

Versuch 252: Aktivierung von Indium und von Silber mit thermischen Neutronen

Yago Obispo Gerster

6. Mai 2024

Inhaltsverzeichnis

1	Einleitung und Motivation	1
1.1	Physikalische Grundlagen	2
1.1.1	Thermische Neutronenquelle	2
1.1.2	Aktivität	2
1.1.3	Beispiel: Aktivierung von Indium und Silber	3
2	Messprotokoll	3
2.1	Ergänzung der Durchführung	6
3	Auswertung	6
3.1	Zerfall der Silberisotope	6
3.1.1	Untergrundmessung	6
3.1.2	Kurvenanpassung	7
3.1.3	Berücksichtigung des Untergrundfehlers	8
3.2	Indiumzerfall	10
3.2.1	Untergrundmessung	10
3.2.2	Kurvenanpassung	10
3.2.3	Berücksichtigung des Untergrundfehlers	12
4	Zusammenfassung und Diskussion	12
4.1	Diskussion	13
5	Quellen und Python Code	16

1 Einleitung und Motivation

Das Ziel des Versuches besteht darin, durch Aktivierung von ^{115}In und $^{107/109}\text{Ag}$ zu radioaktiven Isotopen durch thermische Neutronen, die Halbwertszeit von den entstehenden Kernen zu bestimmen.

1.1 Physikalische Grundlagen

1.1.1 Thermische Neutronenquelle

Bei der Herstellung einer radioaktiven Quelle müssen stabile Isotope durch Kernreaktionen aktiviert werden. Dies geschieht am einfachsten indem Neutronen verwendet werden, da diese nicht geladen sind, somit keine Coulomb-Wechselwirkung aufweisen und deshalb vom Kern leicht eingefangen werden können.

Um Neutronen zu erzeugen, wird eine Quelle verwendet, die Berylliumspäne und einen α -Strahler enthält. Durch eine Kernreaktion



entstehen schnelle Neutronen. Ein Paraffinblock umgibt die Quelle und beim elastischen Stoss der Neutronen mit Wasserstoffkernen werden diese abgebremst bis sie thermische Energie erreicht haben. Der Wirkungsquerschnitt für den Eingang langsamer Neutronen ist bei den meisten Atomen gross.

1.1.2 Aktivität

Die *Aktivität* A ist eine Grösse die die Anzahl der Zerfälle pro Sekunde beschreibt. Diese hängt mit der Bestrahlungsdauer t wie folgt zusammen:

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

Das heisst, sie steigt monoton, bis die Anzahl an pro Sekunde die zerfallen gleich ist wie die Anzahl an neuen radioaktiven Kernen die pro Sekunde gebildet werden. Wird die Aktivierung beendet, so folgt die Aktivität dem Zerfallsgesetz:

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

mit Halbwertszeit

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

Die Halbwertszeit gibt an, wann die Hälfte einer Menge von radioaktiven Kernen zerfallen ist.

Bei letzteren Formeln steht λ immer für die sogenannte *Zerfallskonstante*. Deren Bedeutung kann man sich erschliessen indem man den Bestand N des Zerfallsgesetzes ableitet:

$$\dot{N} = -\lambda \cdot N \quad (5)$$

Obige Differentialgleichung verrät uns, dass die Zerfallskonstante der entscheidende Proportionalitätsfaktor der momentanen Änderungsrate des Bestandes ist.

Die *Lebensdauer* ist über

$$\tau = \frac{1}{\lambda} \quad (6)$$

definiert.

1.1.3 Beispiel: Aktivierung von Indium und Silber

Im heutigen Versuch befassen wir uns mit der Aktivierung von Indium und Silber.

Bestrahlt man Indium mit Neutronen, so geht das stabile Isotop ^{115}In in das radioaktive, β strahlende Isotop ^{116}In über. Genauer betrachtet entstehen zwei unterschiedliche sogenannte *Isomere*, d.h. Nuklide mit gleicher Protonen- und Neutronenanzahl aber unterschiedlichen Energieniveaus (hier: ^{116}In im Grundzustand und ^{116m}In in einem metastabilen Zustand).

Natürliches Silber besteht dabei zu 51% aus ^{107}Ag und zu 49% aus ^{109}Ag . Entsprechend werden bei der Neutronenaktivierung zwei unterschiedliche radioaktive, β -Strahler Isotope erzeugt: ^{108}Ag und ^{110}Ag .

Vor allem interessant ist hierbei, dass die Halbwertszeiten um etwa einen Faktor 6 voneinander abweichen, weshalb durch Variation der Aktivierungszeit, das Aktivitätsverhältnis verändert werden kann. Kurze Aktivierung führt vor allem zur Entstehung von ^{110}Ag während bei längerer Aktivierungszeit mehr ^{108}Ag erzeugt wird (siehe dazu Abbildung 1).

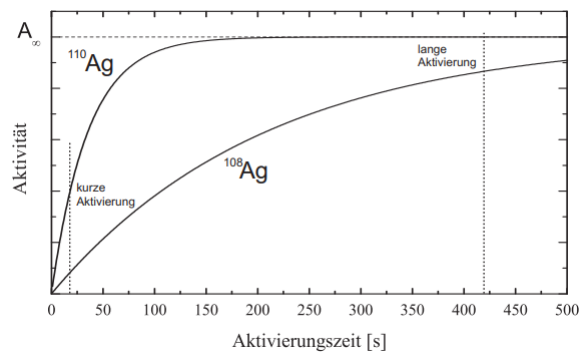


Abbildung 1: Aktivität bei unterschiedlichen Aktivierungszeiten (Quelle:Praktikumsskript Seite 2 V252)

2 Messprotokoll

Physikalisches Laboratorium

Florian Plaswig , WS 23/24

Versuchsprotokoll 252

Lennart Bederke, Yago Gerster

21.02.2024

09-12 Uhr

Inhaltsverzeichnis

1	Materialien	2
2	Versuchsaufbau	2
3	Halbwertszeit von Silber	2
4	Halbwertszeit von Indium	2

1 Materialien

1. Geiger-Müller Zählrohr mit Betriebsgerät
2. Externer Impulszähler
3. PC mit Drucker
4. Neutronenquelle
5. Präparatehalterung
6. Indium- und Silberbleche

2 Versuchsaufbau

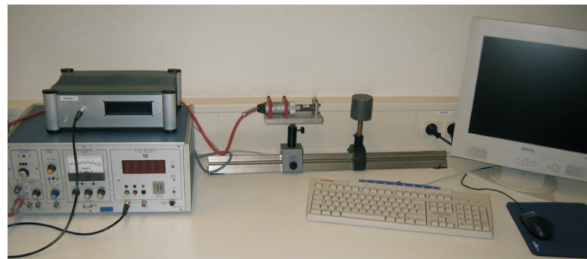


Abbildung 1: Versuchsaufbau

3 Halbwertszeit von Silber

Präparat: Silber

Betriebsspannung: $530 \pm 5\text{V}$

Untergrundmessung: Torzeit: 10s, Messzeit: 8min

Silbermessung: Torzeit: 10s, Messzeit: 7min

4 Halbwertszeit von Indium

Präparat: Indium

Betriebsspannung: $530 \pm 5\text{V}$

~~Untergrund~~ messung: Torzeit: ~~10s~~, Messzeit: 50min

Indium 120s

2.1 Ergänzung der Durchführung

Ergänzend zum Messprotokoll wird die Vorgehensweise während des Versuches dokumentiert um eine möglichst hohe Reproduzierbarkeit der Versuchsbedingungen zu erreichen.

