221

March 4, 2024

[1]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.stats import chi2
import scipy.constants as scp
from tabulate import tabulate
```

1 Clement-Desormes

[2]:
```
#Messwerte:
h1 = np.array([100.5, 163.0, 189.0, 108.0, 154.0])
h3 = np.array([28.0, 44.0, 51.0, 29.5, 49.5])
dh = np.full(5, 1.4)

#Kappa:
kappa = h1 /(h1 - h3)
dkappa = 1/(h1 - h3)**2 * np.sqrt((h3 * dh)**2 + (dh * h1)**2)

#Mittelwert:
kappa_L_1 = np.mean(kappa)
dkappa_L_1 = np.sqrt((np.std(kappa)/np.sqrt(5))**2 + (np.sqrt(np.sum(dkappa**2)/
 ↪5))**2)

print(kappa)
print(dkappa)
print('---')
print('kappa_L_1 =', kappa_L_1, '+/-', dkappa_L_1)
```

```
[1.3862069  1.3697479  1.36956522 1.37579618 1.47368421]
[0.02778762 0.01669147 0.0143911  0.02543536 0.02073797]
---
kappa_L_1 = 1.3950000803945914 +/- 0.027997367118954206
```

[3]:
```
#Signifikanztest:
kappa_N2 = 1.401
sign_L1 = np.abs(kappa_L_1 - kappa_N2)/dkappa_L_1
```

```
print('sigma_L1 =', sign_L1)
```

```
sigma_L1 = 0.2143029942750093
```

2 Rüchard

[4]:
```
#Messwerte:
t_L = (1/50) * np.array([48.78, 50.12, 49.06, 50.12, 49.07, 50.12]) #s
t_Ar = (1/50) * np.array([45.9, 45.01, 45.9, 45.01, 45.90, 45.78])
dt = (1/50) * 0.20 #s

m_L = (1/1000) * 26.006 #kg
dm_L = (1/1000) * 0.002
V_L = 10**(-6) * 5460 #m^3
dV_L = 10**(-6) * 5
r_L = 0.5 * (1/1000) * 15.97 #m
dr_L = 0.5 * (1/1000) * 0.02

m_Ar = (1/1000) * 26.116 #kg
dm_Ar = (1/1000) * 0.002
V_Ar = 10**(-6) * 5370 #m^3
dV_Ar = 10**(-6) * 5
r_Ar = 0.5 * (1/1000) * 15.95 #m
dr_Ar = 0.5 * (1/1000) * 0.02

p = (100) * 1004.4 #Pa
dp = (100) * 0.3

#Druck:
p_L = p + m_L * scp.g /(np.pi * r_L**2)
dp_L = np.sqrt((dp)**2 +
               (scp.g * dm_L /(np.pi * r_L**2))**2 +
               (m_L * scp.g * 2 * dr_L /(np.pi * r_L**3))**2)
p_Ar = p + m_Ar * scp.g /(np.pi * r_Ar**2)
dp_Ar = np.sqrt((dp)**2 +
               (scp.g * dm_Ar /(np.pi * r_Ar**2))**2 +
               (m_Ar * scp.g * 2 * dr_Ar /(np.pi * r_Ar**3))**2)

#Mittelwerte:
T_L = np.mean(t_L)
dT_L = np.sqrt((np.std(t_L)/np.sqrt(6))**2 + (dt)**2)
T_Ar = np.mean(t_Ar)
dT_Ar = np.sqrt((np.std(t_Ar)/np.sqrt(6))**2 + (dt)**2)

#Kappas:
kappa_L_2 = 4 * m_L * V_L/(r_L**4 * T_L**2 * p_L)
dkappa_L_2 = kappa_L_2 * np.sqrt((dm_L/m_L)**2 +
```

```
                                (dV_L/V_L)**2 +
                                (4 * dr_L/r_L)**2 +
                                (2 * dT_L/T_L)**2 +
                                (dp_L/p_L)**2)
kappa_Ar = 4 * m_Ar * V_Ar/(r_Ar**4 * T_Ar**2 * p_Ar)
dkappa_Ar = kappa_Ar * np.sqrt((dm_Ar/m_Ar)**2 +
                                (dV_Ar/V_Ar)**2 +
                                (4 * dr_Ar/r_Ar)**2 +
                                (2 * dT_Ar/T_Ar)**2 +
                                (dp_Ar/p_Ar)**2)

print('p_L =', p_L, '+/-', dp_L)
print('p_Ar =', p_Ar, '+/-', dp_Ar)

print('T_L =', T_L, '+/-', dT_L)
print('T_Ar =', T_Ar, '+/-', dT_Ar)

print('kappa_L_2 =', kappa_L_2, '+/-', dkappa_L_2)
print('kappa_Ar =', kappa_Ar, '+/-', dkappa_Ar)
```

```
p_L = 101713.1938148089 +/- 30.16917429800467
p_Ar = 101721.78763969397 +/- 30.17188656340854
T_L = 0.9908999999999999 +/- 0.006216420906670391
T_Ar = 0.9116666666666667 +/- 0.005203560034921967
kappa_L_2 = 1.3989093150572625 +/- 0.018947564866715418
kappa_Ar = 1.640333046463036 +/- 0.02051604511476058
```

[5]:
```
#Signifikanztests:
kappa_N2 = 1.401
kappa_Ar_ = 1.648
sign_L2 = np.abs(kappa_L_2 - kappa_N2)/dkappa_L_2
print('sigma_L2 =', sign_L2)
sign_Ar = np.abs(kappa_Ar - kappa_Ar_)/dkappa_Ar
print('sigma_Ar =', sign_Ar)
```

```
sigma_L2 = 0.11034056130401239
sigma_Ar = 0.37370523870840483
```