

Name: Marius Pfeiffer
E-Mail: marius.pfeiffer@stud.uni-heidelberg.de

Matrikel-Nr.: 4188573

Betreut durch: Pia Sophie Haas

19.11.2024

Versuch 252: Aktivierung mit thermischen Neutronen

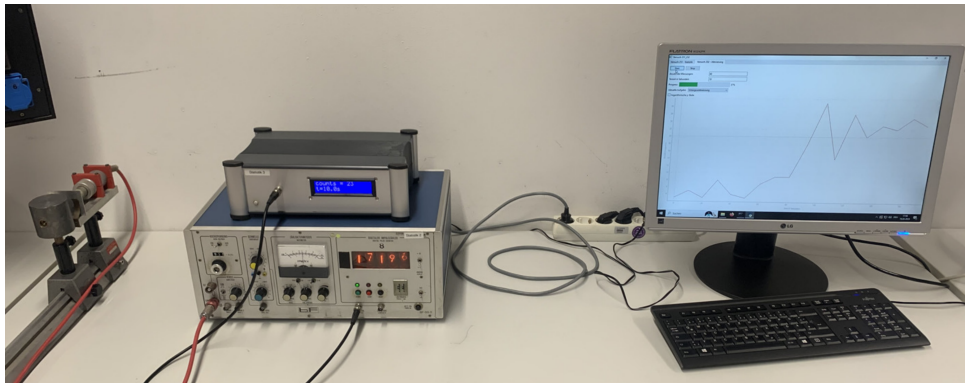


Abbildung 1: Versuchsaufbau

Inhaltsverzeichnis

1	Einleitung	2
1.1	Physikalische Grundlagen	2
1.2	Versuchsdurchführung	3
2	Messprotokoll	4
3	Auswertung	5
3.1	Bestimmung der Halbwertszeit von aktiviertem Silber	5
3.2	Bestimmung der Halbwertszeit von aktiviertem Indium	7
4	Zusammenfassung und Diskussion	9

1 Einleitung

Unter der Aktivierung versteht man die Herstellung radioaktiver Quellen durch die Bestrahlung stabiler Isotope mit verschiedenen Teilchen. In Versuch 252 werden Silber und Indium durch die Bestrahlung mit thermischen Neutronen aktiviert und deren Halbwertszeit bestimmt.

1.1 Physikalische Grundlagen

Neutronenquelle

Als Neutronenquelle werden im Versuch Berylliumspäne verwendet, welche α -Strahlung, also der Bestrahlung mit Helium-4-Kernen ausgesetzt sind. Durch die Reaktion



werden im Prozess freie Neutronen erzeugt. Diese sind zu diesem Zeitpunkt mit einer Energie von 1 bis 10 MeV noch zu schnell um für die Aktivierung verwendet zu werden. Bevor sie das Isotop zur Aktivierung erreichen, müssen die erzeugten Neutronen eine Paraffinschicht passieren. Hier werden sie durch die Kollision mit den Wasserstoffkernen des Paraffins abgebremst. Die langsamen Neutronen bei einer Energie von etwa 0.025eV (bei Zimmertemperatur) werden auch thermische Neutronen genannt, da sie sich mit der Umgebung im thermischen Gleichgewicht befinden.

Aktivierung mit thermischen Neutronen

Die thermischen Neutronen werden von den Atomkernen zur Aktivierung eingefangen. Die meisten Kerne haben hierfür einen großen Wirkungsquerschnitt. Ein anderer Vorteil der Verwendung von Neutronen für die Aktivierung ist, dass diese nicht von der Coulomb-Wechselwirkung betroffen sind. Durch den Einfang der Neutronen erhöht sich die Massenzahl, wodurch instabile, radioaktive Isotope entstehen können. Im Versuch werden das stabile Indium-Isotop ${}^{115}\text{In}$ und Silber, welches zu Teilen aus den stabilen Isotopen ${}^{107}\text{Ag}$ und ${}^{109}\text{Ag}$ besteht, aktiviert. Hierbei entstehen im ersten Fall die Isomere ${}^{116}\text{In}$ und ${}^{116m}\text{In}$ und im zweiten Fall die Isotope ${}^{108}\text{Ag}$ und ${}^{110}\text{Ag}$. Bei all diesen handelt es sich um radioaktive β -Strahler.

Aktivität $A(t)$

Die Zahl der Zerfälle pro Sekunde, genannt Aktivität A , eines Kerns während er aktiviert wird kann beschrieben werden durch die Funktion

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

Hier ist t die Bestrahlungsdauer, λ die Zerfallskonstante des Isotops und A_{∞} der Gleichgewichtszustand. Bis zum Gleichgewichtszustand nimmt die Aktivität mit der Bestrahlungsdauer zu, da die Zahl der Zerfälle nach dem Zerfallsgesetz proportional zur Zahl der zur Verfügung stehenden Kerne ist. Nach der Aktivierung verhält sich die Aktivität nach

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

Dieser Zusammenhang ergibt sich aus der zeitlichen Ableitung des Zerfallsgesetzes.

$$\frac{d}{dt} N_0 e^{-\lambda t} = \underbrace{-\lambda N_0}_{A_0} e^{-\lambda t} \quad (4)$$

Aus der Zerfallskonstante λ lässt sich mit

$$T_{1/2} = \frac{\ln 2}{\lambda} \quad (5)$$

die Halbwertszeit $T_{1/2}$ berechnen.