Im ersten Schritt soll die Halbwertszeit von Silber bestimmt werden. Dazu wird am Betriebsgerät für das Geiger-Müller-Zählrohr eine Spannung von etwa $(530 \pm 5)V$ eingestellt. Um die Zerfälle möglichst ohne äussere Einflüsse zu messen wird ausserdem zuerst eine Untergrundmessung durchgeführt. Dafür wird mit einer Torzeit von $10s$ über ein Zeitraum von $8min$ gemessen und die Daten in Form einer Textdatei abgespeichert.

Die zur Verfügung stehende Silberbleche werden 7 Minuten lang in der thermischen Neutronenquelle aktiviert und dann sehr schnell zum Zählrohr gebracht, da bereits in den ersten Sekunden ein signifikanter Abfall messbar ist. Dabei wird unter einer Torzeit von $10s$, $400s$ lang gemessen. Insgesamt werden vier Mal die gleichen Silbermessungen durchgeführt, wobei zwischendurch immer die Aktivierungszeit gewartet wurde.

Im zweiten Versuchsschritt sind wir an der Halbwertszeit von Indium interessiert. Dazu muss erneut eine Untergrundmessung durchgeführt werden, da das Messintervall nun auf $120s$ eingestellt wird. Diesmal messen wir den Untergrund $6min$ lang. Die Indium-Präparate wurden bereits über Nacht aktiviert, da diese mehr Zeit zur Aktivierung benötigen. Die Messung wird unter einem Zeitraum von $50min$ durchgeführt.

3 Auswertung

Für die Auswertung werden, sofern nicht anders angegeben, alle Fehler gemäss des Gaußsschem Fehlerfortpflanzungsgesetz ermittelt. Die Fehler werden bei den relevanten Formeln weiterhin explizit angegeben.

3.1 Zerfall der Silberisotope

Im ersten Teil des Versuches ist das Ziel die Bestimmung der Zerfallskonstante und Lebensdauer für Silber.

3.1.1 Untergrundmessung

Für den Untergrund bilden wir aus allen gemessenen Anzahlen an Zerfällen pro $10s$ den Mittelwert, wobei für den Fehler der Standardfehler des Mittelwertes verwendet wird. Die Messung wird weiterhin mit 4 multipliziert, da wir 4 Messreihen für Silber durchgeführt haben und diese anschliessend aufaddieren. wir erhalten im Mittel

$$N_{\text{Untergrund}} = 14,6 \pm 0,9 \quad (7)$$

Zerfälle jede $10s$.

Da diese Zahl nicht vernachlässigbar klein ist, wird diese vermutlich einen Ein-

fluss auf unsere Messung haben, weshalb der Untergrund berücksichtigt werden muss.

3.1.2 Kurvenanpassung

Die vier Messreihen werden aufeinanderaddiert, was in Python elegant gemacht werden kann, indem jede Reihe als Numpy-Array abgespeichert wird und alle miteinander addiert werden (für technische Details siehe Python-Code im Anhang). Der Fehler wird als die Wurzel der Zerfälle abgeschätzt. Wir erhalten die Messpunkte in Abbildung 2. An diese Fitten wir im Anschluss die Überlagerung

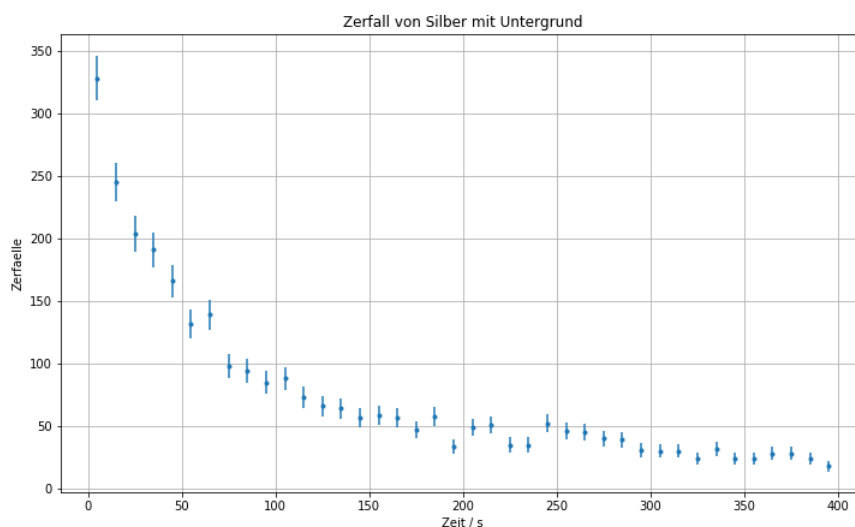


Abbildung 2: Silbermesspunkte

von zwei Exponentialfunktionen:

$$fitfunc(x) = A_1 e^{-\lambda_1 x} + A_2 e^{-\lambda_2 x} + y_0 \quad (8)$$

Dabei sind zwei Exponentialfunktionen nötig, da - wie in den Grundlagen bereits besprochen - natürliches Silber aus zwei Isotopen besteht. Die erste Zerfallskonstante λ_1 bezieht sich somit auf ^{110}Ag und λ_2 auf ^{108}Ag . y_0 beschreibt den Untergrund.

Mit der Python-Funktion `curve_fit()` werden die freie Parameter optimal an die Messpunkte angepasst, sodass wir den Verlauf in Abbildung 3 erhalten. Dabei wurde eine logarithmische Skala gewählt um den Verlauf deutlicher zu machen. Die Fitparameter ergeben sich dabei zu

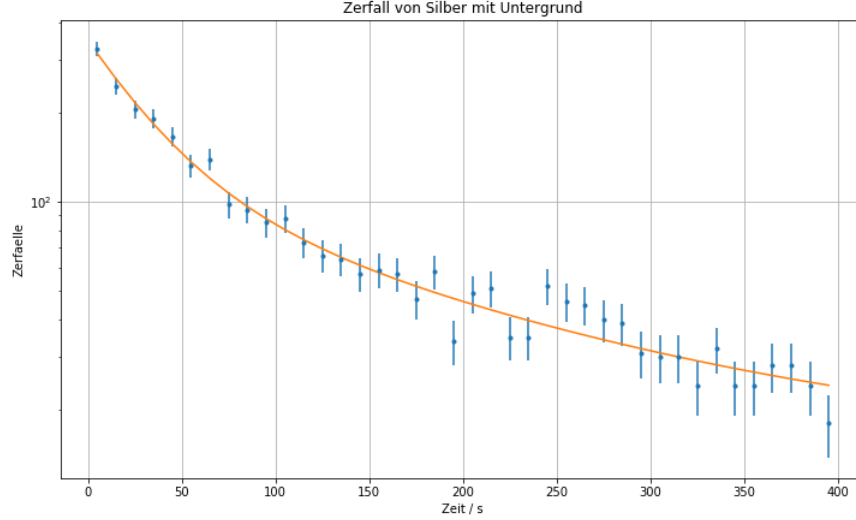


Abbildung 3: Silberanpassung mit Untergrund

$$A_1 = (232 \pm 23) \quad (9)$$

$$A_2 = (103 \pm 23) \quad (10)$$

$$\lambda_1 = (2,9 \pm 0,5) \cdot 10^{-2} \frac{1}{s} \quad (11)$$

$$\lambda_2 = (6,0 \pm 0,9) \cdot 10^{-3} \frac{1}{s} \quad (12)$$

Die Güte dieses Fits wird anhand der χ^2 -Methode in der Diskussion genauer untersucht.

