Fortgeschrittenen Praktikum I

Ultraschall-Phasengitter

Wiebke Herzberg und Kolja Glogowski

12. September 2004

Inhaltsverzeichnis

1. Aufgabenstellung

Amplitudengitter

- Bestimmung der Gitterkonstante eines Sinusgitters aus dem Abstand de 1. Beugungsordnung
- Bestimmung der Gitterkonstante von fünf Amplitudengittern
- Berechnung der Aperturfunktion für Gitter Nr.1 (größte Gitterkonstante, höchste Dichte an Beugungsmaxima) aus den ermittelten Intensitäten der Beugungsordnungen und Zeichnen einer Periode der Aperturfunktion
- Bestimmung des Verhältnisses der Spaltbreite zum Spaltabstand aus der Aperturfunktion
- Bestimmung des Auflösungsvermögens der Gitter bei ihrer vollen Ausleuchtung

Phasengitter

- Messung der Intensitätsverteilung der Beugungsfigur eines Ultraschallwellengitters (Phasengitter) in Abhängigkeit von der Spannung am Ultraschallschwingquarz
- Vergleich der Messergebnisse mit der Raman-Nath-Theorie
- Bestimmung der Schallwellenlänge in Isooktan durch Ausmessen der Beugungsordnungen und Vergleich mit dem rechnerischen Wert

2. Theoretische Grundlagen

2.1. Beugung

Unter Beugung versteht man die Änderung der Richtung elektromagnetischer Wellen an einem Hindernis, die nicht durch Brechung oder Reflexion zustande kommt. In unserem Versuch betrachten wir Beugung an Amplituden- und Phasengittern.

2.2. Amplitudengitter

Die für jedes Gitter charakteristische Größe ist die Gitterkonstante K, nämlich der Abstand zweier benachbarter Spaltmitten. Sie lässt sich bei bekannter Wellenlänge λ aus dem Intensitätsmaximum m-ter Ordnung und dem dazugehörigen Winkel θ, dem Winkel zwischen 0-ter und m-ter Ordnung vom Gitter aus gemessen, bestimmen:

$$K = \frac{m\,\lambda}{\sin\theta}\ . \tag{1}$$

2.3. Aperturfunktion

Jedem Gitter (oder allg. Beugungshindernis) lässt sich auch eine Aperturfunktion zuordnen. Sie beschreibt die Eigenschaften des Beugungshindernisses indem sie jedem Punkt der Blendenebene einen Wert zuordnet. Mit dem Kirchhoffschen Integraltheorem und der daraus erhaltenen Integralformel für die Amplitude einer Kugelwelle auf dem Rand ihres Ausbreitungsgebietes kann man zeigen, dass die Intensitätsverteilung I des Beugungsbildes die Fouriertransformierte der Aperturfunktion g des Beugungshindernisses ist.

$$I = |\Psi(x,y)|^2 = \left| \int_{Blende} g\ e^{-ikr}\ dA \right| \tag{2}$$

Umgekehrt lässt sich aus Symmetriegründen aber auch die Aperturfunktion als Fouriertransformierte der Intensitätsverteilung erhalten.

Die Fouriertransformierte einer Funktion $g(x)$ ist definiert als:

$$F(g) = \int_{-\infty}^{\infty} g(x)\ e^{-ikx}\ dx. \tag{3}$$

Ist nun die Intensitätsverteilung im Einzelnen nicht bekannt, kann man die Apertur-funkrion auch durch eine Fourierreihe nähern, in welcher die Wurzeln der gemessenen

Intensitätsmaxima als Koeffizienten auftreten:

$$g(x) = \sum_{j=0}^{\infty} \pm\sqrt{I_j}\ \cos\left(\frac{x}{K}\ 2\pi j\right) \tag{4}$$

Für ein Sinusgitter z.B. ergibt sich $g(x) = \sqrt{I_0}\ +\ \sqrt{I_1}\ \cos\left(\frac{x}{K}\ 2\pi\right)$, da nur Maxima 0-ter und 1-ter Ordnung entstehen.

Aus dem Graph der Aperturfunktion kann man dann das Verhältnis von Spaltbreite zu Spaltabstand ablesen.

2.4. Auflösungsvermögen

Das Auflösungsvermögen a ist definiert als

$$a = \frac{\lambda}{\Delta\lambda}, \tag{5}$$

wobei λ die Wellenlänge und $\Delta\lambda$ der Wellenlängenabstand ist, bei dem sich eine andere Wellenlänge von λ bei Beugung noch unterscheiden lässt. Außerdem lässt sich zeigen, dass gilt:

$$a = N\ m \tag{6}$$

Das Auflösungsvermögen ist also gleich der Anzahl der lichtdurchsetzten Gitterlinien N mal der maximal beobachteten Beugungsordnung m.

2.5. Phasengitter

In unserem Versuch wird das Phasengitter durch eine laufende Ultraschallwelle in Isooktan erzeugt. Die Schallwelle erzeugt in der Flüssigkeit periodische Druckschwankungen, also Bereiche unterschiedlicher Dichte und somit auch unterschiedlicher Brechungsindizes. Dadurch treten ursprünglich gleichphasige Wellen phasenversetzt aus dem Gitter aus. Für den Brechungsindex n im schalldurchsetzten Medium gilt:

$$n(x) = n_0 + \Delta n\ \sin\left(\frac{2\pi}{\Lambda}\ x\right) \tag{7}$$

Dabei ist die Schallwellenlänge Λ das Analogon zur Gitterkonstante.

Die Schallintensität S ist dem Quadrat der an der Ultraschallzelle angelegten Spannung U proportional, sowie dem Quadrat der relativen Dichte: $S \sim \left(\frac{\Delta\rho}{\rho_0}\right)^2$. Da $\frac{\Delta n}{n-1} = \frac{\Delta\rho}{\rho_0}$ ist auch $\Delta n \sim S$ und wir können somit den Brechungsindex mit der Amplitude des Schallwellenfeldes ändern.

2.6. Raman-Nath-Theorie

- Für die Winkel der Intensitätsmaxima des Beugungssbildes gilt in m-ter Ordnung:

$$\sin\theta = \pm\frac{\lambda}{\Lambda}\, m\ . \tag{8}$$

- Die Intensitäten der Maxima m-ter Ordnung verhalten sich zu denen m'-ter Ordnung wie die Besselfunktionen der m-ten zu denen der m'-ten Ordnung.

$$\frac{I_m}{I_{m'}} = \frac{J_m^2(\Delta n\ D \cdot 2\pi/\lambda)}{J_{m'}^2(\Delta n \cdot 2\pi/\lambda)} \tag{9}$$

3. Versuchsaufbau und Durchführung

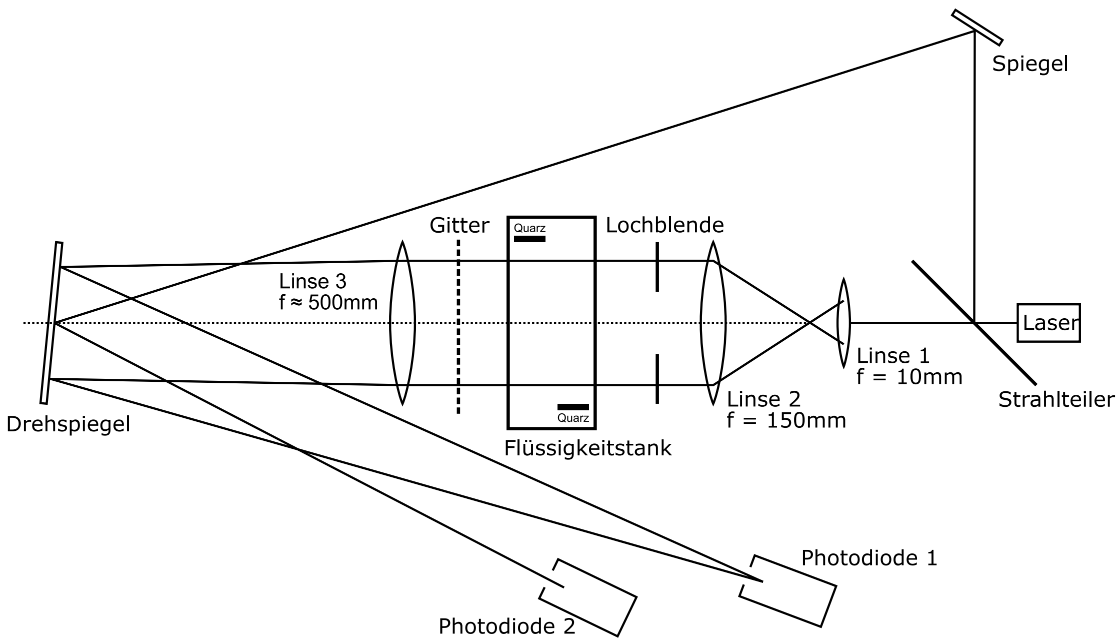

Abbildung 1: Schematische Darstellung der Versuchsaufbau

Versuchsaufbau

Wir verwenden das monochromatische Licht eines He-Ne-Lasers ($\lambda = 632{,}8$ nm). Durch eine erste Linse wird der Strahl aufgeweitet und durch eine zweite (Kollimator) in ein paralleles Strahlenbündel verwandelt, in welches man dann das jeweilige Beugungshindernis (verschiedene Gitter, Ultraschallzelle) einbringt. Diese Anordnung wird Fraunhofer Anordnung genannt. Das Entstehende Beugungsbild wird mittels einer dritten Linse wieder fokussiert und über einen rotierenden Spiegel 12,5 mal pro Sekunde über eine Photodiode geschoben. Das Beugungsbild wird also praktisch sequentiell abgetastet. Das Signal aus der Diode wird vorverstärkt und dann an ein Oszilloskop weitergegeben, an welchem auch ein XY-Schreiber angeschlossen ist. Noch vor der ersten Linse wird der Strahl durch einen Strahlteiler aufgespalten. Dieser zweite Strahl wird dann ebenfalls über den Drehspiegel auf eine zweite Photodiode gelenkt deren Signal zum Triggern des Oszilloskops genutzt werden soll.

Durchführung

Zunächst nahmen wir sämtliche Linsen aus dem Strahlengang um den Laserstrahl möglichst waagerecht auszurichten. Dann brachten wir das Sinusgitter ein und maßen den Abstand zwischen der nullten und ersten Beugungsordnung, sowie den Abstand zwischen Schirm und Gitter. Danach ordneten wir die optischen Elemente wie in Abbildung 1 an. Wir nahmen die Beugungsbilder von 5 verschiedenen Gittern sowie einem Eichungsgitter mit bekannter Gitterkonstante mittels des XY-Schreibers auf. Zur Ermittlung des Auflösungsvermögens versuchten wir den Durchmesser des Lichtstrahls zu bestimmen, dann zählten wir die Anzahl der Beugungsordnungen bei den verschiedenen Gittern.

