

# Versuch 252

## Aktivierung mit thermischen Neutronen

Viktor Ivanov

7. August 2024

### Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
1.1	Motivation . . . . .	2
1.2	Geiger-Müller Zählrohr . . . . .	2
1.2.1	Eigenschaften einer Zählrohr . . . . .	2
1.2.2	Plateaubereich . . . . .	2
1.2.3	Wichtige Bemerkungen über den Zählrohr . . . . .	3
1.3	Physikalische Grundlagen . . . . .	3
<b>2</b>	<b>Messprotokoll und Durchführung des Versuchs</b>	<b>4</b>
<b>3</b>	<b>Auswertung</b>	<b>6</b>
3.1	Zerfall der Silberisotope . . . . .	6
3.1.1	Berücksichtigung des Untergrundfehlers . . . . .	7
3.2	Indiumzerfall . . . . .	9
3.2.1	Berücksichtigung des Untergrundfehlers . . . . .	9
3.3	Zusammenfassung und Diskussion . . . . .	10
3.4	Zusammenfassung . . . . .	10
3.5	Diskussion . . . . .	11
<b>4</b>	<b>Anhang</b>	<b>12</b>
4.1	Quellen . . . . .	12
4.2	Python-Code . . . . .	12

# 1 Einleitung

## 1.1 Motivation

In diesem Versuch wollen wir die Halbwertszeiten, Lebensdauern und Zerfallskonstanten zuerst von  $^{116}\text{In}$  und danach von  $^{108}\text{Ag}$  und  $^{110}\text{Ag}$  bestimmen.

## 1.2 Geiger-Müller Zählrohr

Das Geiger-Müller Zählrohr misst  $\alpha$ -,  $\beta$ -,  $\gamma$ - und  $X$ -Strahlen. Ein Aufbau vom Gerät kann man in Abbildung 1 finden.

Die Hauptkomponenten sind ein Metallzylinder und ein axial verlaufender Anodendraht. Das Rohr ist mit einem geeigneten Gasgemisch erfüllt. Wenn ein schnelles, elektrisch geladenes Teilchen in das Zählrohr gelangt, entstehen durch Ionisation des Gases freie Elektronen und positiv geladene Ionen. Die Elektronen gehen in die Richtung der Anoden und verursachen für eine kurze Zeit Strom. Wie in Abbildung 1 zu sehen ist, ist eine Spannung mit dem Anodendraht verbunden. Wenn der Strom den Widerstand trifft, verursacht er einen Spannungsimpuls, der dann durch einen Verstärker verstärkt wird und dann gemessen wird.

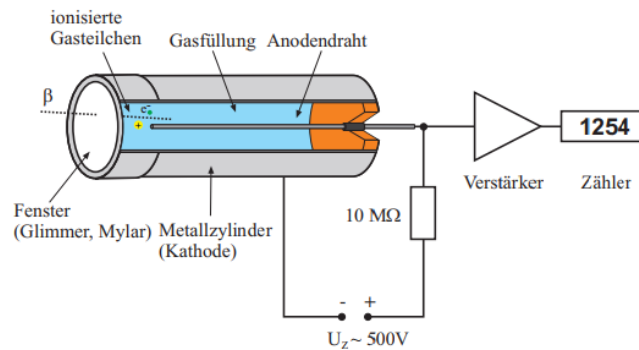


Abbildung 1: Aufbau eines Geiger-Müller Zählrohrs

### 1.2.1 Eigenschaften einer Zählrohr

Bei verschiedenen Anzahlen und Energien von Ladungsträgern misst das Zählrohr verschiedene Ergebnisse, deshalb soll man die Physik eines Zählrohrs beachten.

Bei kleinen Rohrspannungen erreichen nicht alle Primärelektronen den Anodendraht, einige gehen durch Rekombination verloren.

Bei höherer Spannung sinkt die Rekombinationswahrscheinlichkeit und der verursachte Strom ist proportional zur Energie der einfallenden Strahlung.

Bei noch höheren Rohrspannungen werden aus den Primärelektronen auch Sekundärelektronen erzeugt. Die Anzahl der Sekundärelektronen ist proportional zur Primärelektronenanzahl und dieser Bereich heißt "Proportionalbereich".

Wenn man die Rohrspannung noch erhöht, dass das Füllgas komplett ionisiert ist und die einfallenden Teilchen gerade auf den Anodendraht fallen. Das nennt man den Plateaubereich. In diesem Bereich erzeugt jedes einfallende Teilchen ein gleich großes Signal, unabhängig von seiner Energie.

Ein Diagramm von allen Bereichen findet man unter 2.

### 1.2.2 Plateaubereich

Der Plateaubereich ist ziemlich wichtig bei diesem Messgerät, da es uns die Möglichkeit gibt, einzelne Teilchen, unabhängig von ihrer Energie, zu messen.

In der Praxis ist aber der Plateaubereich nicht komplett eben, sondern hat eine Steigung, wie in Abbildung 3 zu sehen ist.

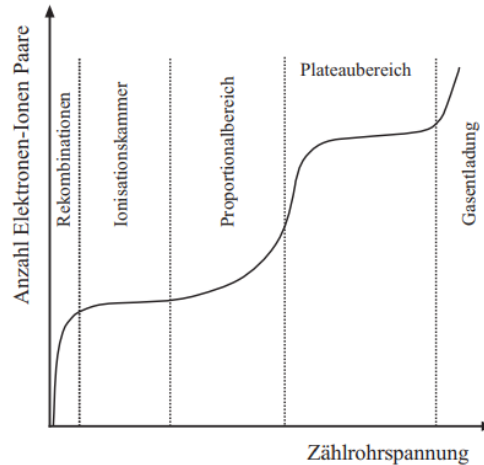


Abbildung 2: Messbereiche eines Geiger-Müller Zählrohrs

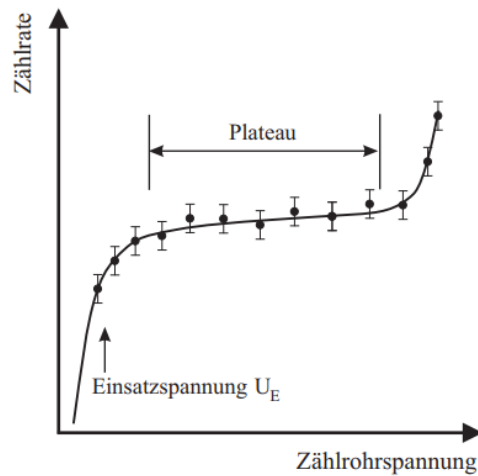


Abbildung 3: Plateaubereich eines Geiger-Müller Zählrohrs

### 1.2.3 Wichtige Bemerkungen über den Zählrohr

Nach einem Entladungsimpuls gibt es eine sogenannte "Totzeit" von c.a.  $10^{-4}s$ , bei denen das Zählrohr keine neuen Teilchen messen kann, deshalb sollen Totzeitkorrekturen vorgenommen werden.

Es gibt auch statistische Schwankungen. Der mittlere statistische Fehler einer Zählung von  $n$  Teilchen beträgt  $\sqrt{n}$  und der mittlere relative Fehler,  $\frac{\sqrt{n}}{n} = \frac{1}{\sqrt{n}}$ . Je mehr Messungen man macht, desto kleiner ist der relative Fehler.

Wegen der natürlichen Radioaktivität zählt das Zählrohr c.a. 50 Ereignisse pro Minute. Das nennt man "Nulleffekt". Alle diese Bemerkungen soll man bei der Auswertung der Resultate berücksichtigen, sonst bekommt er verfälschte Ergebnisse.

## 1.3 Physikalische Grundlagen

Um eine radioaktive Quelle herzustellen, aktiviert man stabile Isotope durch Kernreaktionen. In diesem Versuch verwenden wir Neutronen, da sie nicht der elektromagnetischen Wechselwirkung ausgesetzt sind und vom Kern leicht eingefangen werden können.

Das Präparat der Neutronenquelle enthält die Berylliumspäne und einen  $\alpha$ -Strahler ( $^{241}\text{Am}$ ). Folgende Reaktion

wird durchgeführt:



Wobei  $n$  die entstehenden Neutronen bezeichnen. Sie haben eine Energie von zwischen 1MeV und 10MeV. Die Neutronen stoßen sich mit dem Kern des Wasserstoffs, bis sie nahezu thermische Energie erreicht haben. Das sind elastische Stöße zwischen Teilchen mit ungefähr gleicher Masse. Bei den Stößen verliert das Neutron die Hälfte seiner Energie und entsteht ein Isotop des bestrahlten Elements mit erhöhter Massenzahl.

