

Polarisation

July 1, 2024

1 Fakultät für Physik

1.1 Physikalisches Praktikum P2 für Studierende der Physik

Versuch P2-11 (Stand: April 2024)

[Raum F1-14](#)

2 Polarisation und Doppelbrechung

Name: Vrkic Vorname: Tin E-Mail: uyvpq@student.kit.edu

Name: Nock Vorname: Mika E-Mail: uttziqstudent.kit.edu

Gruppennummer: Mo-32

Betreuer: Kerstin Trost

Versuch durchgeführt am: 01.07.24

Beanstandungen zu Protokoll Version _____:

Testiert am: _____ Testat: _____

3 Durchführung

Die Anleitung zu diesem Versuch finden Sie [hier](#).

```
[1]: import pathlib
import pandas as pd
import numpy as np
import kafe2
import scipy as sc
import matplotlib.pyplot as plt
from uncertainties import ufloat, unumpy as unp
```

```
[2]: # erstellen einer Funktion für kafe2 Fits
def fit_funktion(xy_data, model_function, xy_error, xy_label, title,
    ↪constraint=[], add_error=True):
    xy_data = kafe2.XYContainer(xy_data[0], xy_data[1])
    xy_data.label = title
    fit = kafe2.XYFit(xy_data = xy_data, model_function = model_function)
    if add_error:
        fit.add_error(axis = 'x', err_val = xy_error[0])
        fit.add_error(axis = 'y', err_val = xy_error[1])
    for i in range(len(constraint)):
        fit.add_parameter_constraint(name = constraint[i][0], value =
    ↪constraint[i][1], uncertainty = constraint[i][2])
    fit.do_fit()
    plot = kafe2.Plot(fit)
    plot.x_label, plot.y_label = xy_label[0], xy_label[1]
```

```
return fit.parameter_values, fit.parameter_errors, plot
```

3.1 Aufgabe 1: Polarisiertes Licht aus dem Wasserglas

Hinweise zu Aufgabe 1 finden in der Datei [Hinweise-Versuchsdurchfuehrung.md](#).

- Strahlen Sie Licht durch ein mit Wasser gefülltes Glas und beobachten Sie das austretende Streulicht aus verschiedenen Richtungen durch einen Polarisationsfilter.
- Beschreiben und erklären Sie, was Sie beobachten.

Lösung:

Fügen Sie Ihre Lösung zu dieser Aufgabe hier ein. Löschen Sie hierzu diesen kursiv gestellten Text aus dem Dokument. Um Code-Fragmente und Skripte in [Python](#), sowie ggf. bildliche Darstellungen direkt ins [Jupyter notebook](#) einzubinden fügen Sie dem notebook eine Code-Zelle zu.

3.2 Aufgabe 2: Erzeugung und Untersuchung von Licht mit verschiedener Polarisation

Hinweise zu Aufgabe 2 finden in der Datei [Hinweise-Versuchsdurchfuehrung.md](#).

- Erzeugen und untersuchen Sie die Intensitätsverteilungen von verschieden polarisiertem Licht.
- Bearbeiten Sie hierzu die folgenden Aufgaben.

3.2.1 Aufgabe 2.1: Aufbau des Strahlengangs

- Bauen Sie geeignete Strahlengänge zur Erzeugung von **linear**, **elliptisch** und **zirkular** polarisiertem Licht auf.
- Bestimmen Sie die Intensitätsverteilungen des Lichts jeweils als Funktion des Winkels φ eines zweiten linearen Polarisationsfilters.

