität der 1. Maxima: 1.0

Intensität der 2. Maxima: 1.0

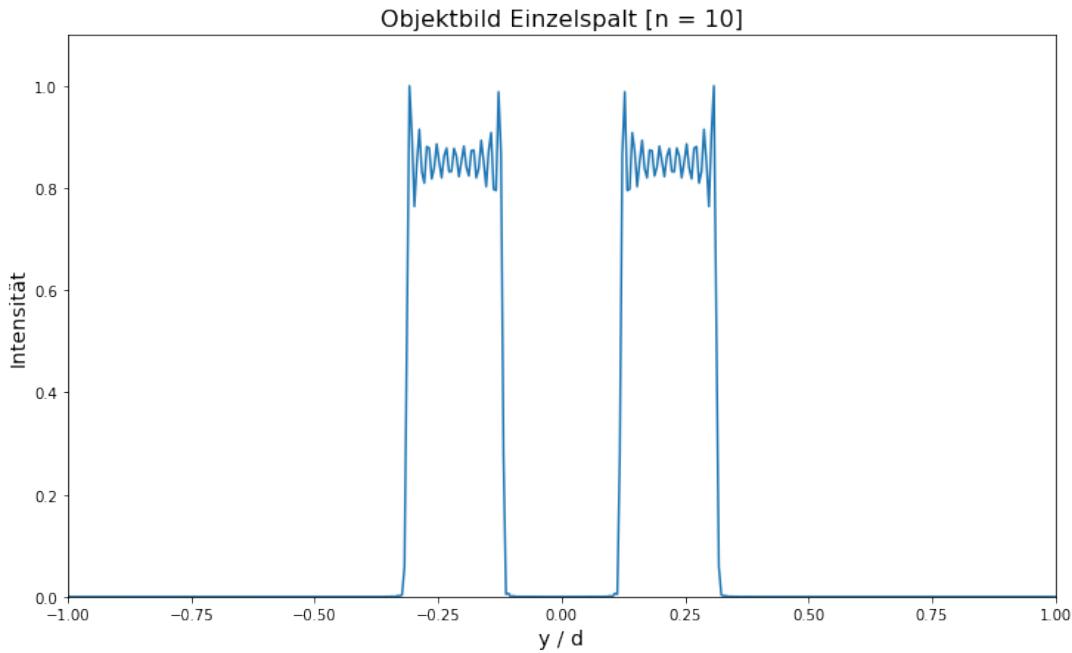


```
[42]: # Plot 13. Ordnung
#maxima_dop13, _ = find_peaks(f_mod_d(Y, 10), height = 0.5)

plt.figure(figsize = (12, 7))
plt.plot(Y, f_mod_d(Y, 11))
# plt.plot(Y[maxima_dop13], f_mod_d(Y[maxima_dop13], 10), 'x',
#          markersize = 12,
#          label = 'Maxima')
plt.xlabel('y / d', size = 14)
plt.ylabel('Intensität', size = 14)
plt.title('Objektbild Einzelpalt [n = 10]', size = 16)
plt.xlim((-1, 1))
plt.ylim((0, 1.1))
# plt.legend(loc = 'best')

plt.savefig('images/233/V233DiagrammIDop13.png')

# Intensität der Maxima, aufsteigend von links nach rechts
#ID_0_theo = intensitaet(f_mod_d(Y, 10), maxima_dop13)
```



