

WS19/20, PAP2.1, Versuch 233/333:
Fourieroptik

Praktikant:
Gerasimov, Vsevolod

zusammen mit
Reiter, Leonardo

Betreuer:
Fischer, Carola

Durchführung am
26. November & 3. Dezember, 2019

Inhaltsverzeichnis

	Seite
1 Einführung	2
2 Versuchsaufbau, Literaturwerte & Vorbereitung	2
3 Durchführung	3
3.1 Quantitative Beobachtungen am Einfachspalt	3
3.2 Beugungsstruktur des Doppelspaltes	4
3.3 Das Objektbild als Fouriersynthese des Beugungsbildes	4
3.4 Fourierbild des Doppelspaltes	5
4 Messergebnisse	5
5 Auswertung	5
5.1 Quantitative Beobachtungen am Einfachspalt	5
5.1.1 Rechnung	8
5.1.2 Source Code & Input	9
5.1.3 Output	14
5.2 Beugungsstruktur des Doppelspaltes	16
5.2.1 Rechnung	16
5.2.2 Source Code & Input	16
5.2.3 Output	21
5.3 Das Objektbild als Fouriersynthese des Beugungsbildes	22
5.3.1 Rechnung	22
5.3.2 Source Code & Input	22
5.3.3 Output	25
5.4 Fourierbild des Doppelspaltes	25
5.4.1 Rechnung	25
5.4.2 Source Code & Input	26
5.4.3 Output	27
6 Fazit	28
7 Anhang	28
Quantitative Beobachtungen am Einfachspalt	29
Beugungsstruktur des Doppelspaltes	35
Fouriersynthese des Beugungsbildes am Einzelspalt	37
Fourierbild des Doppelspaltes	42

1 Einführung¹

In diesem Versuch wollen wir simultan sowohl das Beugungsbild des Objektes (hier Spalte, Gitter, Kreuzgitter etc.) als auch das übliche Bild des Objektes selbst darzustellen. Das gelingt uns mit einem einfachen optischen Aufbau mit wenigen Linsen, einem Strahlteiler sowie einem kleinen Diodenlaser als spektral schmale, intensive Parallellichtquelle.

Durch eine Fouriertransformation eines Signals kann mit ihr das Spektrum gewonnen werden, das Aufschluss über die Bestandteile des Signals liefert. Mit Hilfe von Eingriffen in der „Fourierebene“ - das ist die Ebene, in der die Beugungsstruktur erstmals auftritt - werden gezielt die Beiträge der einzelnen Beugungsordnungen zum Objektbild sichtbar gemacht und können quantitativ verfolgt werden.

Genauer befassen wir uns mit den quantitativen Beobachtungen am Einfachspalt, der Beugungsstruktur des Doppelspaltes, dem Objektbild als Fouriersynthese des Beugungsbildes und dem Fourierbild des Doppelspaltes.

2 Versuchsaufbau¹, Literaturwerte & Vorbereitung

- Singlemode Glasfaser-gekoppelter Diodenlaser (Wellenlänge $\lambda = 635(1) \text{ nm}$) mit regelbarer Stromversorgung und Justieroptik für einen Parallelstrahl

¹Dr. J.Wagner - Physikalisches Anfängerpraktikum - V. 1.1 Stand 01/2018, Versuch 233/333

- 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
- SCCD-Zeilenkamera mit PC und Monitor zur Aufnahme von Beugungsbildern und „Objektbildern“
- Uns schon zuvor Bereitgestellte² Abbildung 1 zeigt den ganzen Aufbau skizziert.

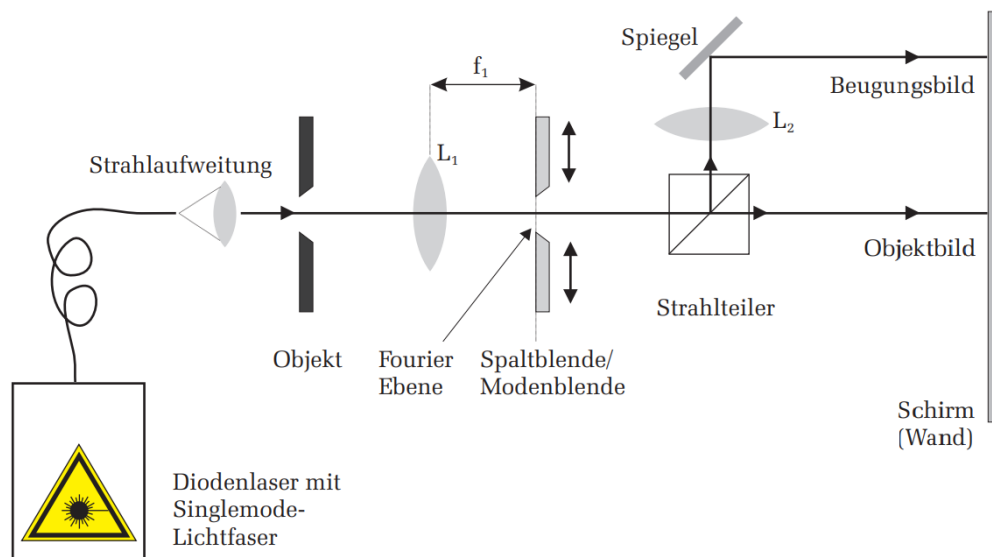


Abbildung 1: Schematische Darstellung des optischen Aufbaus.

3 Durchführung²

3.1 Quantitative Beobachtungen am Einfachspalt

Zunächst wollen wir die Beugungsstruktur des Einfachspaltes „B“ bestimmen: Zur Messung steht uns eine CCD-Zeilenkamera zur Verfügung, die über eine USB-Schnittstelle auf einen PC ausgelesen wird und die es erlaubt, die Lichtintensität als Funktion des Ortes darzustellen. Der lichtempfindliche Teil besteht aus **2048** Pixel der jeweiligen Breite $1\text{ px} = 14.0(2)\,\mu\text{m}$ und Höhe $h_{\text{px}} = 56.0(1)\,\mu\text{m}$ und einer Gesamtbreite $b_{\text{CCD}} = 2048\text{ px} = 28.7(1)\text{ mm}$.

Wir justieren Linse L_2 auf ein scharfes Beugungsbild in der Ebene der Kamera. Sie ist durch einen Doppelpfeil auf dem Kameragehäuse gekennzeichnet. Danach schieben wir diese dazu ein wenig zur Seite und schauen uns das Beugungsbild auf einem daneben gehaltenen Blatt Papier an. Wir schieben nun die Zeilenkamera mittig über die Beugungsstruktur und sehen uns das Ergebnis auf dem PC-Monitor an. Da das Beugungsbild aus einer linearen, symmetrischen Anordnung von hellen Punkten besteht, deren Intensität nach außen stark abnimmt, und diese Punktklinie muss mit der Pixel-Zeile der Kamera zur Deckung gebracht werden. Wir suchen uns also unter den zur Verfügung stehenden optischen Bauteilen nach einer Zylinderlinse, um das Beugungsbild vertikal aufzufächern, und reduzieren

²Dr. J.Wagner - Physikalisches Anfängerpraktikum - V. 1.1 Stand 01/2018, Versuch 233/333

stark die Laserintensität, weit unter die Sichtbarkeitsgrenz, weil die verwendete Zeilenkamera sehr empfindlich ist und sehr rasch überstreut.

Wenn wir schließlich die Beugungsstruktur auf dem Monitor haben, können wir durch Feinjustieren des optischen Aufbaus (Höheneinstellung der Zylinderlinse, L_2 , transversale Position des Spaltes auf der optischen Bank) eine optimale Symmetrie der beobachteten Beugungsstruktur (Abb.3) herstellen. Mit Hilfe des von der Bedienungssoftware zur Verfügung gestellten Cursor können wir nun (Betriebsmode `view` der Kamera) die Lage und Intensitäten der verschiedenen Beugungsmaxima und -minima bis zur 5. Ordnung sowie auch den Untergrund (jeweils rechts und links) bestimmen und protokollieren (Tabelle 1). Wegen des hohen Intensitätsabfalles zu steigender Ordnung der Beugungsmaxima müssen wir zur besseren Vermessung der höheren Beugungsordnungen ein zweites Bild (Abb.4) bei höherer Intensität des Diodenlasers aufnehmen, die wir über den Strom regeln können. Wir können hierüber alle Intensitäten relativ zur Intensität des 0-ten Maximums angeben.

Die Eichung der Abszisse (d.h. wieviel Pixel auf der Kamera entsprechen 1 mm in der Fourierebene) erhalten wir, indem wir in der Fourierebene den Analyserspalt auf verschiedene Weiten einstellen und die zugehörigen Abstände in Pixel auf dem Monitor ablesen.

Die Ablesung der Spaltbreite d erfolgt mit dem integrierten Messfühler. Wir beachten, dass der Messfühler die Verschiebung $\frac{d+d_0}{2}$ nur einer Spaltschneide registriert. Die andere bewegt sich dabei gegenläufig, so dass die wahre Spaltweite dem doppelten und genullten Ablesewert $2\frac{d+d_0}{2} - d_0$ entspricht. Wir wiederholen dies für mehrere gut ablesbare Punkte des Beugungsbildes und protokollieren die Daten (Tabelle 2).

3.2 Beugungsstruktur des Doppelspaltes

Wir ersetzen den Einzelspalt durch das Rähmchen mit dem Doppelspalt ohne den Reiter zu verschieben und verschaffen uns zunächst wieder durch Feinjustieren des neuen Objektes ein scharfes Bild der Beugungsstruktur des Doppelspaltes auf dem Schirm. Wir schauen uns die Strukturen (Abb.12-14) für die verschiedenen Doppelspalte „A“, „B“ und „C“ in dem Dia-Rahmen an und versuchen die Strukturen zu interpretieren. Quantitative Messungen werden im Folgenden an dem mittleren Doppelspalt „B“ durchgeführt. Zur Beobachtung der Beugungsstruktur verwenden wir wieder die CCD-Kamera.

Wie bei der vorangegangenen Vermessung des Einzelspaltes, müssen wir auch hier wieder das Beugungsbild in die Ebene der Kamerazeile scharf einstellen und sich dann mit Hilfe von Zylinderlinse und Einregeln der Laserintensität ein brauchbares Bild der Intensitätsverteilung auf dem Monitor verschaffen. Wir justieren wieder auf Symmetrie und optimalen Kontrast und versuchen vorab schon den beobachteten Maxima und Minima, Beugungsordnungen der Spalt- und Gitterfunktion zuzuordnen. Für die Auswertung zu Hause bestimmen wir beim Einzelspalt mit Hilfe der Cursor Lage und Höhe der Maxima und Minima im Bereich zwischen den 1. Minima (links & rechts) der Spaltfunktion, sowie des Untergrunde (Tabelle 3). Die Eichung der Abszisse auf dem Monitorbild können wir von der entsprechenden Messung am Einzelspalt übernehmen. Es wurde ja lediglich das Rähmchen mit dem Einzelspalt gegen den Doppelspalt ausgetauscht. Gegenstandsweite g und Bildweite b bleiben unverändert.

Hätten wir versehentlich doch an den die Vergrößerung bestimmenden Positionen von Objekt (Dia-Rahmen), Fourierebene, L_2 oder die Position der Kamera nachjustiert, müssten wir die Eichung für diese Messung wiederholen.

3.3 Das Objektbild als Fouriersynthese des Beugungsbildes

Am Beispiel des Spaltes soll nun die Auswirkung der Manipulationen in der Fourierebene auf die Struktur des Bildes quantitativ untersucht werden. Die zunächst nur qualitativ mit dem Auge feststellbaren Änderungen am Bild, etwa: „Die Ränder erscheinen schärfer, wenn man die Beugungsordnungen höherer Ordnung mit zur Abbildung zulässt“, werden nun quantitativ gemessen.

Wir bestimmen die Intensitätsverteilung des Spaltbildes wieder mit Hilfe der CCD-Zeilenkamera und öffnen zunächst den Analyserspalt in der Fourierebene und stellen die Laserintensität am Steuergerät auf Maximum ein. In den Strahlengang zur Kamera stellen wir dicht hinter dem Strahlteiler einen Graufilter (Abschwächung 10-2). Ohne den Graufilter würde die Kamera übersteuern. Man beobachtet eine zunächst mehr oder weniger gut ausgeprägte Rechteckfunktion mit einer großen Zahl von „wiggles“. Wir lassen die Linse L_1 stehen und justieren vielmehr das Objekt selbst bezüglich dieser Linse nach, bis dessen Bild in der Ebene des Photodetektors scharf erscheint. Letzteres wird dadurch kenntlich, dass wir nun eine zunehmend ideale Rechteckfunktion auf dem Monitor beobachten können. Wir stellen auf größtmögliche Steilheit der Kanten des Spaltbildes ein. Dazu verrücken wir den Spalt sehr feinfühlig. Bei diesem Vorgehen bleibt dann die Lage der Fourierebene und damit das Beugungsbild auf dem Schirm unverändert. Danach können wir die Laserintensität geeignet einstellen.

Um sich zunächst einen Überblick über die sukzessiv auftretenden Strukturen zu verschaffen drehen wir den Ana-

lysierspalt nun langsam zu und beobachten simultan die Veränderungen der Beugungsstruktur auf dem Schirm und des Spaltbildes auf dem Monitor. Die einfachste Struktur des Spaltbildes erhalten wir offenbar, nachdem wir bis auf das zentrale Maximum alle Nebenmaxima ausgeblendet haben. Wir symmetrisieren das Fourierbild gleichzeitig durch Nachjustieren des Verschiebereiters, der den Analyserspalt trägt, bezüglich der 0. Ordnung des Beugungsbildes.

Es fällt auf, dass die Kurven nicht so glatt sind, wie wir es erwartet haben, deshalb entfernen wir den Graufilter und reduzieren gleichzeitig die Laserleistung und beobachten sofort einen deutlich bessere Signalqualität. Der Grund dafür ist, dass der verwendete Graufilter optisch nicht absolut homogen ist und beeinflusst Amplitude und Phase der durchlaufenden Teilbündel. Dies führt zu Störungen der Intensität in der Ebene der Kamera, in die die Linse L_1 letztlich alle Bündel zum modifizierten Bild des Spaltes zusammenführt.

Für unser Protokoll wollen wir die Bilder für die fünf ersten Einstellungen des Analyserspalt (Tabelle 4) auf das 1. bis 5. Minimum der Beugungsfigur aufnehmen (Abb.16-20). Zum quantitativen Vergleich mit den später auch zu berechnenden Bildern wollen wir jedoch beispielhaft an den ersten drei Bildern (Abb.16-18) auch die Intensitätsverhältnisse zahlenmäßig überprüfen und protokollieren dazu die Intensitäten der Maxima, Minima und des Untergrundes (Tabelle 5).

Die Zahl der beobachtbaren „Wiggels“ können wir aber bis zu wesentlich höheren Beugungsordnungen (ca. $n = 15$) verfolgen. Wir bestimmen an solch einem Bild (Abb.23) mit $n = 12$ die Abstände der Maxima und Minima (Tabelle 6) und überprüfen auf Äquidistanz. Wir entnehmen weiterhin dem Bild (Abb.22), das einer rechteckigen Spaltfunktion am nächsten kommt, die Bildbreite B in Pixel ($1 \text{ px} = 14.0 \mu\text{m}$). Sie entspricht der Beugungsspaltbreite D , die hier gemäß der optischen Abbildung (L_1) vergrößert erscheint. Die Vergrößerung ist aus Brennweite f der Linse und Gegenstandsweite g bestimmbar, letztere messen wir mit einem Zollstock, dessen Messgenauigkeit hierzu völlig ausreicht und vergleichen den hieraus bestimmten Wert mit dem weiter oben (Auswertung zu 1) gefundenen Wert.

3.4 Fourierbild des Doppelspaltes

Als nächstes schauen wir uns das Bild des Doppelspaltes mit der CCD-Kamera an und justieren wieder auf optimale Kantensteilheit der Spaltbilder und Symmetrie. Wir bestimmen Breite und Abstand (Tabelle 8) der beiden Einzelspalte auf dem Monitor (wie immer bisher in Pixel- Einheiten). In derselben Weise wie bei den Messungen am Einzelspalt können wir aus der aktuellen Gegenstandsweite und Brennweite die Vergrößerung bestimmen und auf die Abmessungen des Doppelspaltes rückschließen. Diese Werte werden später bei der Interpretation des gemessenen Beugungsbildes benötigt. Wir beobachten im Folgenden auch hier wieder die schon beim Einzelspalt untersuchten Beiträge der einzelnen Beugungsordnungen der Spalte, indem wir nun den Analyserspalt in der Fourierebene auf- und zudrehen. Für Fall 1 nehmen wir ein Bild (Abb.25) zur niedriger Beugungsordnung $n = 3$ auf. Als letztes drehen wir nun langsam den Analyserspalt zu und beobachten, wie die beiden rechteckigen Spaltprofile zu zwei gaußähnlichen Profilen verschwimmen (Fall 2: nur noch die 1. Beugungsmaxima beider Spalte tragen zur Abbildung bei), wie sich zunehmend auch der Zwischenraum aufhellt und schließlich die Doppelstruktur endgültig verschwindet und nur noch ein flaches Plateau (Fall 3: auch die dem Doppelspalt zuzuordnenden 1. Gittermaxima werden abgeschnitten) sichtbar bleibt. Wir notieren zu jedem Fall gehörenden Einstellungen des Analyserspalt (Tabelle 7) und nehmen die zugehörigen Bilder (Abb.25-27) auf.

4 Messergebnisse

Alle Messdaten wurden den Versuchsprotokollen (26. November & 3. Dezember, 2019) entnommen und in die Tabellen 1-8 übertragen.

Fehlende Daten im Protokoll wurden zuhause in den entsprechenden CCD-Dateien nachgeschaut und sind in den Tabellen als (*) markiert.

5 Auswertung

5.1 Quantitative Beobachtungen am Einfachspalt

Wir wollen zuerst die Lage der Maxima und Minima sowie die gemessenen Intensitätsverhältnisse der einzelnen Beugungsordnungen mit den theoretisch zu erwartenden Werten und tragen den Abstand (Tab.1) der Minima n -ter Ordnung gegen die Ordnungszahl n in Abbildung 5. Es ergibt sich eine Gerade ergeben, deren Steigung S_1 die Spaltweite des verwendeten Beugungsspalt D zu bestimmen gestattet, wenn wir die Bildweite b kennen. In dasselbe Diagramm tragen Sie nun die gemessenen Werte für die Lage der Beugungsmaxima ein. Um die Hypothese zu testen, dass die Maxima genau zwischen den entsprechenden Werten der Minima bei $n = 1.5, 2.5, \dots$ liegen führen wir eine lineare Fitfunktion durch alle Messwerte und ihre entsprechenden vermuteten Ordnungszahlen durch. Die Parameter

Tabelle 1: Messung der Intensitätsmaxima und -minima am Einzelspalt „B“

Messgröße	1. Messung		2. Messung		Messgröße	1. Messung		2. Messung	
	Position <i>x</i> [px]	Intensität <i>I</i> [counts]	Position <i>x</i> [px]	Intensität <i>I</i> [counts]		Position <i>x</i> [px]	Intensität <i>I</i> [counts]	Position <i>x</i> [px]	Intensität <i>I</i> [counts]
Rauschen, links	90 ± 5	60 ± 2	(*) 20 ± 5	(*) 59 ± 2	Rauschen, rechts	2032 ± 5	58 ± 2	(*) 2039 ± 5	(*) 63 ± 2
Maximum 5. Ordnung, links	/		429 ± 5	237 ± 2	Maximum 5. Ordnung, rechts	/		1252 ± 5	200 ± 2
Minimum 5. Ordnung, links	/		(*) 467 ± 5	(*) 124 ± 2	Minimum 5. Ordnung, rechts	/		(*) 1209 ± 5	(*) 140 ± 2
Maximum 4. Ordnung, links	[1] 512 ± 5	[1] 85 ± 2	502 ± 5	298 ± 2	Maximum 4. Ordnung, rechts	(*) 1182 ± 5	(*) 79 ± 2	1179 ± 5	256 ± 2
Minimum 4. Ordnung, links	(*) 547 ± 5	(*) 66 ± 2	(*) 547 ± 5	(*) 145 ± 2	Minimum 4. Ordnung, rechts	(*) 1141 ± 5	(*) 66 ± 2	(*) 1139 ± 5	(*) 153 ± 2
Maximum 3. Ordnung, links	(*) 588 ± 5	(*) 99 ± 2	588 ± 5	430 ± 2	Maximum 3. Ordnung, rechts	1099 ± 5	94 ± 2	1103 ± 5	405 ± 2
Minimum 3. Ordnung, links	(*) 615 ± 5	(*) 73 ± 2	(*) 615 ± 5	(*) 228 ± 2	Minimum 3. Ordnung, rechts	(*) 1063 ± 5	(*) 71 ± 2	(*) 1065 ± 5	(*) 205 ± 2
Maximum 2. Ordnung, links	662 ± 5	156 ± 2	664 ± 5	1007 ± 2	Maximum 2. Ordnung, rechts	1027 ± 5	130 ± 2	1027 ± 5	758 ± 2
Minimum 2. Ordnung, links	(*) 694 ± 5	(*) 104 ± 2	(*) 695 ± 5	(*) 515 ± 2	Minimum 2. Ordnung, rechts	(*) 993 ± 5	(*) 77 ± 2	(*) 995 ± 5	(*) 227 ± 2
Maximum 1. Ordnung, links	735 ± 5	289 ± 2	736 ± 5	2246 ± 2	Maximum 1. Ordnung, rechts	954 ± 5	244 ± 2	/	
Minimum 1. Ordnung, links	(*) 767 ± 5	(*) 97 ± 2	(*) 767 ± 5	(*) 451 ± 2	Minimum 1. Ordnung, rechts	(*) 921 ± 5	(*) 85 ± 2	/	
Maximum 0. Ordnung	841 ± 5	3891 ± 2	/		/				

[1] Wert war zuvor fälschlicherweise dem linken Maximum 3. Ordnung zugeordnet gewesen.

² Δx und ΔI grob abgeschätzt.

(*) Werte wurden in der entsprechenden CCD-Datei nachträglich nachgeschaut und ergänzt.

Tabelle 2: Eichung der Abszisse

sichtbare Maxima	halbe Spaltbreite $\frac{d+d_0}{2}$ [mm]	Abbildung Nr.
nur Maximum 0.Ordnung	0.19 ± 0.01	6
bis einschließlich Maxima 1.Ordnung	0.36 ± 0.01	7
bis einschließlich Maxima 2.Ordnung	0.60 ± 0.01	8
bis einschließlich Maxima 3.Ordnung	0.81 ± 0.01	9

¹ $\Delta \frac{d}{2}$ folgt aus der angegebenen Ablesegenauigkeit des Potentiometers.

² Beim komplett geschlossenen Spalt beträgt der Nullwert $\frac{d_0}{2} = -0.090(5) \text{ mm}$.

Tabelle 3: Messung der Intensitätsmaxima und -minima am Doppelspalt „B“

Messgröße	Position x [px]	Intensität I [counts]	Messgröße	Position x [px]	Intensität I [counts]
Rauschen, links	91 ± 2	62 ± 2	Rauschen, rechts	90 ± 1998	64 ± 2
Minimum 3. Ordnung, links	771 ± 2	70 ± 2	Minimum 3. Ordnung, rechts	922 ± 2	69 ± 2
Maximum 2. Ordnung, links	790 ± 2	281 ± 2	Maximum 2. Ordnung, rechts	903 ± 2	282 ± 2
Minimum 2. Ordnung, links	798 ± 2	103 ± 2	Minimum 2. Ordnung, rechts	894 ± 2	81 ± 2
Maximum 1. Ordnung, links	818 ± 2	2061 ± 2	Maximum 1. Ordnung, rechts	876 ± 2	2096 ± 2
Minimum 1. Ordnung, links	830 ± 2	140 ± 2	Minimum 1. Ordnung, rechts	863 ± 2	154 ± 2
Maximum 0. Ordnung	845 ± 2	3315 ± 2	/		

¹ Δx und ΔI grob abgeschätzt.

