WS19/20, PAP2.1, Versuch 212: Zähigkeit von Flüssigkeiten

Praktikant:

Gerasimov, Vsevolod

zusammen mit

Reiter, Leonardo

Betreuer:

Jäschke, Cornelia

Durchführung am
7. Januar, 2020

Inhaltsverzeichnis

1 Einführung[1]

In diesem Versuch werden wir die Viskosität $\boldsymbol{\eta}$ von Polyethylenglykol nach Stokes mit einem Kugelfallviskosimeter bestimmen und zusätzlich ist die Gültigkeitsgrenze des Stokes'schen ¨ Gesetzes zu überprüfen, indem der Übergang von laminarer zu turbulenter Umströmung der Kugel (Wirbelablösung) ermittelt wird.

Danach bestimmen wir wieder die Zähigkeit $\boldsymbol{\eta}$ von Polyethylenglykol, aber jetzt nach Hagen-Poiseuille mit einem Kapillarviskosimeter.

Am Ende vergleichen wir die 2 gewonnenen Werte für $\boldsymbol{\eta}$ miteinander.

2 Versuchsaufbau[1], Literaturwerte & Vorbereitung

- Messzylinder (Innendurchmesser: $\boldsymbol{D = 75.0(5)\ mm}$) aus Hartglas mit Messskaler, gefüllt mit Polyethylenglykol. Am unteren Teil des Zylinders befindet sich eine Präzisionskapillare (Länge: $\boldsymbol{L = 100.0(5)\ mm}$) , Kapillardurchmesser: $\boldsymbol{2R = 1.50(1)\ mm}$).

- Kugeln aus „Hostaform C“ mit folgenden Durchmessern:

$$\boldsymbol{d = 2.0/3.0/4.0/5.0/6.0/7.144/8.0/9.0\ mm\ (\pm 1\%)}$$

  Die Dichte der Kugeln und die Dichte von Polyethylenglykol wurden angegeben.

- Abbildung 1 wurde uns bereitgestellt[1] und zeigt die Abhängigkeit von der Temperatur und der Dichte des Polyethylenglykols.

- Literaturwert für Erdbeschleunigung[2]in Heidelberg, Baden Württemberg, Deutschland: $\boldsymbol{g = 9.80984(2)\ m\ s^2}$

[1] Dr. J.Wagner - Physikalisches Anfängerpraktikum - V. 1.1 Stand 01/2018, Versuch 212
[2] Dr. J.Wagner - Physikalisches Anfängerpraktikum - V. 1.2 Stand 06/2016, Versuch 14

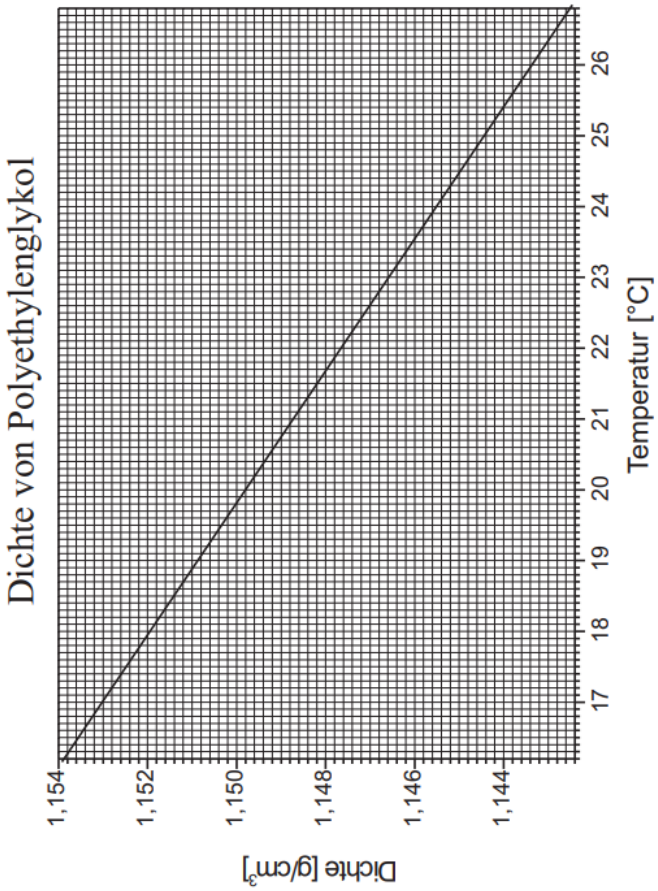

Abbildung 1

3 Durchführung

3.1 Messung der Viskosität nach Stokes

Es wird die Fallzeit $\boldsymbol{t}$ der Kugel zwischen zwei im Abstand $\boldsymbol{h}$ angebrachten Markierungen gemessen. Die Messungen werden mit fallendem Kugelradius durchgeführt und die Werte $\boldsymbol{h}$ und $\boldsymbol{t}$ notiert. Wir notieren die Temperatur der Flüssigkeit während der Messung mit den kleinsten Kugeln. Der Abstand der ersten Messmarke von der Flüssigkeitsoberfläche wird so gewählt, dass sich die Kugel beim Durchlaufen der ersten Messmarke, mit konstanter Geschwindigkeit bewegt. Wir lassen von jedem Durchmesser 5 Kugeln die Fallstrecke möglichst in der Rohrachse durchfallen. Zur Bestimmung der Fallzeit dienen die beigegebenen Handstoppuhren.

Der Innendurchmesser $\boldsymbol{D}$ des Kugelfallgefäßes ist am Viskosimeter angegeben. Dieser Wert wurde ursprünglich im Protokoll vergessen und nachträglich nachgeschaut.

Bei der Durchführung des Experiments wird unbedingt darauf geachtet, dass an den Kugeln keine Luftbläschen haften. Daher sortieren wir vor dem Einbringen der Kugeln in das Fallgefäß, zunächst einige Kugeln des jeweiligen Durchmessers in ein Becherglas und geben etwas Flüssigkeit mit hinein. Wir schwenken das Becherglas vorsichtig um, so dass die Kugeln vollständig benetzt sind und keinerlei Luftbläschen mehr daran zu erkennen sind. Mit der Pinzette werden dann die mit der Flüssigkeit benetzten Kugeln in das Fallgefäß gegeben.

3.2 Messung der Viskosität nach Hagen-Poiseuille

Unter den Ausfluss der Kapillare stellen wir ein Becherglas und öffnen den Hahn, indem wir ihn parallel zur Kapillare drehen. Wir warten nach dem Öffnen so lange ab, bis sich die Strömungsverhältnisse stabilisiert haben und eine gleichmäßige Tropfenfolge zu beobachten ist. Sobald dies der Fall ist, stellen wir einen leeren Messzylinder unter den Ausfluss, starten die Stoppuhr und notieren die Anfangshöhe $\boldsymbol{h_a}$ der Flüssigkeitssäule. Wir messen die Ausströmzeit von $\boldsymbol{25\ cm^3}$ der Flüssigkeit. Nachdem diese Mengen ausgeströmt sind, schließen Sie den Hahn und messen erneut die Höhe des Flüssigkeitsspiegels $\boldsymbol{h_e}$ und notieren die Temperatur der Flüssigkeit. Da eine Wiederholung der Messung relativ lange dauert, bei einer einmaligen Messung aber die Möglichkeit eines Irrtums besteht, nehmen wir zur Kontrolle

Zwischenwerte des Volumens auf. Wir Notieren daher bei laufender Stoppuhr die Ausflusszeit bei **5, 10, 15, 20** und **25** $\boldsymbol{cm^3}$.

4 Messergebnisse

Alle Messdaten wurden dem Versuchsprotokoll (7.Januar, 2020) entnommen und in die Tabellen 1 und 2 übertragen. Der Innendurchmesser $\boldsymbol{D}$ wurde nachträglich im Versuchsraum nachgeschaut und ist in der Beschreibung des Versuchsaufbaus oben aufgezählt.