3.1.3 Berücksichtigung des Untergrundfehlers

Bisher wurde der Fehler des Untergrunds noch nicht berücksichtigt, weshalb im Anschluss die Anpassung zwei Male wiederholt wird, wobei einmal y_0 der 1σ Fehler des Untergrunds abgezogen wird und einmal hinzuaddiert wird. Damit erhalten wir den Verlauf in Abbildung 4. Da anhand des Graphen die Abweichung durch den Untergrundstrom aufgrund ihrer Grösse nicht beobachtet werden kann, wurde in einen Abschnitt des Graphen hereingezoomt, was im oberen rechten Fenster visualisiert ist.

Dabei erhalten wir für die Unterschätzung bzw. Überschätzung für beide Zerfallskonstanten jeweils $\lambda_{\text{Unterschaetzung}}$ bzw. $\lambda_{\text{Ueberschaetzung}}$. Der Fehler der Zerfallskonstante wird damit über quadratische Addition des Untergrundfehlers

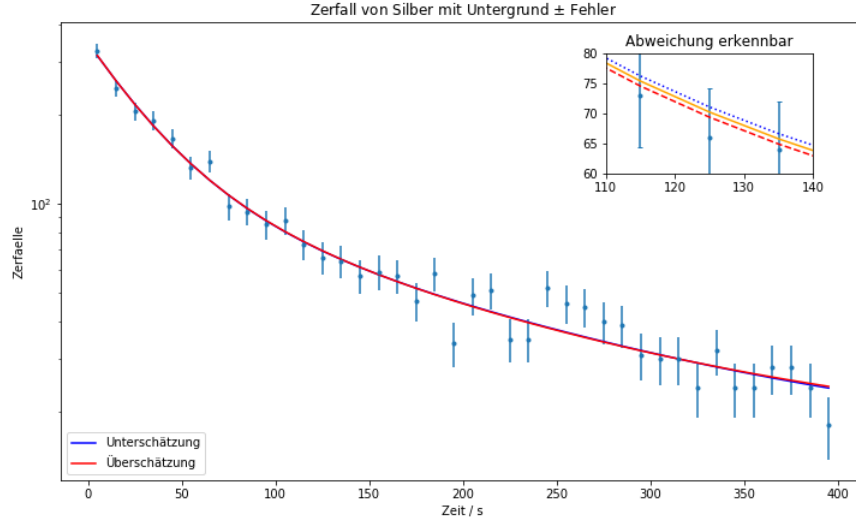


Abbildung 4: Silberverlauf unter Berücksichtigung Untergrundfehler

und des Fehlers davor berechnet:

$$\Delta\lambda_{min,1/2} := |\lambda_{1/2} - \lambda_{min,1/2}| \quad (13)$$

$$\Delta\lambda_{max,1/2} := |\lambda_{1/2} - \lambda_{max,1/2}| \quad (14)$$

$$\Delta\lambda_{neu,1/2} = \sqrt{\left(\frac{\Delta\lambda_{min,1/2} + \Delta\lambda_{max,1/2}}{2}\right)^2 + (\Delta\lambda_{1/2})^2} \quad (15)$$

Damit erhalten wir durch Verwendung der Formeln

$$T_{1/2} = \frac{\ln(2)}{\lambda} \quad (16)$$

$$\Delta T_{1/2} = \frac{\ln(2)\Delta\lambda_{neu}}{\lambda^2} \quad (17)$$

$$\tau = \frac{1}{\lambda} \quad (18)$$

$$\Delta\tau = \frac{\Delta\lambda}{\lambda^2} \quad (19)$$

die Halbwertszeit $T_{1/2}$ und die Lebensdauer τ .
Als Ergebnisse erhalten wir:

$$\lambda_1 = (29 \pm 5) \cdot 10^{-3} \frac{1}{s} \quad (20)$$

$$\lambda_2 = (6,0 \pm 0,9) \cdot 10^{-3} \frac{1}{s} \quad (21)$$

$$T_{1/2,1} = (24 \pm 4)s \quad (22)$$

$$T_{1/2,2} = (115 \pm 18)s \quad (23)$$

$$\tau_1 = (35 \pm 6)s \quad (24)$$

$$\tau_2 = (166 \pm 26)s \quad (25)$$

3.2 Indiumzerfall

Das gleiche Verfahren wird anschliessend für Indium wiederholt.

3.2.1 Untergrundmessung

Dabei musste erneut eine Untergrundmessung durchgeführt werden, da wir nun mit einem Messintervall von 120s die Anzahl an Zerfällen aufnehmen wollen.
Wir erhalten

$$N_{\text{Untergrund}} = 38 \pm 3 \quad (26)$$

Zerfälle jede 120s.

3.2.2 Kurvenanpassung

Die Messpunkte der Indiummessung sind in Abbildung 5 dargestellt. Dieses Mal ist die Anpassungsfunktion nur eine Exponentialfunktion, da wir bei Indium im Gegensatz zu Silber nur eine Halbwertszeit bestimmen müssen:

$$fitfunc(x) = A_1 \cdot e^{-\lambda_1 x} + y_0 \quad (27)$$

Für die Anpassung werde ich den ersten Messwert ausblenden, da dieser wie in Abbildung 5 erkennbar ist, deutlich vom Verlauf der anderen Messwerte abweicht. Diese Ausblendung kann auch aus einer theoretischen Perspektive gerechtfertigt werden: Bekanntlich zerfällt ^{116}In in β^+ und β^- , wobei ersteres nur für eine sehr kurze Zeit gültig ist und somit für den restlichen Verlauf vernachlässigt wird. Dies liefert eine mögliche Erklärung für die Abweichung des ersten Wertes von dem des Restes.

Wir erhalten den Verlauf in Abbildung 6. Für die Fitparameter ergeben sich:

$$A_1 = 1050 \pm 15 \quad (28)$$

$$\lambda_1 = (230 \pm 9) \cdot 10^{-6} \frac{1}{s} \quad (29)$$

Die Fitwahrscheinlichkeit wird in der Diskussion untersucht.