Tabelle 4: Fouriersynthese des Beugungsbildes am Einzelspalt

Einstellung Nr.	äußerstes Maximum in der Fourierebene	halbe Spaltbreite $\frac{d+d_0}{2}$ [mm]	Abbildung Nr.
1.	0. Ordnung	0.13 ± 0.01	16
2.	1. Ordnung	0.29 ± 0.01	17
3.	2. Ordnung	0.53 ± 0.01	18
4.	3. Ordnung	0.71 ± 0.01	19
5.	4. Ordnung	0.79 ± 0.01	20

¹ $\Delta \frac{d}{2}$ folgt aus der angegebenen Ablesegenauigkeit des Potentiometers.

² Beim komplett geschlossenen Spalt beträgt der Nullwert $\frac{d_0}{2} = -0.090(5) \text{ mm}$.

dieser Funktion beinhalten das Hintergrundrauschen h , die Amplitude der Intensität A und die Abszissenskalierung a . Wenn Chi-Quadrat einen Wert liefert der innerhalb unserer Messunsicherheiten liegt, dann bestätigen wir unsere Hypothese und können die Steigung der Fitfunktion als Wert für S_1 benutzen. Als nächstes wollen wir überprüfen ob die Intensitätsverhältnisse unserer zwei Messungen für niedrige (Abb.3) und hohe (Abb.4) Laserintensität unserem theoretischem Modell entsprechen. Bei der Messung hohe Intensität beschneiden wir die Messdaten, dort wo die Kamera überstrahlt, subtrahieren von beiden Messreihen das Rauschen und skalieren die Messwerte der zweiten Messung um einen Faktor, der der Mittelwert aller vergleichbaren markanten Punkte (Intensitätsmaxima und -minima, die in beiden Messungen bestimmt wurden) ist. Wir führen wieder eine Fitfunktion an den so kombinierten Messwerten durch. Anhand des Chi-Quadrat-Werts für diese Fitfunktion und einem qualitativen Vergleich der Fitfunktion und Messdaten in den Abbildungen 3 und 4 bewerten wir, ob die beobachteten Intensitätsverhältnisse vom unseren theoretischen Modell gut beschrieben werden.

Die Eichung der Abszissen erlaubt es uns zu bestimmen, wieviel Pixel auf der Kamera entsprechen 1 mm in der Fourierebene. So können wir den Abbildungsmaßstab β bestimmen und weil wir die Brennweite $f = 80 \text{ mm}$ der Linse L_2 kennen, die Bildweite b bestimmen. Diese verwenden wir um aus S_1 die Breite des Beugungsspaltes D auszurechnen.

5.1.1 Rechnung

Die absolute Amplitude A der Intensität interessiert uns in diesem Versuch nicht, nur die relativen Verhältnisse zueinander. Deswegen werden wir diesen Parameter in der Auswertung nicht wirklich weiter betrachten. Die normierte Intensitätsverteilung in der Fourierebene am Einzelspalt sieht so aus:

$$\hat{I}_1(x) = \left(\frac{\sin(\pi x)}{\pi x} \right)^2 = (\text{sinc}(x))^2 \quad (1)$$

und ist beispielhaft in Abbildung 11 bis zur 2. Beugungsordnung dargestellt. Wir passen nun diese Funktion an unsere Messdaten an:

$$I_1(x) = A \hat{I}_1(a(x - \mu)) + h \quad (2)$$

wobei die Abszissenskalierung so gewählt ist, dass $a x_1 = 1$ und x_1 ist die Position des 1. Hauptminimums bezogen zur Mitte. Für den Abbildungswinkel α_n gelten also folgende Beziehungen:

$$\sin(\alpha_n) = \frac{n \lambda}{D} = \frac{x_n}{b} = \frac{n x_1}{b} \quad (3)$$

also

$$D = \frac{\lambda b}{x_1} = \frac{\lambda}{x_1} f (1 + \beta) = \lambda a f (1 + \beta) \quad (4)$$

mit dem Abbildungsmaßstab β zwischen der Bildebene (Kamera) und der Fourierebene (Spaltblende)

$$\beta = \frac{S_1}{S_2} \quad (5)$$

$$\Delta\beta = \beta \sqrt{\left(\frac{\Delta S_1}{S_1} \right)^2 + \left(\frac{\Delta S_2}{S_2} \right)^2} \quad (6)$$

Für die Eichung der Abszisse bestimmen wir die Steigungen S_1 und S_2 über die Anpassung der Funktionen

$$x_n = S_1 n + \mu \quad (7)$$

$$d_n = S_2 n + d_0 \quad (8)$$

Wir bemerken, dass S_1 und a^{-1} den selben Zusammenhang beschreiben, deswegen können wir in (4) $S_1 = a^{-1}$ substituieren und erhalten

$$D = \lambda f \left(a + \frac{1}{S_2} \right) \quad (9)$$

$$\Delta D = D \sqrt{\left(\frac{\Delta \lambda}{\lambda} \right)^2 + \left(\frac{\Delta f}{f} \right)^2 + \frac{(\Delta a)^2 + \left(\frac{1}{S_2} \right)^2 \left(\frac{\Delta S_2}{S_2} \right)^2}{\left(a + \frac{1}{S_2} \right)^2}} \quad (10)$$

5.1.2 Source Code & Input

So sieht unsere Python-Implementierung aus:

Header:

```
1 %matplotlib inline
2 import matplotlib.pyplot as plt
3 import numpy as np
4 from decimal import Decimal
5 from scipy.stats import chi2
6 from scipy.integrate import quad
7
8 def format_e(n):
9     a = '%e' % Decimal(n)
10    return a.split('e')[0].rstrip('0').rstrip('.')+'e'+a.split('e')[1]
```

Berechnung der normierten Intensität des Fourierbildes am Doppelspalt nach (1):

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

Messwerte der vergleichbaren Extrema aus Tabelle 1:

```
1 x1_ext = np.array([90,512,547,588,615,662,694,735,767,993,1027,1063,1099,1141,1182,2032])
2 Fehler_x1_ext = np.full(x1_ext.size, 5)
3 I1_ext = np.array([60,85,66,99,73,156,104,289,97,77,130,71,94,66,79,58])
4 Fehler_I1_ext = np.full(I1_ext.size, 2)
5
6 x2_ext = np.array([20,502,547,588,615,664,695,736,767,995,1027,1065,1103,1139,1179,2039])
7 Fehler_x2_ext = np.full(x2_ext.size, 5)
8 I2_ext = np.array([59,298,145,430,228,1007,515,2246,451,227,758,205,405,153,256,63])
9 Fehler_I2_ext = np.full(I2_ext.size, 2)
```

Gesamte Messwerte aus aufgenommenen CCD-Dateien:

```
1 x1, I1 = np.loadtxt('data/233.1.1.ccd', skiprows=14, usecols=(0, 1), unpack=True)
2 x2, I2 = np.loadtxt('data/233.1.2.ccd', skiprows=14, usecols=(0, 1), unpack=True)
```

Hintergrundrauschen links und rechts wird gemittelt:

```
1 hintergrund1 = (I1_ext[0]+I1_ext[I1_ext.size-1])/2
2 hintergrund2 = (I2_ext[0]+I2_ext[I2_ext.size-1])/2
```

Messung hoher Intensität wird skaliert:

```
1 scale = np.mean((I1_ext[1:I1_ext.size-1]-hintergrund1)/(I2_ext[1:I2_ext.size-1]-hintergrund2))
2 I2_scaled = (I2-hintergrund2)*scale+hintergrund1
```

Messdaten beider Messungen werden kombiniert, wobei die zweite Messung dort beschnitten wo sie Übersättigt ist. Die genauen Schnittpunkte wurden von Hand so gewählt, dass sie ungefähr in der Mitte zwischen dem übersättigtem Bereich und dem nächsten Minimum liegen.

```
1 x_both = np.array([*x1, *x2[0:778], *x2[975:2048]])
2 I_both = np.array([*I1, *I2_scaled[0:778], *I2_scaled[975:2048]])
3 x_both, I_both = (list(t) for t in zip(*sorted(zip(x_both, I_both))))
```

Absoluter Fehler der skalierten Messung hoher Intensität ist zwar jetzt geringer als bei der Messung niedriger Intensität, wir runden jetzt die entsprechenden Messunsicherheiten einfachheitshalber auf, da in beiden Messungen dieser nur sehr grob abgeschätzt wurde (± 2 counts) und beim Vergleich mit dem Hauptmaximum der größere Fehler der ersten Messung überwiegen wird.

```
1 Fehler_x_both = np.full(len(x_both), 1)
2 Fehler_I_both = np.full(len(I_both), 2)
```

Fitfunktion (2) wird deklariert:

```
1 from scipy import odr
2
3 def fit_func(p, x):
4     (h, A, mu, a) = p
5     return A*beugung_spalt(a*(x-mu))+h
6
7 model = odr.Model(fit_func)
```

darzustellende Daten werden übergeben:

```

1 x = x_both
2 y = I_both
3 delta_x = Fehler_x_both
4 delta_y = Fehler_I_both

```

Startparameter für Ausgleichsrechnung werden gesetzt, sodass Lösung konvergiert. Der Python-Code wird gegebenenfalls mehrmals ausgeführt und die Parameter aus der vorherigen Ausführung als Startparameter verwendet, um eine bessere Anpassung zu erreichen.

```

1 try:
2     popt_iterate
3 except NameError:
4     para0 = [60, 3000, 845, 0.013]
5 else:
6     para0 = popt_iterate
7
8 data = odr.RealData(x, y, sx=delta_x, sy=delta_y)
9 odr = odr.ODR(data, model, beta0=para0)
10 out = odr.run()

```

Endgültige Ausgleichsparameter und ihre Kovarianzmatrix werden ausgelesen:

```

1 popt = out.beta
2 perr = out.sd_beta

```

Angabe welche Sigma-Umgebung der Fitfunktion im Diagramm dargestellt werden soll:

```

1 nstd = 2
2
3 popt_top = popt+nstd*perr
4 popt_bot = popt-nstd*perr

```

Plot-Umgebung wird angegeben:

```

1 x_fit = np.linspace(min(x)-(max(x)-min(x))/10, max(x)+(max(x)-min(x))/10, 1000)
2 fit = fit_func(popt, x_fit)
3 fit_top = fit_func(popt_top, x_fit)
4 fit_bot = fit_func(popt_bot, x_fit)
5
6 fit2 = (fit_func(popt, x_fit)-hintergrund1)/scale+hintergrund2
7 fit2_top = (fit_func(popt_top, x_fit)-hintergrund1)/scale+hintergrund2
8 fit2_bot = (fit_func(popt_bot, x_fit)-hintergrund1)/scale+hintergrund2

```

Ein Chi-Quadrat-Test wird durchgeführt unter Berücksichtigung von Δx und ΔI . D.h. es wird jeweils der senkrechte/orthogonale Abstand (Abb.2) der Messwerte zur Fitfunktion berechnet und normiert³. Die Summe der normierten Abstandsquadrate, der χ^2 -Wert, wird reduziert.

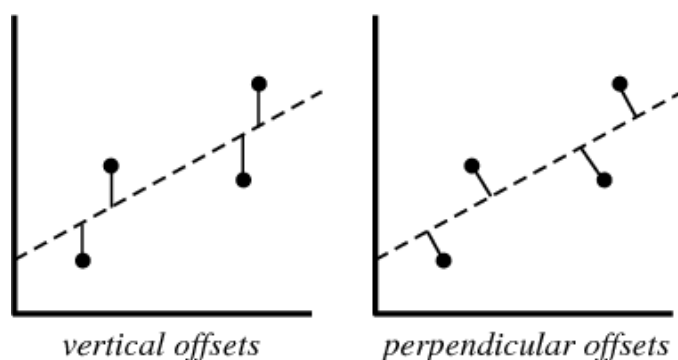


Abbildung 2: Abstände von Messdaten zur Fitfunktion⁴

```

1 chisquare = out.sum_square
2 dof=len(x)-popt.size
3 chisquare_red = chisquare/dof
4 prob = round(1-chi2.cdf(chisquare,dof),2)*100

```

³P. T. Boggs and J. E. Rogers, "Orthogonal Distance Regression," in "Statistical analysis of measurement error models and applications: proceedings of the AMS-IMS-SIAM joint summer research conference held June 10-16, 1989," Contemporary Mathematics, vol. 112, pg. 186, 1990.

⁴<http://mathworld.wolfram.com/LeastSquaresFitting.html>

Diagramm (Abb.3) wird erstellt:

```
1 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
2 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
3 plt.title('Beobachtungen am Einfachspalt, niedrige Laserintensität')
4 plt.grid(True)
5 plt.xlim(300, 1400)
6 plt.plot(x_fit, fit, 'g--', lw=2, label='Fit')
7 ax.fill_between(x_fit, fit_top, fit_bot, color='g', alpha=.25, label=str(nstd)+'$\sigma$-Umgebung
  Fit')
8 plt.xlabel('Position ' + r'${x}$' + ' ' + r'${[px]}$')
9 plt.ylabel('Intensität ' + r'${I}$' + ' ' + r'${[counts]}$')
10 plt.plot(x1, I1, 'r', lw=1, label='Messdaten')
11 plt.legend(loc='best')
12 plt.savefig('figures/233_Fig1.pdf', format='pdf', bbox_inches='tight')
```

Diagramm (Abb.4) wird erstellt:

```
1 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
2 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
3 plt.title('Beobachtungen am Einfachspalt, hohe Laserintensität')
4 plt.grid(True)
5 plt.xlim(300, 1400)
6 plt.ylim(-200,5000)
7 plt.plot(x_fit, fit2, 'g--', lw=2, label='Fit')
8 ax.fill_between(x_fit, fit2_top, fit2_bot, color='g', alpha=.25, label=str(nstd)+'$\sigma$-Umgebung
  Fit')
9 plt.xlabel('Position ' + r'${x}$' + ' ' + r'${[px]}$')
10 plt.ylabel('Intensität ' + r'${I}$' + ' ' + r'${[counts]}$')
11 plt.plot(x2, I2, 'r', lw=1, label='Messdaten')
12 plt.legend(loc='best')
13 plt.savefig('figures/233_Fig2.pdf', format='pdf', bbox_inches='tight')
```

Auslesen der Messergebnisse:

```
1 a = popt[3]
2 Fehler_a = perr[3]
3
4 h = popt[0]
5 Fehler_h = perr[0]
```

Ausgabe der Messergebnisse wird erstellt:

```
1 print('Abzissenskalierung: ')
2 print('a [1/px] =', format_e(a), ' +- ', format_e(Fehler_a))
3 print('Hintergrundrauschen: ')
4 print('h [counts] =', format_e(h), ' +- ', format_e(Fehler_h))
5 print('Chi-Quadrat =', chisquare)
6 print('Freiheitsgrade =', dof)
7 print('Chi-Quadrat reduziert =', chisquare_red)
8 print('Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten =', prob, '%')
9 print('\n')
10 print('Intensitätsskalierung: ')
11 print('k =', format_e(scale))
```

Anpassungsparameter werden gespeichert, um gegebenenfalls bei wiederholter Ausführung als Startparameter zu dienen.

```
1 popt_iterate = popt
```

Messwerte werden aus aufgenommenen CCD-Dateien geladen und Diagramme (Abb.6-9) werden erstellt:

```
1 x, I = np.loadtxt('data/233.1.3.ccd', skiprows=14, usecols=(0, 1), unpack=True)
2 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
3 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
4 plt.title('Beobachtungen am Einfachspalt, Verschwinden Maxima 1. Ordnung')
5 plt.grid(True)
6 plt.xlabel('Position ' + r'${x}$' + ' ' + r'${[px]}$')
7 plt.ylabel('Intensität ' + r'${I}$' + ' ' + r'${[counts]}$')
8 plt.plot(x, I, 'r', lw=1, label='Messdaten')
9 plt.legend(loc='best')
10 plt.savefig('figures/233_Fig3.pdf', format='pdf', bbox_inches='tight')
11
12 x, I = np.loadtxt('data/233.1.4.ccd', skiprows=14, usecols=(0, 1), unpack=True)
13 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
14 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
15 plt.title('Beobachtungen am Einfachspalt, Verschwinden Maxima 2. Ordnung')
```

```

16 plt.grid(True)
17 plt.xlabel('Position ' + r'${x}$' + ' ' + r'${[px]}$')
18 plt.ylabel('Intensiät ' + r'${I}$' + ' ' + r'${[counts]}$')
19 plt.plot(x, I, 'r', lw=1, label='Messdaten')
20 plt.legend(loc='best')
21 plt.savefig('figures/233_Fig4.pdf', format='pdf', bbox_inches='tight')
22
23 x, I = np.loadtxt('data/233.1.5.ccd', skiprows=14, usecols=(0, 1), unpack=True)
24 fig, ax = plt.subplots(1, figsize=[6.4 * 2, 4.8 * 1.5])
25 plt.ticklabel_format(axis='both', style='sci', scilimits=(0, 3), useMathText=True)
26 plt.title('Beobachtungen am Einfachspalt, Verschwinden Maxima 3. Ordnung')
27 plt.grid(True)
28 plt.xlabel('Position ' + r'${x}$' + ' ' + r'${[px]}$')
29 plt.ylabel('Intensiät ' + r'${I}$' + ' ' + r'${[counts]}$')
30 plt.plot(x, I, 'r', lw=1, label='Messdaten')
31 plt.legend(loc='best')
32 plt.savefig('figures/233_Fig5.pdf', format='pdf', bbox_inches='tight')
33
34 x, I = np.loadtxt('data/233.1.6.ccd', skiprows=14, usecols=(0, 1), unpack=True)
35 fig, ax = plt.subplots(1, figsize=[6.4 * 2, 4.8 * 1.5])
36 plt.ticklabel_format(axis='both', style='sci', scilimits=(0, 3), useMathText=True)
37 plt.title('Beobachtungen am Einfachspalt, Verschwinden Maxima 4. Ordnung')
38 plt.grid(True)
39 plt.xlabel('Position ' + r'${x}$' + ' ' + r'${[px]}$')
40 plt.ylabel('Intensiät ' + r'${I}$' + ' ' + r'${[counts]}$')
41 plt.plot(x, I, 'r', lw=1, label='Messdaten')
42 plt.legend(loc='best')
43 plt.savefig('figures/233_Fig6.pdf', format='pdf', bbox_inches='tight')

```

Position und in Hypothese angenommene Ordnungsnummern der Extrema aus Tabelle 1:

```

1 x_ord1 = np.array([512, 547, 588, 615, 662, 694, 735, 767, 841, 921, 954, 993, 1027, 1063, 1099, 1141, 1182])
2 Fehler_x_ord1 = np.full(x_ord1.size, 5)
3 n_ord1 = np.array([-4.5, -4, -3.5, -3, -2.5, -2, -1.5, -1, 0, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5])
4
5 x_ord2 = np.array([429, 467, 502, 547, 588, 615, 664, 695, 736, 767, 995, 1027, 1065, 1103, 1139, 1179, 1209, 1252])
6 Fehler_x_ord2 = np.full(x_ord2.size, 5)
7 n_ord2 = np.array([-5.5, -5, -4.5, -4, -3.5, -3, -2.5, -2, -1.5, -1, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5])

```

darzustellende Daten werden übergeben:

```

1 x = np.concatenate((n_ord1, n_ord2))
2 y = np.concatenate((x_ord1, x_ord2))
3 Fehler_y = np.concatenate((Fehler_x_ord1, Fehler_x_ord2))

```

Fitfunktion (7) wird deklariert:

```

1 from scipy.optimize import curve_fit
2
3 def fit_func(x, s, c):
4     return s*x+c

```

Endgültige Ausgleichungsparameter und ihre Kovarianzmatrix werden ausgelesen:

```

1 popt, pcov = curve_fit(fit_func, x, y)

```

Angabe welche Sigma-Umgebung der Fitfunktion im Diagramm dargestellt werden soll:

```

1 nstd = 2
2
3 popt_top = popt + nstd * np.diag(pcov)
4 popt_bot = popt - nstd * np.diag(pcov)

```

Plot-Umgebung wird angegeben:

```

1 x_fit = np.linspace(min(x) - (max(x) - min(x)) / 10, max(x) + (max(x) - min(x)) / 10, 1000)
2 fit = fit_func(x_fit, *popt)
3 fit_top = fit_func(x_fit, *popt_top)
4 fit_bot = fit_func(x_fit, *popt_bot)

```

Diagramm (Abb.5) wird erstellt:

```

1 fig, ax = plt.subplots(1, figsize=[6.4 * 2, 4.8 * 1.5])
2 plt.ticklabel_format(axis='both', style='sci', scilimits=(0, 3), useMathText=True)
3 plt.plot(x_fit, fit, 'r', lw=1, label='Fit')
4 plt.errorbar(x, y, yerr=Fehler_y, lw=1, ecolor='k', fmt='none', capsize=2, label='Messdaten')

```

```

5 ax.fill_between(x_fit, fit_top, fit_bot, color='C3', alpha=.25, label=str(nstd)+r'$\sigma$'+'-
  Umgebung')
6 plt.title('Abstand der Beugungsmaxima und -minima am Einzelspalt B (Bildebene)')
7 plt.grid(True)
8 plt.xlabel('Ordnungszahl n')
9 plt.ylabel('Position im Bild x [px]')
10 plt.legend(loc='best')
11 plt.savefig('figures/233_Fig23.pdf', format='pdf', bbox_inches='tight')

```

Ein Chi-Quadrat-Test wird durchgeführt:

```

1 dof = x.size-popt.size
2 chisquare = np.sum((fit_func(x,*popt)-y)**2/Fehler_y**2)
3 chisquare_red = chisquare/dof
4 prob = round(1-chi2.cdf(chisquare,dof),2)*100

```

Auslesen der Messergebnisse:

```

1 S_1 = popt[0]
2 Fehler_S_1 = pcov[0][0]

```

Ausgabe der Messergebnisse wird erstellt:

```

1 print('Abstand pro Ordnungszahl in der Bildebene:')
2 print('S_1 [px / 1] =', format_e(S_1), ' +- ', format_e(Fehler_S_1))
3 print('Chi-Quadrat =', chisquare)
4 print('Freiheitsgrade =', dof)
5 print('Chi-Quadrat reduziert =', chisquare_red)
6 print('Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten =', prob, '%')

```

Messwerte aus Tabelle 2:

```

1 d_ord = np.array([0.19,0.36,0.60,0.81])+0.090
2 Fehler_d_ord = np.full(d_ord.size, 0.011)
3 n_ord = np.array([1,2,3,4])

```

Fitfunktion (8) wird deklariert:

```

1 from scipy.optimize import curve_fit
2
3 def fit_func(x,s,c):
4     return s*x+c

```

darzustellende Daten werden übergeben:

```

1 x = n_ord
2 y = d_ord
3 Fehler_y = Fehler_d_ord

```

Endgültige Ausgleichungsparameter und ihre Kovarianzmatrix werden ausgelesen:

```

1 popt, pcov = curve_fit(fit_func, x, y)

```

Angabe welche Sigma-Umgebung der Fitfunktion im Diagramm dargestellt werden soll:

```

1 nstd = 2
2
3 popt_top = popt+nstd*np.diag(pcov)
4 popt_bot = popt-nstd*np.diag(pcov)

```

Plot-Umgebung wird angegeben:

```

1 x_fit = np.linspace(min(x)-(max(x)-min(x))/10, max(x)+(max(x)-min(x))/10, 1000)
2 fit = fit_func(x_fit, *popt)
3 fit_top = fit_func(x_fit, *popt_top)
4 fit_bot = fit_func(x_fit, *popt_bot)

```

Diagramm (Abb.10) wird erstellt:

```

1 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
2 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
3 plt.plot(x_fit, fit, 'r', lw=1, label='Fit')
4 plt.errorbar(x, y, yerr=Fehler_y, lw=1, ecolor='k', fmt='none', capsize=2, label='Messdaten')
5 ax.fill_between(x_fit, fit_top, fit_bot, color='C3', alpha=.25, label=str(nstd)+r'$\sigma$'+'-
  Umgebung')
6 plt.title('Abstand der Beugungsminima am Einzelspalt B (Fourierebene)')
7 plt.grid(True)