5 Auswertung mit Python

5.1 Messung der Viskosität nach Stokes

5.1.1 Rechnung

Wir gehen davon aus, dass das die Sinkgeschwindigkeit sich linear Quadrat des Kugelradius verhält . Daher ist unser funktionales Modell für die Ausgleichungsrechnung wie folgt:

$$\frac{\boldsymbol{v_{lam}}}{\boldsymbol{\rho_k - \rho_f}} = \boldsymbol{S\, r^2} \tag{1}$$

wobei die Steigung

$$\boldsymbol{S = \frac{2}{9}\, g \frac{1}{\eta}} \tag{2}$$

ist und die gemittelte Sinkgeschwindigkeiten ergeben sich aus

$$\langle \boldsymbol{v} \rangle \;= \frac{\boldsymbol{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{v_i} \tag{3}$$

$$\begin{aligned} \boldsymbol{\Delta}\langle \boldsymbol{v} \rangle \;&= \sqrt{\mathrm{Var}\left(\langle \boldsymbol{v} \rangle\right)} = \sqrt{\mathrm{Var}\left(\frac{\boldsymbol{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{v_i}\right)} \\ &= \sqrt{\frac{\boldsymbol{1}}{\boldsymbol{n^2}} \mathrm{Var}\left(\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{v_i}\right)} = \frac{\boldsymbol{1}}{\boldsymbol{n}} \sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \left(\boldsymbol{\Delta v_i}\right)^{\boldsymbol{2}}} \end{aligned} \tag{4}$$

Wegen des endlichen Durchmessers des Fallrohres, wird die Sinkgeschwindigkeit verfälscht und systematisch zu klein gemessen, wobei der Fehler mit wachsendem Kugelradius zunimmt. Dies lässt sich durch die Ladenburg'sche Korrektur $\boldsymbol{\lambda}$ im Stokes'schen Gesetz berücksichtigen:

$$\boldsymbol{v_{lam} = \lambda \langle v \rangle} \tag{5}$$

mit

$$\boldsymbol{\lambda = 1 + 2.1 \frac{d}{D}} \tag{6}$$

Die Fehler hier sind

$$\boldsymbol{\Delta v_{lam} = v_{lam} \sqrt{\left(\frac{\Delta \lambda}{\lambda}\right)^2 + \left(\frac{\Delta \langle v \rangle}{\langle v \rangle}\right)^2}} \tag{7}$$

und

$$\begin{aligned} \boldsymbol{\Delta\lambda} &= \boldsymbol{\Delta}\left(\boldsymbol{1 + 2.1 \frac{d}{D}}\right) \\ &= \boldsymbol{\Delta}\left(\boldsymbol{2.1 \frac{d}{D}}\right) \\ &= \boldsymbol{2.1 \frac{d}{D} \sqrt{\left(\frac{\Delta d}{d}\right)^2 + \left(\frac{\Delta D}{D}\right)^2}} \end{aligned} \tag{8}$$

Zusätzlich werden wir folgende Formeln benutzen:

$$\boldsymbol{r^2 = \left(\frac{d}{2}\right)^2} \tag{9}$$

Tabelle 1: Messung der Fallzeiten

Messung Nr.	Durchmesser $d\ [mm]$	Zeit $t\ [s]$	Fallhöhe $h\ [mm]$	Dichte ρ $[g\ cm^{-3}]$	Durchmesser $d\ [mm]$	Zeit $t\ [s]$	Fallhöhe $h\ [mm]$	Dichte ρ $[g\ cm^{-3}]$	Durchmesser $d\ [mm]$.	Zeit $t\ [s]$	Fallhöhe $h\ [mm]$	Dichte ρ $[g\ cm^{-3}]$
1		**2.87 ± 0.40**				**5.87 ± 0.40**				**40.15 ± 0.40**		
2		**3.01 ± 0.40**				**5.84 ± 0.40**				**40.46 ± 0.40**		
3	**9.00 ± 0.09**	**3.12 ± 0.40**	**100 ± 3**	**1.3625 ± 0.0025**	**6.00 ± 0.06**	**5.80 ± 0.40**	**100 ± 3**	**1.3775 ± 0.0025**	**3.00 ± 0.03**	**40.40 ± 0.40**	**200 ± 3**	**1.3775 ± 0.0025**
4		**3.14 ± 0.40**				**5.86 ± 0.40**				**40.42 ± 0.40**		
5		**3.12 ± 0.40**				**6.01 ± 0.40**				**40.65 ± 0.40**		
1		**3.86 ± 0.40**				**7.78 ± 0.40**				**19.20 ± 0.40**	**50 ± 3**	
2		**4.48 ± 0.40**				**7.78 ± 0.40**				**20.07 ± 0.40**		
3	**8.00 ± 0.08**	**3.81 ± 0.40**	**100 ± 3**	**1.3575 ± 0.0025**	**5.00 ± 0.05**	**7.87 ± 0.40**	**100 ± 3**	**1.3775 ± 0.0025**	**2.00 ± 0.02**	**38.32 ± 0.40**		**1.3775 ± 0.0025**
4		**3.96 ± 0.40**				**7.92 ± 0.40**				**40.50 ± 0.40**	**100 ± 3**	
5		**3.95 ± 0.40**				**8.07 ± 0.40**				**39.68 ± 0.40**		
1		**18.00 ± 0.40**	**400 ± 3**			**11.58 ± 0.40**	**100 ± 3**			**91.84 ± 0.40**		
2		**12.81 ± 0.40**				**11.12 ± 0.40**				**95.61 ± 0.40**		
3	**7.144 ± 0.071**	**13.15 ± 0.40**	**300 ± 3**	**1.3775 ± 0.0025**	**4.00 ± 0.04**	**6.09 ± 0.40**	**50 ± 3**	**1.3775 ± 0.0025**	**1.500 ± 0.015**	**87.18 ± 0.40**	**150 ± 3**	**1.3925 ± 0.0025**
4		**13.23 ± 0.40**				**5.75 ± 0.40**				**99.28 ± 0.40**		
5		**12.89 ± 0.40**				**5.87 ± 0.40**			/			

[1] Δd angegeben als **1%** von d
[2] Δh grob abgeschätzt als Hälfte der Skaleneinteilung von $5\ mm$
[3] Δt grob abgeschätzt als menschliche Reaktionszeit von 300 ms und Faktor $\sqrt{2}$, weil zum Messen die Stoppuhr doppelt betätigt werden muss: $0.3\ s \times \sqrt{2} \approx 0.4\ s = \Delta t$
[4] Temperatur der Flüssigkeit während der Versuchsdurchführung: $T_1 = 21.0(5)\ °C$

$$\Delta r^2 = 2\frac{\Delta d}{d} \tag{10}$$

$$\Delta\left(\frac{v_{lam}}{\rho_k - \rho_f}\right) = \frac{v_{lam}}{\rho_k - \rho_f}\sqrt{\left(\frac{\Delta v_{lam}}{v_{lam}}\right)^2 + \left(\frac{\Delta\left(\rho_k - \rho_f\right)}{\rho_k - \rho_f}\right)^2}$$

$$= \frac{v_{lam}}{\rho_k - \rho_f}\sqrt{\left(\frac{\Delta v_{lam}}{v_{lam}}\right)^2 + \frac{\left(\Delta\rho_k\right)^2 + \left(\Delta\rho_f\right)^2}{\left(\rho_k - \rho_f\right)^2}} \tag{11}$$

$$\Delta\left(\frac{\langle v\rangle}{\rho_k - \rho_f}\right) = \frac{\langle v\rangle}{\rho_k - \rho_f}\sqrt{\left(\frac{\Delta\langle v\rangle}{\langle v\rangle}\right)^2 + \left(\frac{\Delta\left(\rho_k - \rho_f\right)}{\rho_k - \rho_f}\right)^2}$$

$$= \frac{\langle v\rangle}{\rho_k - \rho_f}\sqrt{\left(\frac{\Delta\langle v\rangle}{\langle v\rangle}\right)^2 + \frac{\left(\Delta\rho_k\right)^2 + \left(\Delta\rho_f\right)^2}{\left(\rho_k - \rho_f\right)^2}} \tag{12}$$