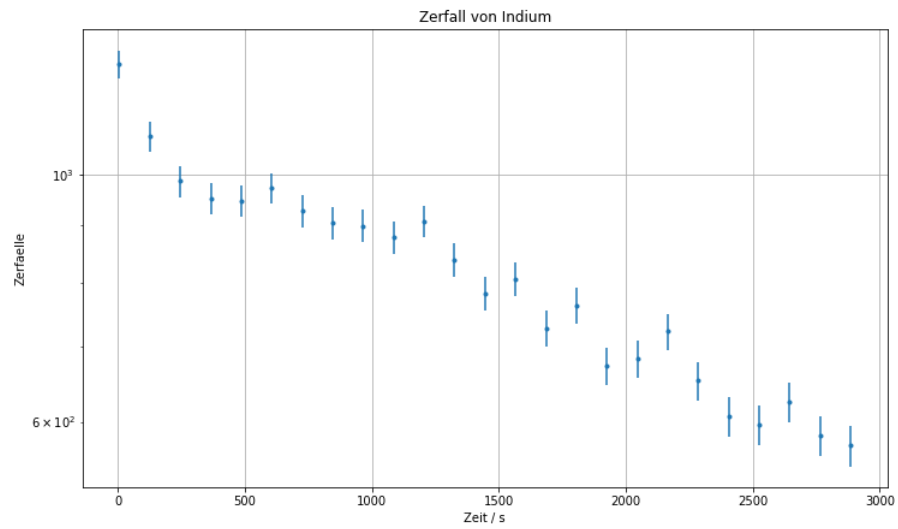


Abbildung 5: Indiummesspunkte

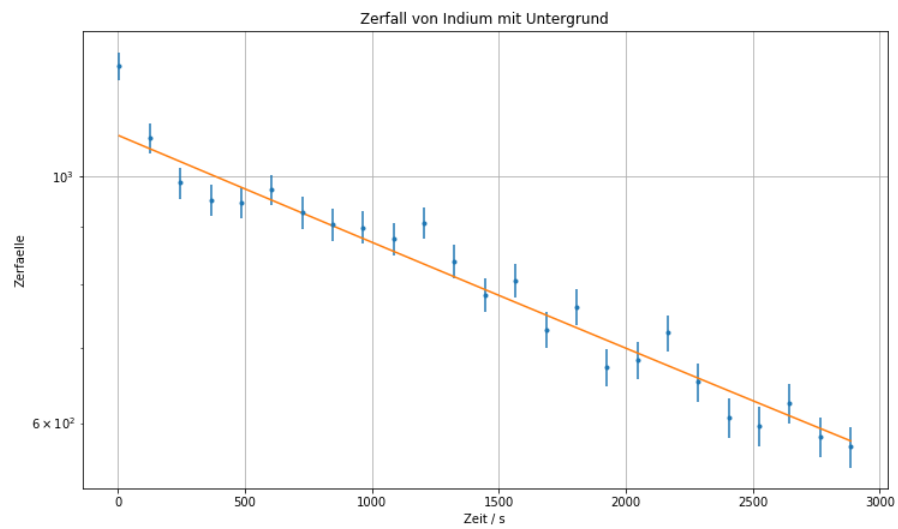


Abbildung 6: Kurvenanpassung an Indiumzerfall

3.2.3 Berücksichtigung des Untergrundfehlers

Analog zu der Vorgehensweise bei Silber addiere ich zunächst einmal den 1σ -Fehler des Untergrunds zum gemessenen Untergrund und subtrahiere diesen anschliessend.

Ich erhalte den Verlauf in Abbildung 7. Dabei wurde erneut im rechten oberen

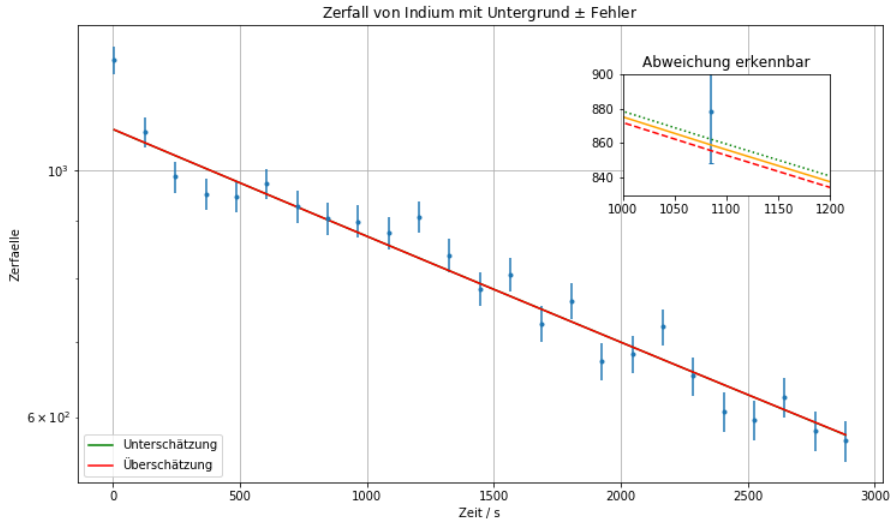


Abbildung 7: Indiumanpassung mit Untergrund

Bereich ein Fenster angegeben, welches einen vergrösserten Bereich des Graphens darstellt damit die Abweichung der Über- bzw. Unterschätzung erkennbar sind.

Für die Zerfallskonstante, Halbwertszeit und Lebensdauer erhalte ich:

$$\lambda_1 = (230 \pm 9) \cdot 10^{-6} \frac{1}{s} \quad (30)$$

$$T_{1/2} = (3010 \pm 120)s \quad (31)$$

$$\tau = (4340 \pm 180)s \quad (32)$$

4 Zusammenfassung und Diskussion

Zusammenfassend haben wir in diesem Versuch eine Methode zur radioaktiven Aktivierung von Isotopen mit einer thermischen Neutronenquelle kennengelernt und uns am Beispiel von Silber und Indium mit den Eigenschaften der entstehenden radioaktiven Isotopen befasst - wie Zerfallskonstante, Halbwertszeit und Lebensdauer.

Dazu wurde für beide Stoffe eine ähnliche Vorgehensweise verwendet. Als erstes wurde jeweils eine Untergrundmessung durchgeführt, um später den gemachten Fehler korrigieren zu können. Die mittlere Anzahl an Zerfällen lag für Silber bei $N_{\text{Untergrund}} = 14,6 \pm 0,9$ pro 10s und für Indium bei $N_{\text{Untergrund}} = 38 \pm 3$ pro 120s.

Daher dass wir nach dem Zerfallsgesetz ein exponentielles Abklingen der Zerfälle mit der Zeit erwartet haben, wurde eine allgemeine Funktion der Form $Ae^{-\lambda x} + \text{Untergrund}$ an die Messwerte angepasst. Im Falle von Silber sogar eine Überlagerung zweier solcher Funktionen, da uns bekannt war, dass radioaktives Silber aus zwei Isotopen mit unterschiedlichen Zerfallskonstanten besteht.

Um den gemachten Fehler der Zerfallskonstante zu verfeinern, musste der Untergrundfehler berücksichtigt werden. Dafür wurde die Anpassung zwei weitere Male durchgeführt, wobei die anzupassende Funktion um den 1σ -Fehler des Untergrunds in positive bzw. negative Richtung verschoben wurde.

Dies führte insgesamt zu allen in Tabelle 1 aufgelisteten Grössen.

	^{110}Ag	^{108}Ag	Indium
Zerfallskonstante [$\frac{1}{s}$]	$(29 \pm 5) \cdot 10^{-3}$	$(6,0 \pm 0,9) \cdot 10^{-3}$	$(230 \pm 9) \cdot 10^{-6}$
Halbwertszeit [s]	24 ± 4	115 ± 18	3010 ± 120
Lebensdauer [s]	$(35 \pm 6)s$	166 ± 26	4340 ± 180

Tabelle 1: Alle relevanten bestimmten Grössen

4.1 Diskussion

Im Anschluss werden die erstellten Graphiken genauer analysiert - insbesondere deren Fitwahrscheinlichkeiten - und die ermittelten Werte werden mit den Literaturwerten verglichen. Dabei werden weiterhin die Fehlerquellen untersucht und einige Vorschläge präsentiert, wie man durch eine Verbesserung der Durchführung die Ergebnisse verbessern könnte.