```

```

8 plt.xlabel('Ordnungszahl n')
9 plt.ylabel('Position der Blende d [mm]')
10 plt.legend(loc='best')
11 plt.savefig('figures/233_Fig24.pdf', format='pdf', bbox_inches='tight')

```

Ein Chi-Quadrat-Test wird durchgeführt:

```

1 dof = x.size-popt.size
2 chisquare = np.sum((fit_func(x,*popt)-y)**2/Fehler_y**2)
3 chisquare_red = chisquare/dof
4 prob = round(1-chi2.cdf(chisquare,dof),2)*100

```

Auslesen der Messergebnisse:

```

1 S_2 = popt[0]
2 Fehler_S_2 = pcov[0][0]
3 d_0 = popt[1]
4 Fehler_d_0 = pcov[1][1]
5
6 lamda = 635 *1e-06
7 Fehler_lamda = 1 *1e-06
8 f = 80
9 Fehler_f = 1
10 b_px = 0.0140
11 Fehler_b_px = 0.0002

```

Berechnung von β und $\Delta\beta$ nach (5) bzw. (6):

```

1 beta_px = S_1/S_2
2 Fehler_beta_px = beta_px*np.sqrt((Fehler_S_1/S_1)**2+(Fehler_S_2/S_2)**2)
3
4 beta = beta_px*b_px
5 Fehler_beta = beta*np.sqrt((Fehler_beta_px/beta_px)**2+(Fehler_b_px/b_px)**2)

```

Berechnung von D und ΔD nach (9) bzw. (10):

```

1 D = lamda*f*((a/b_px)+1/S_2)
2 Fehler_D = D*np.sqrt((Fehler_lamda/lamda)**2+(Fehler_f/f)**2
3               +(((a/b_px)**2)*((Fehler_a/a)**2+(Fehler_b_px/b_px)**2)+(1/S_2)**2*(Fehler_S_2/
4               S_2)**2)
               /(((a/b_px)+1/S_2)**2))

```

Ausgabe der Messergebnisse wird erstellt:

```

1 print('Abstand pro Ordnungszahl in der Fourierebene:')
2 print('S_2 [mm / 1] =', format_e(S_2), ' +- ', format_e(Fehler_S_2))
3 print('Nullwert:')
4 print('d_0_korr [mm] =', format_e(d_0), ' +- ', format_e(Fehler_d_0))
5 print('Chi-Quadrat =', chisquare)
6 print('Freiheitsgrade =', dof)
7 print('Chi-Quadrat reduziert =', chisquare_red)
8 print('Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten =', prob, '%')
9 print('\n')
10 print('Abbildungsmaßstab:')
11 print('beta [px / mm] =', format_e(beta_px), ' +- ', format_e(Fehler_beta_px))
12 print('beta [1] =', format_e(beta), ' +- ', format_e(Fehler_beta))
13 print('\n')
14 print('Beugungsspaltbreite:')
15 print('D [mm] =', format_e(D), ' +- ', format_e(Fehler_D))

```

5.1.3 Output

```

1 Abstand pro Ordnungszahl in der Bildebene:
2 S_1 [px / 1] = 7.428511e+01 +- 3.380067e-02
3 Chi-Quadrat = 17.448390867580066
4 Freiheitsgrade = 33
5 Chi-Quadrat reduziert = 0.5287391171993959
6 Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten = 99.0 %

```

```

1 Abszissenskalierung:
2 a [1/px] = 1.337325e-02 +- 7.494427e-06
3 Hintergrundrauschen:
4 h [counts] = 6.070584e+01 +- 1.091836e-01
5 Chi-Quadrat = 32832.64225691346

```

```

6  Freiheitsgrade = 3895
7  Chi-Quadrat reduziert = 8.429433185343635
8  Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten = 0.0 %
9
10 Intensitätsskalierung:
11 k = 9.740385e-02

```

```

1  Abstand pro Ordnungszahl in der Fourierebene:
2  S_2 [mm / 1] = 2.1e-01 +- 9e-05
3  Nullwert:
4  d_0_korr [mm] = 5.5e-02 +- 6.75e-04
5  Chi-Quadrat = 7.4380165289256235
6  Freiheitsgrade = 2
7  Chi-Quadrat reduziert = 3.7190082644628117
8  Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten = 2.0 %
9
10 Abbildungsmaßstab:
11 beta [px / mm] = 3.537386e+02 +- 2.211107e-01
12 beta [1] = 4.952341e+00 +- 7.081542e-02
13
14 Beugungsspaltbreite:
15 D [mm] = 2.904305e-01 +- 3.725709e-03

```

Daraus folgen die Ergebnisse dieses Versuches. Der Abstand der Minima und Maxima pro Ordnungszahl in der Bildebene (Kamera) beträgt

$$S_1 = 74.285(34) \text{ px}$$

mit einem entsprechenden Ergebnis auf den Anpassungstest:

$$\chi_{red}^2 = 0.53$$

Die Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten ist

$$P \approx 99.0\%$$

Die Abszissenskalierung und das Hintergrundrauschen betragen

$$a = 1.33733(75) \text{ px}^{-1}$$

$$h = 60.71(11) \text{ counts}$$

mit einem entsprechenden Ergebnis auf den Anpassungstest:

$$\chi_{red}^2 = 8.4$$

Die Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten ist

$$P \approx 0.0\%$$

Das gemessene Intensitätsverhältnis zwischen den beiden Messungen beträgt

$$k \approx 0.097$$

Der Abstand der Minima und Maxima pro Ordnungszahl in der Fourierebene (Spatblende) und der Abstandswert bei dem die Blende kein Licht mehr durchlässt betragen

$$S_2 = 2.1000(9) \times 10^{-1} \text{ mm}$$

$$d_0 = 5.500(68) \times 10^{-2} \text{ mm}$$

mit einem entsprechenden Ergebnis auf den Anpassungstest:

$$\chi_{red}^2 = 3.7$$

Die Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten ist

$$P \approx 2.0\%$$

Der Abbildungsmaßstab von unserem optischen Aufbau und die Beugungsspaltbreite betragen

$$\begin{aligned}\beta &= 353.74(22) \text{ px mm}^{-1} \\ &= 4.952(70) \\ D &= 2.904(37) \times 10^{-1} \text{ mm}\end{aligned}$$

Die Abbildungen 3 bis 10 sind im Anhang zu finden.

5.2 Beugungsstruktur des Doppelspaltes

Für jeden der drei Doppelspalte „A“, „B“ und „C“ führen wir an die Messdaten eine Fitfunktion gemäß unserem theoretischem Modell durch und bekommen dadurch einen angepassten Wert für das Verhältnis $v = \frac{g_D}{D}$ zwischen der Breite der Beugungsspalte D und ihrem Abstand g_D an jedem Doppelspalt und die Abszissenskalierung $a = \sin(\alpha) b$. Anhand der Chi-Quadrat-Werte für diese Fitfunktionen und einem qualitativen Vergleich der jeweiligen Fitfunktion und Messdaten in den Abbildungen 12 bis 14 bewerten wir, ob die beobachteten Intensitätsverhältnisse vom unseren theoretischen Modell gut beschrieben werden.

5.2.1 Rechnung

Die normierte Intensitätsverteilung in der Fourierebene am Doppelspalt sieht so aus:

$$\hat{I}(x) = \left(\frac{\sin(\pi x)}{\pi x} \cos(\pi x) \right)^2 = (\text{sinc}(x) \cos(\pi x))^2 \quad (11)$$

und ist beispielhaft in Abbildung 11 bis zur 2. Beugungsordnung und $v = \frac{g_D}{D} = 3$ dargestellt. Wir passen nun diese Funktion an unsere Messdaten an:

$$I_2(x) = A \hat{I}_2(a(x - \mu)) + h \quad (12)$$

Alle anderen Zusammenhänge bleiben genau wie beim Einzelspalt erhalten.

5.2.2 Source Code & Input

So sieht die Fortführung unserer Python-Implementierung aus:

Berechnung der normierten Intensität des Fourierbildes am Doppelspalt nach (11):

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

Diagramm (Abb.11) wird erstellt:

```
1 a=2
2 v=3
3 x=np.linspace(-(a+1),a+1,200)
4 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
5 plt.plot(x,beugung_spalt(x),label='Spalt')
6 plt.plot(x,beugung_doppelspalt(x, v),label='Doppelspalt')
7 plt.xlabel('$\pi x$')
8 plt.ylabel('Intensität I [a.u.]')
9 plt.title('Beugung am Spalt und Doppelspalt (g/d=2)')
10 plt.ylim((0,1.1))
11 plt.legend(loc='best')
12 plt.savefig("figures/233_Fig20.pdf", format="pdf")
```

Messwerte aus aufgenommener CCD-Datei:

```
1 x, I = np.loadtxt('data/233.2.2.ccd',skiprows=14,usecols=(0, 1), unpack=True)
2
3 Fehler_x = np.full(len(x), 1)
4 Fehler_I = np.full(len(I), 2)
```

Fitfunktion (12) wird deklariert:

```
1 from scipy import odr
2
3 def fit_func(p, x):
4     (h, A, mu, a, v) = p
5     return A*beugung_doppelspalt(a*(x-mu), v)+h
6
7 model = odr.Model(fit_func)
```


darzustellende Daten werden übergeben:

```
1 x = x
2 y = I
3 delta_x = Fehler_x
4 delta_y = Fehler_I
```

Startparameter für Ausgleichsrechnung werden gesetzt, sodass Lösung konvergiert. Der Python-Code wird gegebenenfalls mehrmals ausgeführt und die Parameter aus der vorherigen Ausführung als Startparameter verwendet, um eine bessere Anpassung zu erreichen.

```
1 try:
2     popt_iterateA
3 except NameError:
4     para0 = [60, 750, 845, 0.006, 5.4]
5 else:
6     para0 = popt_iterateA
7
8 data = odr.RealData(x, y, sx=delta_x, sy=delta_y)
9 odr = odr.ODR(data, model, beta0=para0)
10 out = odr.run()
```

Endgültige Ausgleichungsparameter und ihre Kovarianzmatrix werden ausgelesen:

```
1 popt = out.beta
2 perr = out.sd_beta
```

Angabe welche Sigma-Umgebung der Fitfunktion im Diagramm dargestellt werden soll:

```
1 nstd = 2
2
3 popt_top = popt+nstd*perr
4 popt_bot = popt-nstd*perr
```

Plot-Umgebung wird angegeben:

```
1 x_fit = np.linspace(min(x)-(max(x)-min(x))/10, max(x)+(max(x)-min(x))/10, 1000)
2 fit = fit_func(popt, x_fit)
3 fit_top = fit_func(popt_top, x_fit)
4 fit_bot = fit_func(popt_bot, x_fit)
5 envelope = popt[1]*beugung_spalt(popt[3]*(x_fit-popt[2]))+popt[0]
```

Ein Chi-Quadrat-Test wird durchgeführt unter Berücksichtigung von Δx und ΔI . D.h. es wird jeweils der senkrechte/orthogonale Abstand (Abb.2) der Messwerte zur Fitfunktion berechnet und normiert⁵. Die Summe der normierten Abstandsquadrate, der χ^2 -Wert, wird reduziert.

```
1 chisquare = out.sum_square
2 dof=len(x)-popt.size
3 chisquare_red = chisquare/dof
4 prob = round(1-chi2.cdf(chisquare,dof),2)*100
```

Diagramm (Abb.12) wird erstellt:

```
1 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
2 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
3 plt.title('Beugungsstruktur des Doppelspalt es A')
4 plt.xlim(300, 1400)
5 plt.plot(x_fit, fit, 'g--', lw=2, label='Fit')
6 ax.fill_between(x_fit, fit_top, fit_bot, color='g', alpha=.25, label=str(nstd)+'$\sigma$-Umgebung Fit')
7 plt.plot(x_fit, envelope, 'b:', lw=1, label='Einhüllende des Fits')
8 plt.xlabel('Position '+r'${x}$'+r'${[px]}$')
9 plt.ylabel('Intensität '+r'${I}$'+r'${[counts]}$')
10 plt.plot(x, I, 'r', lw=1, label='Messdaten')
11 plt.legend(loc='best')
12 plt.savefig('figures/233_Fig8.pdf', format='pdf', bbox_inches='tight')
```

Auslesen der Messergebnisse:

```
1 a = popt[3]
2 Fehler_a = perr[3]
3
```

⁵P. T. Boggs and J. E. Rogers, "Orthogonal Distance Regression," in "Statistical analysis of measurement error models and applications: proceedings of the AMS-IMS-SIAM joint summer research conference held June 10-16, 1989," Contemporary Mathematics, vol. 112, pg. 186, 1990.

```

4 h = popt[0]
5 Fehler_h = perr[0]
6
7 v = popt[4]
8 Fehler_v = perr[4]

```

Berechnung von D und ΔD nach (9) bzw. (10):

```

1 D = lamda*f*((a/b_px)+1/S_2)
2 Fehler_D = D*np.sqrt((Fehler_lamda/lamda)**2+(Fehler_f/f)**2
3             +(((a/b_px)**2)*((Fehler_a/a)**2+(Fehler_b_px/b_px)**2)+(1/S_2)**2*(Fehler_S_2/
4             S_2)**2)
              /(((a/b_px)+1/S_2)**2))

```

Ausgabe der Messergebnisse wird erstellt:

```

1 print('Doppelspalt A: ')
2 print('Abszissenskalierung: ')
3 print('a =', format_e(a), ' +- ', format_e(Fehler_a))
4 print('Hintergrundrauschen: ')
5 print('h =', format_e(h), ' +- ', format_e(Fehler_h))
6 print('Spaltverhältnis: ')
7 print('v =', format_e(v), ' +- ', format_e(Fehler_v))
8 print('Chi-Quadrat =', chisquare)
9 print('Freiheitsgrade =', dof)
10 print('Chi-Quadrat reduziert =', chisquare_red)
11 print('Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten =', prob, '%')
12 print('\n')
13 print('Beugungsspaltbreite:')
14 print('D [mm] =', format_e(D), ' +- ', format_e(Fehler_D))

```

Anpassungsparameter werden gespeichert, um gegebenenfalls bei wiederholter Ausführung als Startparameter zu dienen.

```

1 popt_iterateA = popt

```

Messwerte aus aufgenommener CCD-Datei:

```

1 x, I = np.loadtxt('data/233.2.1.ccd', skiprows=14, usecols=(0, 1), unpack=True)
2
3 Fehler_x = np.full(len(x), 1)
4 Fehler_I = np.full(len(I), 2)

```

darzustellende Daten werden übergeben:

```

1 x = x
2 y = I
3 delta_x = Fehler_x
4 delta_y = Fehler_I

```

Startparameter für Ausgleichungsrechnung werden gesetzt, sodass Lösung konvergiert. Der Python-Code wird gegebenenfalls mehrmals ausgeführt und die Parameter aus der vorherigen Ausführung als Startparameter verwendet, um eine bessere Anpassung zu erreichen.

```

1 try:
2     popt_iterateB
3 except NameError:
4     para0 = [60, 3000, 845, 0.013, 2.5]
5 else:
6     para0 = popt_iterateB
7
8 data = odr.RealData(x, y, sx=delta_x, sy=delta_y)
9 odr = odr.ODR(data, model, beta0=para0)
10 out = odr.run()

```

Endgültige Ausgleichungsparameter und ihre Kovarianzmatrix werden ausgelesen:

```

1 popt = out.beta
2 perr = out.sd_beta

```

Angabe welche Sigma-Umgebung der Fitfunktion im Diagramm dargestellt werden soll:

```

1 nstd = 2
2
3 popt_top = popt+nstd*perr
4 popt_bot = popt-nstd*perr

```

Plot-Umgebung wird angegeben:

```

1 x_fit = np.linspace(min(x)-(max(x)-min(x))/10, max(x)+(max(x)-min(x))/10, 1000)
2 fit = fit_func(popt, x_fit)
3 fit_top = fit_func(popt_top, x_fit)
4 fit_bot = fit_func(popt_bot, x_fit)
5 envelope = popt[1]*beugung_spalt(popt[3]*(x_fit-popt[2]))+popt[0]

```

Ein Chi-Quadrat-Test wird durchgeführt unter Berücksichtigung von Δx und ΔI . D.h. es wird jeweils der senkrechte/orthogonale Abstand (Abb.2) der Messwerte zur Fitfunktion berechnet und normiert⁶. Die Summe der normierten Abstandsquadrate, der χ^2 -Wert, wird reduziert.

```

1
2 chisquare = out.sum_square
3 dof=len(x)-popt.size
4 chisquare_red = chisquare/dof
5 prob = round(1-chi2.cdf(chisquare,dof),2)*100

```

Diagramm (Abb.13) wird erstellt:

```

1 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
2 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
3 plt.title('Beugungsstruktur des Doppelspaltes B')
4 plt.xlim(500, 1200)
5 plt.plot(x_fit, fit, 'g--', lw=2, label='Fit')
6 ax.fill_between(x_fit, fit_top, fit_bot, color='g', alpha=.25, label=str(nstd)+'$\sigma$-Umgebung Fit')
7 plt.plot(x_fit, envelope, 'b:', lw=1, label='Einhüllende des Fits')
8 plt.xlabel('Position '+r'${x}$'+ ' '+r'${[px]}$')
9 plt.ylabel('Intensität '+r'${I}$'+ ' '+r'${[counts]}$')
10 plt.plot(x, I, 'r', lw=1, label='Messdaten')
11 plt.legend(loc='best')
12 plt.savefig('figures/233_Fig7.pdf', format='pdf', bbox_inches='tight')

```

Auslesen der Messergebnisse:

```

1 a = popt[3]
2 Fehler_a = perr[3]
3
4 h = popt[0]
5 Fehler_h = perr[0]
6
7 v = popt[4]
8 Fehler_v = perr[4]

```

Berechnung von D und ΔD nach (9) bzw. (10):

```

1 D = lamda*f*((a/b_px)+1/S_2)
2 Fehler_D = D*np.sqrt((Fehler_lamda/lamda)**2+(Fehler_f/f)**2
3 +(((a/b_px)**2)*((Fehler_a/a)**2+(Fehler_b_px/b_px)**2)+(1/S_2)**2*(Fehler_S_2/
4 /(((a/b_px)+1/S_2)**2)))

```

Ausgabe der Messergebnisse wird erstellt:

```

1 print('Doppelspalt B: ')
2 print('Abziskskalierung: ')
3 print('a =', format_e(a), ' +- ', format_e(Fehler_a))
4 print('Hintergrundrauschen: ')
5 print('h =', format_e(h), ' +- ', format_e(Fehler_h))
6 print('Spaltverhältnis: ')
7 print('v =', format_e(v), ' +- ', format_e(Fehler_v))
8 print('Chi-Quadrat =', chisquare)
9 print('Freiheitsgrade =', dof)
10 print('Chi-Quadrat reduziert =', chisquare_red)
11 print('Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten =', prob, '%')
12 print('\n')
13 print('Beugungsspaltbreite:')
14 print('D [mm] =', format_e(D), ' +- ', format_e(Fehler_D))

```

Anpassungsparameter werden gespeichert, um gegebenenfalls bei wiederholter Ausführung als Startparameter zu dienen.

⁶P. T. Boggs and J. E. Rogers, "Orthogonal Distance Regression," in "Statistical analysis of measurement error models and applications: proceedings of the AMS-IMS-SIAM joint summer research conference held June 10-16, 1989," Contemporary Mathematics, vol. 112, pg. 186, 1990.

```
1 popt_iterateB = popt
```

Messwerte aus aufgenommener CCD-Datei:

```
1 x, I = np.loadtxt('data/233.2.3.ccd', skiprows=14, usecols=(0, 1), unpack=True)
2
3 Fehler_x = np.full(len(x), 1)
4 Fehler_I = np.full(len(I), 2)
```

darzustellende Daten werden übergeben:

```
1 x = x
2 y = I
3 delta_x = Fehler_x
4 delta_y = Fehler_I
```

Startparameter für Ausgleichsrechnung werden gesetzt, sodass Lösung konvergiert. Der Python-Code wird gegebenenfalls mehrmals ausgeführt und die Parameter aus der vorherigen Ausführung als Startparameter verwendet, um eine bessere Anpassung zu erreichen.

```
1 try:
2     popt_iterateC
3 except NameError:
4     para0 = [60, 3500, 845, 0.013, 5]
5 else:
6     para0 = popt_iterateC
7
8 data = odr.RealData(x, y, sx=delta_x, sy=delta_y)
9 odr = odr.ODR(data, model, beta0=para0)
10 out = odr.run()
```

Endgültige Ausgleichsparameter und ihre Kovarianzmatrix werden ausgelesen:

```
1 popt = out.beta
2 perr = out.sd_beta
```

Angabe welche Sigma-Umgebung der Fitfunktion im Diagramm dargestellt werden soll:

```
1 nstd = 2
2
3 popt_top = popt+nstd*perr
4 popt_bot = popt-nstd*perr
```

Plot-Umgebung wird angegeben:

```
1 x_fit = np.linspace(min(x)-(max(x)-min(x))/10, max(x)+(max(x)-min(x))/10, 1000)
2 fit = fit_func(popt, x_fit)
3 fit_top = fit_func(popt_top, x_fit)
4 fit_bot = fit_func(popt_bot, x_fit)
5 envelope = popt[1]*beugung_spalt(popt[3]*(x_fit-popt[2]))+popt[0]
```

Ein Chi-Quadrat-Test wird durchgeführt unter Berücksichtigung von Δx und ΔI . D.h. es wird jeweils der senkrechte/orthogonale Abstand (Abb.2) der Messwerte zur Fitfunktion berechnet und normiert⁷. Die Summe der normierten Abstandsquadrate, der χ^2 -Wert, wird reduziert.

```
1 chisquare = out.sum_square
2 dof=len(x)-popt.size
3 chisquare_red = chisquare/dof
4 prob = round(1-chi2.cdf(chisquare,dof),2)*100
```

Diagramm (Abb.14) wird erstellt:

```
1 fig, ax = plt.subplots(1, figsize=[6.4 * 2, 4.8 * 1.5])
2 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
3 plt.title('Beugungsstruktur des Doppelspaltes C')
4 plt.xlim(500, 1200)
5 plt.plot(x_fit, fit, 'g--', lw=2, label='Fit')
6 ax.fill_between(x_fit, fit_top, fit_bot, color='g', alpha=.25, label=str(nstd)+'$\sigma$-Umgebung Fit')
7 plt.plot(x_fit, envelope, 'b:', lw=1, label='Einhüllende des Fits')
8 plt.xlabel('Position '+r'$\{x\}$'+r'$\{px\}$')
```

⁷P. T. Boggs and J. E. Rogers, "Orthogonal Distance Regression," in "Statistical analysis of measurement error models and applications: proceedings of the AMS-IMS-SIAM joint summer research conference held June 10-16, 1989," Contemporary Mathematics, vol. 112, pg. 186, 1990.

```

9 plt.ylabel('Intensität '+r'${I}$'+ ' '+r'${[counts]}$')
10 plt.plot(x, I, 'r', lw=1, label='Messdaten')
11 plt.legend(loc='best')
12 plt.savefig('figures/233_Fig9.pdf', format='pdf', bbox_inches='tight')

```

Auslesen der Messergebnisse:

```

1 a = popt[3]
2 Fehler_a = perr[3]
3
4 h = popt[0]
5 Fehler_h = perr[0]
6
7 v = popt[4]
8 Fehler_v = perr[4]

```

Berechnung von D und ΔD nach (9) bzw. (10):

```

1 D = lamda*f*((a/b_px)+1/S_2)
2 Fehler_D = D*np.sqrt(((Fehler_lamda/lamda)**2+(Fehler_f/f)**2
3               +(((a/b_px)**2)*((Fehler_a/a)**2+(Fehler_b_px/b_px)**2)+(1/S_2)**2*(Fehler_S_2/
4               S_2)**2))
               /(((a/b_px)+1/S_2)**2))

```

Ausgabe der Messergebnisse wird erstellt:

```

1 print('Doppelspalt C: ')
2 print('Abszissenskalierung: ')
3 print('a [1/px] =', format_e(a), ' +- ', format_e(Fehler_a))
4 print('Hintergrundrauschen: ')
5 print('h [counts] =', format_e(h), ' +- ', format_e(Fehler_h))
6 print('Spaltverhältnis: ')
7 print('v =', format_e(v), ' +- ', format_e(Fehler_v))
8 print('Chi-Quadrat =', chisquare)
9 print('Freiheitsgrade =', dof)
10 print('Chi-Quadrat reduziert =', chisquare_red)
11 print('Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten =', prob, '%')
12 print('\n')
13 print('Beugungsspaltbreite:')
14 print('D [mm] =', format_e(D), ' +- ', format_e(Fehler_D))

```

Anpassungsparameter werden gespeichert, um gegebenenfalls bei wiederholter Ausführung als Startparameter zu dienen.

```

1 popt_iterateC = popt

```

5.2.3 Output

```

1 Doppelspalt A:
2 Abszissenskalierung:
3 a [1/px] = 6.145755e-03 +- 1.286667e-05
4 Hintergrundrauschen:
5 h [counts] = 5.962962e+01 +- 9.645322e-02
6 Spaltverhältniss:
7 v = 5.156191e+00 +- 1.09885e-02
8 Chi-Quadrat = 7642.971826943387
9 Freiheitsgrade = 2043
10 Chi-Quadrat reduziert = 3.74105326820528
11 Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten = 0.0 %
12
13 Beugungsspaltbreite:
14 D [mm] = 2.64206e-01 +- 3.345824e-03

```

```

1 Doppelspalt B:
2 Abszissenskalierung:
3 a [1/px] = 1.337763e-02 +- 1.464026e-05
4 Hintergrundrauschen:
5 h [counts] = 5.937967e+01 +- 1.173951e-01
6 Spaltverhältniss:
7 v = 2.401008e+00 +- 2.705396e-03
8 Chi-Quadrat = 10909.731082876011
9 Freiheitsgrade = 2043
10 Chi-Quadrat reduziert = 5.340054372430744
11 Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten = 0.0 %

```

```

12 Beugungsspaltbreite:
13 D [mm] = 2.904227e-01 +- 3.72586e-03

```

```

1 Doppelspalt C:
2 Abszissenskalierung:
3 a [1/px] = 1.416697e-02 +- 1.234072e-05
4 Hintergrundrauschen:
5 h [counts] = 5.991066e+01 +- 9.829382e-02
6 Spaltverhältnis:
7 v = 4.561324e+00 +- 4.017359e-03
8 Chi-Quadrat = 6505.1842484059
9 Freiheitsgrade = 2043
10 Chi-Quadrat reduziert = 3.1841332591316203
11 Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten = 0.0 %
12 Beugungsspaltbreite:
13 D [mm] = 2.932925e-01 +- 3.769044e-03

```

Die Ergebnisse dieses Versuches sind in Tabelle 9 dargestellt.