Am Ende berechnen wir aus der angepassten Steigung S die Viskosität η vom Polyethylenglykol, die theoretische Sinkgeschwindigkeit v_{theo} bei laminarer Strömung in und einer unendlich ausgedehnten Flüssigkeit und die Reynoldszahl Re für jeden der Kugeldurchmesser d:

$$\eta = \frac{2\,g}{9\,S} \tag{13}$$

$$\Delta\eta = \eta\sqrt{\left(\frac{\Delta g}{g}\right)^2 + \left(\frac{\Delta S}{S}\right)^2} \tag{14}$$

$$v_{theo} = S\,r^2\left(\rho_k - \rho_f\right) \tag{15}$$

$$\Delta v_{theo} = v_{theo}\sqrt{\left(\frac{\Delta\left(S\,r^2\right)}{S\,r^2}\right)^2 + \frac{\left(\Delta\rho_k\right)^2 + \left(\Delta\rho_f\right)^2}{\left(\rho_k - \rho_f\right)^2}} \tag{16}$$

$$Re = \frac{\rho_f\langle v\rangle d}{\eta} \tag{17}$$

$$\Delta Re = Re\sqrt{\left(\frac{\Delta\rho_f}{\rho_f}\right)^2 + \left(\frac{\Delta\langle v\rangle}{\langle v\rangle}\right)^2 + \left(\frac{\Delta d}{d}\right)^2 + \left(\frac{\Delta\eta}{\eta}\right)^2} \tag{18}$$

5.1.2 Source Code & Input

So sieht unsere Python-Implementierung aus:

Header:

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import chi2
from decimal import Decimal

def format_e(n):
    a = '%e' % Decimal(n)
    return a.split('e')[0].rstrip('0').rstrip('.')+'e'+a.split('e')[1]
```

Messwerte aus Tabelle 1 in SI Einheiten:

```
d = np.array([9,
              8,
              7.144,
              6,
              5,
              4,
              3,
              2,
              1.5]) *1e-3
Fehler_d = d*0.01

t = np.array([2.87,3.01,3.12,3.14,3.12,
              3.86,4.48,3.81,3.96,3.95,
```

```
                18.00,12.81,13.15,13.23,12.89,
                5.87,5.84,5.80,5.86,6.01,
                7.78,7.78,7.87,7.92,8.07,
                11.58,11.12,6.09,5.75,5.87,
                40.15,40.46,40.40,40.42,40.65,
                19.20,20.07,38.32,40.50,39.68,
                91.84,95.61,87.18,99.28])
Fehler_t = np.full(t.size,0.4)

h = np.array([100,100,100,100,100,
                100,100,100,100,100,
                400,300,300,300,300,
                100,100,100,100,100,
                100,100,100,100,100,
                100,100,50,50,50,
                200,200,200,200,200,
                50,50,100,100,100,
                150,150,150,150]) *1e-3
Fehler_h = np.full(h.size,3) *1e-3

rho = np.array([1.3625,
                  1.3575,
                  1.3775,
                  1.3775,
                  1.3775,
                  1.3775,
                  1.3775,
                  1.3775,
                  1.3925]) *1e3
Fehler_rho = np.full(rho.size,0.0025) *1e3

D = 75 *1e-3
Fehler_D = 0.5*1e-3

g = 9.80984
Fehler_g = 0.00002
```

Werte für $\boldsymbol{\rho_f}$ aus den Temperaturwerten (Tab.1) und Abbildung 1 entnommen:

```
rho_PEG1 = 1.1487 *1e3
Fehler_rho_PEG1 = 0.0054 *1e3
```

Berechnung von $\langle \boldsymbol{v} \rangle$ und $\boldsymbol{\Delta} \langle \boldsymbol{v} \rangle$ nach (3) bzw. (4):

```
v_mean = np.zeros(9)
v_mean[0] = np.mean((h/t)[0:5])
v_mean[1] = np.mean((h/t)[5:10])
v_mean[2] = np.mean((h/t)[10:15])
v_mean[3] = np.mean((h/t)[15:20])
v_mean[4] = np.mean((h/t)[20:25])
v_mean[5] = np.mean((h/t)[25:30])
v_mean[6] = np.mean((h/t)[30:35])
v_mean[7] = np.mean((h/t)[35:40])
v_mean[8] = np.mean((h/t)[40:44])
Fehler_v_mean = np.zeros(9)
Fehler_v_mean[0] = np.sqrt(np.mean(((h/t)**2*((Fehler_h/h)**2+(Fehler_t/t)**2))[0:5]/5))
Fehler_v_mean[1] = np.sqrt(np.mean(((h/t)**2*((Fehler_h/h)**2+(Fehler_t/t)**2))[5:10]/5))
Fehler_v_mean[2] = np.sqrt(np.mean(((h/t)**2*((Fehler_h/h)**2+(Fehler_t/t)**2))[10:15]/5))
Fehler_v_mean[3] = np.sqrt(np.mean(((h/t)**2*((Fehler_h/h)**2+(Fehler_t/t)**2))[15:20]/5))
Fehler_v_mean[4] = np.sqrt(np.mean(((h/t)**2*((Fehler_h/h)**2+(Fehler_t/t)**2))[20:25]/5))
Fehler_v_mean[5] = np.sqrt(np.mean(((h/t)**2*((Fehler_h/h)**2+(Fehler_t/t)**2))[25:30]/5))
Fehler_v_mean[6] = np.sqrt(np.mean(((h/t)**2*((Fehler_h/h)**2+(Fehler_t/t)**2))[30:35]/5))
Fehler_v_mean[7] = np.sqrt(np.mean(((h/t)**2*((Fehler_h/h)**2+(Fehler_t/t)**2))[35:40]/5))
Fehler_v_mean[8] = np.sqrt(np.mean(((h/t)**2*((Fehler_h/h)**2+(Fehler_t/t)**2))[40:44]/4))
```

Berechnung von $\boldsymbol{\lambda}$ und $\boldsymbol{\Delta\lambda}$ nach (6) bzw. (8):

```
k = (1+2.1*(d/D))
Fehler_k = 2.1*(d/D)*np.sqrt((Fehler_d/d)**2+(Fehler_D/D)**2)
```

Berechnung von $\boldsymbol{v_{lam}}$ und $\boldsymbol{\Delta v_{lam}}$ nach (5) bzw. (7):

```
v_lam = k*v_mean
Fehler_v_lam = v_lam*np.sqrt((Fehler_v_mean/v_mean)**2+(Fehler_k/k)**2)
```

Hier wurde die Abgrenzung des linearen Bereichs in Abbildung 4 gesetzt. Er trennt den Bereich der turbulenten Strömung von dem der laminaren. Der Fit wird deswegen in Abbildung 3 auch nur in diesem Bereich vorgenommen, weil die von uns verwendeten physikalischen Gesetzte und Korrekturen sich mit laminarer Strömung befassen.

```
breakpoint = 5
```

Berechnung von $\boldsymbol{r^2}$ und $\boldsymbol{\Delta r^2}$ nach (9) bzw. (10):

```
r_squared = (d/2)**2
Fehler_r_squared = r_squared*2*(Fehler_d/d)
```

Berechnung von $\frac{v_{lam}}{\rho_k - \rho_f}$ und $\boldsymbol{\Delta}\frac{v_{lam}}{\rho_k - \rho_f}$ nach (11):

```
y1 = v_lam/(rho-rho_PEG1)
Fehler_y1 = y1*np.sqrt((Fehler_v_lam/v_lam)**2+(Fehler_rho**2+Fehler_rho_PEG1**2)/((rho-rho_PEG1)
    **2))
```

Berechnung von $\frac{\langle v \rangle}{\rho_k - \rho_f}$ und $\boldsymbol{\Delta}\frac{\langle v \rangle}{\rho_k - \rho_f}$ nach (12):

```
y2 = v_mean/(rho-rho_PEG1)
Fehler_y2 = y2*np.sqrt((Fehler_v_mean/v_mean)**2+(Fehler_rho**2+Fehler_rho_PEG1**2)/((rho-rho_PEG1)
    **2))
```

Fitfunktion (1) wird deklariert:

```
from scipy import odr

def fit_func(p, x):
    (s) = p
    return s*x

model = odr.Model(fit_func)
```

darzustellende Daten werden übergeben:

```
x = r_squared[5:r_squared.size]
y = y1[5:r_squared.size]
delta_x = Fehler_r_squared[5:r_squared.size]
delta_y = Fehler_y1[5:r_squared.size]
```

Startparameter für Ausgleichungsrechnung werden gesetzt, sodass Lösung konvergiert:

```
para0 = [1]

data = odr.RealData(x, y, sx=delta_x, sy=delta_y)
odr = odr.ODR(data, model, beta0=para0 )
out = odr.run()
```

Endgültige Ausgleichungsparameter und ihre Kovarianzmatrix werden ausgelesen:

```
popt = out.beta
perr = out.sd_beta
```

Angabe welche Sigma-Umgebung der Fitfunktion im Diagramm dargestellt werden soll:

```
nstd = 1

popt_top = popt+nstd*perr
popt_bot = popt-nstd*perr
```

Plot-Umgebung wird angegeben:

```
x_fit0 = np.linspace(min(x)-(max(x)-min(x))/10, max(x)+(max(x)-min(x))/10, 1000)
x_fit = np.linspace(min(r_squared)-(max(r_squared)-min(r_squared))/10, max(r_squared)+(max(r_squared
    )-min(r_squared))/10, 1000)
fit = fit_func(popt, x_fit)
fit_top = fit_func(popt_top, x_fit)
fit_bot = fit_func(popt_bot, x_fit)
```

Diagramm (Abb.3) wird erstellt:

```
fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
plt.errorbar(r_squared, y1, yerr=Fehler_y1, xerr=Fehler_r_squared, lw=1, ecolor='k', fmt='none',
    capsize=2, label='$v_{lam}$ (mit Ladenburgscher Korrektur)')
plt.errorbar(r_squared, y2, yerr=Fehler_y2, xerr=Fehler_r_squared, lw=1, ecolor='r', fmt='none',
    capsize=2, label='$v_{mean}$ (ohne Korrektur)')
plt.plot(x_fit, fit, 'r', lw=1, label='Fit')
plt.plot(x_fit0, np.zeros(x_fit0.size), 'g:', lw=5, label='linearer Bereich, wo der Fit durchgeführt
     wurde')
```

```
ax.fill_between(x_fit, fit_top, fit_bot, color='C3', alpha=.25, label=str(nstd)+r'$\sigma$'+'-Umgebung')
plt.title('Messung nach Stokes')
plt.grid(True)
plt.xlabel('$r^2$ [$m^2$]')
plt.ylabel('$v/({\\rho}-{\\rho}_{PEG})$ [$m^{4}$ $s^{-1}$ $kg^{-1}$]')
plt.legend(loc='best')
plt.savefig('figures/212_Fig1.pdf', format='pdf', bbox_inches='tight')
```

Der Chi-Quadrat-Test wird durchgeführt unter Berücksichtigung von $\boldsymbol{\Delta r^2}$ und $\boldsymbol{\Delta \frac{v_{lam}}{\rho_k - \rho_f}}$. D.h. es wird jeweils der senkrechte/orthogonale Abstand der Messwerte zur Fitfunktion (Abb.2) berechnet und normiert[3]. Die Summe der normierten Abstandsquadrate, der χ^2-Wert, wird reduziert.

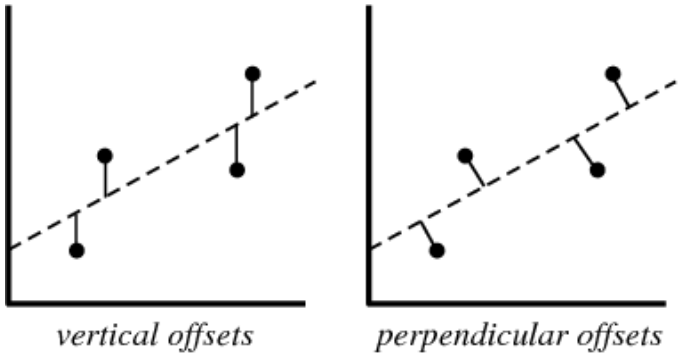

Abbildung 2: Abstände von Messdaten zur Fitfunktion[4]

```
dof = x.size-popt.size
chisquare = out.sum_square
chisquare_red = chisquare/dof
prob = round(1-chi2.cdf(chisquare,dof),2)*100
```

Auslesen der Messergebnisse:

```
S = popt[0]
Fehler_S = perr[0]
```

Berechnung von $\boldsymbol{\eta}$ und $\boldsymbol{\Delta\eta}$ nach (13) bzw. (14):

```
eta1 = 2*g/(9*S)
Fehler_eta1 = eta1*np.sqrt((Fehler_g/g)**2+(Fehler_S/S)**2)
```

Berechnung von $\boldsymbol{v_{theo}}$ und $\boldsymbol{\Delta v_{theo}}$ nach (15) bzw. (16):

```
v_theo = (fit_func(popt, r_squared))*(rho-rho_PEG1)
Fehler_v_theo = v_theo*np.sqrt(((fit_func(popt+perr, r_squared)-fit_func(popt-perr, r_squared))/2/fit_func(popt, r_squared))**2+(Fehler_rho**2+Fehler_rho_PEG1**2)/((rho-rho_PEG1)**2))
```

Berechnung von $\boldsymbol{Re}$ und $\boldsymbol{\Delta Re}$ nach (17) bzw. (18):

```
Re = rho_PEG1*v_mean*d/eta1
Fehler_Re = Re*np.sqrt((Fehler_v_mean/v_mean)**2+(Fehler_rho_PEG1/rho_PEG1)**2+(Fehler_d/d)**2+(Fehler_eta1/eta1)**2)
```

Ausgabe der Messergebnisse wird erstellt:

```
print('Steigung [m^2 / (kg s)]: ')
print('S =', format_e(S), ' +- ', format_e(Fehler_S))
print('Chi-Quadrat =', chisquare)
print('Freiheitsgrade =', dof)
print('Chi-Quadrat reduziert =', chisquare_red)
print('Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten =', prob, '%')
print('\n')
print('Viskosität: ')
print('eta_1 [Pa*s] =',  format_e(eta1), ' +- ',  format_e(Fehler_eta1))
print('\n')
```

[3] P. T. Boggs and J. E. Rogers, "Orthogonal Distance Regression," in "Statistical analysis of measurement error models and applications: proceedings of the AMS-IMS-SIAM joint summer research conference held June 10-16, 1989," Contemporary Mathematics, vol. 112, pg. 186, 1990.

[4] http://mathworld.wolfram.com/LeastSquaresFitting.html

```
print('Tabelle:')
i = 0
while i < 9:
    print('d [m] =',  format_e(d[i]), ' +- ',  format_e(Fehler_d[i]))
    print('v_mean [m / s] =',  format_e(v_mean[i]), ' +- ',  format_e(Fehler_v_mean[i]))
    print('v_theo [m / s] =',  format_e(v_theo[i]), ' +- ',  format_e(Fehler_v_theo[i]))
    print('Abweichung =', round((v_mean[i]-v_theo[i])/np.sqrt(Fehler_v_mean[i]**2+Fehler_v_theo[i
        ]**2),2), 'sigma')
    print('Re =',  format_e(Re[i]), ' +- ',  format_e(Fehler_Re[i]))
    print('\n')
    i = i+1
```

```
k2 = 1/k
Fehler_k2 = k2*Fehler_k/k

x_fit1 = np.linspace(Re[0]*1.1, Re[breakpoint+1], 1000)
x_fit2 = np.linspace(Re[breakpoint-3], Re[Re.size-1]/1.1, 1000)

line1 = k2[0]+np.log(x_fit1/Re[0])*(k2[breakpoint-1]-k2[0])/np.log(Re[breakpoint-1]/Re[0])
line2 = k2[breakpoint]+np.log(x_fit2/Re[breakpoint])*(k2[k2.size-1]-k2[breakpoint])/np.log(Re[Re.
    size-1]/Re[breakpoint])
```

Diagramm (Abb.4) wird erstellt:

```
fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
plt.errorbar(Re, k2, yerr=Fehler_k2, xerr=Fehler_Re, lw=1, ecolor='k', fmt='none', capsize=2, label
    ='Messdaten')
plt.plot(x_fit1, line1, 'r', lw=1, label='1. Gerade')
plt.plot(x_fit2, line2, 'g', lw=1, label='2. Gerade')
plt.title('Bestimmung der kritischen Reynoldszahl ${Re}_{kr}$')
plt.xscale('log')
plt.grid(which='major')
plt.grid(which='minor', linestyle=':')
plt.xlabel('Re')
plt.ylabel('$v/v_{lam}$')
plt.legend(loc='best')
plt.savefig('figures/212_Fig2.pdf', format='pdf', bbox_inches='tight')
```

5.1.3 Output

```
Steigung [m^2 / (kg s)]:
S = 1.1227e+01  +-  4.647375e-01
Chi-Quadrat = 14.872634095929296
Freiheitsgrade = 3
Chi-Quadrat reduziert = 4.957544698643098
Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten = 0.0 %

