

# Vakuum

June 16, 2024

## 1 Fakultät für Physik

### 1.1 Physikalisches Praktikum P2 für Studierende der Physik

Versuch P2-41, 42, 22 (Stand: April 2024)

[Raum F1-19](#)

## 2 Vakuum

Tin Vrkic E-Mail: [uyvpq@student.kit.edu](mailto:uyvpq@student.kit.edu)

Mika Nock E-Mail: [uttzi@student.kit.edu](mailto:uttzi@student.kit.edu)

Gruppennummer: Mo32

Betreuer: Marcel Gaisdörfer

Versuch durchgeführt am: 13.05.2024

---

Beanstandungen zu Protokoll Version \_\_\_\_\_:

Testiert am: \_\_\_\_\_ Testat: \_\_\_\_\_

```
[1]: # importieren aller nötiger Module
from uncertainties import ufloat
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import kafe2
import pathlib
```

```
[2]: # erstellen einer Funktion für kafe2 Fits
def fit_funktion(xy_data, model_function, xy_error, xy_label, title,
↳constraint=[]):
    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)
    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 Durchführung

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

### 3.1 Aufgabe 1: Versuchsaufbau

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

- Machen Sie sich mit dem Versuchsaufbau vertraut.
- Bearbeiten Sie hierzu die folgenden Aufgaben.

#### 3.1.1 Aufgabe 1.1: Orientierung und Beschreibung des Versuchsaufbaus

- Verschaffen Sie sich einen Überblick über die verwendeten Apparaturen und beschreiben Sie sie in eigenen Worten.
- Verfolgen Sie hierzu die Leitungen und identifizieren Sie die verwendeten Elemente in der zugehörigen Skizze.

---

Der vorgefundene Aufbau besteht aus einer Glaskuppel, zwei Pumpen, mehreren Ventilen und verschiedenen Manometern.

In der Glaskuppel (RZ) ist das Verdampferschiffchen zu finden, wobei die Kugelelektroden allerdings fehlen. Von dieser Glaskuppel geht ein Rohr ab, welches das Ionisationsvakuumeter (IM) beinhaltet und durch zwei Ventile von weiteren Teilen abgetrennt ist. V3 führt zu einem Wärmeleitvakuumeter (T3), zu einem Referenzvolumen (RV) und einem Auslassventil (B2). Hinter V2 befinden sich die Turbomolekularpumpe (TMP), ein Wärmeleitvakuumeter (T2), ein austauschbares Leitungselement (L), ein Ventil (V1), noch ein Wärmeleitvakuumeter (V2), ein weiteres Auslassventil und die Drehschieberpumpe (DSP).

Alle Geräte waren mit Atmosphärendruck belüftet und der Verdampfer war mit Indium ausgestattet.

---

#### 3.1.2 Aufgabe 1.2: Gasentladung (Demonstrationsversuch)

- Schalten Sie das Hochspannungsgerät zur Erzeugung der Gasentladungen ein.
- Evakuieren Sie RZ und die Gasentladungsröhre gemeinsam mit Hilfe der DSP.
- Senken Sie den Druck kontinuierlich, bis die Gasentladung erlischt.
- Skizzieren und beschreiben Sie die Gasentladung in Abhängigkeit vom Gasdruck.

---

In diesem Versuch soll die Gasentladung demonstriert werden. Dazu wird nach und nach die Glasröhre evakuiert und gleichzeitig eine Hochspannung angelegt. Dabei treten an einer Anode, die sich in der Glasröhre befindet, Elektronen aus. Diese werden in Richtung der Kathode beschleunigt und können je nach Druck in der Röhre verschiedene Erscheinungen verursachen, wodurch sich das Bild unten ergibt. Hier wurde nichts gemessen, sondern der Effekt wird im folgenden nur qualitativ beschrieben und erklärt.

Zu Beginn befindet sich soviel Luft in der Röhre, dass die mittlere freie Weglänge der Elektronen zu gering ist. Sie stoßen mit den Luftatomen und die Energie, die sie durch die Beschleunigung erhalten wird wieder abgeben. Wird nun die Röhre evakuiert steigt die mittlere freie Weglänge und die Elektronen können immer mehr Energie aufbauen, bevor sie wieder mit einem Atom stoßen. Ab

einem bestimmten Druck können sie schnell genug werden um die Luftmoleküle zu ionisieren, wobei beim Rekombinieren dieser Photonen frei werden. Es ergibt sich somit ein Leuchten, das sich mit sinkendem Druck von der Kathode in Richtung Anode ausbreitet. Bei einer weiteren Entleerung des Rohres können die Elektronen nun auf sehr kurzen Wegen die nötige Energie aufbauen um Photonen freizusetzen. Sie werden somit aus der Anode gelöst, werden stark beschleunigt, treffen auf ein Atom, welches sie zum Leuchten anregen und werden so wieder abgebremst. Dieser Zustand ist in der unteren Grafik zu sehen. Hierbei sind die dunklen Zonen die Strecken, in denen die Elektronen beschleunigt werden und die hellen Zonen die, in denen sie mit Atomen stoßen und so Licht freisetzen.

In der Theorie sollte außerdem das Leuchten bei sehr niedrigen Drücken erlöschen, da dann nicht mehr genug Atome vorhanden sind, um eine wahrnehmbare Menge an Photonen zu erzeugen. Dieser Punkt konnte jedoch im vorliegenden Aufbau nicht erreicht werden.



## 3.2 Aufgabe 2: Saugvermögen und Leitwert

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

- Untersuchen Sie das effektive Saugvermögen von DSP und TMP, sowie den Strömungsleitwert eines dünnen Rohrs.
- Bearbeiten Sie hierzu die folgenden Aufgaben.

### 3.2.1 Aufgabe 2.1: Saugvermögen der DSP

Evakuieren Sie die Apparatur mit Hilfe der DSP und stellen Sie die folgenden funktionalen Zusammenhänge des Druck jeweils **bei T1** geeignet graphisch dar: - Den Druck als Funktion der Zeit  $p(t)$ . - Das Saugvermögen als Funktion des Drucks  $S(p)$ .

In diesem Versuch soll das Saugvermögen der Drehschieberpumpe (DSP) bestimmt werden. Sie wird deshalb genutzt, um die Glasglocke zu evakuieren. Es wird währenddessen der Druck bei  $T_1$  (Wärmeleitvakuummeter) 5 min lang gemessen, wobei alle 5 s ein Messwert genommen wird, bestehend aus dem gemessenen Druck und der Zeit, die seit Anfang der Messung vergangen ist. Es wird dann zunächst der Druck als Funktion der Zeit  $p(t)$  graphisch dargestellt, einmal in normaler Darstellung und einmal logarithmisch im Druck. Danach wird das Saugvermögen für alle Drücke berechnet und ebenfalls dargestellt. In diesem Plot wird dann der Druckbereich identifiziert, in dem das Saugvermögen einigermaßen konstant ist, woraus man dann mit Hilfe eines Fits das Saugvermögen der DSP bestimmen kann.

```
[3]: t_5min = np.linspace(0, 60, 61) * 5 # seconds
t_5min_std = .25 # seconds
druck_std = .1 # 10%

vol_schlauch = 530 * np.pi * (23 * 10**(-3) / 2)**2 * 10**(-3) # m^3
vol_rohr = 530 * np.pi * (2 * 10**(-3) / 2)**2 * 10**(-3) # m^3
volumen_rz_44 = 9.2 * 10**(-3) # m^3
ges_vol_schlauch = vol_schlauch + volumen_rz_44 + .5 * 10**(-3) # m^3
ges_vol_rohr = vol_rohr + volumen_rz_44 + .5 * 10**(-3) # m^3
```

Bei dieser ersten Messungen wurde zunächst ausversehen das Ventil V2 geschlossen gehalten. Zum Vergleich, ob es einen nennenswerten Unterschied macht, wurde die Messreihe zusätzlich zur Richtigen trotzdem durchgeführt.

Die Unsicherheit auf das Ablesen der Zeit wird auf etwa 0.1 s geschätzt, die auf die Druckablesung beträgt für das Messgerät 10 %.

```
[4]: druck_2_1_zu = np.array([1000, 1000, 1.71, 0.266, 0.189, 0.157, 0.139, 0.126, 0.
↪117, 0.108, 0.101, 0.0951,
                                0.00902, 0.00863, 0.00828, 0.00796, 0.00770, 0.00744,
↪0.00716, 0.00692, 0.00669, 0.00646, 0.00629, 0.00614,
                                0.00597, 0.00584, 0.00571, 0.00558, 0.00546, 0.00538,
↪0.00526, 0.00516, 0.00511, 0.00502, 0.00495, 0.00488,
```

```

0.00482, 0.00473, 0.00465, 0.00457, 0.00450, 0.00445,
↪0.00434, 0.00428, 0.00422, 0.00416, 0.00410, 0.00404,
0.00398, 0.00395, 0.00391, 0.00387, 0.00382, 0.00376,
↪0.00373, 0.00369, 0.00367, 0.00362, 0.00357, 0.00355,
0.00352])
druck_2_1_auf = np.array([1000, 1000, 1000, 1000, 1000, 1000, 460, 193, 113, 69.
↪9, 45.8, 31.3,
22.5, 15.9, 11.4, 8.33, 5.98, 4.44, 3.31, 2.48, 1.89,
↪1.48, 1.18, 0.934,
0.763, 0.624, 0.526, 0.450, 0.385, 0.338, 0.301, 0.
↪269, 0.242, 0.222, 0.207, 0.194,
0.181, 0.171, 0.162, 0.154, 0.148, 0.142, 0.137, 0.
↪133, 0.129, 0.126, 0.123, 0.121,
0.118, 0.116, 0.113, 0.111, 0.109, 0.107, 0.106, 0.
↪104, 0.103, 0.101, 0.100, 0.00991,
0.00974])

