Versuch 232 **25. Februar 2022**

Michelson-Interferometer

Physikalisches Anfängerpraktikum II

Juan Provencio Lameiras Betreuer/in: Marcel Fischer

Inhaltsverzeichnis

1 Ziel des Versuches

In diesem Versuch werden wir mithilfe des Michelson-Interferometers grundlegende Eigenschaften von Licht und Optik untersuchen. Wir werden die Wellenlänge eines Lasers bestimmten, den Brechungsindex der Luft und die Kohärenzlänge einer LED ausrechnen. Hier werden wir mit der uns vertrauten Natur des Lichts arbeiten.

2 Grundlagen

2.1 Kohärenz

Kohärenz bei Licht bezieht sich auf die Eigenschaft, dass zwischen verschiedenen Wellenzügen eine konstante Phasenbeziehung gibt. So sendet zum Beispiel ein Laser kohärentes Licht raus, aber eine Leuchtdiode inkohärentes Licht, bei der die Phasenbeziehung nicht fest ist, sondern statistisch verteilt, weshalb das Interferenzmuster beim letzteren verschwindet. Um ein Interferenzmuster aus einer inkohärenten Lichtquelle zu erzeugen muss man das Licht dieser in Teilwellen aufspalten. Eine Methode dazu ist in Abbildung 1 dargestellt. Dabei benutzt man eine Lichtquelle aus der ein Wellenfront rauskommt. Dieser Wellenfront trifft nachher eine Doppellochblende wodurch nach dem Huygenschen Prinzip zwei Sekundärwellen mit gleicher Phase. Diese können dann miteinander interferieren.

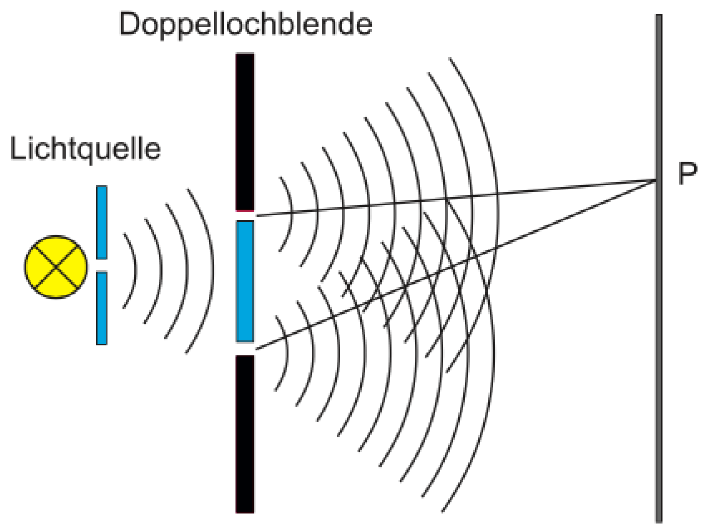


Abbildung 1: Erzeugung von kohärentem Licht aus inkohärenter Lichtquelle

Man kann die Kohärenzlänge bestimmen, indem man die Verfahrensgeschwindigkeit v und die Zeit τ bestimmt, in der die Atomen der Lichtquellen unabhängig voneinander

Wellenzüge aussenden.

$$L = v\tau \tag{1}$$

In unserem Fall gehen wir von einer statistisch verteilten Bandbreite aus. Diese hat die Form einer Gauss-Funktion, welche nach der Fourier-Transformation die Eigenschaften der Gauss-Funktion behält.

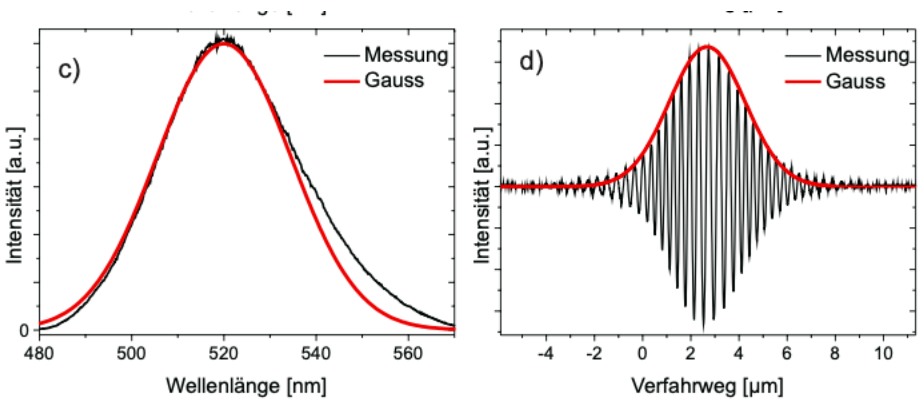


Abbildung 2: Wellenlängeverteilung und zugehöriges Interferogram

Wir werden ein zeitlich verteiltes Spektrum analysieren, deren Halbwertsbreite FWHM genau dieser Zeit τ entspricht. Daraus erhalten wir

$$L = v\text{FWHM} \tag{2}$$

2.2 Interferenz gleicher Neigung / Dicke

2.2.1 Interferenz gleicher Neigung

Am Michelson-Interferometer treffen zwei verschiedene Arten von Interferenz: gleicher Neigung oder gleicher Dicke. Bei Interferenz gleicher Neigung trifft ein Lichtbündel auf eine transparente planparallele Platte der Dicke d mit Brechungsindex n mit dem Einfalls- und Ausgangswinkel α, wie in Abbildung 3 dargestellt. Dabei wird ein Teil an der Oberfläche reflektiert und ein weiterer Teil wird an der Oberfläche gebrochen und an der anderen Seite reflektiert, so dass weitere Lichtstrahlen mit dem Winkel α bezüglich der Platte rauskommen. Zwischen zwei benachbarten Teilbündeln entsteht der Gangunterschied Δ

$$\Delta = n(\overline{AB} + \overline{BC}) - \overline{AD} \tag{3}$$

was sich nach einigen Vereinfachungen reduziert zu

$$\Delta = 2d\sqrt{n^2 - \sin^2\alpha} - \frac{\lambda}{2} \tag{4}$$

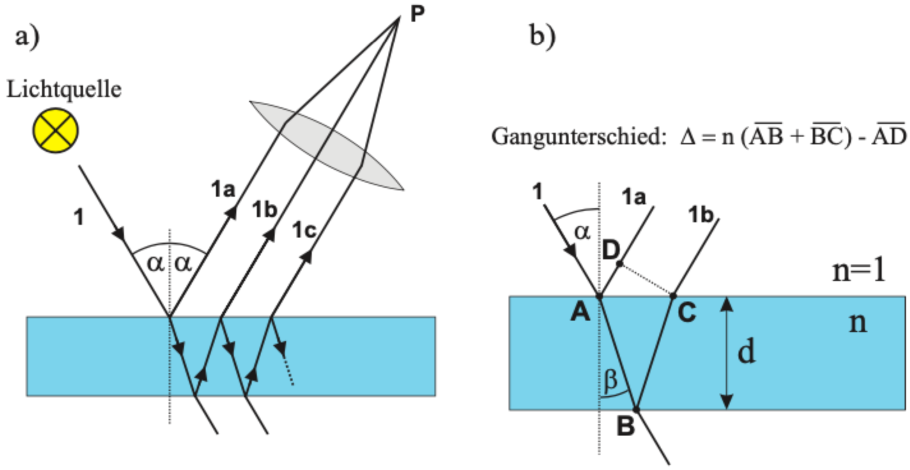


Abbildung 3: Interferenz gleicher Neigung

Alle Parallelen Lichtstrahlen, die durch eine Sammellinse laufen sammeln sich gemeinsam an einem Punkt P. Trifft nicht paralleles Licht auf die Platte auf, so werden alle "paar"weise parallele Strahlen zusammen in einem Punkt gesammelt, wie in Abbildung 4 c) dargestellt. Dadurch, dass wir eine "kreisförmige" Lichtverteilung betrachten, sammeln sich parallele Strahlen in Ringen, sogenannte Haiding'scher Ringe d).

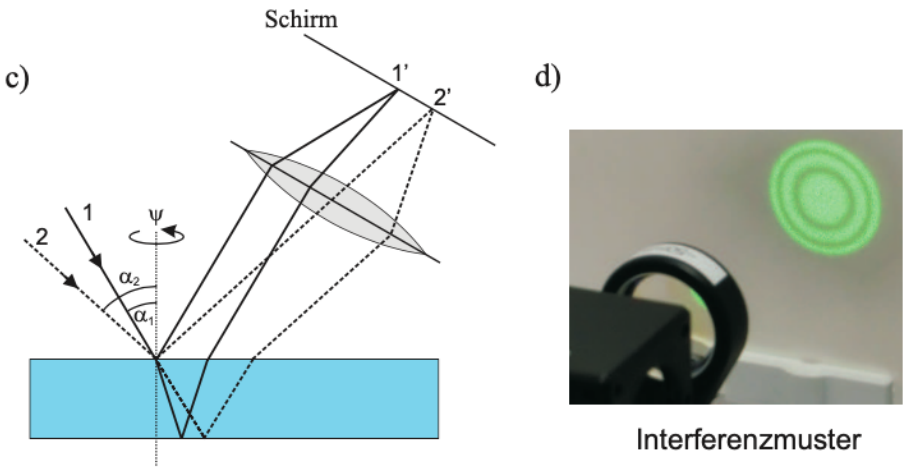


Abbildung 4: Interferenz bei verschiedenen nicht parallelen Strahlen und Haiding'sche Ringe

2.2.2 Interferenz gleicher Dicke

Bei Interferenz gleicher Dicke fällt paralleles Licht auf eine keilförmige Platte auf, so dass die reflektierten Teilbündeln nicht parallel zu einander rauskommen. So entsteht kein kreissymmetrisches Interferenzmuster, sondern fast parallele Interferenzstreifen.

