

## Physikalisches Anfängerpraktikum II

Sommersemester 2023

Versuch 252

Tutor: Daniel Härter

### Aktivierung von Indium und von Silber mit thermischen Neutronen

## 1 Einleitung

### 1.1 Ziel des Versuchs

In diesem Experiment sollen wir uns mit den Methoden zur Messung der Halbwertszeit vertrauter machen sowie das Mechanismus der Aktivierung und des Zerfalls durch Neutronen besser kennenzulernen, indem wir die Halbwertszeiten von  $^{116}\text{In}$  sowie Isotopen  $^{108}\text{Ag}$ ,  $^{110}\text{Ag}$  bestimmen.

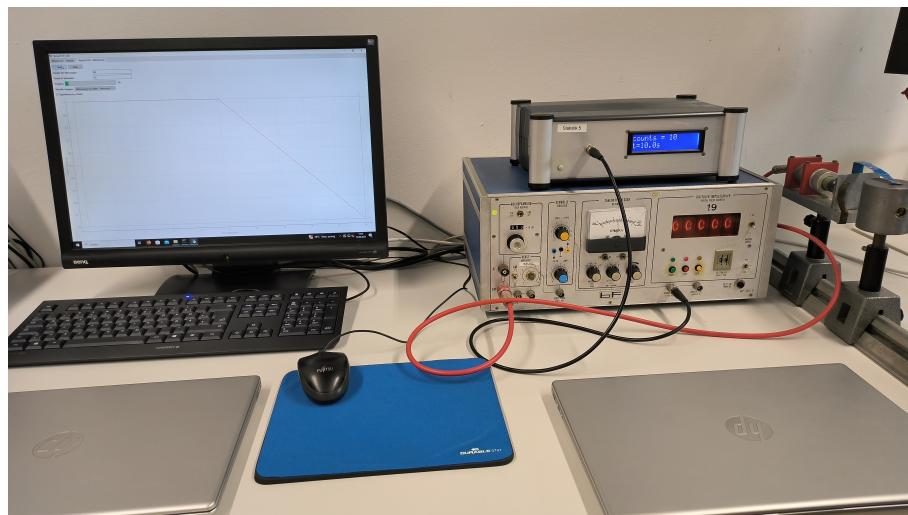


Abbildung 1: Versuchsaufbau

### 1.2 Radioaktiver Zerfall

Für die Herstellung einer radioaktiven Quelle werden stabile Isotope durch Kernreaktionen aktiviert. Besonders geeignet dafür sind Neutronen, da sie nicht von Coulomb-Wechselwirkung beeinflusst und daher vom Kern leicht eingefangen werden können. Nutzen diese Kernreaktion können wir eine Neutronenquelle herstellen und die langsame Neutrale für die Aktivierung des Zerfalls präparieren:



Die energiereichen Neutronen werden im Paraffinblock, der die Neutronenquelle umgibt, durch elastische Stöße mit den Wasserstoffkernen abgebremst, bis sie nahezu

thermische Energie erreicht haben. Viele Atomkerne haben eine starke Fähigkeit, langsame Neutronen einzufangen. Dabei entsteht ein Isotop des bestrahlten Elements mit der um eins erhöhten Massenzahl. Wenn dieses Isotop radioaktiv ist, ist dessen Erzeugung durch langsame Neutronen sogar die einfachste Möglichkeit.

Die Bestrahlung von stabilen  $^{115}\text{In}$  führt zur Bildung der  $\beta$ -Strahler  $^{116}\text{In}$  und  $^{116m}\text{In}$ , die zwei **Isomere** sind, mit gleicher Anzahl von Neutronen und Protonen aber verschiedenen Energiezuständen und Halbwertszeiten:  $^{115}\text{In}$  im Grundzustand und  $^{116m}\text{In}$  im metastabilen Zustand.

Bei der Aktivierung werden immer radioaktive Kerne produziert. Es gibt aber auch Zerfälle unter den vorhandenen radioaktiven Kernen, deren Anzahl proportional zueinander sind. Daher folgt die **Aktivität**  $A$  (die Anzahl der Zerfälle pro Sekunde) als Funktion der Bestrahlungsdauer  $t$  dem Gesetz:

$$A(t) = A_\infty(1 - \exp(-\lambda t)) \quad (2)$$

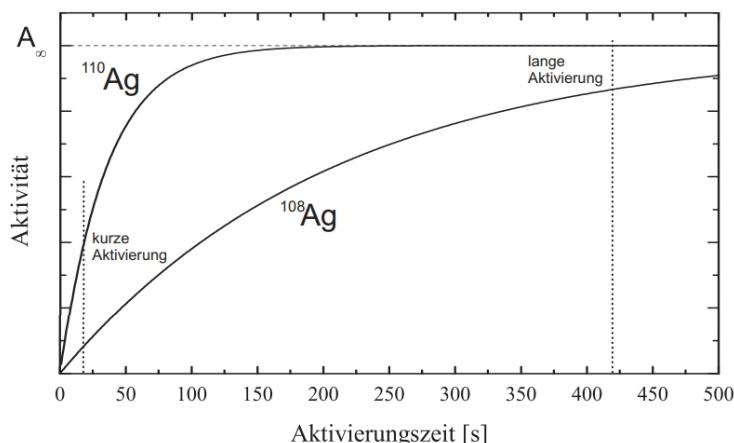


Abbildung 2: Aktivität von  $^{108}\text{Ag}$ ,  $^{110}\text{Ag}$  bei unterschiedlichen Aktivierungszeiten<sup>1</sup>

bis es das Gleichgewicht erreicht. Ohne Aktivierung nimmt die Aktivität allmählich zu nach dem radioaktiven Zerfallsgesetz:

$$A(t) = A_0 \exp(-\lambda t) \quad (3)$$

Die Halbwertszeit wird so definiert:

$$T_{1/2} = \frac{\ln 2}{\lambda} \quad (4)$$

Materie in der Natur besitzt nie nur ein Isotop, auch wenn es sich um eine reine Substanz eines bestimmten Elements handelt. Deswegen ist die Halbwertszeit der Substanz in der Tat die Gewichtsüberlagerung der Halbwertszeiten von mehreren Isotopen.

---

<sup>1</sup>Quelle: Dr. J.Wagner - Physikalisches Anfängerpraktikum - V. 1.1 B.Sc. Stand 10/2012

### 1.3 Das Geiger-Müller Zählrohr

Das Geiger-Müller-Zählrohr erkennt ionisierende Strahlung, wie  $\alpha$ -,  $\beta$ -,  $\gamma$ - und Röntgenstrahlen. Es besteht aus einem Metallzylinder mit einem Anodendraht, der mit einem Gasgemisch gefüllt ist und eine Spannung angelegt wird. Wegen äußerer Strahlung entstehen durch Ionisation des Gases freie Elektronen und positive Ionen, die durch das elektrische Feld beschleunigt, zum Anodendraht wandern und dort einen messbaren Strom erzeugen, der elektronisch verstärkt und gezählt wird. Der Primärstrom kann weiterer Sekundärstrom anregen, was zu einer Gasentladung führt, die jedoch bei entsprechender Detektoreinstellung nach etwa  $10^{-5}$  Sekunden wieder erlischt.