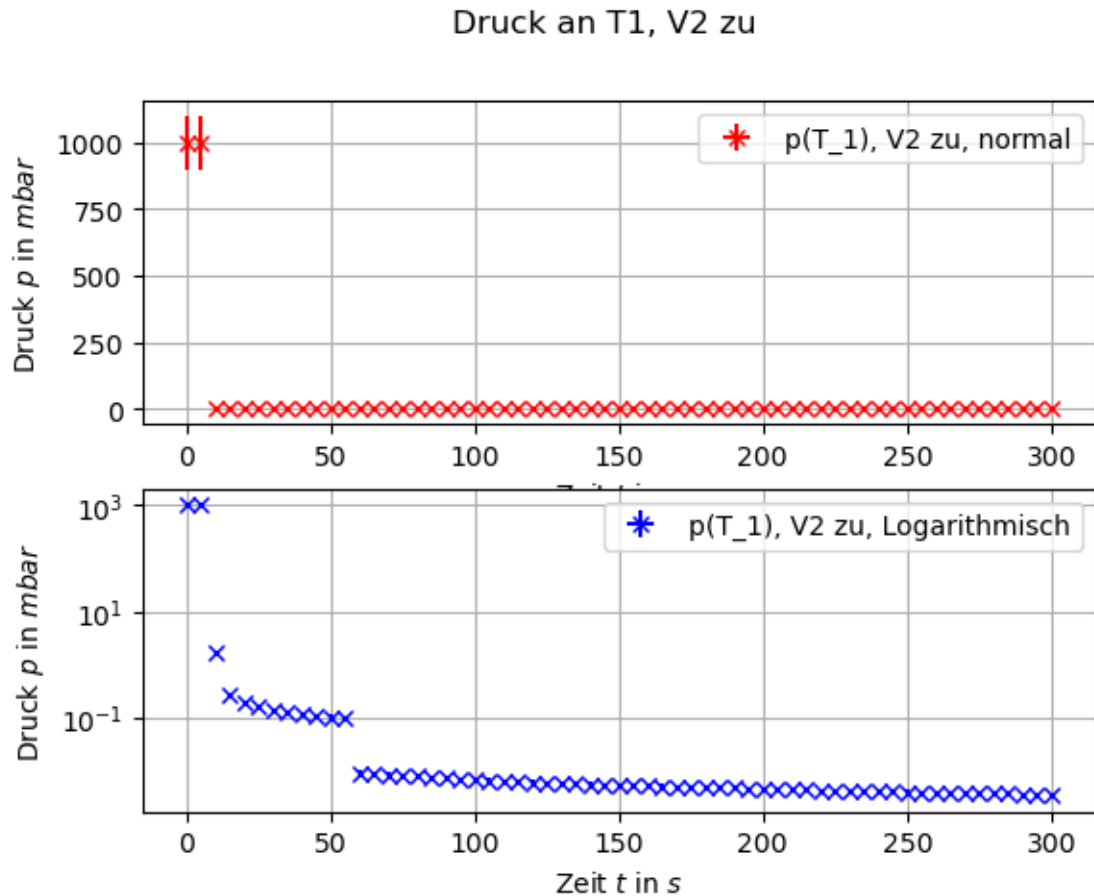
```

```

[5]: fig0, ax0 = plt.subplots(2)
ax0[0].errorbar(t_5min, druck_2_1_zu, xerr=t_5min_std,
↪yerr=druck_2_1_zu*druck_std, fmt="rx", label="p(T_1), V2 zu, normal")
ax0[0].legend(), ax0[0].grid()
ax0[1].errorbar(t_5min, druck_2_1_zu, xerr=t_5min_std,
↪yerr=druck_2_1_zu*druck_std, fmt="bx", label="p(T_1), V2 zu, Logarithmisch")
ax0[1].set_yscale("log")
ax0[1].legend(), ax0[1].grid()
fig0.suptitle("Druck an T1, V2 zu")

for ax in ax0:
    ax.set(xlabel="Zeit $t$ in $s$", ylabel="Druck $p$ in $mbar$")

```



In der logarithmischen Darstellung sieht man gut, welche komischen Dinge die Werte machen, deshalb beschränken wir uns im folgenden, wie von der Aufgabe gefordert, auf die Messreihe mit offenem Ventil V2.

Der erste Wert bei  $t = 0 \text{ s}$  wird jedoch weggelassen, da in der Formel für das Saugvermögen durch den Zeitpunkt  $t_0$  geteilt wird. Um die Division durch 0 zu verhindern, werden dann alle Werte ab einschließlich Sekunde 5 genutzt. Der Vergleichbarkeit wegen wird für folgenden Plot der Wert bei  $0 \text{ s}$  ebenfalls weggelassen.

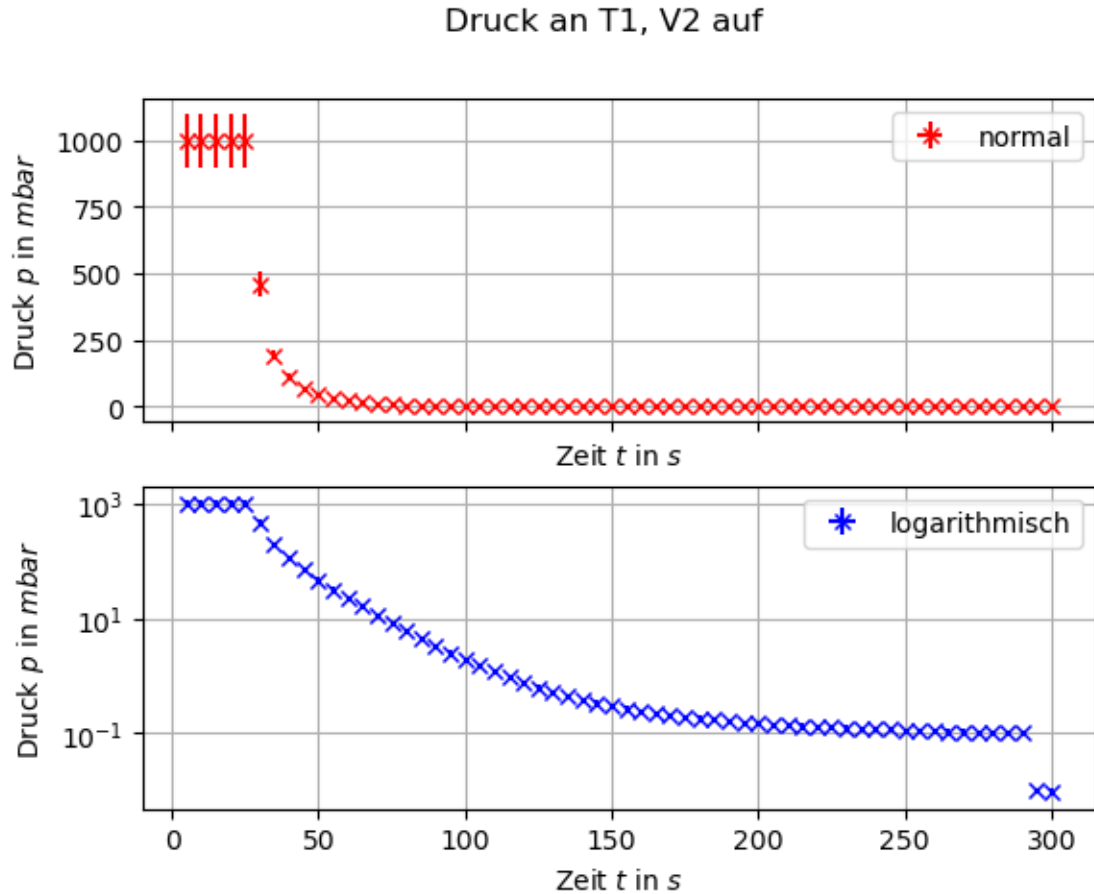
```
[6]: fig, ax = plt.subplots(2, sharex=True)
ax[0].errorbar(t_5min[1:], druck_2_1_auf[1:], xerr=t_5min_std,
               yerr=druck_2_1_auf[1:]*druck_std, fmt="rx", label="normal")
ax[0].legend(), ax[0].grid()

ax[1].errorbar(t_5min[1:], druck_2_1_auf[1:], xerr=t_5min_std,
               yerr=druck_2_1_auf[1:]*druck_std, fmt="bx", label="logarithmisch")
ax[1].set_yscale("log")
ax[1].legend(), ax[1].grid()
```



```
fig.suptitle("Druck an T1, V2 auf")

for ax in ax:
    ax.set(xlabel="Zeit $t$ in $s$", ylabel="Druck $p$ in $mbar$")
```



In der logarithmischen Darstellung sollte der mittlere Teil der Kurve eigentlich einen einigermaßen geraden Verlauf haben. Warum das so ist, ist schwer zu sagen, auf jeden Fall liegt es an den Messwerten, die nicht ganz so ausgefallen sind wie erwartet.

Um nun das Saugvermögen der DSP zu bestimmen, gehen wir von der, in der Vorbereitung hergeleiteten, Formel aus:

$$\ln\left(\frac{p}{p_0}\right) = -\frac{S}{V}(t - t_0)$$

$$\Leftrightarrow S(p) = -\frac{V}{(t - t_0)} \ln\left(\frac{p}{p_0}\right)$$

mit dem Anfangsdruck  $p_0 = 1000 \text{ mbar}$  und dem Volumen des Rezipienten  $V_{RZ} = 9.2 \text{ l} = 9.2 \text{ dm}^3$ .

Das Saugvermögen bei einem Druck von  $10 \text{ mbar}$  beträgt  $S(10 \text{ mbar}) \approx 2.283 \frac{\text{m}^3}{\text{h}}$

```

[7]: saugvermögen = - ( ges_vol_schlauch / t_5min[1:] ) * np.log(druck_2_1_auf[1:] /
↳druck_2_1_auf[1]) * 3600
saugvermögen_2_1_std = 3600*(ges_vol_schlauch/t_5min[1:]) * np.sqrt( (np.
↳log(druck_2_1_auf[1:]/druck_2_1_auf[1]) * t_5min_std / t_5min[1:] )**2 +
↳(druck_std/druck_2_1_auf[1:] )**2)

fig1, ax1 = plt.subplots(2, sharey=True)
ax1[0].errorbar(druck_2_1_auf[1:], saugvermögen, xerr=druck_2_1_auf[1:
↳]*druck_std, yerr=saugvermögen_2_1_std, fmt="rx", label="$S(p)$ normal")
ax1[0].legend(), ax1[0].grid()

ax1[1].errorbar(druck_2_1_auf[1:], saugvermögen, xerr=druck_2_1_auf[1:
↳]*druck_std, yerr=saugvermögen_2_1_std, fmt="bx", label="$S(p)$
↳logarithmisch")
ax1[1].set_xscale("log")
ax1[1].grid()

fig1.suptitle("Saugvermögen der DSP bei versch. Drücken")

for ax in ax1:
    ax.set(xlabel="Druck $p$ in $mbar$", ylabel="Saugvermögen $$$ in $m^3/h$")