Daraufhin setzten wir die mit Isooktan gefüllte Ultraschallzelle ein und nahmen eine Serie von Beugungsbildern in Abhängigkeit zur angelegten Spannung auf. Um die Schallwellenlänge zu bestimmen nahmen wir nochmals ein Bild des Eichgitters auf, diesmal aber mit dem vorgesetzten Isooktantank.

4. Auswertung

4.1. Gitterkonstante des Sinusgitters

Als Abstand zwischen den Beugungsordnungen erhielten wir $a = (4,65 \pm 0,10)\,cm$ und für den Abstand zwischen Schirm und Gitter massen wir $x = (5,3 \pm 0,2)\,cm$. Da Strahl und Schirm senkrecht zueinander standen, ergibt sich für die Hypothenuse $c = \sqrt{x^2 + a^2}$ und somit für den Winkel zwischen den Seiten x und c

$$\sin\theta = \frac{a}{c} = \frac{a}{\sqrt{x^2 + a^2}} \; .$$

Nach Gleichung 1 auf Seite 4 erhält man damit mit $m = 1$

$$K = \frac{\lambda}{\sin\theta} = \frac{\lambda\, c}{a} = \frac{\lambda}{a}\sqrt{x^2 + a^2} = \lambda\sqrt{\frac{x^2}{a^2} + 1}$$

und für den Fehler s_K

$$s_K = \sqrt{\left(\frac{\partial K}{\partial x}\right) s_x^2 + \left(\frac{\partial K}{\partial a}\right) s_a^2} = \frac{x\,\lambda}{a^2}\sqrt{\frac{a^2 s_x^2 + x^2 s_a^2}{x^2 + a^2}} \; .$$

Damit kamen wir für die Gitterkonstante des Sinusgitters auf das Ergebnis:

$$K = (9,5950 \pm 0,2355) \cdot 10^{-7} m \; ,$$

welches mit etwa 1σ Abweichung ziemlich gut zum angegebenen Wert $K_{ref} = 9,8425 \cdot 10^{-7} m$ passt.

4.2. Gitterkonstanten der fünf Amplitudengitter

Zur Bestimmung der Gitterkonstanten der fünf Amplitudengitter musste eine Eichung der Zeitachse vorgenommen werden. Dazu verwendeten wir das Referenzgitter R mit 80 Strichen pro cm, also einer bekannten Gitterkonstante $K_R = 0,0125\,cm$. Dadurch ließen sich mit dem bekannten Zusammenhang aus Gleichung 1 die Werte von $\sin\theta$ zu den verschiedenen Ordnungen m berechnen. Diese trugen wir dann gegen die von uns gemessenen Zeitwerte auf und ermittelten, aus dem so entstehenden linearen Zusammenhang $\sin\theta = at + b$, durch Geradenanpassung die Werte für a und b.

Da das Ablesen der Messwerte vom Papier durch Abstände d in cm erfolgte, mussten diese zuerst in die Zeit t umgerechnet werden. Für einen Umrechnungsfaktor T_B gilt dann $t = T_B \cdot d$ und mir einem Ablesefehler von $s_d = 0,1\,cm$ folgt für den Fehler auf die Zeit: $s_t = T_B \cdot s_d$.

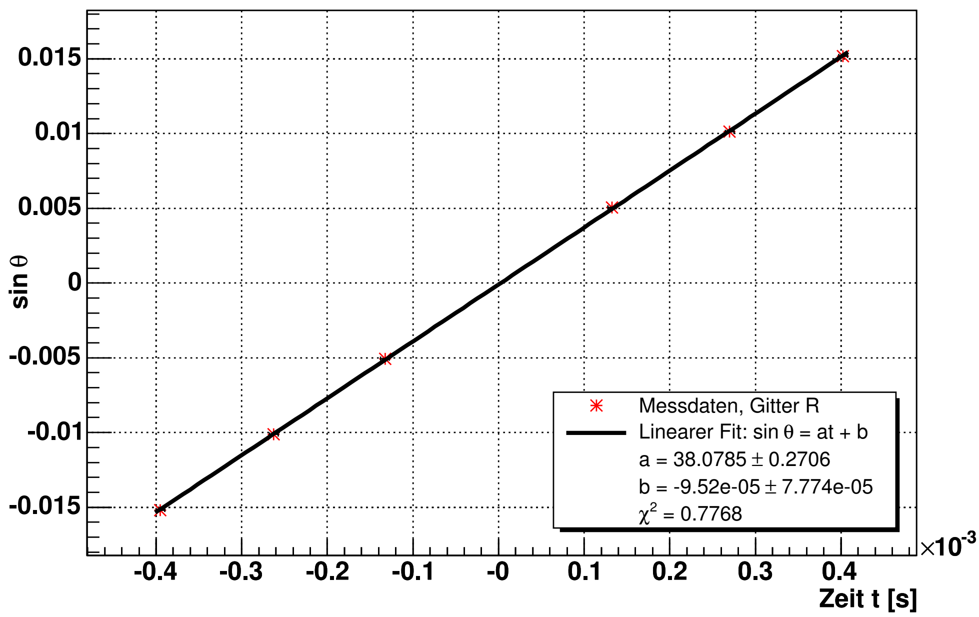

Abbildung 2: Eichung der Zeitachse durch Gitter R

Aus dem Fit (Abbildung 2) erhalten wir also für die Steigung $a = (38,0785 \pm 0,2706)\, s^{-1}$ und für den Y-Achsenabschnitt $b = (-9,5196 \pm 7,7737) \cdot 10^{-5}$. Zur Berechnung und zu den verwendeten Messdaten siehe Anhang A.2 ab Seite 23.

Als Fehler für $\sin\theta$ ergibt sich:

$$s_{\sin\theta} = \sqrt{\left(\frac{\partial \sin\theta}{\partial t}\right)^2 s_t^2 + \left(\frac{\partial \sin\theta}{\partial a}\right)^2 s_a^2 + \left(\frac{\partial \sin\theta}{\partial b}\right)^2 s_b^2} = \sqrt{a^2 s_t^2 + t^2 s_a^2 + s_b^2} \tag{10}$$

Mit dem nun bekannten Zusammenhang $\sin\theta = at + b$ lassen sich aus Gleichung 1 für die fünf Amplitudengitter die Gitterkonstanten K_i bestimmen. Dabei wird der lineare Zussammenhang

$$\sin\theta = \frac{\lambda}{K_i}\, m = a_i\, m \tag{11}$$

ausgenutzt und die Gitterkonstante K_i aus der durch Geradenanpassung ermittelten Steigung a_i berechnet. Für den Fehler von $K_i = \frac{\lambda}{a_i}$ gilt dabei:

$$s_{K_i} = K_i\, \frac{s_{a_i}}{a_i} \tag{12}$$

Als Ergebnis für die Gitterkonstanten der Gitter 1 bis 5 erhielten wir:

$$\begin{aligned}
K_1 &= (137,423 \pm 0,899)\,\mu m \\
K_2 &= (36,735 \pm 0,302)\,\mu m \\
K_3 &= (109,873 \pm 1,320)\,\mu m \\
K_4 &= (109,831 \pm 0,977)\,\mu m \\
K_5 &= (55,155 \pm 0,633)\,\mu m
\end{aligned}$$

Siehe dazu Abbildungen 3 bis 7 und zur Berechnung den Quelltext im Anhang A.2 ab Seite 23.