In dem Fall von dem stabilen Isotop  ${}^{115}\text{In}$  wird der  $\beta$ -Strahler  ${}^{116}\text{In}$  gebildet. Es werden zwei Isomere erzeugt, auf einer Seite  ${}^{116}\text{In}$ , der sich im Grundzustand befindet und auf die andere  ${}^{116m}\text{In}$ , der metastabil ist. Das sind zwei Isomere und auch Nuklide, die mit unterschiedlichen Halbzzeiten in das stabile Isotop  ${}^{116}\text{Sn}$  zerfallen. Die verschiedenen Zerfallsraten sind in Abbildung 4 zu finden. Eine bestimmte Zahl von Kernen wird bei der Aktivierung

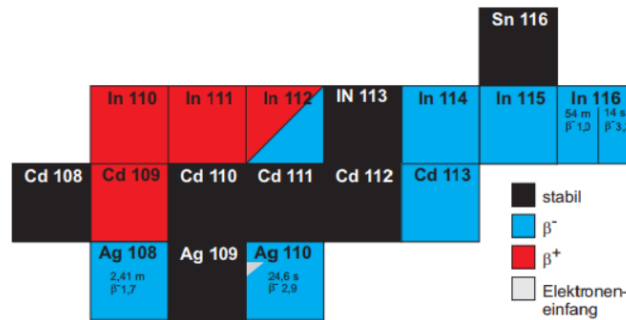


Abbildung 4: Zerfallsraten von Nukliden

erzeugt. Der Zahl folgt dem Zerfallsgesetz, daher gilt für die Aktivität  $A$  als Funktion der Bestrahlungsdauer  $t$ :

$$A(t) = A_{\infty}(1 - e^{-\lambda t}) \quad (2)$$

Wobei  $A_{\infty}$  ein Gleichgewicht, bei dem pro Sekunde gleich viele Kerne entstehen, wie zerfallen. Der Zerfall nach Ende der Aktivierung folgt auch dem Zerfallsgesetz:

$$A(t) = A_0 e^{-\lambda t} \quad (3)$$

Die Halbwertszeit beträgt:

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

Natürliches Silber besteht aus 51%  ${}^{107}\text{Ag}$  und 49%  ${}^{109}\text{Ag}$ , deshalb werden die Isotope  ${}^{108}\text{Ag}$  und  ${}^{110}\text{Ag}$  erzeugt. Sie zerfallen durch  $\beta$ -Zerfall in  ${}^{108}\text{Cd}$  und  ${}^{110}\text{Cd}$ . Die Halbwertszeiten von beiden unterscheiden sich um etwa einen Faktor 6, deshalb können wir das Aktivitätsverhältnis variieren. Bei einer kurzen Zeit entsteht vor allem  ${}^{110}\text{Ag}$ , wobei bei längeren Zeiten  ${}^{110}\text{Ag}$  in Sättigung geht und  ${}^{108}\text{Ag}$  wird vermehrt.

Ein Diagramm von den Aktivitäten bei verschiedenen Aktivierungszeiten von  ${}^{108}\text{Ag}$  und  ${}^{110}\text{Ag}$  ist in Abbildung 5 zu sehen.

## 2 Messprotokoll und Durchführung des Versuchs

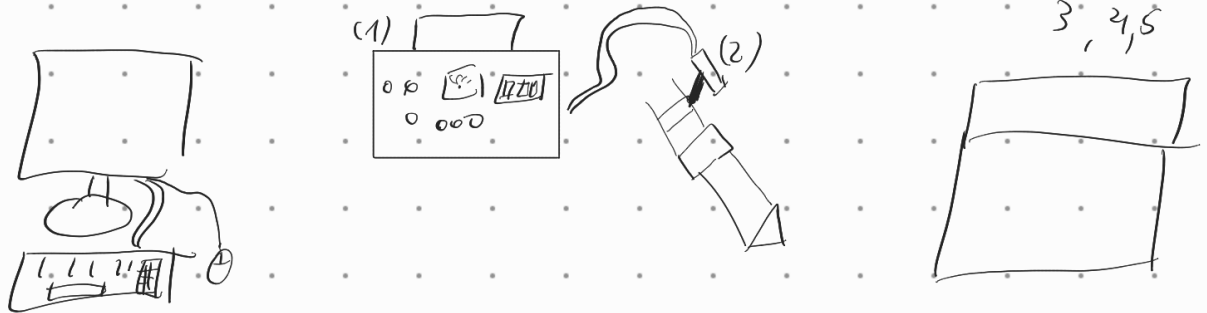
Das Messprotokoll befindet sich auf der nächsten Seite.

3/6/24 Messprotokoll zu Versuch 252  
Aktivierung mit

Viktor Ivanov  
Danae Prontzas  
Versuchsleiter:  
Daniel Häcker

### Messaufbau

- Geiger Müller Zählrohr mit Betriebsgerät (1)
- Externer Impulzzähler (2)
- Neutronenquelle (3)
- Präparatehalterung (4)
- Indium- und Silberbleche (5)



### Skizze 1: Messaufbau

#### II Messung der Halbwertszeit (HWZ) von aktiviertem Silber

- Die Zählrohrspannung des Betriebsgeräts wird auf  $522 \pm 5$  V eingestellt
- Es wird die Untergrundzählrate mit dem entsprechenden Computerprogramm bestimmt.
- Es werden 48 Messungen mit einer Totzeit von 10 s eingestellt
- Das Silberpräparat wird ins Zählrohr platziert und bei  $522 \pm 5$  V gemessen (4 mal)

Das haben wir 4 mal wiederholt

#### III Beim zweiten Teil haben wir den Indium gemessen mit 25 Messungen

Häcker

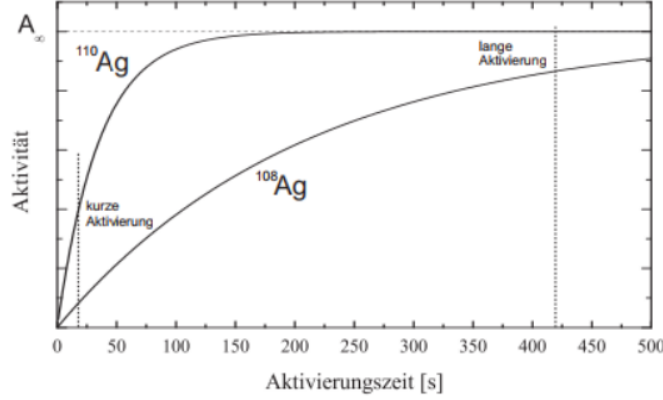


Abbildung 5: Aktivitäten bei verschiedenen Aktivierungszeiten von  $^{108}\text{Ag}$  und  $^{110}\text{Ag}$

### 3 Auswertung

#### 3.1 Zerfall der Silberisotope

Im ersten Auswertungsabschnitt bestimmen wir die Zerfallskonstante und die Lebensdauer von Silber. Da es immer einen Radiationsuntergrund gibt, haben wir eine Untergrundmessung gemacht, deren Wert wir später subtrahieren werden. Wir haben für 8 Minuten 10-Sekunde-Messungen durchgeführt und deren Mittelwert berechnet. Der Fehler ist die Standardabweichung aller Messungen. Die Untergrundmessung beträgt:

$$N_{\text{Untergrund}} = (13,1 \pm 0,9) \frac{\text{Zerfälle}}{10\text{s}} \quad (5)$$

Wir haben dann die vier gemessene Silberzerfälle in einem Diagramm zusammengefügt, Abbildung 6. Aus der Einleitung wissen wir, dass nach dem radioaktiven Zerfallsgesetz die Aktivierung mit einer Exponentialfunktion mal eine Konstante ausgedrückt wird, 3. Deshalb passen wir eine Überlagerung von zwei Exponentialfunktionen auf unsere Daten an:

$$f(x) = A_1 e^{-\lambda_1 x} + A_2 e^{-\lambda_2 x} + y_0 \quad (6)$$

Wobei  $\lambda_1$  auf  $^{110}\text{Ag}$  und  $\lambda_2$  auf  $^{108}\text{Ag}$  sich beziehen und die Konstante  $y_0$  den Untergrund berücksichtigt. Die angepasste Funktion auf die Daten kann man in Abbildung 7 finden.