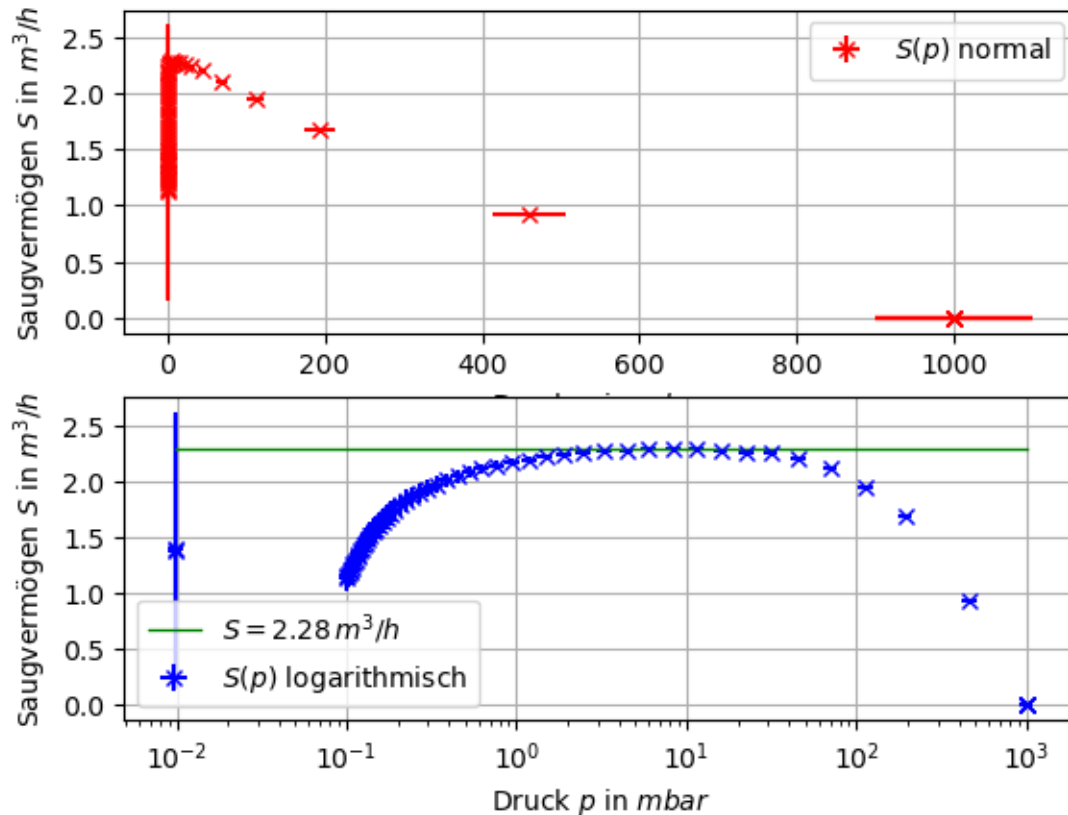
saugvermögen_10mbar = -3600*(ges_vol_schlauch/t_5min[14])*np.
↳log(druck_2_1_auf[14]/druck_2_1_auf[1])
saugvermögen_10mbar_std = 3600*(ges_vol_schlauch/t_5min[14]) * np.sqrt( (np.
↳log(druck_2_1_auf[14]/druck_2_1_auf[1]) * t_5min_std / t_5min[14]) **2 +
↳(druck_std/druck_2_1_auf[14]) **2)
print(f"Saugvermögen bei 10mbar: S(10mbar)={saugvermögen_10mbar:.3f} +/-
↳{saugvermögen_10mbar_std:.3f}) m³/h")
ax1[1].plot(druck_2_1_auf[1:], np.ones(druck_2_1_auf[1:].
↳size)*saugvermögen_10mbar, color="green", linewidth=1.0,
↳label=f"$S=${saugvermögen_10mbar:.2f}$\\,m^3/h$")
ax1[1].legend()

plt.show()

```

Saugvermögen bei 10mbar: S(10mbar)=(2.283 +/- 0.009) m³/h

## Saugvermögen der DSP bei versch. Drücken



In der normalen Darstellung erkennt man wenig von den drei erwarteten Bereichen. Schaut man sich den Graphen jedoch logarithmisch in  $p$  an, kann man grob die drei Bereiche identifizieren:

- Bereich I: Saugvermögen nimmt von  $10^3 \text{ mbar}$  bis etwa  $5 \cdot 10^1 \text{ mbar}$  zu, bis zu einem Saugvermögen von etwa  $(2.283 \pm 0.006) \frac{\text{m}^3}{\text{h}}$
- Bereich II: Saugvermögen bleibt zwischen  $5 \cdot 10^1 \text{ mbar}$  bis etwa  $10^0 \text{ mbar}$  relativ konstant
- Bereich III: Saugvermögen sinkt wieder, da das Grob- in ein Feinvakuum übergeht, wobei der Graph bei einem Druck von  $10^{-1} \text{ mbar}$  einen Knick hat. Das liegt daran, dass hier die Messwerte selbst einen größeren Sprung aufweisen, den man im zweiten Plot oben gegen Ende der Messreihe in der logarithmischen Darstellung sieht.

Laut Datenblatt wird für die DSP unter Normbedingungen ein Saugvermögen von etwa  $2.5 \frac{\text{m}^3}{\text{h}}$  erwartet. Da unsere Messung nicht unbedingt unter exakten Normbedingungen stattgefunden hat, z.B. Umgebungsdruck- und Temperatur, ist unser Wert von  $(2.283 \pm 0.006) \frac{\text{m}^3}{\text{h}}$  relativ gut.

Schließlich noch der exponentielle Fit an  $p(t)$ :

Zunächst haben wir den Fit mit einer explizit exponentiellen Modellfunktion versucht, der aber gar

nicht konvergiert. Deshalb im folgenden nur der Fit mit der “linearisierten” Modellfunktion:

$$p(t) = p_0 \cdot \exp\left(-\frac{S}{V}(t - t_0)\right)$$

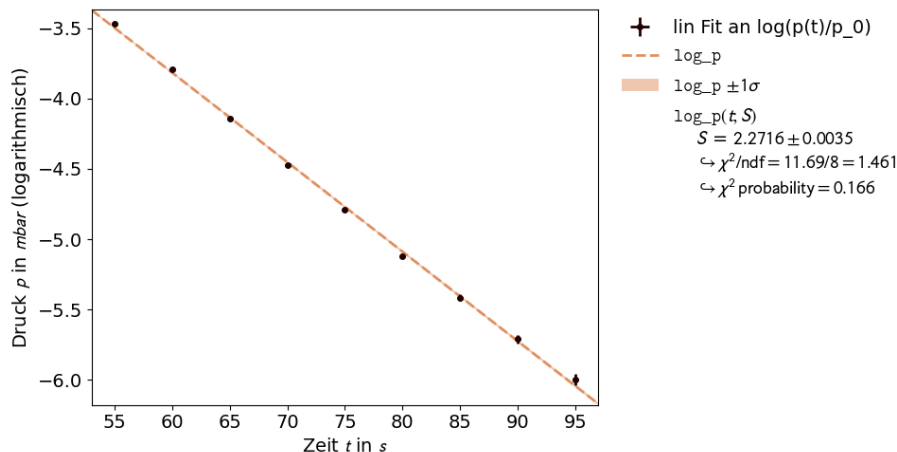
$$\Leftrightarrow \ln\left(\frac{p}{p_0}\right) = -\frac{S}{V}(t - t_0)$$

Die Werte vom Anfangsdruck  $p_0$  und dem Gesamtvolumen der Apparatur  $V$  werden direkt in die Formel eingesetzt, das Saugvermögen  $S$  wird mit dem Anfangswert  $S \approx (2.283 \pm 0.009) \frac{\text{m}^3}{\text{h}}$  übergeben, wird aber für die Anpassung flexibel gelassen.

```
[8]: def log_p(t, S=saugvermögen_10mbar):
      return (- S * t / (ges_vol_schlauch*3600))

fit_2_1_data_log = np.array([ t_5min[11:20] , np.log(druck_2_1_auf[11:20] /
      ↪1000) ])
fit_2_1_error_log = np.array([ np.ones(9)*t_5min_std , np.absolute(druck_std) *
      ↪np.sqrt( (1/druck_2_1_auf[11:20])**2 + (1/1000)**2 ) ] )
fit_2_1_label_log = [ "Zeit $t$ in $s$" , "Druck $p$ in $mbar$ (logarithmisch)"
      ↪]
fit_2_1_title_log = "lin Fit an log(p(t)/p_0)"

fit_res_2_1_log = fit_funktion(fit_2_1_data_log, log_p, fit_2_1_error_log,
      ↪fit_2_1_label_log, fit_2_1_title_log)
fit_res_2_1_log[2].plot()
fit_res_2_1_log[2].show()
print(f"Saugvermögen: S = {fit_res_2_1_log[0][0]:.3f} +/-
      ↪{fit_res_2_1_log[1][0]:.3f} m³/h")
```



Saugvermögen: S = 2.272 +/- 0.004 m³/h

Der Fit liefert ein relativ gutes Ergebnis für das Saugvermögen von  $S = (2.272 \pm 0.004) \frac{m^3}{h}$ , was relativ nah am erwarteten Wert bei Normbedingungen von  $2.5 \frac{m^3}{h}$  liegt, wobei es wie gesagt logisch ist, dass dieser nicht ganz erreicht werden kann, da unsere Messung nicht unter Normbedingungen stattgefunden hat. Die  $\chi^2$  probability ist mit 0.166 in Ordnung, könnte aber auch noch besser sein. Das könnte man zwar durch eine Erhöhung der Unsicherheit auf die Zeitmessung verbessern, allerdings würde das das tatsächliche Ergebnis wahrscheinlich zu sehr verfälschen.

### 3.2.2 Aufgabe 2.2: Leitwert eines dünnen Rohrs

- Bei der Verbindungsleitung L [hier](#) handelt es sich im Originalaufbau um einen Metallwellschlauch. Tauschen Sie diesen gegen das bereitliegende etwa gleichlange dünne Metallrohr aus.
- Bestimmen Sie die Drucke  $p_1$  (bei  $T_1$ ) vor und  $p_2$  (bei  $T_2$ ) hinter dem Rohr als Funktion der Zeit.
- Bestimmen Sie aus den gewonnenen Daten den Leitwert des Rohrs als Funktion des Drucks.

In diesem Versuch soll der Leitwert des dünnen Rohrs, durch das der Metallwellschlauch ersetzt wird, bestimmt werden. Dazu werden diesmal 2 Drücke gemessen, einmal vor ( $p_1$  bei  $T_1$ ) und hinter ( $p_2$  bei  $T_2$ ) dem Rohr. Auch hier wird wieder jeweils der Druck als Funktion der Zeit  $p(t)$  geplottet, dann aus den Drücken jeweils das Saugvermögen vor und hinter dem Rohr bestimmt und ebenfalls geplottet und wieder, komplett analog zu 2.1, aus dem konstanten Druckbereich des Saugvermögens das Saugvermögen vor und hinter dem Rohr bestimmt. Das Saugvermögen hinter dem Rohr entspricht dem effektiven Saugvermögen der DSP, den nominellen Wert haben wir in 2.1 schon bestimmt. Aus diesen beiden Werten lässt sich nun der Leitwert des Rohrs aus der Messung bestimmen, den wir mit der theoretisch erwarteten Wert nach der Knudsen-Gleichung vergleichen.