1.2 Versuchsdurchführung

Der Versuch setzt sich aus zwei Teilen zusammen, welche ähnlich ablaufen. Für die Messungen wird ein nahezu identischer Aufbau wie in Versuch 251 mit einem Geiger-Müller-Zählrohr eingesetzt. In allen Versuchsteilen wurde das Zählrohr bei einer Spannung von 550V betrieben.

Messen der Halbwertszeit von aktiviertem Silber. Bevor das aktivierte Silber in die Vorrichtung eingespannt wird, wird zunächst in einem Leerlauf die Untergrundzählrate bestimmt. Die Messung für diese läuft über 8 Minuten bei einer Torzeit von 10 Sekunden. Nun wird das Silber für 7 Minuten in der Neutronenquelle aktiviert und dann in die Zählvorrichtung eingesetzt. Die Messung wird dann über einen Zeitraum von 400 Sekunden, wieder bei einer Torzeit von 10 Sekunden, durchgeführt. Insgesamt werden 5 Messgänge mit aktiviertem Silber durchgeführt.

Messen der Halbwertszeit von aktiviertem Indium. Die Indiumpräparate wurden bereits im Vorhinein aktiviert. Es wird hierfür ein einzelner Messgang über 50 Minuten bei einer Torzeit von 120 Sekunden durchgeführt.

Messprotokoll 252

Marius Pfeiffer

19.11.2024

Robert Grosch

Einstellungen für Messungen

Spannung Zählrohr: 550V

Untergrundmessung:

Torzeit: 10s

Anzahl d. Messungen: 48

Silbermessungen

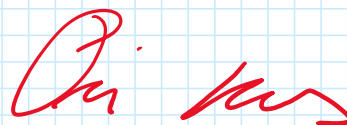
Torzeit: 10s

Anzahl d. Messungen: 40

Indiummessung

Torzeit: 120s

Anzahl d. Messungen: 25



3 Auswertung

Wir betrachten zunächst das Resultat der Untergrundmessung, dieses ist in Abbildung (2) zu sehen.

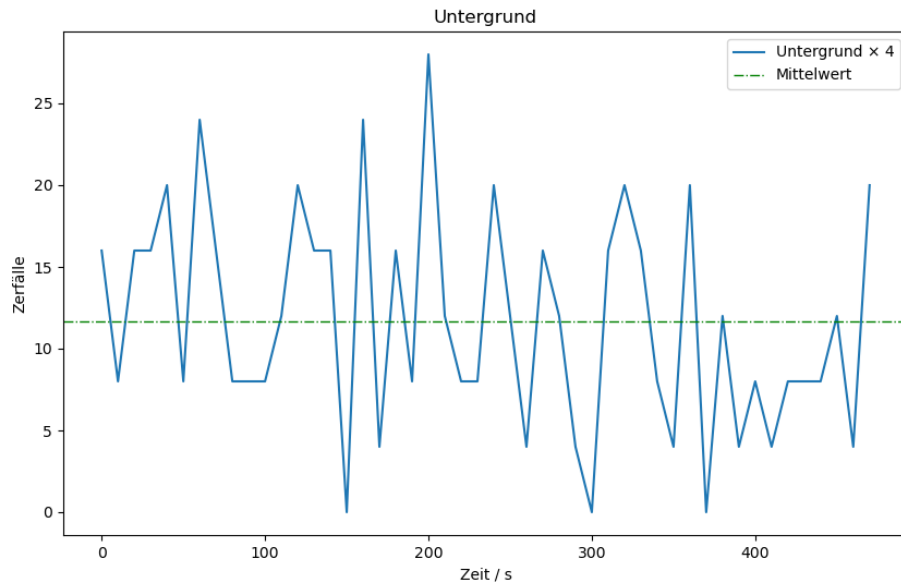


Abbildung 2: Vervierfachter Untergrund mit Mittelwert

Den Mittelwert des Untergrundes, im Plot in grün dargestellt, speichern wir uns ab für die Fits im weiteren Verlauf.

3.1 Bestimmung der Halbwertszeit von aktiviertem Silber

Zur Auswertung betrachten wir die Summe der vier Messgänge des Silberzerfalls. Da sich das aktivierte Silber aus zwei Isomeren mit unterschiedlichen Zerfallsverhalten zusammensetzt, fitten wir die Summe zweier Funktionen an die Daten. Die Fitfunktion für die Aktivität ist daher wie folgt definiert:

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

Hierbei sind A_1 und A_2 die anfänglichen Anzahlen verfügbarer radioaktiver Kerne, λ_1 und λ_2 die Zerfallskonstanten und y_0 der Offset, bestimmt durch den Mittelwert des Untergrundrauschens. Alle Variablen mit Index 1 beziehen sich auf ^{110}Ag , alle mit Index 2 auf ^{108}Ag .

Abbildung (3) zeigt die Daten des Silberzerfalls, gemeinsam mit der angepassten Funktion aus Gleichung 6. Die optimierten Parameter sind Tabelle (1) zu entnehmen.

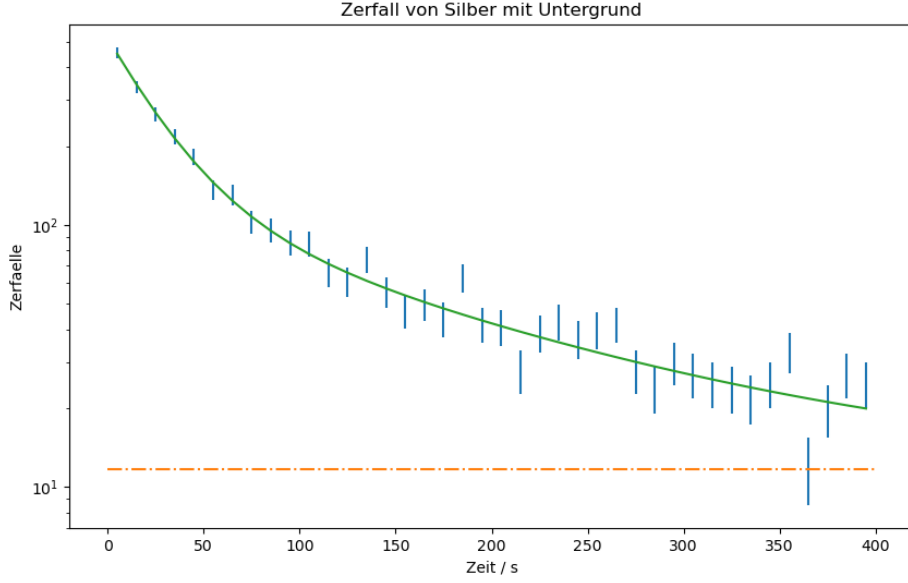


Abbildung 3: Fit der Aktivität von aktiviertem Silber

Parameter	Optimierter Wert
A_1	394 ± 26
λ_1	$(0.036 \pm 0.005) \frac{1}{s}$
A_2	114 ± 20
λ_2	$(0.0066 \pm 0.0008) \frac{1}{s}$

Tabelle 1: Optimierte Parameter für die Aktivität des aktivierten Silbers

Zur Beurteilung der Güte des Fits berechnen wir die üblichen Werte χ^2 , χ_{red}^2 , sowie die Fitwahrscheinlichkeit \mathbb{P}_{Fit} .

$$\chi^2 = 37.56, \quad \chi_{\text{red}}^2 = 1.04, \quad \mathbb{P}_{\text{Fit}} = 40.0\% \quad (7)$$

Diese werden im abschließenden Teil der Ausarbeitung beurteilt werden.

Wie in Gleichung 6 zu erkennen, geht der Mittelwert des Untergrunds als die Variable y_0 zwar in diesen Fit mit ein, allerdings nicht der Fehler des Untergrunds. Um auch diesen Fehler mit Berücksichtigen zu können, führen wir den eben dargestellten Fit noch zwei weitere Male durch. Einmal mit y_0 als Summe vom Mittelwert des Untergrundes und dessen Fehler und einmal als Differenz der Werte. Daraus erhalten wir für die Zerfallskonstanten weitere optimierte Werte λ_1^+, λ_2^+ für den addierten Fehler und λ_1^-, λ_2^- für den subtrahierten Fehler.

Nun Berechnen wir die Abstände zwischen diesen neuen Werten und den zuvor optimierten λ_1, λ_2 in der Form

$$\Delta_{bkg}^+(\lambda_i) = |\lambda_i - \lambda_i^+|, \quad (8)$$

$$\Delta_{bkg}^-(\lambda_i) = |\lambda_i - \lambda_i^-| \quad (9)$$

und erhalten den gesamten Fehler durch den Untergrund als den Mittelwert der Abstände

$$\Delta_{bkg}(\lambda_i) = \frac{\Delta_{bkg}^+(\lambda_i) + \Delta_{bkg}^-(\lambda_i)}{2}. \quad (10)$$

Diese Fehlerwerte summieren wir nun quadratisch mit den unsprünglich durch den Fit für λ_i bestimmten Fehlern nach

$$\Delta\lambda_i = \sqrt{\Delta_{fit}(\lambda_i)^2 + \Delta_{bkg}(\lambda_i)^2} \quad (11)$$

auf, um den gesamten Fehler der Zerfallskonstanten zu erhalten. Somit können wir nun die Zerfallskonstanten wie folgt angeben:

$$\lambda_1 = (0.036 \pm 0.005) \frac{1}{s}, \quad (12)$$

$$\lambda_2 = (0.0066 \pm 0.0009) \frac{1}{s}. \quad (13)$$

Aus diesen Werten berechnen wir die Halbwertszeiten nach

$$T_{1/2,i} = \frac{\ln 2}{\lambda_i} \pm \frac{\ln 2}{\lambda_i^2} \Delta\lambda_i \quad (14)$$

und erhalten

$$T_{1/2,1} = (19.3 \pm 1.7) s, \quad (15)$$

als Halbwertszeit von ^{110}Ag und

$$T_{1/2,2} = (105 \pm 9) s. \quad (16)$$

als Halbwertszeit von ^{108}Ag .

Im abschließenden Diskussionsteil werden diese Werte mit den Literaturwerten verglichen werden. Zunächst fahren wir jedoch damit fort, die Halbwertszeit des Indiumisotops zu berechnen.

3.2 Bestimmung der Halbwertszeit von aktiviertem Indium

Für den vorherigen Versuchsteil nahmen wir die Anzahl an Zerfällen in Intervallen (Torzeit) von zehn Sekunden auf. Entsprechend führten wir auch die Untergrundmessung über zehn Sekunden durch. Für die Messungen am aktivierten Indium wurde die Torzeit auf 120 Sekunden erhöht. Entsprechend multiplizieren wir den Mittelwert und den Fehler des Untergrunds mit 12, um entsprechende Werte für das größere Intervall zu erhalten.

Ab diesem Punkt gehen wir größtenteils analog zur Auswertung des Silberzerfalls vor. Als Modell für die Aktivität verwenden wir eine einfache Exponentialfunktion der Form

$$A(t) = A_0 e^{-\lambda t}. \quad (17)$$

Das Resultat des Fits ist in Abbildung (4) zu sehen, die optimierten Parameter sind erneut in der Tabelle (2) darunter.

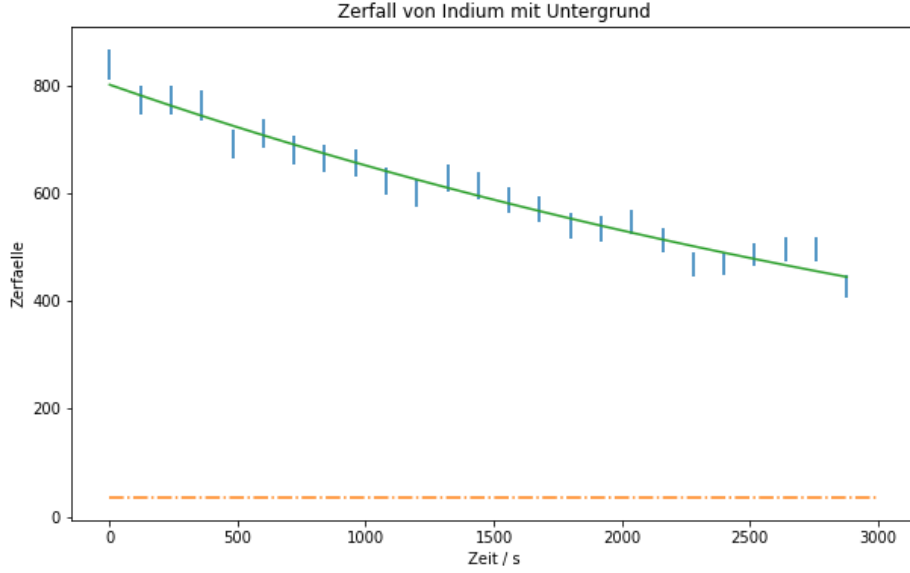


Abbildung 4: Fit der Aktivität von aktiviertem Indium

Parameter	Optimierter Wert
A	766 ± 11
λ	$(0.217 \pm 0.009) \cdot 10^{-3} \frac{1}{s}$

Tabelle 2: Optimierte Parameter für die Aktivität des aktivierten Indiums

Zur späteren Beurteilung der Güte des Fits in der Diskussion werden wir die Werte

$$\chi^2 = 17.7, \quad \chi^2_{\text{red}} = 0.84, \quad \mathbb{P}_{\text{Fit}} = 67.0\% \quad (18)$$

betrachten.

Auch für diesen Fit müssen wir wieder die Auswirkungen des Fehlers des Untergrundes auf den Fehler der Zerfallskonstanten separat berechnen. Hierzu gehen wir analog zum vorherigen Versuchsteil vor und erhalten damit einen Wert für die Zerfallskonstante von

$$\lambda = (0.217 \pm 0.009) \cdot 10^{-3} \frac{1}{s}. \quad (19)$$

Der Fehler durch den Untergrund ist hier mit $\Delta_{\text{bkg}}(\lambda) = 1.2 \cdot 10^{-6}$ so gering, dass er nur außerhalb der signifikanten Stellen des Fehlers eine Auswirkung hat und somit, hier nicht ersichtlich ist.

Mit der soeben bestimmten Zerfallskonstante können wir nun abschließend die Halbwertszeit von Indium-116 zu

$$T_{1/2} = 53.1 \pm 2.2 \text{min} \quad (20)$$

bestimmen.

4 Zusammenfassung und Diskussion

Durch die Bestrahlung stabiler Isotope mit langsamen (thermischen) Neutronen, welche von den Kernen der Isotope eingefangen werden, ist es möglich, radioaktiver Quellen herzustellen. Dieser Prozess, die sogenannte Aktivierung, wurde in Versuch 252 untersucht. Hierzu betrachteten wir sowohl Silber- als auch Indiumisotope. Aus den stabilen Silberisotopen ^{107}Ag und ^{109}Ag werden durch den einfaches von Neutronen die radioaktiven Isotope ^{108}Ag und ^{110}Ag . Aus dem stabilen Indiumisotop ^{115}In werden durch die Aktivierung die radioaktiven Isomere ^{116}In und ^{116m}In erzeugt.

Für den Versuch nahmen wir die Zahl der Zerfälle für das Silber in Intervallen von 10 Sekunden über 40 Messungen und für das Indium in Intervallen von 120 Sekunden über 25 Messungen mit einem Geiger-Müller-Zählrohr auf. Die Zahl der Zerfälle pro Sekunde, genannt Aktivität, folgt, nach der Aktivierung der Isotope, dem negativ exponentiellen Abfall des Zerfallsgesetzes. Ein solches Modell passten wir an die aufgezeichneten Daten an, um primär die Zerfallskonstante λ zu bestimmen.

Betrachten wir zunächst die Güte des Fits an die Aktivität des Silbers. Der χ^2 Wert von 37.56 ist etwas hoch, unter Berücksichtigung der Freiheitsgrade mit einem reduzierten χ^2 von etwa 1.04 liegen wir allerdings im optimalen Bereich. Die Fitwahrscheinlichkeit von 40% ist mit einem Wert um 50% sehr gut und deutet darauf hin, dass das Modell die Messdaten gut beschreibt.

Für den Fit an die Aktivität des Indiums erhalten wir einen etwas niedrigeren χ^2 Wert von etwa 17.7. Mit 0.84 ist der reduzierte χ^2 Wert zwar etwas weiter unter dem optimalen Wert von 1. Dies kann bedeuten, dass das Modell Unsicherheiten in den Messdaten etwas überschätzt. Die sehr gute Fitwahrscheinlichkeit von 67.0%, zeigt aber, ebenso wie die anderen beiden Werte, dass das Modell sehr gut zu den Daten passt.

Aus den durch die Fits ermittelten Zerfallskonstanten berechneten wir in der Auswertung die Halbwertszeiten der Isotope zu

Isotop	Halbwertszeit
^{110}Ag	$(19.3 \pm 1.7) \text{ s}$
^{108}Ag	$(105 \pm 9) \text{ s}$
^{116}In	$(53.1 \pm 2.2) \text{ min}$

Dies Werte möchten wir nun mit den Literaturwerten aus der Nuklidkarte Abbildung (5) vergleichen. Für die Halbwertszeit von ^{108}Ag sind hier 2.41min angegeben, dies entspricht einer Abweichung von 4.4σ vom von uns berechneten Wert. Die angegebene Halbwertszeit von 24.6s für ^{110}Ag weicht um 3.2σ von dem von uns berechneten Wert ab.

Mit einer Abweichung von gerade einmal 0.4σ ist unser berechneter Wert der Halbwertszeit des Indiums sehr nah am Literaturwert von 54min aus der Nuklidkarte.

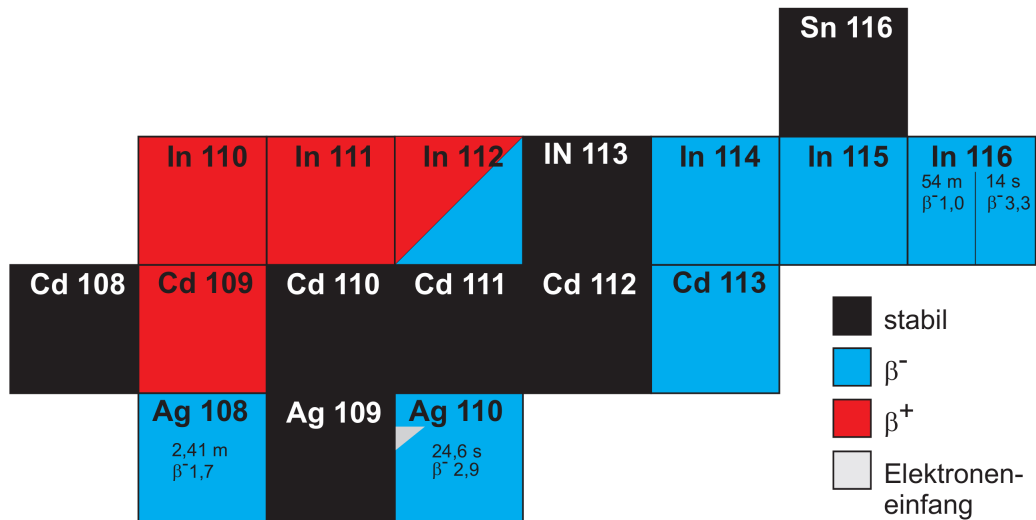


Abbildung 5: Ausschnitt der Nuklidkarte

Ein Großteil der Abweichungen wird hierbei vermutlich aus den Fits entspringen. Hierzu zählt zum einen die Abschätzung des Untergrundes. Bereits Abbildung (2) zeigt, dass die Untergrundzählrate stark rauscht. In die Fits beziehen wir allerdings nur dessen Mittelwert als Konstante mit ein. Zwar betrachten wir noch gesondert die Auswirkungen der Fehler des Mittelwertes, trotzdem werden feinere Auswirkungen größtenteils unterschätzt.

Im Fit der Aktivität des aktivierten Silbers könnte die gleichzeitige Betrachtung zweier zerfallenden Isotope Auswirkungen auf die Genauigkeit haben. Zwar lassen sich diese im Fitmodell mathematisch getrennt darstellen, in Realität kann es aber sein, dass hier dennoch Interaktionen und Überlagerungen stattfinden, welche vom Fitmodell nicht berücksichtigt werden. Betrachten wir die Daten des Indiumzerfalls, so fällt auf, dass der allererste Messpunkt deutlich über der angepassten Funktion liegt.

Wie in der Einleitung erklärt, entstehen bei der Aktivierung des Indium-115 Isotops die Isomere ^{116}In und ^{116m}In . Hierbei hat ^{116m}In die Halbwertszeit von etwa 54m, während sie bei ^{116}In lediglich etwa 14s beträgt. Was wir als Abweichung im ersten Messpunkt sehen ist also der Zerfall des ^{116}In , welcher sich kurz mit dem des ^{116m}In Isotop überlagert. Trotz dieser kleinen Unreinheit in den Daten sehen wir bei der Bestimmung der Halbwertszeit von ^{116m}In auf ein sehr gutes Ergebnis.

Python Code, Hauptprogramm

Versuch_252_Mabert

December 12, 2024

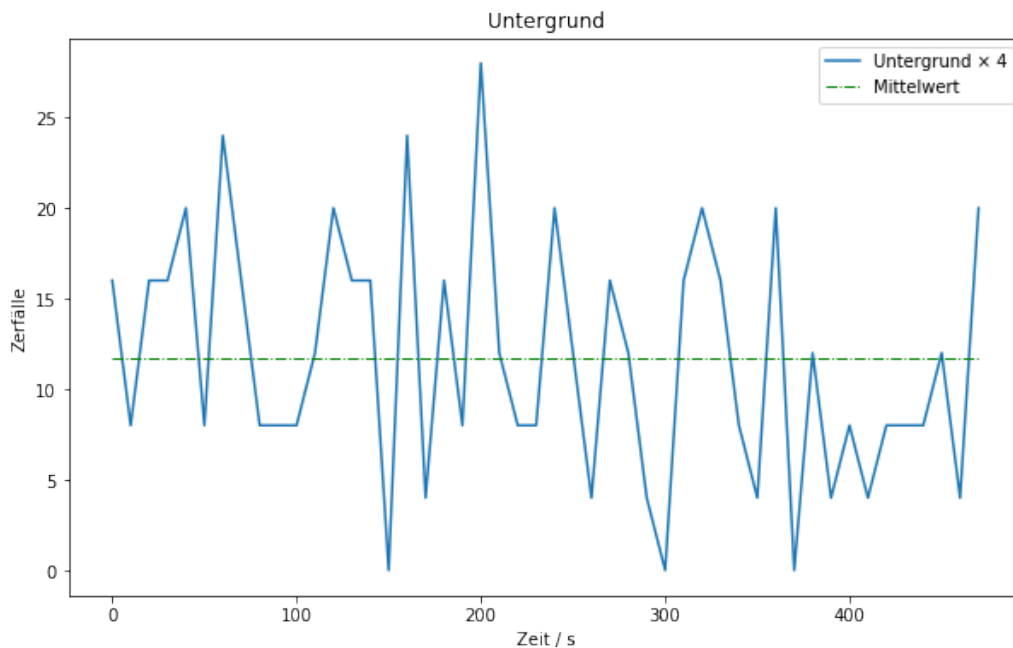
```
[4]: %matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit
from scipy.stats import chi2

[7]: unterg_data=np.loadtxt('untergrund.txt', usecols=(0,1), delimiter=',',
    ↪ skiprows=4)
unterg_t = unterg_data[:,0]
unterg = unterg_data[:,1]

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

plt.figure(figsize=(10,6))
plt.plot(unterg_t, 4*unterg, label='Untergrund × 4')
plt.plot((unterg_t[0], unterg_t[-1]), (mittelw_unterg, mittelw_unterg),
    ↪ linewidth=1, linestyle='-.', color='g', label='Mittelwert')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfälle')
plt.title('Untergrund')
plt.legend()
plt.savefig('untergrund.png',format='png')
#plt.yscale('log')
```

Mittelwert: 11.666666666666666 Fehler: 0.9706336227586749



```
[8]: #Silber
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)

[9]: # Fit with background = mean

y0=mittelw_unterg #Untergrund

def fit_func(x, A1,l1,A2,l2):
    return A1*np.exp(-x*l1) + A2*np.exp(-x*l2) + y0

popt, pcov=curve_fit(fit_func,t,N, p0=[500,0.02,50,0.001], sigma=Fehler_N)

plt.figure(figsize=(10,6))
plt.errorbar(t,N, Fehler_N, linestyle='None')
plt.plot([0,400], [y0, y0], linestyle='-.')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Silber mit Untergrund')
```

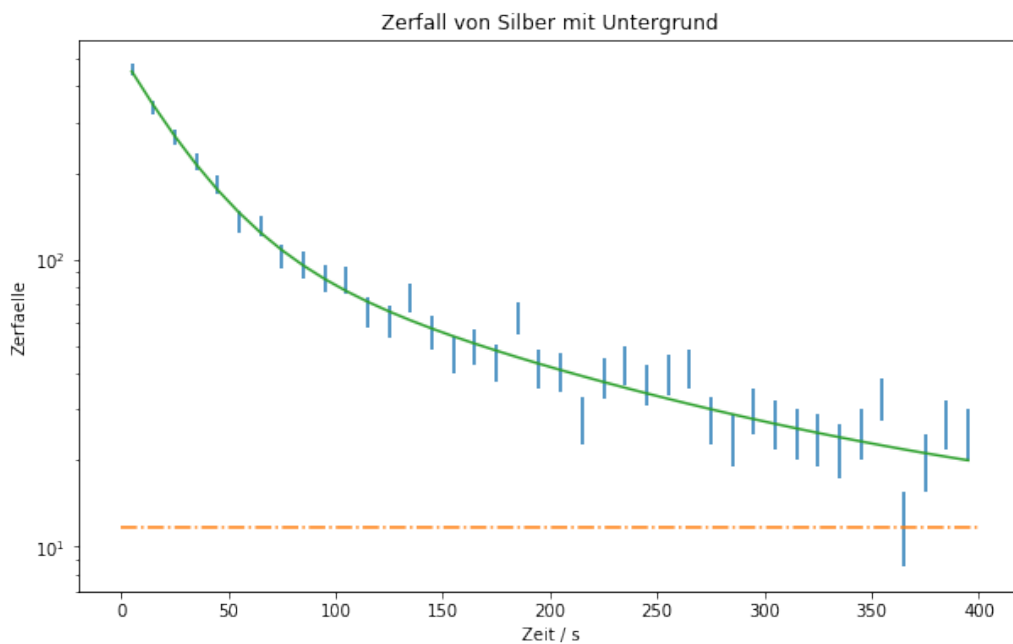
```

plt.yscale('log')
plt.plot(t,fit_func(t,*popt))
plt.savefig('silber.png',format='png')
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]))

l1 = popt[1]
l1_err = np.sqrt(pcov[1][1])
l2 = popt[3]
l2_err = np.sqrt(pcov[3][3])

```

A1= 394.04572684134564 , Standardfehler= 25.75251485363819
l1= 0.0358759181701219 , Standardfehler= 0.004170154725547418
A2= 113.935801849386 , Standardfehler= 19.436365369928318
l2= 0.006632896230054428 , Standardfehler= 0.0007463782296500707



```

[10]: 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)

```

```

prob=round(1-chi2.cdf(chi2_,dof),2)*100
print("Wahrscheinlichkeit=", prob,"%")

```

```

chi2= 37.55902108627056
chi2_red= 1.0433061412852933
Wahrscheinlichkeit= 40.0 %

```

```

[11]: # Fit with background = mean - 1sigma

y0 = mittelw_unterg - fehler_unterg #Untergrund

def fit_func(x, A1,l1,A2,l2):
    return A1*np.exp(-x*l1) + A2*np.exp(-x*l2) + y0

popt_subsig, pcov_subsig = curve_fit(fit_func,t,N, p0=[500,0.02,50,0.001,],  

    ↪sigma=Fehler_N)

l1_sub = popt_subsig[1]
l1_sub_err = np.sqrt(pcov_subsig[1][1])
l2_sub = popt_subsig[3]
l2_sub_err = np.sqrt(pcov_subsig[3][3])

```

```

[12]: # Fit with background = mean - 1sigma

y0 = mittelw_unterg + fehler_unterg #Untergrund

def fit_func(x, A1,l1,A2,l2):
    return A1*np.exp(-x*l1) + A2*np.exp(-x*l2) + y0

popt_addsig, pcov_addsig = curve_fit(fit_func,t,N, p0=[500,0.02,50,0.001,],  

    ↪sigma=Fehler_N)

l1_add = popt_addsig[1]
l1_add_err = np.sqrt(pcov_addsig[1][1])
l2_add = popt_addsig[3]
l2_add_err = np.sqrt(pcov_addsig[3][3])

```

```

[13]: l1_sub_diff = np.abs(l1 - l1_sub)
l1_add_diff = np.abs(l1 - l1_add)
l2_sub_diff = np.abs(l2 - l2_sub)
l2_add_diff = np.abs(l2 - l2_add)

print(f"|l1 - l1^-| = {l1_sub_diff}")
print(f"|l2 - l2^-| = {l2_sub_diff}")
print(f"|l1 - l1^+| = {l1_add_diff}")
print(f"|l2 - l2^+| = {l2_add_diff}")

```

```

l1_err_mean = np.mean([l1_sub_diff, l1_add_diff])
l2_err_mean = np.mean([l2_sub_diff, l2_add_diff])

print(f"l1 bkg err mean: {l1_err_mean}")
print(f"l2 bkg err mean: {l2_err_mean}")

l1_err_total = np.sqrt(pcov[1][1] + l1_err_mean**2)
l2_err_total = np.sqrt(pcov[3][3] + l2_err_mean**2)

print("A1=",popt[0], ", Standardfehler=", np.sqrt(pcov[0][0]))
print("l1=",popt[1], ", Standardfehler=", l1_err_total)
print("A2=",popt[2], ", Standardfehler=", np.sqrt(pcov[2][2]))
print("l2=",popt[3], ", Standardfehler=", l2_err_total)

```

```

|l1 - l1^-| = 0.00042124146933473355
|l2 - l2^-| = 0.0002915609365967088
|l1 - l1^+| = 0.00046744794798004446
|l2 - l2^+| = 0.0003150092715650272
l1 bkg err mean: 0.000444344708657389
l2 bkg err mean: 0.000303285104080868
A1= 394.04572684134564 , Standardfehler= 25.75251485363819
l1= 0.0358759181701219 , Standardfehler= 0.004193761158568438
A2= 113.935801849386 , Standardfehler= 19.436365369928318
l2= 0.006632896230054428 , Standardfehler= 0.0008056440380545968

```

[16]: *# Halbwertszeit berechnen*

```

halbwertszeit_1 = np.log(2) / l1
halbwertszeit_2 = np.log(2) / l2

halbwertszeit_1_err = (np.log(2) / (lebensdauer_1**2)) * lebensdauer_1_err
halbwertszeit_2_err = (np.log(2) / (lebensdauer_2**2)) * lebensdauer_2_err

print(halbwertszeit_1, halbwertszeit_1_err, halbwertszeit_2,
      ↪ halbwertszeit_2_err)

```

```

19.320681279098537 1.5654837562077193 104.50143595179651 8.798076093221706

```

0.0.1 Indiumzerfall

[43]: *#Indium*

```

indium_data = np.loadtxt('indium.txt', usecols=[1], delimiter=',', skiprows=4)

mittelw_120_unterg=np.mean(unterg) * 12
fehler_120_unterg=np.std(unterg)/np.sqrt(len(unterg)) * 12
print('Mittelwert:', mittelw_120_unterg, 'Fehler:',fehler_120_unterg)

```

```

t_120 = t=np.arange(0, 3000, 120)

indium_error = np.sqrt(indium_data)

plt.figure(figsize=(10,6))
plt.errorbar(t_120, indium_data, indium_error, linestyle='None')
plt.plot([0,3000], [y0, y0], linestyle='-.')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Indium mit Untergrund')

#plt.yscale('log')

y0 = mittelw_120_unterg

def fit_func(x, A1,l1):
    return A1*np.exp(-x*l1) + y0

popt_ind, pcov_ind=curve_fit(fit_func,t_120,indium_data, p0=[50,0.001],
    sigma=indium_error)

plt.plot(t_120,fit_func(t_120,*popt_ind))

plt.savefig('indium.png',format='png')

l1_ind = popt_ind[1]

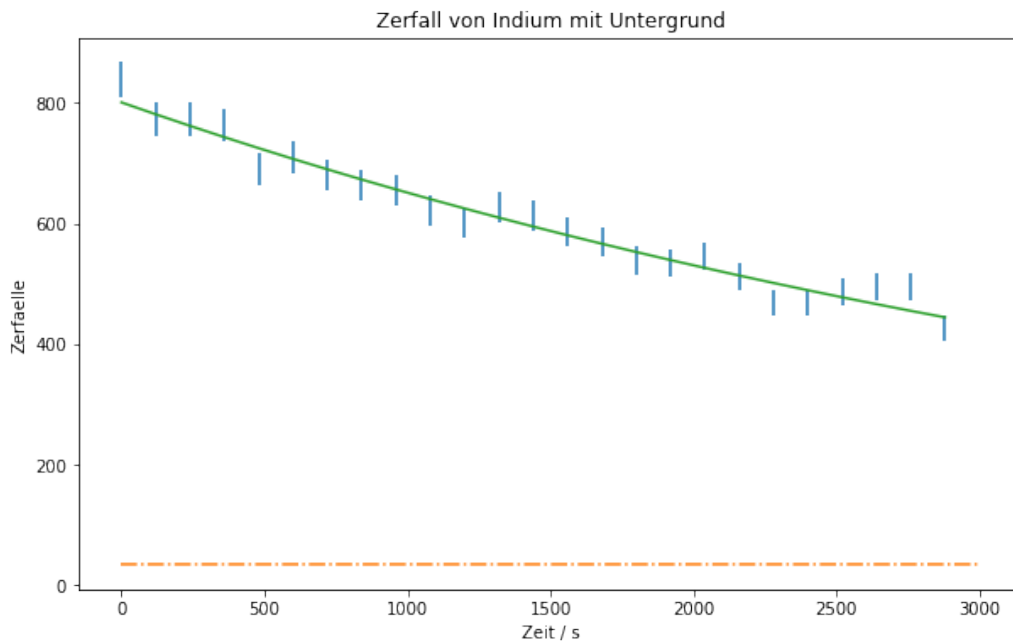
print("A=",popt_ind[0], ", Standardfehler=", np.sqrt(pcov_ind[0][0]))
print("l=",popt_ind[1], ", Standardfehler=", np.sqrt(pcov_ind[1][1]))

```

Mittelwert: 35.0 Fehler: 2.9119008682760246

A= 765.6323496164731 , Standardfehler= 10.412435073562266

l= 0.00021744099371320672 , Standardfehler= 8.899017215255906e-06



```
[18]: chi2=np.sum((fit_func(t_120,*popt_ind)-indium_data)**2/indium_error**2)
      dof=len(indium_data)-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= 17.722120844041417
chi2_red= 0.8439105163829246
Wahrscheinlichkeit= 67.0 %
```

```
[19]: # Fit with background = mean - 1sigma

      y0 = mittelw_120_unterg - fehler_120_unterg #Untergrund

      def fit_func(x, A1,l1):
          return A1*np.exp(-x*l1) + y0

      popt_ind_subsig, pcov_ind_subsig = curve_fit(fit_func, t_120, indium_data,
      ↪p0=[50,0.001], sigma=np.sqrt(indium_data))

      l1_ind_sub = popt_ind_subsig[1]
```

```
l1_ind_sub_err = np.sqrt(pcov_ind_subsig[1][1])
```

```
[20]: # Fit with background = mean + 1sigma

y0 = mittelw_120_unterg + fehler_120_unterg #Untergrund

def fit_func(x, A1,l1):
    return A1*np.exp(-x*l1) + y0

popt_ind_addsig, pcov_ind_addsig = curve_fit(fit_func, t_120, indium_data,
    p0=[50,0.001], sigma=np.sqrt(indium_data))

l1_ind_add = popt_ind_addsig[1]
l1_ind_add_err = np.sqrt(pcov_ind_addsig[1][1])
```

```
[44]: l1_ind_sub_diff = np.abs(l1_ind - l1_ind_sub)
l1_ind_add_diff = np.abs(l1_ind - l1_ind_add)

print(f"|l1 - l1^-| = {l1_ind_sub_diff}")
print(f"|l1 - l1^+| = {l1_ind_add_diff}")

l1_ind_err_mean = np.mean([l1_ind_sub_diff, l1_ind_add_diff])

print(f"l1 bkg err mean: {l1_ind_err_mean}")

l1_ind_err_total = np.sqrt(pcov_ind[1][1] + l1_ind_err_mean**2)

print("A1=",popt_ind[0], ", Standardfehler=", np.sqrt(pcov_ind[0][0]))
print("l1=",popt_ind[1], ", Standardfehler=", l1_ind_err_total)
```

```
|l1 - l1^-| = 1.1299513612909298e-06
|l1 - l1^+| = 1.1417382263016165e-06
l1 bkg err mean: 1.1358447937962732e-06
A1= 765.6323496164731 , Standardfehler= 10.412435073562266
l1= 0.00021744099371320672 , Standardfehler= 8.97121233685922e-06
```

```
[22]: halbwertszeit_ind = np.log(2) / l1_ind

halbwertszeit_ind_err = (np.log(2) / (l1_ind**2)) * l1_ind_err_total

print(halbwertszeit_ind / 60, halbwertszeit_ind_err/60)
```

```
53.12914005796672 2.1920098349228767
```

```
[ ]:
```