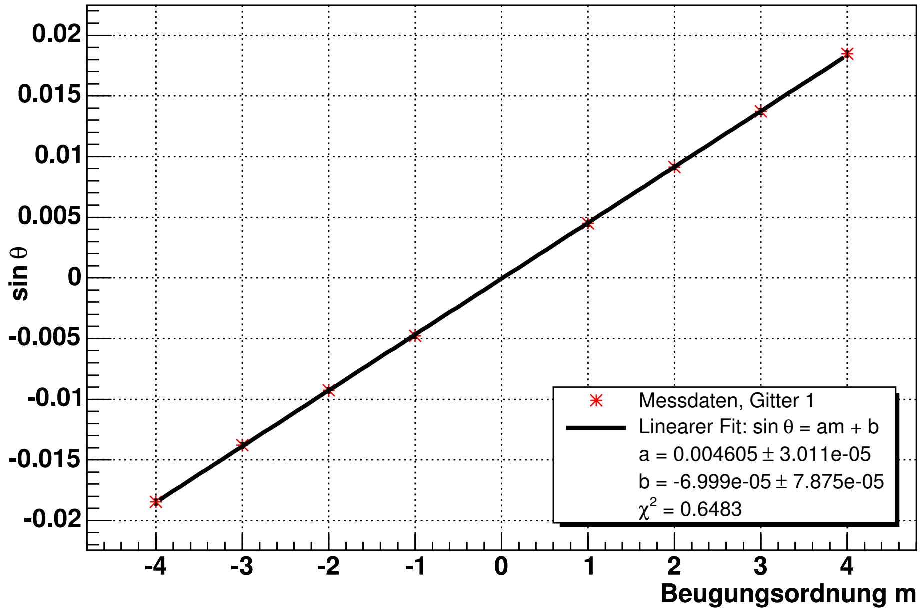

Abbildung 3: Geradenanpassung an die Messwerte zu Gitter 1

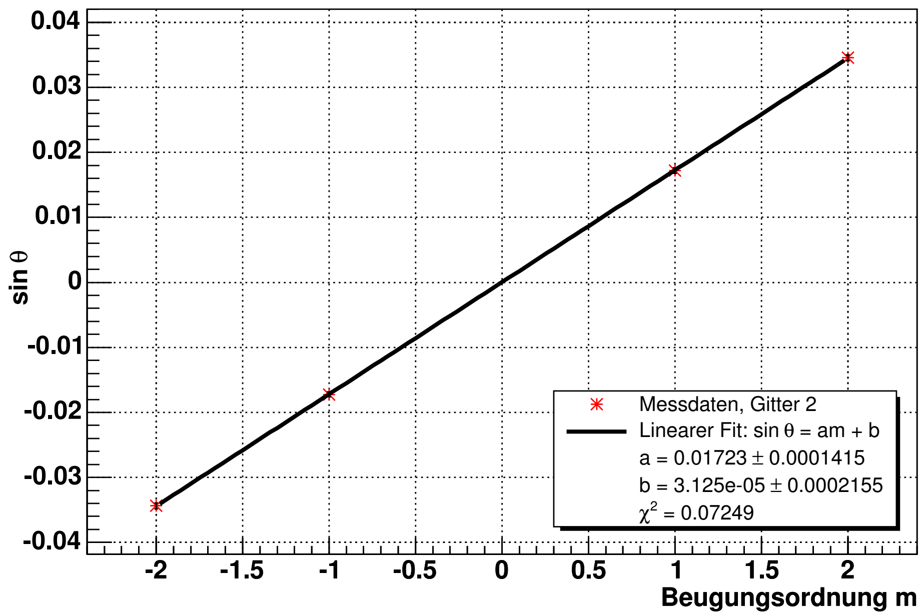

Abbildung 4: Geradenanpassung an die Messwerte zu Gitter 2

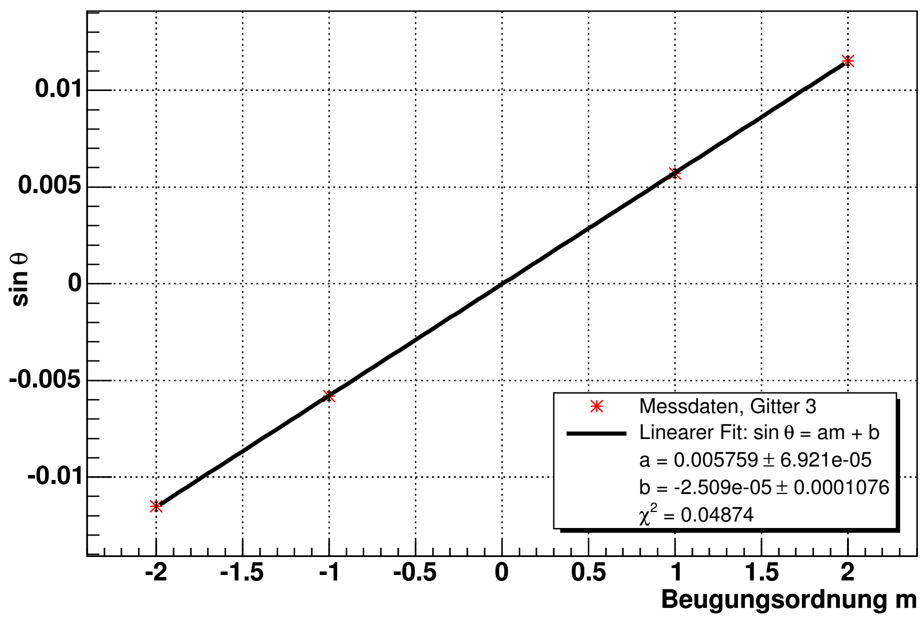

Abbildung 5: Geradenanpassung an die Messwerte zu Gitter 3

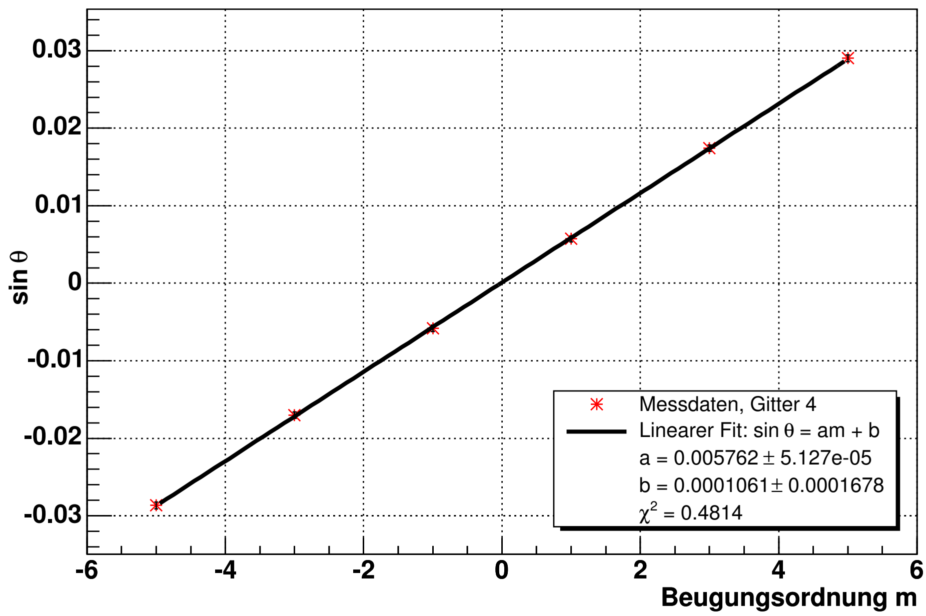

Abbildung 6: Geradenanpassung an die Messwerte zu Gitter 4

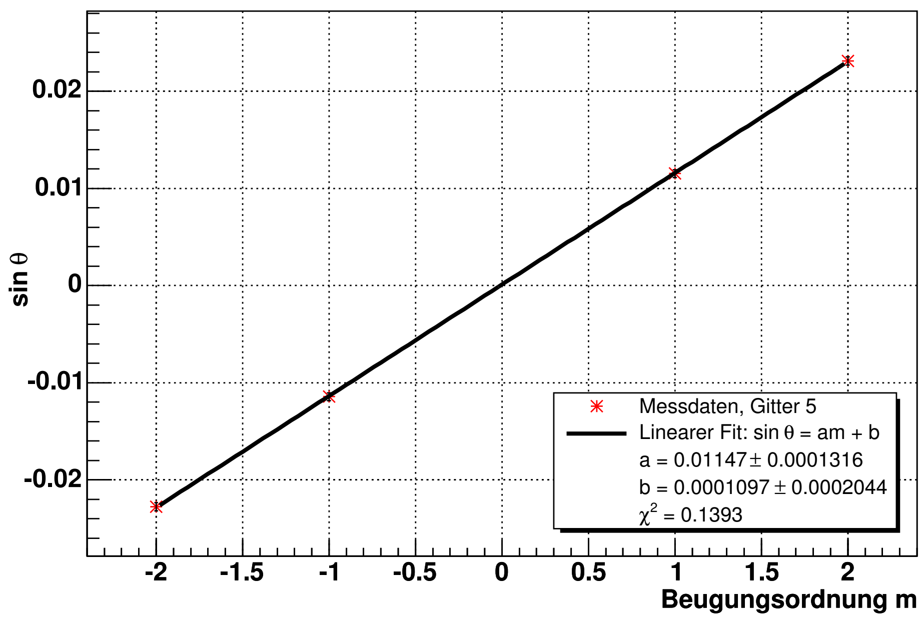

Abbildung 7: Geradenanpassung an die Messwerte zu Gitter 5

4.3. Aperturfunktion und Verhältnis Spaltbreite zu -abstand von Gitters 1

Zur Berechnung der Aperturfunktion g nutzten wir als Näherung die Fourierreihe:

$$g(x) = \sum_{j=0}^{m} \sqrt{I_j} \cos\left(\frac{x}{K} 2\pi j\right) \quad . \tag{13}$$

Als Intensitäten I_j verwendeten wir die Mittelwerte der gemessenen Beugungsmaxima. Dabei erhielten wir mit einem Ablesefehler von $0,1\,cm$ folgenden Intensitäten, die wir zur Berechnung der Aperturfunktion in Abbildung 8 verwendeten (siehe dazu auch Anhang A.3):

$$\begin{aligned} I_0 &= (7,675 \pm 0.035)V \\ I_1 &= (0,345 \pm 0.004)V \\ I_2 &= (0,216 \pm 0.004)V \\ I_3 &= (0,128 \pm 0.004)V \\ I_4 &= (0,060 \pm 0.004)V \\ I_5 &= (0,015 \pm 0.004)V \end{aligned}$$

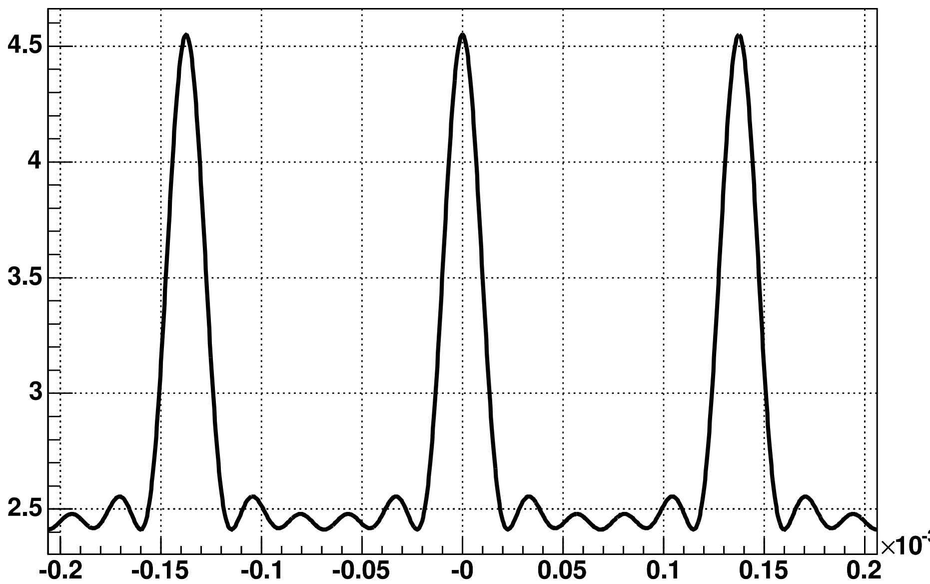

Abbildung 8: Drei Perioden der Aperturfunktion von Gitter 1

Um das Verhältnis von Spaltbreite zu Spaltabstand zu bestimmen, berechneten wir die Spaltbreite w numerisch aus der vollen Breite des halben Maximums der Aperturfunktion. Den Spaltabstand d erhielten wir mit der Gitterkonstante K aus $d = K - w$. Daraus lässt sich nun sofort das Verhältnis $v = \frac{w}{d}$ ausrechnen, wobei für den Fehler $s_v = v\,\frac{s_d}{d}$ gilt:

$$v = 0.1739 \pm 0.0013 \ .$$

4.4. Auflösungsvermögen der Gitter

Der Durchmesser des Laserstrahls betrug $d = (4 \pm 0,5)\,mm$. Mit der Gitterkonstante K des jeweiligen Gitters lässt sich daraus mit $N = \frac{K}{d}$ die Anzahl der ausgeleuchteten Gitterlinien bestimmen. Das Auflösungsvermögen a lässt sich dann durch

$$a = N\,m = \frac{K}{d}\,m \tag{14}$$

berechnen. Für den Fehler gilt dabei:

$$s_a = a\,\frac{s_N}{N} = a\sqrt{\left(\frac{s_d}{d}\right)^2 + \left(\frac{s_K}{K}\right)^2} \ . \tag{15}$$

Für die einzelnen Gitter erhielten wir folgende maximale Auflösungsvermögen:

$$\begin{aligned} a_1 &= 145,54 \pm 18,22 \\ a_2 &= 217,77 \pm 27,28 \\ a_3 &= 145,62 \pm 18,29 \\ a_4 &= 218,52 \pm 27,38 \\ a_5 &= 217,57 \pm 27,31 \end{aligned}$$

4.5. Vergleich der Phasengitter Messung mit der Raman-Nath-Theorie

Zunächst trugen wir den spannungsabhängigen Verlauf der einzelnen Beugungsordnungen auf. Dazu berechneten wir jeweils die Mittelwerte der Maxima aus postiver und negativer Ordnung.