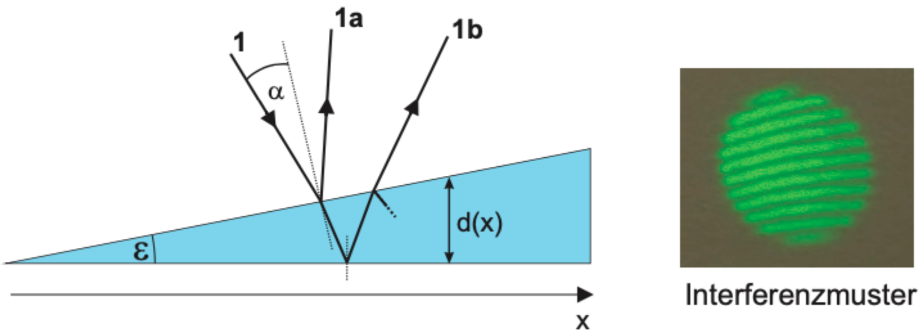


Abbildung 5: Interferenz gleicher dicke

Hier ist der Gangunterschied abhängig von der Form des Keils

$$\Delta \approx 2d(x)\sqrt{n^2 - \sin^2\alpha} - \frac{\lambda}{2} \tag{5}$$

2.3 Michelson-Interferometer

Der Michelson-Interferometer ist folgendermaßen aufgebaut

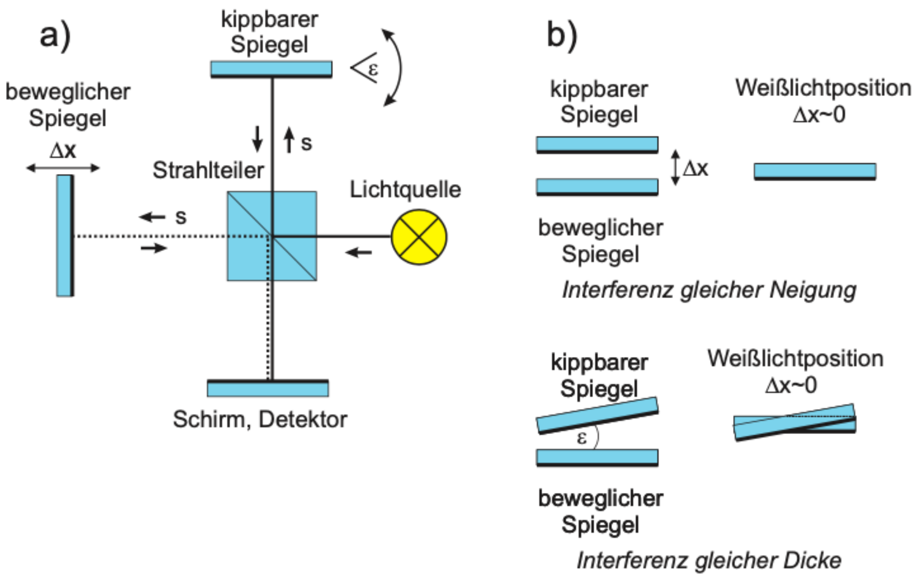


Abbildung 6: Aufbau Michelson-Interferometer

Wir beobachten die Ringmuster der Interferenz gleicher Neigung und konzentrieren uns auf das Zentrum, d.h. $\alpha = 0$ und nehmen einen Brechungsindex $n = 1$ an. Dann wird der Gangunterschied zu

$$\Delta\Big|_{\alpha=0} = 2\Delta x - \frac{\lambda}{2} \tag{6}$$

Eine Verrückung des beweglichen Spiegels um einen Abstand $\Delta x = \frac{3\lambda}{4}$ führt dazu, dass es eine neue Interferenzordnung erscheint. Bei der Interferenz gleicher Dicke entspricht eine Verrückung von $\Delta x = \frac{\lambda}{2}$ eine neue Interferenzordnung. Insgesamt wandern Δm Interferenzstreifen an einer Markierung vorbei und jeder Streifen entspricht einer Änderung von $\Delta x = \frac{\lambda}{2}$, weshalb wir die Wellenlänge anhand dieser zwei Größen bestimmen können als

$$\lambda = \frac{2\Delta x}{\Delta m} \tag{7}$$

Zur Berechnung des Brechungsindex der Luft ist ein ähnliches Prinzip erforderlich. Eine Änderung des Drucks in der Küvette entspricht einer Änderung der optischen Weglänge. Mit einer Länge d der Küvette ist der Gangunterschied

$$\Delta = 2d\Delta n \tag{8}$$

und mit $\Delta = \lambda\Delta m$, und $\Delta n = n_0 - 1$ plus einige Vereinfachungen erhalten wir

$$n_0 = \frac{\lambda}{2d}\Delta m(b) + 1 \qquad \frac{n_0 - 1}{n(p) - 1} = \frac{p_0 T}{p T_0} \tag{9}$$

$$= \frac{\lambda}{2d}\frac{\Delta m}{p}\frac{p_0 T}{T_0} \tag{10}$$

In diesem Fall sind p_0, T_0 die Normalbedingungen bei $T_0 = 273,15\,\mathrm{K}$ und $p_0 = 760\,\mathrm{Torr}$

3 Versuchsaufbau

3.1 Materialen und Geräte

- Michelson Interferometer
- Laser, Leuchtdiode
- Thermometer
- Vakuumpumpe

3.2 Aufbau

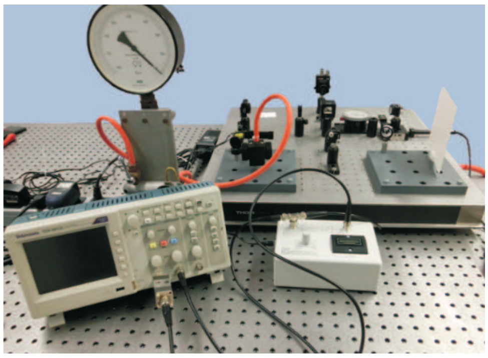


Abbildung 7: Aufbau nach Praktikumsskript

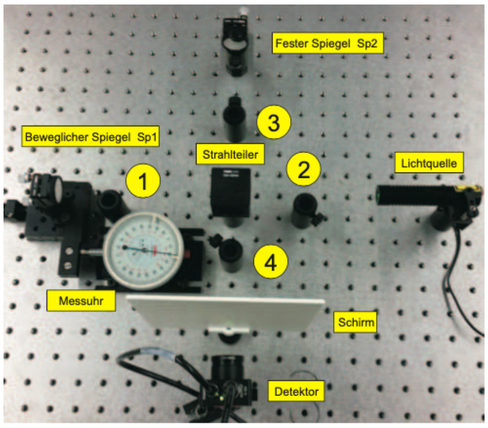
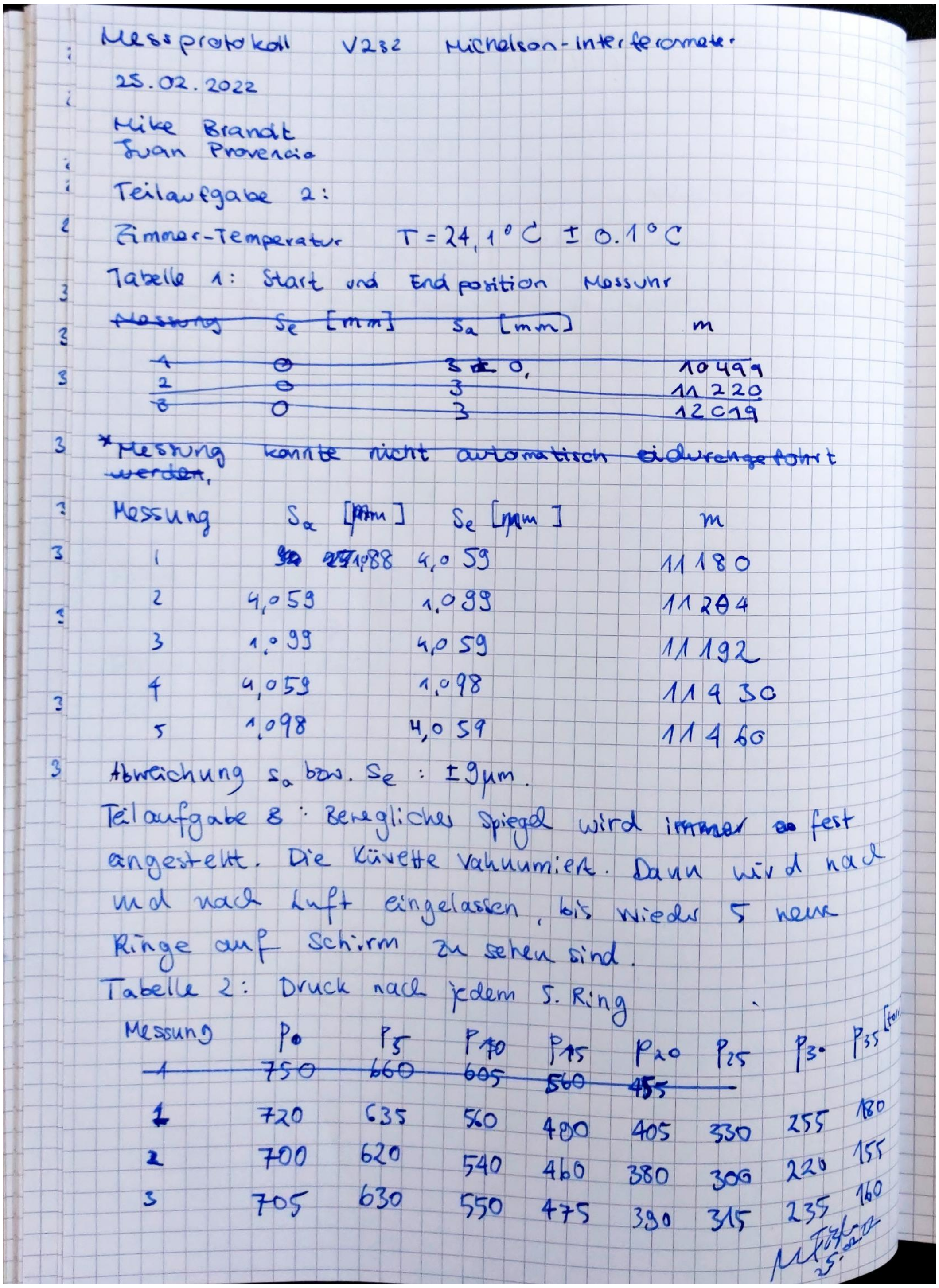


Abbildung 8: Beschriftung des Aufbaus

4 Messung und Auswertung

4.1 Messprotokoll


Messprotokoll V232 Michelson-Interferometer

25.02.2022

Mike Brandt
Juan Provencio

Teilaufgabe 2:

Zimmer-Temperatur $T = 24{,}1\,°C \pm 0.1\,°C$

Tabelle 1: Start und End position Messuhr

~~Messung~~	s_e [mm]	s_a [mm]	m
~~1~~	~~0~~	~~3 0,~~	~~10499~~
~~2~~	~~0~~	~~3~~	~~11220~~
~~3~~	~~0~~	~~3~~	~~12019~~

*~~Messung konnte nicht automatisch durchgeführt werden.~~

Messung	s_a [mm]	s_e [mm]	m
1	1,088	4,059	11180
2	4,059	1,099	11204
3	1,099	4,059	11192
4	4,059	1,098	11430
5	1,098	4,059	11460

Abweichung s_a bzw. s_e : $\pm 9\,\mu m$.

Teilaufgabe 8: Bewegliches Spiegel wird ~~immer~~ fest angestellt. Die Küvette vakuumiert. Dann wird nach und nach Luft eingelassen, bis wieder 5 neue Ringe auf Schirm zu sehen sind.

Tabelle 2: Druck nach jedem 5. Ring

Messung	p_0	p_5	p_{10}	p_{15}	p_{20}	p_{25}	p_{30}	p_{35} [torr]
~~1~~	~~750~~	~~660~~	~~605~~	~~560~~	~~455~~			
1	720	635	560	480	405	330	255	180
2	700	620	540	460	380	306	220	155
3	705	630	550	475	390	315	235	160

25.02.22

Abbildung 9: Messprotokoll

4.2 Auswertung

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

4.2.1 Messung der Wellenlänge

Nach dem der Aufbau richtig kalibriert worden ist, begeben wir uns der Aufgabe die Wellenlänge des Lasers zu bestimmen.