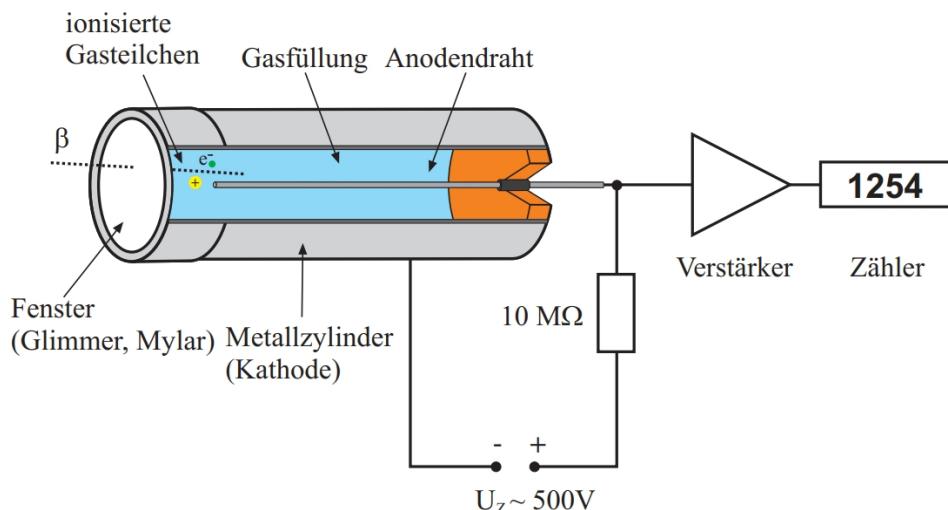


Abbildung 3: Aufbau von dem Geiger-Müller Zählrohr<sup>2</sup>

Ionisierende Strahlung im Zählrohr erzeugt primäre Ladungsträger proportional zur Energie. Bei niedriger Spannung erreicht nur ein Teil der Elektronen den Anoden draht, während der Rest durch Rekombination verloren geht. Mit steigender Spannung erreichen fast alle Elektronen die Anode, die Primärelektronen können selbst auch weitere ionisierte Elektronenlawine hervorrufen, was den Proportionalitätsbereich markiert, in dem der Strom zur Strahlungsenergie proportional ist. Weitere Spannungserhöhung führt zur Bildung von Sekundärelektronen und schließlich zum Plateaubereich, in dem jedes Teilchen ein gleich großes Signal erzeugt.

---

<sup>2</sup>Quelle: Dr. J.Wagner - Physikalisches Anfängerpraktikum - V. 1.1 B.Sc. Stand 10/2012

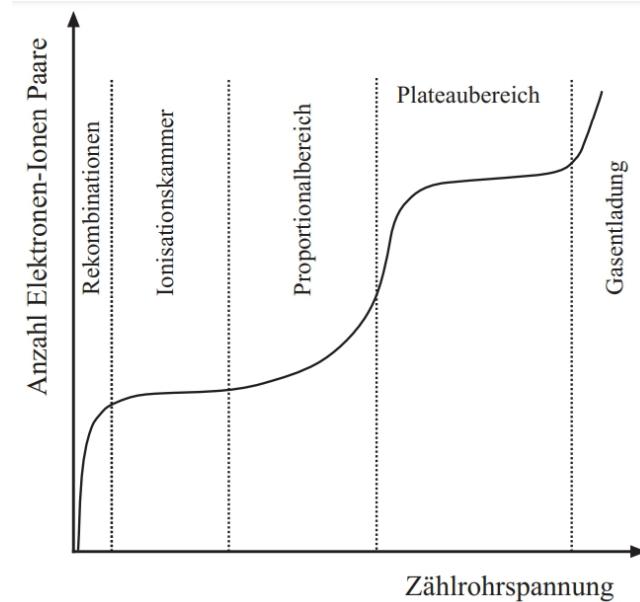


Abbildung 4: Kennlinie des Zählrohrs<sup>3</sup>

Im Plateaubereich des Zählrohrs ist der erzeugte Stromimpuls unabhängig von der Energie der Strahlung. Jedes eintreffende ionisierende Teilchen erzeugt das gleiche Ausgangssignal, das als Triggersignal für einen elektronischen Zähler dient.

## 2 Versuchsdurchführung

Versuchsaufbau, Versuchsdurchführung und Messprotokoll siehe folgende Seiten.

---

<sup>3</sup>Quelle: Dr. J.Wagner - Physikalisches Anfängerpraktikum - V. 1.1 B.Sc. Stand 10/2012

# DAP 2.2 Versuch 252: Aktivierung mit thermischen Neutronen

03. 06. 2024

Yuting Shi

14:00 ~ 17:00

Zulai Shi

## Messaufbau

- Geiger-Müller Zählrohr mit Betriebsgerät
- Externer Impulszähler
- Neutronenquelle
- Präparatehalterung
- Indium- und Silberbleche

## Aufgabe 1: Messung der Halbwertszeit von aktiviertem Silber

Es wurde am Betriebsgerät eine Zählrohrspannung 530 V ( $500 \text{ V} \sim 550 \text{ V}$ ) eingestellt.

① Zunächst sollte die Messung der Untergrundstrahlung durchgeführt werden, indem wir das Messprogramm "Versuch 252" benutzen. die Totzeit des Zählers auf 10 s, Anzahl der Messungen 50 (wurde die Untergrundzählrate über einen Zeitraum von etwa 8 Min zu messen)  $n=40, t=10s$

② Ag-Messu.: Das Ag-Blech (blaue Kennzeichnung)  $\geq 7 \text{ min}$  aktiviert und vor das Zählrohr gebracht, gesamte Messzeit: 400 s; Wir wiederholten die Aktivierung und die Messung mit der gleichen Probe 3 mal (Insges. 4 Aktivierungszyklen)

## Aufgabe 2: Messung der Halbwertszeit von aktiviertem Indium

$n=2t_5$ , Messintervall  $t=120s$ . Das In-Präparat ist bereits aktiviert sodass eine Messung direkt durchgeführt werden kann; Die gesamte Messung dauert 50 min

Häcker



Abbildung 5: Radioaktive Präparate, Neutronenquelle



Abbildung 6: Das Geiger-Müller-Zählrohr

### 3 Auswertung

Alle Text-Dateien werden in Jupyter-Notebook importiert und ausgewertet, dazu wurde zunächst eine Untergrundsmessung durchgeführt. Während der entsprechenden Messung wurde die Torzeit des Zählers auf  $T_{G1} = 10\text{s}$  festgelegt und eine Zählrohrspannung 530 V an das Geiger-Müller-Zählrohr angelegt. Da jedoch die vier Messwerte der Silbermessung hinzugefügt werden, muss die berechnete Hintergrundstrahlung um den Faktor 4 hochskaliert werden:<sup>4</sup>

$$\langle u \rangle_{Ag} = 15.2 \pm 0.9 \quad (5)$$

#### 3.1 Silber-Zerfall

Nun werden die Daten aus den ersten vier Messzyklen, die den Zerfall von Silber beschreiben, im Sinne der Matrizenaddition zusammengefasst. Die Aktivität (Anzahl der registrierten Zerfälle) wird in Abhängigkeit von der Zeit dargestellt. Durch das Zusammenfügen der Daten der einzelnen Messungen können statistische Fehler reduziert werden. Die Daten werden durch eine Anpassungsfunktion  $f$  interpoliert:

$$f(x) = A_1 \exp(-\lambda_1 x) + A_2 \exp(-\lambda_2 x) + \langle u \rangle_{Ag} \quad (6)$$

die einen exponentiellen Abfall der Zerfälle behauptet. Durch geeignete Wahl der Fitparameter ergiven sich:

$$A_1 = 99.12 \pm 8.92 \quad A_2 = 3.39 \pm 5.68 \quad (7)$$

$$\lambda_1 = 0.02256 \pm 0.00397 \quad \lambda_2 = 0.00070 \pm 0.00556 \quad (8)$$