Wir erhalten folgende Werte für den Fitparameter:

$$A_1 = (302 \pm 19) \quad (7)$$

$$A_2 = (54 \pm 16) \quad (8)$$

$$\lambda_1 = (2,6 \pm 0,3) \cdot 10^{-2} \frac{1}{\text{s}} \quad (9)$$

$$\lambda_2 = (4,4 \pm 1,1) \cdot 10^{-3} \frac{1}{\text{s}} \quad (10)$$

Wir können durch die  $\chi^2$ -Abweichung die Güte des Fits quantitativ untersuchen. Die  $\chi^1$ -Summe beträgt:

$$\chi^2 = \sum_i^N \left( \frac{\text{Funktionswert}_i - \text{Messwert}_i}{\text{Fehler}_i} \right)^2 \quad (11)$$

Die reduzierte  $\chi_{\text{red}}^2$ -Summe berechnet man, indem man die  $\chi^2$ -Summe durch die Anzahl der Freiheitsgrade teilt:

$$\chi_{\text{red}}^2 = \frac{\chi^2}{\#\text{Freiheitsgrade}} \quad (12)$$

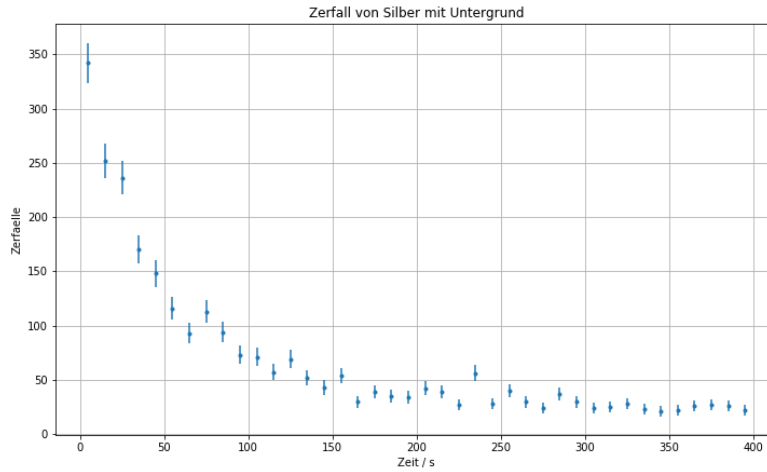


Abbildung 6: Silberzerfall Messdaten aus 4 Messungen

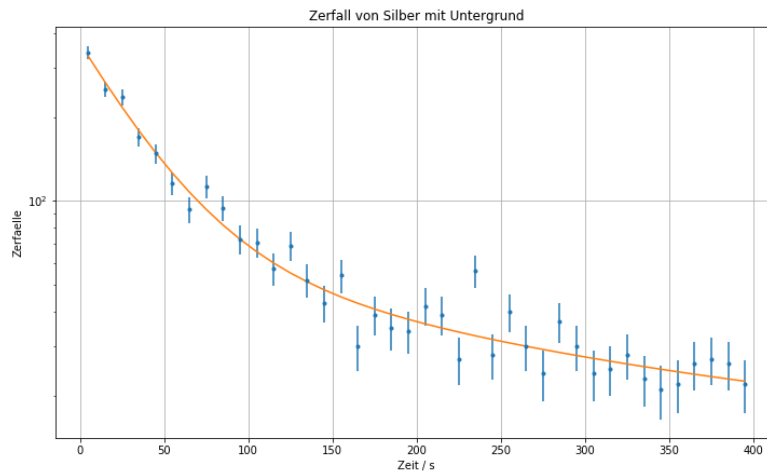


Abbildung 7: Silberzerfall mit angepasste Funktion und berücksichtigte Untergrundmessung

Für die Güte des Fits haben wir folgende Werte berechnet:

$$\boxed{\chi_{Ag}^2 = 44,5} \quad (13)$$

$$\boxed{\chi_{red,Ag}^2 = 1,24} \quad (14)$$

$$\boxed{Fitwahrscheinlichkeit_{Ag} = 16\%} \quad (15)$$

### 3.1.1 Berücksichtigung des Untergrundfehlers

Im letzten Abschnitt haben wir den Untergrund berücksichtigt, jedoch nicht den Fehler des Untergrunds, deshalb habe ich jetzt den Wert  $y_0$  einmal die  $1\sigma$ -Abweichung subtrahiert und einmal addiert. Das Ergebnis kann man in Abbildung 8 finden. Da es schwierig ist, den Fehler ohne reinzuzoomen zu bemerken, habe ich das im rechten oberen Teil des Diagramms in einem Rechteck veranschaulicht. Aus dem Diagramm erhalten wir die Untergeschätzte

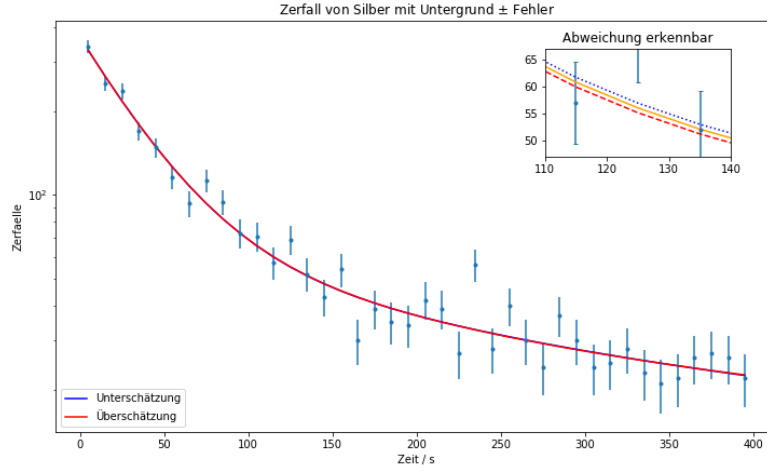


Abbildung 8: Silberzerfall mit angepasste Funktion und berücksichtigte Untergrundmessung mit Fehler

und Übergeschätzte Zerfallskonstanten  $\lambda_{i,min}$  und  $\lambda_{i,max}$ , die wir berücksichtigen können, um eine bessere Fehlerabschätzung über die Halbwertszeiten und Lebensdauern zu bekommen. Den Fehler habe ich wie folgt berechnet:

$$\Delta\lambda_{min,i} := |\lambda_i - \lambda_{min,i}| \quad (16)$$

$$\Delta\lambda_{max,i} := |\lambda_i - \lambda_{max,i}| \quad (17)$$

$$\Delta\lambda_{neu,i} = \sqrt{\left(\frac{\Delta\lambda_{min,\frac{1}{2}} + \Delta\lambda_{max,i}}{2}\right)^2 + (\Delta\lambda_i)^2} \quad (18)$$

Wobei die letzte Formel aus dem Gausschen Fehlerfortpflanzungsgesetz stammt. Mithilfe von Gleichung 4 mit dem Fehler

$$\Delta T_{\frac{1}{2}} = \frac{\ln(2)\Delta\lambda_{neu}}{\lambda^2} \quad (19)$$

können wir die Halbwertszeit bestimmen. Aus der Formel

$$T = \frac{1}{\lambda} \quad (20)$$

mit dem Fehler

$$\Delta\tau = \frac{\Delta\lambda}{\lambda^2} \quad (21)$$

können wir auch leicht die Lebensdauer berechnen. Die Ergebnisse habe ich in Tabelle 1 zusammengefasst.

Tabelle 1: Zusammengefasste Ergebnisse Silber

	$^{110}\text{Ag}$	$\Delta^{110}\text{Ag}$	$^{108}\text{Ag}$	$\Delta^{108}\text{Ag}$
Zerfallskonstante $\lambda$ [ $\frac{1}{s}$ ]	0.026	0.003	0.0044	0.0012
Halbwertszeit $T_{\frac{1}{2}}$ [s]	26	3	157	41
Lebensdauer $\tau$ [s]	38	4	227	59

Die Ergebnisse besprechen wir in der Diskussion.



### 3.2 Indiumzerfall

Bei dem Indiumzerfall ausarbeiten wir die Daten analog zu dem Silberzerfall. Wir verwenden die Untergrundmessung aus dem ersten Teil des Versuchs.