Dafür ist uns gemäß Gleichung (7), dass wir dies mittels des Abstandunterschieds der Messuhr und der gemessenen Anzahl an Interferenzstreifen. Im Versuch haben wir verschiedene Messungen durchgeführt und wir werden die Wellenlänge über den Mittelwert bestimmen. Es ergibt sich

$$\lambda = 2\frac{\Delta s}{m} = 524,7(2,6) \cdot 10^{-9}\,\mathrm{m} \tag{11}$$

Für den Fehler wurde die Gaussche Fehlerentwicklung benutzt, wobei für Δs wurde anhand der Größenordnungen des systematischen Fehlers im Vergleich zum statistischen Fehler bemerkt, dass das erste keine beeinflussende Rolle spielt. Der Fehler der Wellenlänge ist insofern rein statistisch.

4.2.2 Messung des Brechungsindex von Luft

Zur Bestimmung des Brechungsindex von Luft ist das Verhältnis der vorbeigelaufenden Interferenzstreifen und des Drucks benötigt. Dafür haben wir drei Messungen durchgeführt und jeweils mithilfe von einer linearen Anpassung an die Messwerten die Steigung der Geraden a_i und daraus einen Mittelwert a bestimmt.

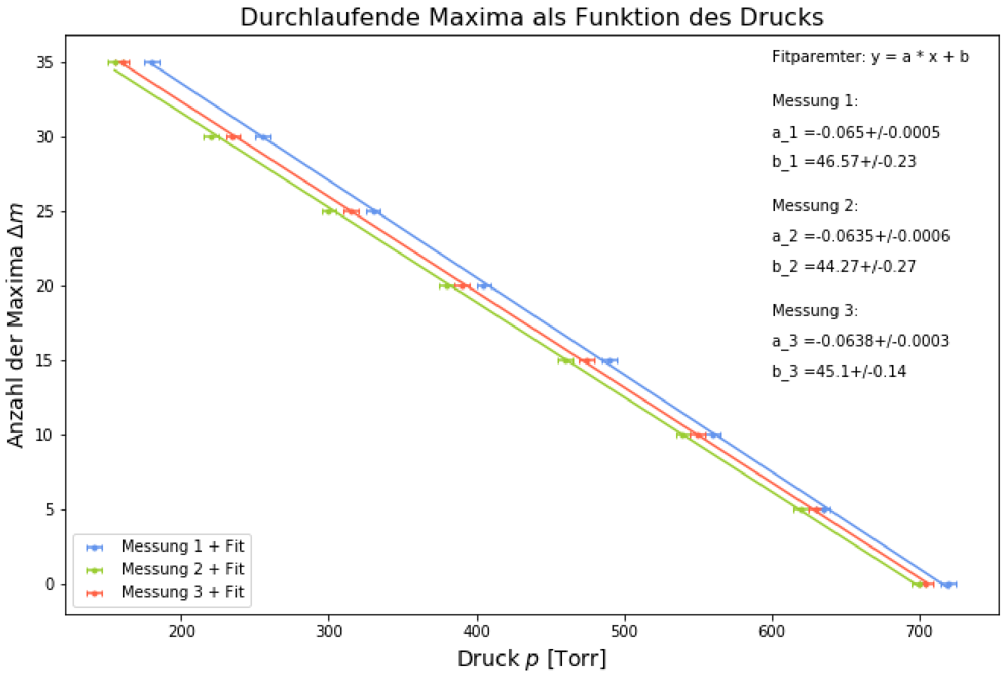


Diagramm 1: Vorgegangene Interferenzstreifen als Funktion des Drucks

Gemäß Gleichung (**??**) ist dann der Brechungsindex

$$n_0 = \frac{\lambda}{2d} \frac{p_0 T}{T_0} a + 1 = 1,000(8) \tag{12}$$

In diesem Fall haben wir den Fehler mittels des relativen Fehlers berechnet als

$$\Delta n_0 = n_0 \sqrt{\left(\frac{\Delta a}{a}\right)^2 + \left(\frac{\Delta T}{T}\right)^2 + \left(\frac{\Delta d}{d}\right)^2 + \left(\frac{\Delta \lambda}{\lambda}\right)^2} \tag{13}$$

4.2.3 Messung der Kohärenzlänge der Leuchtdiode

Zur Bestimmung der Kohärenzlänge der LED wurde der Aufbau so eingestellt, dass man einen gaussförmigen Verlauf erkennen könnte. Hier ist unsere Messung einigermaßen gelungen

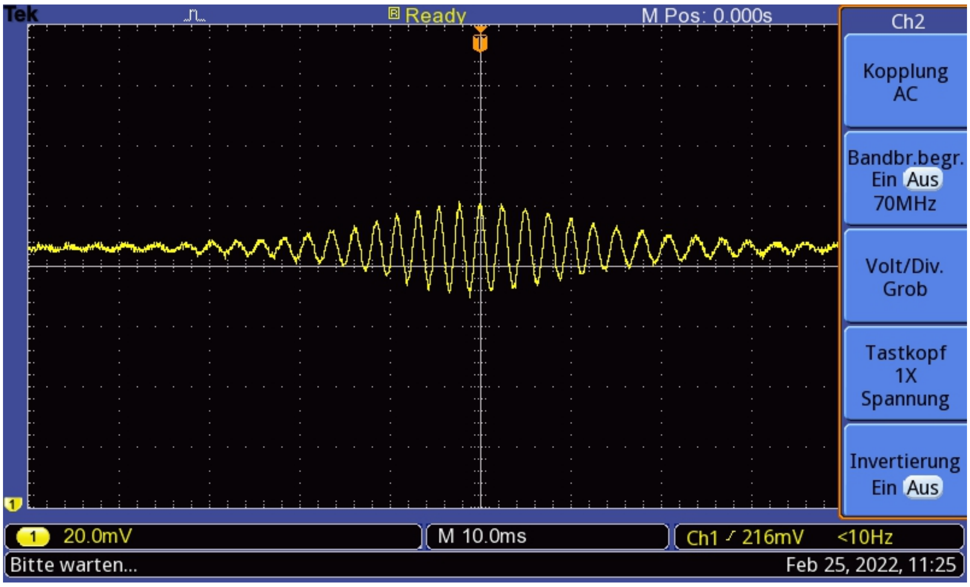


Abbildung 10: Gaussförmiger Verlauf am Oszilloskop (Eigene Messung)

allerdings war die Amplitude sehr klein, weshalb wir nach Angaben des Tutors die Messung unserer Partnergruppe untersuchen werden, welche einen klareren gausschen Verlauf vorzeigt.

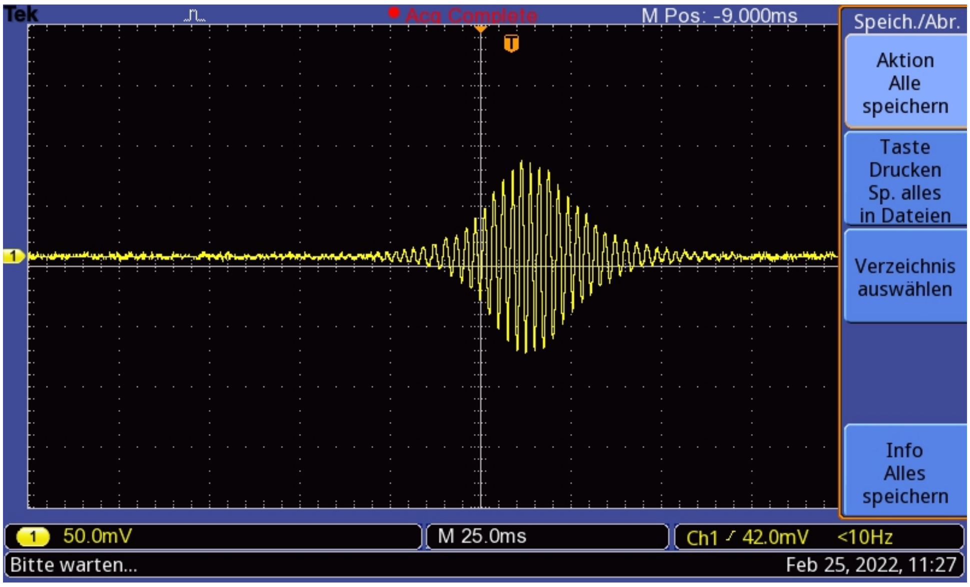


Abbildung 11: Gaussförmiger Verlauf am Oszilloskop (Fremde Messung)

Unsere Auswertung wird dann wie gesagt mit der fremden Messung durchgeführt, und am Ende werden wir zusätzlich die Ergebnisse der beiden vergleichen.

Als erstes stellen wir den Verlauf der Kurve auf Python wieder graphisch dar:

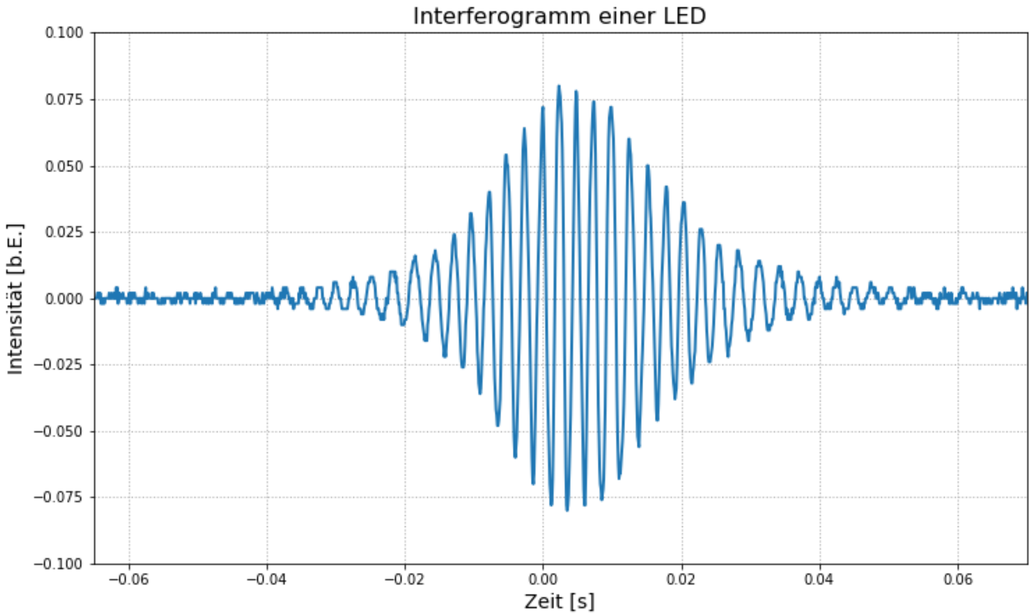


Diagramm 2: Interferogram einer LED

Zur Bestimmung der Parametern einer Gaussverteilung stehen uns zwei Möglichkeiten zur Verfügung: Als erstes können wir manuell die Parametern einer Gauss-Funktion einstellen und den best möglichen Fit abschätzen. Bei der zweiten Methode werden die Peaks rausgesucht und darauf eine Gauss-Kurve angepasst.