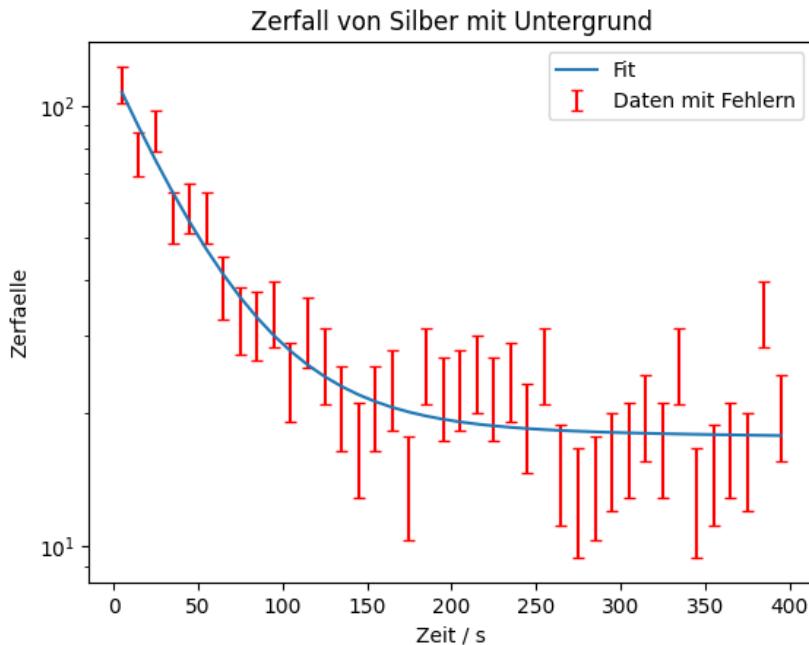


Abbildung 7: Anzahl der detektierten Zerfälle aufgetragen gegen die Zeit

---

<sup>4</sup>Python Code 1

Formel (5) wird verständlich, wenn man den Zerfallsprozess genauer betrachtet. Wie bereits in der Einleitung erwähnt, besteht natürlich vorkommendes Silber zu 51% aus  $^{107}\text{Ag}$  und zu 49% aus  $^{109}\text{Ag}$ . Dies führt bei der Aktivierung zu zwei verschiedenen Isotopen:  $^{108}\text{Ag}$  und  $^{110}\text{Ag}$ . Das erste Glied  $A_1 e^{-x\lambda_1}$  berücksichtigt den Zerfall von  $^{110}\text{Ag}$  und das zweite Glied  $A_2 e^{-x\lambda_2}$  berücksichtigt den Zerfall von  $^{108}\text{Ag}$ , wobei  $\lambda_1$ ,  $\lambda_2$  jeweils die Zerfallskonstante von  $^{110}\text{Ag}$  und  $^{108}\text{Ag}$  sind. Diese Intuition wird durch die sinnvolle Wahl der Anfangsbedingungen für die Optimierung gerechtfertigt.<sup>5</sup>

Des Weiteren soll die Güte des Fits diskutiert werden. Dazu werden die  $\chi^2$ -Summe, die reduzierte  $\chi^2$ -Summe sowie die Fitwahrscheinlichkeit berechnet. Im Allgemeinen gilt:

$$\chi^2 = \sum_i \left( \frac{\text{Funktionswert}_i - \text{Messwert}_i}{\text{Fehler}_i} \right)^2 \quad (9)$$

Außerdem gilt die reduzierte  $\chi^2$ -Summe:

$$\chi_{\text{red}}^2 = \frac{\chi^2}{\text{Freiheitsgrade}} \quad (10)$$

Der Freiheitsgrad lässt sich aus der Anzahl der Messwerte abzüglich der Zahl der Fit-Parameter ermitteln. Es ergibt sich:

$$\chi^2 = 39.44 \quad \chi_{\text{red}}^2 = 1.10 \quad (11)$$

Dabei wird gezeigt, wie gut das Modell zu den Daten passt. Der Wert von  $\chi_{\text{red}}^2$  nahe 1 deutet auf eine gute Anpassung hin. Außerdem möchten wir die Fitwahrscheinlichkeit  $P$  berechnen, dass man bei einer Wiederholungsmessung einen Chi-Quadrat-Wert ( $\chi^2$ ) erhält, der gleich oder größer ist als der Chi-Quadrat-Wert des aktuellen Fits:

$$P = 32.0\% \quad (12)$$

Außerdem kann das Verhältnis der  $^{110}\text{Ag}$ - zu  $^{108}\text{Ag}$ -Isotopen bestimmt werden, da sie mit  $A_1$  und  $A_2$  in Verbindung stehen.

$$P(^{110}\text{Ag}) = \frac{A_1}{A_1 + A_2} \quad (13)$$

$P(^{110}\text{Ag})$  ist der prozentuale Anteil der  $^{110}\text{Ag}$ -Isotope. Die obige Gleichung gilt unter der Bedingung, dass das Verhältnis von  $^{110}\text{Ag}$  zu  $^{108}\text{Ag}$  während der Messung konstant bleibt. Der Fehler wird gemäß der Standardfehlerfortpflanzung berechnet und das Ergebnis lautet:

$$P(^{110}\text{Ag}) = 97\% \quad P(^{108}\text{Ag}) = 3\% \quad (14)$$

Allerdings wurde der Fehler des Untergrunds in dieser Fitroutine nicht berücksichtigt. Da dies einen Einfluss auf den Fehler der Zerfallskonstanten hat, wurde der Fit zweimal wiederholt. Einmal, indem der Hintergrund um den  $1\sigma$ -Fehler nach oben (+)

---

<sup>5</sup>Python Code 2

korrigiert wurde, und einmal, indem der Hintergrund auf die gleiche Weise nach unten (-) korrigiert wurde:<sup>6</sup>

$$A_1^- = 99.15 \pm 8.87 \quad A_2^- = 4.27 \pm 5.36 \quad (15)$$

$$\lambda_1^- = 0.02253 \pm 0.00389 \quad \lambda_2^- = 0.00049 \pm 0.00416 \quad (16)$$

$$A_1^+ = 99.05 \pm 9.11 \quad A_2^+ = 2.60 \pm 6.52 \quad (17)$$

$$\lambda_1^+ = 0.02263 \pm 0.00417 \quad \lambda_2^+ = 0.00122 \pm 0.00842 \quad (18)$$

Wir erhalten zusätzlich zwei unterschiedliche Werte  $\lambda_i^-$  und  $\lambda_i^+$  für die jeweiligen Zerfallskonstanten. Der Fehler der jeweiligen Zerfallskonstanten wird so ermittelt, indem der Mittelwert der berechneten Differenzen quadratisch zum Fehler aus dem ersten Fit addiert wird:

$$(\Delta\lambda_i)_{\text{corr}} = \sqrt{(\Delta\lambda_i)^2 + \left( \frac{|\lambda_i - \lambda_i^+| + |\lambda_i - \lambda_i^-|}{2} \right)^2} \quad \text{für } i \in \{1, 2\} \quad (19)$$

Hierbei ist  $\Delta\lambda_i$  der Fehler der Zerfallskonstanten aus dem ersten Fit. Unter Verwendung der obigen Gleichung und Multiplikation mit den entsprechenden Einheiten kann das endgültige Ergebnis für die Zerfallskonstanten bestimmt werden:

Zerfallskonstante von  $^{110}\text{Ag}$ :  $\lambda_1 = (0.0226 \pm 0.0040) \text{ s}^{-1}$  (Fehler: 17.7 %)

Zerfallskonstante von  $^{108}\text{Ag}$ :  $\lambda_2 = (0.0007 \pm 0.0008) \text{ s}^{-1}$  (Fehler: 1.14)

Damit ergibt sich die Lebensdauer  $L$  sowie die Halbwertszeiten  $T$  aus Gl.(4) mit Fehlern aus Fehlerfortpflanzung der jeweiligen Insotopen:

$$L_1 = \frac{1}{\lambda_1} = (44.25 \pm 7.83) \text{ s} \quad T_1 = \frac{\ln(2)}{\lambda_1} = (30.67 \pm 5.43) \text{ s} \quad (20)$$

$$L_2 = \frac{1}{\lambda_2} = (1429 \pm 1628) \text{ s} \quad T_2 = \frac{\ln(2)}{\lambda_2} = (990 \pm 1128) \text{ s} \quad (21)$$

Vergleicht man die Werte mit den Literaturwerten der Halbwertszeiten von 24.6 s für  $^{110}\text{Ag}$  und 144.6 s für  $^{108}\text{Ag}$ , so stellt man eine Fehlerabweichung  $1.12\sigma$  von  $^{110}\text{Ag}$  und  $0.75\sigma$  fest, die zwar nicht signifikant, aber sehr merkwürdig ist wegen der großen absoluten Abweichung mit dem Literaturwert. Bereits bei der Berechnung des prozentualen Anteils tritt eine Abweichung auf, da der Anteil von  $^{110}\text{Ag}$  mit nur 3% deutlich niedriger ist als erwartet. Diese Anomalie wird später in Diskussion weiter beschäftigt.