Wir passen eine Exponentialfunktion an (da wir bei Indium nicht mit zwei Isotopen arbeiten):

$$g(x) = A_1 e^{-\lambda_1 x} + y_0 \quad (22)$$

Die Daten mit der angepassten Funktion findet man in Abbildung 9. Wir erhalten folgende Werte für den Fitpara-

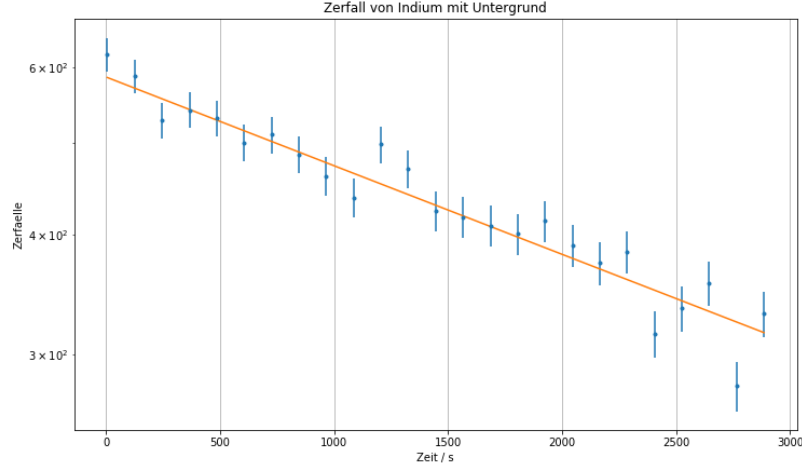


Abbildung 9: Indiumzerfall mit angepasster Funktion und berücksichtigte Untergrundmessung

meter:

$$A_1 = (548 \pm 11) \quad (23)$$

$$\lambda_1 = (2,35 \pm 0,13) \cdot 10^{-4} \frac{1}{s} \quad (24)$$

Für die Güte des Fits haben wir folgende Werte berechnet:

$$\chi_{In}^2 = 28,5 \quad (25)$$

$$\chi_{red,In}^2 = 1,42 \quad (26)$$

$$Fitwahrscheinlichkeit_{In} = 10\% \quad (27)$$

#### 3.2.1 Berücksichtigung des Untergrundfehlers

Analog zum Silber addieren wir und subtrahieren wir die  $1\sigma$ -Abweichung an der Wert  $y_0$ , um den Fehler des Untergrunds zu berücksichtigen, Abbildung 10. Die Ergebnisse habe ich in Tabelle 2 zusammengefasst. Die  $\sigma$ -Abweichung zwischen den theoretischen und unseren Werten können wir mithilfe folgender Formel berechnen:

$$\sigma = \frac{|a - b|}{\sqrt{(\Delta a)^2 + (\Delta b)^2}} \quad (28)$$

Die Ergebnisse besprechen wir in der Diskussion.

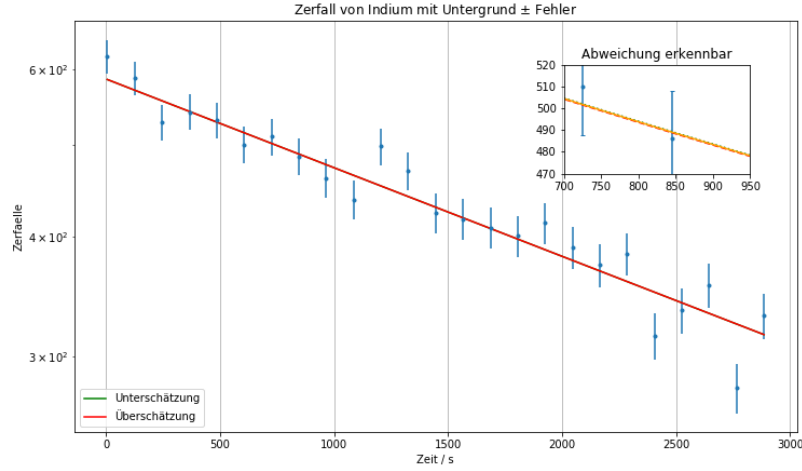


Abbildung 10: Indiumzerfall mit angepasste Funktion und berücksichtigte Untergrundmessung mit Fehler

Tabelle 2: Zusammengefasste Ergebnisse Indium

	Wert Indium	Fehler Indium
Zerfallskonstante $\lambda$ [ $\frac{1}{s}$ ]. $10^{-4}$	2.16	0.12
Halbwertszeit $T_{\frac{1}{2}}$ [s]	3215	165
Lebensdauer $\tau$ [s]	4639	239

### 3.3 Zusammenfassung und Diskussion

#### 3.4 Zusammenfassung

In diesem Versuch haben wir die Silber und Indium aktiviert und ihre Eigenschaften untersucht. Wir haben eine Untergrundmessung gemacht und danach haben wir für entsprechende Zeiten die Zerfallsraten von Silber und Indium gemessen. Die Daten habe ich in Diagrammen dargestellt und Funktionen gemäß dem Zerfallsgesetz angepasst, daraus habe ich die Fitparameter bestimmt und die Zerfallskonstanten, Halbwertszeiten und Lebensdauern berechnet. In Tabelle 3 sind alle Endergebnisse und die Abweichungen vom Literaturwert dargestellt. Die Literaturwerte von  $^{110}\text{Ag}$  lauten:

$$\lambda_{^{110}\text{Ag},theo} = 281,8 \frac{10^{-4}}{s} \quad (29)$$

$$T_{\frac{1}{2},^{110}\text{Ag},theo} = 24,6s \quad (30)$$

$$\tau_{^{110}\text{Ag},theo} = 35,49s \quad (31)$$

von  $^{108}\text{Ag}$ :

$$\lambda_{^{108}\text{Ag},theo} = 48,74 \frac{10^{-4}}{s} \quad (32)$$

$$T_{\frac{1}{2},^{108}\text{Ag},theo} = 161s \quad (33)$$

$$\tau_{^{108}\text{Ag},theo} = 205s \quad (34)$$

und von  $^{116}\text{In}$ :

$$\lambda_{^{116}\text{In},\text{theo}} = 2,13 \frac{10^{-4}}{\text{s}} \quad (35)$$

$$T_{\frac{1}{2},^{116}\text{In},\text{theo}} = 3240\text{s} \quad (36)$$

$$\tau_{^{116}\text{In},\text{theo}} = 4699\text{s} \quad (37)$$

Tabelle 3: Zusammengefasste Ergebnisse mit  $\sigma$ -Abweichungen

	$^{110}\text{Ag}$	$\Delta^{110}\text{Ag}$	$^{108}\text{Ag}$	$\Delta^{108}\text{Ag}$	Wert Indium	Fehler Indium
Zerfallskonstante $\lambda$ [ $\frac{1}{\text{s}}$ ], $10^{-4}$	260	30	44	12	2.16	0.12
Abweichung theoretische experimentelle Wert [ $\sigma$ ]	0.73		0.40		0.27	
Halbwertszeit $T_{\frac{1}{2}}$ [s]	26	3	157	41	3215	165
Abweichung theoretische experimentelle Wert [ $\sigma$ ]	0.52		0.10		0.15	
Lebensdauer $\tau$ [s]	38	4	227	59	4639	239
Abweichung theoretische experimentelle Wert [ $\sigma$ ]	0.57		0.37		0.25	

### 3.5 Diskussion

Die reduzierte  $\chi$ -Wert bei der Anpassung der Fitfunktion auf der Silbermessung beträgt  $\chi_{red,Ag}^2 = 1,24$ . Die optimalste  $\chi$ -Wert ist 1, daher ist unseren Fit nicht der beste, aber ist er auch nicht ziemlich schlecht. Die Fitwahrscheinlichkeit beträgt 16%, was auch niedriger als der optimale Wert von (50 – 70)% ist. Nach meiner Meinung haben wir einfach zu wenige Messdaten um einen perfekten Fit zu haben. In Diagramm 7 beobachten wir, dass die Messungen konsistent mit der Fitfunktion sind, mit einigen Ausnahmen, die Normal wegen der statistischen Natur der radioaktive Zerfall sind.

Ein anderer Grund für die nicht perfekte Fitwahrscheinlichkeit und  $\chi$ -Wert ist, dass während der Grundmessung die Strahlung von dem anderen Versuchsaufbau nicht berücksichtigt wurde, was zu Störungen bei den niedrigeren Zerfallsraten geführt hat.

Für Indium haben wir einen  $\chi$ -Wert von  $\chi_{red,In}^2 = 1,42$  und eine Fitwahrscheinlichkeit von 10%, das ist ein schlechterer Fit als bei Silber. Man kann es stark in Abbildung 9 sehen, dass bei einer Zerfallsrate von weniger als 380 Zerfälle pro Sekunde eine ziemlich große Abweichung von der Fitfunktion gibt. Ich würde annehmen, dass es einfach wegen der kleinen Zerfallsrate der Grundstrom mehr den Daten stört, aber Silber hat eine kleinere Zerfallsrate als Indium, deshalb sollen wir das stärker bei Silber beobachten. Jedoch habe ich den Anteil von Daten am Ende der Messung gelöscht und das Diagramm neu erstellt, Abbildung 11. Für die Güte des Fits haben wir folgende Werte berechnet:

$$\chi_{In}^2 = 11,5 \quad (38)$$

$$\chi_{red,In}^2 = 0,77 \quad (39)$$

$$\text{Fitwahrscheinlichkeit}_{In} = 72\% \quad (40)$$

Dabei ist der Fit ziemlich optimal, sogar ein bisschen overfittet, da die  $\chi$ -Wert kleiner als 1 ist. Die Ergebnisse für die Lebensdauer, Halbwertszeit und Zerfallskonstante sind nicht viel geändert; deshalb verwende ich die früher bestimmten Werte. Ich würde auch sagen, dass wegen der kleinen Zerfallsrate vom Silber die Grundstrahlung mehr die Messdaten stört und daher sind alle Abweichungen größer.