```
[9]: druck_p1_2_2 = np.array([1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 750,
↪380, 245, 171,
                                128, 102, 78.0, 64.4, 52.3, 43.9, 37.4, 31.4, 27.5, 24.
↪1, 20.9, 18.4,
                                16.3, 14.6, 12.9, 11.7, 10.6, 9.61, 8.91, 8.00, 7.25,
↪6.71, 6.26, 5.74,
                                5.29, 4.90, 4.61, 4.29, 4.01, 3.76, 3.54, 3.33, 3.14,
↪2.97, 2.80, 2.67,
                                2.53, 2.41, 2.30, 2.18, 2.09, 2.00, 1.91, 1.84, 1.76,
↪1.70, 1.64, 1.57,
                                1.52])
druck_p2_2_2 = np.array([1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000,
↪1000, 1000, 1000,
                                1000, 1000, 1000, 1000, 1000, 800, 530, 375, 288, 236,
↪196, 166,
                                148, 131, 115, 106, 97.1, 89.3, 81.8, 75.4, 70.2, 66.
↪4, 62.6, 59.3,
                                56.1, 53.0, 50.1, 47.3, 45.3, 43.6, 42.1, 40.5, 39.2,
↪37.9, 36.4, 35.1,
```

```

33.9, 32.8, 31.6, 30.5, 29.3, 28.5, 27.8, 27.1, 26.5,
↪26.0, 25.3, 24.6,
24.1])

```

```

[10]: fig2, ax2 = plt.subplots(2, sharex=True)
ax2[0].errorbar(t_5min, druck_p1_2_2, xerr=t_5min_std,
↪yerr=druck_p1_2_2*druck_std, fmt="rx", label="p1, normal")
ax2[0].errorbar(t_5min, druck_p2_2_2, xerr=t_5min_std,
↪yerr=druck_p2_2_2*druck_std, fmt="bx", label="p2, normal")
ax2[0].legend(), ax2[0].grid()

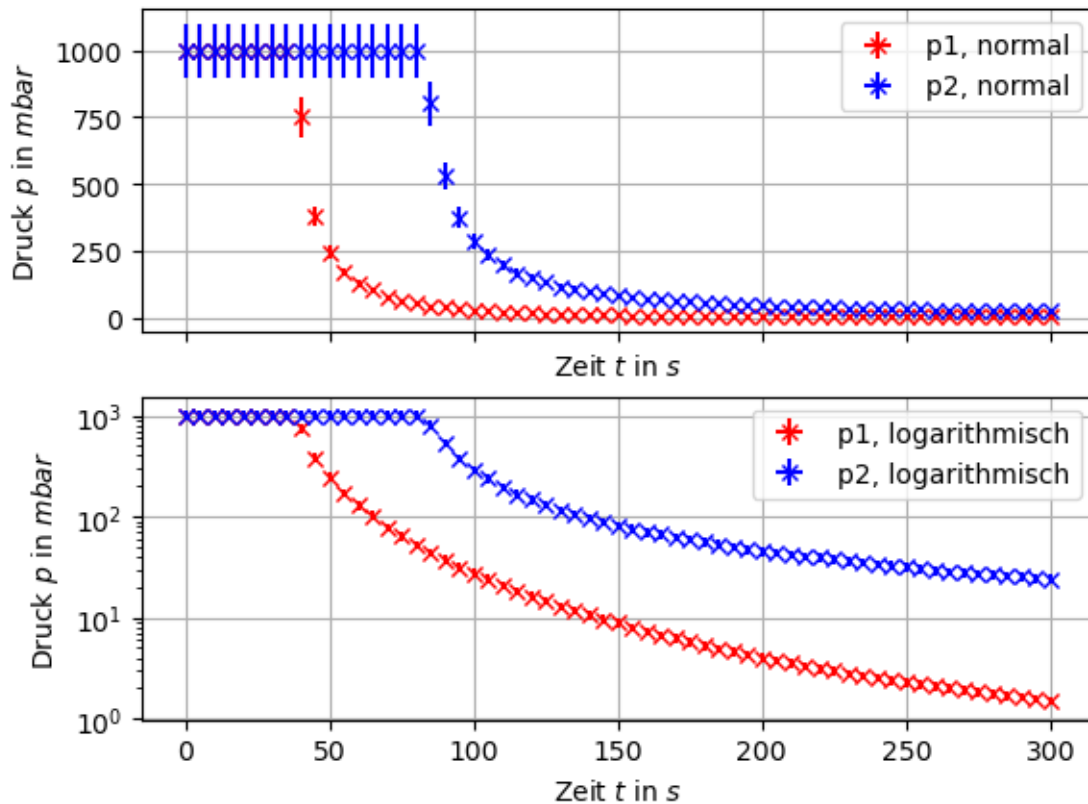
ax2[1].errorbar(t_5min, druck_p1_2_2, xerr=t_5min_std,
↪yerr=druck_p1_2_2*druck_std, fmt="rx", label="p1, logarithmisch")
ax2[1].errorbar(t_5min, druck_p2_2_2, xerr=t_5min_std,
↪yerr=druck_p2_2_2*druck_std, fmt="bx", label="p2, logarithmisch")
ax2[1].set_yscale("log")
ax2[1].legend(), ax2[1].grid()

fig2.suptitle("Drücke p1 (bei T1) und p2 (bei T2)")

for ax in ax2:
    ax.set(xlabel="Zeit $t$ in $s$", ylabel="Druck $p$ in $mbar$")

```

### Drücke p1 (bei T1) und p2 (bei T2)



```
[11]: saugvermögen2_p1 = - ( ges_vol_rohr / t_5min[1:] ) * np.log(druck_p1_2_2[1:] /
↳druck_p1_2_2[1]) * 3600
saugvermögen2_p2 = - ( ges_vol_rohr / t_5min[1:] ) * np.log(druck_p2_2_2[1:] /
↳druck_p2_2_2[1]) * 3600

saugvermögen2_p1_std = 3600*(ges_vol_rohr/t_5min[1:]) * np.sqrt( (np.
↳log(druck_p1_2_2[1:]/druck_p1_2_2[1]) * t_5min_std / t_5min[1:] )**2 +
↳(druck_std/druck_p1_2_2[1:] )**2)
saugvermögen2_p2_std = 3600*(ges_vol_rohr/t_5min[1:]) * np.sqrt( (np.
↳log(druck_p2_2_2[1:]/druck_p2_2_2[1]) * t_5min_std / t_5min[1:] )**2 +
↳(druck_std/druck_p2_2_2[1:] )**2)

fig3, ax3 = plt.subplots()
ax3.errorbar(druck_p1_2_2[1:], saugvermögen2_p1, xerr=druck_p1_2_2[1:
↳]*druck_std, yerr=saugvermögen2_p1_std, fmt="rx", label="p1 bei T1,
↳logarithmisch")
```

```

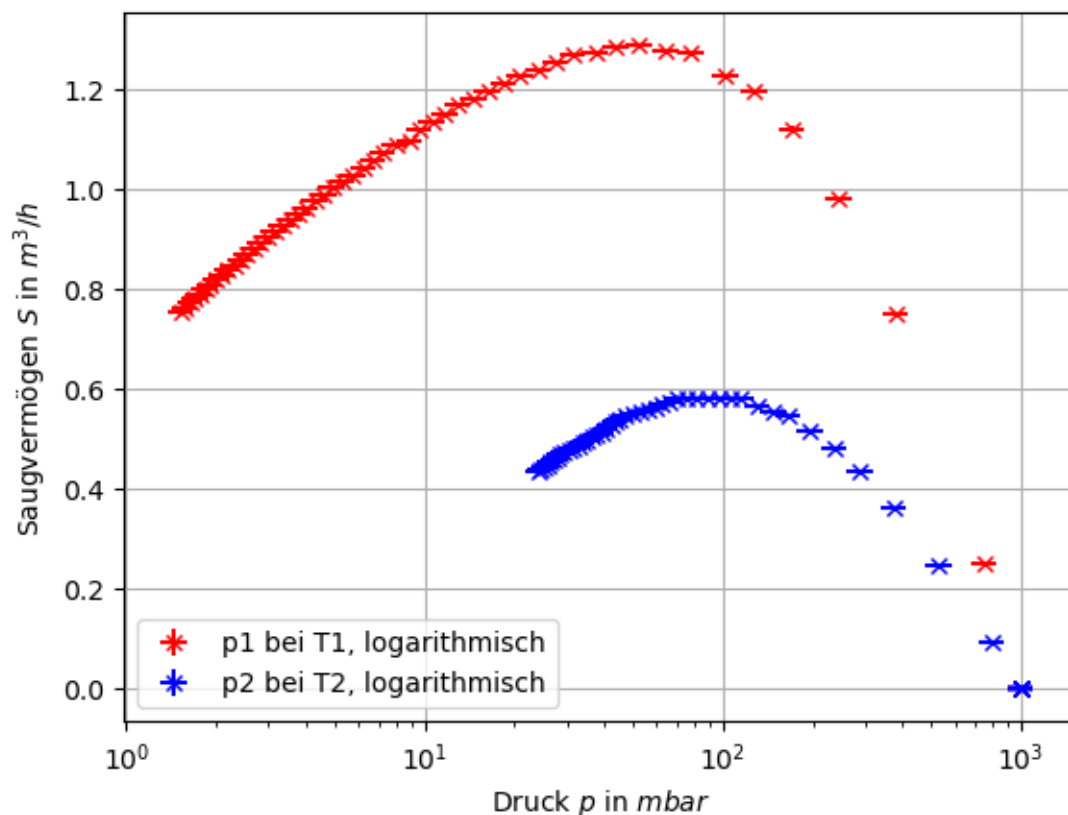
ax3.errorbar(druck_p2_2_2[1:], saugvermögen2_p2, xerr=druck_p2_2_2[1:
    ↳]*druck_std, yerr=saugvermögen2_p2_std, fmt="bx", label="p2 bei T2,␣
    ↳logarithmisch")
ax3.set_xscale("log")
ax3.legend(), ax3.grid()
fig3.suptitle("Saugvermögen der DSP bei versch. Drücken bei T1, T2")
ax3.set(xlabel="Druck $p$ in $mbar$", ylabel="Saugvermögen $$S$ in $m^3/h$")
plt.show()

saugvermögen2_p1_50mbar = -3600*(ges_vol_rohr/t_5min[16])*np.
    ↳log(druck_p1_2_2[16]/druck_p1_2_2[1])
print(f"Saugvermögen bei T1 bei etwa 52.3mbar: S(52.
    ↳3mbar)={saugvermögen2_p1_50mbar} m³/h")