## 3.2 Indium-Zerfall

Analog zur Silbermessung erfolgt die Auswertung der Indiummessung jedoch mit dem Untergrund für ein Zwei-Minuten-Intervall:

$$\langle u \rangle_{120} = 45.6 \pm 2.82 \quad (22)$$

---

<sup>6</sup>Python Code 3

Zur Interpolation wird eine einfache Exponentialfunktion benutzt:

$$f(x) = A \exp(-kx) + \langle u \rangle_{120} \quad (23)$$

und die Fit-Parameter lauten:

$$A = 178 \pm 9 \quad k = 0.00025 \pm 0.00389 \quad (24)$$

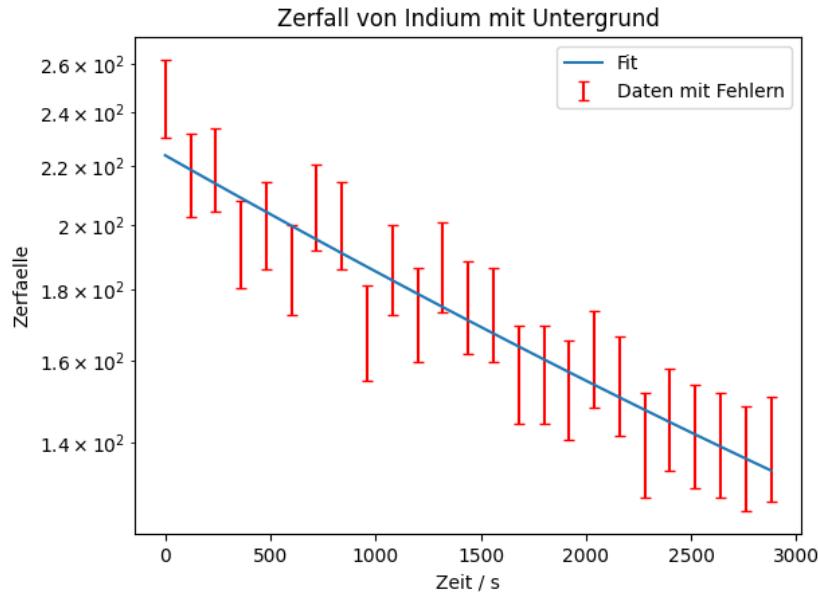


Abbildung 8: Anzahl der detektierten Zerfälle aufgetragen gegen die Zeit

Die Chi-Quadrat sowie Fitwahrscheinlichkeit werden ausgerechnet:

$$\chi^2 = 10.23 \quad \chi^2_{\text{red}} = 0.49 \quad P = 98\% \quad (25)$$

Mit großer Wahrscheinlichkeit dass der Chi-Quadrat-Wert größer wird, was uns eine schlechte Anpassung liefert. Der Grund lässt sich darauf zurückführen, dass es eine starke Fluktuation der Daten gibt (siehe Abb. 8) und die Messung deswegen statistisch schlecht korreliert ist.

Ganz Analog wie die erste Aufgabe soll der Fehler des Untergrunds ins Betracht gezogen werden, und es ergibt sich:

$$A^- = 181 \pm 9 \quad k^- = 0.00024 \pm 0.00389 \quad (26)$$

$$A^+ = 175 \pm 9 \quad k^+ = 0.00024 \pm 0.00390 \quad (27)$$

Benutzt man die gleichen Argumentationen wie Aufgabe 1 um die Zerfallskonstante, die Halbwertszeit sowie die Lebensdauer auszuwerten:

$$k = (0.00025 \pm 0.00533) \text{ s}^{-1} \quad (28)$$

$$L = (4000 \pm 2442) \text{ s} \quad (29)$$

$$T = (2772 \pm 1540) \text{ s} \quad (30)$$

Mit einer Fehlerabweichung der Halbwertszeiten von  $0.56\sigma$  von dem Literaturwerten  $3240\text{ s}$  ist die Abweichung nicht signifikant, zeigt jedoch gleich wie vorher, eine großen Fehler gegen sich selbst.

## 4 Zusammenfassung und Diskussion

In diesem Experiment wurde die Halbwertszeit von  $^{108}\text{Ag}$ ,  $^{110}\text{Ag}$  und  $^{116}\text{In}$  gemessen, nachdem sie durch eine Neutronenquelle aktiviert wurden. Um eine Verfälschung der Messwerte durch Hintergrundrauschen aus anderen Quellen zu minimieren, wurde zuerst die Hintergrundsstrahlung gemessen und anschließend die jeweiligen Proben vor den Detektor platziert und eine Zerfallsmessung durchgeführt. Für Silber wurde eine Torzeit von  $t_T = 10\text{ s}$  gewählt und vier Messzyklen mit jeweils 40 Einzelmessungen durchgeführt. Für Indium wurde eine Torzeit auf  $t_T = 120\text{ s}$  eingestellt und eine Messreihe mit 25 Einzelmessungen durchgeführt. Die erhaltenen Daten wurden in Auswertung gegen die Zeit in einem Diagramm aufgetragen und durch eine Exponentialfunktion angepasst (Abbildung 7 und Abbildung 8 entsprechend).

Die verwendete Anpassungsfunktion kann wie folgt verallgemeinert werden:  
Betrachtet man eine Messung von Strahlung, die von  $N$  unabhängigen radioaktiven Isotopen emittiert wird, gestört durch einen Hintergrund von  $\langle u \rangle$ . Die physikalische Natur des Zerfalls rechtfertigt die folgende Anpassungsfunktion:

$$f(x) = \langle u \rangle + \sum_{i=1}^N A_i e^{-\lambda_i x} \quad (31)$$

Hierbei sind  $A_i$  und  $\lambda_i$  die Anpassungsparameter. Diese Gleichung beschreibt die Überlagerung der von den einzelnen Quellen emittierten Strahlung und des Hintergrunds. Eine kluge Wahl der Anfangsbedingungen der Optimierung stellt sicher, dass die Parameter zu den jeweiligen Intensitäten ( $A_i$ ) und Zerfallskonstanten ( $\lambda_i$ ) der Isotope konvergieren, sodass wir die Anpassungsparameter mit diesen physikalischen Größen identifizieren können.

Als Ergebnis bekommen wir die Halbwertszeiten von  $^{108}\text{Ag}$   $T_1 = (30.67 \pm 5.43)\text{ s}$ ,  $^{110}\text{Ag}$   $T_2 = (990 \pm 1128)\text{ s}$ , wobei sich die Halbwertszeit von  $^{110}\text{Ag}$   $T_2 = (990 \pm 1128)\text{ s}$  betragsmäßig signifikant von dem Literaturwerten abweicht. Graphisch lässt sich erkennen, dass im Laufe der Zeit eine immer schwächere Korrelation vorliegt, was auf die kurze Aktivierungszeit zurückzuführen ist. Der Zerfallprozess findet so schnell statt, dass ein charakteristischer exponentieller Abfall nicht mehr erkennbar und stattdessen nur noch der Grundrauschen zu sehen ist. Auf der anderen Seite ist der Anteil von  $^{110}\text{Ag}$  so gering (nach Gleichung (14) nur 3 %), dass der Term  $A_2 \exp(-\lambda_2 x)$  in der Anpassungsfunktion fast keine Rolle mehr spielt und  $^{110}\text{Ag}$  dominiert, was die Messung von  $^{110}\text{Ag}$  beeinträchtigt.