Die Ergebnisse bei der ersten Methode sind folgende:

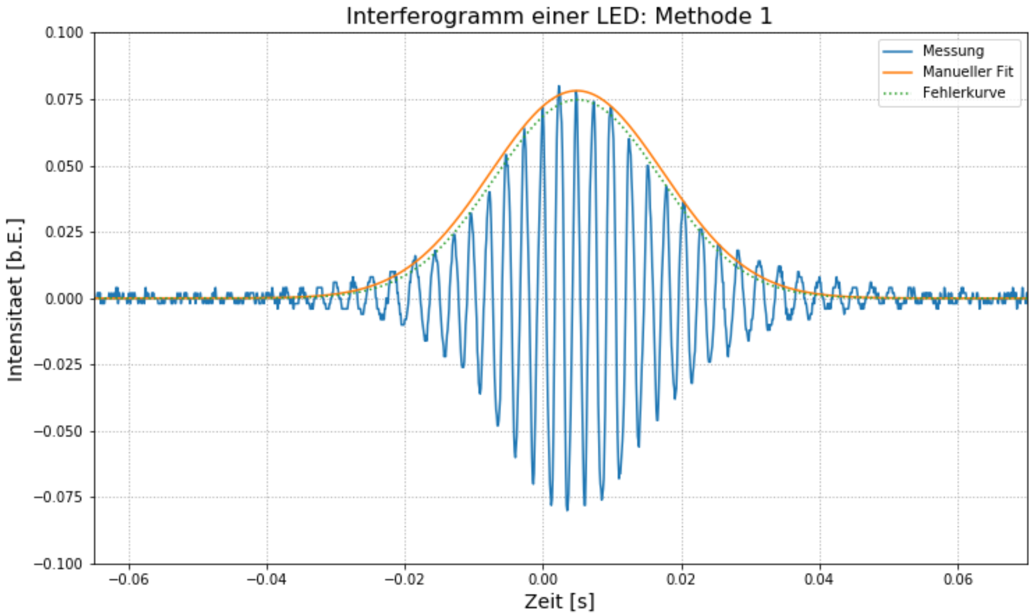


Diagramm 3: Interferogram einer LED: Fit mit manueller Anpassung

Dafür haben wir für eine Funktion der Form

$$g(t) = \frac{a}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(t-\mu)^2}{2\sigma^2}} \tag{14}$$

folgende Parameter eingestllt:

$$a_{4_1} = 0,00245(20) \tag{15}$$
$$\mu_{4_1} = 0,0049(1) \tag{16}$$
$$\sigma_{4_1} = 0,0125(5) \tag{17}$$

Besonders relevant ist hier die Breite der Verteilung σ. Wir haben für die Abschätzung eines Fehlers eine weitere Kurve angepasst, die als Fehlerkurve dienen soll.

Mit der zweiten Methode haben wir folgende Anpassung erhalten:

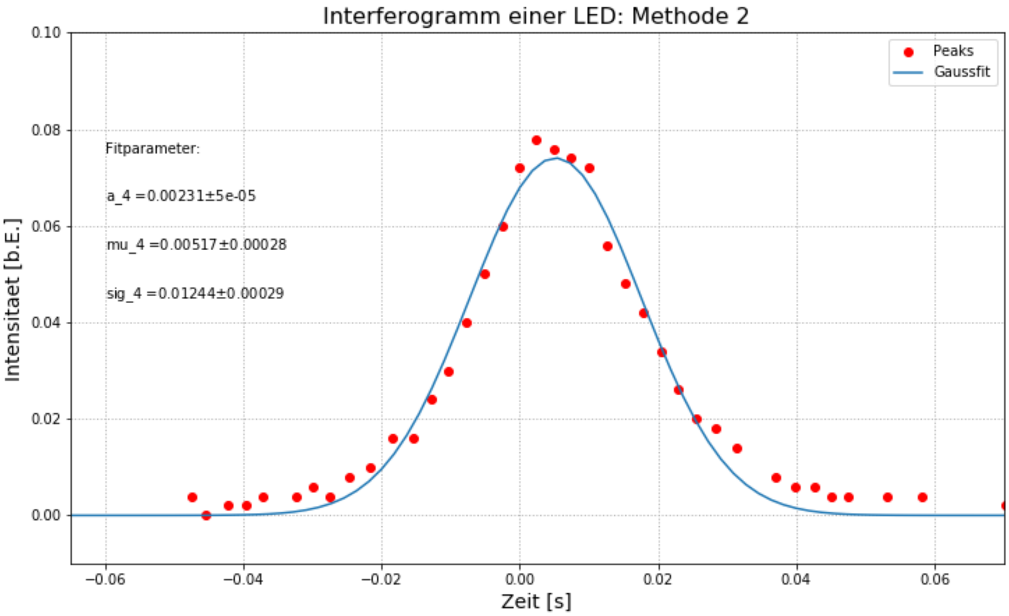


Diagramm 4: Interferogram einer LED: Fit mit Kurvenanpassung

$$a_{4_2} = 0,00231(5) \tag{18}$$
$$\mu_{4_2} = 0,00517(28) \tag{19}$$
$$\sigma_{4_2} = 0,01244(29) \tag{20}$$

Mit der Breite dieser Verteilung bestimmen wir hier die Halbwertsbreite und daraus die Kohärenzlänge der LED gemäß Gleichung (**??**)

$$\mathrm{FWHM}_{4_1} = 0,0294(12)\,\mathrm{s} \tag{21}$$

und somit die Kohärenzlänge

$$L_1 = v \cdot \mathrm{FWHM}_{4_1} = 2,94(12) \cdot 10^{-6}\,\mathrm{m} \tag{22}$$

Für die zweite Methode erhalten wir

$$\mathrm{FWHM}_{4_2} = 0,0293(7)\,\mathrm{s} \tag{23}$$

und

$$L_2 = 2,93(7) \cdot 10^{-6}\,\mathrm{m} \tag{24}$$

Vollständigkeitshalber vergleichen wir mit unserer eigenen Messung. Als Vergleih stellen wir beide Interferograme überlagert auf Python dar:

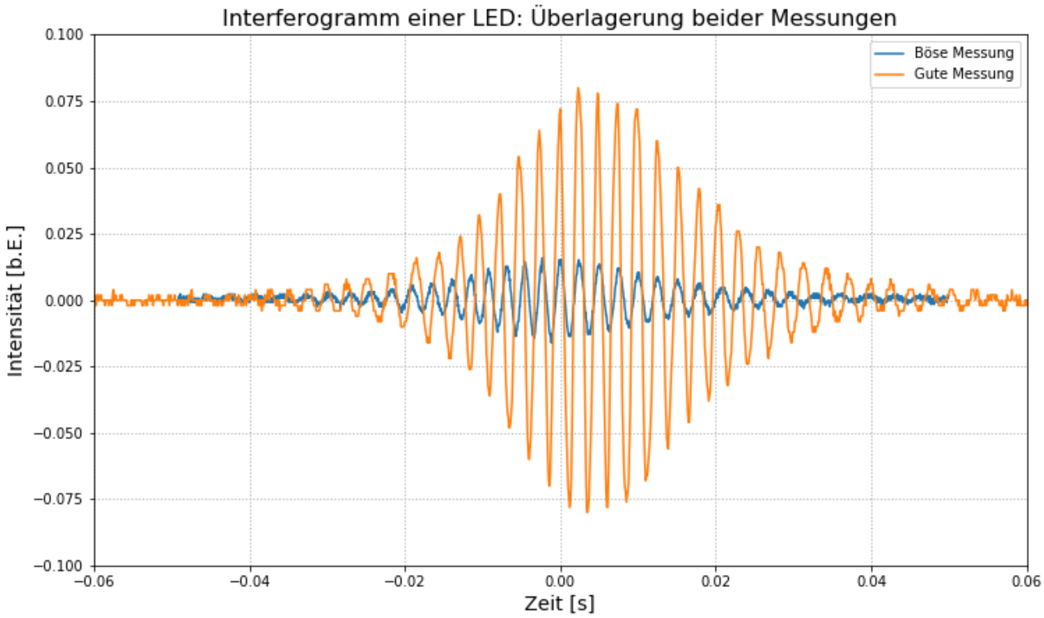


Diagramm 5: Überlagerung beider Messungen

Man erkennt aufgrund der deutlich kleineren Amplitude, dass wir vermutlich bessere Ergebnisse mit der fremden Messung erhalten könnten. Wir erhalten hier für die erste Methode tatsächlich die gleiche Kohärenzlänge

$$L_{b_1} = 2,94(12) \cdot 10^{-6}\,\mathrm{m} \tag{25}$$

und für die zweite Mehode weicht das Ergebnis schon signifikant ab:

$$L_{b_2} = 3,30(14) \cdot 10^{-6}\,\mathrm{m} \tag{26}$$

Ausführlicher ist dies dem Python-Skript zu sehen.

5 Zusammenfassung und Diskussion

5.1 Zusammenfassung

In diesem Versuch haben wir drei wichtige Aufgaben erledigt. Als erstes haben wir durch eine Messung der vorbeilaufenden Maxima durch einen automatischen Zähler und eine automatisch eingestellte Messuhr die Wellenlänge eines Lasers bestimmt. Dafür haben wir fünf verschiedene Messungen gemacht und einen Mittelwert darüber gebildet.

Als nächstes haben wir den Brechungsindex der Luft bestimmt, dies erfolgt analog durch eine Aufzählung vorbeilaufender Interferenzstreifen, jedoch dieses Mal manuell gezählt und in Abhängigkeit des Drucks in einer Glasküvette, wodurch der Laser gestrahlt hat. Hier haben wir dieses Verhältnis auf einem Graph geplottet und darüber eine lineare Anpassung durchgeführt. Aus der Steigung der Gerade und anderen Faktoren, unter anderem die bereits ermittelte Wellenlänge ließ sich der Brechungsindex der Luft bestimmen.

Schließlich haben wir den Laser für eine Leuchtdiode getauscht und über die zeitliche Intensitätsverteilung eine Gausskurve angepasst, mittels zweier Methoden.

5.2 Diskussion

Als erstes wollen wir kurz unsere Ergebnisse zusammenfassen und vergleichen. Dies machen wir am besten tabellarisch, indem wir die σ-Abweichung der berechneten Messwerten und den Literaturwerten gemäß

$$\frac{|G - G_{\mathrm{Lit}}|}{\sqrt{(\Delta G)^2 + (G_{\mathrm{Lit}})^2}} \tag{27}$$

berechnen.