Linse bei rotem Licht (lin. pol.) vor dem zweiten Polfilter

Polfilter sind leicht Wellenlängenabhängig, deshalb wird beim roten Licht (monochrom.) der Nullpunkt besser erreicht.

```
[181]: # Messungen
phi = unp.uarray(np.arange(0, 181, 5, dtype=np.float32), [1.])
# Weißes Licht, linear polarisiert
I_weiß_lin = unp.uarray([2.09, 2.18, 2.27, 2.32, 2.37, 2.36, 2.32, 2.26, 2.17,
↪ 2.04, 1.90, 1.76, 1.60, 1.43, 1.24, 1.07, 0.90, 0.76, 0.63, 0.52, 0.44, 0.
↪ 39, 0.37, 0.39, 0.44, 0.52, 0.64, 0.77, 0.93, 1.09, 1.29, 1.48, 1.67, 1.84,
↪ 2.04, 2.2, 2.32] , [0.02]) # in Volt
# Monochromatisches Rotes Licht, linear polarisiert
```

```

I_rot_lin = unp.uarray([1.84, 1.95, 2.04, 2.08, 2.10, 2.08, 2.05, 1.98, 1.88, 1.
    ↪76, 1.63, 1.47, 1.28, 1.13, 0.96, 0.78, 0.63, 0.49, 0.37, 0.27, 0.18, 0.15,
    ↪0.13, 0.14, 0.19, 0.26, 0.35, 0.48, 0.62, 0.76, 0.92, 1.08, 1.24, 1.41, 1.
    ↪56, 1.70, 1.80] , [0.01])
# Monochromatisches Rotes Licht, elliptisch polarisiert, Dicke Glimmerplättchen:
    ↪ 60 mikrometer
I_rot_ell_60 = unp.uarray([0.26, 0.33, 0.40, 0.49, 0.57, 0.66, 0.74, 0.82, 0.
    ↪89, 0.96, 1.01, 1.05, 1.08, 1.09, 1.07, 1.05, 1.02, 0.97, 0.91, 0.84, 0.76,
    ↪0.67, 0.58, 0.49, 0.41, 0.33, 0.26, 0.20, 0.16, 0.12, 0.10, 0.09, 0.10, 0.
    ↪12, 0.16, 0.21, 0.27] , [0.01])
# Monochromatisches Rotes Licht, elliptisch polarisiert, Dicke Glimmerplättchen:
    ↪ 50 mikrometer
I_rot_ell_50 = unp.uarray([0.11, 0.13, 0.15, 0.17, 0.19, 0.21, 0.23, 0.25, 0.
    ↪26, 0.28, 0.29, 0.30, 0.30, 0.31, 0.31, 0.30, 0.29, 0.28, 0.27, 0.25, 0.23,
    ↪0.21, 0.19, 0.16, 0.15, 0.13, 0.11, 0.10, 0.09, 0.08, 0.07, 0.07, 0.07, 0.
    ↪08, 0.09, 0.10, 0.11] , [0.01])
# Monochromatisches Rotes Licht, zirkular polarisiert, Lambda/4 Plättchen ->
    ↪Zirkular
I_rot_zir = unp.uarray([0.564, 0.562, 0.562, 0.562, 0.559, 0.560, 0.560, 0.559,
    ↪0.559, 0.558, 0.556, 0.553, 0.552, 0.552, 0.552, 0.551, 0.554, 0.553, 0.554,
    ↪0.556, 0.558, 0.560, 0.561, 0.561, 0.563, 0.563, 0.560, 0.561, 0.561, 0.561,
    ↪0.561, 0.560, 0.560, 0.559, 0.558, 0.554, 0.553] , [0.002])

fig0, ax0 = plt.subplots()
ax0.errorbar(unp.nominal_values(phi), unp.nominal_values(I_weiß_lin), xerr=unp.
    ↪std_devs(phi), yerr=unp.std_devs(I_weiß_lin), fmt="g.", label="weißes Licht,
    ↪lin. pol")
#ax0.errorbar(unp.nominal_values(phi), unp.nominal_values(I_rot_lin), xerr=unp.
    ↪std_devs(phi), yerr=unp.std_devs(I_rot_lin), fmt="r.", label="rotes Licht,
    ↪lin. pol")
#ax0.errorbar(unp.nominal_values(phi), unp.nominal_values(I_rot_ell_60),
    ↪xerr=unp.std_devs(phi), yerr=unp.std_devs(I_rot_ell_60), fmt="b.",
    ↪label="rotes Licht, ell. pol. 60$, \mu m$")
#ax0.errorbar(unp.nominal_values(phi), unp.nominal_values(I_rot_ell_50),
    ↪xerr=unp.std_devs(phi), yerr=unp.std_devs(I_rot_ell_50), fmt="k.",
    ↪label="rotes Licht, ell. pol. 50$, \mu m$")
#ax0.errorbar(unp.nominal_values(phi), unp.nominal_values(I_rot_zir), xerr=unp.
    ↪std_devs(phi), yerr=unp.std_devs(I_rot_zir), fmt="y.", label="rotes Licht,
    ↪zir. pol")
ax0.grid(), ax0.legend()

# Fun with polar coordinates
fig, ax = plt.subplots(subplot_kw={"projection": "polar"})