saugvermögen2_p2_100mbar = -3600*(ges_vol_rohr/t_5min[28])*np.
    ↳log(druck_p2_2_2[28]/druck_p2_2_2[1])
print(f"Saugvermögen bei T2 bei etwa 97.1mbar: S(97.
    ↳1mbar)={saugvermögen2_p2_100mbar} m³/h")

```

Saugvermögen der DSP bei versch. Drücken bei T1, T2





Saugvermögen bei T1 bei etwa 52.3mbar:  $S(52.3\text{mbar})=1.2882273547710714 \text{ m}^3/\text{h}$

Saugvermögen bei T2 bei etwa 97.1mbar:  $S(97.1\text{mbar})=0.58177074270213 \text{ m}^3/\text{h}$

Obiger Plot zeigt wieder den Druck logarithmisch gegen das Saugvermögen aufgetragen an, wie in 2.1, nur diesmal mit den beiden Drücken  $p_1$  und  $p_2$  bei  $T_1$  und  $T_2$ . Es wird wieder der konstante zweite Bereich gesucht, der für

- $p_1$  zwischen 20.9 mbar und 78 mbar

- $p_2$  zwischen 70.2 mbar und 115 mbar

liegt (die Werte sind schon aus dem Array der Messung oben identifiziert worden).

Als Anfangswert für das Saugvermögen für den Fit werden die Werte

- $S_1 \approx 1.289 \frac{\text{m}^3}{\text{h}}$  bei  $p_1 = 52.3 \text{ mbar}$

- $S_2 \approx 0.582 \frac{\text{m}^3}{\text{h}}$  bei  $p_2 = 97.1 \text{ mbar}$

genutzt.

Für die tatsächliche Bestimmung von  $S_1$  und  $S_2$  gehen wir wie in 2.1 vor und nutzen die Modellfunktion in logarithmischer Darstellung:

```
[12]: def lin_modell_p1(t, S1=saugvermögen2_p1_50mbar):
        return (- S1 * t / (ges_vol_rohr*3600) )

def lin_modell_p2(t, S2=saugvermögen2_p2_100mbar):
        return (- S2 * t / (ges_vol_rohr*3600) )

fit_2_2_data1 = np.array([ t_5min[14:20] , np.log(druck_p1_2_2[14:20] / 1000) ])
fit_2_2_data2 = np.array([ t_5min[26:33] , np.log(druck_p2_2_2[26:33] / 1000) ])

fit_2_2_error1 = np.array([ np.ones(6)*t_5min_std , np.absolute(druck_std) * np.
    ↪sqrt( (1/druck_p1_2_2[14:20])**2 + (1/1000)**2 ) ] )
fit_2_2_error2 = np.array([ np.ones(7)*t_5min_std , np.absolute(druck_std) * np.
    ↪sqrt( (1/druck_p2_2_2[26:33])**2 + (1/1000)**2 ) ] )

fit_2_2_label1 = [ "Zeit $t$ in $$s$" , "Druck $p1$ in $mbar$ (logarithmisch)" ]
fit_2_2_label2 = [ "Zeit $t$ in $$s$" , "Druck $p2$ in $mbar$ (logarithmisch)" ]

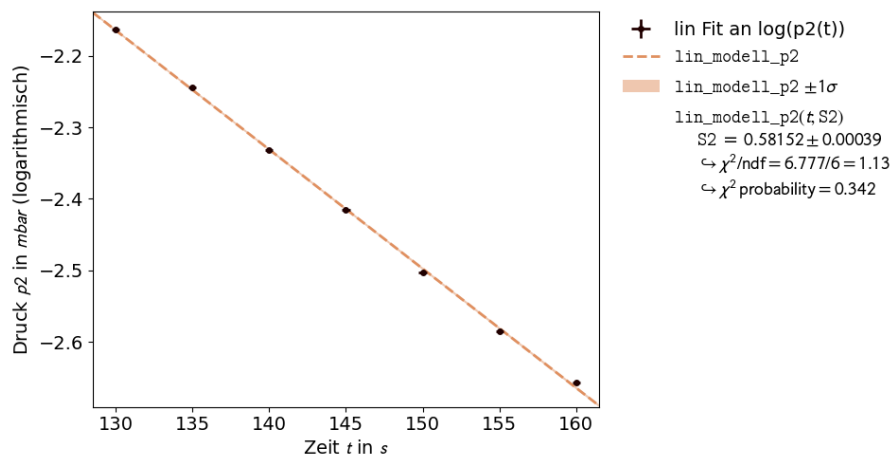
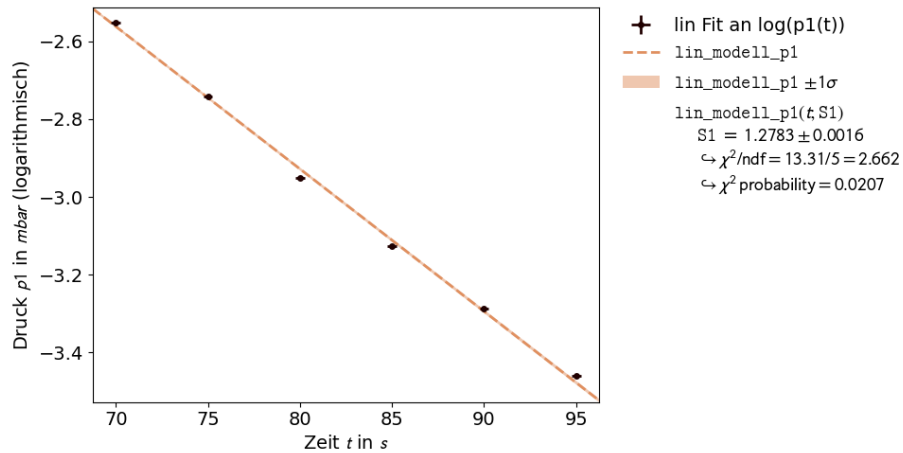
fit_2_2_title1 = "lin Fit an log(p1(t))"
fit_2_2_title2 = "lin Fit an log(p2(t))"

fit_res1_2_2 = fit_funktion(fit_2_2_data1, lin_modell_p1, fit_2_2_error1,
    ↪fit_2_2_label1, fit_2_2_title1)
fit_res2_2_2 = fit_funktion(fit_2_2_data2, lin_modell_p2, fit_2_2_error2,
    ↪fit_2_2_label2, fit_2_2_title2)

fit_res1_2_2[2].plot()
fit_res2_2_2[2].plot()
```

```
fit_res1_2_2[2].show()
fit_res2_2_2[2].show()

fit_res2_2_2[1][0]
```



[12]: 0.0003928166748655592

Bei  $T_1$  beträgt das Saugvermögen  $S_1 = (1.278 \pm 0.002) \frac{m^3}{h}$  mit einer sehr niedrigen  $\chi^2$  probability von 0.0207.

Bei  $T_2$  hingegen beträgt das Saugvermögen  $S_2 = (0.5815 \pm 0.0004) \frac{m^3}{h}$  mit einer sehr guten  $\chi^2$  probability von 0.342.

Schließlich soll noch der Strömungsleitwert berechnet werden. Dieser lässt sich über die effektive Saugleistung berechnen:

$$\frac{1}{S_{eff}} = \frac{1}{L} + \frac{1}{S}$$

$$L = \frac{1}{\frac{1}{S_{eff}} - \frac{1}{S}}$$

$S_{eff}$  entspricht hier dem  $S_2 = (0.5815 \pm 0.0004) \frac{m^3}{h}$  von eben, also hinter dem Rohr, und  $S$  dem Saugvermögen der DSP aus der 2.1  $S = (2.272 \pm 0.004) \frac{m^3}{h}$ . Zusätzlich soll dieser bestimmte Wert mit dem erwarteten Wert nach der Knudsen-Gleichung verglichen werden. Die Vorbereitung verweist hierzu auf einen [englischsprachigen Wikipedia-Artikel](#) zur Knudsen-Gleichung, die dann folgendermaßen gegeben ist:

$$L = 12 \frac{d^3}{l} \frac{l}{s}$$

wobei  $[L] = \frac{l}{s}$  und  $[d] = [l] = dm$

Auch wenn sich diese Gleichung in der Dimensionsanalyse als fehlerhaft erweist, wird sie im Folgenden für die Berechnung des erwarteten Wertes von  $L$  genutzt, denn die Vorbereitung verweist zusätzlich auf eine Gleichung, die als Ingenieursformel bezeichnet wird, in der noch ein durchschnittlicher Druck  $\bar{p}$  vorkommt. Die Dimensionsanalyse zeigt auch bei dieser Gleichung, dass sich die Einheiten nicht zu *Liter/Sekunde* vereinfachen lassen. Mit dem durchschnittlichen Druck sind wahrscheinlich die Drücke vor und hinter dem Rohr gemeint, da sich diese aber innerhalb von 5 min sehr stark gegeneinander ändern, bleibt der durchschnittliche Druck nicht konstant, und damit erhält man keinen einzelnen Wert für den Leitwert. Außerdem würde dann der theoretisch erwartete Wert von  $L$  wieder von der Messung selbst abhängen, was eigentlich nicht gewollt ist. Deshalb wird obige Formel für die Berechnung des Leitwerts genutzt.

Zunächst der aus der Messung bestimmte Wert von  $L$ :

```
[13]: # Maße des Rohrs in cm
d_rohr = .2 * 10 #dm
l_rohr = 53. * 10 #dm

saugvermögen_eff = fit_res2_2_2[0][0] / 3.6      # :3.6 für m^3/h -> l/s
saugvermögen_eff_std = fit_res2_2_2[1][0] / 3.6
saugvermögen3 = fit_res_2_1_log[0][0] / 3.6
saugvermögen3_std = fit_res_2_1_log[1][0] / 3.6

L_mess = ( saugvermögen_eff * saugvermögen3 ) / ( saugvermögen3 -
↳ saugvermögen_eff )
# Fehlerfortpflanzung
L_mess_std = np.sqrt( ( saugvermögen_eff**2 * saugvermögen3_std )**2 + (
↳ saugvermögen3**2 * saugvermögen_eff_std )**2 ) / ( saugvermögen3 -
↳ saugvermögen_eff )**2