Tabelle 3: Vergleich Wellenlänge

$\lambda\ [10^{-9}\,\mathrm{m}]$	$\frac{\Delta\lambda}{\lambda}$ [%]	$\lambda_{\mathrm{Lit}}\ [10^{-9}\,\mathrm{m}]$	$\frac{\Delta\lambda_{\mathrm{Lit}}}{\lambda_{\mathrm{Lit}}}$ [%]	σ
524,7(2,6)	0,5	532(1)	0,19	2,6

In unserer Rechnung erhalten wir eine σ-Abweichung was innerhalb des akzeptablen Bereiches liegt, aber an der Grenze dieses liegt. Außerdem ist zu bemerken, dass der tatsächliche Wert sich außerhalb des Fehlerbereiches unserer gemessenen Wellenlänge befindet. Ein möglicher Grund für die relativ große σ-Abweichung ist der ziemlich klein abgeschätzter relativer Fehler, der unter 1% liegt. Es lässt sich vermuten, dass vielleicht dieser zu klein abgeschätzt wurde, inesbesondere da wir anscheinend den Abstands-Unterschied Δs bis auf 5 Stellen ziemlich genau gemessen haben. Eine weitere Fehlerquelle ist hier die Sensibilität des Detektors und des Lasers. Beispielsweise verursachen Stimmen und allgemein Geräusche, wie wir experimentell bei der Durchführung beobachtet haben, kurze Schläge in der Zählung der Intereferenzstreifen, da wahrscheinlich die kleinsten Vibrationen schon signifikante Unterschiede bei der Aufzählung der Interferenzstreifen verursachen. Wir haben versucht diese Messungen in möglichster Ruhe durchzuführen, aber da wir mit einer anderen Gruppe gearbeitet haben, die gleichzeitig ihre eigenen Messungen und Besprechungen durchführen muss, gibt es immernoch Raum für kleine Störungen. Diese Störungen hätten die Anzahl der gezählten Maxima erhöht und dementsprechend die Wellenlänge geringer gemacht, was bei uns offensichtlich der Fall ist. Außerdem, da ganz viele Interferenzstreifen pro Sekunde gemessen werden, spielt unsere Reaktionszeit eine große Rolle. Wenn wir nicht sofort die angezeigte Anzahl ablesen, dann sind vielleicht mit den Störungen weitere 50 bis 100 dazugekommen, was die gemessene Anzahl natürlich erhöht und die Wellenlänge verkleinert. Hier hätte ein Ablesefehler zusätzlich beigetragen, einen größeren Spielraum zu haben. Diesen haben wir allerdings nicht berücksichtigt.

Nun kommen wir zur nächsten Messung, und zwar die Berechnung des Brechungsindex der Luft. Hier erhalten wir, angepasst auf die signifikanten Nachkommastellen des Fehlers:

Tabelle 4: Vergleich Brechungsindex

n_0	$\frac{\Delta n_0}{n_0}$ [%]	$n_{0\mathrm{Lit}}$	σ
1,000(8)	0,8	1,00028	$0,035$

Schließlich haben wir die Kohärenzlänge der Leuchtdiode berechnet. Hierzu gibt es keinen Literaturwert zum Vergleich vor, aber wir können die uns 4 zu Verfügung stehenden Werten miteinander vergleichen. Da beide manuelle Anpassungen exakt miteinander übereinstimmen (die Breite) benutzen wir nun die anderen 3.

Tabelle 5: Vergleich Kohärenzlänge

Größe	Wert	Rel. Fehler [%]	σ	
L_1 [10^{-6} m]	2,94(12)	4	σ_{L_1,L_2}	0,1
L_2 [10^{-6} m]	2,93(7)	2,3	$\sigma_{L_1,L_{2_b}}$	2
L_{2_b} [10^{-6} m]	3,30(14)	4,2	$\sigma_{L_2,L_{2_b}}$	2,4

Man erkennt hier zwischen den Messungen mit dem selben Datenpaket erhalten wir sehr gute Ergebnisse. Beide Werte liegen innerhalb der Fehlerebereiche der anderen und die Abweichung beträgt nur $0,1$. Zusätzlich stimmen die Ergebnisse auch mit dem zweiten Datenpaket zum größten Teil überein, mit immernoch akzeptable Abweichungen. Allerdings ist der Vergleich hier schwieriger, da viele Faktoren mehr eine Rolle spielen. Eine ähnliche Kohärenzlänge ist natürlich bei LEDs gleicher Manufaktur zu erwarten, da wir aber hier mit zwei gleichen aber verschiedenen LED's rechnen ist nicht auszuschließen, dass es einen kleinen Unterschied gibt, woraus eine Abweichung nicht auf die Messmethode, sondern auf die Herstellung zurückzuführen ist. Außerdem hätte es Unterschiede in der Justierung geben können, die das Ergebnis verfälschen. Aber mit diesen Faktoren in Betracht, ist es vielleicht desto beeindruckender, dass die gemessenen Kohärenzlängen so (relativ) gut mit einander übereinstimmen. Ein Faktor was wir nicht berücksichtigt haben ist die Verfahrensgeschwindigkeit v, welche wir ohne Fehler aus dem Praktikumsskript entnommen haben. Man könnte diese möglicherweise bestimmen, dafür wäre aber die Zeit im Versuch nicht ausreichend gewesen.

6 Quellen

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

Anhang

Versuch Michelson-Interferometer

20. März 2022

7 Anhang

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

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


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

VII.2 Messung der Wellenlänge

[2]:
```
## Messdaten ##
s_a = np.array([1.088, 4.059, 1.099, 4.059, 1.098]) * 1e-3 # m
s_e = np.array([4.059, 1.099, 4.059, 1.098, 4.059]) * 1e-3
sig_sys_s = 9 * 1e-9 # m
Delta_s = np.abs(s_a - s_e)
m = np.array([11180, 11204, 11192, 11430, 11460])

# Fehler
Delta_s_mean = np.mean(Delta_s)
sig_std_Delta_s = 1 / sqrt(5) * np.std(Delta_s)
sig_sys_Delta_s = sqrt(2) * sig_sys_s
sig_Delta_s = sqrt(sig_std_Delta_s ** 2 + sig_sys_Delta_s ** 2)

m_mean = np.mean(m)
sig_std_m = 1 / sqrt(5) * np.std(m)

# Ausgabe
print('Delta_s =', np.round(1e3 * Delta_s_mean, 4), '+/-',
      np.round(1e3 * sig_std_Delta_s, 4), '[std] +/-',
      np.round(sig_sys_Delta_s,8),  '[sys] [10^-3 m]',
     '=', np.round(1e3 * Delta_s_mean, 4), '+/-',
      np.round(1e3 * sig_std_Delta_s, 4), '[10^-3 m]')
print('m =', np.round(m_mean, -1), '+/-', np.round(sig_std_m, -1))
```

```
Delta_s = 2.9626 +/- 0.0019 [std] +/- 1e-08 [sys] [10^-3 m] = 2.9626 +/-␣
↪0.0019
[10^-3 m]
m = 11290.0 +/- 60.0
```

[3]:
```
# Berechnung der Wellenlänge
lambda_1 = 2 * Delta_s_mean / m_mean # m
sig_lambda_1 = 2 * sqrt( (sig_Delta_s / m_mean) ** 2
                        + (Delta_s_mean * sig_std_m / m_mean ** 2) ** 2␣
↪)
```

```
print('lambda =', np.round(1e9 * lambda_1, 1), '+/-', np.round(1e9 *␣
↪sig_lambda_1, 1), '[10^-9 m]')
```

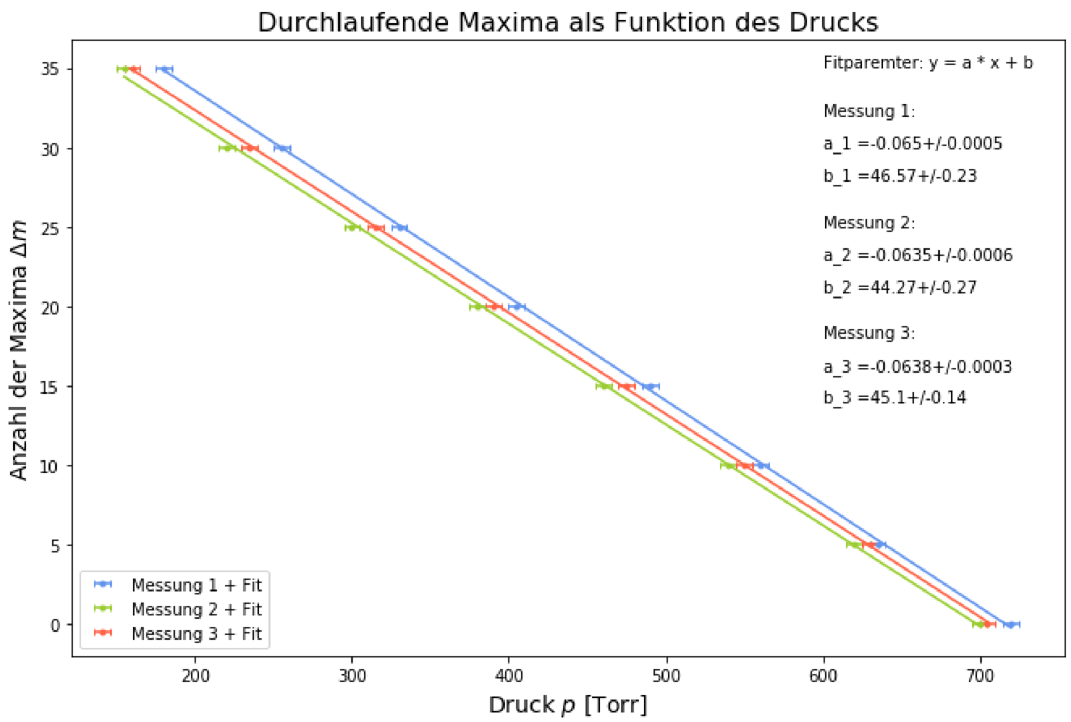
```
lambda = 524.7 +/- 2.6 [10^-9 m]
```

VII.3 Messung des Brechungsindex von Luft

[4]:
```
## Messdaten ##
p1 = np.array([720, 635, 560, 490, 405, 330, 255, 180]) # Torr
p2 = np.array([700, 620, 540, 460, 380, 300, 220, 155])
p3 = np.array([705, 630, 550, 475, 390, 315, 235, 160])
sig_p = 5
Delta_m = np.array([0, 5, 10, 15, 20, 25, 30, 35])

# Fit
def line(x, a, b):
    return a * x + b

popt_1, pcov_1 = curve_fit(line, p1, Delta_m)
popt_2, pcov_2 = curve_fit(line, p2, Delta_m)
popt_3, pcov_3 = curve_fit(line, p3, Delta_m)

a_1 = popt_1[0] # Torr^-1
sig_a_1 = sqrt(pcov_1[0, 0])
a_2 = popt_2[0]
sig_a_2 = sqrt(pcov_2[0, 0])
a_3 = popt_3[0]
sig_a_3 = sqrt(pcov_3[0, 0])

b_1 = popt_1[1]
sig_b_1 = sqrt(pcov_1[1, 1])
b_2 = popt_2[1]
sig_b_2 = sqrt(pcov_2[1, 1])
b_3 = popt_3[1]
sig_b_3 = sqrt(pcov_3[1, 1])

# Plot
```

```
plt.figure(figsize = (11, 7))
plt.errorbar(p1, Delta_m, xerr = sig_p, linestyle = 'None', fmt = '.',
             capsize = 2, label = 'Messung 1 + Fit', color =␣
 ↪'cornflowerblue')
plt.errorbar(p2, Delta_m, xerr = sig_p, linestyle = 'None', fmt = '.',
             capsize = 2, label = 'Messung 2 + Fit', color =␣
 ↪'yellowgreen')
plt.errorbar(p3, Delta_m, xerr = sig_p, linestyle = 'None', fmt = '.',
             capsize = 2, label = 'Messung 3 + Fit', color = 'tomato')
plt.ylabel('Anzahl der Maxima $\Delta m$', size = 14)
plt.xlabel('Druck $p$ [Torr]', size = 14)
plt.title('Durchlaufende Maxima als Funktion des Drucks', size = 16)

plt.plot(p1, line(p1, *popt_1), color = 'cornflowerblue')
plt.plot(p2, line(p2, *popt_2), color = 'yellowgreen')
plt.plot(p3, line(p3, *popt_3), color = 'tomato')

plt.text(600, 35, 'Fitparemter: y = a * x + b')
plt.text(600, 32, 'Messung 1:')
plt.text(600, 30, 'a_1 =' + str(np.round(a_1, 4)) + '+/-'
         + str(np.round(sig_a_1, 4)))
plt.text(600, 28, 'b_1 =' + str(np.round(b_1, 2)) + '+/-'
         + str(np.round(sig_b_1, 2)))

plt.text(600, 25, 'Messung 2:')
plt.text(600, 23, 'a_2 =' + str(np.round(a_2, 4)) + '+/-'
         + str(np.round(sig_a_2, 4)))
plt.text(600, 21, 'b_2 =' + str(np.round(b_2, 2)) + '+/-'
         + str(np.round(sig_b_2, 2)))

plt.text(600, 18, 'Messung 3:')
plt.text(600, 16, 'a_3 =' + str(np.round(a_3, 4)) + '+/-'
         + str(np.round(sig_a_3, 4)))
plt.text(600, 14, 'b_3 =' + str(np.round(b_3, 2)) + '+/-'
         + str(np.round(sig_b_3, 2)))
```

```
plt.legend(loc = 'lower left')
plt.savefig('images/232/V232Diagramm1.png')
```