```

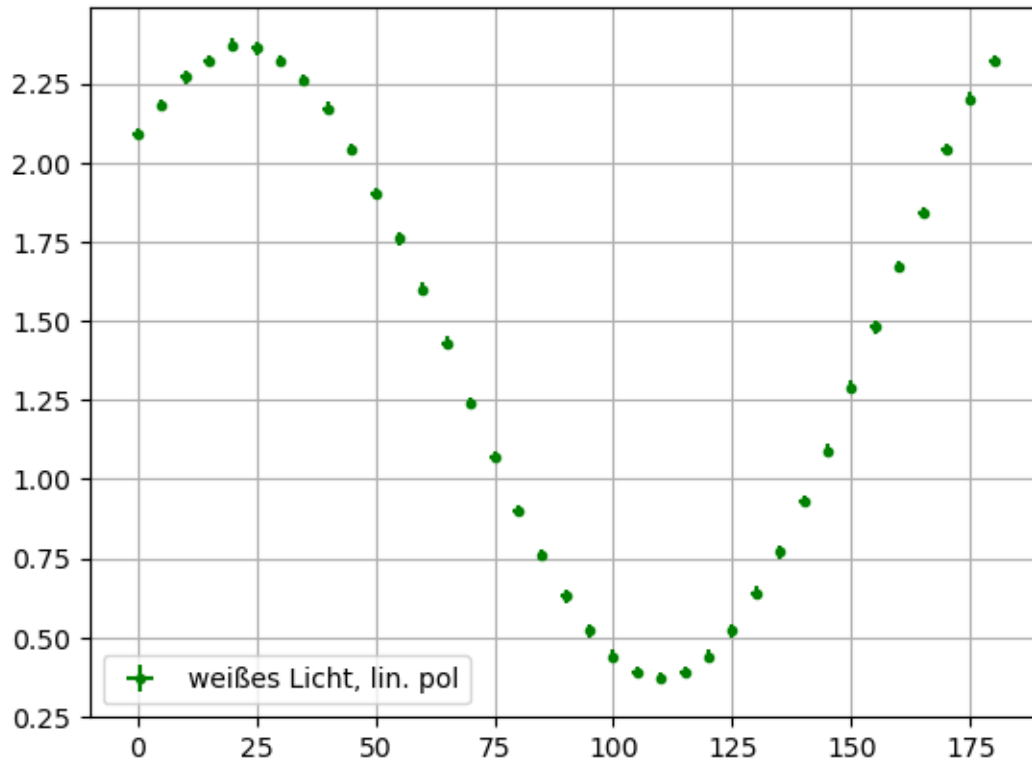
```

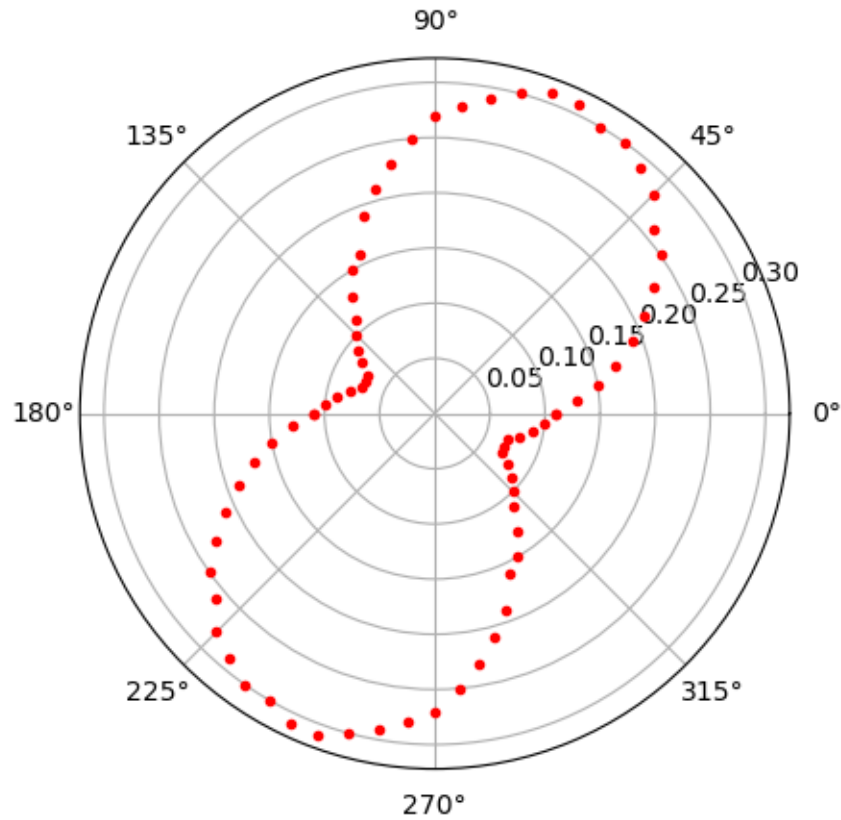
ax.plot(udp.nominal_values(phi) * np.pi/180, udp.nominal_values(I_rot_ell_50),  

↪ "r.")
ax.plot(udp.nominal_values(phi) * np.pi/180 - np.pi, udp.  

↪ nominal_values(I_rot_ell_50), "r.")
plt.show()

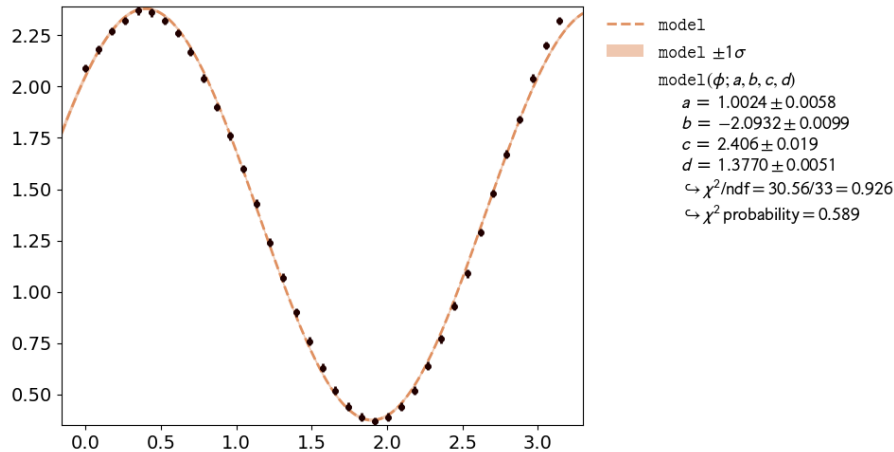
```





```
[182]: def model(phi, a=0.2, b=1, c=-np.pi/4, d=1.3):
        return a * np.sin(b*phi + c) + d
weiß_lin_data = np.array([unp.nominal_values(phi*np.pi/180), unp.
    ↳ nominal_values(I_weiß_lin)])
weiß_lin_error = np.array([unp.std_devs(phi*np.pi/180), unp.
    ↳ std_devs(I_weiß_lin)])
label = ["", ""]
title = ""

weiß_lin_res = fit_funktion(weiß_lin_data, model, weiß_lin_error, label, title)
weiß_lin_res[2].plot()
weiß_lin_res[2].show()
```