L_exp = 12 * d_rohr**3 / l_rohr

print(f"Bestimmter Wert von L: L = {L_mess:.4f} +/- {L_mess_std:.4f} l/s")
print(f"Erwarteter Wert von L: L = {L_exp:.4f} l/s")
```

Bestimmter Wert von L:  $L = 0.2171 \pm 0.0002 \text{ l/s}$

Erwarteter Wert von L:  $L = 0.1811 \text{ l/s}$

Der aus der Messung bestimmte Wert beträgt  $L_{mess} = (0.2171 \pm 0.0002) \frac{\text{l}}{\text{s}}$ . Der erwartete Wert liegt mit  $L_{exp} = 0.1811 \frac{\text{l}}{\text{s}}$  zwar außerhalb des Unsicherheitsbereichs unseres bestimmten Wertes, ihr Verhältnis ist mit  $L_{mess}/L_{exp} \approx 1.20 = 120\%$  trotzdem noch relativ in Ordnung. Unser Wert liegt nur etwa 20% über dem erwarteten.

### 3.2.3 Aufgabe 2.3: Saugvermögen der TMP

- Nehmen Sie die TMP in Betrieb und bestimmen Sie analog zu **Aufgabe 2.1** das Saugvermögen der TMP als Funktion des Drucks bei IM.
- Stellen Sie  $S(p)$  für die DSP und die TMP in einem Diagramm graphisch dar und diskutieren Sie Ihr Ergebnis.
- Schätzen Sie die Kraft ab, mit der die Glasglocke auf die Gummidichtung gedrückt wird.

In diesem Versuch soll das Saugvermögen der Turbomolekularpumpe (TMP) bestimmt werden. Da diese aber einige Minuten Vorlaufzeit benötigt und nicht erst bei sehr niedrigen Drücken eingeschalten werden soll, wird sie schon ca.  $5 \text{ min}$  nach Beginn der Messung bei  $T_3$  angeschalten und ist nach etwa  $1.5 \text{ min}$  im Normalbetrieb. Bis die TMP also bereit ist, wird der Druck bei  $T_3$  gemessen, ab dann beim Ionisationsvakuummeter  $IM$ . Es wird dann wieder der zeitliche Verlauf des Drucks dargestellt, aber nur bei  $IM$ . Zur Bestimmung des Saugvermögens wird es wieder geplottet und der Wert im einigermaßen konstanten Bereich schätzungsweise abgelesen. Schließlich werden die Saugvermögen der DSP und TMP zusammen dargestellt und verglichen. Schließlich wird noch die Kraft abgeschätzt, die auf die Glasglocke aufgrund des Vakuums in ihrem Inneren wirkt.

```
[14]: # Anfang 3:20, Ende: 10:15, IM Start: 8:05, IM Norm: 9:25 (T3 bei 0.00338)
druck_t3 = np.array([0.204, 0.196, 0.189, 0.180, 0.174, 0.168, 0.163, 0.158, 0.
↪153, 0.150, 0.147, 0.144,
                        0.141, 0.138, 0.136, 0.134, 0.132, 0.130, 0.128, 0.127, 0.
↪125, 0.124, 0.122, 0.121,
                        0.120, 0.119, 0.118, 0.117, 0.116, 0.114, 0.113, 0.112, 0.
↪111, 0.110, 0.109, 0.109,
                        0.108, 0.107, 0.106, 0.106, 0.104, 0.104, 0.103, 0.103, 0.
↪102, 0.101, 0.101, 0.100,
                        0.0996, 0.0987, 0.0982, 0.0977, 0.0972, 0.0966, 0.0962, 0.
↪0954, 0.0950, 0.0944, 0.0936, 0.0873,
                        0.0791, 0.0675, 0.0543, 0.0410, 0.0297, 0.0206, 0.0144, 0.
↪0104, 0.00773, 0.00576, 0.00478, 0.00404,
                        0.00359, 0.00338, 0.00319, 0.00306, 0.00296, 0.00287, 0.
↪00281, 0.00271, 0.00266, 0.00264, 0.00259, 0.00254])
# IM Messung: 9:25
druck_im = np.array([4.33, 4.08, 3.83, 3.64, 3.51, 3.41, 3.33, 3.26, 3.18, 3.
↪12, 3.06,
                        3.01, 2.97, 2.92, 2.89, 2.85, 2.81, 2.77, 2.75, 2.71, 2.
↪68, 2.66, 2.64,
```

```

        2.62, 2.60, 2.57, 2.56, 2.54, 2.52, 2.51, 2.50, 2.47, 2.
↪46, 2.44, 2.43,
        2.41, 2.40, 2.39, 2.38, 2.36, 2.36, 2.34, 2.33, 2.32, 2.
↪31, 2.29, 2.29,
        2.28, 2.26, 2.25, 2.25, 2.24, 2.23, 2.22, 2.21, 2.20, 2.
↪19, 2.18, 2.17,
        2.17, 2.17, 2.15, 2.15, 2.15, 1.80]) * 10**(-4)

t_14min = np.linspace(0, 138, 139) * 5

```

Verlauf des Drucks bei IM, nach der Vorlaufzeit von, in unserem Fall, etwa 1.5 *min* der TMP. Man sieht keinen großen Unterschied zwischen der normalen und der logarithmischen Darstellung.

```

[15]: fig4, ax4 = plt.subplots(2, sharex=True)
ax4[0].errorbar(t_14min[74:], druck_im, xerr=t_5min_std,
↪yerr=druck_im*druck_std, fmt="rx", label="$p(t)$ normal")
ax4[0].grid(), ax4[0].legend()

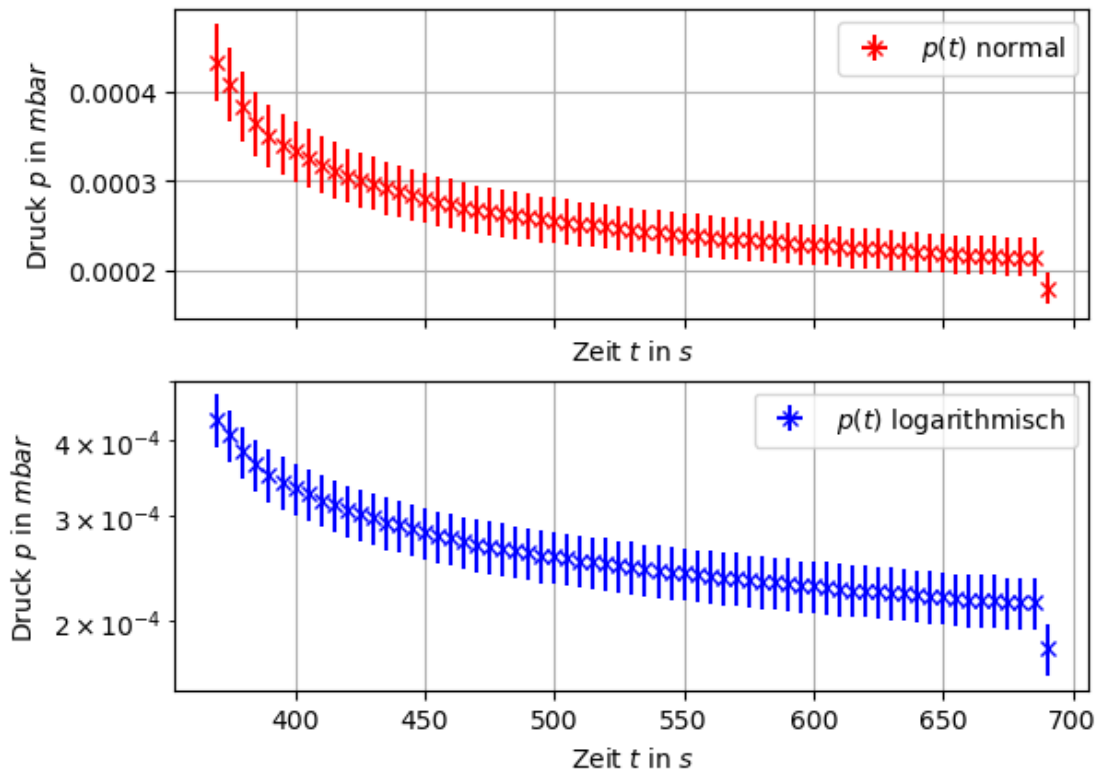
ax4[1].errorbar(t_14min[74:], druck_im, xerr=t_5min_std,
↪yerr=druck_im*druck_std, fmt="bx", label="$p(t)$ logarithmisch")
ax4[1].set_yscale("log")
ax4[1].grid(), ax4[1].legend()

fig4.suptitle("Verlauf des Drucks bei IM")

for ax in ax4:
    ax.set(xlabel="Zeit $t$ in $s$", ylabel="Druck $p$ in $mbar$")

```

### Verlauf des Drucks bei IM



```
[16]: saugvermögen4 = - ( ges_vol_schlauch * 10**3 / t_14min[74:] ) * np.log(druck_im_
    ↪ / druck_im[0])
saugvermögen4_std = 10**3 * (ges_vol_schlauch/t_14min[74:]) * np.sqrt( (np.
    ↪ log(druck_im/druck_im[0]) * t_5min_std / t_14min[74:] )**2 + np.ones(druck_im.
    ↪ size)*druck_std**2)

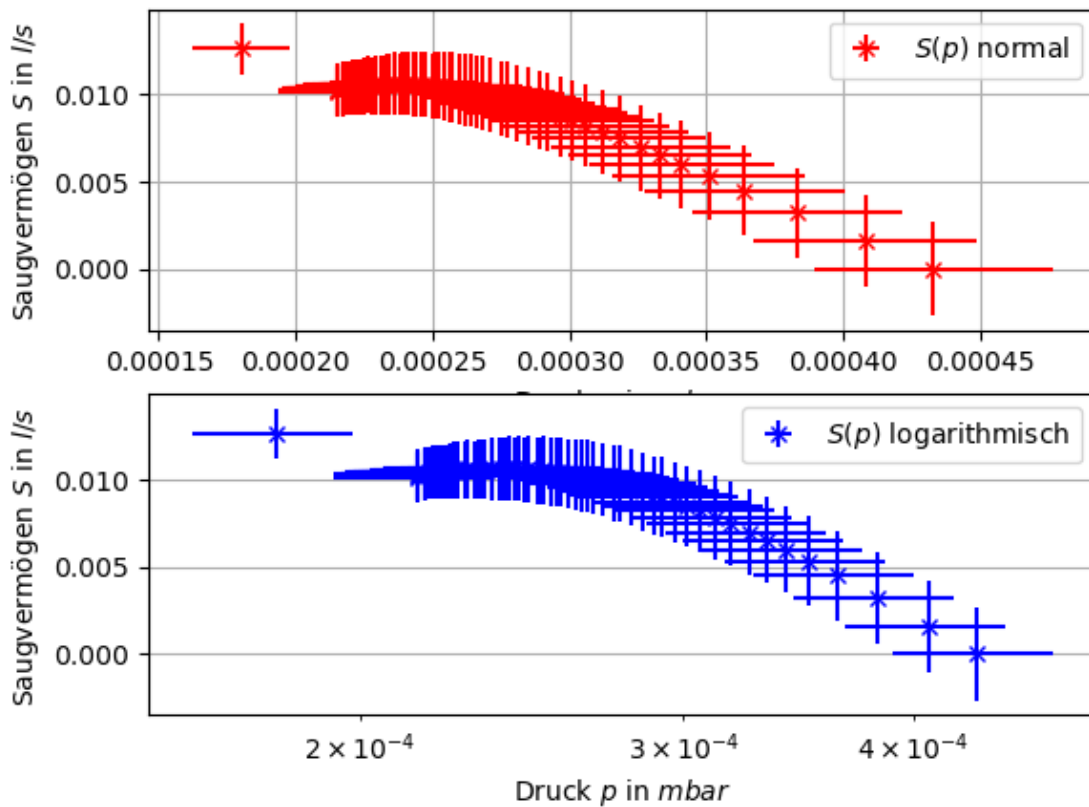
fig5, ax5 = plt.subplots(2, sharey=True)
ax5[0].errorbar(druck_im, saugvermögen4, xerr=druck_im*druck_std,
    ↪ yerr=saugvermögen4_std, fmt="rx", label="$S(p)$ normal")
ax5[0].grid(), ax5[0].legend()