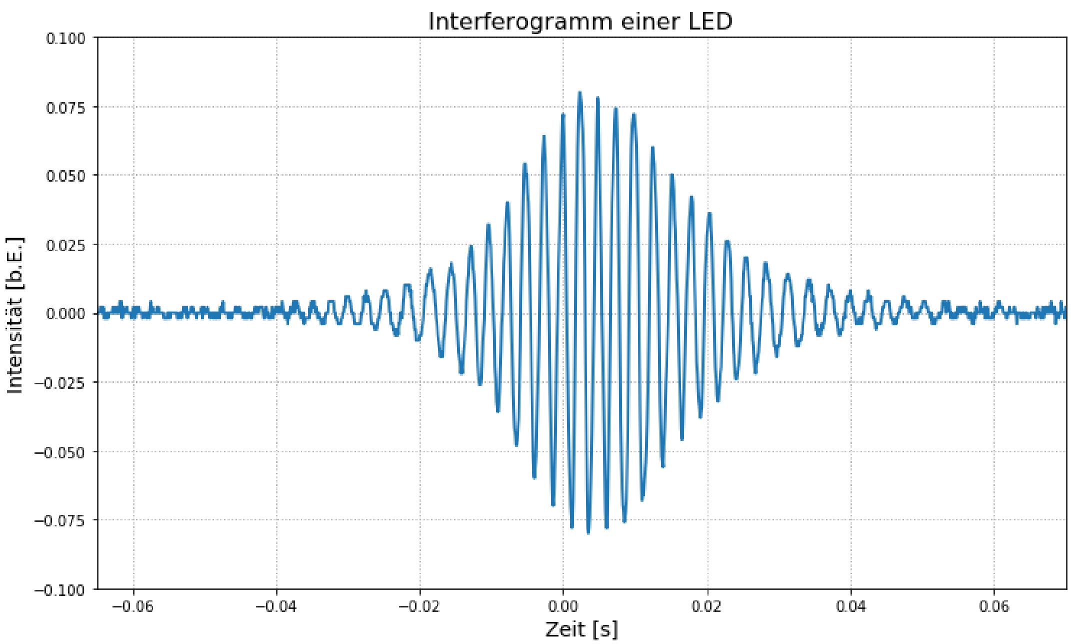
[5]:
```
# Brechungsindex
a = np.abs(np.mean([a_1, a_2, a_3]))
sig_a = 1 / sqrt(3) * np.std([a_1, a_2, a_3])
T0 = 273.15 # K
T = 24.1 + 273.15
sig_T = 0.1
p0 = 760 # Torr
d = 0.05 # m
sig_d = 0.00005

n_0 = a * lambda_1 * p0 * T / (2 * d * T0) + 1
sig_n_0 = n_0 * sqrt( (sig_a / a) ** 2 + (sig_T / T) ** 2 + (sig_d / d)
↪** 2
               + (sig_lambda_1 / lambda_1) ** 2)

print('n_0 =', np.round(n_0, 3), '+/-', np.round(sig_n_0, 3))
```

```
n_0 = 1.0 +/- 0.008
```

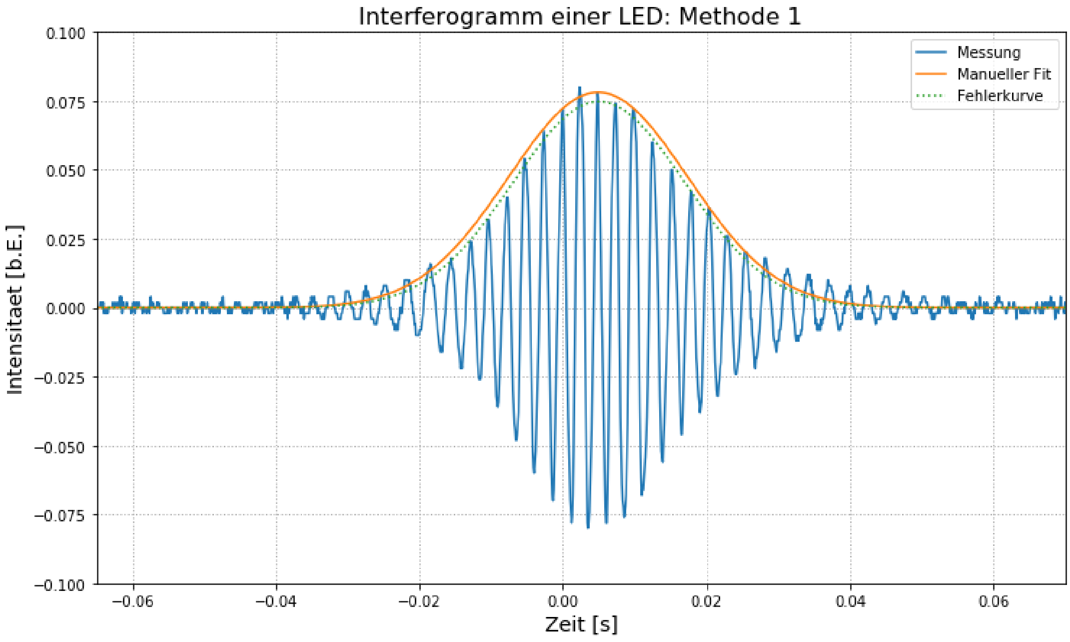
VII.4 Messung der Kohärenzlänge der Leuchtdiode

[6]:
```
## Messdaten ##
data=np.genfromtxt('data/232/F0001CH1.CSV',␣
↪delimiter=",",skip_header=18)

t = data[:,3:4]
t = t[:, 0]

U = data[:,4:5]
U = U[:, 0]
```

[7]:
```
# Plot
plt.figure(figsize = (12, 7))
plt.plot(t, U,  linewidth = 2)
plt.axis([-0.065, 0.07, -0.1, 0.1])
plt.grid(linestyle = "dotted", linewidth = 1)
plt.title('Interferogramm einer LED', size = 16)
plt.ylabel('Intensität [b.E.]', size = 14)
plt.xlabel('Zeit [s]', size = 14)

plt.savefig('images/232/V232Diagramm2.png')
```

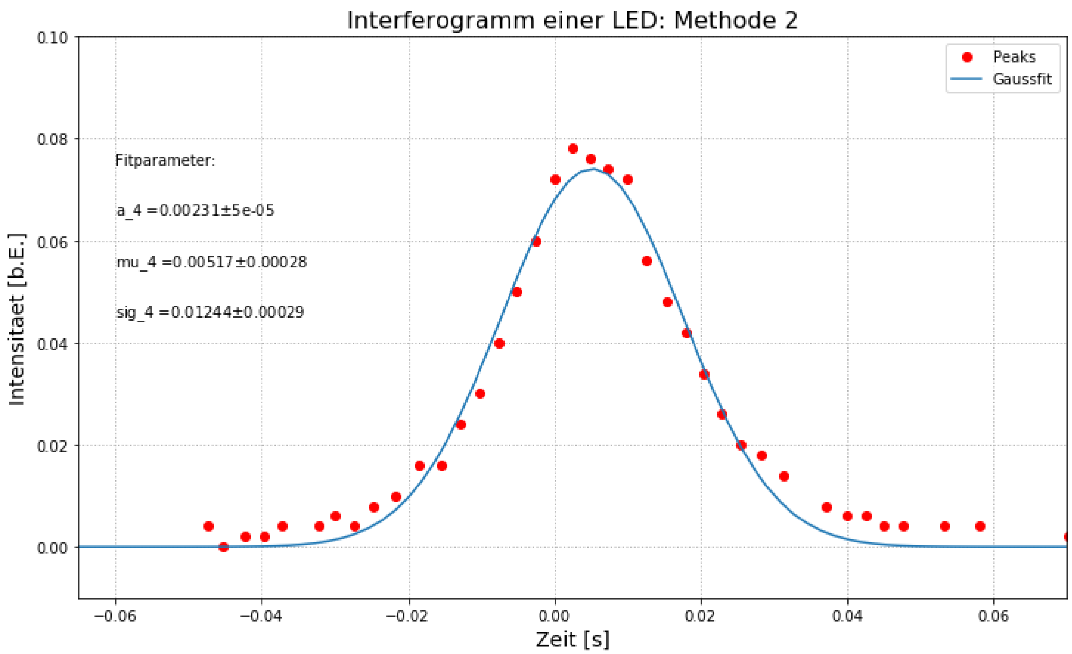


[8]:
```
# Gauss Fit: Methode 1
def gauss(t, a, mu, sig):
    return a / (sqrt(2 * pi) * sig) * exp(-(t - mu) ** 2 / (2 * sig **␣
 ↪2))

a_4_a = 0.00245
mu_4_a = 0.0049
sig_4_a = 0.0125
sig_sig_4_a = 0.0005

a_4_b = 0.00225
mu_4_b = 0.005
sig_4_b = 0.012

plt.figure(figsize = (12, 7))
plt.plot(t, U, label = 'Messung')
plt.xlabel('Zeit [s]', size = 14)
plt.ylabel('Intensitaet [b.E.]', size = 14)
plt.title('Interferogramm einer LED: Methode 1', size = 16)
plt.axis([-0.065, 0.07, -0.1, 0.1])
plt.plot(t, gauss(t, a_4_a, mu_4_a, sig_4_a), label = 'Manueller Fit')
plt.plot(t, gauss(t, a_4_b, mu_4_b, sig_4_b), label = 'Fehlerkurve',
         linestyle = 'dotted')
plt.grid(linestyle = "dotted", linewidth = 1)
plt.legend(loc = 'best')

plt.savefig('images/232/V232Diagramm3.png')
```