Viskosität:
eta_1 [Pa*s] = 1.941716e-01  +-  8.037662e-03


Tabelle:
d [m] = 9e-03  +-  9e-05
v_mean [m / s] = 3.28031e-02  +-  1.981428e-03
v_theo [m / s] = 4.860673e-02  +-  2.424582e-03
Abweichung = -5.05 sigma
Re = 1.746539e+00  +-  1.293409e-01


d [m] = 8e-03  +-  8e-05
v_mean [m / s] = 2.500877e-02  +-  1.177701e-03
v_theo [m / s] = 3.750716e-02  +-  1.88498e-03
Abweichung = -5.62 sigma
Re = 1.183595e+00  +-  7.535353e-02


d [m] = 7.144e-03  +-  7.144e-05
v_mean [m / s] = 2.288094e-02  +-  3.15728e-04
v_theo [m / s] = 3.277499e-02  +-  1.602269e-03
Abweichung = -6.06 sigma
Re = 9.670218e-01  +-  4.352693e-02


```

```
d [m] = 6e-03  +-  6e-05
v_mean [m / s] = 1.702084e-02  +-  5.66537e-04
v_theo [m / s] = 2.311864e-02  +-  1.130199e-03
Abweichung = -4.82 sigma
Re = 6.041618e-01  +-  3.277821e-02


d [m] = 5e-03  +-  5e-05
v_mean [m / s] = 1.268625e-02  +-  3.345911e-04
v_theo [m / s] = 1.605461e-02  +-  7.848605e-04
Abweichung = -3.95 sigma
Re = 3.752531e-01  +-  1.887944e-02


d [m] = 4e-03  +-  4e-05
v_mean [m / s] = 8.610421e-03  +-  2.900494e-04
v_theo [m / s] = 1.027495e-02  +-  5.023107e-04
Abweichung = -2.87 sigma
Re = 2.037536e-01  +-  1.110476e-02


d [m] = 3e-03  +-  3e-05
v_mean [m / s] = 4.948613e-03  +-  3.977206e-05
v_theo [m / s] = 5.779659e-03  +-  2.825498e-04
Abweichung = -2.91 sigma
Re = 8.782651e-02  +-  3.828479e-03


d [m] = 2e-03  +-  2e-05
v_mean [m / s] = 2.53887e-03  +-  5.347162e-05
v_theo [m / s] = 2.568737e-03  +-  1.255777e-04
Abweichung = -0.22 sigma
Re = 3.00394e-02  +-  1.434106e-03


d [m] = 1.5e-03  +-  1.5e-05
v_mean [m / s] = 1.608401e-03  +-  1.647327e-05
v_theo [m / s] = 1.539643e-03  +-  7.398712e-05
Abweichung = 0.91 sigma
Re = 1.427271e-02  +-  6.287311e-04
```

Wir erfahren also sofort, dass

$$\boldsymbol{S = 1.123(46) \times 10^{1}\ m^{2}\ kg^{-1}\ s^{-1}}$$
$$\boldsymbol{\eta_1 = 1.942(80) \times 10^{-1}\ Pa\ s}$$

und als Ergebnis auf unseren Anpassungstest:

$$\boldsymbol{\chi^2_{red} = 5.0}$$

Die Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten ist

$$\boldsymbol{P \approx 0.0\%}$$

und wir erhalten die Diagramme in Abb.3 und 4 und die Ergebnisse in Tabelle 3.

Abbildung 4 können wir entnehmen, dass in unserem Versuch

$$\boldsymbol{Re_{kr} \approx 0.30(5)}$$

gelten muss, da in diesem Bereich die laminare Strömung zu Turbulenzen übergeht.

5.2 Messung der Viskosität nach Hagen-Poiseuille

5.2.1 Rechnung

Wir gehen davon aus, dass das die Viskosität konstant ist. Daher ist unser funktionales Modell für die Ausgleichungsrechnung wie folgt:

$$\boldsymbol{\eta = konst.} \tag{19}$$

Wir wissen, dass

$$\boldsymbol{\frac{dV}{dt} = \frac{\pi \rho_f h g R^4}{8\, L\, \eta}} \tag{20}$$

Tabelle 2: Messung des Volumenstroms

Volumenänderung ΔV $[cm^3]$	Zeit t $[s]$
5.0 ± 0.2	**42.42 ± 0.30**
10.0 ± 0.2	**286.19 ± 0.30**
15.0 ± 0.2	**426.84 ± 0.30**
20.0 ± 0.2	**576.70 ± 0.30**
25.0 ± 0.2	**721.65 ± 0.30**

[1] Anfangshöhe: $h_a = 499(3)\ mm$
[2] Endhöhe: $h_e = 495(3)\ mm$
[3] Δt grob abgeschätzt als menschliche Reaktionszeit von 300 ms
[4] Temperatur der Flüssigkeit während der Versuchsdurchführung: $T_2 = 22.0(5)\ °C$

Tabelle 3: Ergebnisse der Python-Auswertung zum Versuch nach Stokes

Kugeldurchmesser	Sinkgeschwindigkeit			Reynoldszahl	Strömungsart
d $[mm]$	$\langle v \rangle$ $[m\ s^{-1}]$	v_{theo} $[m\ s^{-1}]$	Abweichung		
9.00 ± 0.09	$\mathbf{3.28 \times 10^{-2} \pm 0.20 \times 10^{-2}}$	$\mathbf{4.86 \times 10^{-2} \pm 0.24 \times 10^{-2}}$	$-5.1\,\sigma$	**1.75 ± 0.13**	turbulent
8.00 ± 0.08	$\mathbf{2.50 \times 10^{-2} \pm 0.12 \times 10^{-2}}$	$\mathbf{3.75 \times 10^{-2} \pm 0.19 \times 10^{-2}}$	$-5.6\,\sigma$	**1.184 ± 0.075**	turbulent
7.114 ± 0.071	$\mathbf{2.288 \times 10^{-2} \pm 0.032 \times 10^{-2}}$	$\mathbf{3.28 \times 10^{-2} \pm 0.16 \times 10^{-2}}$	$-6.1\,\sigma$	**0.967 ± 0.044**	turbulent
6.00 ± 0.06	$\mathbf{1.702 \times 10^{-2} \pm 0.057 \times 10^{-2}}$	$\mathbf{2.31 \times 10^{-2} \pm 0.11 \times 10^{-2}}$	$-4.8\,\sigma$	**0.604 ± 0.033**	turbulent
5.00 ± 0.05	$\mathbf{1.269 \times 10^{-2} \pm 0.033 \times 10^{-2}}$	$\mathbf{1.605 \times 10^{-2} \pm 0.078 \times 10^{-2}}$	$-4.0\,\sigma$	**0.375 ± 0.019**	turbulent
4.00 ± 0.04	$\mathbf{8.61 \times 10^{-3} \pm 0.29 \times 10^{-3}}$	$\mathbf{10.27 \times 10^{-3} \pm 0.50 \times 10^{-3}}$	$-2.9\,\sigma$	**0.204 ± 0.011**	laminar
3.00 ± 0.03	$\mathbf{4.949 \times 10^{-3} \pm 0.040 \times 10^{-3}}$	$\mathbf{5.78 \times 10^{-3} \pm 0.28 \times 10^{-3}}$	$-2.9\,\sigma$	**0.0878 ± 0.0038**	laminar
2.00 ± 0.02	$\mathbf{2.539 \times 10^{-3} \pm 0.053 \times 10^{-3}}$	$\mathbf{2.57 \times 10^{-3} \pm 0.13 \times 10^{-3}}$	$-0.2\,\sigma$	**0.0300 ± 0.0014**	laminar
1.500 ± 0.015	$\mathbf{1.608 \times 10^{-3} \pm 0.016 \times 10^{-3}}$	$\mathbf{1.540 \times 10^{-3} \pm 0.074 \times 10^{-3}}$	$+0.9\,\sigma$	**0.01427 ± 0.00063**	laminar

[1] Nur der Bereich der laminaren Strömung wurde zum Anpassen der Funktion benutzt. Zudem ist $\langle v \rangle$ der noch nicht der korrigierte Wert, im Gegensatz zu v_{lam}.

gilt, also

$$\eta = \frac{\pi \rho_f h g R^4 t}{8\, L\, (\Delta V)\, \ln\left(\frac{h_a}{h}\right)} \tag{21}$$

Weil $h_e \approx h_a$ bzw. $\ln\left(\frac{h_a}{h}\right) \approx 1$ nähern wir für jedes Paar unserer Messwerte aus Tabelle 2 und berechnen die Viskosität nach

$$\eta = \frac{\pi \rho_f \langle h \rangle g R^4 t}{8\, L\, (\Delta V)} \tag{22}$$

$$\Delta\eta = \eta \sqrt{\left(\frac{\Delta\rho_f}{\rho_f}\right)^2 + \left(\frac{\Delta\langle h\rangle}{\langle h\rangle}\right)^2 + \left(\frac{\Delta g}{g}\right)^2 + 4\left(\frac{\Delta R}{R}\right)^2 + \left(\frac{\Delta t}{t}\right)^2 + \left(\frac{\Delta L}{L}\right)^2 + \left(\frac{\Delta(\Delta V)}{(\Delta V)}\right)^2} \tag{23}$$

mit

$$\langle h \rangle \approx h_a + (h_e - h_a)\, q \quad \text{mit} \quad q = \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, \frac{5}{5} \tag{24}$$

Der Faktor q folgt aus den Verhältnissen Volumenänderungen zu denen wir gemessen haben.

$$\Delta\langle h \rangle \approx \sqrt{(\Delta h_a\, (1-q))^2 + (\Delta h_e\, q)^2} \tag{25}$$

Am Ende berechnen wir aus der Viskosität η vom Polyethylenglykol die Reynoldszahl Re für die Kapillare:

$$Re = \frac{\rho_f v\, (2R)}{(\pi R^2)\, \eta} = \frac{2\rho_f \Delta V}{\pi R \eta t} \tag{26}$$

$$\Delta Re = Re \sqrt{\left(\frac{\Delta\rho_f}{\rho_f}\right)^2 + \left(\frac{\Delta(\Delta V)}{\Delta V}\right)^2 + \left(\frac{\Delta R}{R}\right)^2 + \left(\frac{\Delta\eta}{\eta}\right)^2 + \left(\frac{\Delta t}{t}\right)^2} \tag{27}$$

5.2.2 Source Code & Input

So sieht die Fortführung unserer Python-Implementierung aus:

Messwerte aus Tabelle 2 in SI Einheiten:

```
V = np.array([5,10,15,20,25]) *1e-6
Fehler_V = np.full(V.size,0.2) *1e-6

t2 = np.array([142.42,286.19,426.84,576.70,721.65])
Fehler_t2 = np.full(t2.size,0.3)

R = 1.5 *1e-3 /2
Fehler_R = 0.01 *1e-3 /2

L = 100 *1e-3
Fehler_L = 0.5 *1e-3

h1 = 499 *1e-3
Fehler_h1 = 2 *1e-3

h2 = 495 *1e-3
Fehler_h2 = 2 *1e-3

q=np.array([0.2,0.4,0.6,0.8,1])
```

Berechnung von $\langle h \rangle$ und $\Delta\langle h \rangle$ nach (24) bzw. (25):

```
h_mean = h2 + (h1-h2)*q
Fehler_h_mean = np.sqrt((Fehler_h2*q)**2+(Fehler_h1*(1-q))**2) *1e-3
```

Werte für ρ_f aus den Temperaturwerten (Tab.2) und Abbildung 1 entnommen:

```
rho_PEG2 = 1.1476 *1e3
Fehler_rho_PEG2 = 0.0054 *1e3
```

Berechnung von η und $\Delta\eta$ nach (22) bzw. (23):

```
eta2 = np.pi*(rho_PEG2*h_mean*g)*(R**4)*t2/(8*L*V)
Fehler_eta2 = eta2*np.sqrt((Fehler_g/g)**2+(Fehler_rho_PEG2/rho_PEG2)**2+(Fehler_h_mean/h_mean)**2
                           +4*(Fehler_R/R)**2+(Fehler_t2/t2)**2+(Fehler_L/L)**2+(Fehler_V/V)**2)
```

Fitfunktion (19) wird deklariert:

```
from scipy import odr

def fit_func(p, x):
    (c) = p
    return 0*x+c

model = odr.Model(fit_func)
```

darzustellende Daten werden übergeben:

```
x = np.array([1,2,3,4,5])
y = eta2
delta_x = np.array([1,1,1,1,1])
delta_y = Fehler_eta2
```

Startparameter für Ausgleichungsrechnung werden gesetzt, sodass Lösung konvergiert:

```
para0 = [1]

data = odr.RealData(x, y, sx=delta_x, sy=delta_y)
odr = odr.ODR(data, model, beta0=para0 )
out = odr.run()
```

Endgültige Ausgleichungsparameter und ihre Kovarianzmatrix werden ausgelesen:

```
popt = out.beta
perr = out.sd_beta
```

Angabe welche Sigma-Umgebung der Fitfunktion im Diagramm dargestellt werden soll:

```
nstd = 1

popt_top = popt+nstd*perr
popt_bot = popt-nstd*perr
```

Plot-Umgebung wird angegeben:

```
x_fit = np.linspace(min(x)-(max(x)-min(x))/10, max(x)+(max(x)-min(x))/10, 1000)
fit = fit_func(popt, x_fit)
fit_top = fit_func(popt_top, x_fit)
fit_bot = fit_func(popt_bot, x_fit)
```

Diagramm (Abb.5) wird erstellt:

```
fig, ax = plt.subplots(1, figsize=[6.4 *2, 4.8 *2])
plt.ticklabel_format(axis='both', style='sci', scilimits=(0,3), useMathText=True)
plt.errorbar(x, y, yerr=delta_y, lw=1, ecolor='k', fmt='none', capsize=10, label='Messdaten')
plt.plot(x_fit, fit, 'r', lw=1, label='Fit')
ax.fill_between(x_fit, fit_top, fit_bot, color='C3', alpha=.25, label=str(nstd)+r'$\sigma$'+'-
    Umgebung')
plt.title('Messung nach Hagen-Poiseuille')
plt.grid(True)
plt.xlabel('Messung Nr.')
plt.ylabel('${\eta}$ [$Pa$ $s$]')
plt.legend(loc='best')
plt.savefig('figures/212_Fig3.pdf', format='pdf', bbox_inches='tight')
```

Der Chi-Quadrat-Test wird durchgeführt:

```
dof = x.size-popt.size
chisquare = out.sum_square
chisquare_red = chisquare/dof
prob = round(1-chi2.cdf(chisquare,dof),2)*100
```

Auslesen der Messergebnisse:

```
eta2_mean = popt[0]
Fehler_eta2_mean = perr[0]
```

Berechnung und Mittelwertbildung von $\boldsymbol{Re}$ und $\boldsymbol{\Delta Re}$ nach (26) bzw. (27):

```
Re2 = 2*rho_PEG2*V/(t2*np.pi*R*eta2)
Fehler_Re2 = Re2*np.sqrt((Fehler_V/V)**2+(Fehler_rho_PEG2/rho_PEG2)**2+(Fehler_eta2/eta2)**2
                         +(Fehler_R/R)**2+(Fehler_t2/t2)**2)

Re2_mean = np.mean(Re2)
Fehler_Re2_mean = np.sqrt(np.mean(Fehler_Re2**2)/Re2.size)
```

Ausgabe der Messergebnisse wird erstellt:

```
print('Viskosität: ')
print('eta_2 [Pa*s] =',  format_e(eta2_mean), ' +- ',  format_e(Fehler_eta2_mean))
print('Chi-Quadrat =', chisquare)
print('Freiheitsgrade =', dof)
print('Chi-Quadrat reduziert =', chisquare_red)
print('Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten =', prob, '%')
print('\n')
print('Re =',  format_e(Re2_mean), ' +- ',  format_e(Fehler_Re2_mean))
```

5.2.3 Output

```
Viskosität:
eta_2 [Pa*s] = 1.999588e-01  +-  7.65919e-04
Chi-Quadrat = 0.6548238407073612
Freiheitsgrade = 4
Chi-Quadrat reduziert = 0.1637059601768403
Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten = 96.0 %