Für die σ -Abweichung wird die bekannte Formel

$$\frac{|Wert1 - Wert2|}{\sqrt{Fehler1^2 + Fehler2^2}} \quad (33)$$

verwendet.

Als erstes will ich betonen, dass die ermittelten Fehler der Zerfallskonstanten, sowohl bei Silber als auch bei Indium, durch die Berücksichtigung des Untergrundfehlers kaum verändert werden, da der statistische Fehler bereits sehr gross ist. Die Abweichung der Fehler bei Silber beträgt (absolut) $1,1 \cdot 10^{-5} \frac{1}{s}$ bzw. $3,5 \cdot 10^{-5} \frac{1}{s}$ und bei Indium $5,24 \cdot 10^{-8} \frac{1}{s}$. Diese Veränderung entspricht bei Silber 0,2% bzw. 3,7% des Gesamtfehlers und bei Indium 0,57%. Unter Berücksichtigung der signifikanten Stellen bleibt der Unterschied zunächst unbemerkbar. Diesen Sachverhalt konnten wir auch grafisch feststellen, da wir sowohl in Abbildung 6 als auch in Abbildung 3 extra ein Fenster in der Grafik

einführen mussten, welches in einen Bereich des Graphens hereinzoomte, damit überhaupt die Abweichung erkennbar war. Der Fehler des Untergrundes spielte somit keine signifikante Rolle. Jedoch kann diese Aussage wahrscheinlich nicht auf alle radioaktiven Substanzen erweitert werden. Für solche mit enormen Halbwertszeiten beispielsweise (was für viele Isotope für die wir uns teilweise in der modernen Physik interessieren der Fall ist) ist eine äusserst genaue Messung des Untergrundes notwendig und entsprechend auch eine bessere Isolation erforderlich. In unserem Fall waren die Halbwertszeiten klein genug, sodass wir keine besondere Isolierung benötigten.

Bei der Kurvenanpassung bei Silber wurde eine Überlagerung des Untergrundes mit zwei Exponentialfunktionen unterschiedlicher Zerfallskonstanten gewählt, da bereits bekannt war, dass natürliches Silber aus zwei Isotopen bestand. Analysiert man die Güte des Fits Abbildung 4 mit der χ^2 -Methode, so erhält man die folgenden Werte in Tabelle 2. Die χ^2 -Summe wurde dabei nach

χ^2	32,17
χ^2_{red}	0,89
<i>Fitwahrscheinlichkeit</i>	65,0%

Tabelle 2: Güte des Fits bei Silber

$$\chi^2 = \sum_i^N \left(\frac{Funktionswert_i - Messwert_i}{Fehler_i} \right)^2 \quad (34)$$

berechnet.

Die Fitwahrscheinlichkeit beschreibt dabei die Wahrscheinlichkeit, bei einer Wiederholungsmessung ein χ^2 -Wert zu erhalten, welcher grösser oder gleich dem χ^2 -Wert ist. Hier ist dabei ist die Fitwahrscheinlichkeit von 65% ziemlich gut, weshalb unsere Wahl von zwei Exponentialfunktionen für den Fit geeignet war und die Untergrundmessung wahrscheinlich auch keine groben Fehler beinhaltet.

Die gleiche Fitanalyse kann für Indium wiederholt werden. Hier wurde in Abbildung 6 eine Exponentialfunktion mit dem Untergrund angepasst. Wir erhalten die Werte in Tabelle 3. Hier ist der Fit also etwas schlechter als im Silber-Fall,

χ^2	28,09
χ^2_{red}	1,40
<i>Fitwahrscheinlichkeit</i>	11,0%

Tabelle 3: Güte des Fits bei Indium

jedoch erweist sich unser Anpassungsmodell immer noch als gut genug um die Physik des Indiumszerfalls zu beschreiben.

Zuletzt werden die ermittelten Halbwertszeiten mit den Literaturwerten verglichen (Tabelle 4). Diese werden der Nuklidkarte entnommen (Quelle: Praktikumsskript Versuch 252 Seite 6).

Halbwertszeit von Isotop	Versuchswert [s]	Literaturwert	Abs. Abweichung	σ -Abweichung
^{110}Ag	24 ± 4	24,6 s	0,5s	0.12
^{108}Ag	115 ± 18	2,41 min	30s	1.7
Indium	3010 ± 120	54 min	230s	1,9

Tabelle 4: Vergleich der Halbwertszeiten mit Literaturwerten

Wir können dabei also feststellen, dass die Halbwertszeiten in allen drei Fällen eine Sigma-Abweichung unter der üblichen Signifikanzgrenze von 3σ besitzen und somit nicht signifikant sind. Damit haben wir in dem Versuch sinnvolle Messwerte erhalten, welche mit der Theorie kompatibel sind. Die Abweichungen lassen sich dabei grösstenteils durch statistische Effekte deuten. Zum einen wurde bereits in einem vorherigen Versuch der statistische Charakter des radioaktiven Zerfalls bestätigt, welcher die statistischen Abweichungen unserer Messung verschärft.

Dabei ist jedoch auffallend, dass ^{108}Ag eine deutlich höhere Sigma-Abweichung als das andere Silber-Isotop besitzt, obwohl beide ja vom selben Fit stammen. Dabei liegt das weiterhin auch nicht an der Grösse des gewählten Fehlers, da ^{110}Ag einen relativen Fehler von 17% aufweist und das andere Isotop einen von 16%, also relativ ähnliche Fehlergrössen. Dies könnte dadurch erklärt werden, dass die Halbwertszeit bei ^{108}Ag deutlich grösser ist, d.h. der Zerfall langsamer verläuft. Im Versuch haben wir zwar vier unterschiedliche Silbermessungen durchgeführt, jedoch haben wir bei allen dieselben Messzeiten verwendet. Vielleicht wäre es sinnvoll, stattdessen für unterschiedliche Zeiten zu messen - einmal für eine die sich dem ersten Isotop und einmal eine die sich dem zweiten Isotop anpasst. Damit könnte die Genauigkeit der Silberergebnisse gesteigert werden. Bei Indium ist die Halbwertszeit ebenfalls sehr gross worunter wahrscheinlich auch die Genauigkeit unseres Ergebnisses in geringem Masse gelitten hat.

Weiter können wir an der Genauigkeit der Ergebnisse von Indium feststellen, dass die Vernachlässigung des sehr schnellen Zerfalls am Anfang keine weiterführenden Konsequenzen mitsich gebracht hat, was daran liegt, dass der Literaturwert für die Halbwertszeit dieses schnell zerfallenden Indiumisotops bei 14s liegt und somit gegenüber der verwendeten Torzeit vernachlässigbar ist.