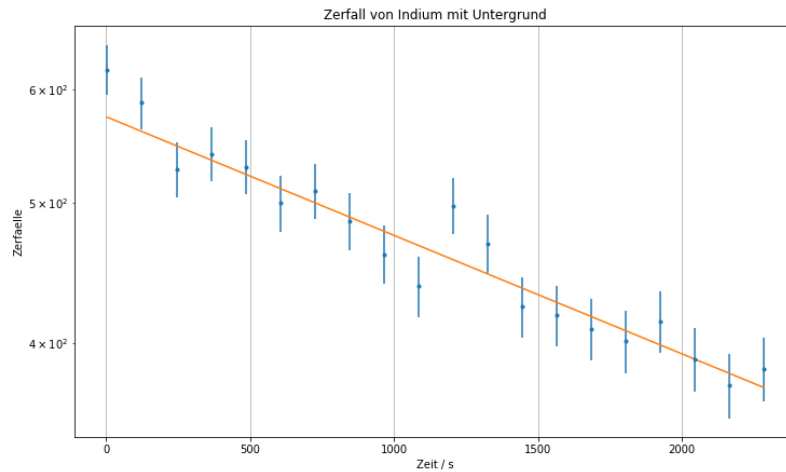


Abbildung 11: Indiumzerfall mit angepasste Funktion und berücksichtigte Untergrundmessung

Wenn wir Tabelle 3 betrachten, sehen wir die Endergebnisse von den Konstanten, die wir gesucht haben. Der relative Fehler von  $^{110}\text{Ag}$  beträgt zwischen 1% und 11%, wobei der relative Fehler von  $^{108}\text{Ag}$  liegt zwischen 20% und 27%, deshalb sind auch die  $\sigma$ -Abweichungen bei  $^{108}\text{Ag}$  kleiner. Der Grund für die großen Fehler ist genau das, was wir früher besprochen haben, Silber hat eine kleine Zerfallsrate und daher sind die Daten mehr wegen der Untergrundstrahlung gestört, daher ist auch die Standardabweichung von den Messdaten, die wir als relative Fehler betrachtet haben, größer.

Für Indium haben wir, wie erwartet, kleinere relative Fehler, im Bereich von 5%. Die  $\sigma$ -Abweichungen liegen auch unter dem  $0,3\sigma$ -Bereich, was ziemlich nah zu dem theoretischen Wert ist.

Da alle  $\sigma$ -Abweichungen unter dem  $1\sigma$ -Bereich liegen, sind alle insignifikant.

Im Großen und Ganzen ist der Versuch erfolgreich. Wenn wir jedoch bessere Ergebnisse kriegen wollen, sollen wir die Messung irgendwo, wo es weniger Untergrundstrahlung gibt, wie z.B. unter der Erde. Bessere Messgeräte werden auch darüber helfen, aber das Wichtigste wäre, die Strahlung von draußen abzuschirmen.

Mir hat der Versuch gefällt, da wir uns mehr mit Strahlung beschäftigt hat, was mich viel interessiert.

## 4 Anhang

### 4.1 Quellen

Alle Informationen, die ich im Protokoll verwendet habe, stammen aus der Praktikumsanleitung, Ausgabe 4.2023 und Wikipedia.com.

### 4.2 Python-Code

Der Python-Code befindet sich auf der nächsten Seite.

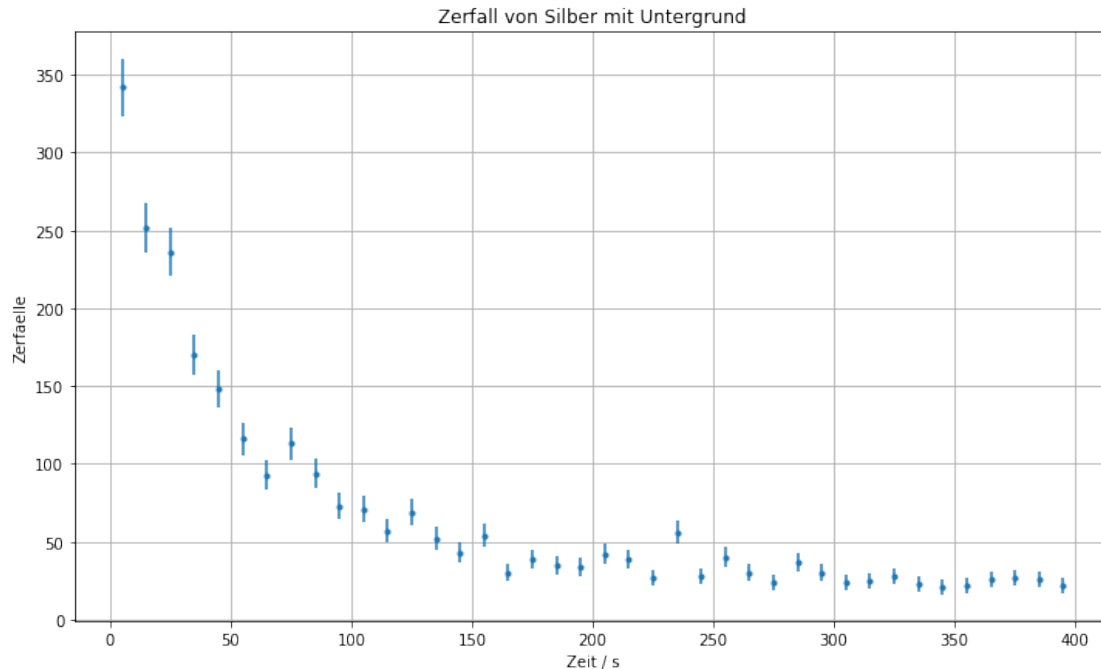
August 7, 2024

```
[1]: #Aufgabe 1: Untergrundbestimmung
      %matplotlib inline
      import matplotlib.pyplot as plt
      import numpy as np
```

```
[2]: unterg = np.loadtxt('untergrund.txt', usecols=[1], skiprows=4)
      #Da 4 Messreihen addiert werden muss Untergrund um Faktor 4 angehoben werden
      mittelw_unterg = np.mean(4*unterg)
      fehler_unterg = np.std(4*unterg)/np.sqrt(len(unterg))
      print('Mittelwert:', mittelw_unterg, 'Fehler:', fehler_unterg)
```

Mittelwert: 13.083333333333334 Fehler: 0.9408889364535757

```
[3]: #Bestimmung Zerfallskonstante
      n1 = np.loadtxt('silber1.txt', usecols=[1], delimiter = ",", skiprows=4)
      n2 = np.loadtxt('silber2.txt', usecols=[1], delimiter = ",", skiprows=4)
      n3 = np.loadtxt('silber3.txt', usecols=[1], delimiter = ",", skiprows=4)
      n4 = np.loadtxt('silber4.txt', usecols=[1], delimiter = ",", skiprows=4)
      N = n1 + n2 + n3 + n4
      Fehler_N = np.sqrt(N)
      t = np.arange(5, 405, 10)
      plt.figure(figsize = (12, 7))
      plt.errorbar(t, N, Fehler_N, linestyle='None', fmt=".")
      plt.xlabel('Zeit / s')
      plt.ylabel('Zerfaelle')
      plt.grid()
      plt.title('Zerfall von Silber mit Untergrund')
      plt.savefig("SilberohneKurve.png", format="png")
```



```
[4]: #Fit der Zerfallsfunktion an die Daten
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=[250,0.2,50,0.
    ↳001],sigma=Fehler_N,absolute_sigma=True)
plt.figure(figsize = (12,7))
plt.errorbar(t,N, Fehler_N, linestyle='None',fmt=".")
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Silber mit Untergrund')
plt.yscale('log')
plt.grid()
plt.plot(t,fit_func(t,*popt))
plt.savefig('Silber.png',format='png')
#Fitparameter
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]))
A_1 = popt[0]
sig_A_1 = np.sqrt(pcov[0][0])
l_1 = popt[1]
sig_l_1 = np.sqrt(pcov[1][1])
```

```

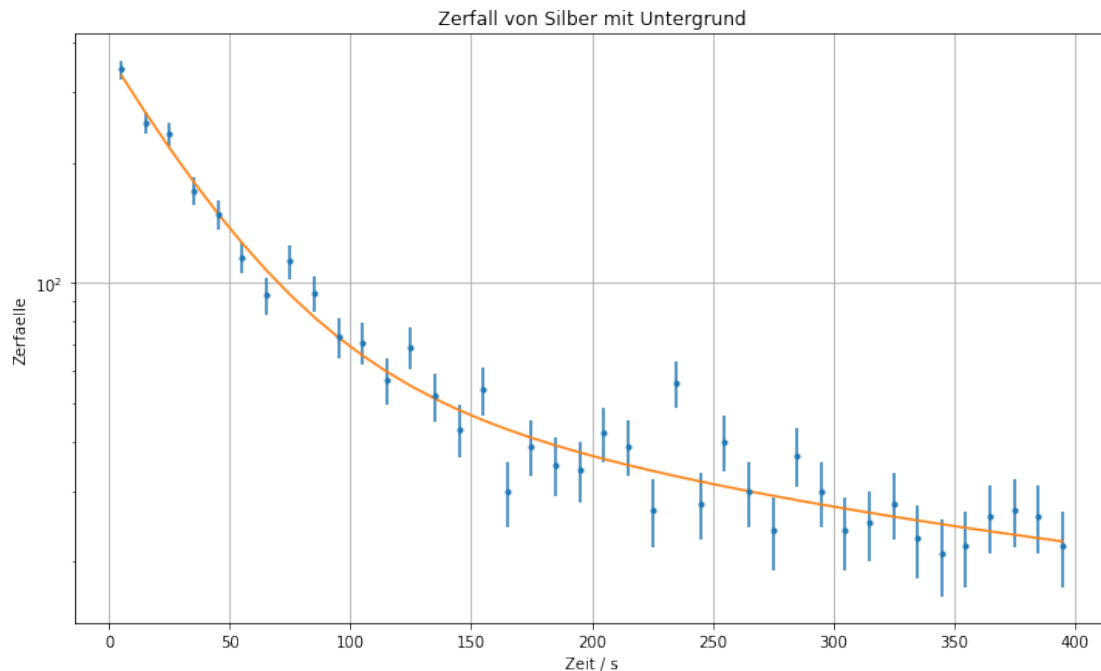
A_2 = popt[2]
sig_A_2 = np.sqrt(pcov[2][2])
l_2 = popt[3]
sig_l_2 = np.sqrt(pcov[3][3])