Re = 1.705669e-01  +-  2.698329e-03
```

Wir erfahren also sofort, dass

$$\eta_2 = 1.9996(77) \times 10^{-1}\ Pa\ s$$
$$Re = 1.706(27) \times 10^{-1}$$

und als Ergebnis auf unseren Anpassungstest:

$$\chi^2_{red} = 1.6 \times 10^{-1}$$

Die Wahrscheinlichkeit ein größeres oder gleiches Chi-Quadrat zu erhalten ist

$$P \approx 96.0\%$$

und wir erhalten das Diagramm in Abb.5.

6 Fazit

Unsere zwei verschiedenen Messmethoden liefern uns zum Schluss folgende Ergebnisse:

$$\eta_1 = 1.942(80) \times 10^{-1}\ Pa\ s$$
$$\eta_2 = 1.9996(77) \times 10^{-1}\ Pa\ s$$

Der Unterschied zwischen diesen Werten berägt nur $0.7\,\sigma$ und ist somit nicht signifikant. Man sollte trotzdem nicht vergessen, dass die Temperaturen der Flüssigkeit sich leicht unterschieden haben:

$$T_1 = 21.0(5)\ °C$$
$$T_2 = 22.0(5)\ °C$$

Das bedeutet, dass die Änderung der Viskosität mit der Temperatur innerhalb unserer Messunsicherheiten liegt. Um dies genauer zu überprüfen, sollte man denselben Versuch bei verschiedenen Temperaturen wiederholen, um systematische Fehler zwischen den verschiedenen Versuchen auszuschließen und am besten dafür den Versuch verwenden, der eine geringere Messunsicherheit aufweist und leichter durchzuführen ist (in diesem Fall die Messung mit der Kapillare).

Die Chi-Quadrat-Tests liefern uns:

$$\chi^2_{red,1} = 5.0$$
$$\chi^2_{red,2} = 1.6 \times 10^{-1}$$

Die Werte lassen darauf schließen, dass beim Fit an die kleinen Kugeldurchmesser im ersten Versuch trotzdem signifikante Fehler aufgetreten sind. Zum Beispiel konnten uns auf einzelnen Kugeln Luftbläschen nicht aufgefallen sein, oder die Kugeln zu nah am Gefäßrand fallen gelassen worden sein. Der geringe Wert im Kapillarversuch ist möglicherweise rein statistisch bedingt oder durch zu grobe Abschätzung der Ablesegenauigkeit an den entsprechenden Gefäßen

bedingt.