Eine andere Erläuterung für die Abweichung könnte die unvollständige Hintergrundsmessung sein, da wir es leider nicht schaffen, während der Messung alle möglichen

radioaktiven Quellen zu entfernen. Da das Labor selbst ein Lagerraum für radioaktive Stoffe ist, verändern die Versuchsteams die Radioaktivität in der Umgebung, indem sie von Zeit zu Zeit ein- und ausgehen, um radioaktive Stoffe zu entnehmen. Dies führt zu großen Schwankungen und Instabilität der Messungen.

Für die Halbwertszeit von Indium ist  $T = (2772 \pm 1540)$ , die nicht signifikant von dem Literaturwert abweicht. Man sieht, dass die Zerfallskonstante und der Verlauf deswegen fast linear ist. Als Result ergibt sich eine viel größere Halbwertszeit vergleichbar mit dem Silber. Es ist jedoch zu erkennen, dass eine Fitwahrscheinlichkeit von 98% und die reduzierte  $\chi^2$ -Summe von 0.49 für Indium berechnet wird. Das bedeutet, dass die  $\chi^2_{\text{red}}$  bereits deutlich kleiner als 1 ist, was darauf hindeutet, dass es mit großer Wahrscheinlichkeit ist, dass wir bei einer Wiederholungsmessung einen größeren  $\chi^2$ -Wert erhalten werden. Die Situation könnte darauf zurückzuführen sein, dass wir die Messungen für Indium nicht viermal durchgeführt haben, um sie zu mitteln. Daher ist die Messung anfälliger für statistische Fehler.

## 5 Anhang

# Python Code Versuch 252

June 3, 2024

## 0.0.1 PAP 2.2 Versuch 252: Aktivierung von Indium und Silber mit den thermischen Neutronen

Yuting Shi

```
[42]: import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
```

Python Code 1: Untergrundsmessung

```
[43]: unterg =np.loadtxt('C:\\\\Users\\\\shiy0\\\\OneDrive\\\\Dokumente\\\\PAP 2.2\\\\Versuch_\\\\
\u252\\\\Shi 252\\\\untergrund.txt', delimiter=',', skiprows= 4, unpack=True,\\\\
\u252\\\\usecols=[1]) #nur die 2. Spalte auslesen
```

Wir benötigen im Folgenden den Mittelwert der Untergrundrate und dessen Fehler. Da wir später die vier Messreihen addieren, muss der Untergrund um den Faktor vier erhöht werden.

```
[44]: mittelw_unterg=np.mean(4*unterg)
fehler_unterg=np.std(4*unterg)/np.sqrt(len(unterg))
print('Mittelwert:', mittelw_unterg, 'Fehler:', fehler_unterg)
```

Mittelwert: 15.2 Fehler: 0.9397872099576584

Python Code 2: Silbermessung

```
[45]: #Importieren der Daten
n1=np.loadtxt('C:\\\\Users\\\\shiy0\\\\OneDrive\\\\Dokumente\\\\PAP 2.2\\\\Versuch 252\\\\Shi_\\\\
\u252\\\\silber1.txt', delimiter=',', skiprows= 4, unpack=True, usecols=[1])
n2=np.loadtxt('C:\\\\Users\\\\shiy0\\\\OneDrive\\\\Dokumente\\\\PAP 2.2\\\\Versuch 252\\\\Shi_\\\\
\u252\\\\silber2.txt', delimiter=',', skiprows= 4, unpack=True, usecols=[1])
n3=np.loadtxt('C:\\\\Users\\\\shiy0\\\\OneDrive\\\\Dokumente\\\\PAP 2.2\\\\Versuch 252\\\\Shi_\\\\
\u252\\\\silber3.txt', delimiter=',', skiprows= 4, unpack=True, usecols=[1])
n4=np.loadtxt('C:\\\\Users\\\\shiy0\\\\OneDrive\\\\Dokumente\\\\PAP 2.2\\\\Versuch 252\\\\Shi_\\\\
\u252\\\\silber4.txt', delimiter=',', skiprows= 4, unpack=True, usecols=[1])

N=n1+n2+n3+n4
Fehler_N=np.sqrt(N)
t=np.arange(5,405,10)
```

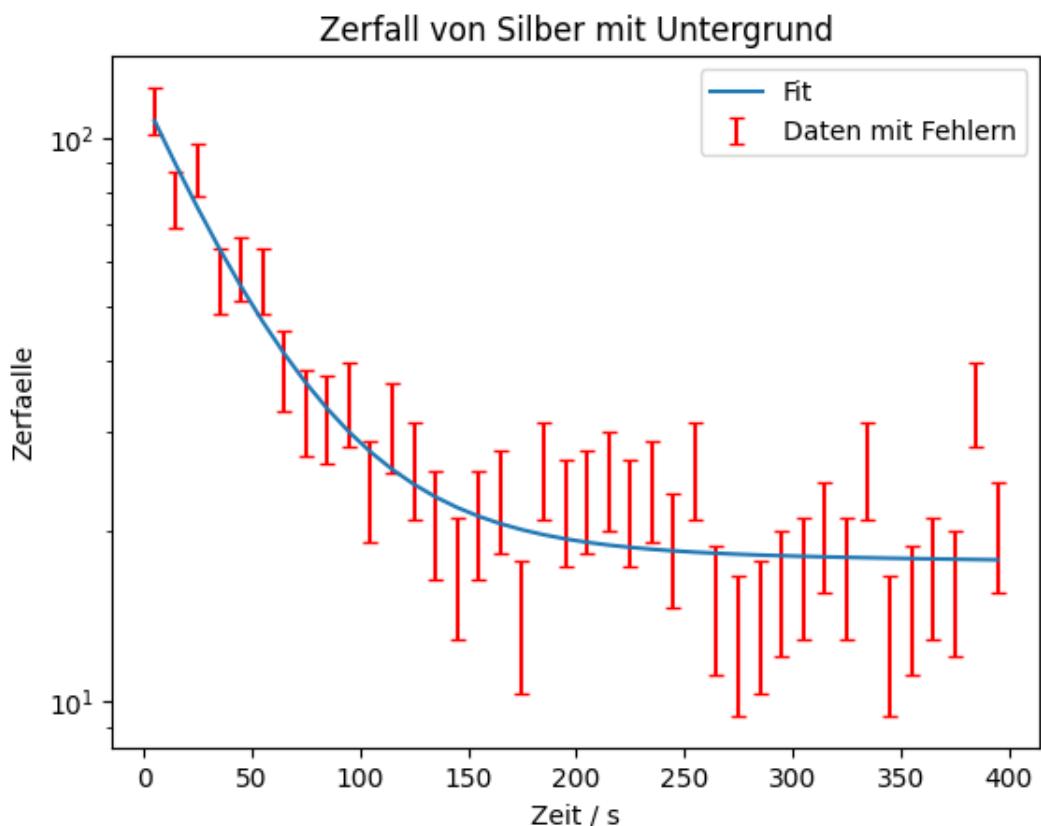
```

y0=mittelw_unterg #Untergrund
def fit_func(x, A1,l1,A2,l2):
    return A1*np.exp(-x*l1) + A2*np.exp(-x*l2) + y0
from scipy.optimize import curve_fit
popt, pcov=curve_fit(fit_func,t,N, p0=[500,0.02,50,0.001], sigma=Fehler_N,
                     absolute_sigma=True)

plt.errorbar(t,N, Fehler_N, fmt='None', ecolor='r', capsize=3, marker='s',
             mec='black', mfc='black', ms='2', label='Daten mit Fehlern')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Silber mit Untergrund')
plt.yscale('log')
plt.plot(t,fit_func(t,*popt), label='Fit')
plt.legend()