Zum Vergleich mit der Raman-Nath-Theorie mussten wir alle Messwerte auf $I_0(0) = J_0(0) = 1$ normieren. Ausserdem benötigten wir den Umrechnungsfaktor b, der die Spannung der Generators in das passende Argument für die Besselfunktion verwandelt. Dazu ermittelten wir den x-Wert des ersten spannungsabhängigen Minimums im Verlauf der 0. Ordnung und teilten ihn durch den x-Wert des Minimums der Besselfunktion der selben Ordnung. Analog verfuhren wir bei dem x-Wert des ersten Maximums der 1. Ordnung. Der Mittelwert der beiden so gewonnenen Faktoren ergab unseren Umrechnungsfaktor $b = 0,14846$, sodass wir nun die quadrierten Besselfunktionen der jeweiligen Ordnungen in die Diagramme einfügen konnten. Wie man in Abbildungen 10 bis 14 sieht ist jeweils der Verlauf der beiden Kurven sehr ähnlich.

Die normierten Intensitäten I werden, wie schon erwähnt, aus dem Mittelwert der beiden gemessenen Intensitäten I_a und I_b berechnet und durch I_0 normiert; es gilt also:

$$I = \frac{I_a + I_b}{2\,I_0} \tag{16}$$

wobei I_0 den Fehler s_0 und sowohl I_a als auch I_b den Fehler s_{ab} haben. Für den Fehler von I erhält man daraus:

$$s_I = \sqrt{\left(\left(\frac{\partial I}{\partial I_a}\right)^2 + \left(\frac{\partial I}{\partial I_b}\right)^2\right) s_{ab}^2 + \left(\frac{\partial I}{\partial I_0}\right)^2 s_0^2} = \sqrt{\frac{s_{ab}^2}{2\,I_0^2} + I^2\,s_0^2} \tag{17}$$

Als Fehler für die Generatorspannung U nahmen wir einen Fehler von $s_U = 0,2\,Skt$ an.

Die Berechnungen und Auswertungen zu diesem Aufgabenteil finden sich im Anhang A.5 ab Seite 30.