```

```

A1= 302.6671818145436 , Standardfehler= 18.893246518147127
l1= 0.026483334840926635 , Standardfehler= 0.003080802799231616
A2= 53.815730144687926 , Standardfehler= 16.37035714593234
l2= 0.004414393334994159 , Standardfehler= 0.001119818299992263

```



```

[5]: #Güte des Fits mit Chi^2
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)
from scipy.stats import chi2
prob=round(1-chi2.cdf(chi2_,dof),2)*100
print("Wahrscheinlichkeit=", prob,"%")

```

```

chi2= 44.48147144369391
chi2_red= 1.2355964289914976
Wahrscheinlichkeit= 16.0 %

```

```

[6]: #Berücksichtigung des Untergrunds
plt.figure(figsize=(12,7))
plt.errorbar(t,N, yerr=Fehler_N, linestyle='None',fmt=".")
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.grid()
plt.title('Zerfall von Silber mit Untergrund  $\lambda_{pm}$  Fehler')
plt.yscale('log')
plt.grid()
#Fit bei Subtrahieren des Fehlers
y0 = mittelw_unterg - fehler_unterg
def fit_func(x, A1,l1,A2,l2):
    return A1*np.exp(-x*l1) + A2*np.exp(-x*l2) + y0
popt_min, pcov_min=curve_fit(fit_func,t,N, p0=[250,0.2,50,0.
    ↳001],sigma=Fehler_N, absolute_sigma=True)
plt.plot(t,fit_func(t,*popt_min),label = "Unterschätzung", color="blue")
print("A1min=",popt_min[0], ", Standardfehler=", np.sqrt(pcov_min[0][0]))
print("l1min=",popt_min[1], ", Standardfehler=", np.sqrt(pcov_min[1][1]))
print("A2min=",popt_min[2], ", Standardfehler=", np.sqrt(pcov_min[2][2]))
print("l2min=",popt_min[3], ", Standardfehler=", np.sqrt(pcov_min[3][3]))
A_min_1 = popt_min[0]
sig_A_min_1 = np.sqrt(pcov_min[0][0])
l_min_1 = popt_min[1]
sig_l_min_1 = np.sqrt(pcov_min[1][1])
A_min_2 = popt_min[2]
sig_A_min_2 = np.sqrt(pcov_min[2][2])
l_min_2 = popt_min[3]
sig_l_min_2 = np.sqrt(pcov_min[3][3])
#Fit bei Addieren des Fehlers
y0 = mittelw_unterg + fehler_unterg
def fit_func(x, A1,l1,A2,l2):
    return A1*np.exp(-x*l1) + A2*np.exp(-x*l2) + y0
popt_max, pcov_max=curve_fit(fit_func,t,N, p0=[250,0.2,50,0.
    ↳001],sigma=Fehler_N, absolute_sigma=True)
plt.plot(t,fit_func(t,*popt_max),label = "Überschätzung", color="red")
print("A1max=",popt_max[0], ", Standardfehler=", np.sqrt(pcov_max[0][0]))
print("l1max=",popt_max[1], ", Standardfehler=", np.sqrt(pcov_max[1][1]))
print("A2max=",popt_max[2], ", Standardfehler=", np.sqrt(pcov_max[2][2]))
print("l2max=",popt_max[3], ", Standardfehler=", np.sqrt(pcov_max[3][3]))
A_max_1 = popt_max[0]
sig_A_max_1 = np.sqrt(pcov_max[0][0])
l_max_1 = popt_max[1]
sig_l_max_1 = np.sqrt(pcov_max[1][1])
A_max_2 = popt_max[2]
sig_A_max_2 = np.sqrt(pcov_max[2][2])
l_max_2 = popt_max[3]
sig_l_max_2 = np.sqrt(pcov_max[3][3])

```

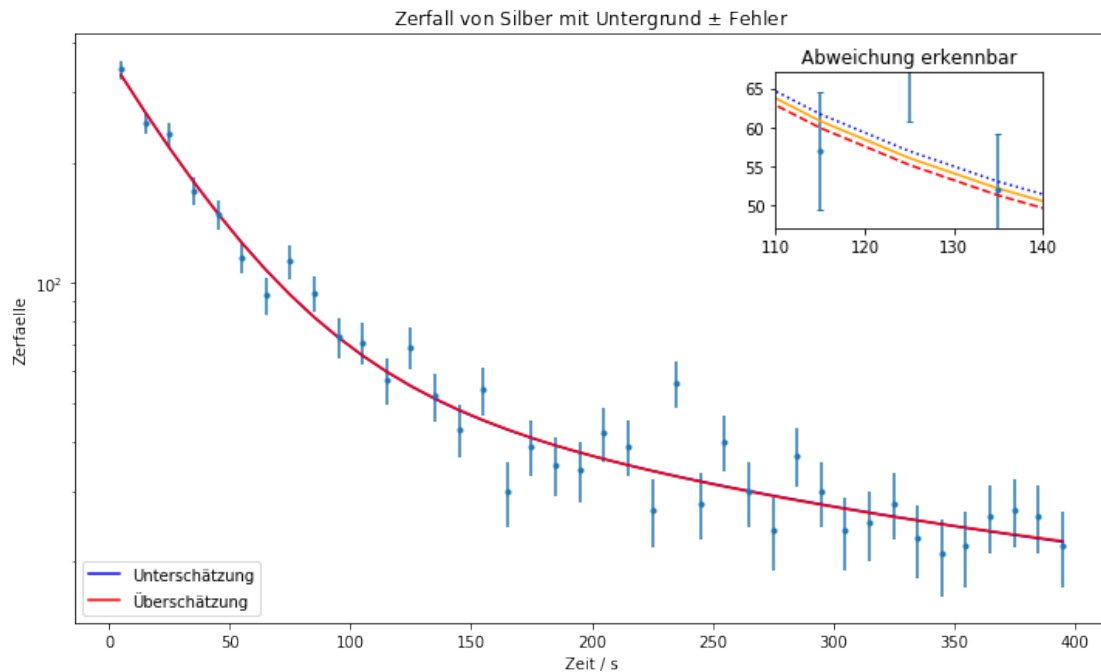


```

plt.legend(loc="lower left")
#Fenster zur besseren Visualisierung des Unterschiedes
a = plt.axes([.65, .63, .2, .2], facecolor = 'white')
plt.plot(t, fit_func(t, *popt_min), color = 'blue', ls = 'dotted')
plt.plot(t, fit_func(t, *popt_max), color = 'red', ls = '--')
plt.plot(t, fit_func(t, *popt), color = 'orange')
plt.errorbar(t, N, yerr = Fehler_N, fmt = '.', capsize = 2)
plt.xlim(110, 140)
plt.ylim(47, 67)
plt.title('Abweichung erkennbar')
plt.savefig("SilbermitUntergrundfehler.png",format="png")