```

[45]: <matplotlib.legend.Legend at 0x2188711f5f0>



```
[46]: print("A1=",popt[0], ", Standardfehler=", np.sqrt(pcov[0][0]))
print("l1=",popt[1], ", Standardfehler=", np.sqrt(pcov[1][1]))
print("A2=",popt[2], ", Standardfehler=", np.sqrt(pcov[2][2]))
print("l2=",popt[3], ", Standardfehler=", np.sqrt(pcov[3][3]))
```

A1= 99.12296350656041 , Standardfehler= 8.921627902790485  
l1= 0.02255891416658196 , Standardfehler= 0.003972891073473391  
A2= 3.384638976099674 , Standardfehler= 5.67524557636918  
l2= 0.0006974867680178984 , Standardfehler= 0.005564553126057568

Güte des Fits

```
[47]: chi2_=np.sum((fit_func(t,*popt)-N)**2/Fehler_N**2)
dof=len(N)-4 #dof: degrees of freedom, Freiheitsgrad
chi2_red=chi2_/dof
print("chi2=", chi2_)
print("chi2_red=",chi2_red)
```

chi2= 39.437623258143084  
chi2\_red= 1.095489534948419

```
[48]: from scipy.stats import chi2
prob=round(1-chi2.cdf(chi2_,dof),2)*100
print("Wahrscheinlichkeit=", prob, "%")
```

Wahrscheinlichkeit= 32.0 %

Python Code 3: Wiederholung der Fits mit Berücksichtigung der Fehler des Untergrunds

```
[49]: #Korrektur nach unten
unter_unten=mittelw_unterg-fehler_unterg
y0_minus=unter_unten
print('korrigierter Untergrund unten', unter_unten)
def fit_func_minus(x, A1,l1_minus,A2,l2_minus):
    return A1*np.exp(-x*l1_minus) + A2*np.exp(-x*l2_minus) + y0_minus

popt, pcov=curve_fit(fit_func_minus,t,N, p0=[500,0.02,50,0.001],  

                     sigma=Fehler_N, absolute_sigma=True)

plt.errorbar(t,N, Fehler_N, fmt='None', ecolor='r', capsize=3, marker='s',  

             mec='black', mfc='black', ms='2', label='Daten mit Fehlern')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Silber mit korrigierter Untergrund')
plt.yscale('log')
plt.plot(t,fit_func_minus(t,*popt), label='Fit')
plt.legend()

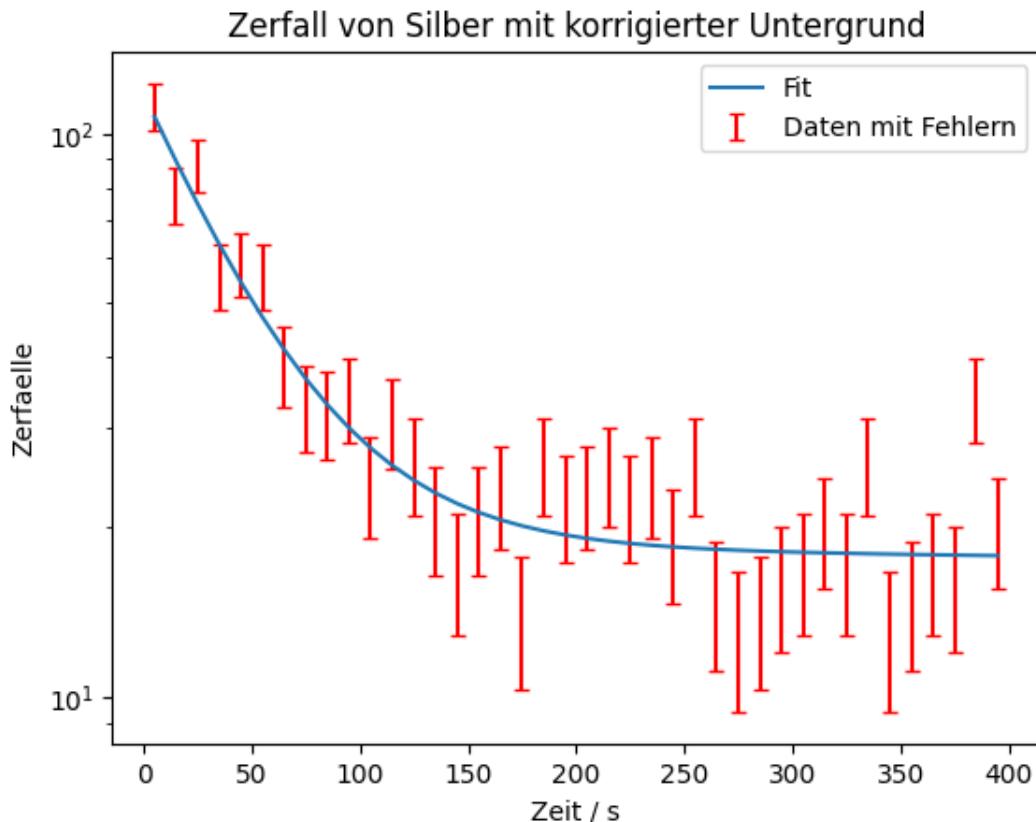
print("A1_u=",popt[0], ", Standardfehler=", np.sqrt(pcov[0][0]))
```

```

print("l1_u=",popt[1], " Standardfehler=", np.sqrt(pcov[1][1]))
print("A2_u=",popt[2], " Standardfehler=", np.sqrt(pcov[2][2]))
print("l2_u=",popt[3], " Standardfehler=", np.sqrt(pcov[3][3]))

```

korrigierter Untergrund unten 14.26021279004234  
 $A_1 = 99.1492980249903$ , Standardfehler= 8.865176788194109  
 $l_1 = 0.02252871001211929$ , Standardfehler= 0.0038947029506337023  
 $A_2 = 4.2641293996479295$ , Standardfehler= 5.363231731153289  
 $l_2 = 0.0004853589050744501$ , Standardfehler= 0.004157030668576867



```

[50]: #Korrektur nach oben
unter_oben=mittelw_unterg + fehler_unterg
y0_plus=unter_oben
print('korrigierter Untergrund unten', unter_oben)
def fit_func_plus(x, A1,l1_plus,A2,l2_plus):
    return A1*np.exp(-x*l1_plus) + A2*np.exp(-x*l2_plus) + y0_plus

popt2, pcov2=curve_fit(fit_func_plus,t,N, p0=[500,0.02,50,0.001],sigma=True)