Insgesamt empfand ich den Versuch als sehr lehrreich. Obwohl letztendlich vielmehr der Zerfall der Isotope eine bedeutende Rolle gespielt hat und weniger deren Aktivierung durch thermische Neutronen, fand ich das theoretisch erlernte Wissen zur radioaktiven Aktivierung besonders interessant, da ich mich davor nie damit beschäftigt hatte, wie radioaktive Quellen überhaupt künstlich aktiviert werden können.

5 Quellen und Python Code

- PAP 2.1 Anleitung der Universität Heidelberg

Versuch252

March 2, 2024

1 Auswertung Versuch 252

1.1 Yago Obispo Gerster | mn304

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

unterg = np.loadtxt('252/data/untergrund (1).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)
```

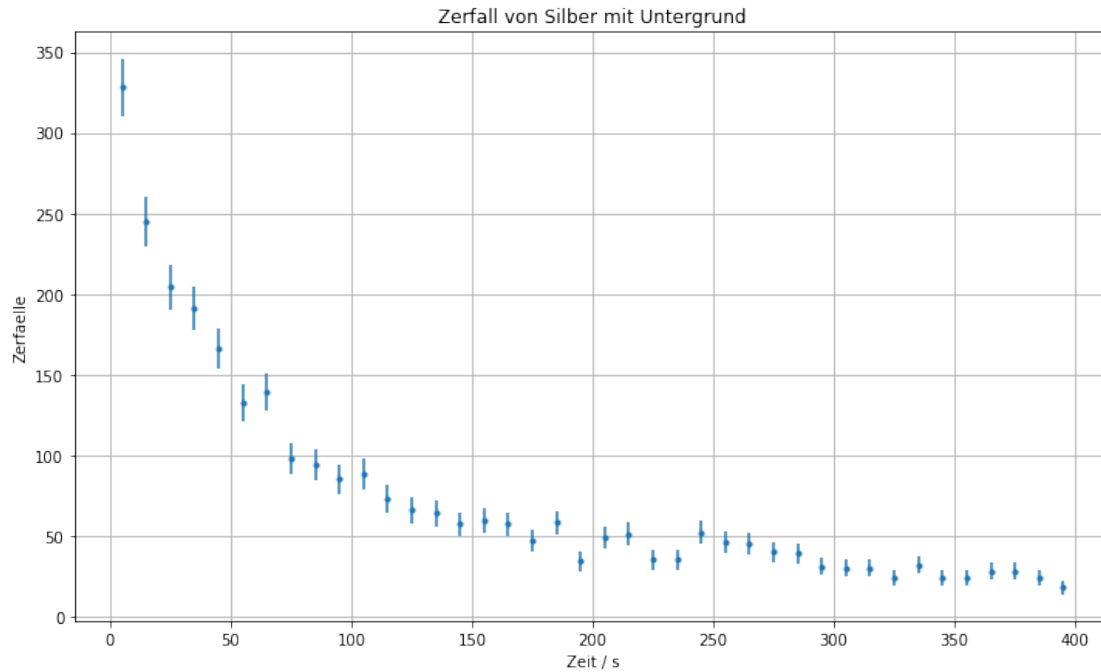
Mittelwert: 14.583333333333334 Fehler: 0.8935694909659714

```
[2]: #Bestimmung Zerfallskonstante
n1 =np.loadtxt('252/data/silber1.txt', usecols=[1],delimiter = ",",skiprows=1)
n2 =np.loadtxt('252/data/silber2.txt', usecols=[1],delimiter = ",",skiprows=1)
n3 =np.loadtxt('252/data/silber3.txt', usecols=[1],delimiter = ",",skiprows=1)
n4 =np.loadtxt('252/data/silber4.txt', usecols=[1],delimiter = ",",skiprows=1)

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("252/SilberohneKurve.png",format="png")
```



```
[3]: #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('252/Silber.png',format='png')

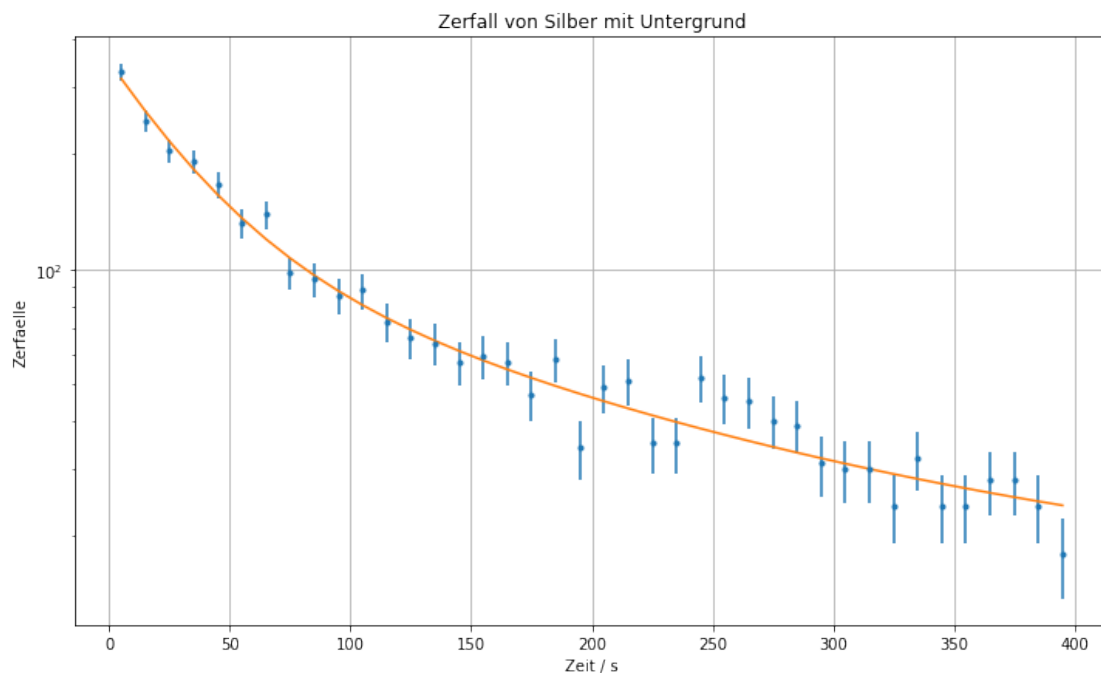
#Fitparameter
print("A1=",popt[0], ", Standardfehler=", np.sqrt(pcov[0][0]))
print("l1=",popt[1], ", Standardfehler=", np.sqrt(pcov[1][1]))
print("A2=",popt[2], ", Standardfehler=", np.sqrt(pcov[2][2]))
print("l2=",popt[3], ", Standardfehler=", np.sqrt(pcov[3][3]))
```

```

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= 231.83311209928468 , Standardfehler= 23.190843100053407
 l1= 0.02882086465732568 , Standardfehler= 0.005261414603358678
 A2= 103.06708040732667 , Standardfehler= 23.38369434844112
 l2= 0.006037710367733205 , Standardfehler= 0.0008944565166232641



```

[4]: #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= 32.169587571951624

```
chi2_red= 0.893599654776434
Wahrscheinlichkeit= 65.0 %
```

```
[5]: #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  $\pm$  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")