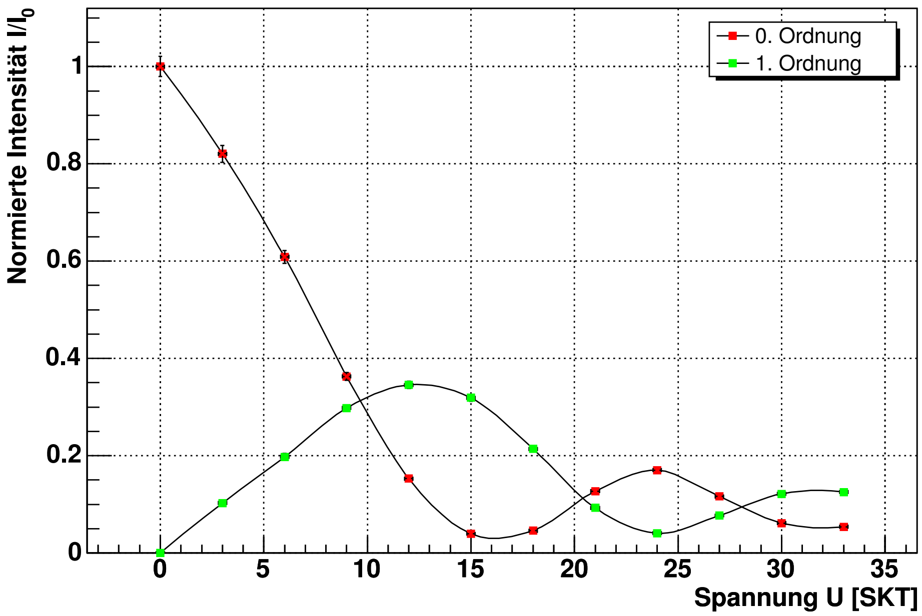

Abbildung 9: Spannungsabhängiger Verlauf der 0. und 1. Ordnung des Phasengitters

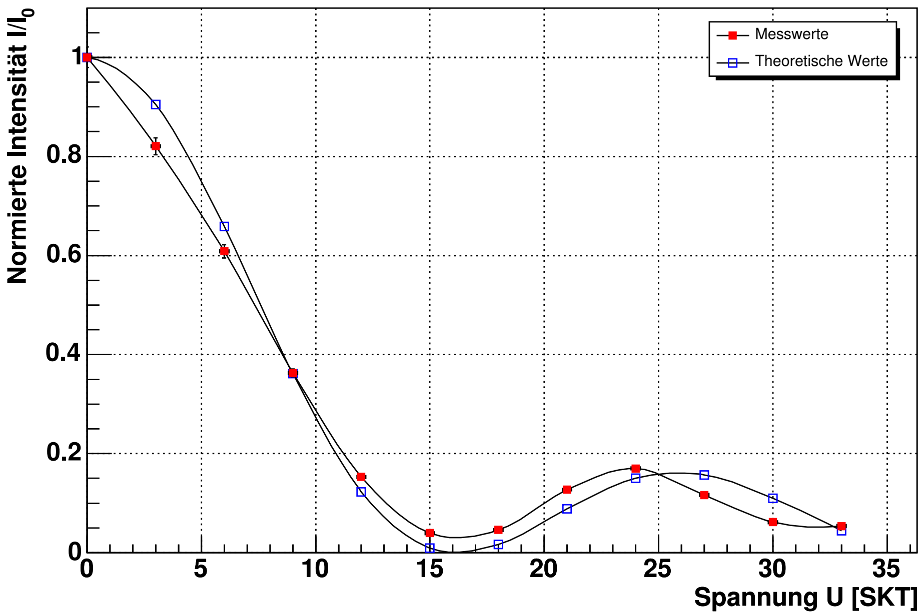

Abbildung 10: Vergleich der 0. Beugungsordnung mit der Besselfunktion J_0^2

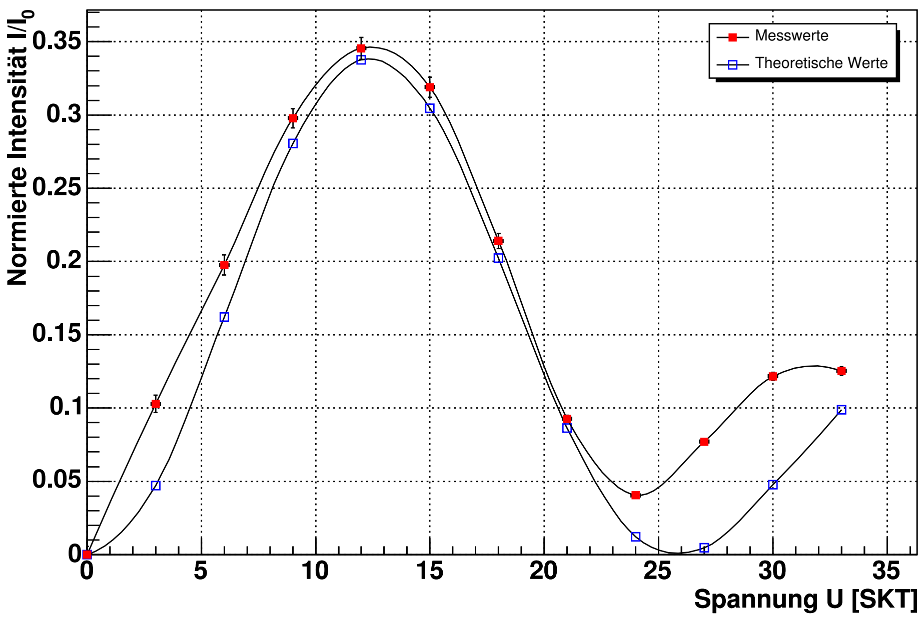

Abbildung 11: Vergleich der 1. Beugungsordnung mit der Besselfunktion J_1^2

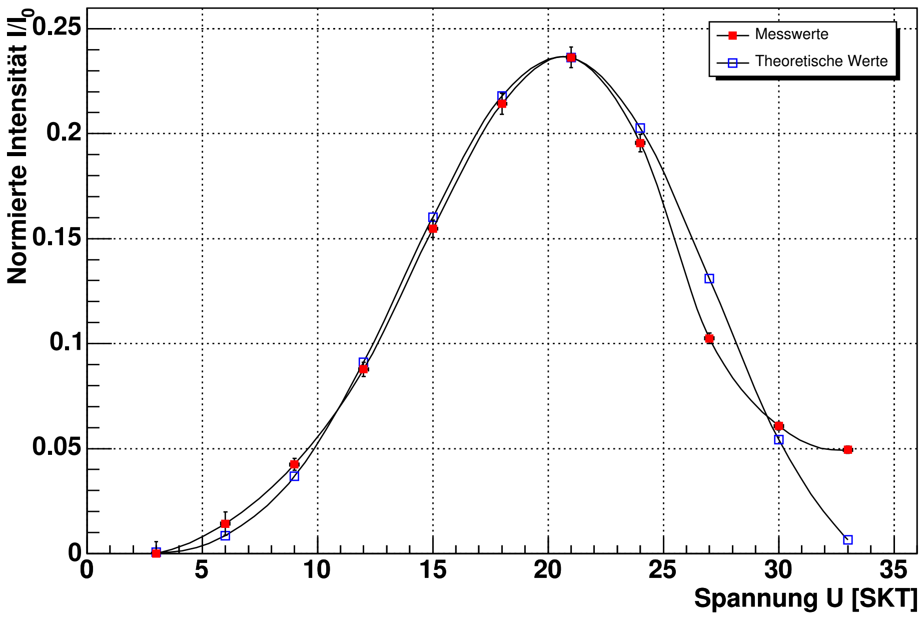

Abbildung 12: Vergleich der 2. Beugungsordnung mit der Besselfunktion J_2^2

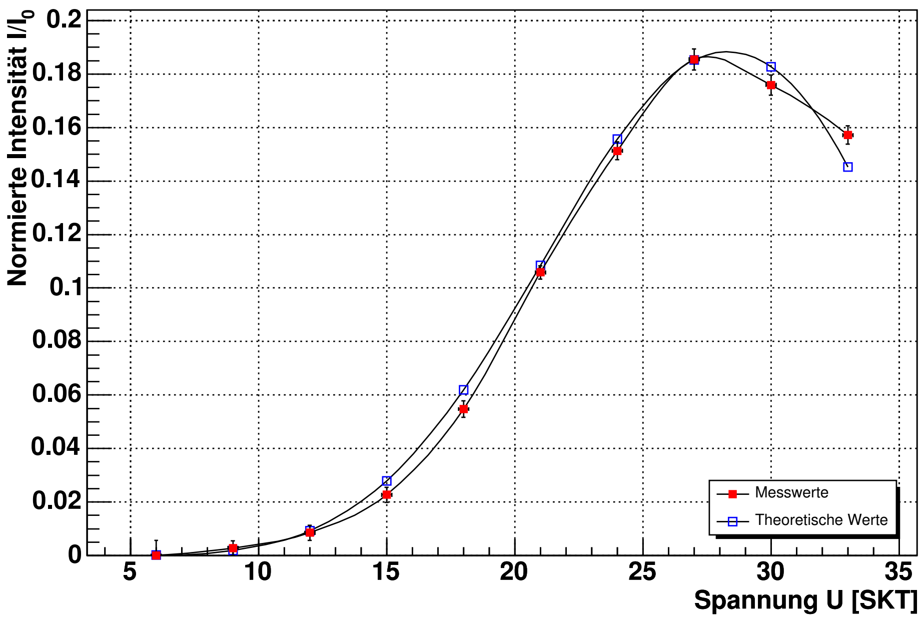

Abbildung 13: Vergleich der 3. Beugungsordnung mit der Besselfunktion J_3^2

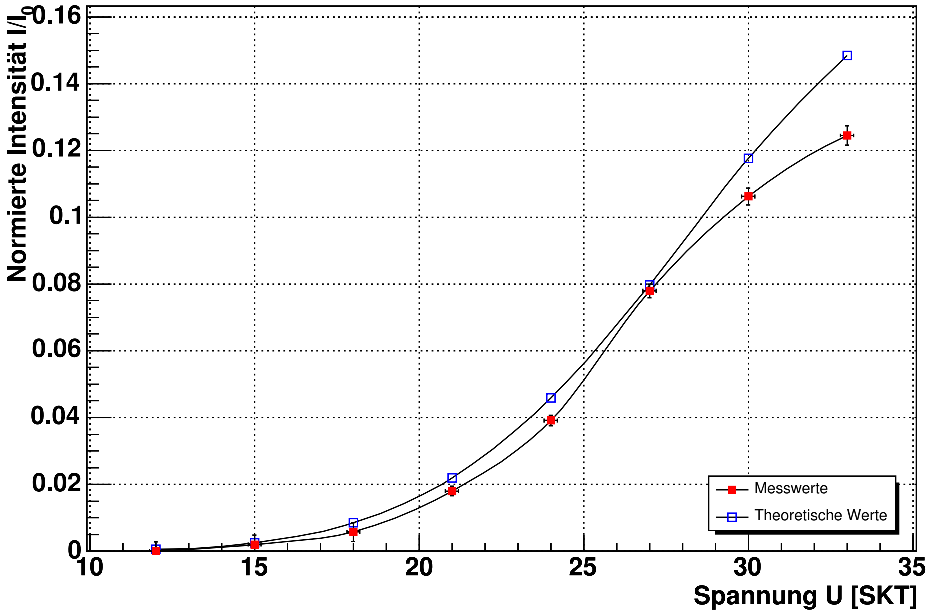

Abbildung 14: Vergleich der 4. Beugungsordnung mit der Besselfunktion J_4^2

4.6. Bestimmung der Schallwellenlänge

Wie schon in Abschnitt 4.2 nutzten wir das Referenzgitter R, um den Zusammenhang zwischen $\sin\theta$ und t zu bestimmen (siehe Abbildung 15).

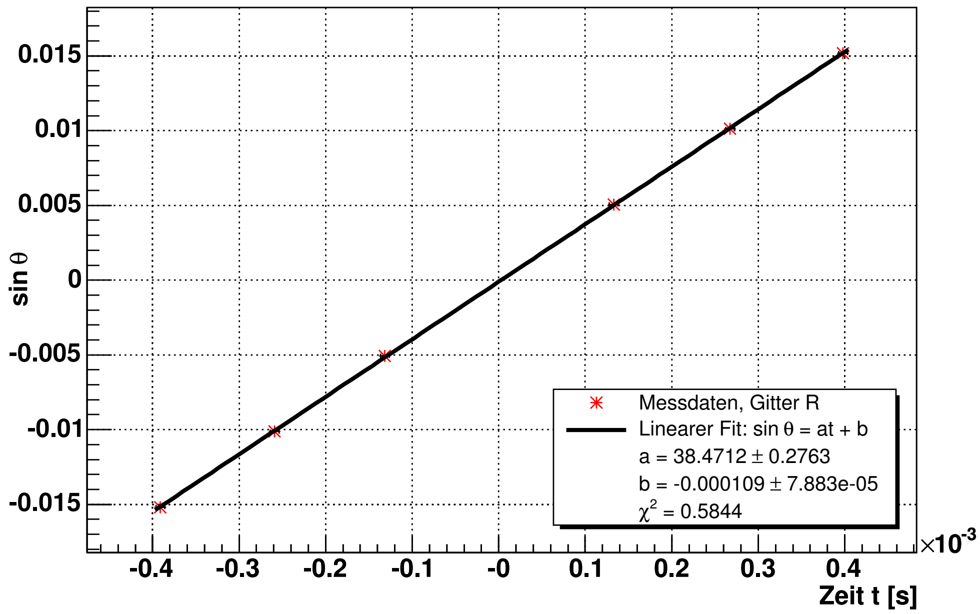

Abbildung 15: Eichung der Zeitachse durch Gitter R

Durch den Fit erhielten wir für $\sin\theta = at + b$ die Parameter $a = (38,471 \pm 0,276)\,s^{-1}$ und $b = (-1,090 \pm 0,788) \cdot 10^{-4}$.

Analog zur Bestimmung der Gitterkonstanten lässt sich die Schallwellenlänge Λ, durch Auftragen der zu den verschiedenen Ordnungen m des Phasengitters gehörenden Werte für $\sin\theta$ (siehe Abbildung 16), aus der Steigung $a = (24,960 \pm 0,2797) \cdot 10^{-4}$ der Ausgleichsgerade berechnen:

$$\Lambda = \frac{\lambda}{a} = (253,525 \pm 2,841)\,\mu m$$

Mit der Schallgeschwindigkeit $c_s = 1111\,m/s$ in Isooktan ergibt sich als Ultraschallfrequenz ν:

$$\nu = \frac{c_s}{\Lambda} = (4,382 \pm 0,049)\,MHz\ .$$

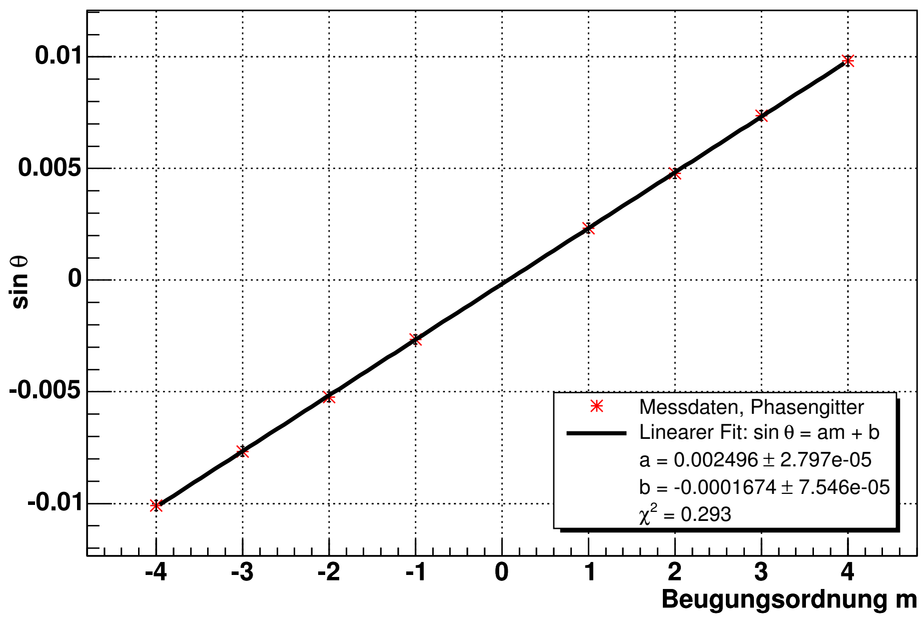

Abbildung 16: Geradenanpassung an die Messwerte zum Phasengitter

5. Zusammenfassung

- Für das Sinusgitter erhielten wir als Gitterkonstante:

$$K_S = (9,5950 \pm 0,2355) \cdot 10^{-7} m \ ,$$

die mit etwa 1σ Abweichung ziemlich gut mit dem angegebenen Wert $K_{S,ref} = 9,8425 \cdot 10^{-7} m$ übereinstimt.

- Für die fünf Amplitudengitter erhielten wir als Gitterkonstanten:

$$\begin{aligned} K_1 &= (137,423 \pm 0,899)\,\mu m \\ K_2 &= (36,735 \pm 0,302)\,\mu m \\ K_3 &= (109,873 \pm 1,320)\,\mu m \\ K_4 &= (109,831 \pm 0,977)\,\mu m \\ K_5 &= (55,155 \pm 0,633)\,\mu m \end{aligned}$$

- Für Gitter 1 ermittelten wir über die Aperturfunktion das Verhältnis v von Spaltbreite zu Spaltabstand:

$$v = 0.1739 \pm 0.0013$$

- Für das maximale Auflösungsvermögen der Gitter erhielten wir:

$$\begin{aligned} a_1 &= 145,54 \pm 18,22 \\ a_2 &= 217,77 \pm 27,28 \\ a_3 &= 145,62 \pm 18,29 \\ a_4 &= 218,52 \pm 27,38 \\ a_5 &= 217,57 \pm 27,31 \end{aligned}$$

- Durch unsere Messergebnisse mit dem Phasengitter ließ sich die Raman-Nath-Theorie gut verifizieren. Wie man an den Abbildungen 10-14 sieht, stimmen die Kurven der verschiedenen Beugungsordnungen gut mit den Quadraten der Besselfunktionen der jeweiligen Ordnung überein.
- Der für die Schallwellenlänge ermittelte Wert ist:

$$\Lambda = (253,525 \pm 2,841)\,\mu m$$

Fazit

Insgesamt erzielten wir bei diesem Versuch befriedigende Ergebnisse. Einzig die Auflösungsvermögen der Amplitudengitter waren aufgrund der hohen Ungenauigkeit bei der Messung des Strahldurchmessers nicht ganz so gut bestimmbar.