Die Abbildungen 11 bis 14 sind im Anhang zu finden.

5.3 Das Objektbild als Fouriersynthese des Beugungsbildes

Wir verwenden die aus der Messung bei weit offener Blendenöffnung ($n=25$) abgeschätzte Breite $B = 131(3) \text{ px}$ des Bildes an den Flanken, Lage $\mu = 1027(3) \text{ px}$ um eine Näherung des erwarteten Beugungsbildes in den Abbildungen 16-23 abzubilden, das Hintergrundrauschen wurde hierfür aus vorherigen Messungen als $h = 60(1) \text{ counts}$ abgeschätzt und die Intensität für alle Näherungen an dem Maximum 0. Ordnung der ersten Messung (Abb.16) normiert.

Wir vergleichen danach die Zahl und Lage der „Wiggels“ (der Maxima) und die Intensitäten der Maxima und Minima der ersten drei Bildkurven, normiert auf das Maximum des zur 0. Beugungsordnung gehörenden modifizierten Spaltbildes, also die von uns gezeichneten Näherungen in den Abbildungen. Zum Schluss überlegen wir uns warum bei einem fast geschlossenem Analyserspalt (d.h. nur die 0. Ordnung wird durchgelassen) die Intensität in der Bildmitte höher ist, als wenn dieser weit geöffnet ist.

5.3.1 Rechnung

Um ein theoretisches Bild des Einzelspaltes zu erhalten muss folgendes Integral berechnet werden:

$$\hat{I}_1(x) = \frac{D}{\pi} \int_0^{\frac{2n\pi}{D}} \frac{\sin(kD/2)}{kD/2} \cos(kx) dk \quad (13)$$

und ist beispielhaft in Abbildung 15 bis zur Beugungsordnung $n = 3$ dargestellt. Wir nähern nun diese Funktion an unsere Messdaten an:

$$I_1(x) = A\hat{I}_1(x - \mu) + h \quad (14)$$

Die Parameter μ und h sind uns bekannt, A wird an dem Peak in Abbildung 16 normiert und für D setzen wir die Bildbreite B ein, weil wir für x auch nur die Abstände x in der Bild- und nicht der Objektebene betrachten.

Wir kennen die Bildweite $B = 131(3) \text{ px}$, die Pixelweite $1 \text{ px} = 14.0(2) \text{ }\mu\text{m}$, die Brennweite $f = 80(1) \text{ mm}$ und die Gegenstandsweite $g = 98(2) \text{ mm}$. Daraus folgt:

$$D = \frac{B(1 \text{ px})g}{b} = B(1 \text{ px})g \left(\frac{1}{f} - \frac{1}{g} \right) = B(1 \text{ px}) \frac{g - f}{f} = 0.42(5) \text{ mm} \quad (15)$$

, was eine 2.6-Sigma Differenz zum Wert $D = 0.2904(37) \text{ mm}$ für Einzelspalt „B“ aus der ersten Aufgabe darstellt.

5.3.2 Source Code & Input

So sieht die Fortführung unserer Python-Implementierung aus:

Funktion innerhalb des Integrals nach (13) wird deklariert:

```

1 def spalt(k):
2     return d/np.pi*np.sin(k*d/2)/(k*d/2)*np.cos(y*k)

```

Diagramm (Abb.15) wird erstellt:

```

1 d=1 #Spaltbreite auf 1 setzen
2 n=3 #Integrationsbereich
3 Y=np.linspace(-1,1,200)*d
4
5 f_modifiziert=[]
6 for i in range(len(Y)):
7     y=Y[i]
8     result, error = quad(spalt, 0, 2*np.pi*n/d)
9     f_modifiziert.append(result**2)
10
11 f_modifiziert=f_modifiziert/np.max(f_modifiziert)
12
13 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
14 plt.plot(Y,f_modifiziert, label='theoretischer Intesitätsverlauf')
15 plt.xlabel('y / d')
16 plt.ylabel('Intensität I [a.u.]')
17 plt.title('modifiziertes Objektbild am Einzelspalt (n=3)')
18 plt.ylim((0,1.1))
19 plt.legend(loc='best')
20 plt.savefig("figures/233_Fig21.pdf", format="pdf")

```

Parameter der Näherung:

```

1 mu = 1027
2 h = 60
3 d=131

```

Messwerte werden aus aufgenommenen CCD-Dateien geladen, Näherung nach (14) berechnet und Diagramme (Abb.16-23) werden erstellt:

```

1 x, I = np.loadtxt('data/233.3.1.ccd', skiprows=14, usecols=(0, 1), unpack=True)
2 n=1 #Integrationsbereich
3
4 f_modifiziert=[]
5 for i in range(len(x)):
6     y=x[i]-mu
7     result, error = quad(spalt, 0, 2*np.pi*n/d)
8     f_modifiziert.append(result**2)
9 A = np.max(I)/np.max(f_modifiziert)
10 f_modifiziert=A*np.array(f_modifiziert)+h
11 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
12 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
13 plt.title('Beugungsbild des Einzelspalts B bei nur 1 verbleibenden Beugungsordnung')
14 plt.grid(True)
15 plt.plot(x,f_modifiziert, 'b--', lw=1, label='theoretische Annäherung \nnormiert auf Peak, \nn=1')
16 plt.xlim(900, 1150)
17 plt.xlabel('Position '+r'${x}$'+ ' '+r'${[px]}$')
18 plt.ylabel('Intensiät '+r'${I}$'+ ' '+r'${[counts]}$')
19 plt.plot(x, I, 'r', lw=1, label='Messdaten')
20 plt.legend(loc='best')
21 plt.savefig('figures/233_Fig10.pdf', format='pdf', bbox_inches='tight')
22
23 x, I = np.loadtxt('data/233.3.2.ccd', skiprows=14, usecols=(0, 1), unpack=True)
24 n=2 #Integrationsbereich
25 f_modifiziert=[]
26 for i in range(len(x)):
27     y=x[i]-mu
28     result, error = quad(spalt, 0, 2*np.pi*n/d)
29     f_modifiziert.append(result**2)
30 f_modifiziert=A*np.array(f_modifiziert)+h
31 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
32 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
33 plt.title('Beugungsbild des Einzelspalts B bei nur 2 verbleibende Beugungsordnungen')
34 plt.grid(True)
35 plt.plot(x,f_modifiziert, 'b--', lw=1, label='theoretische Annäherung n=2\nmit gleicher Normierung wie in Abb. für n=1.')
36 plt.xlim(900, 1150)
37 plt.xlabel('Position '+r'${x}$'+ ' '+r'${[px]}$')
38 plt.ylabel('Intensiät '+r'${I}$'+ ' '+r'${[counts]}$')
39 plt.plot(x, I, 'r', lw=1, label='Messdaten')
40 plt.legend(loc='best')
41 plt.savefig('figures/233_Fig11.pdf', format='pdf', bbox_inches='tight')
42
43 x, I = np.loadtxt('data/233.3.3.ccd', skiprows=14, usecols=(0, 1), unpack=True)
44 n=3 #Integrationsbereich
45 f_modifiziert=[]

```

```

46 for i in range(len(x)):
47     y=x[i]-mu
48     result, error = quad(spalt, 0, 2*np.pi*n/d)
49     f_modifiziert.append(result**2)
50 f_modifiziert=A*np.array(f_modifiziert)+h
51 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
52 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
53 plt.title('Beugungsbild des Einzelspalts B bei nur 3 verbleibende Beugungsordnungen')
54 plt.grid(True)
55 plt.plot(x,f_modifiziert, 'b--', lw=1, label='theoretische Annäherung n=3\nmit gleicher Normierung
    wie in Abb. für n=1.')
56 plt.xlim(900, 1150)
57 plt.xlabel('Position ' +r'${x}$'+ ' +r'${[px]}$')
58 plt.ylabel('Intensität ' +r'${I}$'+ ' +r'${[counts]}$')
59 plt.plot(x, I, 'r', lw=1, label='Messdaten')
60 plt.legend(loc='best')
61 plt.savefig('figures/233_Fig12.pdf', format='pdf', bbox_inches='tight')
62
63 x, I = np.loadtxt('data/233.3.4.ccd',skiprows=14,usecols=(0, 1), unpack=True)
64 n=4 #Integrationsbereich
65 f_modifiziert=[]
66 for i in range(len(x)):
67     y=x[i]-mu
68     result, error = quad(spalt, 0, 2*np.pi*n/d)
69     f_modifiziert.append(result**2)
70 f_modifiziert=A*np.array(f_modifiziert)+h
71 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
72 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
73 plt.title('Beugungsbild des Einzelspalts B bei nur 4 verbleibende Beugungsordnungen')
74 plt.grid(True)
75 plt.plot(x,f_modifiziert, 'b--', lw=1, label='theoretische Annäherung n=4\nmit gleicher Normierung
    wie in Abb. für n=1.')
76 plt.xlim(900, 1150)
77 plt.xlabel('Position ' +r'${x}$'+ ' +r'${[px]}$')
78 plt.ylabel('Intensität ' +r'${I}$'+ ' +r'${[counts]}$')
79 plt.plot(x, I, 'r', lw=1, label='Messdaten')
80 plt.legend(loc='best')
81 plt.savefig('figures/233_Fig13.pdf', format='pdf', bbox_inches='tight')
82
83 x, I = np.loadtxt('data/233.3.5.ccd',skiprows=14,usecols=(0, 1), unpack=True)
84 n=5 #Integrationsbereich
85 f_modifiziert=[]
86 for i in range(len(x)):
87     y=x[i]-mu
88     result, error = quad(spalt, 0, 2*np.pi*n/d)
89     f_modifiziert.append(result**2)
90 f_modifiziert=A*np.array(f_modifiziert)+h
91 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
92 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
93 plt.title('Beugungsbild des Einzelspalts B bei nur 5 verbleibende Beugungsordnungen')
94 plt.grid(True)
95 plt.plot(x,f_modifiziert, 'b--', lw=1, label='theoretische Annäherung n=5\nmit gleicher Normierung
    wie in Abb. für n=1.')
96 plt.xlim(900, 1150)
97 plt.xlabel('Position ' +r'${x}$'+ ' +r'${[px]}$')
98 plt.ylabel('Intensität ' +r'${I}$'+ ' +r'${[counts]}$')
99 plt.plot(x, I, 'r', lw=1, label='Messdaten')
100 plt.legend(loc='best')
101 plt.savefig('figures/233_Fig14.pdf', format='pdf', bbox_inches='tight')
102
103 x, I = np.loadtxt('data/233.3.6 (alles offen).ccd',skiprows=14,usecols=(0, 1), unpack=True)
104 n=25 #Integrationsbereich
105 f_modifiziert=[]
106 for i in range(len(x)):
107     y=x[i]-mu
108     result, error = quad(spalt, 0, 2*np.pi*n/d)
109     f_modifiziert.append(result**2)
110 f_modifiziert=A*np.array(f_modifiziert)+h
111 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
112 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
113 plt.title('Beugungsbild des Einzelspalts B bei weit geöffneten Spaltblende (n=25)')
114 plt.grid(True)
115 plt.plot(x,f_modifiziert, 'b--', lw=1, label='theoretische Annäherung n=25\nmit gleicher Normierung
    wie in Abb. für n=1.')
116 plt.xlim(900, 1150)
117 plt.xlabel('Position ' +r'${x}$'+ ' +r'${[px]}$')

```



```

118 plt.ylabel('Intensität '+r'${I}$'+ ' '+r'${[counts]}$')
119 plt.plot(x, I, 'r', lw=1, label='Messdaten')
120 plt.legend(loc='best')
121 plt.savefig('figures/233_Fig15.pdf', format='pdf', bbox_inches='tight')
122
123 x, I = np.loadtxt('data/233.3.7.ccd', skiprows=14, usecols=(0, 1), unpack=True)
124 n=12 #Integrationsbereich
125 f_modifiziert=[]
126 for i in range(len(x)):
127     y=x[i]-mu
128     result, error = quad(spalt, 0, 2*np.pi*n/d)
129     f_modifiziert.append(result**2)
130 f_modifiziert=A*np.array(f_modifiziert)+h
131 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
132 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
133 plt.title('Beugungsbild des Einzelspalts B bei 12 Beugungsordnungen')
134 plt.grid(True)
135 plt.plot(x, f_modifiziert, 'b--', lw=1, label='theoretische Annäherung n=12\nmit gleicher Normierung\nwie in Abb. für n=1.')
136 plt.xlim(900, 1150)
137 plt.xlabel('Position '+r'${x}$'+ ' '+r'${[px]}$')
138 plt.ylabel('Intensität '+r'${I}$'+ ' '+r'${[counts]}$')
139 plt.plot(x, I, 'r', lw=1, label='Messdaten')
140 plt.legend(loc='best')
141 plt.savefig('figures/233_Fig16.pdf', format='pdf', bbox_inches='tight')

```

5.3.3 Output

```

1 Abstand pro Ordnungszahl in der Bildebene:
2 S_3 [px / 1] = 1.118577e+01 +- 1.80183e-03
3 Chi-Quadrat = 2.393280632411085
4 Freiheitsgrade = 21
5 Chi-Quadrat reduziert = 0.11396574440052787
6 Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten = 100.0 %

```

Daraus folgt das Ergebnis dieses Versuches, dass der durchschnittliche Abstand der Minima und Maxima

$$S_3 = 11.19(18) \text{ px}$$

mit einem entsprechenden Ergebnis auf den Anpassungstest

$$\chi_{red}^2 = 0.11$$

beträgt. Die Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten ist

$$P \approx 100.0\%$$

Die Abbildungen 15 bis 23 sind im Anhang zu finden.

5.4 Fourierbild des Doppelspaltes

Wir verwenden die aus der Messung bei weit offener Blendenöffnung (Tab8) gemessene Breite $B = 132(2) \text{ px}$ der Teilbilder an den Flanken, ihr Abstand $g_B = 307(2) \text{ px}$ und ihre Lage $\mu = 1026(2) \text{ px}$ um eine Näherung des erwarteten Beugungsbildes in den Abbildungen 25-27 abzubilden, das Hintergrundrauschen wurde hierfür aus vorherigen Messungen als $h = 60(1) \text{ counts}$ abgeschätzt und die Intensität für alle Näherungen an dem Peak der ersten der Messung (Abb.25) normiert.

Wir vergleichen danach die Zahl und Lage der „Wiggels“ (der Maxima) und die Intensitäten der Maxima und Minima der drei Bildkurven. Die Ordnungszahl n , ab der wir den Doppelspalt von einem Einzelspalt nicht mehr unterscheidet, soll für die letzte Messung (Abb.27) theoretisch hergeleitet werden und die so erwartete Intensität (Näherung) mit der tatsächlich gemessenen qualitativ verglichen werden.

5.4.1 Rechnung

Um ein theoretisches Bild des Doppelspaltes zu erhalten muss folgendes Integral berechnet werden:

$$\hat{I}_2(x) = \frac{D}{\pi} \int_0^{\frac{2n\pi}{D}} \cos(kg_D) \frac{\sin(kD/2)}{kD/2} \cos(kx) dk \quad (16)$$

und ist beispielhaft für $g_D = 2D$ in Abbildung 24 bis zur Beugungsordnung $n = 3$ dargestellt. Wir nähern nun diese Funktion an unsere Messdaten an:

$$I_2(x) = A\hat{I}_2(x - \mu) + h \quad (17)$$

Die Parameter μ und h sind uns bekannt, A wird an dem Peak in Abbildung 25 normiert und für D setzen wir die Bildbreite B ein (selbes für g_D und g_B), weil wir für x auch nur die Abstände x in der Bild- und nicht der Objektebene betrachten.

Wir wollen bestimmen, wann wir im Objektbild die 2 Spalte nicht mehr von einander unterscheiden können. Dafür betrachten wir die 2 Zusammenhänge für die ersten Haupt- und Nebenminima von \hat{I} :

$$k_{1,Haupt} = \frac{\pi}{kD/2} \quad (18)$$

und

$$k_{1,Neben} = \frac{\pi/2}{kg_D} \quad (19)$$

weil in diesen Fällen der Term unter dem Integral in (16) Null wird und somit die Ableitungen von \hat{I} und I nach x auch verschwinden. Es folgt also

$$n_{kr} = \frac{k_{1,Neben}}{k_{1,Haupt}} = \frac{D}{gD} = v^{-1} \quad (20)$$

für die kritische Ordnungszahl n_{kr} . Wenn man also nur diesen Anteil der ersten Beugungsordnung in der Fourierebene durchlässt und den Rest abschneidet, so können wir am Objektbild den Doppelspalt von einem Einzelspalt nicht mehr unterscheiden.

Wir kennen die Bildweite $B = 132(2) \text{ px}$, die Pixelweite $1 \text{ px} = 14.0(2) \mu\text{m}$, die Brennweite $f = 80(1) \text{ mm}$ und die Gegenstandsweite $g = 98(2) \text{ mm}$. Daraus folgt:

$$D = \frac{B(1 \text{ px})g}{b} = B(1 \text{ px})g \left(\frac{1}{f} - \frac{1}{g} \right) = B(1 \text{ px}) \frac{g-f}{f} = 0.42(5) \text{ mm} \quad (21)$$

, was eine 2.6-Sigma Differenz zum Wert für Doppelspalt „B“ in Tabelle 9 darstellt.

5.4.2 Source Code & Input

So sieht die Fortführung unserer Python-Implementierung aus:

Funktion innerhalb des Integrals nach (16) wird deklariert:

```
1 def doppelspalt(k):
2     return d/np.pi*np.cos(k*g/2)*np.sin(k*d/2)/(k*d/2)*np.cos(y*k)
```

Diagramm (Abb.24) wird erstellt:

```
1 d=1
2 n=3
3 g=2*d
4 Y=np.linspace(-1,1,200)*3*d
5
6 f_modifiziert=[]
7 for i in range(len(Y)):
8     y=Y[i]
9     result, error = quad(doppelspalt, 0, 2*np.pi*n/d)
10    f_modifiziert.append(result**2)
11
12 f_modifiziert=f_modifiziert/np.max(f_modifiziert)
13
14 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
15 plt.plot(Y,f_modifiziert, label='theoretischer Intesitätsverlauf')
16 plt.xlabel('y / d')
17 plt.ylabel('Intensität I [a.u.]')
18 plt.title('modifiziertes Objektbild am Doppelspalt (n=3, g/d=2)')
19 plt.ylim((0,1.1))
20 plt.legend()
21 plt.savefig("figures/233_Fig22.pdf", format="pdf")
```

Parameter der Näherung:

```

1 mu = 1026
2 h = 6
3 d=132
4 Fehler_d = 2
5 g = 307
6 Fehler_g = 2

```

Messwerte werden aus aufgenommenen CCD-Dateien geladen, Näherung nach (17) berechnet und Diagramme (Abb.25-27) werden erstellt:

```

1 n=3
2 x, I = np.loadtxt('data/233.4.1.ccd', skiprows=14, usecols=(0, 1), unpack=True)
3 f_modifiziert=[]
4 for i in range(len(x)):
5     y=x[i]-mu
6     result, error = quad(doppelspalt, 0, 2*np.pi*n/d)
7     f_modifiziert.append(result**2)
8 A=np.max(I)/np.max(f_modifiziert)
9 f_modifiziert=A*np.array(f_modifiziert)+h
10 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
11 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
12 plt.title('Beugungsbild des Doppelspalts B im 1. Fall ')
13 plt.grid(True)
14 plt.plot(x,f_modifiziert, 'b--', lw=1, label='theoretische Annäherung \nnormiert auf Peak, \nn=3')
15 plt.xlabel('Position '+r'${x}$'+ ' '+r'${[px]}$')
16 plt.ylabel('Intensiät '+r'${I}$'+ ' '+r'${[counts]}$')
17 plt.plot(x, I, 'r', lw=1, label='Messdaten')
18 plt.legend(loc='best')
19 plt.savefig('figures/233_Fig17.pdf', format='pdf', bbox_inches='tight')
20
21 n=1 #Integrationsbereich
22 x, I = np.loadtxt('data/233.4.2.ccd', skiprows=14, usecols=(0, 1), unpack=True)
23 f_modifiziert=[]
24 for i in range(len(x)):
25     y=x[i]-mu
26     result, error = quad(doppelspalt, 0, 2*np.pi*n/d)
27     f_modifiziert.append(result**2)
28 f_modifiziert=A*np.array(f_modifiziert)+h
29 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
30 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
31 plt.title('Beugungsbild des Doppelspalts B im 2. Fall ')
32 plt.grid(True)
33 plt.plot(x,f_modifiziert, 'b--', lw=1, label='theoretische Annäherung n=1\nmit gleicher Normierung wie in Abb. für n=3.')
34 plt.xlabel('Position '+r'${x}$'+ ' '+r'${[px]}$')
35 plt.ylabel('Intensiät '+r'${I}$'+ ' '+r'${[counts]}$')
36 plt.plot(x, I, 'r', lw=1, label='Messdaten')
37 plt.legend(loc='best')
38 plt.savefig('figures/233_Fig18.pdf', format='pdf', bbox_inches='tight')
39
40 n=d/g/2 #Integrationsbereich
41 Fehler_n = n*np.sqrt((Fehler_d/d)**2+(Fehler_g/g)**2)
42 print('kritische Ordnungszahl: ')
43 print('n =', format_e(n), ' +- ', format_e(Fehler_n))
44 x, I = np.loadtxt('data/233.4.3.ccd', skiprows=14, usecols=(0, 1), unpack=True)
45 f_modifiziert=[]
46 for i in range(len(x)):
47     y=x[i]-mu
48     result, error = quad(doppelspalt, 0, 2*np.pi*n/d)
49     f_modifiziert.append(result**2)
50 f_modifiziert=A*np.array(f_modifiziert)+h
51 fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *1.5])
52 plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
53 plt.title('Beugungsbild des Doppelspalts B im 3. Fall ')
54 plt.grid(True)
55 plt.plot(x,f_modifiziert, 'b--', lw=1, label='theoretische Annäherung n=d/2g='+str(round(n,2))+'\nmit gleicher Normierung wie in Abb. für n=1.')
56 plt.xlabel('Position '+r'${x}$'+ ' '+r'${[px]}$')
57 plt.ylabel('Intensiät '+r'${I}$'+ ' '+r'${[counts]}$')
58 plt.plot(x, I, 'r', lw=1, label='Messdaten')
59 plt.legend(loc='best')
60 plt.savefig('figures/233_Fig19.pdf', format='pdf', bbox_inches='tight')

```

5.4.3 Output

```

1 kritische Ordnungszahl:
2 n = 2.149837e-01 +- 3.545662e-03

```

Daraus folgt das Ergebnis dieses Versuches, dass die kritische Ordnungszahl in diesem Versuch

$$n_{kr} = 2.15(4)$$

beträgt.

Die Abbildungen 24 bis 27 sind im Anhang zu finden.

6 Fazit

Insgesamt sind wir mit der Ausführung des Versuches zufrieden. Qualitativ stimmen alle unsere Näherungen und Anpassungsfunktionen sehr gut mit den entsprechenden Messdaten überein. Bei den Messungen, wo wir die Extremstellen selber Abschätzen sollten, hatten wir den Fehler Δx wahrscheinlich überschätzt, da die entsprechenden Chi-Quadrat-Werte, bei der Bestimmung von S_1 und S_2 sehr niedrig ausgefallen sind. Bei den Fitfunktionen an die Messdaten vielen die Chi-Quadrat-Werte dafür zu groß aus, weil wir in unserem Modell nicht für Absorptionsunterschiede und Asymmetrie in unserem optischen Aufbau korrigiert haben. Das ist deutlich zu erkennen an Abbildungen 3, 12, 14, 16, 22 und 23, wo in der Bildmitte die Intensität der Messdaten immer runter geht. Die Asymmetrie ist auch in allen Messdaten bemerkbar. In Abbildung 4 ist sie jedoch durch das Überstrahlen der Kamera verstärkt worden, was größtenteils durch das Abschneiden des betroffenen Bereiches behoben wurde.

Die 2.6-Sigma-Differenzen der bestimmten Werte der Beugungsspaltbreiten D für den Einzelspalt „B“ und den Doppelspalt „B“ in (15) und (21) kommen höchstwahrscheinlich aus einem systematischen Fehler, bei der Vermessung der Gegenstandsweite g . D.h. wir haben wahrscheinlich an der Objekt- und/oder Linsenebene falsch angesetzt und ein solcher systematischer Fehler von mehreren Millimetern führt zu einem sehr großen relativen Fehler im Term $\frac{g-f}{f}$, und somit auch D , weil $g - f \approx 18 \text{ mm}$. Deswegen wäre es besser gewesen, wenn wir (wie im Skript ursprünglich angegeben) die Bildweite b direkt gemessen hätten. Der absolute Fehler Δb wäre zwar mit dem Zollstock größer gewesen als Δg an der Skala. Der ausschlaggebende relative Fehler $\frac{\Delta D}{D}$ wäre aber niedriger. Bei der Versuchsdurchführung fiel uns das jedoch nicht auf und wir hielten die Messung der Gegenstandsweite für gut genug.

7 Anhang

Hier sind Abbildungen 3 bis 27, sowie alle übrige Tabellen angehängt.

Tabelle 5: Intensitätsverhältnisse der Beugungsbilder

Intensität I [<i>counts</i>] der Messgröße	Einstellung Nr.		
	1.	2.	3.
Rauschen, links	61 ± 2	61 ± 2	61 ± 2
1. Maximum von links	3516 ± 2	3240 ± 2	3394 ± 2
1. Minimum von links	2264 ± 2	2250 ± 2	/
2. Maximum von links	3520 ± 2	3022 ± 2	/
2. Minimum von links	2601 ± 2	/	
3. Maximum von links	3550 ± 2	/	
Rauschen, rechts	61 ± 2	63 ± 2	62 ± 2

¹ ΔI grob abgeschätzt.

Abbildung 3

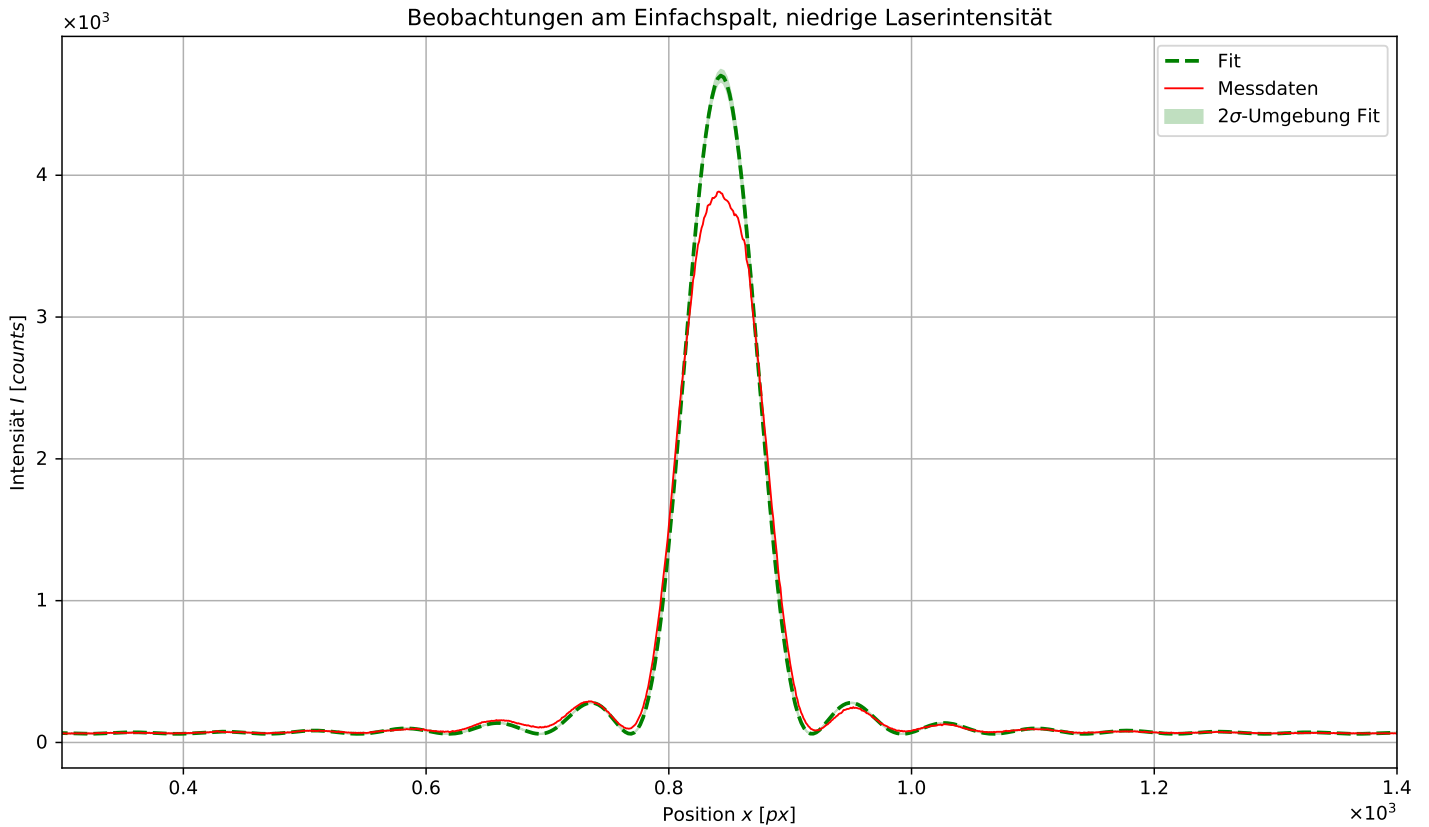


Tabelle 6: Abstände der Beugungsbildmaxima und -minima

Messgröße	Position x [px]	Messgröße	Position x [px]
1. Maximum von links	965 ± 2	1. Minimum von links	970 ± 2
2. Maximum von links	975 ± 2	2. Minimum von links	982 ± 2
3. Maximum von links	987 ± 2	3. Minimum von links	992 ± 2
4. Maximum von links	998 ± 2	4. Minimum von links	1004 ± 2
5. Maximum von links	1007 ± 2	5. Minimum von links	1014 ± 2
6. Maximum von links	1021 ± 2	6. Minimum von links	1025 ± 2
7. Maximum von links	1032 ± 2	7. Minimum von links	1037 ± 2
8. Maximum von links	1042 ± 2	8. Minimum von links	1048 ± 2
9. Maximum von links	1054 ± 2	9. Minimum von links	1059 ± 2
10. Maximum von links	1065 ± 2	10. Minimum von links	1071 ± 2
11. Maximum von links	1076 ± 2	11. Minimum von links	1082 ± 2
12. Maximum von links	1088 ± 2	/	

¹ Δx grob abgeschätzt.

² halbe Spaltbreite $\frac{d+d_0}{2} = 2.27(1) \text{ mm}$.

³ Beim komplett geschlossenen Spalt beträgt der Nullwert $\frac{d_0}{2} = -0.090(5) \text{ mm}$.

⁴ Breite des Bildes an den Flanken $B = 131(3) \text{ px}$.

⁵ Mitte des Bildes (*) $\mu = 1027(3) \text{ px}$.

⁶ Gegenstandsweite $g = 98(2) \text{ mm}$. Fehler aufgrund doppelter Ablesung (zur Bildung der Differenz) und Skaleneinteilung (1 mm).

⁷ Brennweite der Linse $f = 80(1) \text{ mm}$. Fehler abgeschätzt.

(*) Werte wurden in der entsprechenden CCD-Datei nachträglich nachgeschaut und ergänzt.

Tabelle 7: Fouriersynthese des Beugungsbildes am Doppelspalt

Fall		halbe Spaltbreite	Abbildung
Nr.	Beschreibung	$\frac{d+d_0}{2}$ [mm]	Nr.
1.	3 Beugungsmaxima beider Spalte sichtbar	0.52 ± 0.01	25
2.	Nur 1 Beugungsmaximum pro Spalt sichtbar	0.13 ± 0.01	26
3.	Nur noch ein einziges Plateau sichtbar	-0.03 ± 0.01	27

¹ $\Delta \frac{d}{2}$ folgt aus der angegebenen Ablesegenauigkeit des Potentiometers.

² Beim komplett geschlossenen Spalt beträgt der Nullwert $\frac{d_0}{2} = -0.090(5)$ mm.

Tabelle 8: Abstände der Flanken am Doppelspalt

Position x [px]			
linkes Plateau		rechtes Plateau	
linke Flanke	rechte Flanke	linke Flanke	rechte Flanke
808 ± 2	938 ± 2	1113 ± 2	1247 ± 2

¹ Δx grob abgeschätzt.

Tabelle 9: Ergebnisse der Messung der Beugungsstrukturen am Doppelspalt

Doppelspalt	A	B	C
Abszissenskalierung a [px ⁻¹]	6.146×10^{-3} $\pm 0.013 \times 10^{-3}$	13.378×10^{-3} $\pm 0.015 \times 10^{-3}$	14.167×10^{-3} $\pm 0.012 \times 10^{-3}$
Hintergrundrauschen h [counts]	59.63 ± 0.10	59.38 ± 0.12	59.91 ± 0.10
Spaltverhältnis $v = \frac{gD}{D}$	5.156 ± 0.011	2.401 ± 0.003	4.561 ± 0.004
Chi-Quadrat reduziert χ_{red}^2	3.7	5.3	3.2
Wahrscheinlichkeit P ein größeres oder gleiches Chi-Quadrat zu erhalten	0.0%	0.0%	0.0%
Beugungsspaltbreite D [mm]	0.2642 ± 0.0033	0.2904 ± 0.0037	0.2933 ± 0.0038