[9]:
```python
# Gauss Fit: Methode 2
peakind = signal.find_peaks_cwt(U, np.arange(1, 30), noise_perc = 100)

# Fit
def gaussian(t, a, mu, sig):
    return a / sqrt(2 * pi) / sig * exp(-(t - mu) ** 2 / (2 * sig **
 ↪2))

p0 = [0.0025, 0.005, 0.0125]
popt_4, pcov_4 = curve_fit(gaussian, t[peakind], U[peakind], p0)

a_4_2 = popt_4[0]
sig_a_4_2 = sqrt(pcov_4[0, 0])
mu_4_2 = popt_4[1]
sig_mu_4_2 = sqrt(pcov_4[1, 1])
sig_4_2 = popt_4[2]
sig_sig_4_2 = sqrt(pcov_4[2, 2])

x=np.linspace(-0.08,0.1,100)
```

```
plt.figure(figsize = (12, 7))
plt.plot(t[peakind], U[peakind], marker = 'o',linewidth=0, color =␣
↪'red',
        label = 'Peaks')
plt.plot(x, gaussian(x, *popt_4), label = "Gaussfit")
plt.text(-0.06, 0.075, 'Fitparameter:')
plt.text(-0.06, 0.065, 'a_4 =' + str(np.round(a_4_2, 5)) + '$\pm$'
         + str(np.round(sig_a_4_2, 5)))
plt.text(-0.06, 0.055, 'mu_4 =' + str(np.round(mu_4_2, 5)) + '$\pm$'
         + str(np.round(sig_mu_4_2, 5)))
plt.text(-0.06, 0.045, 'sig_4 =' + str(np.round(sig_4_2, 5)) + '$\pm$'
         + str(np.round(sig_sig_4_2, 5)))
plt.xlabel('Zeit [s]', size = 14)
plt.ylabel('Intensitaet [b.E.]', size = 14)
plt.title('Interferogramm einer LED: Methode 2', size = 16)
plt.axis([-0.065, 0.07, -0.01, 0.1])
plt.grid(linestyle = "dotted", linewidth = 1)
plt.legend(loc = 'best')

plt.savefig('images/232/V232Diagramm4.png')
```



[10]:
```
# Bestimmung der Kohärenzlänge: Methode 1
v = 1e-4 # m s^-1 Verfahrengeschwindigkeit

fwhm_1 = 2 * sqrt(2 * log(2)) * sig_4_a
sig_fwhm_1 = 2 * sqrt(2 * log(2)) * sig_sig_4_a

l_k_1 = fwhm_1 * v
sig_l_k_1 = sig_fwhm_1 * v

print('FWHM_1 =', np.round(fwhm_1, 4), '+/-', np.round(sig_fwhm_1, 4),␣
 ↪'[s]')
print('l_k_1 =', np.round(1e6 * l_k_1, 2), '+/-', np.round(1e6 *␣
 ↪sig_l_k_1, 2), '[10^-6 m]')
```

```
FWHM_1 = 0.0294 +/- 0.0012 [s]
l_k_1 = 2.94 +/- 0.12 [10^-6 m]
```

[11]:
```
# Bestimmung der Kohärenzlänge: Methode 2

fwhm_2 = 2 * sqrt(2 * log(2)) * sig_4_2
sig_fwhm_2 = 2 * sqrt(2 * log(2)) * sig_sig_4_2

l_k_2 = fwhm_2 * v
sig_l_k_2 = sig_fwhm_2 * v

print('FWHM_2 =', np.round(fwhm_2, 4), '+/-', np.round(sig_fwhm_2, 4),␣
 ↪'[s]')
print('l_k_2 =', np.round(1e6 * l_k_2, 2), '+/-', np.round(1e6 *␣
 ↪sig_l_k_2, 2), '[10^-6 m]')
```

```
FWHM_2 = 0.0293 +/- 0.0007 [s]
l_k_2 = 2.93 +/- 0.07 [10^-6 m]
```

VII.4.2 Messung der Kohärenzlänge der Leuchtdiode: Alternative Messdaten

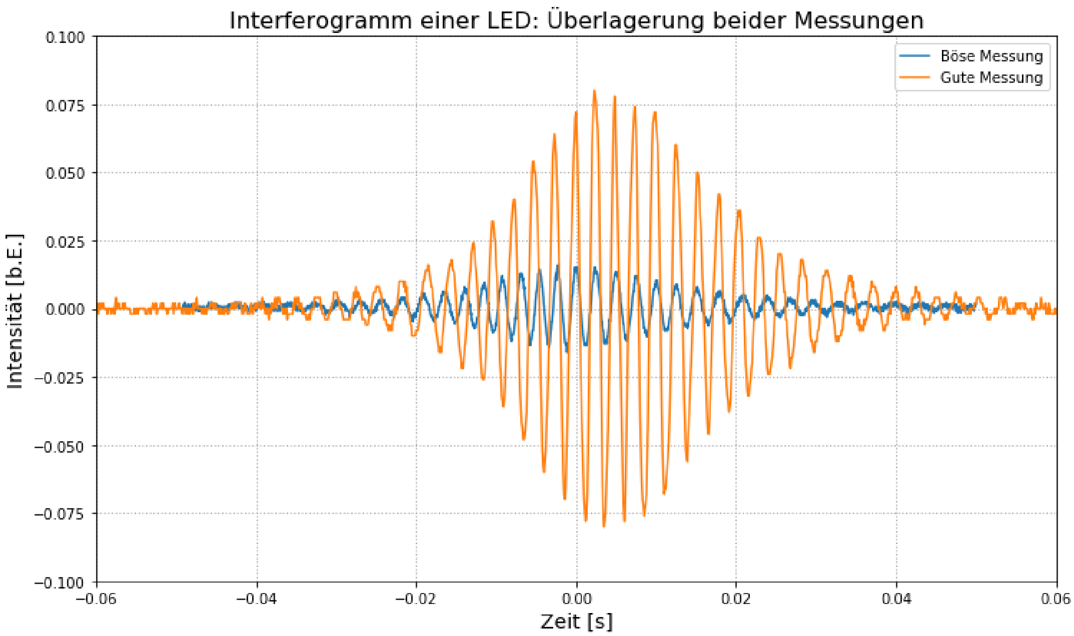
[12]:
```
## Messdaten ##
# Neue Messung: In einer Zelle um die Variablen im Vakuum zu halten
data = np.genfromtxt('data/232/F0000CH1.CSV',␣
 ↪delimiter=",",skip_header=18)

t = data[:,3:4]
t = t[:, 0]

U = data[:,4:5]
U = U[:, 0] - 0.212 # Anpassung weil Mittelwert um 0.2

# Alte
data_og = np.genfromtxt('data/232/F0001CH1.CSV',␣
 ↪delimiter=",",skip_header=18)

t_og = data_og[:,3:4]
t_og = t_og[:, 0]

U_og = data_og[:,4:5]
U_og = U_og[:, 0]

# Gegeneinander
plt.figure(figsize = (12, 7))
plt.plot(t, U, label = 'Böse Messung')
plt.plot(t_og, U_og, label = 'Gute Messung')
plt.axis([-0.06, 0.06, -0.1, 0.1])
plt.grid(linestyle = "dotted", linewidth = 1)
plt.title('Interferogramm einer LED: Überlagerung beider Messungen',␣
 ↪size = 16)
plt.ylabel('Intensität [b.E.]', size = 14)
plt.xlabel('Zeit [s]', size = 14)
plt.legend(loc = 'best')
```

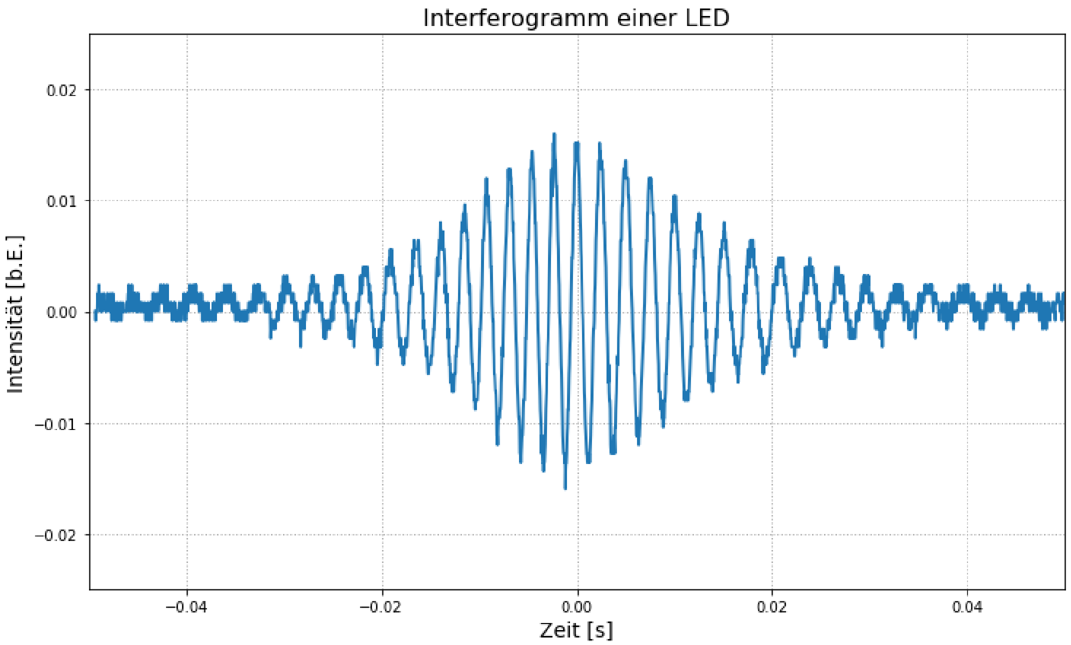
```
plt.savefig('images/232/V232Diagramm5.png')

# Plot
plt.figure(figsize = (12, 7))
plt.plot(t, U,  linewidth = 2)
plt.axis([-0.05, 0.05, -0.025, 0.025])
plt.grid(linestyle = "dotted", linewidth = 1)
plt.title('Interferogramm einer LED', size = 16)
plt.ylabel('Intensität [b.E.]', size = 14)
plt.xlabel('Zeit [s]', size = 14)

plt.savefig('images/232/V232Diagramm2.2.png')



# Gauss Fit: Methode 1

a_b = 0.00048
mu_b = 0.0001
sig_b = 0.0125
sig_sig_b = 0.0005

plt.figure(figsize = (12, 7))
plt.plot(t, U, label = 'Messung')
plt.xlabel('Zeit [s]', size = 14)
plt.ylabel('Intensitaet [b.E.]', size = 14)
plt.title('Interferogramm einer LED: Methode 1', size = 16)
plt.axis([-0.05, 0.05, -0.025, 0.025])
plt.plot(t, gauss(t, a_b, mu_b, sig_b), color = 'orange', label =␣
 ↪'Manueller Fit')
plt.grid(linestyle = "dotted", linewidth = 1)
plt.legend(loc = 'best')

plt.savefig('images/232/V232Diagramm3.2.png')



# Gauss Fit: Methode 2
```

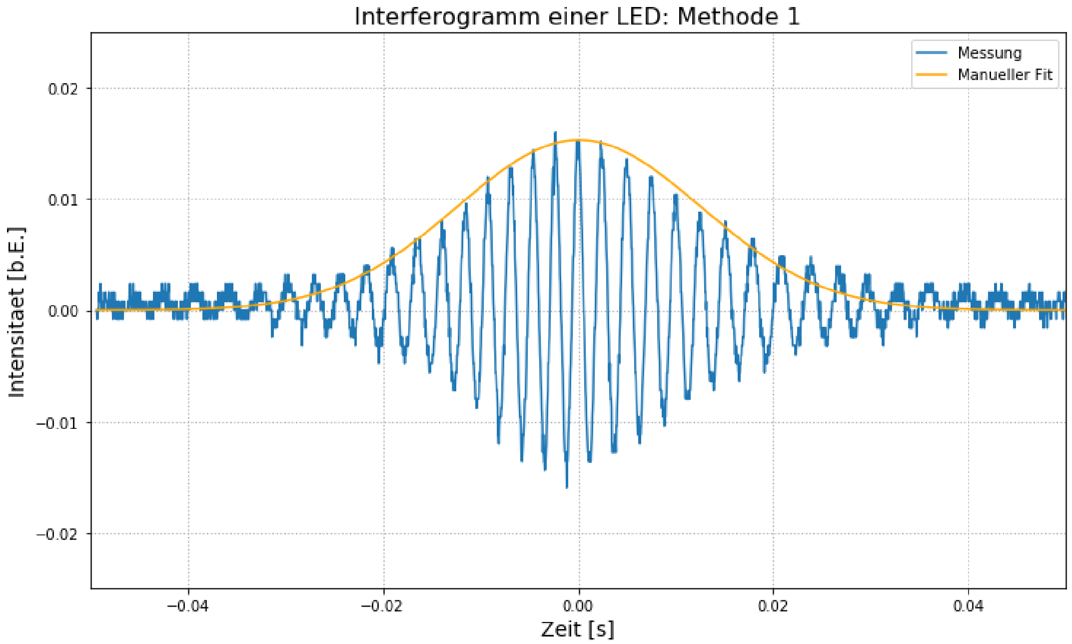
```
peakind = signal.find_peaks_cwt(U, np.arange(1, 30), noise_perc = 100)

# Fit
def gaussian(t, a, mu, sig):
    return a / sqrt(2 * pi) / sig * exp(-(t - mu) ** 2 / (2 * sig **␣
↪2))

p0 = [0.0005, 0.005, 0.0125]
popt_4, pcov_4 = curve_fit(gaussian, t[peakind], U[peakind], p0)

a_b_2 = popt_4[0]
sig_a_b_2 = sqrt(pcov_4[0, 0])
mu_b_2 = popt_4[1]
sig_mu_b_2 = sqrt(pcov_4[1, 1])
sig_b_2 = popt_4[2]
sig_sig_b_2 = sqrt(pcov_4[2, 2])

x=np.linspace(-0.08,0.1,100)

plt.figure(figsize = (12, 7))
plt.plot(t[peakind], U[peakind], marker = 'o',linewidth=0, color =␣
↪'red',
        label = 'Peaks')
plt.plot(x, gaussian(x, *popt_4), label = "Gaussfit")
plt.text(-0.04, 0.020, 'Fitparameter:')
plt.text(-0.04, 0.018, 'a_4 =' + str(np.round(a_b_2, 5)) + '$\pm$' +␣
↪str(np.round(sig_a_b_2, 5)))
plt.text(-0.04, 0.016, 'mu_4 =' + str(np.round(mu_b_2, 4)) + '$\pm$' +␣
↪str(np.round(sig_mu_b_2, 4)))
plt.text(-0.04, 0.014, 'sig_4 =' + str(np.round(sig_b_2, 4)) + '$\pm$'␣
↪+ str(np.round(sig_sig_b_2, 4)))
plt.xlabel('Zeit [s]', size = 14)
plt.ylabel('Intensitaet [b.E.]', size = 14)
plt.title('Interferogramm einer LED: Methode 2', size = 16)
plt.axis([-0.06, 0.06, -0.001, 0.025])
plt.grid(linestyle = "dotted", linewidth = 1)
```

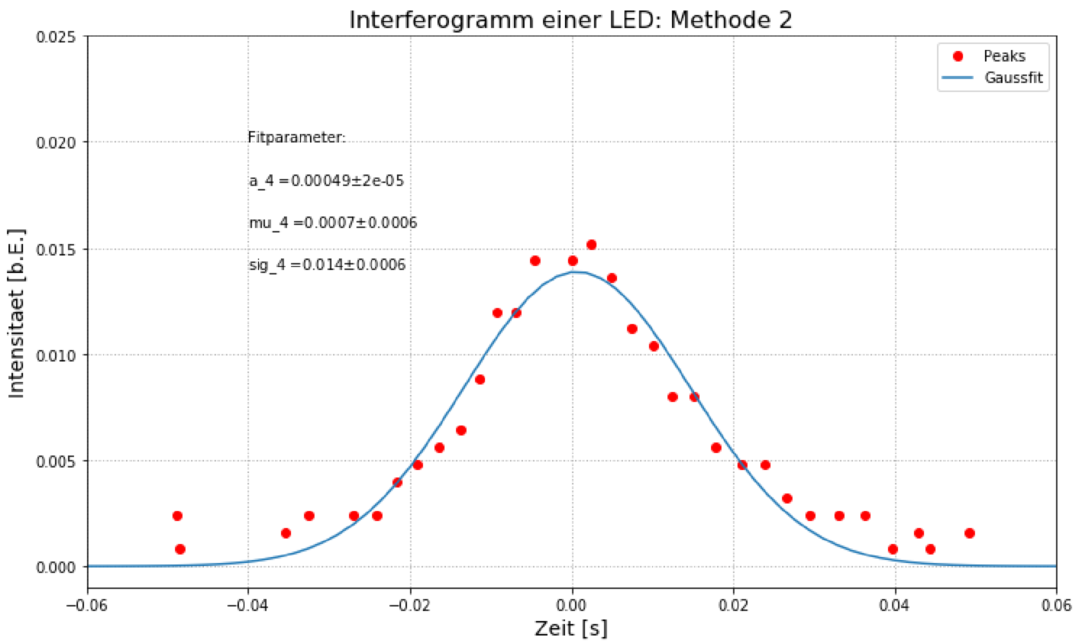
```
plt.legend(loc = 'best')

plt.savefig('images/232/V232Diagramm4.2.png')

# Bestimmung der Kohärenzlänge: Methode 1
print('Methode 1:')
fwhm_b_1 = 2 * sqrt(2 * log(2)) * sig_b
sig_fwhm_b_1 = 2 * sqrt(2 * log(2)) * sig_sig_b

l_k_b_1 = fwhm_b_1 * v
sig_l_k_b_1 = sig_fwhm_b_1 * v

print('FWHM_b_1 =', np.round(fwhm_b_1, 4), '+/-', np.
 ↪round(sig_fwhm_b_1, 4), '[s]')
print('l_k_b_1 =', np.round(1e6 * l_k_b_1, 2), '+/-', np.round(1e6 *␣
 ↪sig_l_k_b_1, 2), '[10^-6 m]')

# Methode 2
print('\nMethode 2:')
fwhm_b_2 = 2 * sqrt(2 * log(2)) * sig_b_2
sig_fwhm_b_2 = 2 * sqrt(2 * log(2)) * sig_sig_b_2

l_k_b_2 = fwhm_b_2 * v
sig_l_k_b_2 = sig_fwhm_b_2 * v

print('FWHM_b_2 =', np.round(fwhm_b_2, 4), '+/-', np.
 ↪round(sig_fwhm_b_2, 4), '[s]')
print('l_k_b_2 =', np.round(1e6 * l_k_b_2, 2), '+/-', np.round(1e6 *␣
 ↪sig_l_k_b_2, 2), '[10^-6 m]')
```

```
Methode 1:
FWHM_b_1 = 0.0294 +/- 0.0012 [s]
l_k_b_1 = 2.94 +/- 0.12 [10^-6 m]

Methode 2:
FWHM_b_2 = 0.033 +/- 0.0014 [s]
l_k_b_2 = 3.3 +/- 0.14 [10^-6 m]
```


Interferogramm einer LED: Überlagerung beider Messungen
Böse Messung
Gute Messung
Intensität [b.E.]
Zeit [s]
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.06
−0.04
−0.02
0.00
0.02
0.04
0.06


Interferogramm einer LED
Intensität [b.E.]
Zeit [s]
0.02
0.01
0.00
−0.01
−0.02
−0.04
−0.02
0.00
0.02
0.04


Interferogramm einer LED: Methode 1
Messung
Manueller Fit
Intensitaet [b.E.]
Zeit [s]
0.02
0.01
0.00
−0.01
−0.02
−0.04
−0.02
0.00
0.02
0.04


Interferogramm einer LED: Methode 2
Peaks
Gaussfit
Fitparameter:
a_4 =0.00049±2e-05
mu_4 =0.0007±0.0006
sig_4 =0.014±0.0006
Intensitaet [b.E.]
Zeit [s]
0.025
0.020
0.015
0.010
0.005
0.000
−0.06
−0.04
−0.02
0.00
0.02
0.04
0.06

Sigmas

[13]:
```
# Lambda Lit
lambda_lit = 532 * 1e-9 # m
sig_lambda_lit = 1e-9
fehler('lambda',np.round(lambda_1, 10), np.round(sig_lambda_1, 10),␣
↪lambda_lit, sig_lambda_lit)
```

```
lambda
Relativer Fehler:  0.49552125023823135
Rel. Fehler (Vergleich): 0.18796992481203006
Absoluter Fehler:  7.300000000000019e-09
Verhältnis: 0.9862781954887218
Sigma-Abweichung:  2.6205472789547506
```

[14]:
```
# n Lit
n_lit = 1.00028
fehler('n_0', np.round(n_0, 3), np.round(sig_n_0, 3), n_lit, 0)
```

```
n_0
Relativer Fehler:  0.8
Rel. Fehler (Vergleich): 0.0
Absoluter Fehler:  0.000280000000000058
Verhältnis: 0.9997200783780541
Sigma-Abweichung:  0.03500000000000725
```

[15]:
```
# l_k_1 - l_k_2
fehler('l_k_1 - l_k_2', l_k_1, sig_l_k_1, l_k_2, sig_l_k_2)
```

```
l_k_1 - l_k_2
Relativer Fehler:  4.0
Rel. Fehler (Vergleich): 2.3244312585105344
Absoluter Fehler:  1.300488462450178e-08
Verhältnis: 1.0044377393304604
Sigma-Abweichung:  0.09560607324327101
```

[16]:
```
# l_k_1 - l_k_b_2
fehler('l_k_1 - l_k_b_2', l_k_1, sig_l_k_1, l_k_b_2, sig_l_k_b_2)
```

```
l_k_1 - l_k_b_2
Relativer Fehler:  4.0
Rel. Fehler (Vergleich): 4.163149950709182
Absoluter Fehler:  3.5579988885165485e-07
Verhältnis: 0.8921597918459891
Sigma-Abweichung:  1.966688810520832
```

[17]:
```
# l_k_2 - l_k_b_2
fehler('l_k_2 - l_k_b_2', l_k_2, sig_l_k_2, l_k_b_2, sig_l_k_b_2)
```

```
l_k_2 - l_k_b_2
Relativer Fehler:  2.3244312585105344
Rel. Fehler (Vergleich): 4.163149950709182
Absoluter Fehler:  3.6880477347615663e-07
Verhältnis: 0.8882181113990065
Sigma-Abweichung:  2.4054752080717097
```

[ ]:
```

```