A. Quelltexte [1]

Wir verwendeten zur Auswertung die Programmiersprache Python mit dem Datenanalyse Framework ROOT.

A.1. Gitterkonstante des Sinusgitters (`konst_sinus.py`)

```python
#!/usr/bin/python
# -*- coding: iso-8859-1 -*-

from math import sqrt

# ------------------------------------------------------------------
# Berechnung der Gitterkonstant des Sinusgitters
# ------------------------------------------------------------------

l = 6.328e-7     # Wellenlänge des Lasers [m]

a = 4.65 / 100.  # Abstand zur ersten Beugungsordnung [m]
sa = 0.1 / 100.  # Fehler [m]
x = 5.3  / 100.  # Abstand zwischen Schirm und Gitter [m]
sx = 0.2 / 100.  # Fehler [m]

Ks_real = 1./1016./1000.  # Angegeberne Gitterkonstante [m]

# Berechnung der Gitterkonstante und ihres Fehlers
Ks = l * sqrt(x**2/a**2 + 1)
sKs = x*l/a**2 * sqrt( ((sx*a)**2 + (sa*x)**2) / (x**2 + a**2) )

# Abweichung vom angegebenen Wert
d = abs(Ks-Ks_real)
d_sigma = d/sKs

print "Abstand, 1.Ordnung [m]: %.4f +- %.4f" % (a,sa)
print "Abstand, Schirm-Gitter [m]: %.4f +- %.4f" % (x,sx)
print "Gitterkonstante [m]: %g +- %g" % (Ks, sKs)
print "Angegebene Gitterkonstante [m]: %g" % Ks_real
print "Abweichung vom Nennwert: %gm (%f sigma)" % (d,d_sigma)
```

Programmausgabe

```
Abstand, 1.Ordnung [m]: 0.0465 +- 0.0010
Abstand, Schirm-Gitter [m]: 0.0530 +- 0.0020
Gitterkonstante [m]: 9.59503e-07 +- 2.35482e-08
Angegebene Gitterkonstante [m]: 9.84252e-07
Abweichung vom Nennwert: 2.4749e-08m (1.050993 sigma)
```

[1] Siehe *http://www.physik.uni-freiburg.de/˜kolja/fp1/ultraschall/*

A.2. Gitterkonstanten der fünf Amplitudengitter (konst.py)

```
#!/usr/bin/python
# -*- coding: iso-8859-1 -*-

from array import array
from math import sqrt
from pickle import dump
from ROOT import gROOT, TCanvas, TGraphErrors, TF1, TLegend

gROOT.SetStyle("Plain")

# ------------------------------------------------------------------
# Bestimmung der Gitterkonstanten der Amplitudengitter
# ------------------------------------------------------------------

L = 6.328e-7  # Wellenlaenge des Lasers [m]
sd = 0.1      # Ablesefehler vom Papier [cm]

# Eichung der Zeitachse mit Gitter R -------------------------------

Kr = 1./80./100.  # Gitterkonstante von R [m]

# Gitter R: Abstand zur 0.Ordnung in cm bei 50us/cm (-3..+3)
rd = [ -7.9, -5.25, -2.65, 2.65, 5.4, 8.05 ]
rm = [   -3,    -2,    -1,    1,   2,    3 ]
rcount = len(rd)

rt_pro_cm = 5e-5
rt = [z * rt_pro_cm for z in rd]  # Umrechnung von cm nach s
srt = [sd * rt_pro_cm] * rcount   # Fehler auf die Zeit

# sin(theta) aus den Ordnungen und der Gitterkonstante berechnen
rsinth = [m * L / Kr for m in rm]

# beim Fit gibts Probleme, wenn hier 0 steht
srsinth = [1e-10] * rcount

# Plotte sin(theta) nach t
cr = TCanvas('cr', 'Eichung der Zeitachse')
cr.SetGrid()
gr = TGraphErrors(rcount, array('d',rt), array('d',rsinth),
                  array('d',srt), array('d',srsinth))
gr.SetTitle(';Zeit t [s];sin #theta')
gr.GetYaxis().CenterTitle()
gr.SetMarkerColor(2)
gr.SetMarkerStyle(3)
gr.Draw('AP')

```

```
# Linearer Fit
fr = TF1('fr', '[0]*x + [1]')
gr.Fit(fr, 'Q')
ar, sar = fr.GetParameter(0), fr.GetParError(0)
br, sbr = fr.GetParameter(1), fr.GetParError(1)

lr = TLegend(0.55, 0.14, 0.88, 0.34)
lr.SetFillColor(0)
lr.AddEntry(gr, 'Messdaten, Gitter R', 'p')
lr.AddEntry(fr, 'Linearer Fit: sin #theta = at + b', 'l')
lr.AddEntry(fr, 'a = %.4f #pm %.4f' % (ar,sar), '')
lr.AddEntry(fr, 'b = %.4g #pm %.4g' % (br,sbr), '')
lr.AddEntry(fr, '#chi^{2} = %.4g' % fr.GetChisquare(), '')
lr.Draw()
cr.Update()

print "Eichung, Gitter R:"
print 'sin(theta) = a * t + b'
print 'a: %.5f +- %.5f' % (ar, sar)
print 'b: %.5g +- %.5g' % (br, sbr)


# Gitterkonstanten von Gitter 1 bis 5 ------------------------------

# Gitter 1: Abstand zur 0.Ordnung in cm bei 50us/cm (-4..+4)
d1 = [ -9.65, -7.20, -4.82, -2.45, 2.40, 4.85, 7.25, 9.75 ]
m1 = [   - 4,    -3,    -2,    -1,    1,    2,    3,    4 ]

# Gitter 2: Abstand zur 0.Ordnung in cm bei 100us/cm (-2..+2)
d2 = [ -9.00, -4.50, 4.54, 9.10 ]
m2 = [    -2,    -1,    1,    2 ]

# Gitter 3: Abstand zur 0.Ordnung in cm bei 50us/cm (-2..+2)
d3 = [ -6.00, -3.00, 3.05, 6.10 ]
m3 = [    -2,    -1,    1,    2 ]

# Gitter 4: Abstand zur 0.Ordnung in cm bei 50us/cm (-3..+3)
d4 = [ -7.50, -4.45, -1.50, 1.53, 4.60, 7.65 ]
m4 = [    -5,    -3,    -1,    1,    3,    5 ]

# Gitter 5: Abstand zur 0.Ordnung in cm bei 100us/cm (-2..+2)
d5 = [ -5.95, -2.98, 3.05, 6.10 ]
m5 = [    -2,    -1,    1,    2 ]

# Zeit pro cm fuer Gitter 1-5
t_pro_cm = [5e-5, 1e-4, 5e-5, 1e-4, 1e-4]

d = [d1, d2, d3, d4, d5]
m = [m1, m2, m3, m4, m5]
```

```
# Listen c fuer GUI-Instanzen und K fuer die Gitterkonstanten
c, K = [], []

# Fitte Gitter 1-5 und berechne die Gitterkonstanten K[i]
for i in range(len(d)):

    di = d[i]
    count = len(di)
    tpcm = t_pro_cm[i]

    # Beugungsorgnungen m
    mi = m[i]
    smi = [0]*count

    # Zeit und Fehler in s
    ti = [z * tpcm for z in di]
    sti = tpcm * sd

    # sin(theta) und Fehler anhand der Eichung berechnen
    sinth = [ar*z+br for z in ti]
    ssinth = [sqrt((z*sar)**2 + (ar*sti)**2 + sbr**2) for z in ti]

    # Plotte sin(theta) nach m
    ci = TCanvas('c%d' % (i+1), '%d. Gitter' % (i+1))
    ci.SetGrid()

    gi = TGraphErrors(count, array('d',mi), array('d',sinth),
                      array('d',smi), array('d',ssinth))
    gi.SetTitle('; Beugungsordnung m; sin #theta')
    gi.GetYaxis().CenterTitle()
    gi.SetMarkerColor(2)
    gi.SetMarkerStyle(3)
    gi.Draw('AP')

    # Linearer Fit
    fi = TF1('f%d' % (i+1), '[0]*x + [1]')
    gi.Fit(fi, 'Q')
    a, sa = fi.GetParameter(0), fi.GetParError(0)
    b, sb = fi.GetParameter(1), fi.GetParError(1)

    l = TLegend(0.55, 0.14, 0.88, 0.34)
    l.SetFillColor(0)
    l.AddEntry(gi, 'Messdaten, Gitter %d' % (i+1), 'p')
    l.AddEntry(fi, 'Linearer Fit: sin #theta = am + b', 'l')
    l.AddEntry(fi, 'a = %.4g #pm %.4g' % (a,sa), '')
    l.AddEntry(fi, 'b = %.4g #pm %.4g' % (b,sb), '')
    l.AddEntry(fi, '#chi^{2} = %.4g' % fi.GetChisquare(), '')
    l.Draw()
```

```
    ci.Update()

    Ki = L/a       # Gitterkonstante [m]
    sKi = Ki*sa/a  # Fehler [m]

    print '\nGitter %d:' % (i+1)
    print 'a: %.5e +- %.5e,' % (a,sa),
    print 'b: %.5g +- %.5g' % (b,sb)
    print 'K [m]: %.5e +- %.5e' % (Ki,sKi)

    # Referenzen sichern, damit sie nicht entsorgt werden
    c += [(ci,gi,fi,l)]

    # fuege Ki und sKi in die Liste K der Gitterkonstanten ein
    K += [(Ki,sKi)]

# Schreibe die ermittelten Gitterkonstanten K1..K5 zur weiteren
# Verwendung (aufloesung.py) in eine Datei
dump(K, open('k.dat', 'w'))
```