```

A1min= 303.9996710346695 , Standardfehler= 18.43087048179739  
 l1min= 0.026304772443081277 , Standardfehler= 0.002977137996982717  
 A2min= 53.04227820381195 , Standardfehler= 15.403343834070018  
 l2min= 0.004148314117161798 , Standardfehler= 0.0010590696589290874  
 A1max= 301.0626451521881 , Standardfehler= 19.48695005503778  
 l1max= 0.026693434378221942 , Standardfehler= 0.003205279050644499  
 A2max= 54.91539980296562 , Standardfehler= 17.52939604228597  
 l2max= 0.004715671138302823 , Standardfehler= 0.0011878495960178433



```

[7]: #Differenz Zerfallskonstante
diff_min_l_1 = np.abs(l_1-l_min_1)
diff_max_l_1 = np.abs(l_1-l_max_1)
diff_min_l_2 = np.abs(l_2-l_min_2)

```

```

diff_max_1_2 = np.abs(l_2-l_max_2)
feh_l_1 = np.sqrt( ((diff_min_l_1 + diff_max_l_1) / 2) ** 2 + sig_l_1 ** 2 )
feh_l_2 = np.sqrt( ((diff_min_l_2 + diff_max_l_2) / 2) ** 2 + sig_l_2 ** 2 )
print("Zerfallskonstante l_1 ",np.round(l_1,5),"+/-",np.round(feh_l_1,5))
print("Zerfallskonstante l_2 ",np.round(l_2,7),"+/-",np.round(feh_l_2,6))
print("Halbwertszeit T_1 ", np.round(np.log(2)/l_1,3),"+/-",np.round(np.log(2)*
    ↪feh_l_1 / (l_1 ** 2),3))
print("Halbwertszeit T_2 ",np.round(np.log(2)/l_2,1),"+/-",np.round(np.log(2)*
    ↪feh_l_2 / (l_2 ** 2),1))
print("Lebensdauer tau_1 ",np.round((1/l_1),3),"+/-", (np.round(1 * feh_l_1 /
    ↪(l_1 ** 2),3)))
print("Lebensdauer tau_2 ",np.round((1/l_2),3),"+/-", (np.round(1 * feh_l_2 /
    ↪(l_2 ** 2),1)))

```

```

Zerfallskonstante l_1  0.02648 +/- 0.00309
Zerfallskonstante l_2  0.0044144 +/- 0.001155
Halbwertszeit T_1    26.173 +/- 3.051
Halbwertszeit T_2    157.0 +/- 41.1
Lebensdauer tau_1    37.76 +/- 4.401
Lebensdauer tau_2    226.532 +/- 59.3

```

```

[8]: #Vergleich mit Literaturwerten aus Nuklidkarte
def literaturVergleich(name,mess,sig_mess,lit,sig_lit):
    print(name,":")
    print("Absolute Abweichung: ",np.abs(mess-lit))
    print("Sigma-Abweichung: ",np.abs(mess-lit)/np.sqrt(sig_mess**2
    ↪+sig_lit**2))

```

```

[9]: #Aufgabe 2: Indiumzerfall
unterg =np.loadtxt('untergrund.txt', usecols=[1],skiprows=4)
mittelw_unterg=np.mean(unterg)
fehler_unterg=np.std(unterg)/np.sqrt(len(unterg))
print('Mittelwert:', mittelw_unterg, 'Fehler:',fehler_unterg)

```

```

Mittelwert: 3.2708333333333335 Fehler: 0.23522223411339394

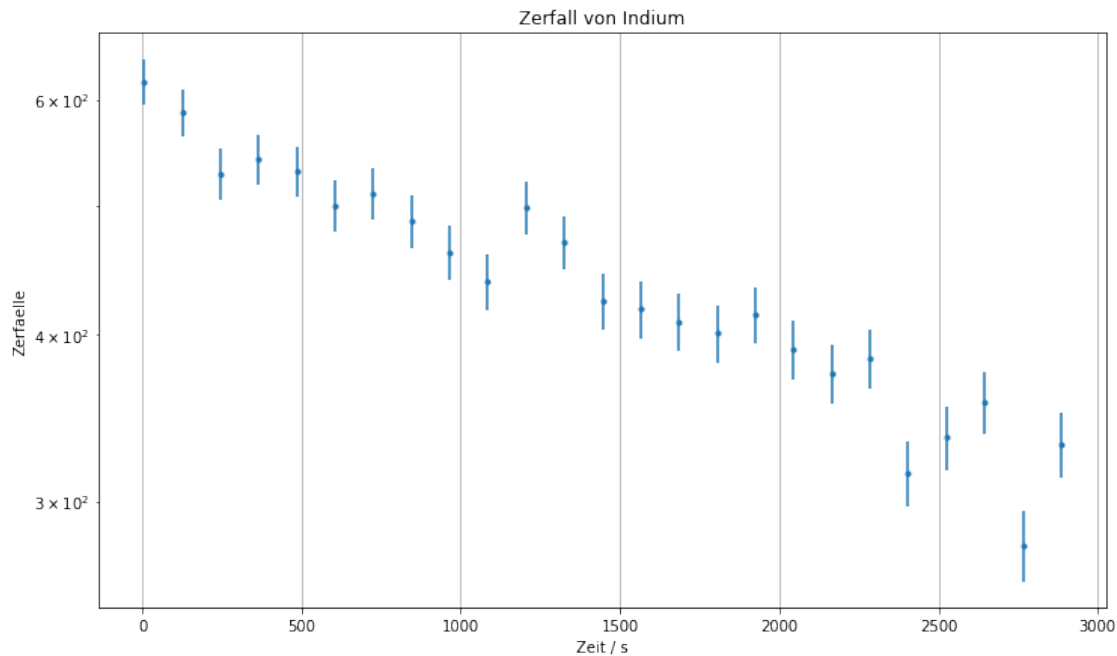
```

```

[10]: #Bestimmung Zerfallskonstante
N =np.loadtxt('indium.txt', usecols=[1],skiprows=4)
Fehler_N=np.sqrt(N)
#Messintervall 120s; 50 Minuten lang
t=np.arange(5,3005,120)
plt.figure(figsize=(12,7))
plt.errorbar(t,N, Fehler_N, linestyle='None',fmt=".")
plt.xlabel('Zeit / s')
plt.grid()
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Indium')

```

```
plt.yscale('log')
plt.savefig("IndiumohneFit.png",format="png")
```



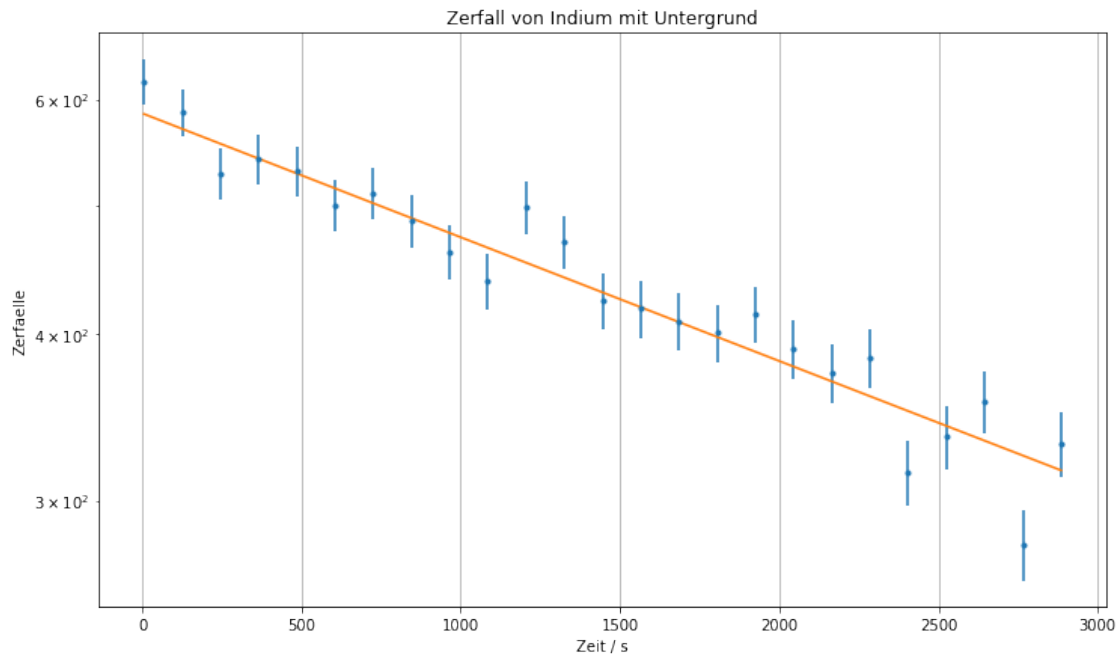
```
[11]: #Fit der Zerfallsfunktion an die Daten
y0=mittelw_unterg #Untergrund
#Nur noch eine Zerfallskonstante
def fit_func(x, A1,l1):
    return A1*np.exp(-x*l1) + y0
#Erster Messwert wird wegen Abweichung nicht für Fit verwendet
from scipy.optimize import curve_fit
popt, pcov=curve_fit(fit_func,t[1:],N[1:], p0=[500,0.02],sigma=Fehler_N[1:
↪],absolute_sigma=True)
plt.figure(figsize = (12,7))
plt.errorbar(t,N, Fehler_N, linestyle='None',fmt=".")
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Indium mit Untergrund')
plt.yscale('log')
plt.grid()
plt.plot(t,fit_func(t,*popt))
plt.savefig('IndiummitUntergrund.png',format='png')
A_1 = popt[0]
sig_A_1 = np.sqrt(pcov[0][0])
l_1 = popt[1]
sig_l_1 = np.sqrt(pcov[1][1])
```

```
#Fitparameter
print("A1=",A_1, ", Standardfehler=", sig_A_1)
print("l1=",l_1, ", Standardfehler=", sig_l_1)
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:5: RuntimeWarning:  
overflow encountered in exp

"""