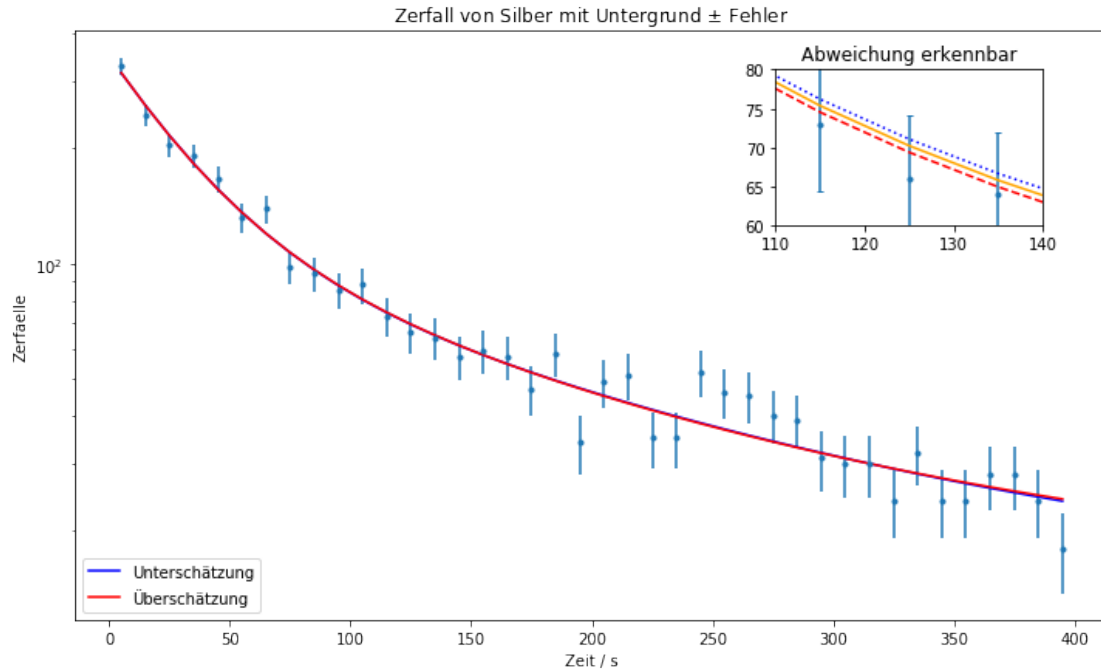
#Fenster zur besseren Visualisierung des Unterschiedes
a = plt.axes([.65, .63, .2, .2], facecolor = 'white')
plt.plot(t, fit_func(t, *popt_min), color = 'blue', ls = 'dotted')
plt.plot(t, fit_func(t, *popt_max), color = 'red', ls = '--')
plt.plot(t, fit_func(t, *popt), color = 'orange')
plt.errorbar(t, N, yerr = Fehler_N, fmt = '.', capsize = 2)
plt.xlim(110, 140)
plt.ylim(60, 80)
plt.title('Abweichung erkennbar')
plt.savefig("252/SilbermitUntergrundfehler.png",format="png")

```

```

A1min= 234.46210932873387 , Standardfehler= 22.487368550418964
l1min= 0.02850017688268341 , Standardfehler= 0.005039139968242843
A2min= 101.04256037821959 , Standardfehler= 22.352978432402786
l2min= 0.005795143847900907 , Standardfehler= 0.000861428118318056
A1max= 228.90042989929927 , Standardfehler= 23.989604201389838
l1max= 0.029181283659841704 , Standardfehler= 0.005516767833294191
A2max= 105.419313513687 , Standardfehler= 24.522517874507198
l2max= 0.006299025197570174 , Standardfehler= 0.000929741817627534

```



```
[6]: #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_l_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  0.02882 +/- 0.00527
Zerfallskonstante l_2  0.0060377 +/- 0.000929
Halbwertszeit T_1    24.05 +/- 4.4
Halbwertszeit T_2   114.8 +/- 17.7
Lebensdauer tau_1   34.697 +/- 6.347
```

Lebensdauer tau_2 165.626 +/- 25.5

```
[7]: #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))
literaturVergleich("Halbwertszeit Silber 1",np.log(2)/l_1,np.log(2) * feh_l_1 /
    ↳(l_1 ** 2),24.6,0)
literaturVergleich("Halbwertszeit Silber 2",np.log(2)/l_2,np.log(2) * feh_l_2 /
    ↳(l_2 ** 2),2.41*60,0)
```

Halbwertszeit Silber 1 :
Absolute Abweichung: 0.5498131370683481
Sigma-Abweichung: 0.12496640978837525
Halbwertszeit Silber 2 :
Absolute Abweichung: 29.797013711643146
Sigma-Abweichung: 1.686374321325686

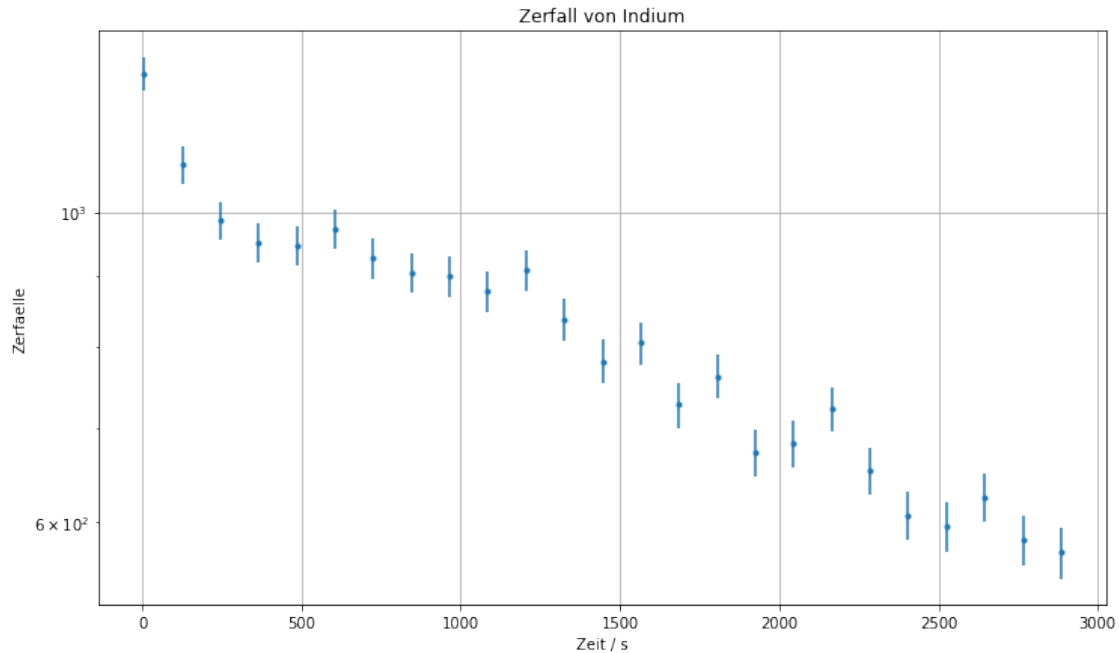
```
[8]: #Aufgabe 2: Indiumzerfall
unterg =np.loadtxt('untergrund120.txt', usecols=[1],skiprows=4)