Programmausgabe

```
Eichung, Gitter R:
sin(theta) = a * t + b
a: 38.07850 +- 0.27064
b: -9.5196e-05 +- 7.7737e-05

Gitter 1:
a: 4.60477e-03 +- 3.01148e-05, b: -6.9988e-05 +- 7.8754e-05
K [m]: 1.37423e-04 +- 8.98732e-07

Gitter 2:
a: 1.72259e-02 +- 1.41482e-04, b: 3.1251e-05 +- 0.00021554
K [m]: 3.67355e-05 +- 3.01722e-07

Gitter 3:
a: 5.75935e-03 +- 6.92097e-05, b: -2.5086e-05 +- 0.00010764
K [m]: 1.09873e-04 +- 1.32034e-06

Gitter 4:
a: 5.76159e-03 +- 5.12670e-05, b: 0.00010612 +- 0.00016784
K [m]: 1.09831e-04 +- 9.77282e-07

Gitter 5:
a: 1.14731e-02 +- 1.31625e-04, b: 0.00010971 +- 0.00020437
K [m]: 5.51549e-05 +- 6.32764e-07
```

A.3. Aperturfunktion von Gitter 1 (`apertur.py`)

```
#!/usr/bin/python
# -*- coding: iso-8859-1 -*-

from array import array
from math import sqrt, cos, pi
from ROOT import gROOT, TCanvas, TF1
from pickle import load

gROOT.SetStyle("Plain")

# ---------------------------------------------------------------------
# Bestimmung der Aperturfunktion von Gitter 1
# ---------------------------------------------------------------------

L = 6.328 * 1e-7  # Wellenlaenge des Lasers [m]

# Lade die in konst.py berechneten Werte der Gitterkonstanten
Ks = load(open('k.dat', 'r'))

# Gitterkonstante von Gitter 1 [m], Fehler [m]
K, sK = Ks[0]

# Messung der Intensitäten ------------------------------------------

# Die gemessenen Intensitäten [cm]: ( (I-, I+), [V/cm] )
messI = [ ( ( 15.35, 15.35 ), 0.50 ),    # 0. Ordnung
          ( (  6.90,  6.90 ), 0.05 ),    # 1. Ordnung
          ( (  4.50,  4.15 ), 0.05 ),    # 2. Ordnung
          ( (  2.84,  2.28 ), 0.05 ),    # 3. Ordnung
          ( (  1.18,  1.22 ), 0.05 ),    # 4. Ordnung
          ( (  0.31,  0.30 ), 0.05 ) ]   # 5. Ordnung

count = len(messI)
sIa = 0.1  # Ablesefehler [cm]

# Mittelwerte der Intensitäten [V]
I = [(z[0][0] + z[0][1])/2 * z[1] for z in messI]

# Fehler der Mittelwerte [V]
sI = [z[1]/sqrt(2) * sIa for z in messI]

print 'Intensitäten:'
for i in range(count):
    print 'I%d = (%.5f +- %.5f)V' % (i, I[i], sI[i])


# Erzeuge Aperturfunktion g ------------------------------------------
```

```
expr = 'sqrt([0])'
for i in range(1,count):
    expr += ' + sqrt([%d])*cos(x/[%d]*2*pi*%d)' % (i,count,i)

g = TF1('g', expr, -K/2, K/2)
g.SetParameters(array('d', I+[K]))
g.SetNpx(1000)


# Spaltbreite / Spaltabstand ----------------------------------------

# Bestimme Spaltbreite (volle Breite des halben Maximums, fwhm)
gmax, gmin = g.GetMaximum(), g.GetMinimum()
h = (gmax-gmin)/2 + gmin
fwhm = abs(g.GetX(h))*2
w, d, sd = fwhm, K-fwhm, sK

# Verhaeltnis und Fehler
v, sv = w/d, sd/d*v

print '\nSpaltbreite / Spaltabstand:'
print 'Spaltbreite w [m]: %e' % w
print 'Spaltabstand d [m]: %e +- %e' % (d,sd)
print 'Verhältnis w/d: %f +- %f' % (v,sv)


# Plot der Aperturfunktion ------------------------------------------

g.SetRange(-3*K/2, 3*K/2)
c = TCanvas('c', 'Aperturfunktion von Gitter 1')
c.SetGrid()
g.SetTitle('')
g.Draw()
c.Update()
```

Programmausgabe

```
Intensitäten:
I0 = (7.67500 +- 0.03536)V
I1 = (0.34500 +- 0.00354)V
I2 = (0.21625 +- 0.00354)V
I3 = (0.12800 +- 0.00354)V
I4 = (0.06000 +- 0.00354)V
I5 = (0.01525 +- 0.00354)V

Spaltbreite / Spaltabstand:
Spaltbreite w [m]: 2.035560e-05
Spaltabstand d [m]: 1.170670e-04 +- 8.987320e-07
Verhältnis w/d: 0.173880 +- 0.001335
```

A.4. Auflösungsvermögen (`aufloesung.py`)

```
#!/usr/bin/python
# -*- coding: iso-8859-1 -*-

from math import sqrt
from pickle import load

# -------------------------------------------------------------------
# Bestimmung der maximalen Auflösungsvermögen der fünf Gitter
# -------------------------------------------------------------------

# Lade die in konst.py berechneten Werte der Gitterkonstanten
K = load(open('k.dat', 'r'))

d  = 4e-3    # Durchmesser des Laserstrahls [m]
sd = 0.5e-3  # Fehler [m]

# Maximale Ordnung
m = [ 5, 2, 4, 6, 3 ]

for i, m, (Ki,sKi) in zip(range(len(K)), m, K):

    # Anzahl der Linien und deren Fehler
    N = d/Ki
    sN = N * sqrt((sd/d)**2 + (sKi/Ki)**2)

    # Maximales Auflösungsvermögen und dess Fehler
    a = N * m
    sa = a * sqrt((sd/d)**2 + (sKi/Ki)**2)

    print '\nGitter %d:' % (i+1)
    print 'N: %f +- %f' % (N, sN)
    print 'a: %f +- %f' % (a, sa)
```

Programmausgabe

```
Gitter 1:
N: 29.107287 +- 3.643387
a: 145.536435 +- 18.216936

Gitter 2:
N: 108.886607 +- 13.640176
a: 217.773213 +- 27.280352

Gitter 3:
N: 36.405521 +- 4.571670
a: 145.622083 +- 18.286682

Gitter 4:
N: 36.419638 +- 4.563974
a: 218.517829 +- 27.383846

Gitter 5:
N: 72.522988 +- 9.103475
a: 217.568963 +- 27.310424
```

A.5. Vergleich mit Raman-Nath-Theorie (raman_nath.py)

```
#!/usr/bin/python
# -*- coding: iso-8859-1 -*-

from array import array
from math import sqrt, pi
from ROOT import gROOT, TCanvas, TGraph, TGraphErrors, TLegend

# ROOT verfuegt leider nur ueber die Besselfunktionen J0 und J1,
# deshalb wird hier die Jn Implementierung von SciPy verwendet.
from scipy.special import jn

gROOT.SetStyle("Plain")

# -------------------------------------------------------------------
# Vergleich der Messwerte mit der Raman-Nath-Theorie
# -------------------------------------------------------------------


# Die Messdaten -----------------------------------------------------

sa = 0.1           # Ablesefehler in cm
I0 = 12.65 * 0.20  # 0.Intensitaetsmaximum, 0.Ordnung [V]
sI0 = sa * 0.20    # Fehler von I0

```

```
# 0.Ordnung: (Spannung, [Intensitaet], Einheit der Intensitaet in V/cm)
order0 = [ (  0, [12.65]*2, 0.20 ),
           (  3, [10.38]*2, 0.20 ),
           (  6, [ 7.70]*2, 0.20 ),
           (  9, [ 9.18]*2, 0.10 ),
           ( 12, [ 3.87]*2, 0.10 ),
           ( 15, [ 1.00]*2, 0.10 ),
           ( 18, [ 1.16]*2, 0.10 ),
           ( 21, [ 6.44]*2, 0.05 ),
           ( 24, [ 8.61]*2, 0.05 ),
           ( 27, [ 5.89]*2, 0.05 ),
           ( 30, [ 3.11]*2, 0.05 ),
           ( 33, [ 2.73]*2, 0.05 ) ]

# 1.Ordnung: (Spannung, (Intensitaet), Einheit der Intensitaet in V/cm)
order1 = [ (  0, ( 0.00, 0.00 ), 0.20 ),
           (  3, ( 1.30, 1.30 ), 0.20 ),
           (  6, ( 2.54, 2.46 ), 0.20 ),
           (  9, ( 7.70, 7.37 ), 0.10 ),
           ( 12, ( 9.18, 8.30 ), 0.10 ),
           ( 15, ( 8.90, 7.24 ), 0.10 ),
           ( 18, ( 6.48, 4.35 ), 0.10 ),
           ( 21, ( 6.26, 3.12 ), 0.05 ),
           ( 24, ( 2.35, 1.75 ), 0.05 ),
           ( 27, ( 3.12, 4.68 ), 0.05 ),
           ( 30, ( 5.74, 6.57 ), 0.05 ),
           ( 33, ( 6.15, 6.54 ), 0.05 ) ]

# 2.Ordnung: (Spannung, (Intensitaet), Einheit der Intensitaet in V/cm)
order2 = [ (  3, (  0.00,  0.00 ), 0.20 ),
           (  6, (  0.13,  0.23 ), 0.20 ),
           (  9, (  0.95,  1.20 ), 0.10 ),
           ( 12, (  2.00,  2.44 ), 0.10 ),
           ( 15, (  3.63,  4.20 ), 0.10 ),
           ( 18, (  5.47,  5.37 ), 0.10 ),
           ( 21, ( 13.48, 10.43 ), 0.05 ),
           ( 24, ( 12.83,  6.96 ), 0.05 ),
           ( 27, (  7.87,  2.50 ), 0.05 ),
           ( 30, (  4.24,  1.91 ), 0.05 ),
           ( 33, (  2.61,  2.39 ), 0.05 ) ]

# 3.Ordnung: (Spannung, (Intensitaet), Einheit der Intensitaet in V/cm)
order3 = [ (  6, (  0.00, 0.00 ), 0.20 ),
           (  9, (  0.04, 0.10 ), 0.10 ),
           ( 12, (  0.13, 0.30 ), 0.10 ),
           ( 15, (  0.35, 0.80 ), 0.10 ),
           ( 18, (  0.93, 1.84 ), 0.10 ),
           ( 21, (  4.11, 6.60 ), 0.05 ),
           ( 24, (  6.82, 8.49 ), 0.05 ),
```

```
           ( 27, ( 10.57, 8.20 ), 0.05 ),
           ( 30, ( 11.51, 6.29 ), 0.05 ),
           ( 33, (  9.90, 6.01 ), 0.05 ) ]

# 4.Ordnung: (Spannung, (Intensitaet), Einheit der Intensitaet in V/cm)
order4 = [ ( 12, ( 0.00, 0.00 ), 0.10 ),
           ( 15, ( 0.00, 0.10 ), 0.10 ),
           ( 18, ( 0.05, 0.24 ), 0.10 ),
           ( 21, ( 0.34, 1.48 ), 0.05 ),
           ( 24, ( 1.00, 2.96 ), 0.05 ),
           ( 27, ( 2.64, 5.25 ), 0.05 ),
           ( 30, ( 4.17, 6.58 ), 0.05 ),
           ( 33, ( 6.09, 6.51 ), 0.05 ) ]

orders = [order0, order1, order2, order3, order4]
ocount = len(orders)


# Berechnung der Mittelwerte und Fehler -----------------------------

g = []
U, sU, I, sI, count = [], [], [], [], []
for o in orders:
    counti = len(o)

    Ui = [z[0] for z in o]
    sUi = [0.2]*counti

    # Normierten Intensitäten und ihr Fehler
    Ii = [(z[1][0]+z[1][1])/2 * z[2] / I0 for z in o]
    sIi = [sqrt((sa*z[2]/I0)**2 / 2 + (i*sI0)**2) for z,i in zip(o,Ii)]

    U += [Ui]
    I += [Ii]
    count += [counti]
    sU += [sUi]
    sI += [sIi]

    gi = TGraphErrors(counti, array('d',Ui), array('d',Ii),
                      array('d',sUi), array('d',sIi))
    gi.SetTitle(';Spannung U [SKT];'
                'Normierte Intensit#ddot{a}t I/I_{0}')
    gi.SetMarkerColor(2)
    gi.SetMarkerStyle(21)
    gi.SetMarkerSize(0.7)

    g += [gi]