ax5[1].errorbar(druck_im, saugvermögen4, xerr=druck_im*druck_std,
    ↪ yerr=saugvermögen4_std, fmt="bx", label="$S(p)$ logarithmisch")
ax5[1].set_xscale("log"), ax5[1].grid(), ax5[1].legend()

fig5.suptitle("Saugvermögen der TMP bei versch. Drücken")

for ax in ax5:
    ax.set(xlabel="Druck $p$ in $mbar$", ylabel="Saugvermögen $$$ in $l/s$")
```

### Saugvermögen der TMP bei versch. Drücken



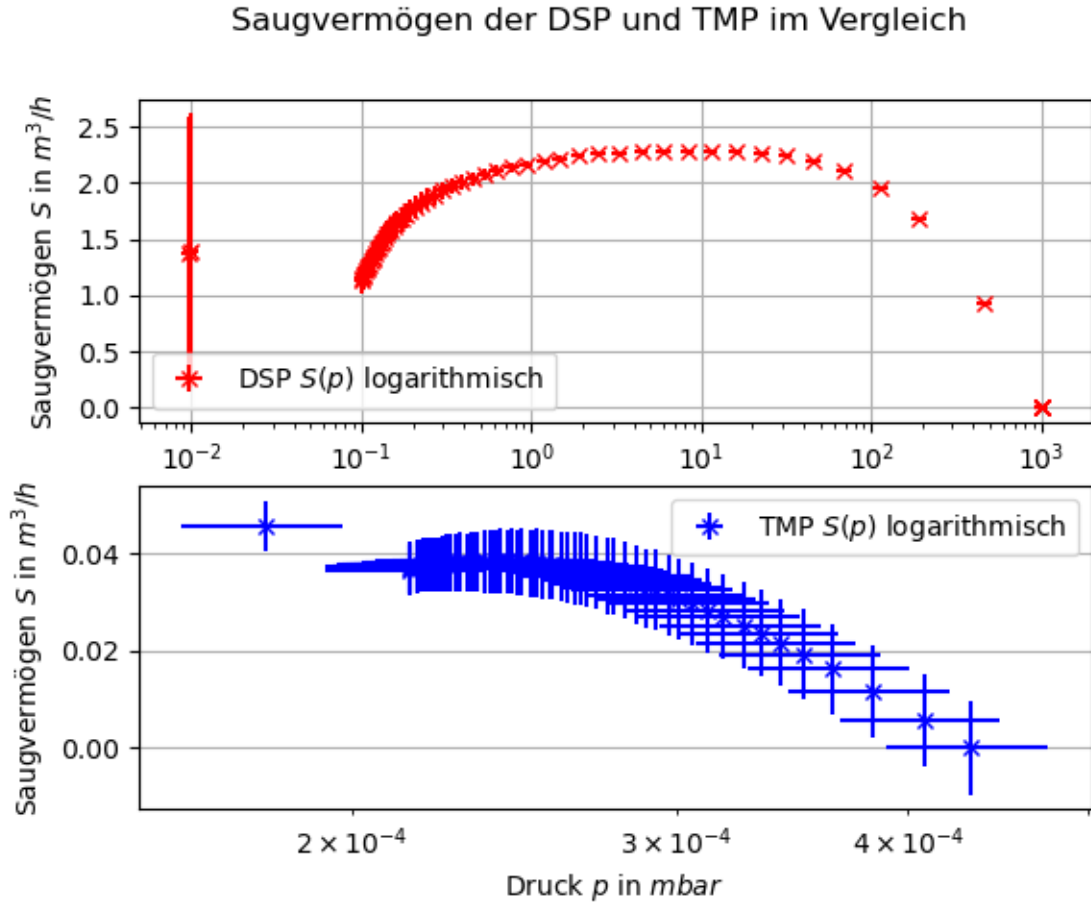
Der Plot sieht durch die Fehlerbalken nicht sehr schön aus, man kann aber erkennen, dass zwischen  $p \approx 2.15 \cdot 10^{-4} \text{ mbar}$  und  $p \approx 2.5 \cdot 10^{-4} \text{ mbar}$  das Saugvermögen der TMP relativ konstant bleibt. Allerdings liegt es in diesem konstanten Bereich in der Größenordnung  $10^{-2}$ , obwohl in der Vorbereitung  $30 \text{ l/s}$  gegeben waren, also etwa 3 Größenordnungen Unterschied.

```
[17]: fig6, ax6 = plt.subplots(2)

ax6[0].errorbar(druck_2_1_auf[1:], saugvermögen, xerr=druck_2_1_auf[1:
    ↪]*druck_std, yerr=saugvermögen_2_1_std, fmt="rx", label="DSP $S(p)$",
    ↪logarithmisch")
ax6[1].errorbar(druck_im, saugvermögen4*3.6, xerr=druck_im*druck_std,
    ↪yerr=saugvermögen4_std*3.6, fmt="bx", label="TMP $S(p)$ logarithmisch")
ax6[0].set_xscale("log")
ax6[1].set_xscale("log")
ax6[0].legend(), ax6[0].grid()
ax6[1].legend(), ax6[1].grid()
ax6[0].set(xlabel="Druck $p$ in $mbar$", ylabel="Saugvermögen $$$ in $m^3/h$")
ax6[1].set(xlabel="Druck $p$ in $mbar$", ylabel="Saugvermögen $$$ in $m^3/h$")
```

```
fig6.suptitle("Saugvermögen der DSP und TMP im Vergleich")
```

```
[17]: Text(0.5, 0.98, 'Saugvermögen der DSP und TMP im Vergleich')
```



Wie man in den vergangenen Aufgaben gesehen hat, ist das Saugvermögen der DSP mit  $S_{DSP} = (2.272 \pm 0.004) \frac{m^3}{h}$  relativ nah am erwarteten Wert von  $S_{DSP,exp} = 2.5 \frac{m^3}{h}$ . Das der TMP ist leider viel zu niedrig ausgefallen, in der Größenordnung  $10^{-2} \frac{l}{s}$ , während der erwartete Wert  $\geq 30 \frac{l}{s}$  beträgt. Die Verläufe der Saugvermögen weisen aber Ähnlichkeiten auf: Zunächst steigt es zu niedrigeren Drücken hin, bis über eine größere Druckspanne ein relativ konstanter Wert erreicht wird. Bei der DSP nimmt das Saugvermögen danach wieder stark ab, bei der TMP würde man wahrscheinlich ein ähnliches Verhalten zeigen, hätten wir noch länger gemessen. Beide Pumpen haben also ihre optimalen Drücke, bei der TMP sind diese aber sehr viel niedriger als bei der DSP, da sie besonders gut wenige Moleküle bewegen kann, daher auch der Name. Das Saugvermögen der TMP sollte eigentlich auch noch um etwa einen Faktor 4 größer sein, ist aber komischerweise um einiges niedriger.

Zur Kraftabschätzung nähern wir die Glasglocke als Zylinder, damit können wir die Kraft folgendermaßen abschätzen:  $F = A \cdot |\Delta p|$ . Der Druck außen sind natürlich die  $1000 \text{ mbar}$ , in der Glasglocke nehmen wir den letzten gemessenen Wert von  $1.8 \cdot 10^{-4} \text{ mbar}$  und die Fläche beträgt



$A = \pi \cdot (0.11\text{m})^2 \approx 0.038\text{m}^2$ . Wir erhalten als Kraft letztlich:

```
[18]: f_glocke = np.pi * .11**2 * np.absolute(1000 - 1.8 * 10**(-4)) * 10**2 # in_
      ↪Newton
      print(f"Kraftabschätzung: F = {f_glocke:.1f}")
```

Kraftabschätzung: F = 3801.3

Die Kraft, die auf die Glasglocke nach unten wirkt, beträgt  $F_{Glocke} \approx 3801.3\text{ N} \approx 3.8\text{ kN}$

### 3.3 Aufgabe 3: Experimente im Vakuum

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

- Für diese Aufgabe führen Sie einige einfache Experimente im Vakuum durch.
- Bearbeiten Sie hierzu die folgenden Aufgaben.

#### 3.3.1 Aufgabe 3.1: Statische Kalibration von T3

Führen Sie mit Hilfe des Referenzvolumens RV eine statische Kalibration des Vakuummeters T3 durch.