[43]: # Plot 1. Ordnung

```

n = 0.3076
m = 0.308
sig_n = 0.0005

# Suche Übergang von 2 auf 1 Maxima
maxima_dop1, _ = find_peaks(f_mod_d(Y, n), height = 0.5)
maxima_dop1_m, _ = find_peaks(f_mod_d(Y, m), height = 0.5)

plt.figure(figsize = (12, 7))
plt.plot(Y, f_mod_d(Y, n), label = 'Plateaukurve mit n = ' + str(n))
plt.plot(Y, f_mod_d(Y, m), label = 'Fehlerkurve mit n = ' + str(m),  

         linestyle = '--')
plt.plot(Y, f_mod_d(Y, 1), label = 'Gut erkennbare Doppelspaltstruktur')
plt.plot(Y[maxima_dop1], f_mod_d(Y[maxima_dop1], n), 'x', markersize =  

         12,  

         label = 'Maxima der Plateaukurve')
plt.plot(Y[maxima_dop1_m], f_mod_d(Y[maxima_dop1_m], m), 'x',  

         markersize = 12, label = 'Maxima der Fehlerkurve')
plt.xlabel('y / d', size = 14)

```

```

plt.ylabel('Intensität', size = 14)
plt.title('Übergang zur Auflösung der Doppelspaltstruktur', size = 16)
plt.xlim((-1, 1))
plt.ylim((0,1.1))
plt.legend(loc = 'best')

plt.savefig('images/233/V233DiagrammIDopAuf.png')

```

```

# Intensität der Maxima, aufsteigend von links nach rechts
print('Maxima der Plateaukurve')
ID_n_theo = intensitaet(f_mod_d(Y, n), maxima_dop1)
print('Maxima der Fehlerkurve')
ID_m_theo = intensitaet(f_mod_d(Y, m), maxima_dop1_m)