```

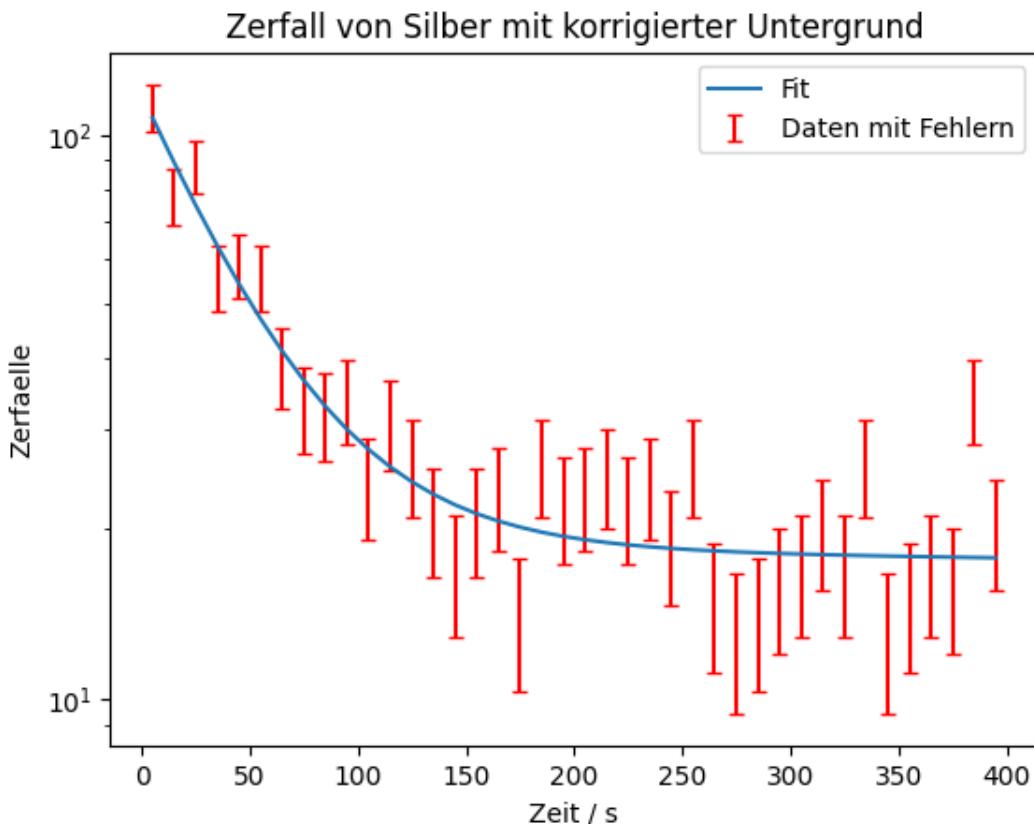
```

plt.errorbar(t,N, Fehler_N, fmt='None', ecolor='r', capsize=3, marker='s', mew=1, mec='black', mfc='black', ms=2, label='Daten mit Fehlern')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Silber mit korrigierter Untergrund')
plt.yscale('log')
plt.plot(t,fit_func_plus(t,*popt2), label='Fit')
plt.legend()

print("A1_o=",popt2[0], ", Standardfehler=", np.sqrt(pcov2[0][0]))
print("l1_o=",popt2[1], ", Standardfehler=", np.sqrt(pcov2[1][1]))
print("A2_o=",popt2[2], ", Standardfehler=", np.sqrt(pcov2[2][2]))
print("l2_o=",popt2[3], ", Standardfehler=", np.sqrt(pcov2[3][3]))

```

korrigierter Untergrund unten 16.139787209957657  
A1\_o= 99.04856423749831 , Standardfehler= 9.11195869389798  
l1\_o= 0.022630535724347475 , Standardfehler= 0.004174799642077707  
A2\_o= 2.5968404543177153 , Standardfehler= 6.517077295970586  
l2\_o= 0.0012161918962943807 , Standardfehler= 0.008415912682107816



#### Python Code 4: Zerfall vom In

```
[57]: mittelw_unterg=np.mean(12*unterg)
fehler_unterg=np.std(12*unterg)/np.sqrt(len(unterg))
print('Mittelwert:', mittelw_unterg, 'Fehler:', fehler_unterg)

#Importieren der Daten
n1=np.loadtxt('C:\\\\Users\\\\shiy0\\\\OneDrive\\\\Dokumente\\\\PAP 2.2\\\\Versuch 252\\\\Shi\\\\
    ↪252\\\\indium.txt', delimiter=',', skiprows= 4, unpack=True, usecols=[1])
N=n1
Fehler_N=np.sqrt(N)
t=np.arange(0,3000,120)      #Zeitintervall 120s

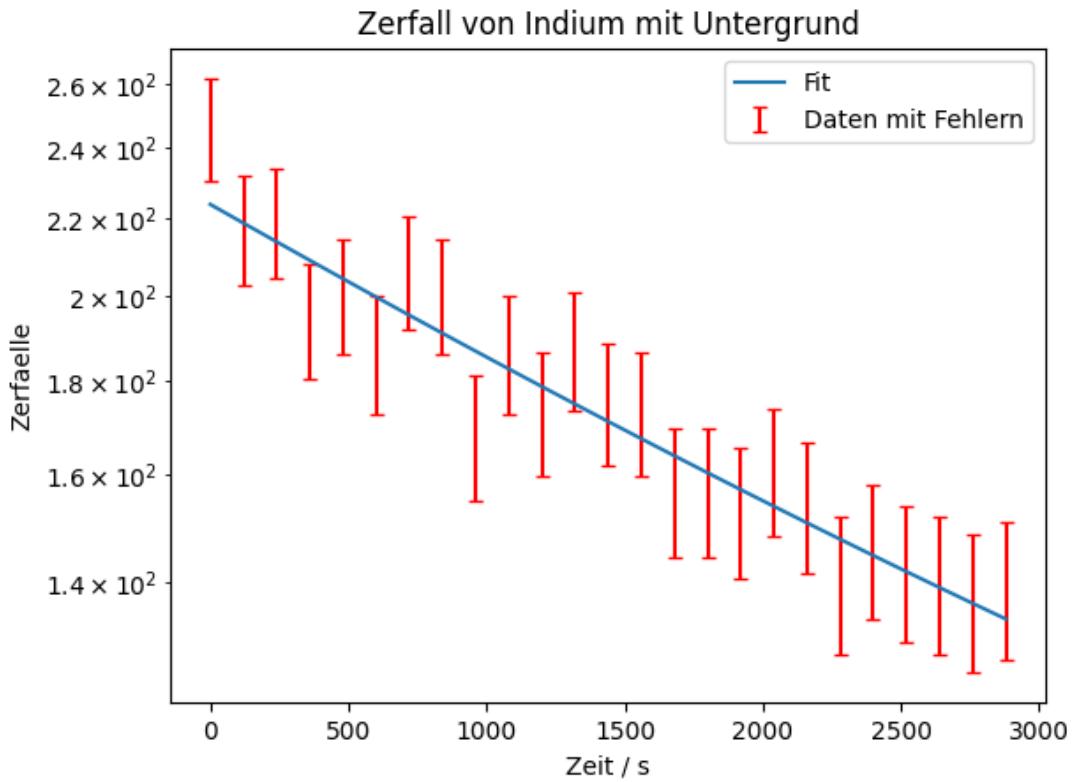
y0=mittelw_unterg #Untergrund
def fit_func2(x,A,k):
    return A*np.exp(-k*x) + y0

popt3, pcov3=curve_fit(fit_func2,t,N, p0=[700, 0.01], sigma=Fehler_N, u
    ↪absolute_sigma=True)

plt.errorbar(t,N, Fehler_N, fmt='None', ecolor='r', capsize=3, marker='s', u
    ↪mec='black', mfc='black', ms='2', label='Daten mit Fehlern')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Indium mit Untergrund')
plt.yscale('log')
plt.plot(t,fit_func2(t,*popt3), label='Fit')
plt.legend()

print("A=",popt3[0], ", Standardfehler=", np.sqrt(pcov[0][0]))
print("k=",popt3[1], ", Standardfehler=", np.sqrt(pcov[1][1]))
```

Mittelwert: 45.6 Fehler: 2.8193616298729753  
A= 178.24277624630034 , Standardfehler= 8.865176788194109  
k= 0.00024417010740752806 , Standardfehler= 0.0038947029506337023



```
[58]: chi2_=np.sum((fit_func2(t,*popt3)-N)**2/Fehler_N**2)
dof=len(N)-4 #dof: degrees of freedom, Freiheitsgrad
chi2_red=chi2_/dof
print("chi2=", chi2_)
print("chi2_red=",chi2_red)
prob=round(1-chi2_.cdf(chi2_,dof),2)*100
print("Wahrscheinlichkeit=", prob, "%")
```

```
chi2= 10.232802409300081
chi2_red= 0.4872763052047658
Wahrscheinlichkeit= 98.0 %
```

```
[72]: y0_unten=mittelw_unterg-fehler_unterg
def fit_func2_unten(x,A,k):
    return A*np.exp(-k*x) + y0_unten

popt3, pcov3=curve_fit(fit_func2_unten,t,N, p0=[700, 0.01], sigma=Fehler_N, u
↪absolute_sigma=True)

plt.errorbar(t,N, Fehler_N, fmt='None', ecolor='r', capsize=3, marker='s', u
↪mec='black', mfc='black', ms='2', label='Daten mit Fehlern')
```