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

Mittelwert: 37.75 Fehler: 3.2282928925362393

```
[9]: #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("252/IndiumohneFit.png",format="png")
```



```
[10]: #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('252/IndiummitUntergrund.png',format='png')

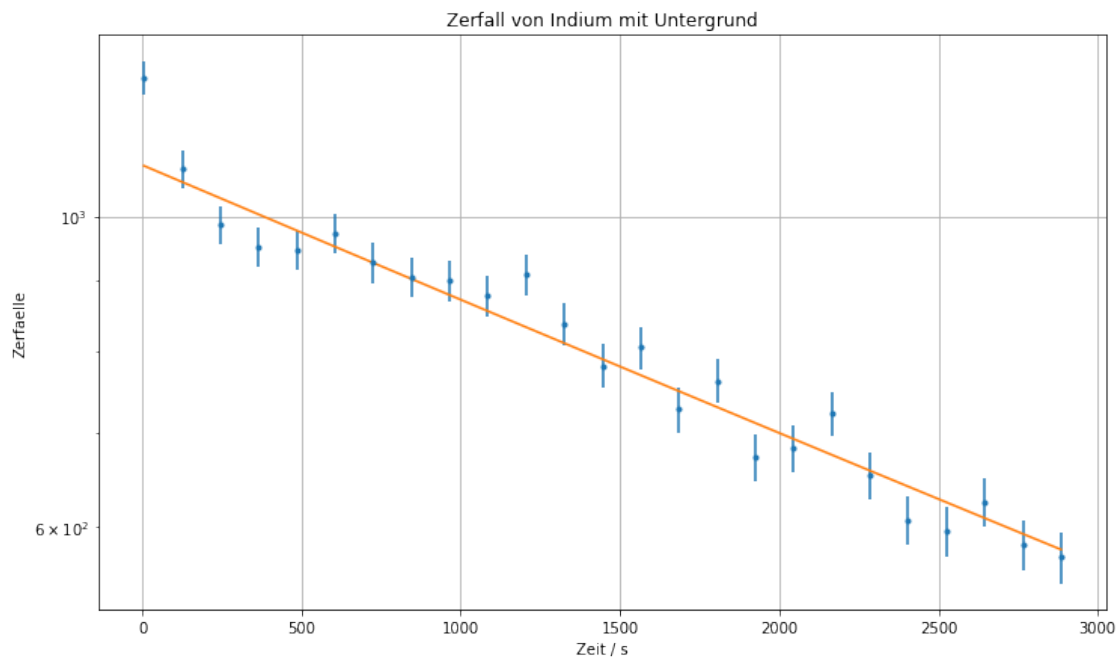
A_1 = popt[0]
sig_A_1 = np.sqrt(pcov[0][0])
l_1 = popt[1]
sig_l_1 = np.sqrt(pcov[1][1])
```



```
#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= 1050.1612249741117 , Standardfehler= 15.222210484161797
l1= 0.00023026304559071274 , Standardfehler= 9.210699359561367e-06
```



```
[11]: #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= 28.085885551206733
chi2_red= 1.4042942775603366
Wahrscheinlichkeit= 11.0 %
```

```

[12]: #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 $\pm$ 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(1000, 1200)
plt.ylim(830,900)
plt.title('Abweichung erkennbar')
plt.savefig("252/IndiummitFehlerUntergrund.png",format = "png")

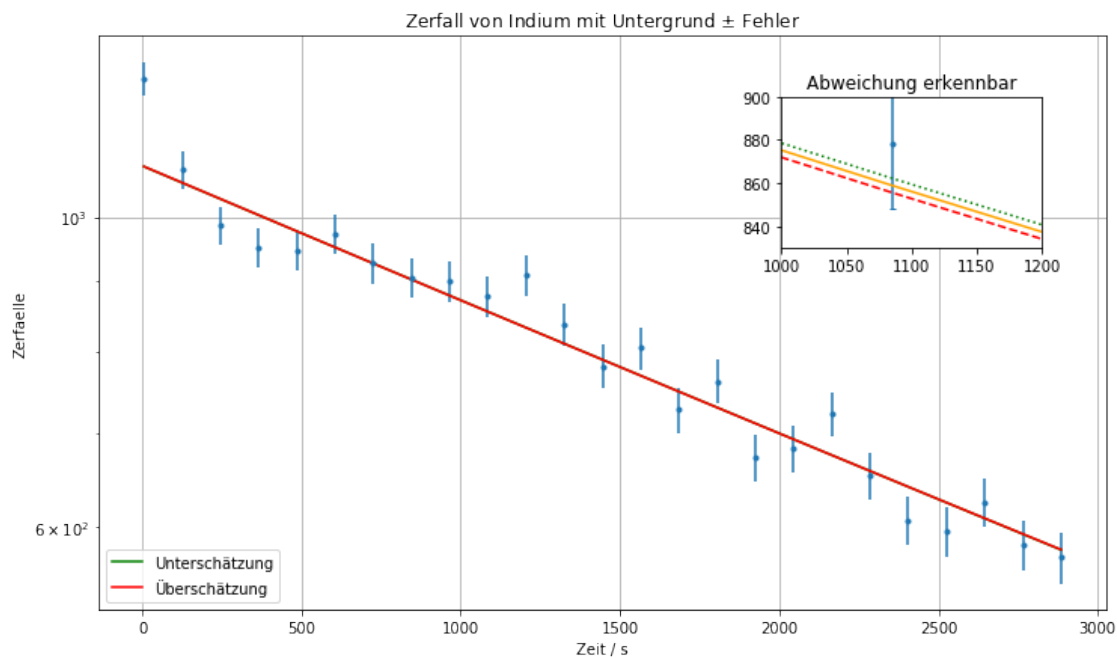
```

A1= 1053.2540576473384 , Standardfehler= 15.211109525298207
 l1= 0.0002292837164907388 , Standardfehler= 9.16984437032992e-06
 A1= 1047.069465860701 , Standardfehler= 15.233408145401093
 l1= 0.00023125071131660128 , Standardfehler= 9.251927842268567e-06

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:13: RuntimeWarning:
 overflow encountered in exp

del sys.path[0]

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:29: RuntimeWarning:
 overflow encountered in exp



```

[13]: #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  $\lambda_1$  ", np.round( $\lambda_1$ ,5), "+/-", np.round(feh_l_1,8))
print("Halbwertszeit  $T_1$  ", np.round(np.log(2)/ $\lambda_1$ ,1), "+/-", np.round(np.log(2) *
    ↳ feh_l_1 / ( $\lambda_1$  ** 2),1))
print("Lebensdauer  $\tau_1$  ", np.round((1/ $\lambda_1$ ),1), "+/-", (np.round(1 * feh_l_1 /
    ↳ ( $\lambda_1$  ** 2),1)))

```

```

9.263058342333087e-06
Zerfallskonstante  $\lambda_1$   0.00023 +/- 9.26e-06
Halbwertszeit  $T_1$   3010.2 +/- 121.1
Lebensdauer  $\tau_1$   4342.9 +/- 174.7

```

```

[14]: #Vergleich mit Literaturwert und Sigma-Abweichung
literaturVergleich("Halbwertszeit In:", np.log(2)/ $\lambda_1$ , np.log(2) * feh_l_1 / ( $\lambda_1$ 
    ↳ ** 2), 54*60, 0)

```

```

Halbwertszeit In: :
Absolute Abweichung:  229.75934769837795
Sigma-Abweichung:  1.8973253891913928

```