Abbildung 4

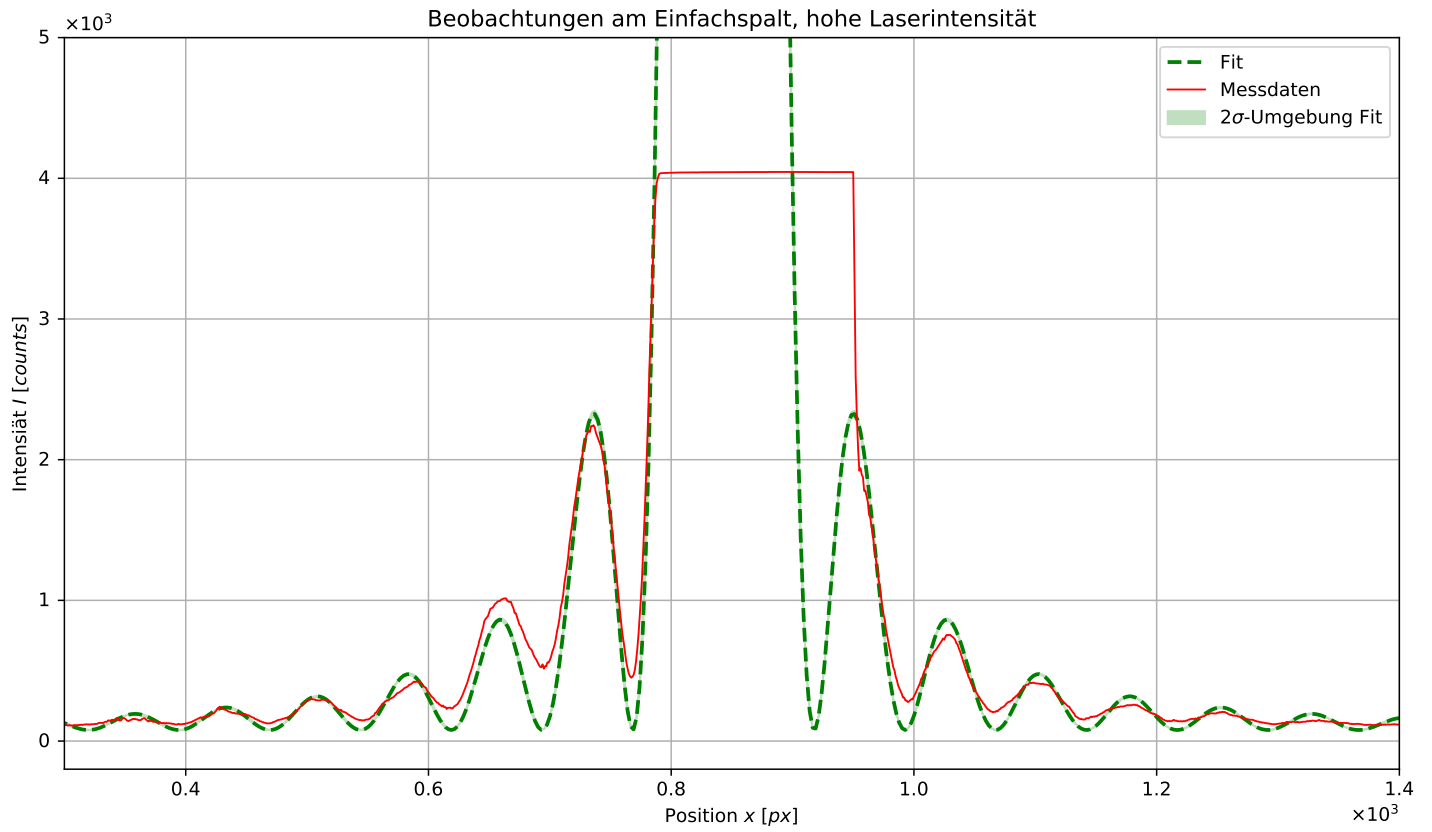


Abbildung 5

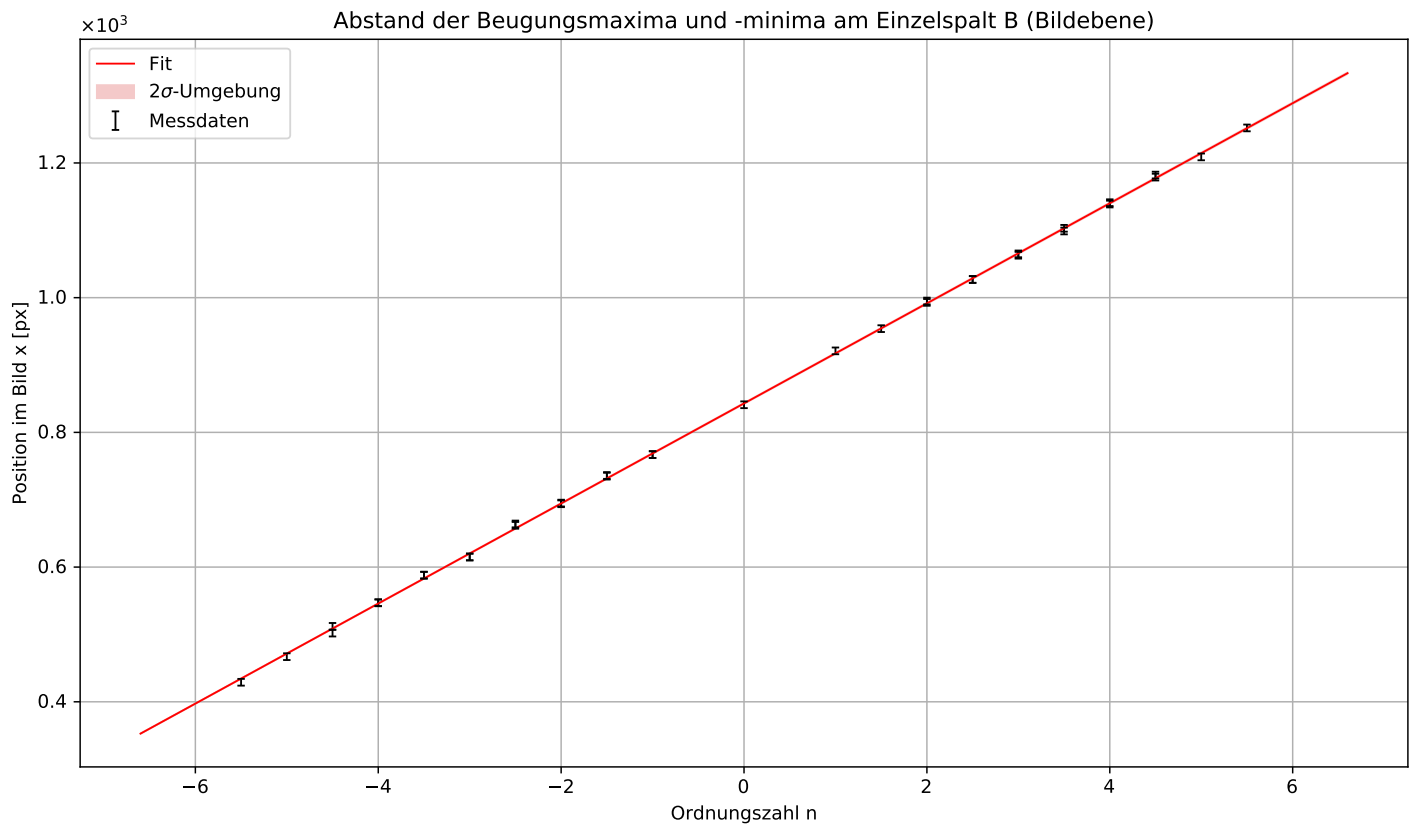


Abbildung 6

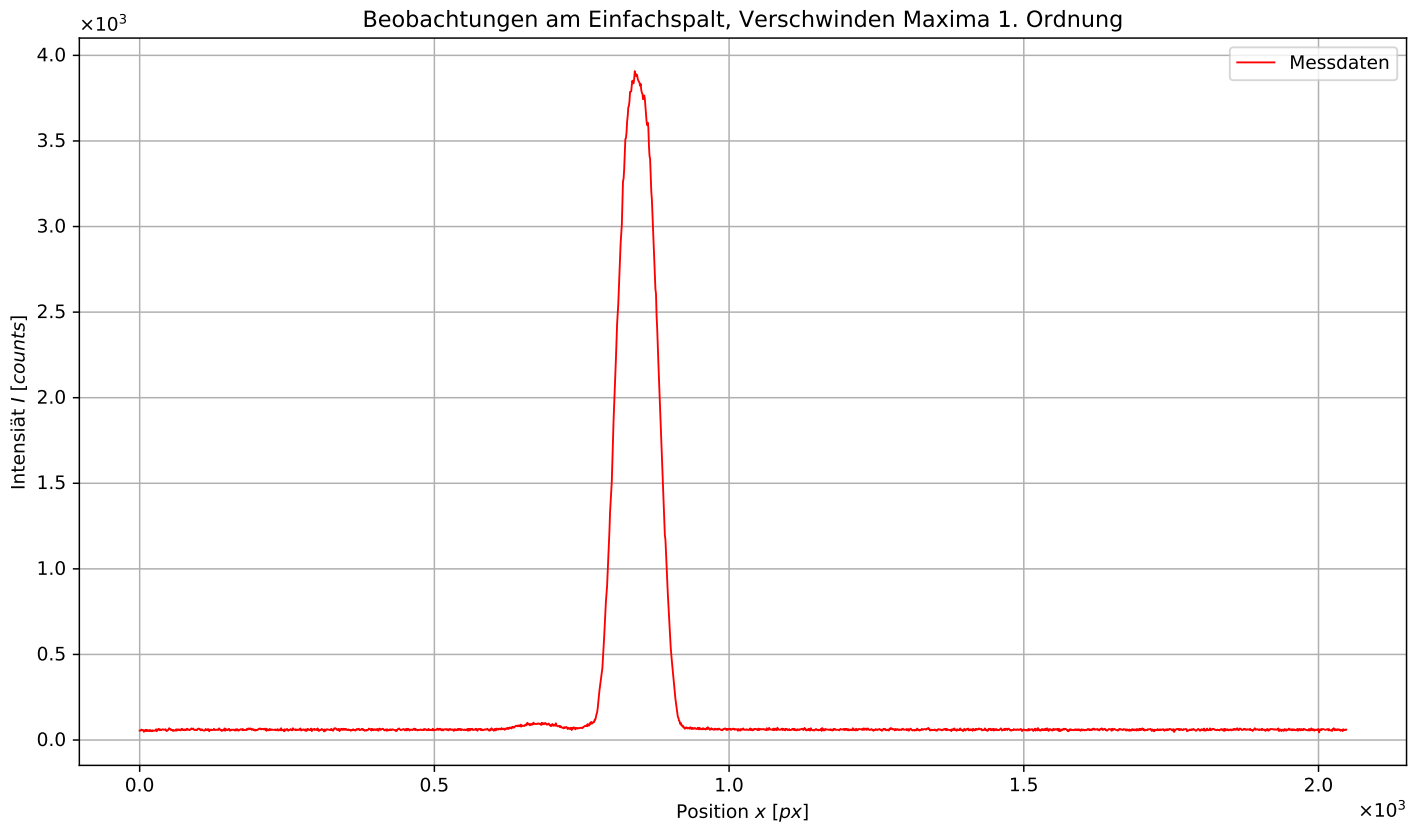


Abbildung 7

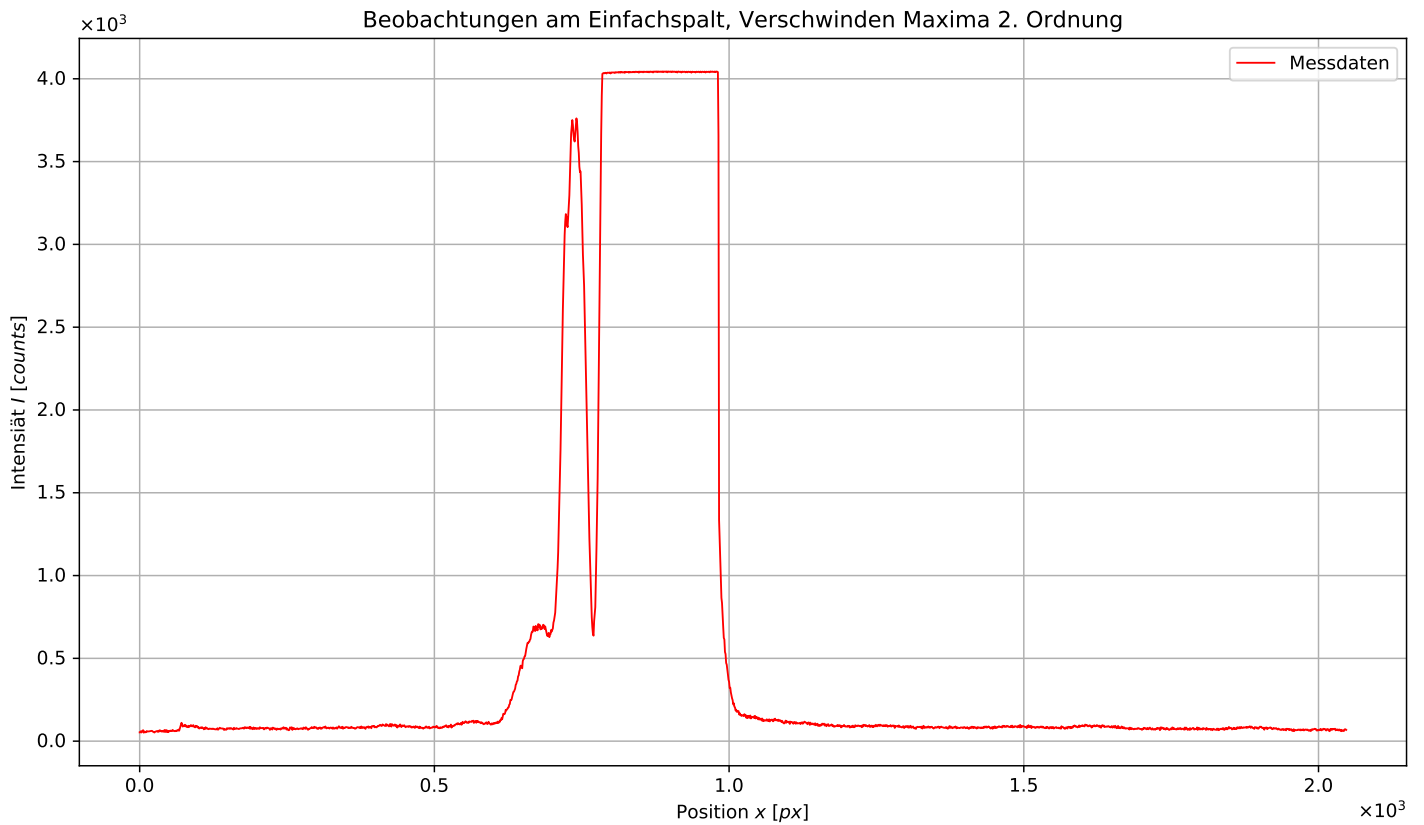


Abbildung 8

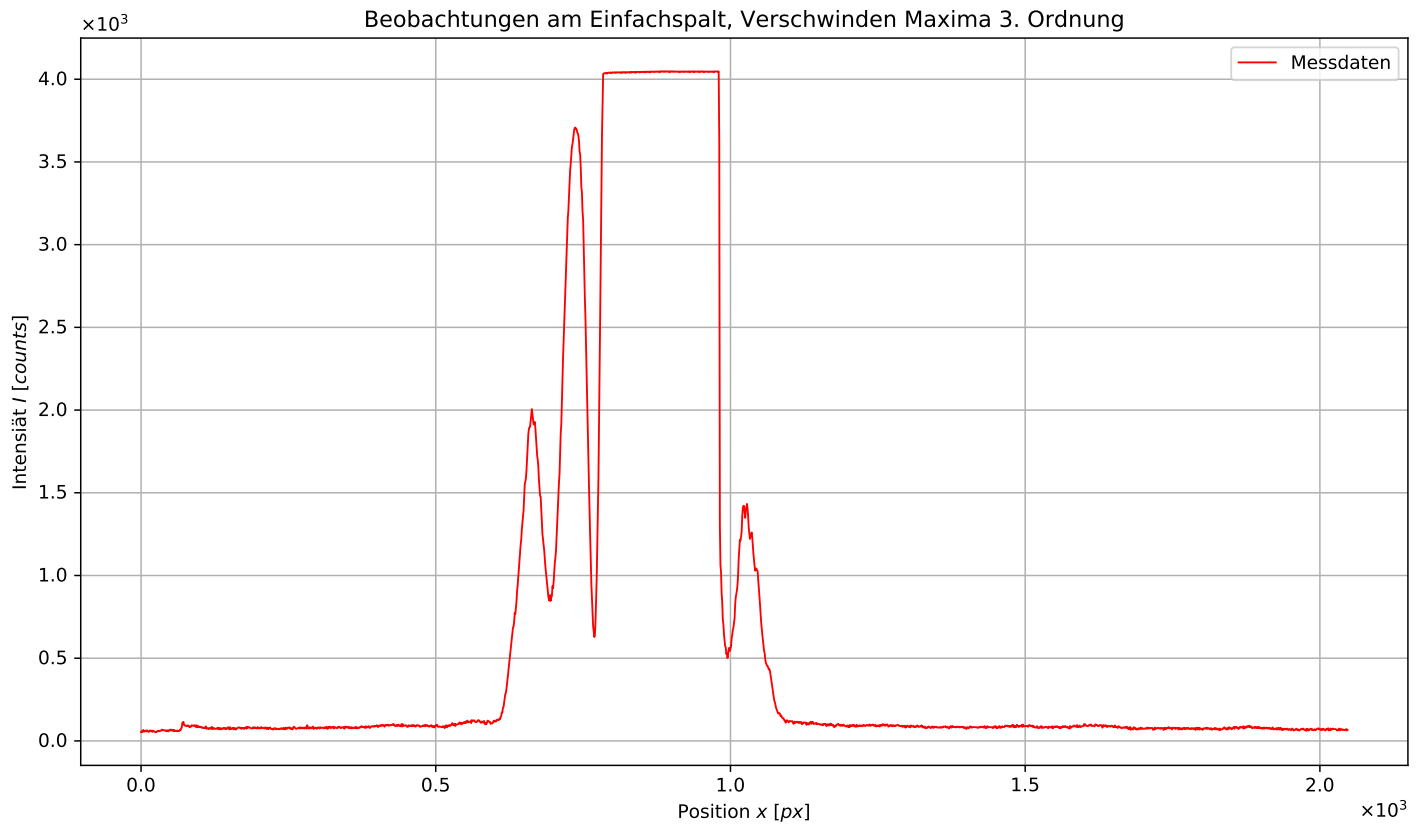


Abbildung 9

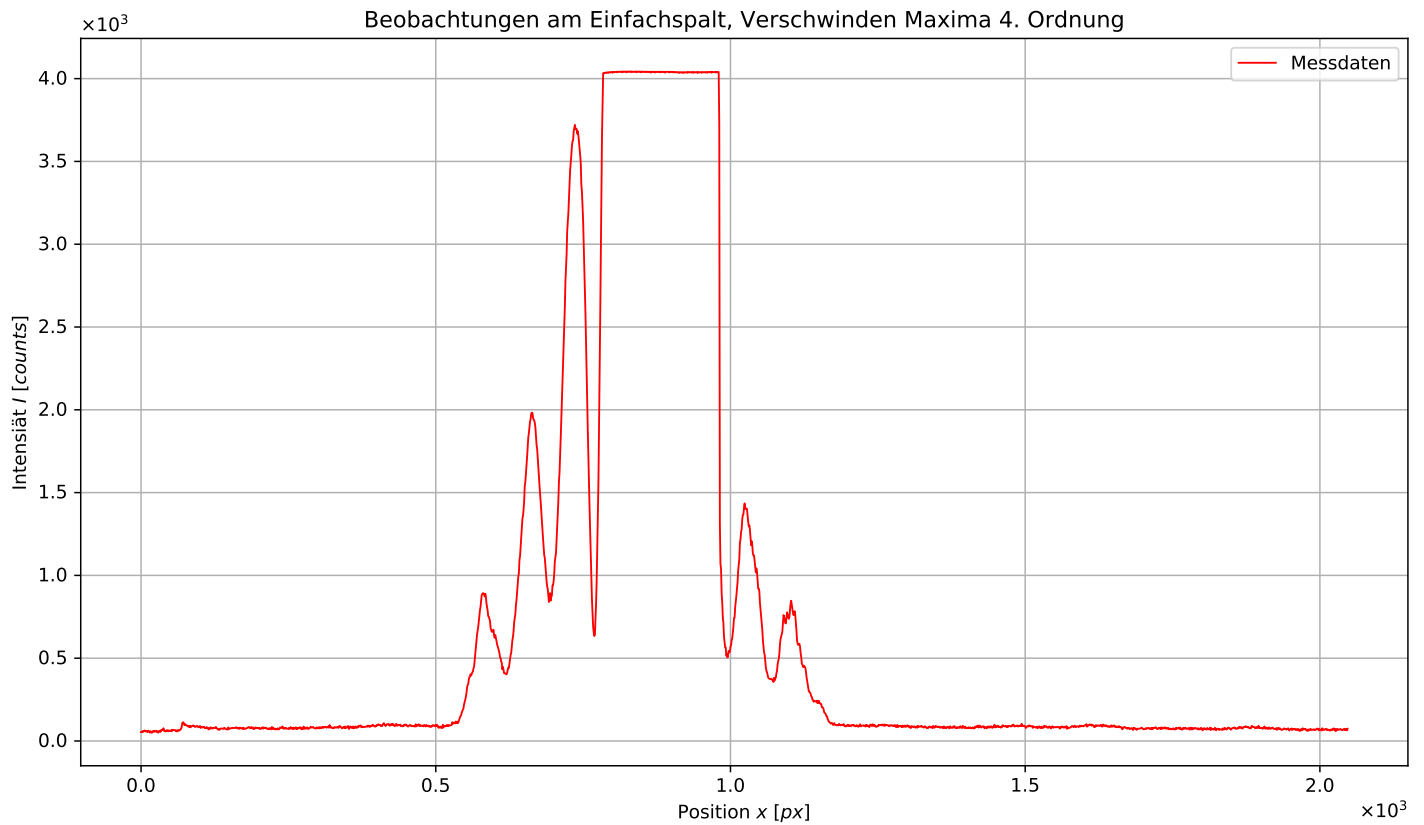


Abbildung 10

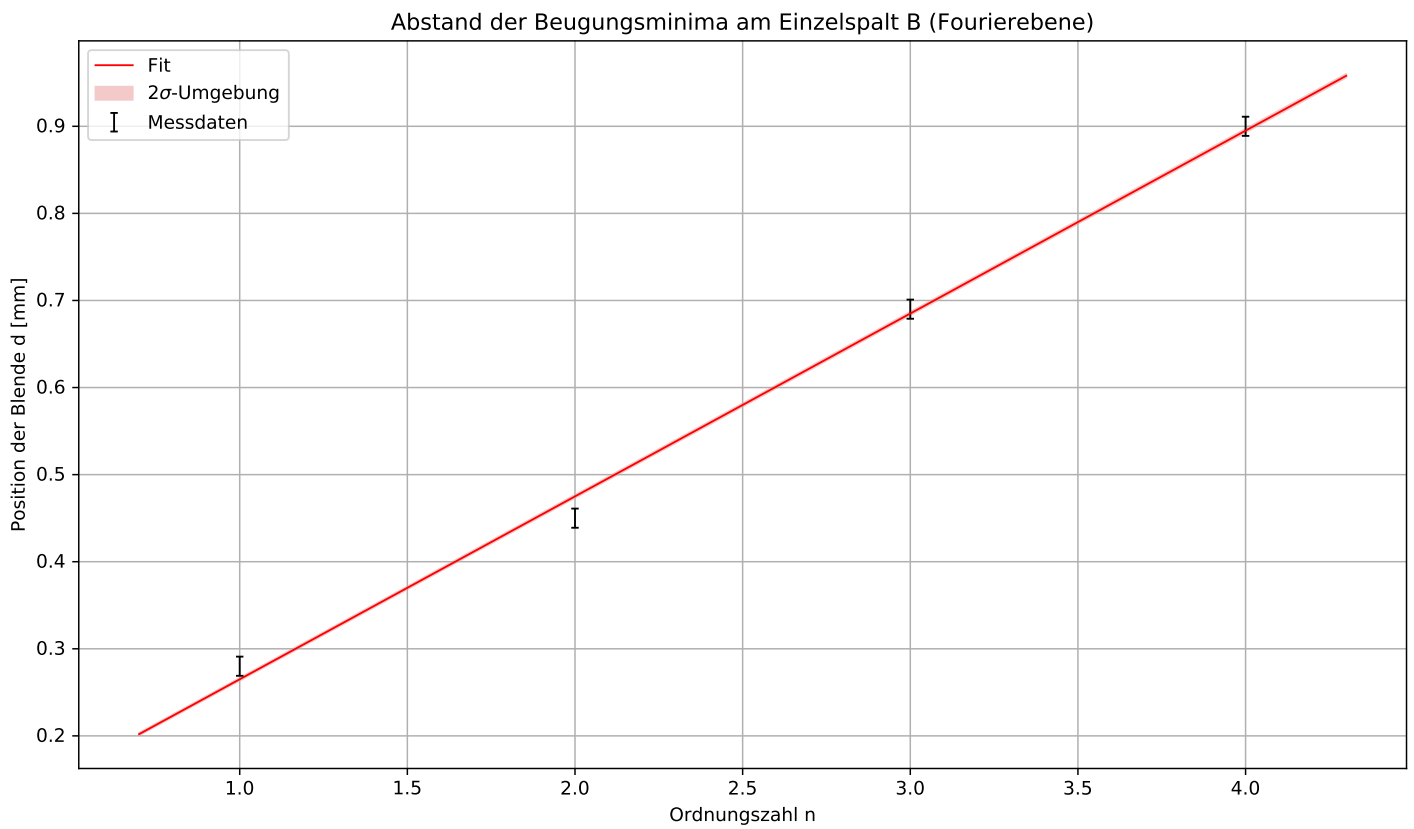


Abbildung 11

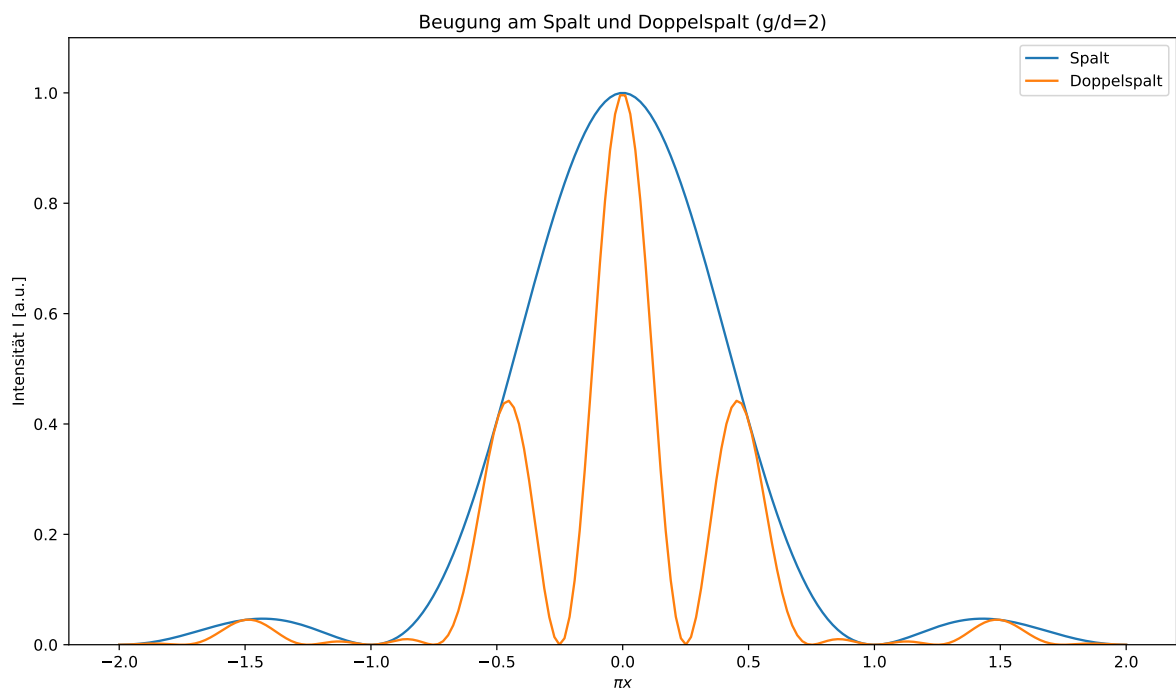