```
A1= 582.5640174445759 , Standardfehler= 11.084907864197579
l1= 0.00021558800824819104 , Standardfehler= 1.1980505301228857e-05
```



```
[12]: #Güte des Fits mit Chi^2
chi2=np.sum((fit_func(t[1:],*popt)-N[1:])**2/Fehler_N[1:]**2)
dof=len(N[1:])-4 #dof:degrees of freedom, Freiheitsgrad
chi2_red=chi2/dof
print("chi2=", chi2_)
print("chi2_red=",chi2_red)
from scipy.stats import chi2
prob=round(1-chi2.cdf(chi2_,dof),2)*100
print("Wahrscheinlichkeit=", prob,"%")
```

```
chi2= 28.16065094821038
chi2_red= 1.408032547410519
Wahrscheinlichkeit= 11.0 %
```

```

[13]: #Berücksichtigung des Untergrunds
plt.figure(figsize=(12,7))
plt.errorbar(t,N, yerr=Fehler_N, linestyle='None',fmt=".")
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Indium mit Untergrund  $\lambda_{pm}$  Fehler')
plt.yscale('log')
plt.grid()
#Fit bei Subtrahieren des Fehlers
y0 = mittelw_unterg - fehler_unterg
def fit_func(x, A1,l1):
    return A1*np.exp(-x*l1) + y0
popt_min, pcov_min=curve_fit(fit_func,t[1:],N[1:], p0=[500,0.
    ↳02],sigma=Fehler_N[1:], absolute_sigma=True)
plt.plot(t,fit_func(t,*popt_min),label = "Unterschätzung", color="green")
print("A1=",popt_min[0], ", Standardfehler=", np.sqrt(pcov_min[0][0]))
print("l1=",popt_min[1], ", Standardfehler=", np.sqrt(pcov_min[1][1]))
A_min_1 = popt_min[0]
sig_A_min_1 = np.sqrt(pcov_min[0][0])
l_min_1 = popt_min[1]
sig_l_min_1 = np.sqrt(pcov_min[1][1])
#Fit bei Addieren des Fehlers
y0 = mittelw_unterg + fehler_unterg
def fit_func(x, A1,l1):
    return A1*np.exp(-x*l1) + y0
popt_max, pcov_max=curve_fit(fit_func,t[1:],N[1:], p0=[500,0.
    ↳02],sigma=Fehler_N[1:], absolute_sigma=True)
plt.plot(t,fit_func(t,*popt_max),label = "Überschätzung",color="red")
print("A1=",popt_max[0], ", Standardfehler=", np.sqrt(pcov_max[0][0]))
print("l1=",popt_max[1], ", Standardfehler=", np.sqrt(pcov_max[1][1]))
A_max_1 = popt_max[0]
sig_A_max_1 = np.sqrt(pcov_max[0][0])
l_max_1 = popt_max[1]
sig_l_max_1 = np.sqrt(pcov_max[1][1])
plt.legend(loc = "lower left")
#Fenster
a = plt.axes([.65, .6, .2, .2], facecolor = 'white')
plt.plot(t, fit_func(t, *popt_min), color = 'green', ls = 'dotted')
plt.plot(t, fit_func(t, *popt_max), color = 'red', ls = '--')
plt.plot(t, fit_func(t, *popt), color = 'orange')
plt.errorbar(t, N, yerr = Fehler_N, fmt = '.', capsize = 2)
plt.xlim(700, 950)
plt.ylim(470,520)
plt.title('Abweichung erkennbar')
plt.savefig("IndiummitFehlerUntergrund.png",format = "png")

```

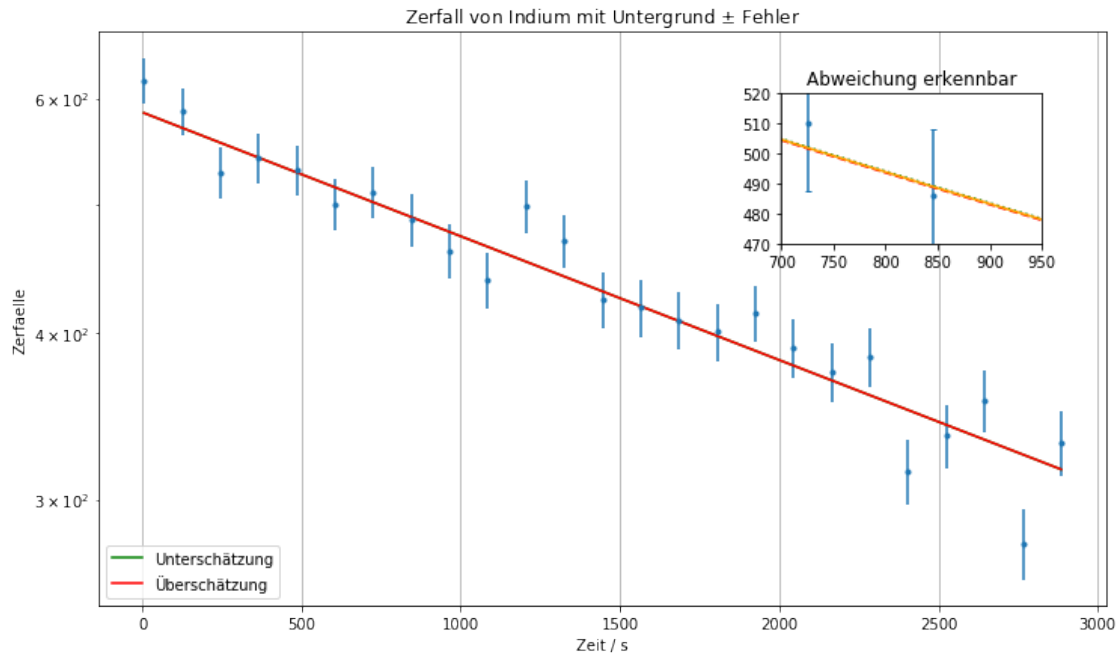
A1= 582.7902917440501 , Standardfehler= 11.083911067181033

```
l1= 0.00021546918032481295 , Standardfehler= 1.1973660269872267e-05
A1= 582.3377517325305 , Standardfehler= 11.085905870340698
l1= 0.00021570696570804654 , Standardfehler= 1.1987358637317317e-05
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:12: RuntimeWarning:
overflow encountered in exp
```

```
if sys.path[0] == '':
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:24: RuntimeWarning:
overflow encountered in exp
```



```
[14]: #Differenz Zerfallskonstante
diff_min_l_1 = np.abs(l_1-l_min_1)
diff_max_l_1 = np.abs(l_1-l_max_1)
feh_l_1 = np.sqrt( ((diff_min_l_1 + diff_max_l_1) / 2) ** 2 + sig_l_1**2 )
print(feh_l_1)
print("Zerfallskonstante l_1 ",np.round(l_1,6),"+/-",np.round(feh_l_1,8))
print("Halbwertszeit T_1 ", np.round(np.log(2)/l_1,1),"+/-",np.round(np.log(2)*_
↪feh_l_1 / (l_1 ** 2),1))
print("Lebensdauer tau_1 ",np.round((1/l_1),1),"+/-", (np.round(1 * feh_l_1 /
↪(l_1 ** 2),1)))
```

```
1.1981095223095952e-05
Zerfallskonstante l_1  0.000216 +/- 1.198e-05
Halbwertszeit T_1  3215.1 +/- 178.7
Lebensdauer tau_1  4638.5 +/- 257.8
```