```

```
# Ermittle Umrechnungsfaktor ---------------------------------------

# Plotte 0. und 1. Ordnung
c01 = TCanvas('c01', '0. und 1. Ordnung')
c01.SetGrid()

g0 = g[0].Clone()
g0.Draw('ACP')

g1 = g[1].Clone()
g1.SetMarkerColor(3)
g1.Draw('CP')

l01 = TLegend(0.7, 0.8, 0.88, 0.88)
l01.SetFillColor(0)
l01.AddEntry(g0, '0. Ordnung', 'pl')
l01.AddEntry(g1, '1. Ordnung', 'pl')
l01.Draw()

c01.Update()

xg0min = 16.2   # Erstes Minimum der 0.Ordnung
xj0min = 2.4048 # Erstes Minimum von J0

xg1max = 12.4   # Erstes Maximum der 1.Ordnung
xj1max = 1.8412 # Erstes Maximum von J1

# Umrechungsfaktor
b = (xj0min/xg0min + xj1max/xg1max) / 2.

print 'Erstes Minimum der 0.Ordnung: %.1f' % xg0min
print 'Erstes Minimum von J0: %.4f' % xj0min
print 'Erstes Maximum der 1.Ordnung: %.1f' % xg1max
print 'Erstes Maximum von J1: %.4f' % xj1max
print 'Umrechnungsfaktor b: %f' % b

# Vergleiche Messkurven mit Besselfunktionen -----------------------

# Erstelle Graphen der Besselfunktionen
gt = []
for i in range(ocount):
    Iti = [(jn(i,z*b))**2 for z in U[i]]
    gti = TGraph(len(Iti), array('d', U[i]), array('d', Iti))
    gti.SetTitle(';Spannung U [SKT];'
                 'Normierte Intensit#ddot{a}t I/I_{0}')
    gti.SetMarkerColor(4)
    gti.SetMarkerStyle(25)
    gti.SetMarkerSize(0.7)
    gt += [gti]
```

```

# Plotte Besselfunktionen und Messungen der jeweiligen Ordnungen
c = [] # Liste fuer GUI Instanzen
ly = [ 0.8, 0.8, 0.8, 0.13, 0.13 ] # Y-Koordinatrn fuer Legenden
for i in range(ocount):
    ci = TCanvas('c%d' % i, '%d. Ordnung' % i)
    ci.SetGrid()

    gt[i].Draw('ACP')
    g[i].Draw('CP')

    l = TLegend(0.7, ly[i], 0.88, ly[i]+0.08)
    l.SetFillColor(0)
    l.AddEntry(g[i], 'Messwerte', 'pl')
    l.AddEntry(gt[i], 'Theoretische Werte', 'pl')
    l.Draw()

    ci.Update()

    c += [(ci,l)]
```

Programmausgabe

```
Erstes Minimum der 0.Ordnung: 16.2
Erstes Minimum von J0: 2.4048
Erstes Maximum der 1.Ordnung: 12.4
Erstes Maximum von J1: 1.8412
Umrechnungsfaktor b: 0.148464
```

A.6. Bestimmung der Schallwellenlänge (`schall.py`)

```
#!/usr/bin/python
# -*- coding: iso-8859-1 -*-

from array import array
from math import sqrt
from ROOT import gROOT, TCanvas, TGraphErrors, TF1, TLegend

gROOT.SetStyle("Plain")

# ---------------------------------------------------------------------
# Bestimmung der Schallwellenlaenge
# ---------------------------------------------------------------------

L = 6.328e-7  # Wellenlaenge des Lasers [m]
sd = 0.1      # Ablesefehler vom Papier [cm]

# Eichung der Zeitachse mit Gitter R --------------------------------
```

```

Kr = 1./80./100.  # Gitterkonstante von R [m]

# Gitter R: Abstand zur 0.Ordnung in cm bei 50us/cm (-4..+4)
rd = [ -7.82, -5.18, -2.63, 2.66, 5.35, 7.96 ]
rm = [    -3,    -2,    -1,    1,    2,    3 ]

rcount = len(rd)

rt_pro_cm = 5e-5
rt = [z * rt_pro_cm for z in rd]  # Umrechnung von cm nach s
srt = [sd * rt_pro_cm] * rcount   # Fehler auf die Zeit

# sin(theta) aus den Ordnungen und der Gitterkonstante berechnen
rsinth = [m * L / Kr for m in rm]

# beim Fit gibts Probleme, wenn hier 0 steht
srsinth = [1e-10] * rcount

# Plotte sin(theta) nach t
cr = TCanvas('cr', 'Eichung der Zeitachse')
cr.SetGrid()
gr = TGraphErrors(rcount, array('d',rt), array('d',rsinth),
                  array('d',srt), array('d',srsinth))
gr.SetTitle(';Zeit t [s];sin #theta')
gr.GetYaxis().CenterTitle()
gr.SetMarkerColor(2)
gr.SetMarkerStyle(3)
gr.Draw('AP')

# Linearer Fit
fr = TF1('fr', '[0]*x + [1]')
gr.Fit(fr, 'Q')
ar, sar = fr.GetParameter(0), fr.GetParError(0)
br, sbr = fr.GetParameter(1), fr.GetParError(1)

lr = TLegend(0.55, 0.14, 0.88, 0.34)
lr.SetFillColor(0)
lr.AddEntry(gr, 'Messdaten, Gitter R', 'p')
lr.AddEntry(fr, 'Linearer Fit: sin #theta = at + b', 'l')
lr.AddEntry(fr, 'a = %.4f #pm %.4f' % (ar,sar), '')
lr.AddEntry(fr, 'b = %.4g #pm %.4g' % (br,sbr), '')
lr.AddEntry(fr, '#chi^{2} = %.4g' % fr.GetChisquare(), '')
lr.Draw()
cr.Update()

print "Eichung, Gitter R:"
print 'sin(theta) = a * t + b'
print 'a: %.5f +- %.5f' % (ar, sar)
```

```
print 'b: %.5g +- %.5g' % (br, sbr)


# Bestimmung der Schallwellenlaenge ---------------------------------

# Abstand zur 0.Ordnung in cm bei 50us/cm (-4..+4)
d = [ -5.19, -3.93, -2.66, -1.32, 1.27, 2.54, 3.89, 5.16 ]
m = [    -4,    -3,    -2,    -1,    1,    2,    3,    4 ]

count = len(d)
tpcm = 5e-5    # Zeit pro cm
sm = [0]*count # Kein Fehler auf die Beugungsorgnungen m

# Zeit und Fehler in s
t = [z * tpcm for z in d]
st = tpcm * sd

# sin(theta) und Fehler anhand der Eichung berechnen
sinth = [ar*z+br for z in t]
ssinth = [sqrt((z*sar)**2 + (ar*st)**2 + sbr**2) for z in t]

# Plotte sin(theta) nach m
c = TCanvas('c', 'Phasengitter')
c.SetGrid()

g = TGraphErrors(count, array('d',m), array('d',sinth),
                 array('d',sm), array('d',ssinth))
g.SetTitle(';Beugungsordnung m;sin #theta')
g.GetYaxis().CenterTitle()
g.SetMarkerColor(2)
g.SetMarkerStyle(3)
g.Draw('AP')

# Linearer Fit
f = TF1('f', '[0]*x + [1]')
g.Fit(f, 'Q')
a, sa = f.GetParameter(0), f.GetParError(0)
b, sb = f.GetParameter(1), f.GetParError(1)

l = TLegend(0.55, 0.14, 0.88, 0.34)
l.SetFillColor(0)
l.AddEntry(g, 'Messdaten, Phasengitter', 'p')
l.AddEntry(f, 'Linearer Fit: sin #theta = am + b', 'l')
l.AddEntry(f, 'a = %.4g #pm %.4g' % (a,sa), '')
l.AddEntry(f, 'b = %.4g #pm %.4g' % (b,sb), '')
l.AddEntry(f, '#chi^{2} = %.4g' % f.GetChisquare(), '')
l.Draw()

c.Update()
```

```

Ls = L/a       # Schallwellenlaenge [m]
sLs = Ls*sa/a  # Fehler [m]

print '\nPhasengitter:'
print 'a: %.5e +- %.5e,' % (a,sa),
print 'b: %.5g +- %.5g' % (b,sb)
print 'Schallwellenlaenge Ls [m]: %.5e +- %.5e' % (Ls,sLs)

cs = 1111.       # Schallgeschwindigkeit in Isooktan [m/s]
nu = cs/Ls       # Ultraschallfrequenz [Hz]
snu = nu*sLs/Ls  # Fehler der Frequenz [Hz]

print 'Ultraschallfrequenz nu [MHz]: %f +- %f' % (nu/1e6, snu/1e6)
```

Programmausgabe

```
Eichung, Gitter R:
sin(theta) = a * t + b
a: 38.47122 +- 0.27625
b: -0.000109 +- 7.8828e-05

Phasengitter:
a: 2.49601e-03 +- 2.79719e-05, b: -0.00016741 +- 7.5459e-05
Schallwellenlaenge Ls [m]: 2.53525e-04 +- 2.84116e-06
Ultraschallfrequenz nu [MHz]: 4.382218 +- 0.049110
```