Wir betrachten Tabelle 3. Die Ladenburgsche Korrektur (6) ist für genügend kleine Kugeldurchmesser (in unserem Fall für $\boldsymbol{d < 3mm}$) nicht notwendig, da ihre Sinkgeschwindigkeiten $\boldsymbol{\langle v \rangle}$ dann nicht mehr signifikant (Differenz: $\boldsymbol{< 1\,\sigma}$) vom theoretischen Wert $\boldsymbol{v_{theo}}$ abweichen. Bei größere Kugeldurchmesser nimmt dieser systematische Fehler eine signifikante Größe an und ab einem bestimmten Punkt ($\boldsymbol{\approx Re_{kr}}$) schlägt die laminare Strömung um und wird turbulent. Das erhöht die Abweichung des praktischen vom theoretischen Wert, zusätzlich zum Fehlen der Korrektur.

Bei der Messung mit dem Kugelfallviskosimeter lässt sich die kritische Reynoldszahl als

$$\boldsymbol{Re_{kr} \approx 0.30(5)}$$

aus Abbildung 4 abschätzen und in der Kapillare haben wir eine gemittelte Reynoldszahl von

$$\boldsymbol{Re = 1.706(27) \times 10^{-1}}$$

bestimmt. Damit liegt in der Messung nach Hagen-Poiseuille $\boldsymbol{Re}$ deutlich unterhalb der zuvor bestimmten kritischen Reynoldszahl $\boldsymbol{Re_{kr}}$ (Differenz: $\boldsymbol{-2.6\sigma}$). Wir können also mit großer Sicherheit von einer laminaren Strömung ausgehen und somit die Gültigkeit von (26) untermauern.

7 Abbildungen

Hier sind Abbildungen 3 bis 5 angehängt.

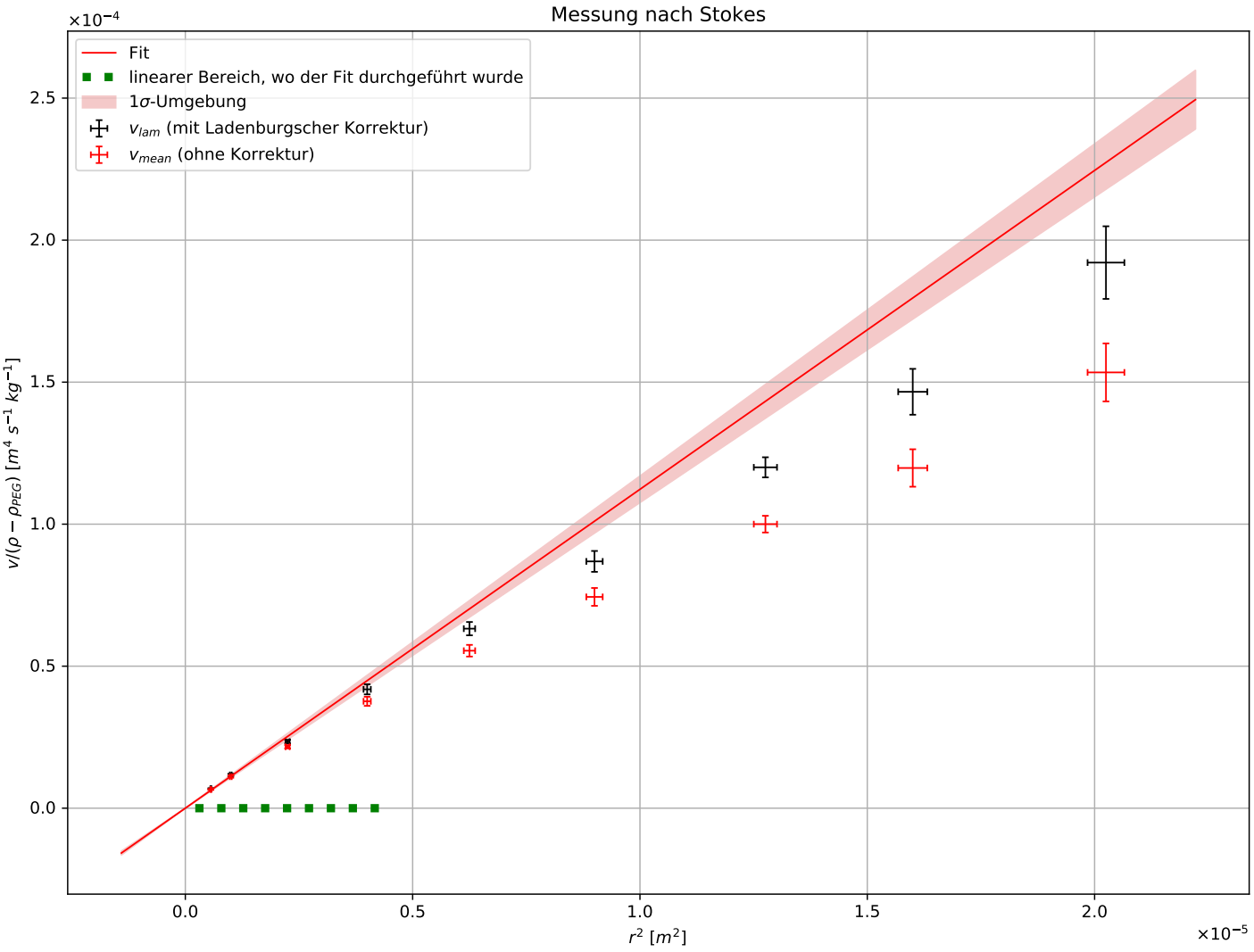
Messung nach Stokes
×10⁻⁴
Fit
linearer Bereich, wo der Fit durchgeführt wurde
1σ-Umgebung
v_{lam} (mit Ladenburgscher Korrektur)
v_{mean} (ohne Korrektur)
$v/(\rho - \rho_{PEG})$ [$m^4\ s^{-1}\ kg^{-1}$]
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
r^2 [m^2]
×10⁻⁵

Abbildung 3

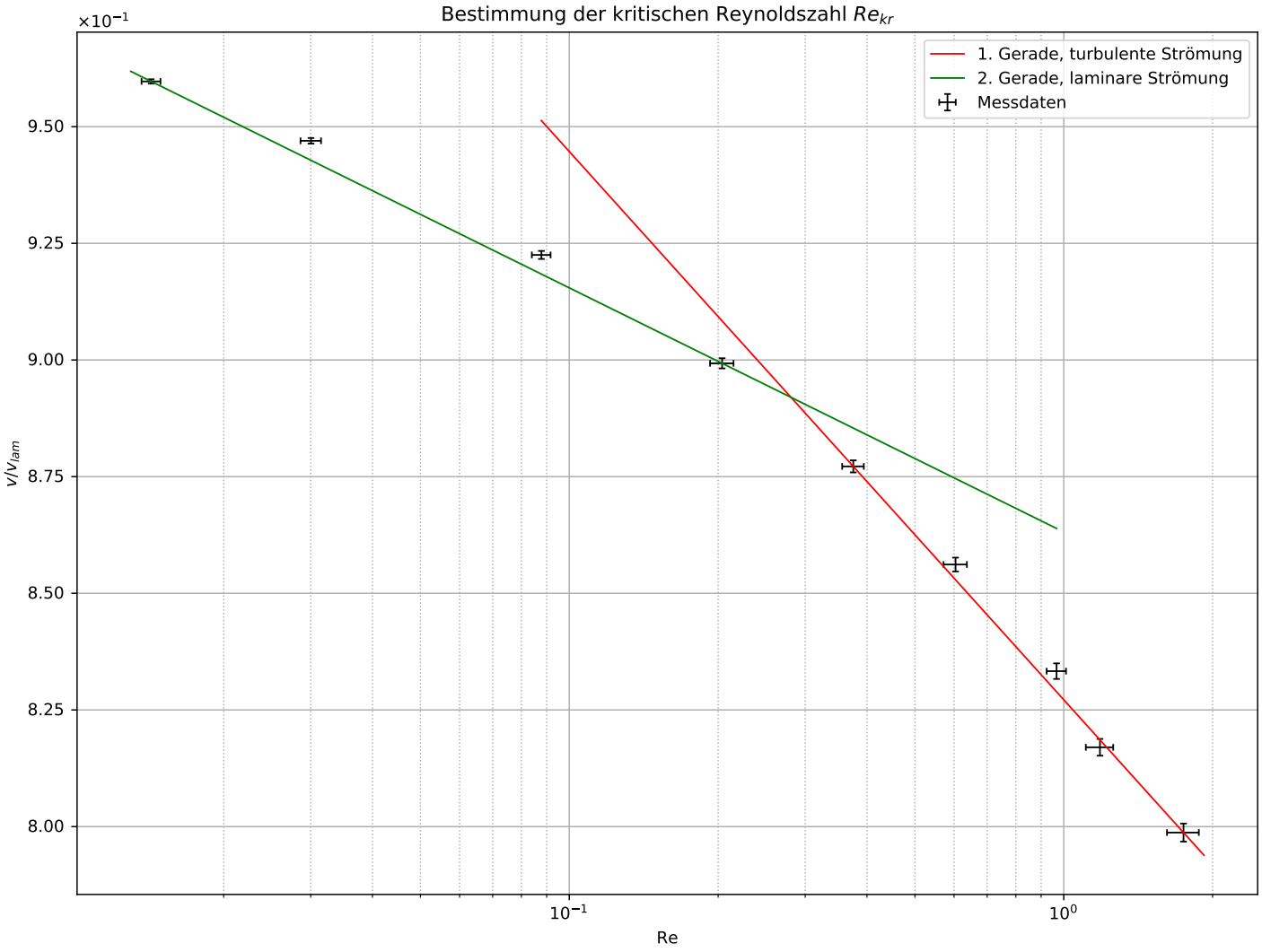
Bestimmung der kritischen Reynoldszahl Re_{kr}
$\times 10^{-1}$
1. Gerade, turbulente Strömung
2. Gerade, laminare Strömung
Messdaten
9.50
9.25
9.00
8.75
8.50
8.25
8.00
v/v_{lam}
10^{-1}
10^{0}
Re

Abbildung 4

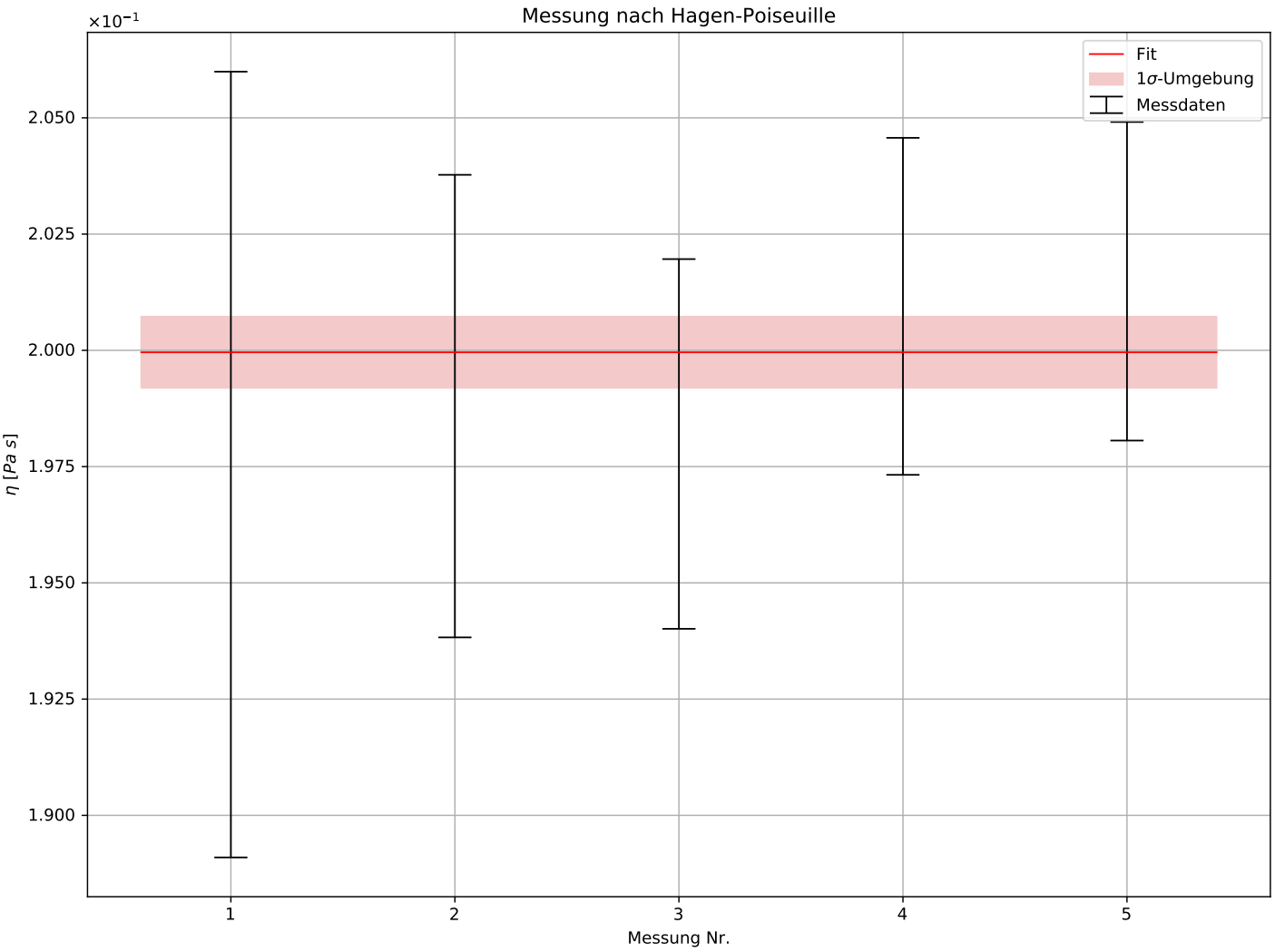
Messung nach Hagen-Poiseuille
×10⁻¹
Fit
1σ-Umgebung
Messdaten
2.050
2.025
2.000
1.975
1.950
1.925
1.900
η [Pa s]
1
2
3
4
5
Messung Nr.

Abbildung 5