```
[19]: def linear_model(x,b,epsilon):
      return 100000*epsilon*x + b

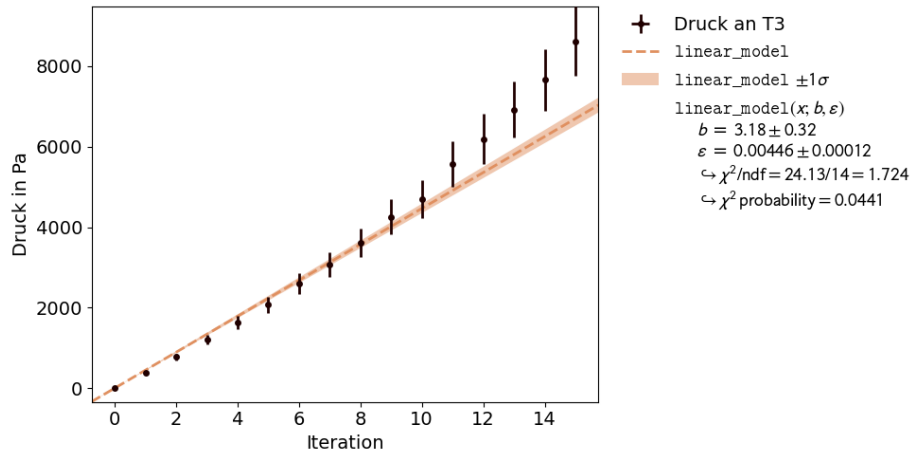
[20]: druck_t3 = np.array([3.19*10**(-2), 3.73, 7.78, 12.1, 16.3, 20.7, 26.0, 30.6,
      ↪36.1, 42.6, 47.0, 55.7, 61.9, 69.2, 76.6, 86.2])*100
      iter = 16

      xy_data = kafe2.XYContainer(np.arange(iter),druck_t3)
      xy_data.label = 'Druck an T3'

      fit = kafe2.XYFit(xy_data, model_function=linear_model)
      fit.add_error(axis='y', err_val=0.1, relative=True)
      fit.data_container.axis_labels = ('Iteration', 'Druck in Pa')
      fit.do_fit()

      plot=kafe2.Plot(fit)
      plot.plot()
      plt.show()

      print(f'Das rechnerisch bestimmte Epsilon beträgt {0.034 / 9.2}')
```



Das rechnerisch bestimmte Epsilon beträgt 0.003695652173913044

In diesem Versuch wird eine statische Kalibration des Wärmeleitvakuumeters T3 durchgeführt. Hierfür wird der Rezipient evakuiert und dann das Referenzvolumen damit verbunden. Danach wird diese Verbindung gekappt, das Referenzvolumen belüftet und die Belüftung wieder geschlossen. Dieser Prozess wird solange durchgeführt, bis im Rezipient Normaldruck herrscht.

Im oben stehenden Fit wurde der gemessene Druck in RV gegenüber der Anzahl an erfolgten Iterationen aufgetragen und dann durch einen Fit die Steigung bestimmt. Diese Steigung ist das Verdichtungsverhältnis und beträgt für die genommenen Messdaten  $\epsilon = 0.00446 \pm 0.00012$ . Das, mithilfe der Formel  $\epsilon = \frac{V_{RV}}{V_{RZ}}$  berechnete, beträgt für den vorliegenden Aufbau  $\epsilon = 0.0037$ . Ein Grund für die Abweichung könnte sein, dass die Apparatur nicht vollständig luftdicht ist, wodurch der Druck schneller steigt als erwartet.

Der Fit selbst entspricht mit  $\chi^2_{\text{probability}} = 0.0441$  zwar eigentlich nicht den Anforderungen, wird jedoch trotzdem akzeptiert. Änderungen in den Startwerten führen zu keiner Veränderung der bestimmten Werte. Ebenso ist eine Variation der Unsicherheiten nahezu Ergebnislos.

Außerdem lassen sich die Daten sehr gut durch eine Gerade nähern, man erkennt jedoch gerade bei hohen Drücken, dass sich ein nichtlinearer Fehler dazuaddiert. Dieser Fehler ist aber laut der Vorbereitung erwartet.

### 3.3.2 Aufgabe 3.2: Elektrische Durchschlagfestigkeit

- Bestimmen Sie die elektrische Durchschlagfestigkeit der KE als Funktion des Umgebungsdrucks in Luft.
- Diese Aufgabe kann nur an den Apparaturen 41 und 42 durchgeführt werden.

```
[21]: # Druck an T3
p = [1000, 600, 276, 145, 71.4, 35.0, 17.4, 8.01, 4.15, 2.01, 1.00, 0.51, 0.
     ↪ 266, 0.15, 0.09] # mbar
```

```

U = [319.5, 88, 80, 72, 61, 50, 41, 39, 36, 36, 33, 45, 48, 52, 47] # es fehlt
↪Faktor 10

p_rev = [5.2*10**(-4), 8.8*10**(-4), 68*10**(-4), 92*10**(-4), 1260*10**(-4)]
↪#mbar

U_rev = [251, 150, 72, 62, 56] # es fehlt Faktor 10

data_p = np.array(p + p_rev[::-1])
data_U = np.array(U + U_rev[::-1]) * 10

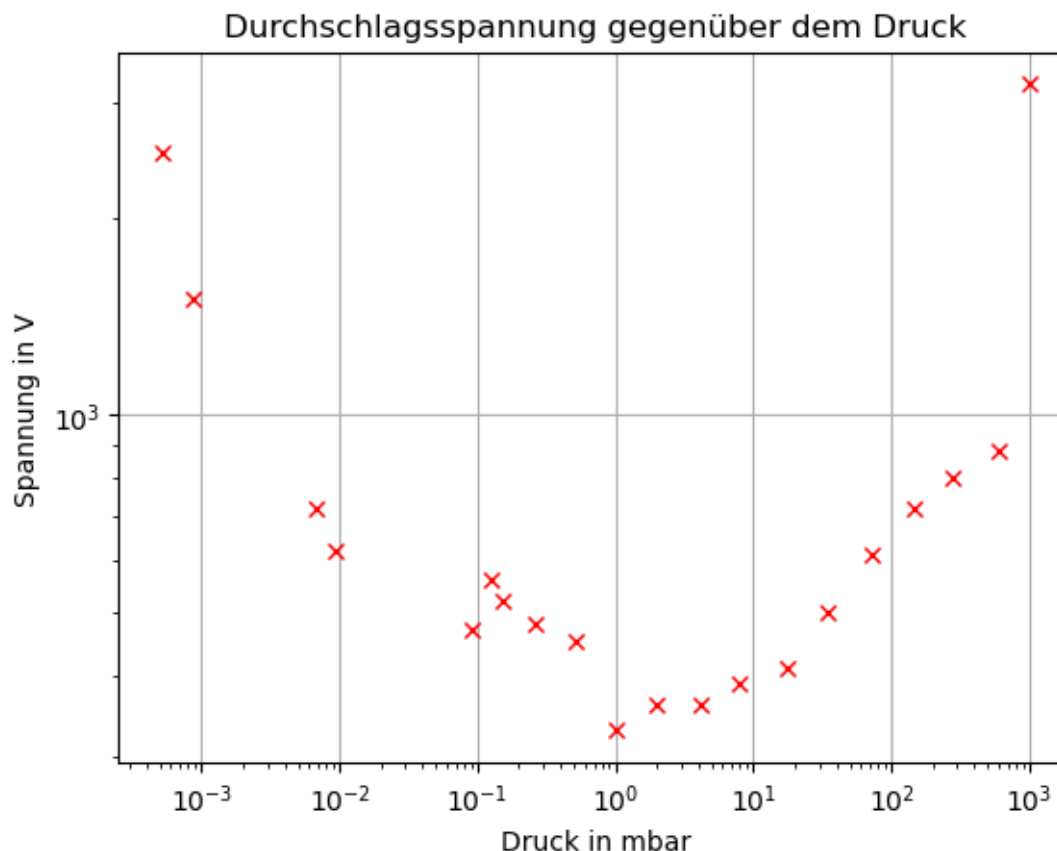
U_std = 30

fig = plt.figure()
ax = fig.add_subplot()

ax.plot(data_p, data_U, 'rx')
ax.set_xscale('log')
ax.set_yscale('log')

ax.grid()
ax.set_title('Durchschlagsspannung gegenüber dem Druck')
ax.set_xlabel('Druck in mbar')
ax.set_ylabel('Spannung in V')
plt.show()

```



In diesem Versuch wird für verschieden starke Vakua die minimale Spannung bestimmt, bei der ein Überschlag zwischen zwei Kugelelektroden im Rezipienten stattfindet.

Im oben gezeigten Diagramm ist die Durchschlagsfestigkeit gegenüber dem Druck im inneren der Kuppel aufgetragen. Da dieser Versuch in der vorhandenen Apparatur fehlte, wird hier mit den Daten einer anderen Gruppe gearbeitet. Die Durchschlagsfestigkeit bezieht sich hierbei auf die gemessene Spannung, die benötigt wird, um zwischen zwei Kugeln, die sich unter der Vakuumglocke befinden, einen Übersprung der Elektronen zu erzeugen.

Zu Beginn ist die mittlere freie Weglänge gering, weshalb sich die Elektronen aus der Anode nicht sehr weit bewegen können, bevor sie stoßen. Die Entladungen finden also erst bei sehr hohen Spannungen statt. Sinkt nun der Druck, steigt die freie Weglänge und die Elektronen können sich weiter bewegen, wodurch auch bei geringeren Spannungen ein Durschlag stattfindet. Unter einer bestimmten Schwelle muss jedoch wieder mehr Spannung aufgebracht werden, da nicht genug zu ionisierende Moleküle vorhanden sind.

Außerdem war zu beobachten, dass sich für einige Messungen kein Überschlag mehr erkennen lässt, sondern lediglich die Spannung nicht weiter ansteigt. Ebenso ist ab einem bestimmten Punkt der Druck nichtmehr genau einstellbar, da hier durch bedienen der Ventile und der Pumpen nur bedingt Dinge verändert werden können.

```
[22]: p = ufloat(1,0.1)

M_m = 28
f = 0.74
roh_fl = 807
N_A = 6.022 * 10**23
k_B = 1.38 * 10**-20
T = 293.15
V = M_m * f / (N_A * roh_fl)
r = (3*V/(4*np.pi))**(1/3)
sigma = np.pi * r**2
lamda = k_B*T/(sigma*p)

lamda
```

[22]: 0.2741750144179074+/-0.02741750144179075

Aus dem oben gezeigten Plot wird der Druck bestimmt, bei dem die benötigte Spannung am niedrigsten ist. Dieser Druck wird dann mithilfe von den unten aufgeführten Formeln aus der Vorbereitung umgerechnet:

$$\lambda = \frac{k_B T}{\sigma p}$$

$$\sigma = \pi \left( \frac{3}{4\pi} V \right)^{\frac{2}{3}}$$

$$V = \frac{M_m f}{N_A \rho_{fl}}$$

Hierbei meint  $M_m$  die molare Masse,  $\rho_{fl}$  die Dichte von flüssigem Stickstoff und  $f$  den Füllfaktor der dichtesten Kugelpackung. Insgesamt ergibt sich so eine Abschätzung des Abstandes der Kugelelektroden von  $\lambda = 0.274 \pm 0.0274 \text{ mm}$ .

---

### 3.3.3 Aufgabe 3.3: Aufdampfen von Indium

Dampfen Sie mit Hilfe einer Schablone bei zwei verschiedenen Drucken Indium auf eine Plexiglasplatte auf.

---

Es wird hier eine Plexiglasplatte, die sich im Rezipienten befindet mit Indium bedampft. Es werden zwei Durchführungen für verschiedenen Drücke unternommen.

In diesem Versuch zeigte sich das Ablenkvermögen der Luftmoleküle und man erkennt klar die unterschiedlichen Drücke. Bei höherem Druck sind mehr Luftmoleküle vorhanden, mit denen die aufsteigenden Indiumatome stoßen und so gestreut werden. Diese Streuung resultiert in einem

größeren Punkt auf der Plexiglasplatte. Der untere Punkt wurde somit bei hohem und der untere bei niedrigem Druck aufgedampft.