3.2.2 Aufgabe 2.2: Differenz der Brechungsindizes der beobachteten Strahlen

Bestimmen Sie die Differenz $\Delta n = (n_\beta - n_\gamma)$ der beobachteten Strahlen. Verwenden Sie hierzu die gemessenen Intensitätsverteilungen für den Fall elliptisch polarisierten Lichts.

Lösung:

Fügen Sie Ihre Lösung zu dieser Aufgabe hier ein. Löschen Sie hierzu diesen kursiv gestellten Text aus dem Dokument. Um Code-Fragmente und Skripte in [Python](#), sowie ggf. bildliche Darstellungen direkt ins [Jupyter notebook](#) einzubinden fügen Sie dem notebook eine Code-Zelle zu.

3.3 Aufgabe 3: Beobachtungen mit polarisiertem Licht

Hinweise zu Aufgabe 3 finden in der Datei [Hinweise-Versuchsdurchfuehrung.md](#).

- Beobachten Sie einige Beispiele, wo Doppelbrechung im Alltag auftaucht und z.T. auch technisch angewandt wird.
- Bearbeiten Sie hierzu die folgenden Aufgaben.

3.3.1 Aufgabe 3.1: Doppelbrechung am Klebefilm

- An handelsüblichen Klebefilmen tritt Doppelbrechung auf, die Sie mit linear polarisiertem Licht sichtbar machen können.
- Untersuchen Sie die am Versuch ausliegenden Klebefilme und beschreiben Sie, was Sie beobachten.
- Stellen Sie eigene Klebefilmkonstruktionen her, um sich mit dem beobachteten Phänomen vertraut zu machen.
- Fügen Sie Ihrem Protokoll entsprechende Aufnahmen zu.

3.3.2 Aufgabe 3.2: Doppelbrechung unter mechanischer Spannung

- Eine technische Anwendung der Doppelbrechung besteht in der Sichtbarmachung von Stellen an Materialien, die besonderer mechanischer Spannung unterliegen.
 - Am Versuch liegen einige Modelle aus. Versetzen Sie diese mechanischer Spannung und beobachten Sie die Transmission linear polarisierten Lichts.
 - Beschreiben Sie, was Sie beobachten.
 - Fügen Sie Ihrem Protokoll entsprechende Aufnahmen zu.
-

Lösung:

Fügen Sie Ihre Lösung zu dieser Aufgabe hier ein. Löschen Sie hierzu diesen kursiv gestellten Text aus dem Dokument. Um Code-Fragmente und Skripte in [Python](#), sowie ggf. bildliche Darstellungen direkt ins [Jupyter notebook](#) einzubinden fügen Sie dem notebook eine Code-Zelle zu.