```

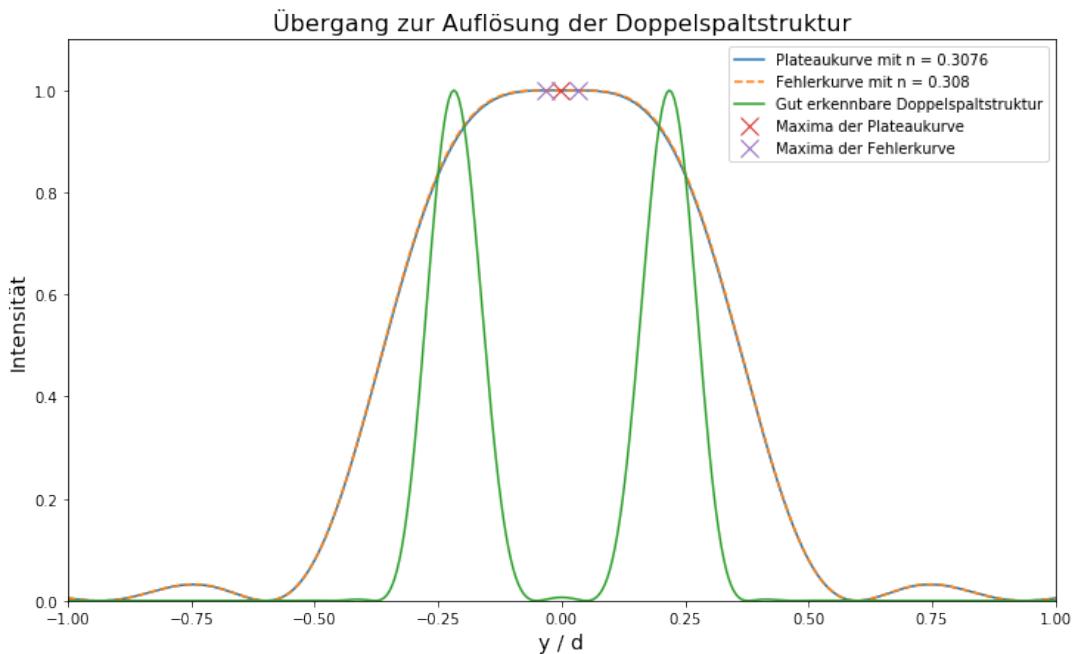
Maxima der Plateaukurve

Intensität der 1. Maxima: 1.0

Maxima der Fehlerkurve

Intensität der 1. Maxima: 1.0

Intensität der 2. Maxima: 1.0



[48]: # Berechnung k: Theoretisch

```
k_yb_theo = 2 * n * pi / (G_m2 * 1e-3)
k_ya_theo = 2 * pi / (G_m2 * 1e-3)
sig_k_yb_theo = k_yb_theo * sqrt((sig_n / n) ** 2 + (sig_G_m2 / G_m2) ** 2)
sig_k_ya_theo = k_ya_theo * sqrt((0 / n) ** 2 + (sig_G_m2 / G_m2) ** 2)

ergebnis('Theoretisches k_ya_theo', k_ya_theo, sig_k_ya_theo, 1, '[10^-3 m^-1]')
ergebnis('Theoretisches k_yb_theo', k_yb_theo, sig_k_yb_theo, 1, '[10^-3 m^-1]')
```

Theoretisches k_ya_theo = 31.6 +/- 2.3 [10^-3 m^-1]

Theoretisches k_yb_theo = 9.7 +/- 0.7 [10^-3 m^-1]

[49]: # Berechnung k: Experimentell

```
d_a = 0.27 # mm
sig_d_a = 0.01
d_b = 0.07
sig_d_b = 0.01
f = 0.08 # m

k_ya_exp = (2 * pi * d_a) / (lambda_k * f) * 1e-6 # m^-1
sig_k_ya_exp = k_ya_exp * sig_d_a / d
k_yb_exp = (2 * pi * d_b) / (lambda_k * f) * 1e-6
sig_k_yb_exp = k_yb_exp * sig_d_b / d

ergebnis('Experimentelles k_ya_exp', k_ya_exp, sig_k_ya_exp, 1, '[10^-3 m^-1]')
ergebnis('Experimentelles k_yb_exp', k_yb_exp, sig_k_yb_exp, 1, '[10^-3 m^-1]')
```

Experimentelles k_ya_exp = 33.4 +/- 1.7 [10^-3 m^-1]

Experimentelles k_yb_exp = 8.7 +/- 0.4 [10^-3 m^-1]

Sigmas

VIII.1

```
[50]: # Mittlerer Abstand und Steigung der Gerade
fehler('Delta Minima:', np.abs(Delta_min_mean), sig_Delta_min_mean, np.
       ↪abs(m_2), sig_m_2)
```

Delta Minima:
 Relativer Fehler: 1.038565648770192
 Rel. Fehler (Vergleich): 0.4152166755667483
 Absoluter Fehler: -0.055084152360535654
 Sigma-Abweichung: 0.049616965332528

```
[51]: # Lage der Maxima
fehler('Maxima n:', npi, sig_npi, npi_lit, 0)
```

Maxima n:
 Relativer Fehler: [-162.12547099 1.37733047 0.83201939 0.
 ↪64972361
 0.57389313
 0.52848179]
 Rel. Fehler (Vergleich): [nan 0. 0. 0. 0. 0.]
 Absoluter Fehler: [-0.03431683 -0.26367047 -0.01021122 0.22848085 -0.
 ↪02283125
 -0.20352267]
 Sigma-Abweichung: [0.61680623 4.5187145 0.15904753 3.1586731 0.
 ↪28328284
 2.26303112]

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:17:_
 ↪RuntimeWarning:

invalid value encountered in true_divide

```
[52]: # Intensitätsverhältnisse: VIII.1
fehler('Intensitätsverhältnisse: Beugung am Einfachspalt vw', vw,
       ↪sig_vw, vw_theo, sig_vw_theo)
```

Intensitätsverhältnisse: Beugung am Einfachspalt vw

Relativer Fehler: [10.27855513 35.30482875 18.55634294 26.48000676 32.
 ↵71810507
 36.77129899]

Rel. Fehler (Vergleich): [nan 16.22138395 32.12141664 19.06588898
 1.3428207 64.47780936]

Absoluter Fehler: [0. 0.03676927 0.01184094 0.00553987 0.
 ↵00405718
 0.00383066]

Sigma-Abweichung: [nan 1.20108047 1.58741191 1.38334078 1.
 ↵36444534
 1.12046697]

VIII.2

```
[53]: # Intensitätsverhältnisse: VIII.2
print(vwD)
print(vwD_theo)
fehler('Intensitätsverhältnisse: Beugung am Doppelspalt vw:', vwD,
      ↵sig_vwD, vwD_theo, sig_vwD_theo)
```

[1. 0.6798082 0.08519115 0.04015745 0.04302584 0.02294711]
[0.99477724 0.69480014 0.19164735 0.04284635 0.02930816 0.01400354]

Intensitätsverhältnisse: Beugung am Doppelspalt vw:

Relativer Fehler: [10.40342049 12.59635937 29.10451798 42.37622549 40.
 ↵94023828
 56.05095871]

Rel. Fehler (Vergleich): [0.50262509 0.7196314 2.6089586 11.66960545
 17.06009253 35.70526377]

Absoluter Fehler: [0.00522276 -0.01499194 -0.10645619 -0.0026889 0.
 ↵01371767
 0.00894357]

Sigma-Abweichung: [0.05014446 0.17477818 4.20882009 0.15160233 0.
 ↵74915897
 0.64809703]

VIII.3

```
[54]: # Intensitätsverhältnisse: VIII.3, graphisch bestimmt

# n = 1
fehler('Intensitätsverhältnis [n = 1]:', I_1_ex_gr, sig_I, I_1_theo, 0.
       ↪001)

# n = 2
fehler('Intensitätsverhältnis [n = 2]:', I_2_ex_gr, sig_I, I_2_theo, 0.
       ↪001)

# n = 13
#fehler('Intensitätsverhältnis [n = 13]:', I_13_ex_gr, sig_I,
       ↪I_13_theo, 0.001)
```

Intensitätsverhältnis [n = 1]:

Relativer Fehler: [1.03223541 1.19194326 1.03250098]

Rel. Fehler (Vergleich): [0.11028891 0.1704849 0.11028891]

Absoluter Fehler: [0.06206182 0.2524039 0.06181265]

Sigma-Abweichung: [6.17538222 25.11512629 6.15058815]

Intensitätsverhältnis [n = 2]:

Relativer Fehler: [1.02441229 1.20083353 1.10379406 1.19739451 1.
 ↪01296673]

Rel. Fehler (Vergleich): [0.11287798 0.16193485 0.12227421 0.16193485
0.11287798]

Absoluter Fehler: [0.0902571 0.21522258 0.08813216 0.21761433 0.
 ↪10128688]

Sigma-Abweichung: [8.98091663 21.41544752 8.76947753 21.65343504 10.
 ↪07842093]

```
[55]: # Intensitätsverhältnisse: VIII.3, manuell bestimmt

# n = 1
fehler('Intensitätsverhältnis [n = 1]:', IF_10, sig_IF_10, I_1_theo, 0.
       ↪001)

# n = 2
fehler('Intensitätsverhältnis [n = 2]:', IF_20, sig_IF_20, I_2_theo, 0.
       ↪001)

# n = 3
```

```
fehler('Intensitätsverhältnis [n = 3] : ', IF_30, sig_IF_30, I_3_theo, 0.  
→001)
```

Intensitätsverhältnis [n = 1]:

Relativer Fehler: [2.00533191 2.18033438 2.0234913]

Rel. Fehler (Vergleich): [0.11028891 0.1704849 0.11028891]

Absoluter Fehler: [0.06273501 0.24677112 0.04606834]

Sigma-Abweichung: [3.22275251 13.56112446 2.38630623]

Intensitätsverhältnis [n = 2]:

Relativer Fehler: [1.99653963 2.17393916 2.08831396 2.17182444 1.
→98651484]

Rel. Fehler (Vergleich): [0.11287798 0.16193485 0.12227421 0.16193485
0.11287798]