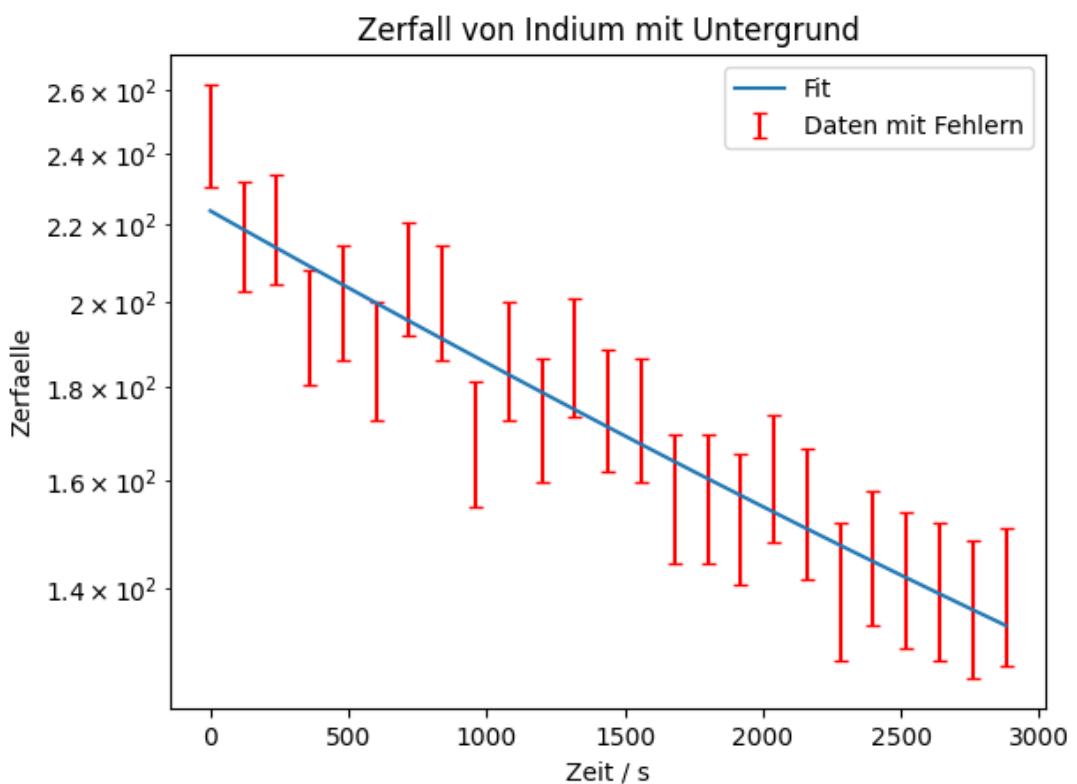
```

plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Indium mit Untergrund')
plt.yscale('log')
plt.plot(t,fit_func2_unten(t,*popt3), label='Fit')
plt.legend()

print("A_u=",popt3[0], ", Standardfehler=", np.sqrt(pcov[0][0]))
print("k_u=",popt3[1], ", Standardfehler=", np.sqrt(pcov[1][1]))

```

A\_u= 180.92775222246783 , Standardfehler= 8.865176788194109  
k\_u= 0.0002387800571678792 , Standardfehler= 0.0038947029506337023



```
[73]: y0_oben=mittelw_unterg+fehler_unterg
def fit_func2_oben(x,A,k):
    return A*np.exp(-k*x) + y0_oben

popt3, pcov3=curve_fit(fit_func2_oben,t,N, p0=[700, 0.01], sigma=Fehler_N,
absolute_sigma=True)
```

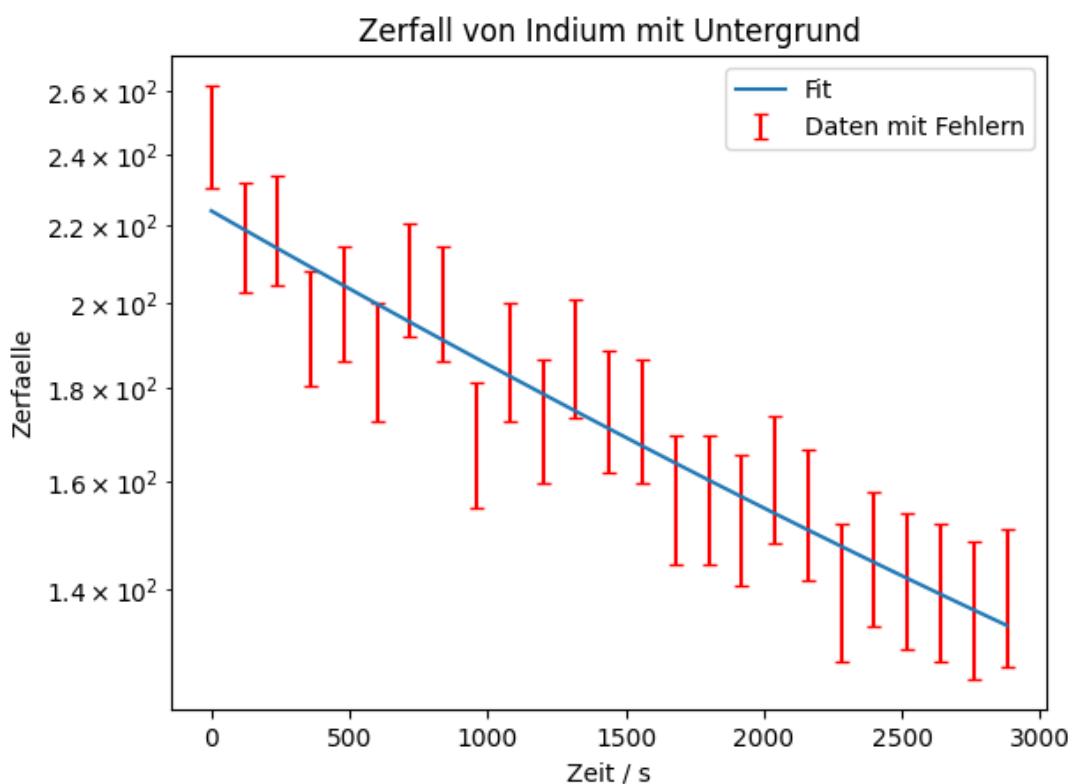
```

plt.errorbar(t,N, Fehler_N, fmt='None', ecolor='r', capsize=3, marker='s', ms=2, label='Daten mit Fehlern')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Indium mit Untergrund')
plt.yscale('log')
plt.plot(t,fit_func2_oben(t,*popt3), label='Fit')
plt.legend()

print("A_o=",popt3[0], ", Standardfehler=", np.sqrt(pcov[0][0]))
print("k_o=",popt3[1], ", Standardfehler=", np.sqrt(pcov[1][1]))

```

A\_o= 175.56347416888534 , Standardfehler= 8.865176788194109  
k\_o= 0.00024980716697075114 , Standardfehler= 0.0038947029506337023



[ ]:

## **6 Quelle**

- Wagner, J. (April 2022). Physikalisches Praktikum PAP 2.2 für Studierende der Physik [Praktikumsanleitung]. Ruprecht-Karls-Universität Heidelberg. Abgerufen am 05. April 2024, von [https://www.physi.uni-heidelberg.de/Einrichtungen/AP/info/Corona/2\\_2\\_2022.pdf](https://www.physi.uni-heidelberg.de/Einrichtungen/AP/info/Corona/2_2_2022.pdf)