Abbildung 12

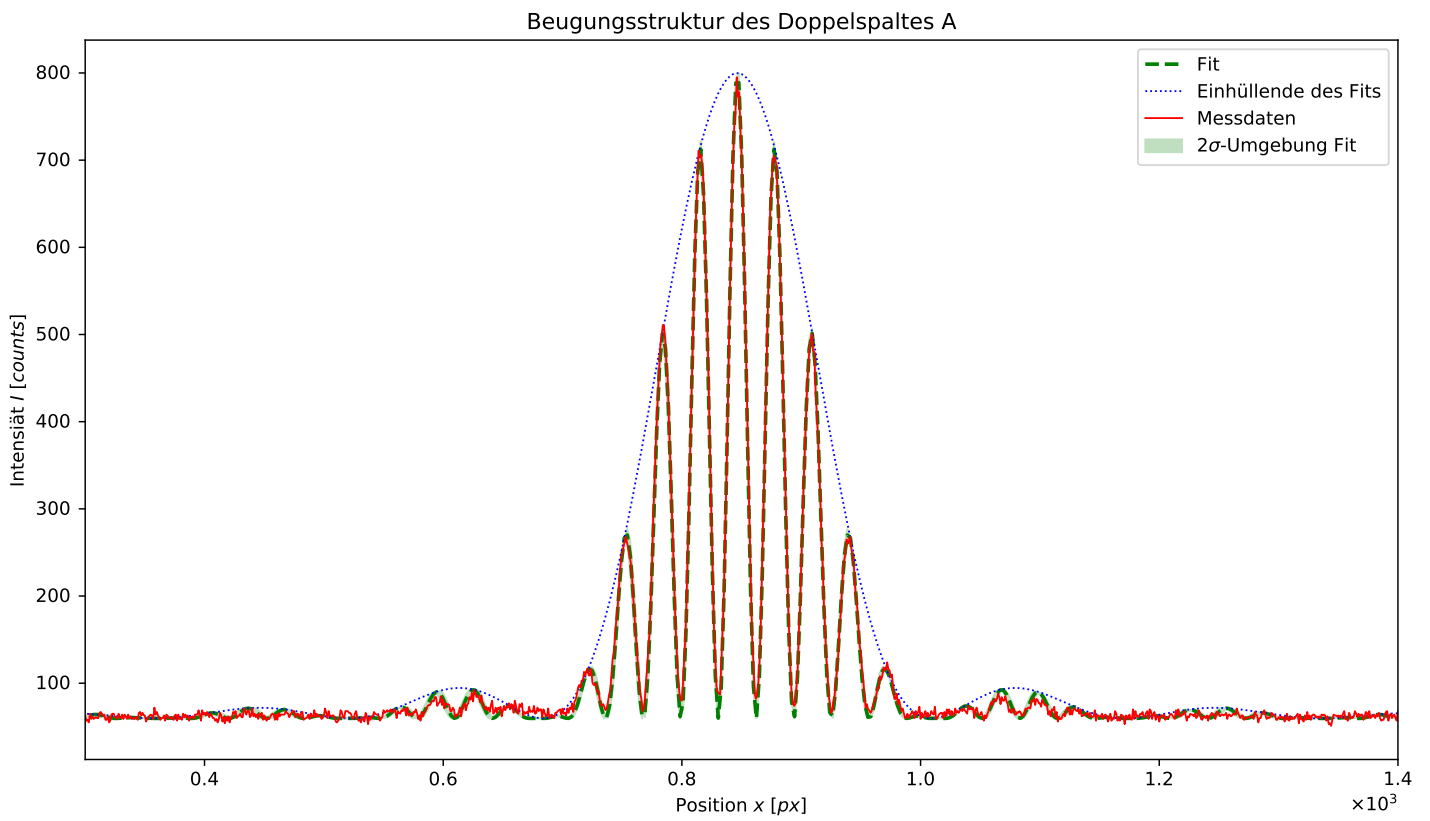


Abbildung 13

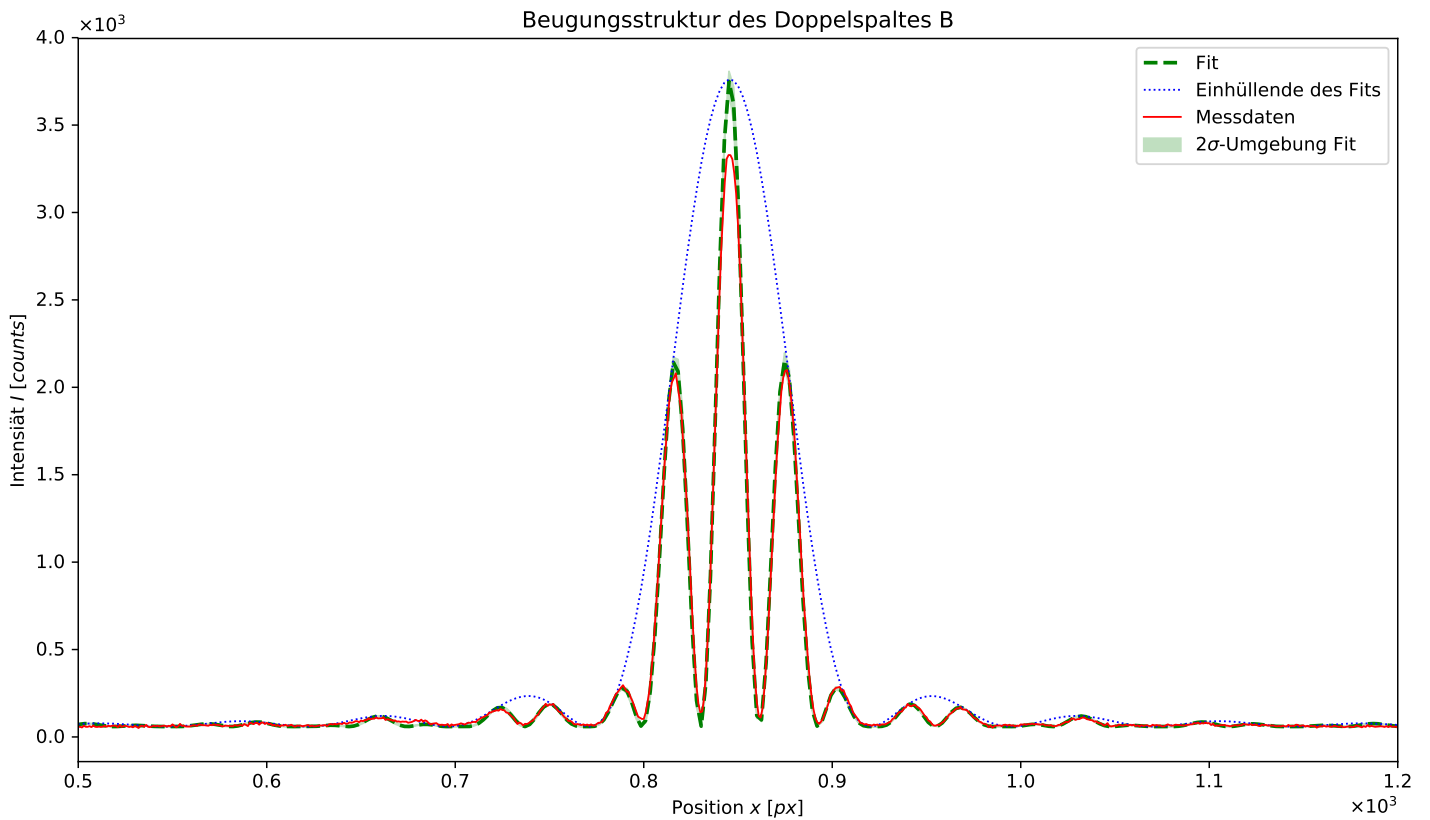


Abbildung 14

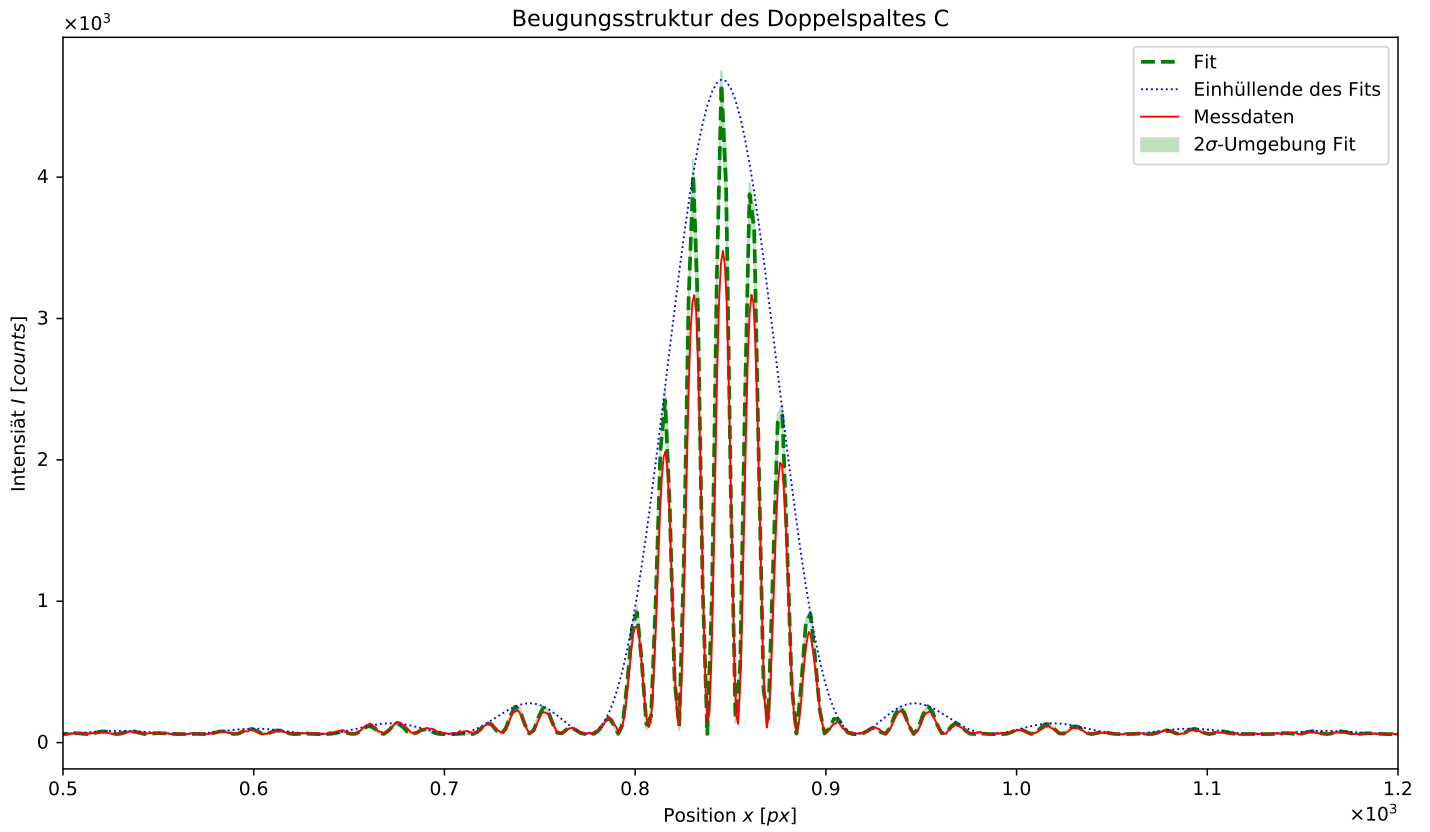


Abbildung 15

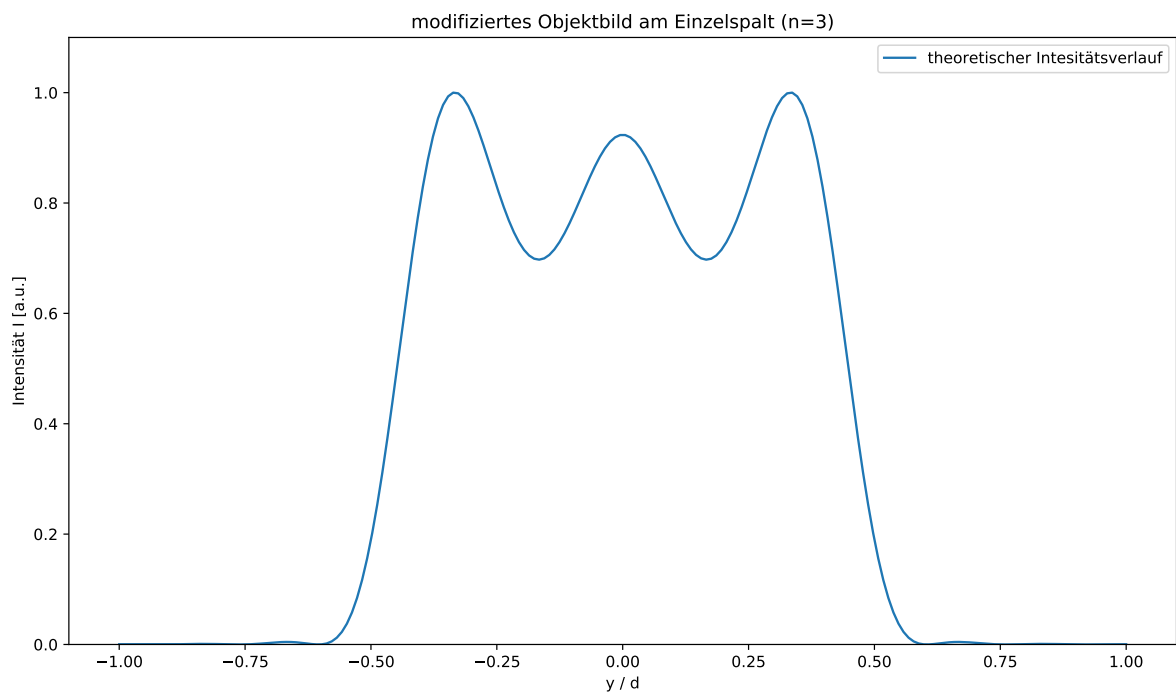


Abbildung 16

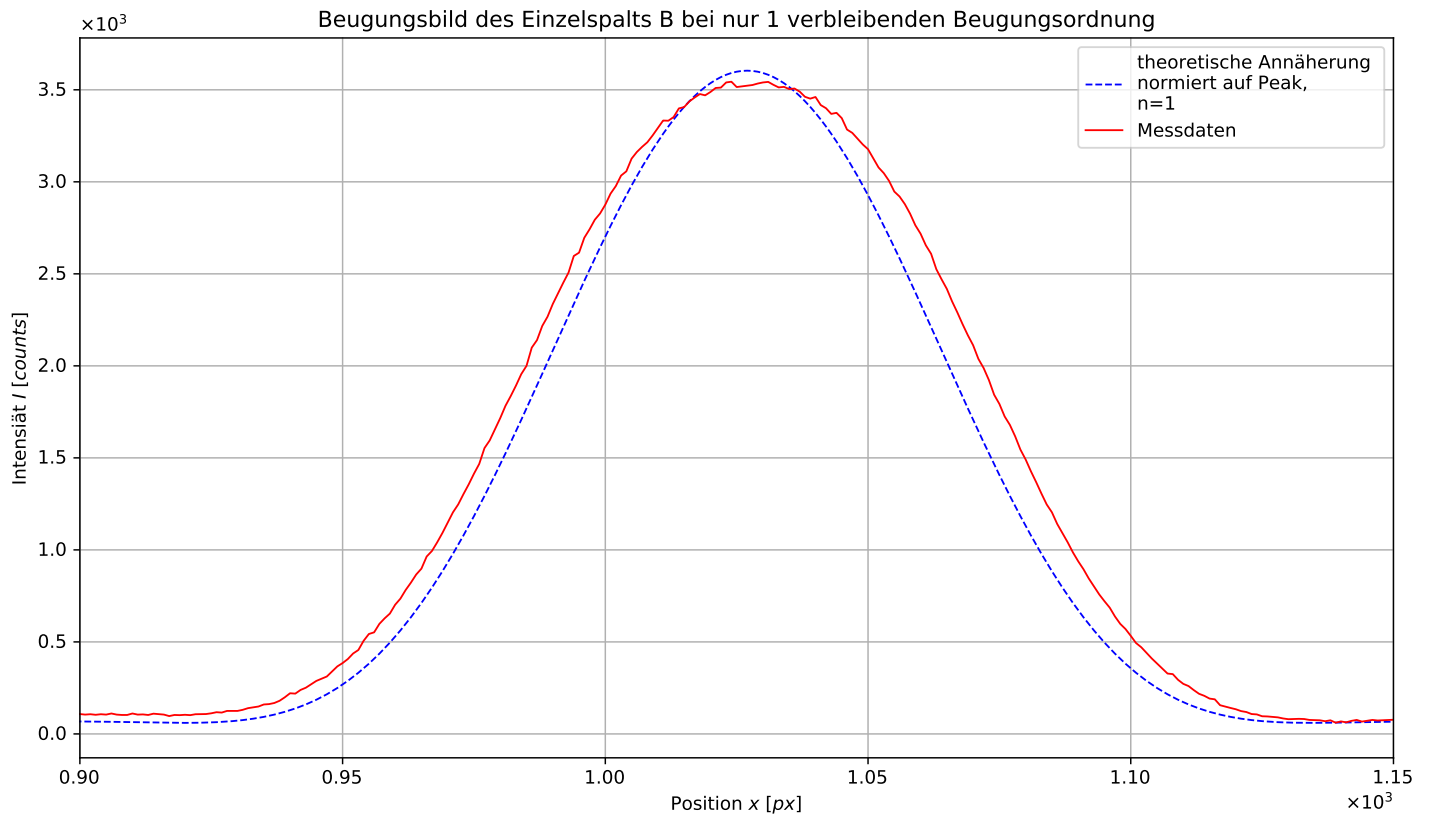


Abbildung 17

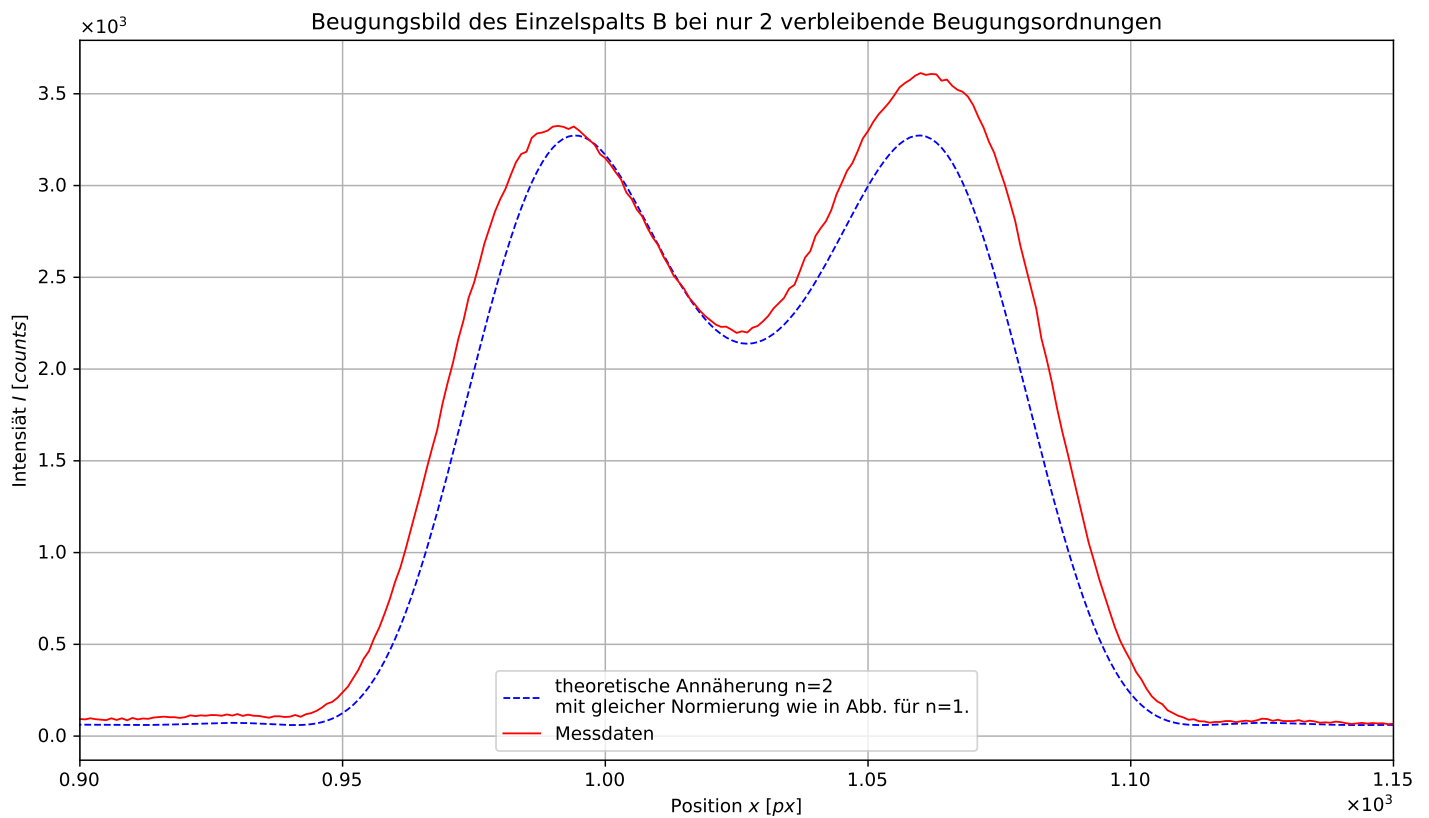


Abbildung 18

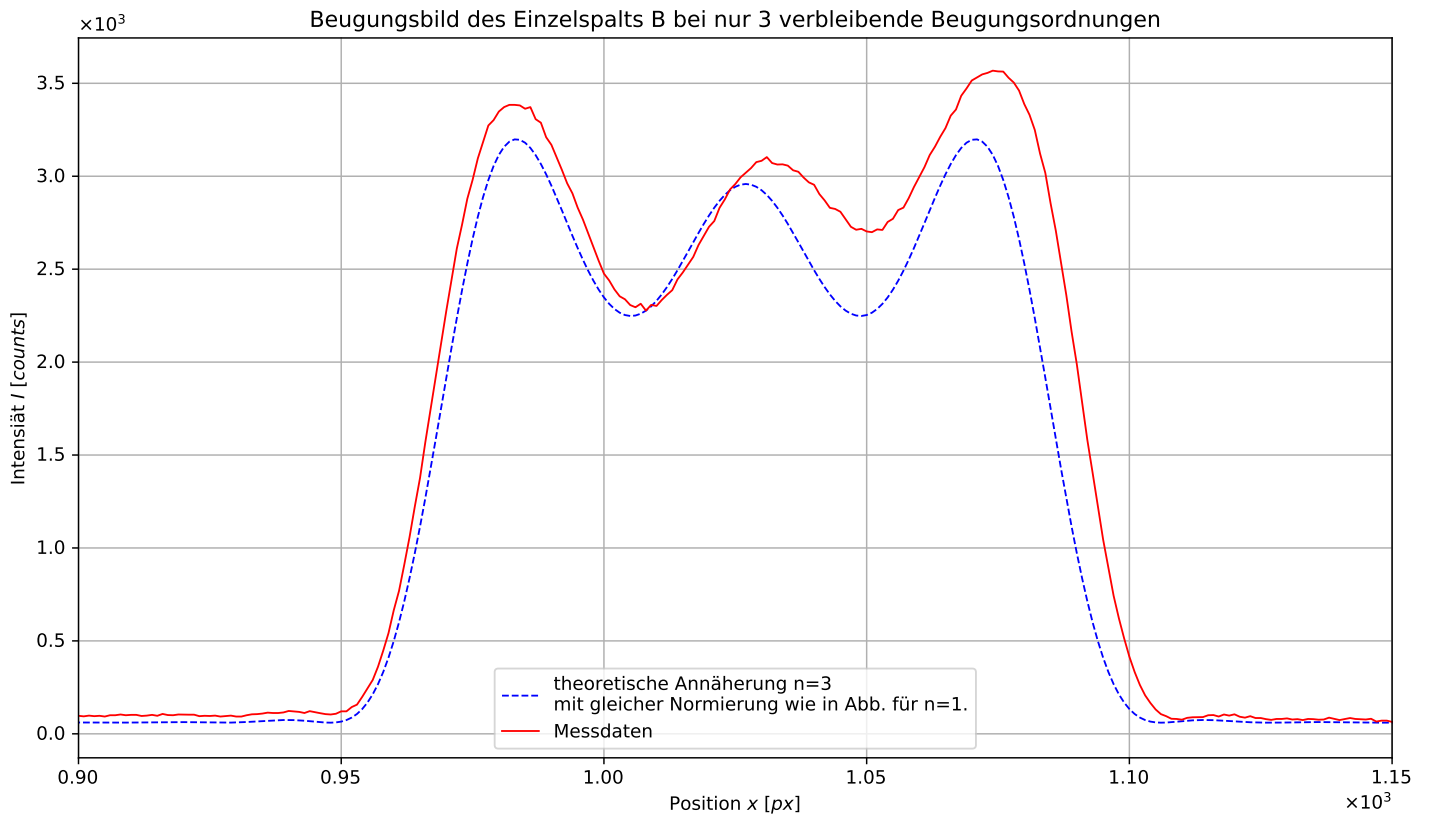


Abbildung 19

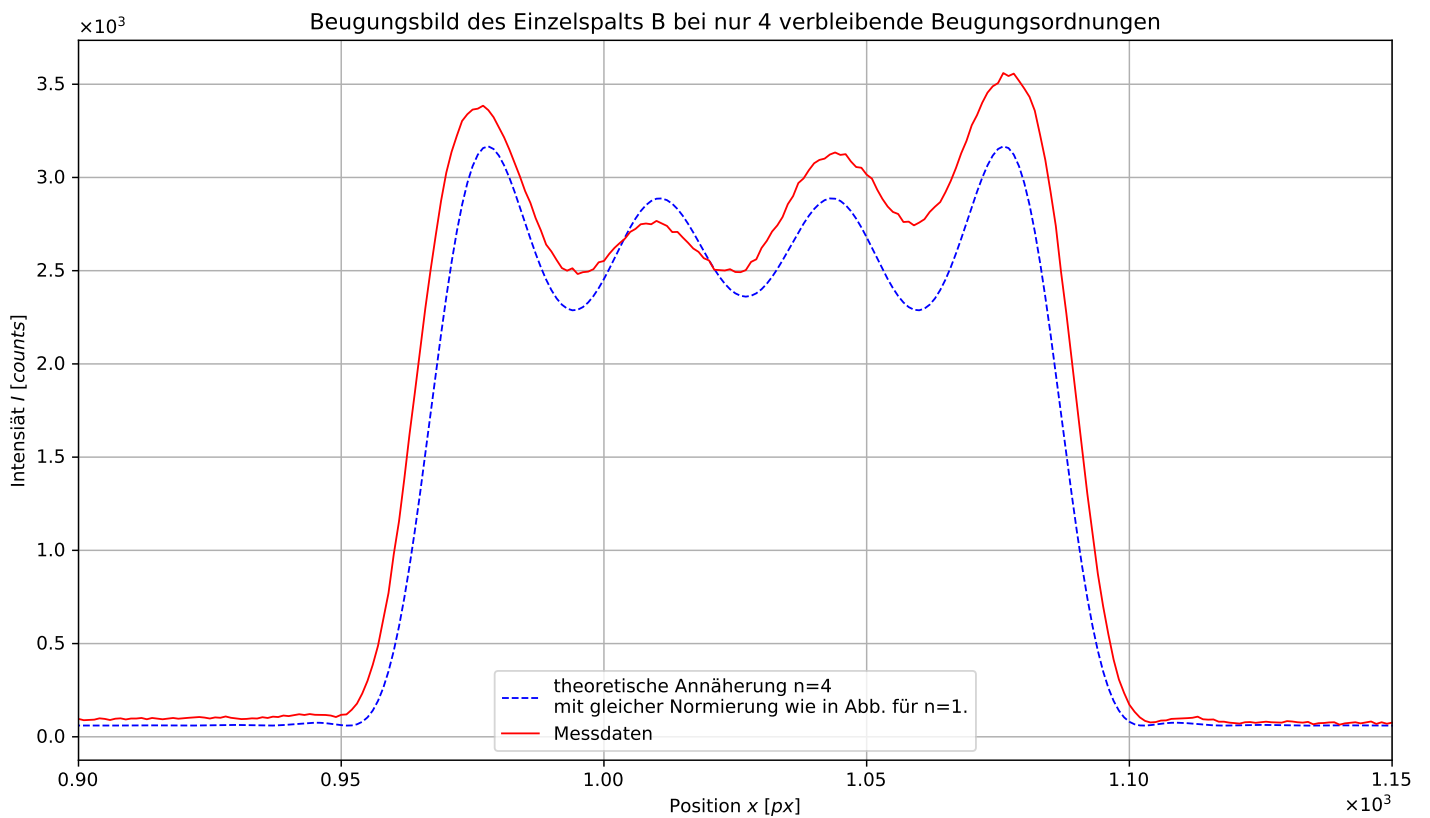


Abbildung 20

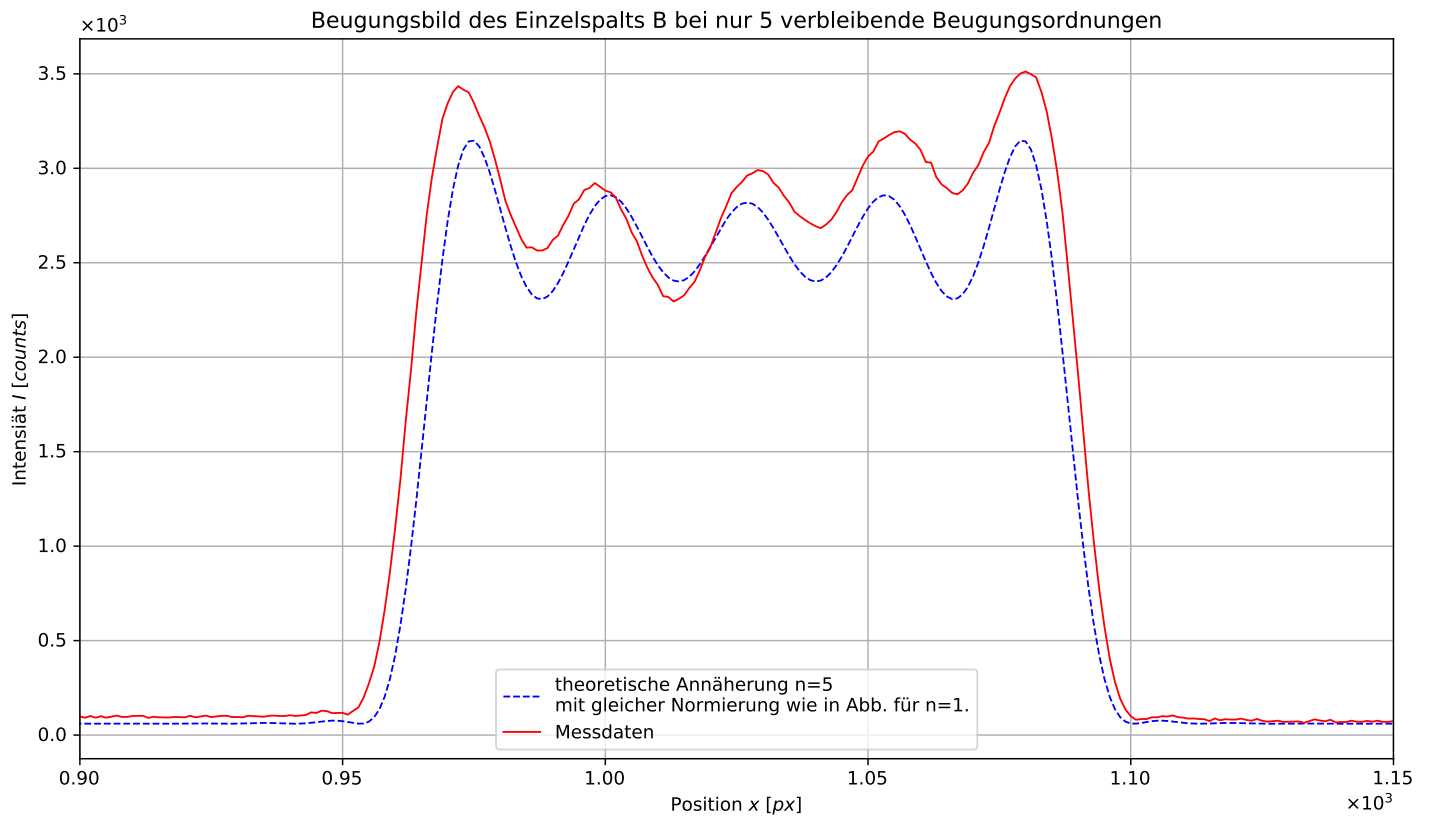


Abbildung 21

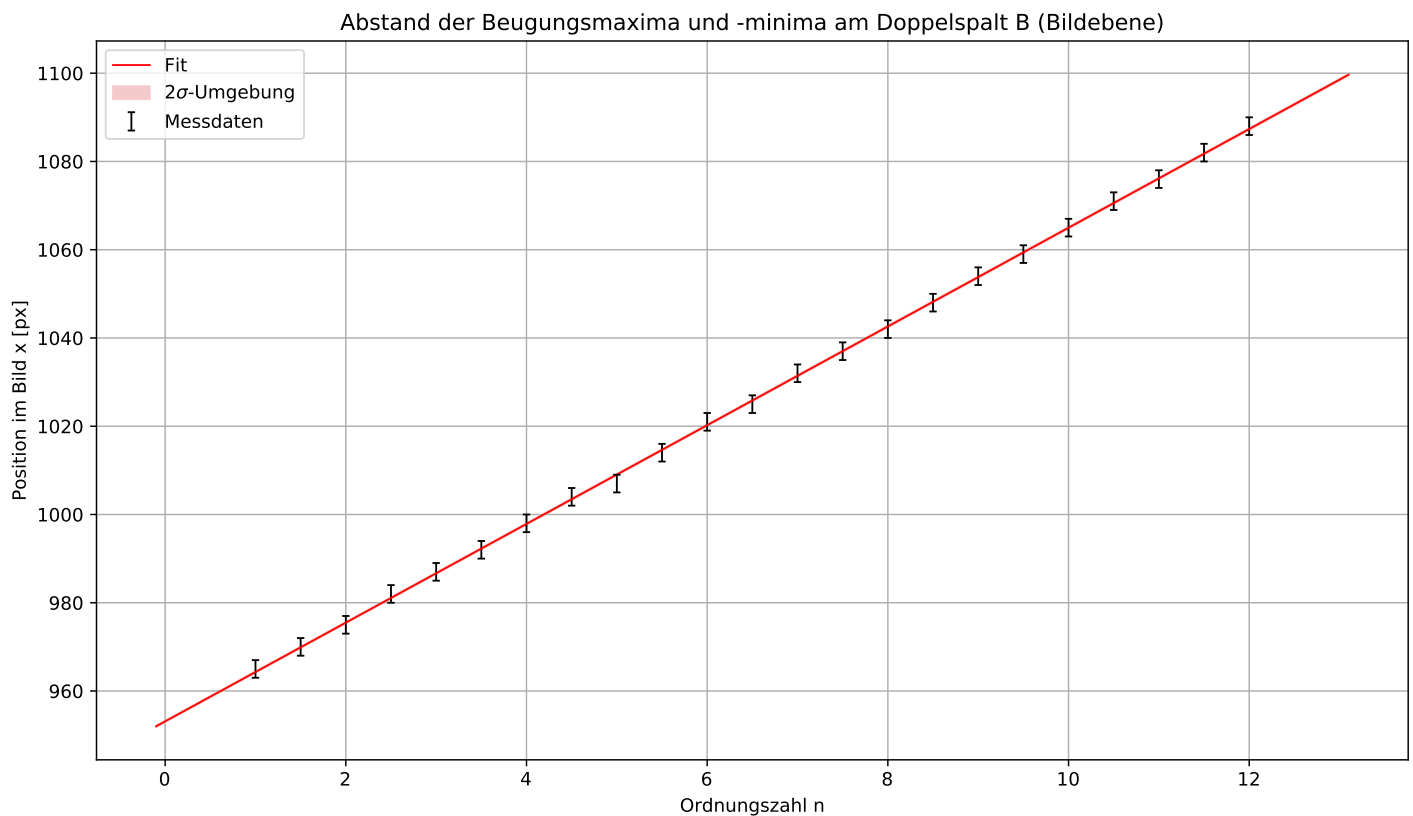


Abbildung 22

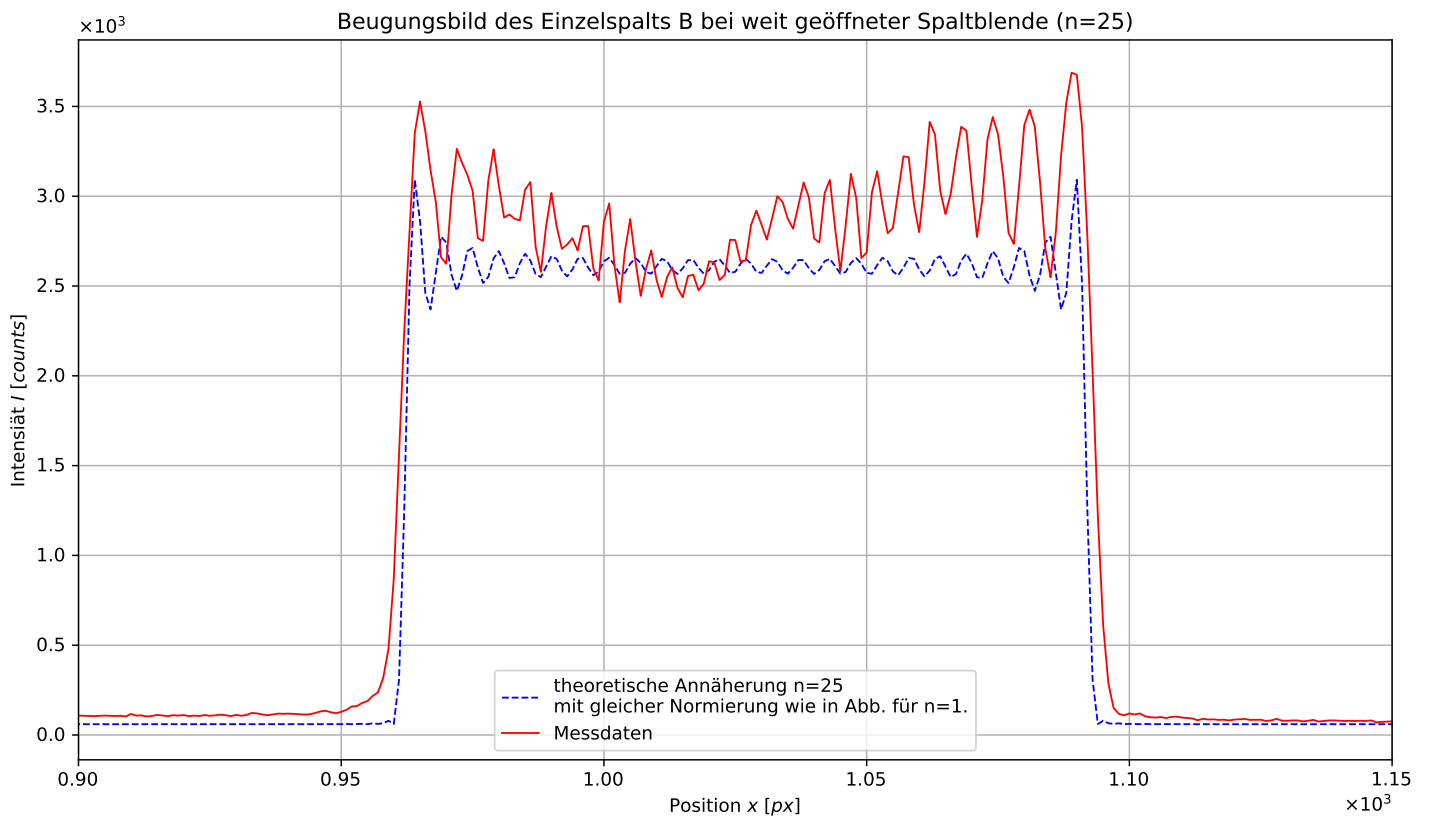


Abbildung 23

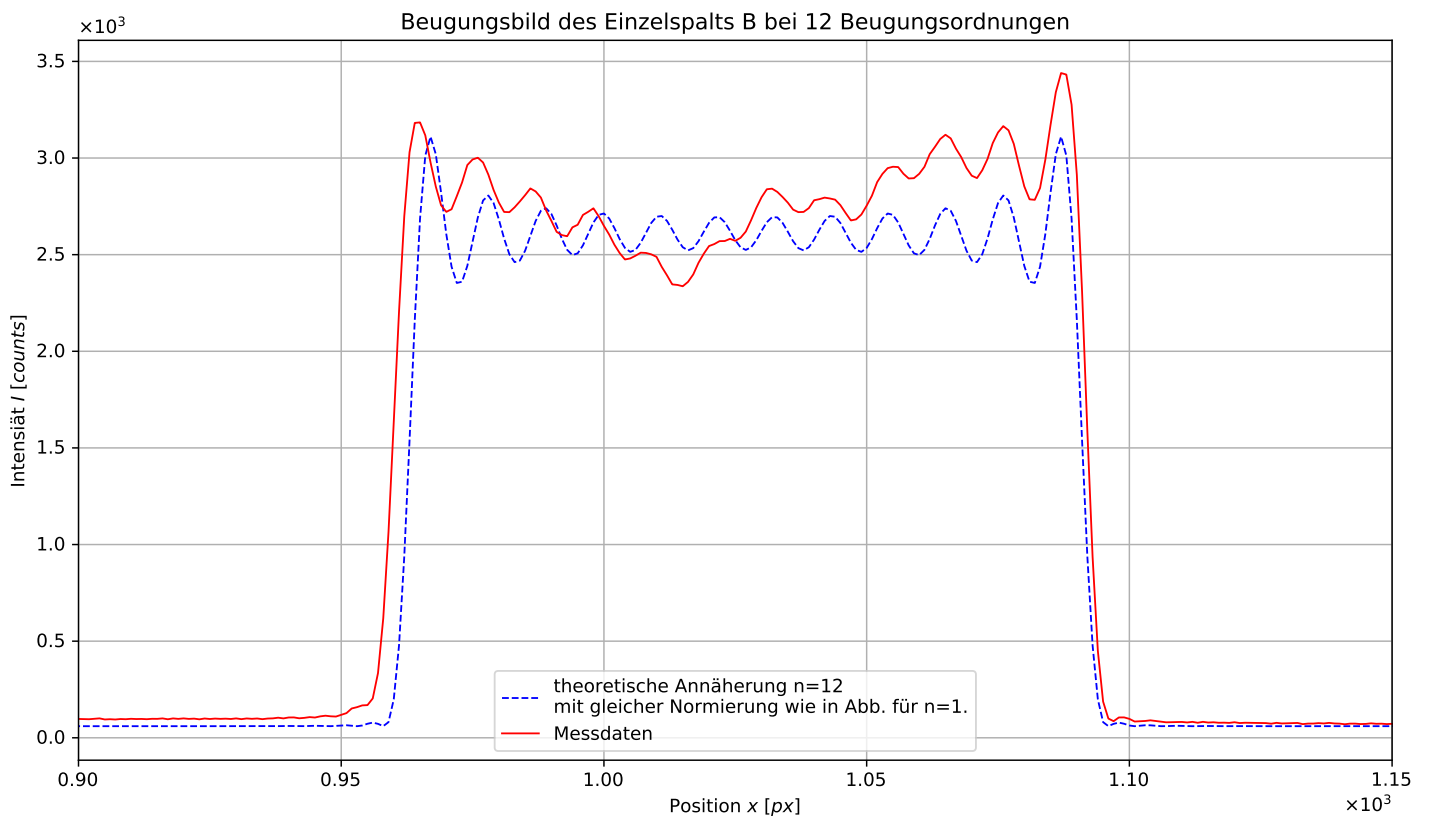


Abbildung 24

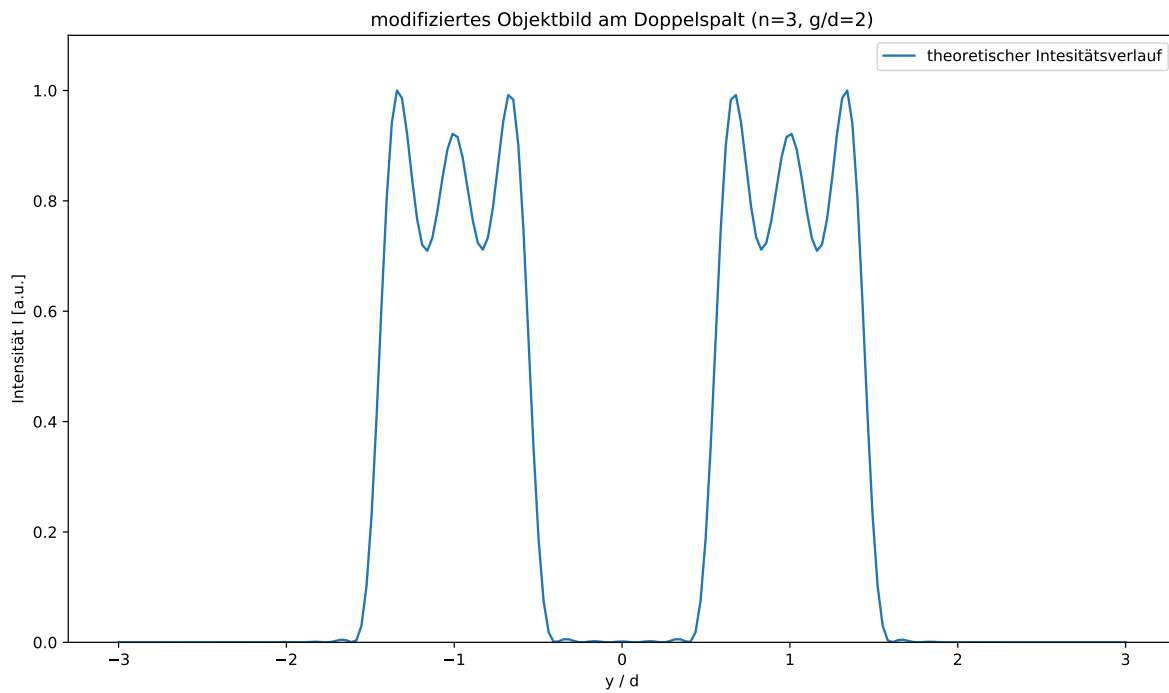


Abbildung 25

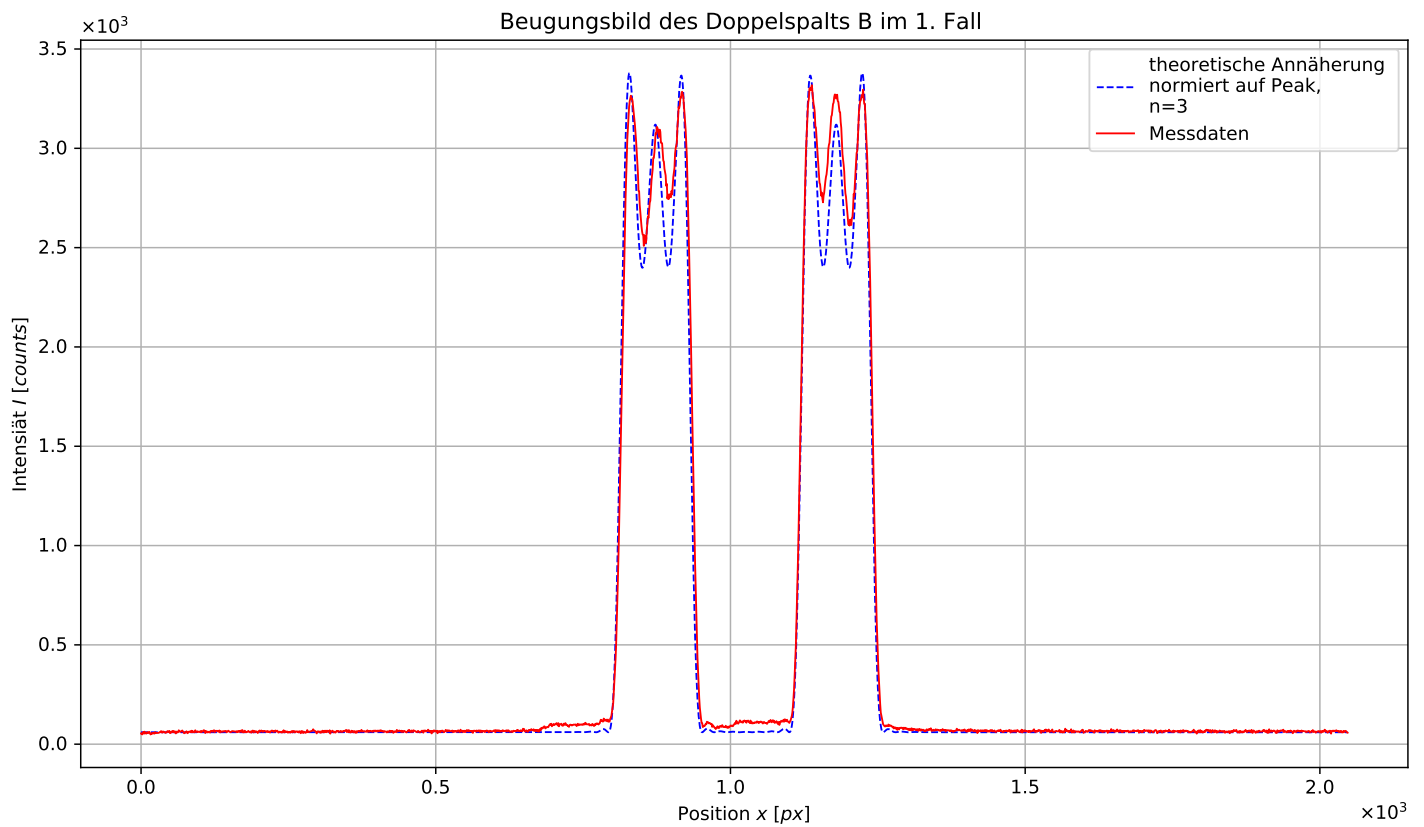


Abbildung 26

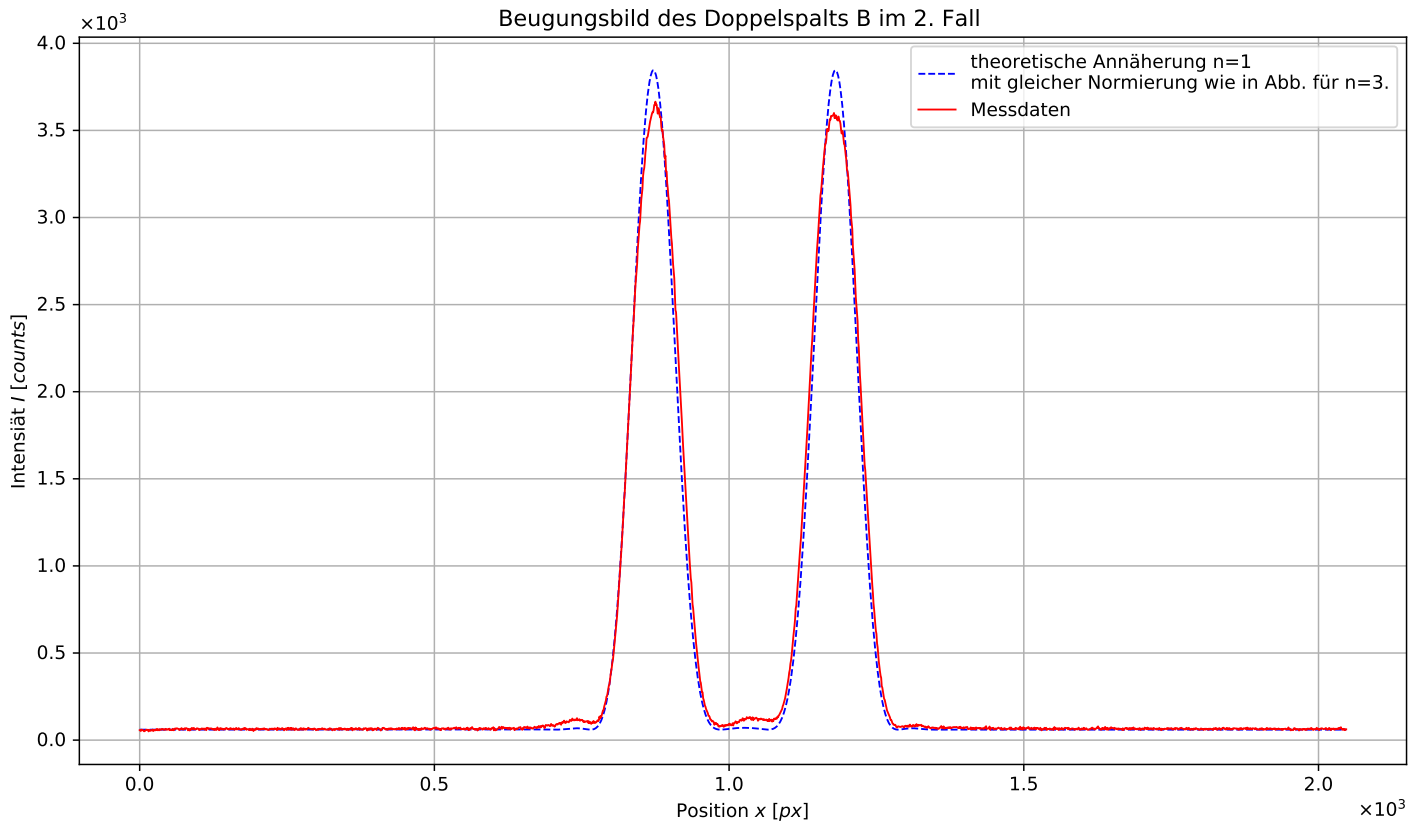


Abbildung 27