Absoluter Fehler: [0.0918654 0.21996769 0.08077717 0.22135658 0.
→10158762]

Sigma-Abweichung: [4.69964263 12.06347974 4.29838479 12.13137635 5.
→17187875]

Intensitätsverhältnis [n = 3]:

Relativer Fehler: [2.02972077 2.70379159 2.05556666 2.12735713 2.
→08653161

2.26350272

1.95918617]

Rel. Fehler (Vergleich): [0.11406327 0.1590753 0.12530989 0.15397712 0.
→12530989

0.15902925

0.11406327]

Absoluter Fehler: [0.07051582 -0.02585532 0.12697839 0.21999732 0.
→10197839

0.15451818

0.13857137]

Sigma-Abweichung: [3.66278739 1.58344662 6.66893836 11.87685183 5.
→42282709

8.7008633

6.95768566]

```
[56]: # Äquidistanz der Maxima
fehler('Äquidistanz der Maxima:', dist_max13, sig_dist_max13, good_dist,_
      ↵sig_good_dist)
```

Äquidistanz der Maxima:

Relativer Fehler: 9.777441643154155

Rel. Fehler (Vergleich): 4.1305523852638615

Absoluter Fehler: -0.018700597647965722

Sigma-Abweichung: 0.016713867183029182

```
[57]: # Äquidistanz der Minima
fehler('Äquidistanz der Minima:', dist_min13, sig_dist_min13,_
      ↵good_dist_min, sig_good_dist_min)
```

Äquidistanz der Minima:

Relativer Fehler: 7.438887884551808

Rel. Fehler (Vergleich): 4.140728437338061

Absoluter Fehler: -0.890495469442838

Sigma-Abweichung: 1.058174557675539

VIII.4

```
[58]: # Spaltbreite, VIII.1, 3
fehler('Spaltbreite b_S, G_m:', 1e10 * b_S, 1e10 * sig_b_S, G_m,_
      ↵sig_G_m)
```

Spaltbreite b_S, G_m:

Relativer Fehler: 0.4154333834607962

Rel. Fehler (Vergleich): 7.725667168394465

Absoluter Fehler: -10.01745303127251

Sigma-Abweichung: 0.5951024915149186

```
[59]: # Spaltbreite, VIII.1, 4
fehler('Spaltbreite b_S, G_m2:', 1e10 * b_S, 1e10 * sig_b_S, G_m2,_
      ↵sig_G_m2)
```

Spaltbreite b_S, G_m2:

Relativer Fehler: 0.4154333834607962

Rel. Fehler (Vergleich): 7.35410303093698

Absoluter Fehler: 8.471435857616399

Sigma-Abweichung: 0.577537198886647

[60]: # Spaltbreite, VIII.3, 4

```
fehler('Spaltbreite G_m, G_m2:', G_m, sig_G_m, G_m2, sig_G_m2)
```

Spaltbreite G_m, G_m2:

Relativer Fehler: 7.725667168394465

Rel. Fehler (Vergleich): 7.35410303093698

Absoluter Fehler: 18.4888888888891

Sigma-Abweichung: 0.8293198915589404

[61]: # k_ya

```
fehler('k_ya:', k_ya_exp, sig_k_ya_exp, k_ya_theo, sig_k_ya_theo)
```

k_ya:

Relativer Fehler: 5.025125628140704

Rel. Fehler (Vergleich): 7.35410303093698

Absoluter Fehler: 1.8387063302260316

Sigma-Abweichung: 0.6420390065954883

[62]: # k_yb

```
fehler('k_yb:', k_yb_exp, sig_k_yb_exp, k_yb_theo, sig_k_yb_theo)
```

k_yb:

Relativer Fehler: 5.025125628140703

Rel. Fehler (Vergleich): 7.355899230584866

Absoluter Fehler: -1.0487472916671159

Sigma-Abweichung: 1.2542966418203396

[]: