

Lichtgeschwindigkeit.nbconvert

December 17, 2023

1 Fakultät für Physik

1.1 Physikalisches Praktikum P1 für Studierende der Physik

Versuch P1-43, 44, 45 (Stand: August 2023)

[Raum F1-12](#)

2 Lichtgeschwindigkeit

Name: *surname1* Vorname: *name1* E-Mail: *mail1*

Name: *surname2* Vorname: *name2* E-Mail: *mail2*

Gruppennummer: *groupnumber*

Betreuer:

Versuch durchgeführt am: *date*

Beanstandungen:

Testiert am: _____ Testat: _____

```
[1]: # Hilfsfunktionen. Nicht weiter beachten
from IPython.core.display import Markdown
from IPython.display import display
def M_helper_func(txt):
    #print(txt)
    #print('fr'+txt.replace("$", "$").replace("{", "{{").replace("}", "}}").
    ↪replace("[", "{").replace("!", "!")+''')
    excreturnvalue = ""
    loc = {}
    exec('excreturnvalue = fr"<div style=\'margin-left: 8px;\'>'+txt.
    ↪replace("{", "{{").replace("}", "}}").replace("[", "{").replace("!", "!")+</
    ↪div>'", globals(), loc)
    display(Markdown(loc["excreturnvalue"]))
    return loc["excreturnvalue"]

from IPython.core.magic import (register_line_magic,
                                register_cell_magic)

from IPython.core import magic

def register_line_magic(f):
    setattr(f, magic.MAGIC_NO_VAR_EXPAND_ATTR, True)
    return magic.register_line_magic(f)

@register_line_magic
def m(line):
    M_helper_func(line)
```

```

import os
import numpy as np
from numpy import pi
import matplotlib.pyplot as plt
import matplotlib as mpl
import scipy as sci
import kafe2
import pandas as pd
import uncertainties
from uncertainties import ufloat, unumpy
from uncertainties.unumpy import uarray as uarr
from uncertainties.unumpy import nominal_values, nominal_values as nomv, u
↳std_devs, std_devs as stdv
from uncertainties.umath import sqrt as usqrt, sin as usin, exp as uexp
import scipy
import matplotlib.image as mpimg
from io import StringIO

mpl.rcParams['figure.dpi'] = 100
class ng:
    def __init__(self, x=None, stat=None, syst=None):
        if x == None:
            self.syst = None
            self.stat = None
        else:
            self.syst = ufloat(x, 0)
            self.stat = ufloat(x, 0)
            if syst != None:
                self.syst = ufloat(x, syst)
            if stat != None:
                self.stat = ufloat(x, stat)
    def SetSyst(self, syst):
        self.syst = ufloat(nomv(self.syst), syst)
    def SetStat(self, stat):
        self.stat = ufloat(nomv(self.stat), stat)
    def __str__(self):
        return f'{{nomv(self.stat):.2}}± {{stdv(self.stat):.2}}± {{stdv(self.syst):
↳.2}}'
    def getTotErr(self):
        return np.sqrt(stdv(self.syst)**2 + stdv(self.stat)**2)
    def nom(self):
        print(type(nomv(self.syst)))
        return float(nomv(self.syst))
    def __add__(self, o):
        res = ng()
        if type(o) == ng:
            res.syst = self.syst + o.syst

```

```

        res.stat = self.stat + o.stat
    else:
        res.syst = self.syst + o
        res.stat = self.stat + o
    return res
def __iadd__(self,o):
    return self.__add__(o)
def __sub__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst - o.syst
        res.stat = self.stat - o.stat
    else:
        res.syst = self.syst - o
        res.stat = self.stat - o
    return res
def __isub__(self,o):
    return self.__sub__(o)
def __mul__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst * o.syst
        res.stat = self.stat * o.stat
    else:
        res.syst = self.syst * o
        res.stat = self.stat * o
    return res
def __imul__(self,o):
    return self.__mul__(o)
def __truediv__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst / o.syst
        res.stat = self.stat / o.stat
    else:
        res.syst = self.syst / o
        res.stat = self.stat / o
    return res
def __itruediv__(self,o):
    return self.__truediv__(o)
def __pow__(self,o):
    res = ng()
    if type(o) == ng:
        res.syst = self.syst**o.syst
        res.stat = self.stat**o.stat
    else:
        res.syst = self.syst**o

```

```

        res.stat = self.stat**o
    return res
def __ipow__(self,o):
    return self.__pow__(o)
def sin(o):
    if type(o) == ng:
        res = ng()
        res.syst = usin(o.syst)
        res.stat = usin(o.stat)
        return res
    else:
        return np.sin(o)
def exp(o):
    if type(o) == ng:
        res = ng()
        res.syst = uexp(o.syst)
        res.stat = uexp(o.stat)
        return res
    else:
        return np.exp(o)
def sqrt(o):
    if type(o) == ng:
        res = ng()
        res.syst = usqrt(o.syst)
        res.stat = usqrt(o.stat)
        return res
    else:
        return np.sqrt(o)

def uplot(x,y,*args, **kwargs):
    utypes = (uncertainties.core.
Variable,uncertainties.core.AffineScalarFunc)
    xplt = x
    yplt = y
    if type(x[0]) in utypes:
        xplt = nomv(x)
        kwargs["xerr"] = stdv(x)
    if type(y[0]) in utypes:
        yplt = nomv(y)
        kwargs["yerr"] = stdv(y)
    plt.errorbar(xplt,yplt,*args,**kwargs)

def plting(path):
    try:
        img = mpimg.imread(path)
        imgplot = plt.imshow(img)
        plt.axis('off')

```

```
plt.show()
except Exception as e:
    print("Image Not Found")
    print(e)
```

3 Durchführung

3.1 Aufgabe 1: Drehspiegelmethode

Hinweise zu allen hier durchzuführenden Messungen finden Sie in der Datei [Hinweise-Aufgabe-1.md](#).

3.1.1 Aufgabe 1.1: Vorbereitung

Diskutieren Sie das Messprinzip, den Aufbau und die Eigenschaften des Strahlengangs, basierend auf den Angaben zum Versuchsaufbau. Beantworten Sie dabei, vor Beginn der Messung, die folgenden Fragen:

- Welche Aufgabe erfüllt die Linse im Strahlengang?
- Wo sollten Sie die Linse im Strahlengang positionieren, damit sie den Ausgang des Lasers auf den Endspiegel abbildet?
- Berechnen Sie, mit Hilfe der [Linsengleichung](#), die Position ℓ , für die der Laser als Gegenstand G in ein Bild B im Endspiegel abgebildet wird (siehe [Skizze](#)).
- Wieso ruht der beobachtete Leuchtfleck während der Messung auf der Skala, obwohl sich der Spiegel doch kontinuierlich dreht?
- Geben Sie die Formel zur Berechnung der Lichtgeschwindigkeit an.
- Geben Sie eine Abschätzung für die Größe des zu erwartenden Versatzes s des Lichtflecks, relativ zum Fall des ruhenden Drehspiegels an.

In dieser Aufgabe sollen die Messprinzipien des Versuchs diskutiert werden.

- Die Linse im Strahlengang hat hauptsächlich zwei Aufgaben. Sie sorgt dafür, dass nur Licht, dessen Ausbreitungsrichtung beim Rücklauf parallel zur optischen Achse liegt wieder auf den Drehspiegel fokussiert wird. Damit wird einmal der Einfluss von nicht parallelen Streustrahlen auf dem Schirm minimiert und andererseits wird dafür gesorgt, dass nur bei einer bestimmten Ausrichtung des Drehspiegels, nämlich dann wenn der Laserstrahl bei der ersten Reflektion perfekt in parallele Richtung reflektiert wird, der Laserstrahl wieder auf den Schirm zurücklaufen kann. Für alle anderen Winkel kommt das Licht nicht wieder auf dem Schirm an. Außerdem sorgt die Linse dafür, dass das Licht auf dem Weg zwischen Linse und Endspiegel parallel läuft, sodass der Strahl nicht weiter divergiert sondern einen sauberen Punkt auf dem Schirm abbildet.
- Die Linse muss zwischen Laser und Endspiegel plziert werden. Da zusätzlich noch der Drehspiegel im Fokus der Linse liegen soll muss die Linse zwischen Drehspiegel und Umlenkspiegel plziert werden. Aus der Linsengleichung folgt, wie in der Dokumentation ausgeführt ist, dass $\frac{1}{f} = \frac{1}{g'} + \frac{1}{b'}$ $= \frac{1}{\ell+f} + \frac{1}{a+b-f}$ gelten muss.
- Da die Brennweite der Linse fest ist muss die Fokussierung des Bildes durch den Abstand des

Lasers von der Linse erreicht werden. Es ergibt sich aus obiger Gleichung dass $\ell = \frac{f^2}{a+b-2f} = 6.57895 \text{ m}$.

- Wie oben erwähnt werden nur diejenigen Strahlen, die vom Fokuspunkt in die Linse einfallen und dann parallel zur optischen Achse verlaufen auch wieder zurück durch die Linse auf den drehspiegel Reflektiert. Strahlen die nicht parallel laufen verlassen zur Seite den Aufbau und nicht wieder fokussiert. Nur dann, wenn der Spiegel in einem genauen Winkel, nämlich orthogonal zur Winkelhalbierenden zwischen der Achse des Laser-geräts und optischen Achse der Linse, wird das Licht in dieser Weise durch die Linse gelenkt. Dementsprechend sieht man nur dann einen Punkt, wenn der Spiegel einen genau definierten Winkel hat aber unser Auge ist nicht schnell genug das Flackern des Punktes wahr zu nehmen.
- $c_{Luft} = 8\pi \frac{\ell \nu(a+b)}{s}$ ist die Formel für die Lichtgeschwindigkeit im Medium, hier Luft. Diese muss dann noch mit n_{Luft} multipliziert werden.
- Der Versatz s ist proportional zur Frequenz des Drehspiegels. Es gilt $s = 8\pi \frac{\ell \nu(a+b)}{c}$. Das ergibt bei einer maximalen Rotationsgeschwindigkeit von 450 Hz einen Versatz von 3.42505mm.

3.1.2 Aufgabe 1.2: Justierung der Apparatur

Überprüfen Sie mit Hilfe Ihres:r Tutor:in die Justierung des Versuchs. Gehen Sie dabei am besten entlang des Strahlengangs, wie folgt, vor:

- **Laser – Drehspiegel:** Der horizontal ausgesandte Laserstrahl sollte den ruhenden Drehspiegel mittig treffen und horizontal wieder verlassen.
- **Drehspiegel – Umlenkspiegel:** Richten Sie den Drehspiegelwinkel so aus, dass der reflektierte Strahl (noch ohne Linse) auf die Mitte des Umlenkspiegels fällt. Decken Sie hierzu den Endspiegel ab.
- **Umlenkspiegel – Endspiegel:** Richten Sie den Umlenkspiegel so aus, dass der Strahl auf die Mitte des Endspiegels fällt.
- **Linse:** Richten Sie die Linse aus.
- **Endspiegel – Umlenkspiegel:** Richten Sie den Endspiegel so aus, dass der Strahl in sich reflektiert wird (beachten Sie die Lichtflecke auf dem Umlenkspiegel).
- **Kontrolle und Feinjustierung:** Stellen Sie durch Feinabstimmung sicher, dass der zurückkehrende Strahl den Drehspiegel und die Laseraustrittsöffnung trifft.
- **Schirm und USB-Mikroskop:** Positionieren Sie das USB-Mikroskop hinter dem Schirm mit der Skala. Laser und Skala sind gleich weit vom Strahlteiler entfernt.
- **Frequenzmessung am Drehspiegel:** Positionieren Sie den Phototransistor für die Messung der Rotationsfrequenz des Drehspiegels.

Notieren Sie sich alle Besonderheiten, die Ihnen dabei auffallen.

Achtung: Optische Apparaturen sind sehr empfindlich. Eine grundsätzliche Neujustierung des Strahlengangs kann mehrere Stunden in Anspruch nehmen.

Der Aufbau war wie erwartet justiert. Es gab keine Besonderheiten oder Probleme mit dem Aufbau.

3.1.3 Aufgabe 1.3: Messung

- Stellen Sie, zur Kalibration des Frequenzzählers, die Rotationsfrequenz des Drehspiegels, mit Hilfe der auftretenden Schwebung zwischen dem Motorengeräusch des Drehspiegels und dem Stimmgabelton für den Kammerton A (440 Hz) ein. Vergleichen Sie den erhaltenen Wert mit der elektronischen Frequenzanzeige und halten Sie den Kalibrationsfaktor in Ihrem Protokoll fest.
- Tragen Sie eine geeignete Anzahl an Messpunkten für den Versatz s des beobachteten Lichtpunkts für verschiedene Rotationsfrequenzen des Drehspiegels auf und bestimmen Sie daraus c .

```
[2]: uf = ufloat(5,0.01)
ub = ufloat(7.23,0.01)
ua = ufloat(6.75,0.01)
ul = uf**2/(ua+ub-2*uf)

uv_corr = ufloat(440,3)
ucorr = 440 / uv_corr

%m In dieser Aufgabe geht es darum die Lichtgeschwindigkeit über den Versatz
↳ der Laserstrahlen mit dem Drehspiegel zu bestimmen. Wenn der Motor synchron
↳ mit der Schwingung der Stimmgabel dreht zeigt die Anzeige einen Wert von [!
↳ uv_corr.n!]Hz mit einer Unsicherheit von [!uv_corr.s!]Hz. Damit ergibt sich
↳ ein Korrekturfaktor von [!ucorr.n:.2f!].

%m Die Messung bei verschiedenen Frequenzen ergibt die folgenden Messwerte:

a11 = """
v,dv,s,ds
1550,120, 24e-3,0.1e-3
5330,120, 24.5e-3,0.1e-3
9250,120, 25e-3,0.1e-3
13820,120, 25.5e-3,0.1e-3
17300,120, 26e-3,0.1e-3
21700,120, 26.5e-3,0.1e-3
25280,120, 27e-3,0.1e-3
29250,120, 27.5e-3,0.1e-3
"""

mpg = pd.read_csv(StringIO(a11))
mpg["v"] = mpg["v"]/60
mpg["dv"] = mpg["dv"]/60
mpg.rename(columns={"v": 'Drehzahl in Hz', "dv": "Unsicherheit in Hz","s":
↳ "Distanz in m", "ds": "Unsicherheit in m"}, inplace=True)
display(mpg)
```

```

v, dv, s, ds = np.genfromtxt(StringIO(a11), delimiter=',')[1:].transpose()
uv = uarr(v/60,dv/60)
us = uarr(s,ds)

%m Da der Laserstrahl einen Durchmesser von 1mm hat und die Skala auch nur auf
↳ einen Millimeter genau ist wird der Fehler auf 0.1mm abgeschätzt. Die
↳ Frequenz schwankte stark im Bereich von  $\pm 120$  U/min. Deswegen wurden 2
↳ Hz als Unsicherheit gewählt.

%m Über eine lineare Regresion kann nun aus diesen Werten die
↳ Lichtgeschwindigkeit im Medium bestimmt werden. Dabei wird es einen
↳ y-Achsenabschnitt geben, weil nicht genau bestimmt wurde wo der Strahl bei
↳ Stillstand des Rotationsspiegels auftrifft. Das ist aber kein Problem, das
↳ sich die Lichtgeschwindigkeit nach der in der letzten Aufgabe genannten
↳ Formel aus der Steigung bestimmen lässt. Das modell ist also  $s =$ 
↳  $8\pi \frac{v}{c} \nu (a+b)$  mit dem
↳ y-Achsenabschnitt  $d$ .

def model(v,a,b,c,l,d):
    s = 8*pi*l*v*(a+b)/c + d
    return s

xy_data = kafe2.XYContainer(x_data=nomv(uv), y_data=nomv(us))
xy_data.add_error(axis='x', err_val=stdv(uv))
xy_data.add_error(axis='y', err_val=stdv(us))

lin_fit = kafe2.Fit(data=xy_data, model_function=model)
lin_fit.add_parameter_constraint(name='l',value=ul.n,uncertainty=ul.s)
lin_fit.add_parameter_constraint(name='a',value=ua.n,uncertainty=ua.s)
lin_fit.add_parameter_constraint(name='b',value=ub.n,uncertainty=ub.s)
lin_fit.do_fit()

p = kafe2.Plot(lin_fit)
p.x_label = 'Frequenz in Hz'
p.y_label = 'Distanz in m'
p.plot()
p.show()

_,_,c_luft,_,d = lin_fit.parameter_values
_,_,Δc_luft,_,Δd = lin_fit.parameter_errors
uc_luft = ufloat(c_luft,Δc_luft)

uc = uc_luft*1.000292
ud = ufloat(d,Δd)

```

Die Anpassung mit Kafe2 ergibt einen Wert von $\frac{m}{s} \pm$
 $\frac{m}{s}$, wobei noch mit dem Literaturwert von Luft
 $n_{\text{Luft}} = 1.000292$ skaliert wurde (Quelle <https://de.wikipedia.org/wiki/Brechungsindex> abgerufen am 01.12.2023). Folglich ist der gemessene Wert im
Rahmen der Unsicherheit mit der Definition für die Lichtgeschwindigkeit
kompatibel. An den Werten für χ^2/dof und χ^2_{prop} ist zu sehen,
dass es sich um einen einigermaßen guten Fit handelt, die Unsicherheiten für
die Position des Lichtpunktes aber zu groß gewählt wurden. Hier wäre eine
Auswertung des Kamerabildes mit dem Computer sicherlich genauer.

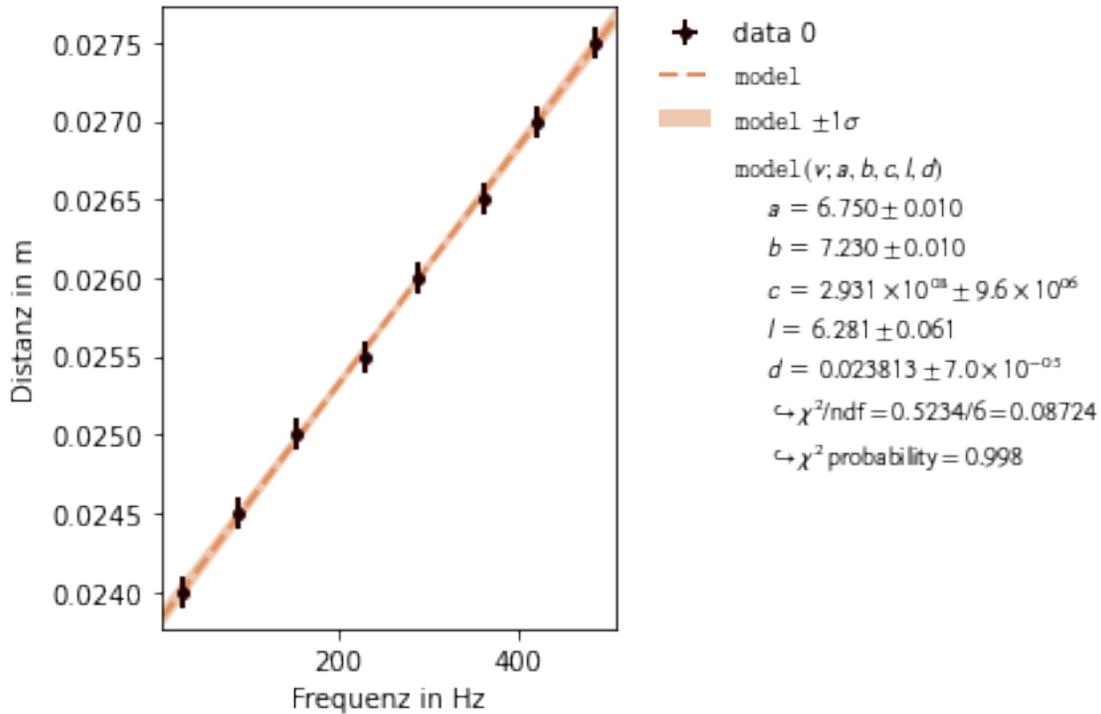
In dieser Aufgabe geht es darum die Lichtgeschwindigkeit über den Versatz der Laserstrahlen mit dem Drehspiegel zu bestimmen. Wenn der Motor synchron mit der Schwingung der Stimmgabel dreht zeigt die Anzeige einen Wert von 440.0Hz mit einer Unsicherheit von 3.0Hz. Damit ergibt sich ein Korrekturfaktor von 1.00.

Die Messung bei verschiedenen Frequenzen ergibt die folgenden Messwerte:

	Drehzahl in Hz	Unsicherheit in Hz	Distanz in m	Unsicherheit in m
0	25.833333	2.0	0.0240	0.0001
1	88.833333	2.0	0.0245	0.0001
2	154.166667	2.0	0.0250	0.0001
3	230.333333	2.0	0.0255	0.0001
4	288.333333	2.0	0.0260	0.0001
5	361.666667	2.0	0.0265	0.0001
6	421.333333	2.0	0.0270	0.0001
7	487.500000	2.0	0.0275	0.0001

Da der Laserstrahl einen Durchmesser von 1mm hat und die Skala auch nur auf einen Millimeter genau ist wird der Fehler auf 0.1mm abgeschätzt. Die Frequenz schwankte stark im Bereich von $\pm 120U/\text{min}$. Deswegen wurden 2 Hz als Unsicherheit gewählt.

Über eine lineare Regresion kann nun aus diesen Werten die Lichtgeschwindigkeit im Medium bestimmt werden. Dabei wird es einen y-Achsenabschnitt geben, weil nicht genau bestimmt wurde wo der Strahl bei Stillstand des Rotationsspiegels auftrifft. Das ist aber kein Problem, das sich die Lichtgeschwindigkeit nach der in der letzten Aufgabe genannten Formel aus der Steigung bestimmen lässt. Das Modell ist also $s = 8\pi \frac{\ell \nu (a+b)}{c_{\text{Luft}}} + d$ mit dem y-Achsenabschnitt d .



Die Anpassung mit Kafé2 ergibt einen Wert von $293219457.21 \frac{m}{s} \pm 9578567.51 \frac{m}{s}$, wobei noch mit dem Literaturwert von Luft $\epsilon_{Luft} = 1.000292$ skaliert wurde (Quelle <https://de.wikipedia.org/wiki/Brechungsindex> abgerufen am 01.12.2023). Folglich ist der gemessene Wert im Rahmen der Unsicherheit mit der Definition für die Lichtgeschwindigkeit kompatibel. An den Werten für χ^2/dof und χ^2_{prop} ist zu sehen, dass es sich um einen einigermaßen guten Fit handelt, die Unsicherheiten für die Position des Lichtpunktes aber zu groß gewählt wurden. Hier wäre eine Auswertung des Kamerabildes mit dem Computer sicherlich genauer.

3.2 Aufgabe 2: Phasenvergleichsmethode

Hinweise zu allen hier durchzuführenden Messungen finden Sie in der Datei [Hinweise-Aufgabe-2.md](#).

3.2.1 Aufgabe 2.1: Vorbereitung

Beantworten Sie zur Vorbereitung auf den Versuch die folgenden Fragen:

- Welche Größenordnung sollte die Frequenz ν der Lichtwelle haben, damit Sie mit dem vorliegenden Aufbau (siehe Datenblatt.md), zwischen Empfänger- und Referenzsignal, eine Phasendifferenz von $\Delta\varphi = 0, 2\pi$, d.h. einem 1/10 der Periodendauer, auflösen können?
- Welche Ablenkgeschwindigkeit u sollte ein Oszilloskop entlang der Zeitachse haben, damit eine solche Phasenverschiebung zu einer wahrnehmbaren Verschiebung z.B. von $\Delta d = 5 \text{ mm}$ führt? Zu Ihrer Information: Für ein einfaches Oszilloskops beträgt die Ablenkgeschwindigkeit entlang der Zeitachse $u \lesssim 10 \text{ cm/s}$.

```

[3]: %m Beim folgenden Experiment wird nun der an einem Oszilloskop angezeigte
      ↳ zeitliche Versatz zwischen Referenzsignal und Lichtstrahl bei bestimmter
      ↳ Distanz gemessen und durch Überlagern des Signals mit einem frequentiell
      ↳ nahen Signal die zeitliche Auflösung des Oszilloskops verbessert.

l = 1 #(Annahme)
c_lit = 2.99792458E8 #Literaturwert
      = 2*np.pi*60E6
_dot = 2*np.pi*59.9E6
_bar = -_dot

      = 1/10*c_lit/l
%m Für eine Phasendifferenz zwischen Empfänger- und Referenzsignal von  $\Delta\varphi = \frac{2\pi}{10}$ 
      ↳ bei Annahme der Lichtgeschwindigkeit im Vakuum
      ↳ und einer Distanz von einem Meter sollte die Lichtwelle eine Frequenz von ca.
      ↳  $\nu = [! :.0f!] \backslash, \text{Hz}$ , also in der Größenordnung von  $10^{\lceil \text{int}(\text{np.log}()/\text{np.}
      ↳ \log(10)) \rceil} \backslash, \text{Hz}$ , haben.

Ad = 5E-3

u_erw = _bar / * c_lit * Ad / l
%m Für eine Länge der Apparatur von einem Meter und die
      ↳ Vakuumlichtgeschwindigkeit erwartet man eine Ablenkgeschwindigkeit  $u = [!
      ↳ u_erw * 1E-4 : .2f!] \backslash, \frac{\text{cm}}{\mu\text{s}}$ , um eine Verschiebung von  $\Delta d = [!
      ↳ 5 \backslash, \text{mm}$  zu erhalten. Dies ist kleiner als die übliche Ablenkgeschwindigkeit
      ↳ von  $u = 10 \backslash, \frac{\text{cm}}{\mu\text{s}}$  eines Oszilloskops, also gut beobachtbar.

```

Beim folgenden Experiment wird nun der an einem Oszilloskop angezeigte zeitliche Versatz zwischen Referenzsignal und Lichtstrahl bei bestimmter Distanz gemessen und durch Überlagern des Signals mit einem frequentiell nahen Signal die zeitliche Auflösung des Oszilloskops verbessert.

Für eine Phasendifferenz zwischen Empfänger- und Referenzsignal von $\Delta\varphi = \frac{2\pi}{10}$ bei Annahme der Lichtgeschwindigkeit im Vakuum und einer Distanz von einem Meter sollte die Lichtwelle eine Frequenz von ca. $\nu = 29979246 \text{ Hz}$, also in der Größenordnung von 10^7 Hz , haben.

Für eine Länge der Apparatur von einem Meter und die Vakuumlichtgeschwindigkeit erwartet man eine Ablenkgeschwindigkeit $u = 0.25 \frac{\text{cm}}{\mu\text{s}}$, um eine Verschiebung von $\Delta d = 5 \text{ mm}$ zu erhalten. Dies ist kleiner als die übliche Ablenkgeschwindigkeit von $u = 10 \frac{\text{cm}}{\mu\text{s}}$ eines Oszilloskops, also gut beobachtbar.

Aufgabe 2.2: Justierung der Apparatur

Überprüfen Sie die Verbindungskabel anhand des Anschlussschemas auf der Frontplatte des Hauptnetzgeräts (siehe Skizze). Richten Sie einen möglichst parallelen Strahl vom Sender in Richtung des Empfängers ein. Nutzen Sie hierzu die Justierschrauben zur Zentrierung der Leuchtdiode des Senders und des verschiebbaren Kondensators am Lichtsendergehäuse. Stellen Sie die Sammellinse vor der Photodiode des Empfängers so auf, dass die Photodiode optimal ausgeleuchtet wird. Achtung: Stellen Sie sicher, dass das Lichtsignal des Senders auch bei maximalem Abstand zwischen Sender und Empfänger noch erkannt wird. Das Empfängersignal sollte bei jedem Abstand zwischen Sender

und Empfänger ausreichend groß sein, die Amplitude am Empfänger sollte bei allen Entfernungen $\pm 2V$ jedoch nicht überschreiten, weil die internen Mischerstufen der Elektronik sonst übersteuern und dabei die Phasenlage verfälschen.

Notieren Sie sich alle Besonderheiten, die Ihnen dabei auffallen.

Am Oszilloskop gab es ein Rädchen zum Einstellen eines zusätzlichen Phasenversatzes am Oszilloskop, welches leider verstellt wurde. Um diesen Fehler zu beheben, wurde ein neuer Offset gemessen. Dieser wurde bestimmt, indem für die kleinste messbare Distanz des Laserdistanzmessers der Phasenversatz auf 0 gesetzt wurde und diese Länge als neue 0 von allen Messwerten abgezogen wird. Er ergab sich zu $l_{off} = (0.2 \pm 0.005) m$.

3.2.2 Aufgabe 2.3: Messung

- Überprüfen und notieren Sie mit dem Oszilloskop die Differenzfrequenz $\bar{\omega} = (\omega - \omega')$. Dabei entspricht $\omega = 2\pi \cdot 60 \text{ MHz}$ der Kreisfrequenz des für die Messung verwendeten Referenzsignals und $\omega' = 2\pi \cdot 59,9 \text{ MHz}$ der Kreisfrequenz des beigemischten Signals zum Zweck der Amplitudenmodulation. **Achtung:** Für eine zuverlässige Frequenzmessung benötigt das Oszilloskop mehrere vollständige Wellenzüge im Anzeigebereich.
- Messen Sie, aus der Phasenverschiebung $\Delta\varphi$ zwischen Empfänger- und Referenzsignal, die zeitliche Differenz Δt zwischen Sender- und Empfängersignal, als Funktion des Abstands ℓ des Senders vom Empfänger. Bestimmen Sie aus der Auftragung einer geeigneten Anzahl an Messpunkten c mit entsprechender Unsicherheit Δc .
- Beschließen Sie diese Aufgabe damit, dass Sie c durch die Bestimmung von $\lambda/2$ mit Hilfe einer [Lissajous-Figur](#) bestimmen. Betreiben Sie das Oszilloskop hierzu im X/Y-Modus. **Achtung:** Da Sie hierzu fast die gesamte zur Verfügung stehende Strecke der Führungsschiene benötigen, müssen Sie die Phase am Betriebsgerät bei minimalem Abstand geeignet einstellen.

```
[4]: l_off = ufloat(0.2,0.005)
      = 2*np.pi*ufloat(60E6,1e3)
      _dot = 2*np.pi*ufloat(59.9E6,1e3)
      #AbleSEN und Messen der zeitlichen Ablenkung bei gegebener Länge
      _bar = ufloat( .n- _dot.n,20) #Abgelesen aus FT
      Δl = 1.5e-3
      arrl = [ufloat(0.261,Δl),
              ufloat(334e-3,Δl),
              ufloat(412e-3,Δl),
              ufloat(507e-3,Δl),
              ufloat(587e-3,Δl),
              ufloat(686e-3,Δl),
              ufloat(742e-3,Δl),
              ufloat(825e-3,Δl),
              ufloat(909e-3,Δl),
              ufloat(999e-3,Δl),
              ufloat(1072e-3,Δl),
              ufloat(1167e-3,Δl),
              ufloat(1360e-3,Δl),
              ufloat(1622e-3,Δl),
```

```

    ufloat(1872e-3,Δl),
    ufloat(2128e-3,Δl),
    ufloat(2353e-3,Δl),
    ufloat(2598e-3,Δl)] #Messen bei 12 verschiedenen Längen, eintragen hier
↳#TODO Fehler abschätzen
Δt_bar = 15e-9
arrΔt_bar = [ufloat(138.8e-9,Δt_bar),
             ufloat(285.7e-9,Δt_bar),
             ufloat(449e-9,Δt_bar),
             ufloat(625.9e-9,Δt_bar),
             ufloat(789.1e-9,Δt_bar),
             ufloat(993.2e-9,Δt_bar),
             ufloat(1067e-9,Δt_bar),
             ufloat(1246e-9,Δt_bar),
             ufloat(1418e-9,Δt_bar),
             ufloat(1614e-9,Δt_bar),
             ufloat(1763e-9,Δt_bar),
             ufloat(1951e-9,Δt_bar),
             ufloat(2240e-9,Δt_bar),
             ufloat(2743e-9,Δt_bar),
             ufloat(3222e-9,Δt_bar),
             ufloat(3733e-9,Δt_bar),
             ufloat(4169e-9,Δt_bar),
             ufloat(4653e-9,Δt_bar)] #abgelesene Zeitdifferenzen #TODO Fehler
↳abschätzen
arrl = [i-1_off for i in arrl]
%m Bei der Durchführung des Experiments wurden nun folgende Zeiten gemessen:
table = pd.DataFrame({
    "l [m]": arrl,
    r"$\Delta\bar{t}$ [s]": arrΔt_bar,
})
display(Markdown(table.to_markdown()))
%m Dabei wurde eine Unsicherheit von $[!_bar.s!]\,Hz$ auf $\bar{\omega}$
↳angenommen, bestimmt aus der am Oszilloskop abgelesenen
↳Fouriertransformationen. Auf die einzelnen Längen wurde eine Unsicherheit
↳von $[!arrl[0].s*1e3:.3!]\,mm$ angenommen, größer als die im Datenblatt
↳angegebene Unsicherheit von $1.5\,mm$ des Laserdistanzmessers, um Messfehler
↳beim Verschieben der Schiene und insbesondere des Setzens des Nullpunktes zu
↳berücksichtigen und auf das Ablesen des Zeitversatzes eine Unsicherheit von
↳$[!\Delta_bar*1e9:.3!]\,ns$, da die Schwingungen rauschten.
%m Die Daten werden nun ein erstes Mal gegen die annähernd erwartete Gerade der
↳Vakuumllichtgeschwindigkeit geplottet.
#erstes Plotten
arrΔ = [_bar * i for i in arrΔt_bar]
x = [i.n for i in arrl]
xerror = [arrl.s for arrl in arrl]
y = [i.n for i in arrΔ]

```

```

yerror = [arrΔ .s for arrΔ in arrΔ ]
plt.scatter(x,y)
k = np.linspace(x[0],x[len(x)-1])
g = [ .n/c_lit*i for i in k]
plt.plot(k,g)
plt.title("Gemessener Phasenversatz bei gegebener Distanz")
plt.xlabel("Distanz zum Empfänger in m")
plt.ylabel("Phasenversatz in Rad")
plt.show()
tmp=''
plt.errorbar(x,y,fmt='.',xerr=xerror,yerr=yerror)
plt.title("Phasenverschiebung in Abhängigkeit von der Distanz")
plt.xlabel("Distanz des Empfängers in m")
plt.ylabel(r"$\Delta \varphi$ in Rad")
plt.show()'''
%m Es ist gut zu erkennen, dass die ersten 12 Punkte der Geraden sehr gut
↳ folgen, während die späteren einen konstanten Offset nach unten zu haben
↳ scheinen, der durch eine Pausierung und eine Beeinflussung der Messung
↳ zustande kam. Da zur Bestimmung der Lichtgeschwindigkeit nur die Steigung
↳ relevant ist, wird nun ein Multifit für die beiden Hälften durchgeführt, die
↳ sich um den Offset unterscheiden.Es wird ein Modell der Form  $\bar{\Delta t}$ 
↳  $= \frac{\omega * l}{c * \bar{\omega}} + b$  gefittet.

#Splitting the data
x1 = x[:12]
y1 = y[:12]
xerror1 = xerror[:12]
yerror1 = yerror[:12]

x2 = x[12:]
y2 = y[12:]
xerror2 = xerror[12:]
yerror2 = yerror[12:]

# Defining Models
def diffModel1(l,w= .n,c=2.98e8,b1=0):
    return w/c * l +b1

def diffModel2(l,w= .n,c=2.98e8,b2=0):
    return w/c * l +b2

#Creating the Fits
fitlist=["",""]

xy_data = kafe2.XYContainer(x_data = x1, y_data=y1)
xy_data.label = 'Erste Hälfte'
fitlist[0] = kafe2.Fit(data=xy_data, model_function=diffModel1)

```

```

fitlist[0].add_error("x",xerror1)
fitlist[0].add_error("y",yerror1)

xy_data = kafe2.XYContainer(x_data = x2, y_data=y2)
xy_data.label = 'Zweite Hälfte'
fitlist[1] = kafe2.Fit(data=xy_data, model_function=diffModel2)
fitlist[1].add_error("x",xerror2)
fitlist[1].add_error("y",yerror2)

#Defining the multiFit
multi_fit = kafe2.MultiFit(fit_list=fitlist, minimizer='iminuit')
multi_fit.add_parameter_constraint("w",value= .n,uncertainty= .s)

# Do the fit
multi_fit.do_fit()

# Plotting
p = kafe2.Plot(multi_fit, separate_figures=False)
p.x_label = "Distanz des Empfängers in m"
p.y_label = r"$\Delta \varphi$ in Rad"
p.plot()
p.show()

#Values
#multi_fit.report()
_,c,_,_ = multi_fit.parameter_values
_,Δc,_,_ = multi_fit.parameter_errors

tmp=''
##Fitting
%m An diese Daten wird nun ein Modell der Form  $\bar{\Delta t} = \frac{\omega}{c} l + b$ 
↳ gefittet. Der konstante Versatz b soll hier den
↳ eingestellten Offset am Oszilloskop berücksichtigen. und wird in der
↳ nachfolgenden Aufgabe zu Lissajous-Figuren von den Phasenwerten abgezogen.
def diffModel(l,w= .n,c=2.98e8,b=0):
    return w/c * l +b
xy_data = kafe2.XYContainer(x_data = x, y_data=y)
xy_data.label = 'Phasenverschiebung in Abhängigkeit von der Distanz'
diff_fit = kafe2.Fit(data=xy_data, model_function=diffModel)
diff_fit.add_error("x",xerror)
diff_fit.add_error("y",yerror)
diff_fit.add_parameter_constraint(name="w",value= .n,uncertainty= .s)
diff_fit.do_fit()
_,c,b = diff_fit.parameter_values
_,Δc,Δb = diff_fit.parameter_errors
#offset = ufloat(offset,Δoffset)
p = kafe2.Plot(diff_fit)

```

```

p.x_label = "Distanz des Empfängers in m"
p.y_label = r"$\Delta \varphi$ in Rad"
p.plot()
p.show()'''

#Auswertung
%m Die globale  $\chi^2$ -Wahrscheinlichkeit und  $\frac{\chi^2}{\text{ndof}}$  sind recht
↳nahe an eins, der Fit ist also akzeptabel. Der Literaturwert ist innerhalb
↳des drei Sigma Intervalls, ist also durchaus unwahrscheinlich, aber immer
↳noch kompatibel. Wie im ersten Plot erkennbar, streuen die Punkte alle ein
↳wenig um die Gerade, sodass nicht nur ein systematischer, sondern auch ein
↳statistischer Fehler relevant ist. Die größten systematische Fehlerquellen
↳waren vermutlich die Nullsetzung der Phase am Oszilloskop, führte diese doch
↳dazu die Daten einmal zu splitten, und das korrekte Ablesen des zeitlichen
↳Versatzes am Oszilloskops. Sofern man diese beiden allerdings bei mehreren
↳Messreihen auch immer neu einstellen würde, würden allerdings auch diese
↳Fehler zu einem statistischen werden, der sich im Grenzwert unendlicher
↳Messungen (bzw. hier Messreihen), herausmitteln würde.

#Man erkennt, dass die Punkte dem Fit im Plot scheinbar sehr gut folgen,
↳allerdings ist die  $\chi^2$ -Wahrscheinlichkeit recht niedrig, dafür ist
↳ $\frac{\chi^2}{\text{ndof}}$  recht nah an 1. Kritisch ist die gemessene
↳Lichtgeschwindigkeit, #TODO

### Messung durch eine Lissajous Figur #TODO Wie viele Werte? #TODO: Auch
↳fitt*en?

%m Es wird das Oszilloskop nun auf X/Y-Betrieb gestellt. Es wird das
↳Eingangssignal gegen das Ausgangssignal dargestellt und der Sender so lange
↳verschoben, bis feste Lissajous-Figuren entstehen. Für diese ist die
↳Phasenverschiebung bekannt und es wird die zugehörige Länge gemessen.
Δ_Liss = np.pi/2
l_Liss = ufloat(1394e-3,30e-3)-ufloat(153e-3,30e-3)#[153e-3,1394e-3]
c_Liss = .n*l_Liss/Δ_Liss
#print(c_Liss)

%m Es ergibt sich eine Lichtgeschwindigkeit von  $c = (c_Liss.n:.3!)\pm(c_Liss.s:.3!)\backslash\frac{m}{s}$ . Die hohe Unsicherheit lässt sich durch die
↳begrenzte Schiene des Lasers erklären, die es lediglich erlaubte zwei
↳Lissajous-Figuren aufzunehmen, von denen eine der Kreis war, der durch die
↳verschiedenen Achsenskalierungen und die Unterscheidung von perfektem Kreis
↳von leicht verschobenen schwierig zu bestimmen war. Die Messung ist äußerst
↳kompatibel mit dem Literaturwert.

```

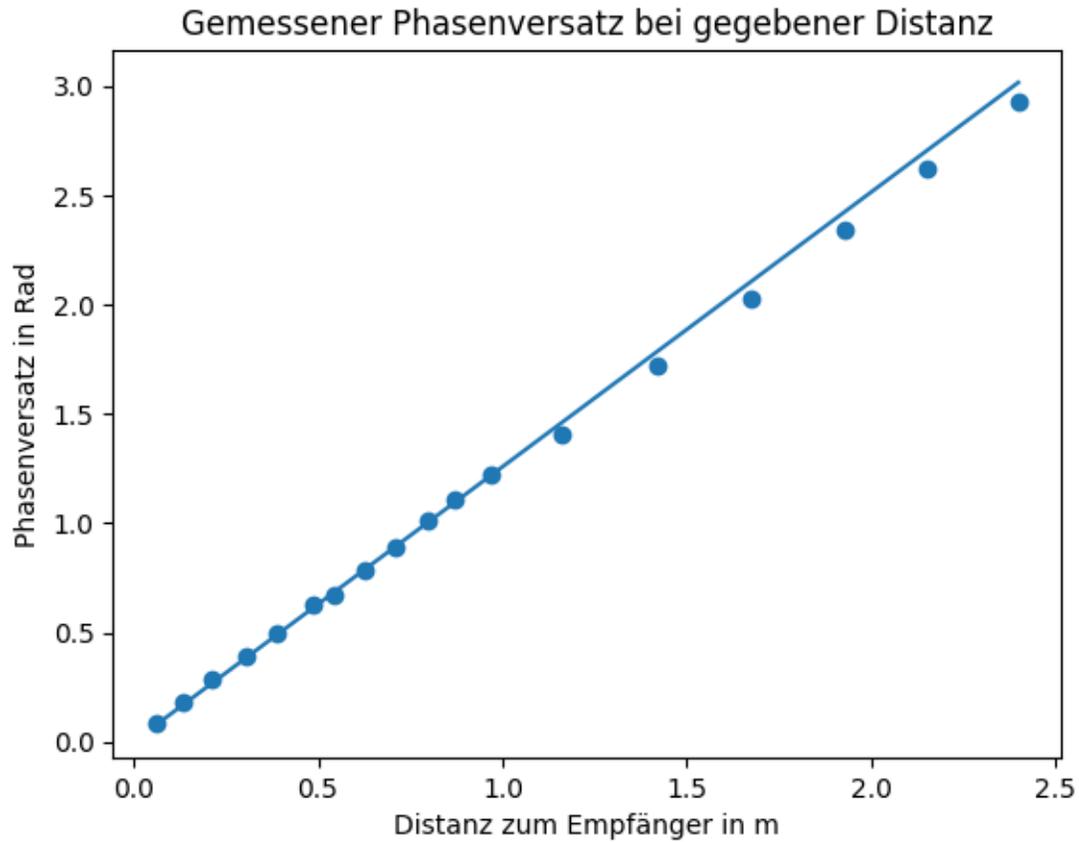
Bei der Durchführung des Experiments wurden nun folgende Zeiten gemessen:

	l [m]	$\Delta \bar{t}$ [s]
0	0.061±0.005	(1.39±0.15)e-07
1	0.134±0.005	(2.86±0.15)e-07
2	0.212±0.005	(4.49±0.15)e-07

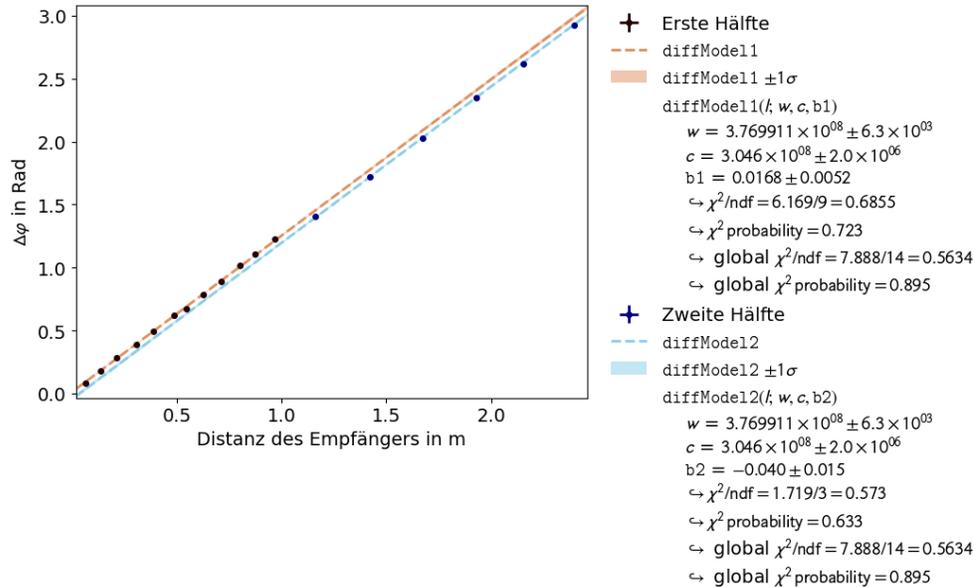
	l [m]	$\Delta\bar{t}$ [s]
3	0.307+/-0.005	(6.26+/-0.15)e-07
4	0.387+/-0.005	(7.89+/-0.15)e-07
5	0.486+/-0.005	(9.93+/-0.15)e-07
6	0.542+/-0.005	(1.067+/-0.015)e-06
7	0.625+/-0.005	(1.246+/-0.015)e-06
8	0.709+/-0.005	(1.418+/-0.015)e-06
9	0.799+/-0.005	(1.614+/-0.015)e-06
10	0.872+/-0.005	(1.763+/-0.015)e-06
11	0.967+/-0.005	(1.951+/-0.015)e-06
12	1.160+/-0.005	(2.240+/-0.015)e-06
13	1.422+/-0.005	(2.743+/-0.015)e-06
14	1.672+/-0.005	(3.222+/-0.015)e-06
15	1.928+/-0.005	(3.733+/-0.015)e-06
16	2.153+/-0.005	(4.169+/-0.015)e-06
17	2.398+/-0.005	(4.653+/-0.015)e-06

Dabei wurde eine Unsicherheit von 20.0 Hz auf $\bar{\omega}$ angenommen, bestimmt aus der am Oszilloskop abgelesenen Fouriertransformationen. Auf die einzelnen Längen wurde eine Unsicherheit von 5.22 mm angenommen, größer als die im Datenblatt angegebene Unsicherheit von 1.5 mm des Laserdistanzmessers, um Messfehler beim Verschieben der Schiene und insbesondere des Setzens des Nullpunktes zu berücksichtigen und auf das Ablesen des Zeitversatzes eine Unsicherheit von 15.0 ns , da die Schwingungen rauschten.

Die Daten werden nun ein erstes Mal gegen die annähernd erwartete Gerade der Vakuumlichtgeschwindigkeit geplottet.



Es ist gut zu erkennen, dass die ersten 12 Punkte der Geraden sehr gut folgen, während die späteren einen konstanten Offset nach unten zu haben scheinen, der durch eine Pausierung und eine Beeinflussung der Messung zustande kam. Da zur Bestimmung der Lichtgeschwindigkeit nur die Steigung relevant ist, wird nun ein Multifit für die beiden Hälften durchgeführt, die sich um den Offset unterscheiden. Es wird ein Modell der Form $\bar{\Delta}t = \frac{\omega * l}{c * \bar{\omega}} + b$ gefittet.



Die globale χ^2 -Wahrscheinlichkeit und $\frac{\chi^2}{ndof}$ sind recht nahe an eins, der Fit ist also akzeptabel. Der Literaturwert ist innerhalb des drei Sigma Intervalls, ist also durchaus unwahrscheinlich, aber immer noch kompatibel. Wie im ersten Plot erkennbar, streuen die Punkte alle ein wenig um die Gerade, sodass nicht nur ein systematischer, sondern auch ein statistischer Fehler relevant ist. Die größten systematische Fehlerquellen waren vermutlich die Nullsetzung der Phase am Oszilloskop, führte diese doch dazu die Daten einmal zu splitten, und das korrekte Ablesen des zeitlichen Versatzes am Oszilloskops. Sofern man diese beiden allerdings bei mehreren Messreihen auch immer neu einstellen würde, würden allerdings auch diese Fehler zu einem statistischen werden, der sich im Grenzwert unendlicher Messungen (bzw. hier Messreihen), herausmitteln würde.

Es wird das Oszilloskop nun auf X/Y-Betrieb gestellt. Es wird das Eingangssignal gegen das Ausgangssignal dargestellt und der Sender so lange verschoben, bis feste Lissajous-Figuren entstehen. Für diese ist die Phasenverschiebung bekannt und es wird die zugehörige Länge gemessen.

Es ergibt sich eine Lichtgeschwindigkeit von $c = (2.98e + 08 \pm 1.02e + 07) \frac{m}{s}$. Die hohe Unsicherheit lässt sich durch die begrenzte Schiene des Lasers erklären, die es lediglich erlaubte zwei Lissajous-Figuren aufzunehmen, von denen eine der Kreis war, der durch die verschiedenen Achsenskalierungen und die Unterscheidung von perfektem Kreis von leicht verschobenen schwierig zu bestimmen war. Die Messung ist äußerst kompatibel mit dem Literaturwert.

3.3 Aufgabe 3: Bestimmung des Brechungsindex n von Wasser oder Plexiglas

Hinweise zu allen hier durchzuführenden Messungen finden Sie in der Datei [Hinweise-Aufgabe-3.md](#).

3.3.1 Aufgabe 3.1: Bestimmung von n mit der Apperatur von Aufgabe 2

Bestimmen Sie **mit dem Aufbau von Aufgabe 2** aus der Messung der Laufzeitdifferenz des Lichtstrahls im optisch dichteren Medium den Brechungsindex von (n_{H_2O}) Wasser oder (n_{Plex})

Plexiglas. Für Wasser ersetzen Sie 1m Lichtweg in Luft durch 1m Lichtweg in Wasser. Für Plexiglas verwenden Sie die 8 und 30cm langen Plexiglaszylinder. Die Messung mit *einem* der angebotenen Medien erfüllt diese Aufgabe bereits.

```
[5]: %m Es wird der Brechungsindex von Wasser und Plexiglas vermessen. Dazu wird der
↳ Aufbau aus 3.1 genutzt, allerdings ein Teil der Laufstrecke mit dem zu
↳ vermessenden Medium gefüllt.
%m Vor den Messungen wurde der Phasenversatz am Oszilloskop bei $[!l_off!]\,m$
↳ auf 0 gesetzt und dieser Wert wird nun von allen vermessenen Längen
↳ abgezogen.
%m Die Daten werden jeweils an ein Modell der Form $\bar{\Delta t}$
↳ = $\frac{\omega}{\bar{\omega}c}(d*(n_{Material}-n_{Luft})+1)$ gefittet.
tmp=''
d = ufloat(,) #Strecke im Medium #TODO: Fehler abschätzen
l = ufloat(,) -l_off#Gesamtlichtweg #TODO: Fehler abschätzen
#TODO: Für c Literaturwert nehmen?
At_bar_Medium = ufloat(,) #gemessener Zeitversatz #TODO: Fehler abschätzen

At_Medium = At_bar_Medium * _bar/

n = (c_lit*At_Medium-l)/d +1
%m Es ergibt sich ein Brechungsindex von $n = [!nomv(n):.3!]\pm [!stdv(n):.3!]$.
↳ Dies ist im Rahmen der Messungenauigkeiten #TODO Vergleich'''
Δt = 15e-9
#l_off_P1 = ufloat(432e-3,0.0015)
d_P1 = 30e-2#40.1e-2
a_P1 = ""
l,dt
0539e-3, 1034e-9
0700e-3, 1344e-9
0879e-3, 1703e-9
1058e-3, 2057e-9
1239e-3, 2449e-9
1419e-3, 2781e-9
1599e-3, 3151e-9
1776e-3, 3505e-9
1957e-3, 3842e-9
""

mpg = pd.read_csv(StringIO(a_P1))
mpg["l"] = mpg["l"] - l_off.n
%m Messung in Plexiglas der Länge $[!d_P1!]\,m$
mpg.rename(columns={"l":"Distanz in m","dt":"Gemessener Zeitversatz"},
↳ inplace=True)
display(mpg)
```

```

l_P1,dt_P1 = np.genfromtxt(StringIO(a_P1), delimiter=',')[1:].transpose()
l_P1 = [i-l_off.n for i in l_P1] #offset abziehen

def NP1Model(l,n=1,w_bar=_bar.n,w=.n,d=d_P1):
    return 1/2.99792458e8*w/w_bar*(d*(n-1.000292)+l)
xy_data = kafe2.XYContainer(x_data=l_P1 , y_data=dt_P1)
xy_data.label = 'Zeitverschiebung in Abhängigkeit von der Distanz'
n_fit = kafe2.Fit(data=xy_data, model_function=NP1Model)
n_fit.add_error("x",0.015)
n_fit.add_error("y",15e-9)
n_fit.add_parameter_constraint(name="w",value=.n,uncertainty=.s)
n_fit.add_parameter_constraint(name="w_bar",value=_bar.n,uncertainty=_bar.s)
n_fit.add_parameter_constraint(name="d",value=d_P1,uncertainty=1e-5)
n_fit.do_fit()
n_P1,_,_,_ = n_fit.parameter_values
Δn_P1,_,_,_ = n_fit.parameter_errors
p = kafe2.Plot(n_fit)
p.x_label = "Distanz des Empfängers in m"
p.y_label = "Zeitversatz in s"
p.plot()
p.show()

#l_off_P2 = ufloat(211e-3,0.0015)
d_P2 = 8e-2#18e-2 #Strecke im Medium #
a_P2 = """
l,dt
0339e-3, 424.5e-9
0514e-3, 794.6e-9
0699e-3, 1137e-9
0878e-3, 1518e-9
1053e-3, 1839e-9
1238e-3, 2210e-9
1418e-3, 2580e-9
1599e-3, 2957e-9
1777e-3, 3271e-9
1957e-3, 3603e-9
"""

mpg = pd.read_csv(StringIO(a_P2))
mpg["l"] = mpg["l"] - l_off.n
%m Messung in Plexiglas der Länge $[!d_P2!]\,m$
mpg.rename(columns={"l":"Distanz in m","dt":"Gemessener Zeitversatz"},  

           ↪inplace=True)
display(mpg)

l_P2,dt_P2 = np.genfromtxt(StringIO(a_P2), delimiter=',')[1:].transpose()
l_P2 = [i-l_off.n for i in l_P2]

```

```

def NP2Model(l,n=1,w_bar= _bar.n,w= .n,d=d_P2):
    return 1/2.99792458e8*w/w_bar*(d*(n-1.000292)+1)
xy_data = kafe2.XYContainer(x_data =l_P2 , y_data=dt_P2)
xy_data.label = 'Zeitverschiebung in Abhängigkeit von der Distanz'
n_fit = kafe2.Fit(data=xy_data, model_function=NP2Model)
n_fit.add_error("x",0.015)
n_fit.add_error("y",15e-9)
n_fit.add_parameter_constraint(name="w",value= .n,uncertainty= .s)
n_fit.add_parameter_constraint(name="w_bar",value= _bar.n,uncertainty= _bar.s)
n_fit.add_parameter_constraint(name="d",value=d_P2,uncertainty=1e-5)
n_fit.do_fit()
n_P2,_,_,_ = n_fit.parameter_values
Δn_P2,_,_,_ = n_fit.parameter_errors
p = kafe2.Plot(n_fit)
p.x_label = "Distanz des Empfängers in m"
p.y_label = "Zeitversatz in s"
p.plot()
p.show()

```

%m Wie man erkennen kann, haben beide Fits für Plexiglas eine recht gute χ^2 -Wahrscheinlichkeit und einen akzeptablen Wert für $\frac{\chi^2}{\text{ndof}}$. Allerdings weichen die bestimmten Brechzahlen stark voneinander ab und sind im Rahmen ihrer Unsicherheiten nicht kompatibel. Für das l_{P1} lange Rohr ergab sich eine Brechzahl von $n_{\text{Plexiglas}} = [n_{P1} \pm \Delta n_{P1}]$ und für das l_{P2} lange Rohr ergab sich eine Brechzahl von $n_{\text{Plexiglas}} = [n_{P2} \pm \Delta n_{P2}]$. Besonders der zweite Wert weicht dabei stark vom Literaturwert von $n_{\text{Plexiglas}} = 1.49$ (nach <https://www.chemie.de/lexikon/Brechzahl.html> aufgerufen am 15.12.2023 um 17:36) ab. Da die Fits beide jeweils recht gute Wahrscheinlichkeiten haben, ist davon auszugehen, dass systematische Fehler begangen wurden, vermutlich die Bestimmung des Nullpunktes, die auch bei Aufgabe 2 für Probleme gesorgt hat, hier aber wirklich relevant wird, da die zu bestimmende Größe vom y-Achsenabschnitt abhängt. In Kombination damit wurden die Fehler auf diese Nullsetzung zu klein abgeschätzt.

```

#l_off_W = ufloat(1149e-3,0.0015)
d_W = 994e-3 #Strecke im Medium #TODO
a_W = ""
l,dt
1196e-3, 2721e-9
1361e-3, 3020e-9
1537e-3, 3391e-9
1719e-3, 3739e-9
1899e-3, 4093e-9
2079e-3, 4457e-9
""

```

```

mpg = pd.read_csv(StringIO(a_W))
mpg["l"] = mpg["l"] - l_off.n
%m Messung in Wasser der Länge  $l$  in m
mpg.rename(columns={"l": "Distanz in m", "dt": "Gemessener Zeitversatz"},
           inplace=True)
display(mpg)

l_W, dt_W = np.genfromtxt(StringIO(a_W), delimiter=',')[1:].transpose()
l_W = [i-l_off.n for i in l_W]

def NWModel(l, n=1, w_bar= _bar.n, w= .n, d=d_W):
    return 1/2.99792458e8*w/w_bar*(d*(n-1.000292)+l)
xy_data = kafe2.XYContainer(x_data=l_W, y_data=dt_W)
xy_data.label = 'Zeitverschiebung in Abhängigkeit von der Distanz'
n_fit = kafe2.Fit(data=xy_data, model_function=NWModel)
n_fit.add_error("x", 0.005)
n_fit.add_error("y", dt)
n_fit.add_parameter_constraint(name="w", value= .n, uncertainty= .s)
n_fit.add_parameter_constraint(name="w_bar", value= _bar.n, uncertainty= _bar.s)
n_fit.add_parameter_constraint(name="d", value=d_W, uncertainty=1e-3)
n_fit.do_fit()
n_W, _, _, _ = n_fit.parameter_values
Δn_W, _, _, _ = n_fit.parameter_errors
p = kafe2.Plot(n_fit)
p.x_label = "Distanz des Empfängers in m"
p.y_label = "Zeitversatz in s"
p.plot()
p.show()
%m Die  $\chi^2$ -Wahrscheinlichkeit des Fits ist etwas geringer als erhofft,
↳ zusammen mit  $\frac{\chi^2}{ndof}$  aber in einem akzeptablen Bereich. Der
↳ bestimmte Wert von  $n_{\text{Wasser}} = [n_W : .3] \pm [\Delta n_W : .3]$  ist gut mit dem
↳ Literaturwert von  $n_{\text{Wasser}} = 1.3330$  (vom Datenblatt nach Wikipedia)
↳ kompatibel, wobei auch hier der angegebene Fehler etwas gering erscheint, die
↳ Distanzmessung allerdings mit einem digital abzulesenden genauen Messgerät
↳ erfolgte (Unsicherheit von  $0.0015$  m) und die Zeitmessung zwar
↳ rauschbehaftet, allerdings durch ihre digitale Natur ebenfalls nicht zu
↳ unsicher (angenommene Unsicherheit von  $[\Delta t * 1e9 : .3]$  ns) war. Der Fehler
↳ des Messgeräts wurde auf  $0.005$  m erhöht, um die Unsicherheit in der Wahl
↳ des Nullpunktes zu berücksichtigen, diese Schätzung scheint allerdings zu
↳ konservativ gewesen zu sein. Andere aufgetretene Unsicherheiten waren eine
↳ von  $[.s]$  s auf die Hauptfrequenz und die zuvor bereits angesprochene
↳ auf  $\bar{\omega}$ .

```

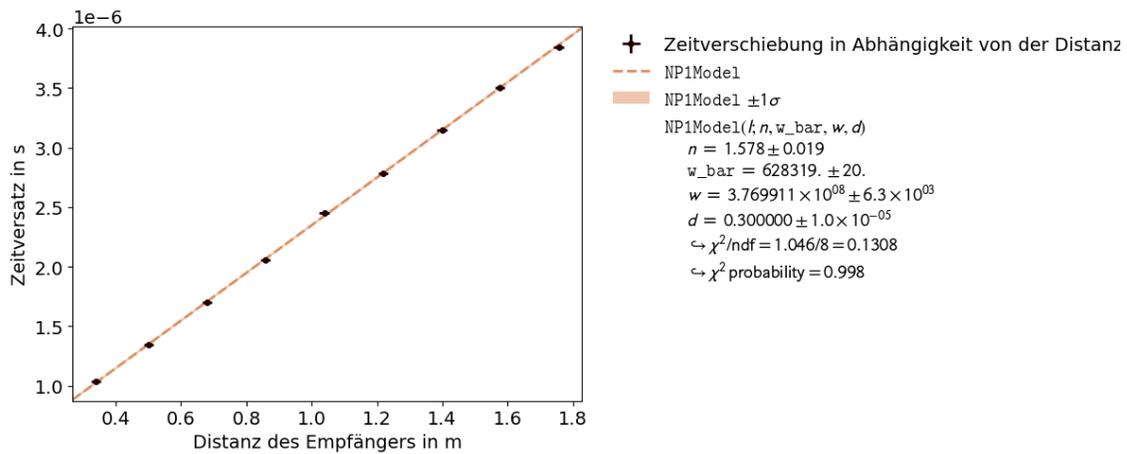
Es wird der Brechungsindex von Wasser und Plexiglas vermessen. Dazu wird der Aufbau aus 3.1 genutzt, allerdings ein Teil der Laufstrecke mit dem zu vermessenden Medium gefüllt.

Vor den Messungen wurde der Phasenversatz am Oszilloskop bei $0.200 + / - 0.005$ m auf 0 gesetzt und dieser Wert wird nun von allen vermessenen Längen abgezogen.

Die Daten werden jeweils an ein Modell der Form $\bar{\Delta t} = \frac{\omega}{\omega_c} (d * (n_{Material} - n_{Luft}) + l)$ gefittet.

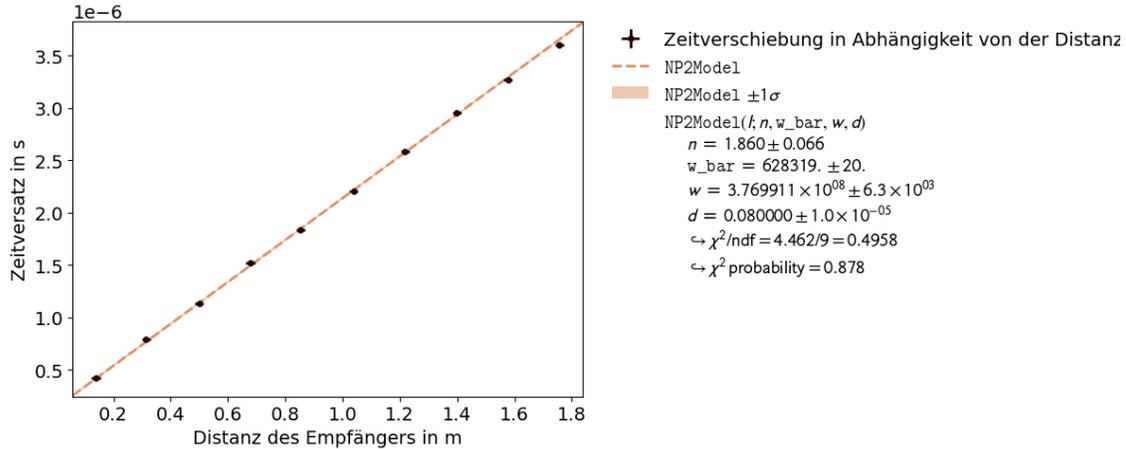
Messung in Plexiglas der Länge 0.3 m

	Distanz in m	Gemessener Zeitversatz
0	0.339	0.000001
1	0.500	0.000001
2	0.679	0.000002
3	0.858	0.000002
4	1.039	0.000002
5	1.219	0.000003
6	1.399	0.000003
7	1.576	0.000004
8	1.757	0.000004



Messung in Plexiglas der Länge 0.08 m

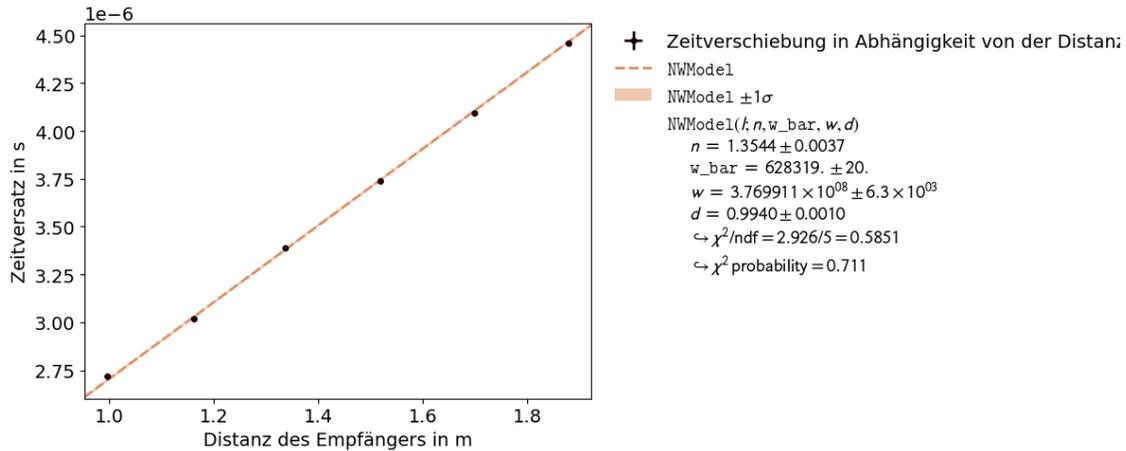
	Distanz in m	Gemessener Zeitversatz
0	0.139	4.245000e-07
1	0.314	7.946000e-07
2	0.499	1.137000e-06
3	0.678	1.518000e-06
4	0.853	1.839000e-06
5	1.038	2.210000e-06
6	1.218	2.580000e-06
7	1.399	2.957000e-06
8	1.577	3.271000e-06
9	1.757	3.603000e-06



Wie man erkennen kann, haben beide Fits für Plexiglas eine recht gute χ^2 -Wahrscheinlichkeit und einen akzeptablen Wert für $\frac{\chi^2}{ndof}$. Allerdings weichen die bestimmten Brechzahlen stark voneinander ab und sind im Rahmen ihrer Unsicherheiten nicht kompatibel. Für das 0.3 m lange Rohr ergab sich eine Brechzahl von $n_{Plexiglas} = 1.58 \pm 0.0186$ und für das 0.08 m lange Rohr ergab sich eine Brechzahl von $n_{Plexiglas} = 1.86 \pm 0.0663$. Besonders der zweite Wert weicht dabei stark vom Literaturwert von $n_{Plexiglas} = 1.49$ (nach <https://www.chemie.de/lexikon/Brechzahl.html> aufgerufen am 15.12.2023 um 17:36) ab. Da die Fits beide jeweils recht gute Wahrscheinlichkeiten haben, ist davon auszugehen, dass systematische Fehler begangen wurden, vermutlich die Bestimmung des Nullpunktes, die auch bei Aufgabe 2 für Probleme gesorgt hat, hier aber wirklich relevant wird, da die zu bestimmende Größe vom y-Achsenabschnitt abhängt. In Kombination damit wurden die Fehler auf diese Nullsetzung zu klein abgeschätzt.

Messung in Wasser der Länge 0.994 m

	Distanz in m	Gemessener Zeitversatz
0	0.996	0.000003
1	1.161	0.000003
2	1.337	0.000003
3	1.519	0.000004
4	1.699	0.000004
5	1.879	0.000004



Die χ^2 -Wahrscheinlichkeit des Fits ist etwas geringer als erhofft, zusammen mit $\frac{\chi^2}{\text{ndof}}$ aber in einem akzeptablen Bereich. Der bestimmte Wert von $n_{\text{Wasser}} = 1.35 \pm 0.00372$ ist gut mit dem Literaturwert von $n_{\text{Wasser}} = 1.3330$ (vom Datenblatt nach Wikipedia) kompatibel, wobei auch hier der angegebene Fehler etwas gering erscheint, die Distanzmessung allerdings mit einem digital abzulesenden genauen Messgerät erfolgte (Unsicherheit von 0.0015 m) und die Zeitmessung zwar rauschbehaftet, allerdings durch ihre digitale Natur ebenfalls nicht zu unsicher (angenommene Unsicherheit von 15.0 ns) war. Der Fehler des Messgeräts wurde auf 0.005 m erhöht, um die Unsicherheit in der Wahl des Nullpunktes zu berücksichtigen, diese Schätzung scheint allerdings zu konservativ gewesen zu sein. Andere aufgetretene Unsicherheiten waren eine von $6283.185307179586 \text{ s}$ auf die Hauptfrequenz und die zuvor bereits angesprochene auf \bar{w} .

3.3.2 Aufgabe 3.2: Bestimmung von n mit Hilfe eines Laserentfernungsmessers

Heute benutzt fast jeder Handwerker einen handelsüblichen Laserentfernungsmesser, um z.B. den Abstand zweier Wandflächen eines Neubaus zu vermessen. Wiederholen Sie die Messung aus Aufgabe 3.1 mit dem im Praktikum zur Verfügung stehenden Laserentfernungsmesser. Stellen Sie hierzu eine der bereitstehenden Küvetten jeweils einmal längs und einmal quer in den Strahlengang zwischen dem Entfernungsmesser und einem beliebigen Reflektor. Berechnen Sie aus den vom Gerät angezeigten Entfernungen und den Abmessungen der Küvetten den entsprechenden Brechungsindex.

Es stehen Ihnen hierzu Küvetten gefüllt mit Wasser und Silikonöl zur Verfügung. Es genügt, wenn Sie für diese Aufgabe die Messung mit *einem* Material durchführen.

```
[6]: import warnings
      warnings.filterwarnings('ignore')
```

```
%m In diesem Experiment soll der Brechungsindex von verschiedenen Medien mit
↳ Hilfe eines Laserentfernungsmessgeräts bestimmt werden. Dabei werden die
↳ Strecke im Medium d und die durchs medium gemessene verfälschte Streche l'
↳ bestimmt. Die Differenz zu den Längenmessungen mit der leeren Küvette gibt
↳ die durchs Medium erzeugte scheinbare Längenänderung. Werte für d wurden aus
↳ dem Datenblatt entnommen, da leider kein akkurates Messinstrument für die
↳ Dimensionsbestimmung vorhanden ist. Allerdings sind so die Unsicherheiten
↳ nicht bekannt, da sie im Datenblatt nicht aufgeführt sind. Deswegen wurde
↳ die Unsicherheiten auf 1mm geschätzt. Die Unsicherheit des Entfernungsmesser
↳ ist laut Hersteller 1.5 mm.
```

```
Null = 0.351
```

```
d1 = 0.1
```

```
d2 = 0.05
```

```
ohne_l1 = 0.352
```

```
ohne_l2 = 0.353
```

```
silikon_l1 = 0.369
```

```
silikon_l2 = 0.394
```

```
wasser_l1 = 0.356
```

```
wasser_l1 = 0.382
```

```
d = [d1,d2,d1,d2,d1,d2]
```

```
ls = [ohne_l1,ohne_l2,silikon_l1,silikon_l2,wasser_l1,wasser_l1]
```

```
l = [ohne_l1,ohne_l2,ohne_l1,ohne_l2,ohne_l1,ohne_l2]
```

```
uls = uarr(ls, 0.0015)
```

```
ul = uarr(l, 0.0015)
```

```
ud = uarr(d, 0.001)
```

```
un = (uls-ul)/ud + 1
```

```
mpg = pd.DataFrame()
```

```
mpg["Material"] = ["Luft", "Luft", "Silikonöl", "Silikonöl", "Wasser", "Wasser"]
```

```
mpg["d"] = nomv(ud)
```

```
mpg["Unsicherheit d"] = stdv(ud)
```

```
mpg["n"] = nomv(un)
```

```
mpg["Unsicherheit zu n"] = stdv(un)*[0,0,1,1,1,1]
```

```
mpg["n"][:2] = [None, None]
```

```
mpg["Unsicherheit zu n"][:2] = [None, None]
```

```
display(mpg)
```

```
%m Es ist zu sehen, dass die einzelnen Brechzahlen für Silikonöl nicht im
↳ Rahmen der Messunsicherheit miteinander kompatibel sind. Es kann sein, dass
↳ hier eine Reflektion gemessen wird, denn es schwierig Reflektionen in der
↳ Messung auszuschließen. Keiner der beiden Werte scheint besonders plausibel.
```

```

n_ges = np.sum(nomv(un[2:4])/stdv(un[2:4]))/np.sum(1/stdv(un[2:4]))
Δn_ges = np.sqrt(np.sum((nomv(un[2:4])-n_ges)**2/stdv(un[2:4]))/np.sum(1/
↳stdv(un[2:4])))

%m Damit ergibt sich $n = [!n_ges:.4f!] \pm [!Δn_ges:.4f!]$ für Silikonöl als
↳gewichteter Mittelwert aus der Formel $n =
↳\frac{\ell_{\text{Medium}}-\ell_{\text{Luft}}}{d}+1$.

n_ges = np.sum(nomv(un[4:])/stdv(un[4:]))/np.sum(1/stdv(un[4:]))
Δn_ges = np.sqrt(np.sum((nomv(un[4:])-n_ges)**2/stdv(un[4:]))/np.sum(1/
↳stdv(un[4:])))

%m Für Wasser ergibt sich $n = [!n_ges:.4f!] \pm [!Δn_ges:.4f!]$.

%m Der gewichtete Mittelwert für das Silikonöl ist mit Vorsicht zu genießen, da
↳er aus nicht kompatiblen Werten errechnet wurde und wenig Aussagekraft hat.
↳Deswegen ist die Unsicherheit für diesen Wert auch sehr groß. Die Messung
↳für Wasser ist im Rahmen der Unsicherheit mit dem Erwarteten Wert von 1.33
↳kompatibel.

```

In diesem Experiment soll der Brechungsindex von verschiedenen Medien mit Hilfe eines Laserentfernungsmessgeräts bestimmt werden. Dabei werden die Strecke im Medium d und die durchs medium gemessene verfälschte Strecke l' bestimmt. Die Differenz zu den Längenmessungen mit der leeren Küvette gibt die durchs Medium erzeugte scheinbare Längenänderung. Werte für d wurden aus dem Datenblatt entnommen, da leider kein akkurates Messinstrument für die Dimensionsbestimmung vorhanden ist. Allerdings sind so die Unsicherheiten nicht bekannt, da sie im Datenblatt nicht aufgeführt sind. Deswegen wurde die Unsicherheiten auf 1mm geschätzt. Die Unsicherheit des Entfernungsmesser ist laut Hersteller 1.5 mm.

	Material	d	Unsicherheit d	n	Unsicherheit zu n
0	Luft	0.10	0.001	None	None
1	Luft	0.05	0.001	None	None
2	Silikonöl	0.10	0.001	1.17	0.021281
3	Silikonöl	0.05	0.001	1.82	0.045486
4	Wasser	0.10	0.001	1.3	0.021424
5	Wasser	0.05	0.001	1.58	0.043984

Es ist zu sehen, dass die einzelnen Brechzahlen für Silikonöl nicht im Rahmen der Messunsicherheit miteinander kompatibel sind. Es kann sein, dass hier eine Reflektion gemessen wird, denn es schwierig Reflektionen in der Messung auszuschließen. Keiner der beiden Werte scheint besonders plausibel.

Damit ergibt sich $n = 1.3772 \pm 0.3029$ für Silikonöl als gewichteter Mittelwert aus der Formel $n = \frac{\ell_{\text{Medium}} - \ell_{\text{Luft}}}{d} + 1$.

Für Wasser ergibt sich $n = 1.3917 \pm 0.1314$.

Der gewichtete Mittelwert für das Silikonöl ist mit Vorsicht zu genießen, da er aus nicht kompatiblen Werten errechnet wurde und wenig Aussagekraft hat. Deswegen ist die Unsicherheit für diesen Wert

auch sehr groß. Die Messung für Wasser ist im Rahmen der Unsicherheit mit dem Erwarteten Wert von 1.33 kompatibel.

[]: