

Versuch 252 - Radioaktiver Zerfall

Marius Mangold

January 5, 2022

Contents

1	Grundlagen	2
2	Durchführung	3
2.1	Untergrundmessung	3
2.2	Halbwertszeit Silber	3
2.3	Halbwertszeit Indium	3
3	Messprotokoll	3
3.1	Aufgaben	3
3.2	Messaufbau	3
3.3	Untergrund	4
3.4	Halbwertszeit Silber	5
3.5	Halbwertszeit Indium	6
4	Auswertung	7
4.1	Untergrundmessung	7
4.2	Halbwertszeit Silber	7
4.3	Halbwertszeit Indium	9
5	Zusammenfassung und Diskussion	11
5.1	Untergrundmessung	11
5.2	Halbwertszeit Silber	11
5.3	Halbwertszeit Indium	11
5.4	Abschluss	12
6	Anhang	12
6.1	Abbildungsverzeichniss	12
6.2	Jupyter Notebook	13

1 Grundlagen

In diesem Versuch möchten wir den Zerfall von Silber-, und Indium-isotopen betrachten. Hierfür aktivieren wir stabile Isotope durch Kernreaktionen. Verwendet werden die elektrisch neutralen, thermischen Neutronen, welche von einem Atom gut eingefangen werden können, da sie keine Abstoßung erfahren. Ausgesendet werden diese von einer Neutronenquelle, welche Berylliumspäne und einen Alphastrahler enthält.



Durch diese Kernreaktion erhalten wir Neutronen mit Energien von 1 bis 10 Megaelektronenvolt. Diese sind also noch weit über thermischen Geschwindigkeiten, und werden daher durch den umgebenden Paraffinblock durch Stöße mit dem Wasserstoffkern abgebremst, bis sie im thermischen Bereich liegen. Durch die Bestrahlung mit diesen Neutronen wird aus dem stabilen Isotop ^{115}In ^{116}In , welches Betastrahlung aussendet.

Durch die Aktivierung werden also radioaktive Kerne erzeugt. Die Rate der Erzeugung ist jedoch nach dem Zerfallsge setz abhängig von den vorhandenen radioaktiven Kernen. Die Aktivität nimmt daher wie folgt zu:

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

Wir sehen, dass der Verlauf asymptotisch einen Gleichgewichtszustand annehmen muss, bei welchem ebenso viele radioaktive Kerne erzeugt werden, wie zerfallen. Wird die Aktivierung beendet finden wir einen klassischen radioaktiven Zerfall und dessen Halbwertszeit nach:

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

mit der Halbwertszeit

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

Natürliches Silber besteht zu 51% aus ^{107}Ag und zu 49% aus ^{109}Ag . Daher werden bei der Aktivierung zwei unterschiedliche Isotope erzeugt : ^{108}Ag und ^{110}Ag . Sie zerfallen weiterhin durch Betazerfall in ^{108}CD und ^{110}CD . Die Halbwertszeit dieser erzeugten Isotope ist jedoch etwa um Faktor sechs verschieden, wodurch durch unterschiedliche Aktivierungszeiten auf das Isotopenverhältnis Einfluss genommen werden kann. Bei kurzer Aktivierung (20 Sekunden) finden wir hauptsächlich ^{110}Ag vor. Mit längerer Aktivierungszeit nähert sich ^{110}Ag bereits der Sättigung an, wodurch sich das Verhältnis aufteilt. Siehe hierzu auch [Abbildung 1](#), welche zur besseren Seitenaufteilung angehängt wurde.

2 Durchführung

2.1 Untergrundmessung

Wir werden damit beginnen, die Untergrundmessung durchzuführen, um bei den späteren Messungen äußere Einflüsse zu minimieren, bzw. herauszurechnen. Hierfür stellen wir eine Zählrohrspannung zwischen 500 und 550 Volt ein, und messen über 400 Sekunden bei einer Torzeit von 10 Sekunden die Untergrundstrahlung. Hierbei ist darauf zu achten, alle Strahlungsquellen aus dem Raum zu entfernen bzw. abzudecken.

2.2 Halbwertszeit Silber

Wir legen das Silberblech in die Neutronenquelle und lassen es für mindestens sieben Minuten aktivieren. Währenddessen stellen wir die Torzeit im Computerprogramm auf 10 Sekunden ein. Anschließend werden wir so schnell wie möglich das Silberblech in den Probenhalter führen, und die Messung im Programmfenster starten. Diese Messungen werden wir viermal wiederholen, wobei jedes Mal das Silberblech zu aktivieren ist. Wir werden die Messdaten speichern und uns elektronisch zusenden.

2.3 Halbwertszeit Indium

Für die Indium-Messung gehen wir symmetrisch zum ersten Versuchsteil vor. Wir achten erneut darauf, dass sich keine aktiven Quellen im Raum befinden oder diese abgeschirmt sind und führen eine Nullmessung über fünf Minuten durch. Anschließend führen wir die Indio Messungen durch, wobei ein Messintervall von 120 Sekunden gewählt wird, und die Messung über einen Zeitraum von 50 Minuten aufgenommen wird. Auch diese Messung speichern wir erneut ab und senden sie uns elektronisch zu.

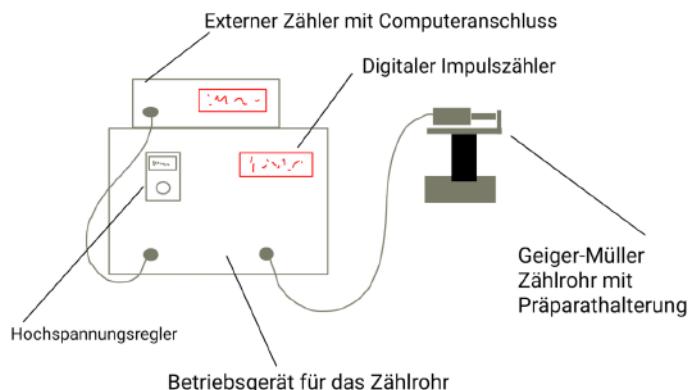
3 Messprotokoll

3.1 Aufgaben

1. Bestimmung der Halbwertszeit von ^{116}In .
2. Bestimmung der Halbwertszeiten von ^{108}Ag und ^{110}Ag

3.2 Messaufbau

- Geiger-Müller Zählrohr
- mit Betriebsgerät
- Externer Impulszähler
- PC mit Drucker
- Neutronenquelle
- Präparathalterung
- Indium- und Silberbleche



Skizze 1: Versuchsaufbau und Materialliste

3.3 Untergrund

Wir versichern uns, dass alle radioaktiven Quellen aus dem Raum entfernt oder abgeschirmt sind, und stellen anschließend am Betriebsgerät eine Spannung von ca. 500 Volt ein. Am Computer stellen wir im Messprogramm eine Torzeit von 10 Sekunden ein. Nun vermessen wir den Untergrund über 8 Minuten.

Intervall	counts
1	2
2	4
3	5
4	4
5	1
6	4
7	5
8	4
9	5
10	3
11	2
12	3
13	8
14	1
15	3
16	4
17	2
18	5
19	2
20	5
21	3
22	3
23	6
24	2
25	2
26	4
27	6
28	2
29	2
30	4
31	5
32	7
33	6
34	3
35	1
36	3
37	2
38	1
39	2
40	2
41	4
42	2
43	5
44	5
45	4
46	3
47	3

Tabelle 1: Untergrundmessung

3.4 Halbwertszeit Silber

Wir aktivieren die Silberprobe mit dem ^{107}Ag und ^{109}Ag 7 Minuten lang in der Neutronenquelle. Das Computerprogramm stellen wir in der Zwischenzeit auf eine Messdauer von 400 Sekunden ein, und erhalten die Torzeit von 10 Sekunden. Nach Ablauf der Aktivierungszeit schieben wir das aktivierte Silberblech möglichst schnell in die vorgesehene Halterung, und starten die Messung sofort.

Dieses Vorgehen wiederholen wir viermal. Die Messdaten speichern wir jeweils ab.

Inter	counts
1	24
2	31
3	24
4	25
5	15
6	8
7	14
8	7
9	10
10	5
11	8
12	6
13	6
14	9
15	5
16	6
17	8
18	3
19	2
20	4
21	6
22	5
23	4
24	6
25	7
26	3
27	2
28	6
29	5
30	3
31	5
32	2
33	2
34	8
35	4
36	5
37	5
38	6
39	2
40	6

Inter	counts
1	88
2	70
3	71
4	64
5	45
6	48
7	45
8	30
9	25
10	27
11	11
12	19
13	11
14	15
15	12
16	16
17	11
18	12
19	8
20	9
21	7
22	9
23	9
24	15
25	11
26	10
27	10
28	8
29	9
30	7
31	8
32	5
33	8
34	3
35	3
36	7
37	5
38	7
39	5
40	4

Inter	counts
1	123
2	95
3	69
4	58
5	50
6	44
7	40
8	28
9	23
10	27
11	18
12	21
13	21
14	19
15	19
16	19
17	10
18	12
19	9
20	10
21	7
22	17
23	15
24	13
25	14
26	8
27	8
28	12
29	13
30	12
31	9
32	11
33	9
34	9
35	8
36	11
37	5
38	8
39	10
40	8

Intervall	counts
1	130
2	84
3	72
4	59
5	55
6	31
7	34
8	33
9	25
10	25
11	21
12	21
13	9
14	12
15	15
16	12
17	18
18	12
19	12
20	14
21	16
22	7
23	14
24	14
25	5
26	10
27	8
28	10
29	6
30	9
31	8
32	3
33	5
34	6
35	8
36	12
37	7
38	10
39	9
40	9

Tabelle 2: Ag 1

Tabelle 3: Ag 2

Tabelle 4: Ag 3

Tabelle 5: Ag 4

3.5 Halbwertszeit Indium

Wir erhöhen die Torzeit auf 120 Sekunden und setzen das bereits aktivierte Indium-präparat in die Halterung. Wir führen die Messung nun 50 Minuten durch, und speichern uns die Messdaten anschließend ab.

Intervall	counts
1	742
2	728
3	736
4	697
5	713
6	702
7	649
8	639
9	655
10	673
11	626
12	579
13	596
14	567
15	567
16	580
17	518
18	557
19	508
20	501
21	456
22	484
23	433
24	455
25	445

Tabelle 6: Indium

4 Auswertung

4.1 Untergrundmessung

Wir möchten damit beginnen den Mittelwert und dessen Fehler der Untergrund Rate zu bestimmen. Hierbei vervierfachen wir den Untergrund, da wir später vier Messreihen addieren. Wir erhalten:

$$\bar{n}_0 = (14.0 \pm 1.0) \frac{1}{s} \quad (5)$$

4.2 Halbwertszeit Silber

Nachdem wir nun den Untergrund bestimmt haben, können wir mit der Analyse der Silber Messung beginnen. Hierfür addieren wir die Messreihen und bestimmen die zugehörigen Fehler. Wir plotten die Zerfälle halblogarithmisch gegen die Zeit. Weiterhin fitten wir die Zerfallsfunktion an die Messdaten, und lassen sowohl die Fitparameter, als auch die Güte des Fits und die Fitwahrscheinlichkeit ausgeben, wobei A die Anzahl der Kerne zu Beginn der Messung, und l die Zerfallskonstante repräsentiert:

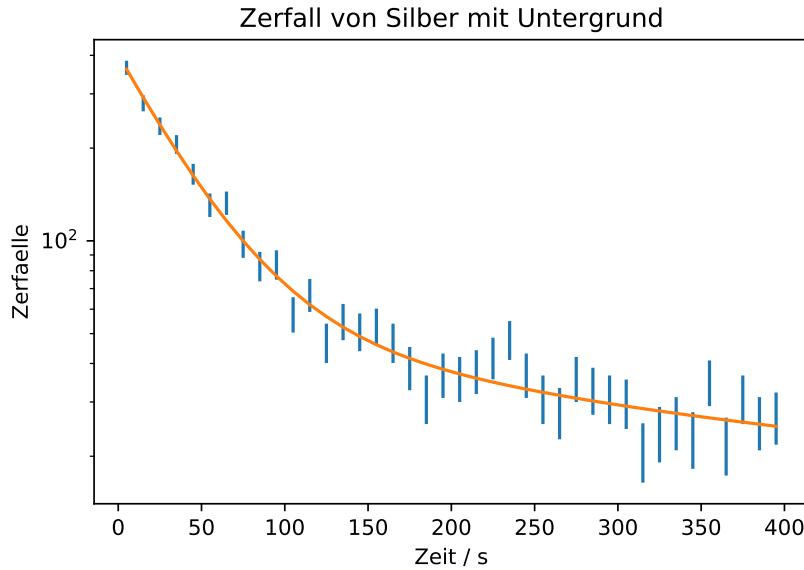


Diagramm 1: Die Fitfunktion des Zerfalls von Silber mit Untergrund

$$l_1 = (0.0247 \pm 0.0021) \frac{1}{s} \quad (6)$$

$$l_2 = (0.0033 \pm 0.0011) \frac{1}{s} \quad (7)$$

$$\chi^2 = 28 \quad (8)$$

$$\chi^2_{red} = 0.8 \quad (9)$$

$$P = 82.0\% \quad (10)$$

Die Güte des Fits ist (auch unter der Betrachtung des Graphen) ausreichend, eine genauere Analyse hierzu ist im Kapitel Diskussion zu finden.

Zu beachten ist, dass bei diesem Fit der Fehler des Untergrunds nicht berücksichtigt wurde. Daher möchten wir nun vom gemessenen Untergrund erst den 1σ Fehler des Untergrunds addieren und anschließend subtrahieren. Diese Werte wählen wir anschließend als Fitparameter $y0$, und

erhalten so jeweils die untere und obere Schranke der Zerfallskonstanten. Den Fehler dieser Zerfallskonstanten erhalten wir über die quadratische Addition des Mittelwerts der berechneten Differenzen zum Fehler des ersten Fits. Wir erhalten:

$$l_1^+ = 0.0248 \frac{1}{s} \quad (11)$$

$$l_2^+ = 0.0035 \frac{1}{s} \quad (12)$$

$$l_1^- = 0.0246 \frac{1}{s} \quad (13)$$

$$l_2^- = 0.0031 \frac{1}{s} \quad (14)$$

Hieraus finden wir weiterhin die Zerfallskonstantenfehler und letztendlich:

$$l_1 = (0.0247 \pm 0.0021) \frac{1}{s} \quad (15)$$

$$l_2 = (0.0033 \pm 0.0021) \frac{1}{s} \quad (16)$$

Mit der Formel (4) aus den Grundlagen:

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

finden wir den Fehler

$$\Delta T_{1/2} = \frac{\Delta \lambda \ln 2}{\lambda^2} \quad (18)$$

Nach diesen Gleichungen erhalten wir:

$$T_{1/2, AG^{110}} = (28 \pm 3)s \quad (19)$$

$$T_{1/2, AG^{108}} = (220 \pm 150)s \quad (20)$$

und wiederum aus dem Kehrbruch die Lebensdauer:

$$\tau = \frac{1}{\lambda} \quad (21)$$

$$\Delta \tau = \frac{\lambda}{\Delta \lambda} \quad (22)$$

$$\tau_{AG^{110}} = (0.035 \pm 0.003) \frac{1}{s} \quad (23)$$

$$\tau_{AG^{108}} = (0.005 \pm 0.003) \frac{1}{s} \quad (24)$$

Den Vergleich dieser errechneten Werte zu den Literaturwerten werden wir in der Diskussion vornehmen.

4.3 Halbwertszeit Indium

Das Vorgehen bei der Silber Messung wiederholen wir nun für den Zerfall von Indium. Hierbei müssen wir, da wir fälschlicherweise keine zweite Grundmessung durchgeführt haben, die vorhandene Untergrundmessung mit dem Faktor 12 multiplizieren und verwenden. Wir erhalten den Mittelwert und Fehler:

$$\overline{n'_o} = (42 \pm 3) \frac{1}{s} \quad (25)$$

Als Fitfunktion können wir nun eine Exponentialfunktion verwenden, wobei wir die ersten Werte aus der Messreihe nicht bei dem fit beachten, da wir die langanhaltende Strahlung beobachten möchten. In der nachfolgenden Grafik ist klar zu sehen, dass es kurzlebige Isotope geben muss, welche schneller zerfallen und anfangs zu einer zu großen Rate führen. Wir finden:

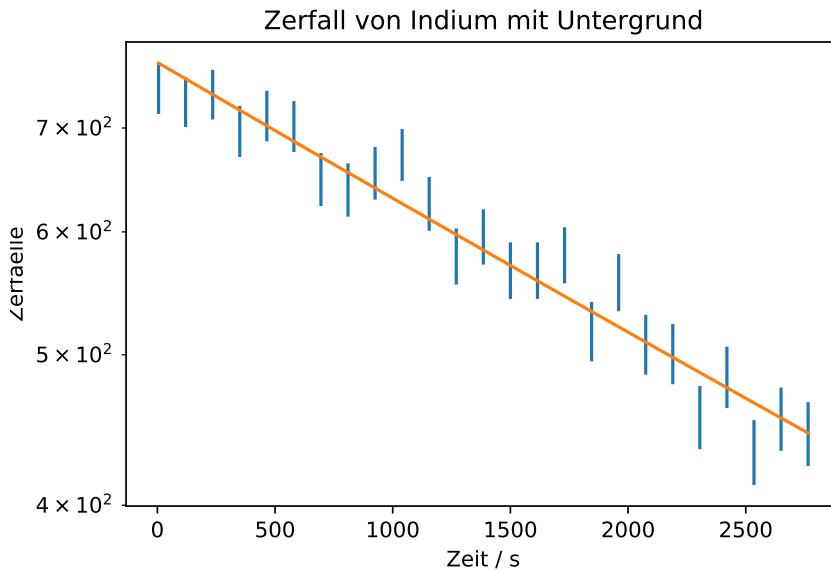


Diagramm 1: Die Fitfunktion des Zerfalls von Indium mit Untergrund

$$l = (2.16 \pm 0.10) * 10^4 \frac{1}{s} \quad (26)$$

$$\chi^2 = 19 \quad (27)$$

$$\chi^2_{red} = 0.8 \quad (28)$$

$$P = 71.0\% \quad (29)$$

Symmetrisch wird nun zum vorherigen Versuchsteil der Fehler des Untergrundes einmal subtrahiert und addiert. Wir erhalten:

$$l^+ = 0.000216 \frac{1}{s} \quad (30)$$

$$l^- = 0.000213 \frac{1}{s} \quad (31)$$

also:

$$l = (2.16 \pm 0.10) * 10^4 \frac{1}{s} \quad (32)$$

Was wie zuvor zu einer Halbwertszeit von:

$$T_{1/2,In} = (3230 \pm 160)s \quad (33)$$

führt, und eine Lebensdauer von:

$$\tau_{In} = (0.036 \pm 0.003)\frac{1}{s} \quad (34)$$

impliziert.

5 Zusammenfassung und Diskussion

5.1 Untergrundmessung

In diesem Versuch begannen wir damit, über die ausgelesene Messdatei Mittelwert und Fehler der Untergrundrate zu bestimmen.

5.2 Halbwertszeit Silber

Wir lasen die Messdaten der Silbermessung aus und fitteten die Zerfallsfunktion an die Messdaten. Für die Fitwahrscheinlichkeit erhielten wir überzeugende 82%, was uns in der Durchführung als auch in der Auswertung des Versuchs bestätigt. Um den Fehler der Zerfallskonstante zu ermitteln, bestimmten wir die Fitfunktion erneut, jedoch addierten wir diesmal den $1 - \sigma$ Fehler des Untergrundes hinzu. Dieses Vorgehen wiederholten wir mit der Subtraktion, um das Wertintervall und somit, nach quadratischer Addition, den Fehler zu erhalten. Aus diesen Werten fanden wir schlussendlich die Halbwertszeit, und somit die Lebensdauer der Isotope. Über die, der Praktikumsanleitung angehängten, Nuklidkarte konnten wir die Literaturwerte der Halbwertszeiten entnehmen, über diese können wir unsere Messwerte überprüfen, und die Abweichung bestimmen. Die Literaturwerte lauten:

$$T_{1/2, AG^{110}, Lit} = 24.6s \quad (35)$$

$$T_{1/2, AG^{108}, Lit} = 2.41min \quad (36)$$

was uns zu einer Sigmaabweichung von 1.2σ bei ^{110}AG und 0.5σ bei ^{108}AG führt.

Jedoch ist anzumerken, dass die geringe Sigmaabweichung bei 108 nicht aufgrund der großen Nähe der Werte auftritt, sondern vielmehr aufgrund des extrem großen Fehlers. Tatsächlich entspricht der gemessene Wert dem 1.5-fachen des Literaturwertes. Die große Abweichung vom Literaturwert, als auch der große Fehler ist eventuell auf die große Halbwertszeit relativ zum vermessenen Zeitraum zurückzuführen. Ebenso kann der Zerfall des 110 die Auswertung des 108 beeinflusst haben. All diese möglichen Fehlerquellen erlauben trotz allem keine schlüssige Erklärung dieses Ergebnisses, vor allem, da die 110 Messung ein so gutes Ergebnis lieferte. Ein möglicher Fehler in der Auswertung wurde mehrfach überprüft, ist aber selbstverständlich nicht auszuschließen.

5.3 Halbwertszeit Indium

Die Indiummessung wurde symmetrisch zur Silbervermessung durchgeführt. Hierbei fanden wir eine Fitwahrscheinlichkeit von 71.0% vor, welche etwas schlechter als die vorherige, aber trotzdem noch ausreichend gut ist.

Beim Betrachten der Nuklidkarte sehen wir, dass 116 in zwei Variationen zerfallen kann, einmal mit 54 Minuten Halbwertszeit, und einmal mit 14 Sekunden. Da die Messung grundsätzlich auf die Bestimmung der 54 Minuten-Variante ausgelegt ist, vernachlässigen wir bei der Auswertung den ersten Messwert, um die Verunreinigungen der Daten zu verhindern.

$$T_{1/2, Ag, Lit} = 54min \quad (37)$$

Wir sehen direkt, dass unsere bestimmte Halbwertszeit um weniger als 10 Zerfälle pro Sekunde abweicht, was uns zu einer extrem guten Sigmaabweichung von gerade einmal 0,05 Sigma führt. Somit ist die Theorie als auch unsere Versuchsdurchführung bestätigt. Dieses sehr gute Ergebnis lässt sich auch auf die sehr große Messzeit von 50 Minuten zurückführen, welche aufgrund der großen Halbwertszeit von 54 Minuten zwar knapp bemessen erscheint, jedoch wegen der Vielzahl an Zerfällen (rund 14800) wesentlich unanfälliger ist für äußere Einflüsse als bei der Silbervermessung, welche lediglich mit rund 2800 Messungen auskommen musste.

5.4 Abschluss

Abschließend bleibt zu sagen, dass der Versuch, besonders aufgrund der sehr guten Ergebnisse bei der Indium Messung, positiv in Erinnerung bleibt. Aufgrund des recht simplen Versuchsaufbaus, sind nur schwer maßgebliche Verbesserungsvorschläge aufzubringen. Lediglich das längere Aufnehmen einer Grundmessung, eventuell bereits während der Praktikumsbesprechung, und das Durchführen des Versuchs in separaten Versuchsräumen zur Vermeidung der Einflüsse vorbeilaufender Praktikanten mit radioaktivem Material auf die eigene Messung, sind realistische Überlegungen.

6 Anhang

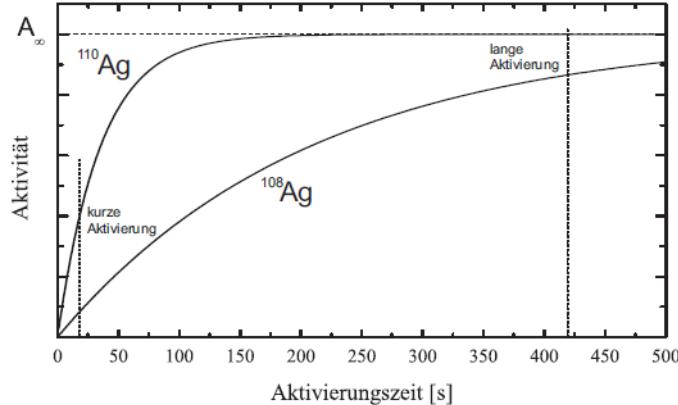


Abbildung 1: Die Aktivierungszeit von ^{108}Ag und ^{110}Ag .¹



Abbildung 2 : Ein Ausschnitt der Nuklidkarte.²

In dieser Abbildung sehen wir den für unseren Versuch relevanten Ausschnitt der Nuklidkarte. Die Farben geben hierbei das Verhalten der einzelnen Isotope an; schwarz eingefärbte sind hierbei stabil, während die rote und blaue Einfärbung einen alpha-, bzw. beta-Zerfall impliziert. Bei den für uns relevanten Isotopen ist zusätzlich die Halbwertszeit und die Energie der Strahlung angegeben. Besonders hervorzuheben ist das geteilte Feld bei ^{116}In , welches anzeigen, dass es sich um ein Isotop handelt, welches auf zwei verschiedene Weisen zerfällt und somit zwei Halbwertszeiten und Energien der Strahlung besitzt.

6.1 Abbildungsverzeichniss

^{1,2} Physikalisches Anfängerpraktikum - V. 1.1 Stand 03/2010, Seite 70,74

Imports und laden des Untergrunds, Mittelwertbestimmung

```
In [1]: import matplotlib.pyplot as plt
import numpy as np

unterg = np.loadtxt('Untergrund.dat', usecols=[1])
print(unterg)

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

[2. 4. 5. 4. 1. 4. 5. 4. 3. 2. 3. 8. 1. 3. 4. 2. 5. 2. 5. 3. 3. 6. 2.
 2. 4. 6. 2. 2. 4. 5. 7. 6. 3. 1. 3. 2. 1. 2. 2. 4. 2. 5. 5. 4. 3. 3.]
Mittelwert: 13.957446808510639 Fehler: 0.9543483893077498
```

Bestimmung der Zeitkonstante

Einladen der Messdaten für Ag und graphische Darstellung

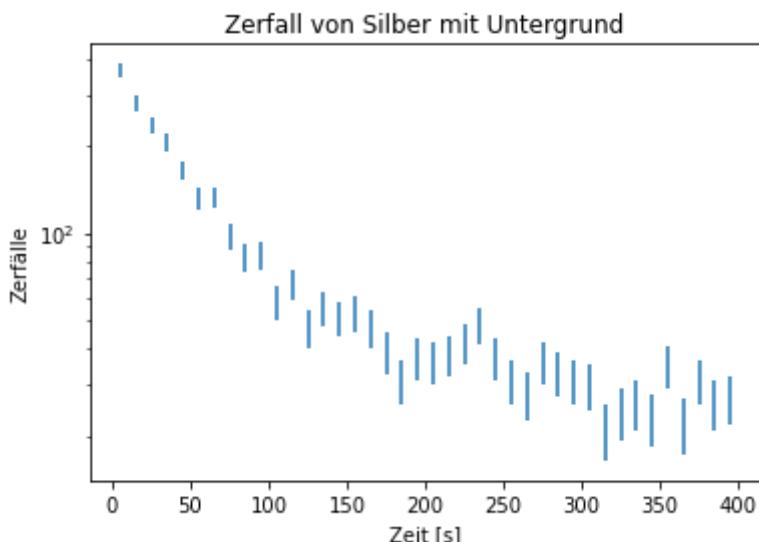
```
In [2]: n1 = np.loadtxt('Ag1.dat', usecols=[1])
n2 = np.loadtxt('Ag2.dat', usecols=[1])
n3 = np.loadtxt('Ag3.dat', usecols=[1])
n4 = np.loadtxt('Ag4.dat', usecols=[1])

t=np.arange(5,405,10)

N=n1+n2+n3+n4
Fehler_N=np.sqrt(N)

plt.errorbar(t, N, Fehler_N, linestyle='None')
plt.xlabel('Zeit [s]')
plt.ylabel('Zerfälle')

plt.title('Zerfall von Silber mit Untergrund')
plt.yscale('log')
```



Graphische Darstellung mit Ausgleichsgerade und betrachtung des Untergrunds

In [3]:

```

y0 = mw_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
p0 = [500, 0.02, 50, 0.001]
popt, pcov=curve_fit(fit_func,t,N, p0, sigma=Fehler_N)

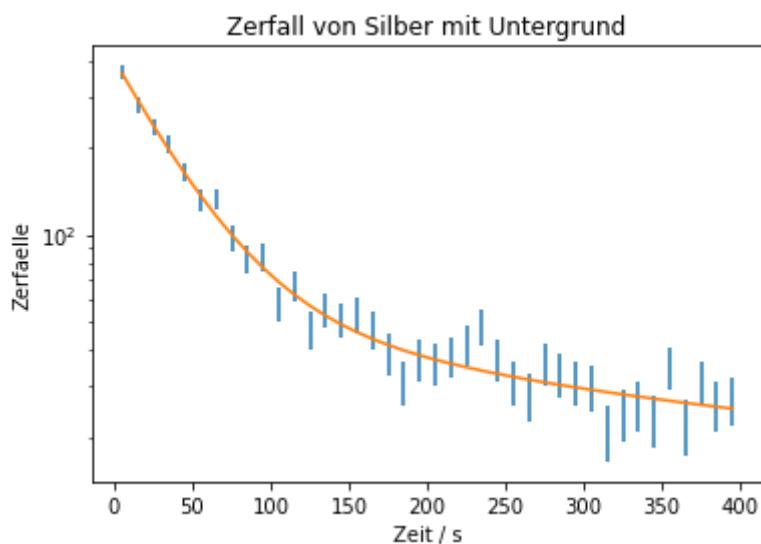
plt.errorbar(t,N, Fehler_N, linestyle='None')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Silber mit Untergrund')
plt.yscale('log')
plt.plot(t,fit_func(t,*popt))
plt.savefig('Silber1.pdf')

print("A1=",round(popt[0],1), ", Standardfehler=", round(np.sqrt(pcov[0][0]),1))
print("l1=",round(popt[1],4), ", Standardfehler=", round(np.sqrt(pcov[1][1]),4))
print("A2=",round(popt[2],1), ", Standardfehler=", round(np.sqrt(pcov[2][2]),1))
print("l2=",round(popt[3],4), ", Standardfehler=", round(np.sqrt(pcov[3][3]),4))

l1_0 = popt[1]
l2_0 = popt[3]
f_l1_0 = np.sqrt(pcov[1][1])
f_l2_0 = np.sqrt(pcov[3][3])

```

A1= 346.1 , Standardfehler= 15.8
 l1= 0.0249 , Standardfehler= 0.0022
 A2= 41.6 , Standardfehler= 12.6
 l2= 0.0034 , Standardfehler= 0.0011



Berechnung der Charakteristika des Fits

In [4]:

```

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= 28.32104042176444
chi2_red= 0.7866955672712345
Wahrscheinlichkeit= 82.0 %
```

Addition des Fehlers

```
In [5]: y0 = mw_unterg+fehler_unterg
def fit_func(x, A1,l1,A2,l2):
    return A1*np.exp(-x*l1) + A2*np.exp(-x*l2) + y0

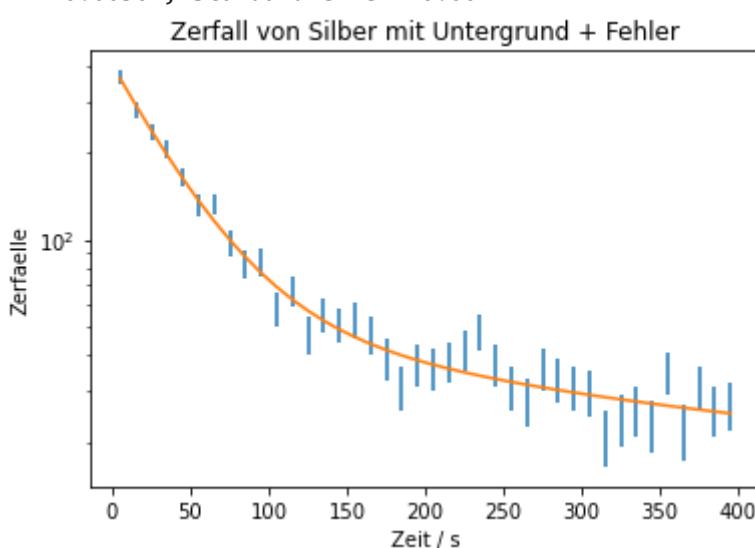
from scipy.optimize import curve_fit
p0 = [500, 0.02, 50, 0.001]
popt, pcov=curve_fit(fit_func,t,N, p0, sigma=Fehler_N)

plt.errorbar(t,N, Fehler_N, linestyle='None')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Silber mit Untergrund + Fehler')
plt.yscale('log')
plt.plot(t,fit_func(t,*popt))

print("A1=",round(popt[0],1), ", Standardfehler=", round(np.sqrt(pcov[0][0]),1))
print("l1=",round(popt[1],4), ", Standardfehler=", round(np.sqrt(pcov[1][1]),4))
print("A2=",round(popt[2],1), ", Standardfehler=", round(np.sqrt(pcov[2][2]),1))
print("l2=",round(popt[3],4), ", Standardfehler=", round(np.sqrt(pcov[3][3]),4))

l1_0 = popt[1]
l2_0 = popt[3]
f_l1_0 = np.sqrt(pcov[1][1])
f_l2_0 = np.sqrt(pcov[1][1])
```

A1= 345.4 , Standardfehler= 16.0
l1= 0.0249 , Standardfehler= 0.0022
A2= 41.5 , Standardfehler= 13.3
l2= 0.0036 , Standardfehler= 0.0011



```
In [6]: 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= 28.332663427791978
chi2_red= 0.7870184285497772
Wahrscheinlichkeit= 82.0 %

```

Speichern der Maxima

```
In [7]: l1_plu = popt[1]
l2_plu = popt[3]
f_l1_plu = np.sqrt(pcov[1][1])
f_l2_plu = np.sqrt(pcov[3][3])
```

Abzug des Fehlers

```
In [8]: y0 = mw_unterg-fehler_unterg
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
p0 = [500, 0.02, 50, 0.001]
popt, pcov=curve_fit(fit_func,t,N, p0, sigma=Fehler_N)

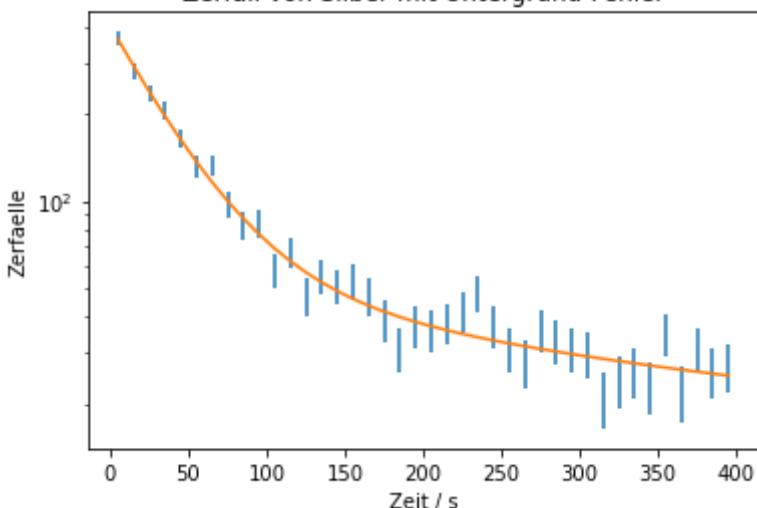
plt.errorbar(t,N, Fehler_N, linestyle='None')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Silber mit Untergrund-Fehler')
plt.yscale('log')
plt.plot(t,fit_func(t,*popt))

print("A1=",round(popt[0],1), ", Standardfehler=", round(np.sqrt(pcov[0][0]),1))
print("l1=",round(popt[1],4), ", Standardfehler=", round(np.sqrt(pcov[1][1]),4))
print("A2=",round(popt[2],1), ", Standardfehler=", round(np.sqrt(pcov[2][2]),1))
print("l2=",round(popt[3],4), ", Standardfehler=", round(np.sqrt(pcov[3][3]),4))

l1_0 = popt[1]
l2_0 = popt[3]
f_l1_0 = np.sqrt(pcov[1][1])
f_l2_0 = np.sqrt(pcov[3][3])
```

A1= 346.8 , Standardfehler= 15.6
l1= 0.0248 , Standardfehler= 0.0021
A2= 41.9 , Standardfehler= 12.0
l2= 0.0032 , Standardfehler= 0.001

Zerfall von Silber mit Untergrund-Fehler



Berechnung der Charakteristika des Fits

```
In [9]: 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= 28.311783773032968
chi2_red= 0.7864384381398046
Wahrscheinlichkeit= 82.0 %
```

Speichern der Minima

```
In [10]: l1_min = popt[1]
l2_min = popt[3]
f_l1_min = np.sqrt(pcov[1][1])
f_l2_min = np.sqrt(pcov[3][3])
```

Berechnung des Fehlers

```
In [11]: d1_l1 = abs(l1_0 - l1_min)
d2_l1 = abs(l1_0 - l1_plu)
d1_l2 = abs(l2_0 - l2_min)
d2_l2 = abs(l2_0 - l2_plu)
f_l1 = np.sqrt(((d1_l1+d2_l1)/2)**2+(f_l1_0)**2)
f_l2 = np.sqrt(((d1_l2+d2_l2)/2)**2+(f_l2_0)**2)

print('Zerfallskonstantenfehler l1:', f_l1, 'Zerfallskonstantenfehler l2:', f_l2)
```

Zerfallskonstantenfehler l1: 0.0021115029323168243 Zerfallskonstantenfehler l2: 0.002119903810450295

```
In [12]: T_1 = np.log(2)/l1_0
f_T_1 = f_l1*np.log(2)/(l1_0**2)
T_2 = np.log(2)/l2_0
f_T_2 = f_l2*np.log(2)/(l2_0**2)

print('Halbwertszeit AG110=',T_1,'+-',f_T_1)
print('Halbwertszeit AG108=',T_2,'+-',f_T_2)
```

Halbwertszeit AG110= 27.940407838207708 +- 2.3781087454814096
 Halbwertszeit AG108= 218.4742397618491 +- 145.97926343330383

```
In [13]: tau_AG110 = 1/T_1
f_tau_AG110 = f_T_1/(T_1**2)
tau_AG108 = 1/T_2
f_tau_AG108 = f_T_2/(T_2**2)

print('tau_AG110=',tau_AG110,'+-',f_tau_AG110)
print('tau_AG108=',tau_AG108,'+-',f_tau_AG108)
```

tau_AG110= 0.03579045824207794 +- 0.0030462548092759878
 tau_AG108= 0.004577198671523306 +- 0.003058374714498257

Indium Zerfall

```
In [14]: unterg = np.loadtxt('Untergrund.dat', usecols=[1])*12
```

```

print(unterg)

mw_unterg=np.mean(unterg)
fehler_unterg=np.std(unterg)/np.sqrt(len(unterg))
print('Mittelwert:', mw_unterg, 'Fehler:', fehler_unterg)
n1 = np.loadtxt('In1.dat', usecols=[1])

N=n1
Fehler_N=np.sqrt(N)

t=np.arange(5,2880,115)

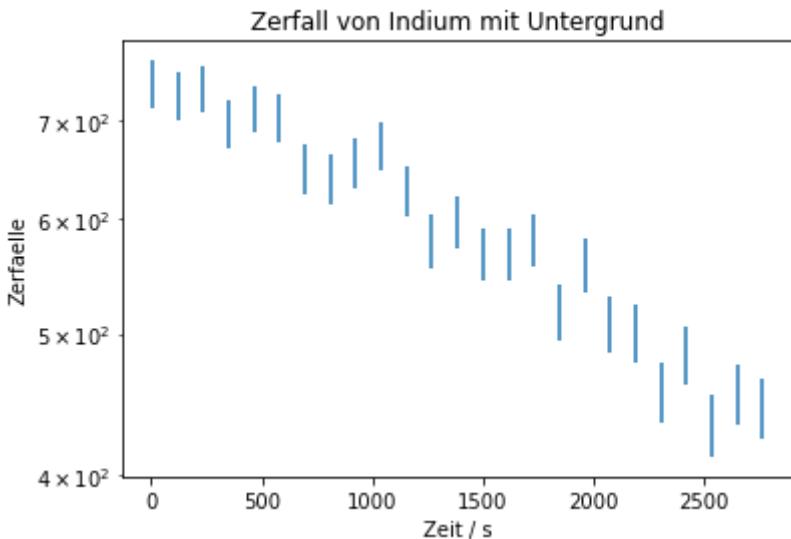
plt.errorbar(t, N, Fehler_N, linestyle='None')
plt.xlabel('Zeit / s')

plt.ylabel('Zerfaelle')

plt.title('Zerfall von Indium mit Untergrund')
plt.yscale('log')

```

[24. 48. 60. 48. 12. 48. 60. 48. 60. 36. 24. 36. 96. 12. 36. 48. 24. 60.
 24. 60. 36. 36. 72. 24. 24. 48. 72. 24. 24. 48. 60. 84. 72. 36. 12. 36.
 24. 12. 24. 24. 48. 24. 60. 60. 48. 36. 36.]
 Mittelwert: 41.87234042553192 Fehler: 2.8630451679232496



Ausgleichsgerade

```

In [15]: y0=mw_unterg #Untergrund Indium
def fit_func(x, A1,l1):
    return A1*np.exp(-x*l1) + y0

p0 = [680, 0.0002]
popt, pcov=curve_fit(fit_func,t[1:],N[1:], p0, sigma=Fehler_N[1:])

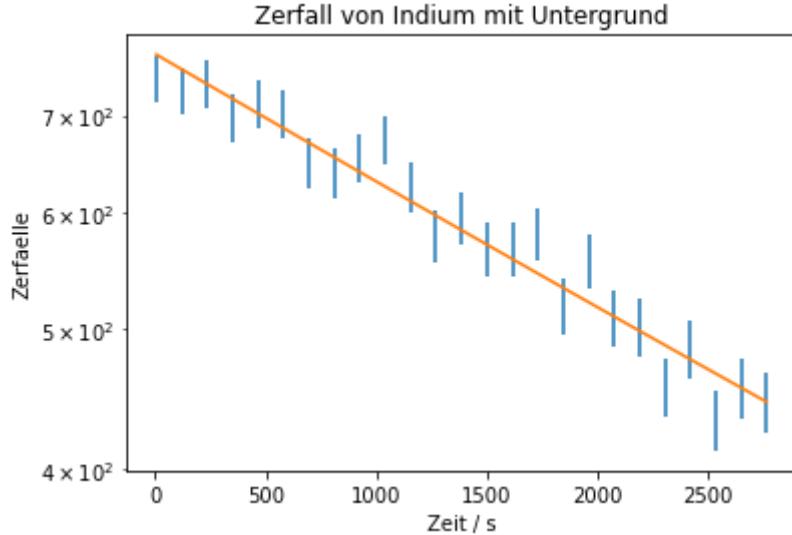
plt.errorbar(t,N, Fehler_N, linestyle='None')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Indium mit Untergrund')
plt.yscale('log')
plt.plot(t, fit_func(t, *popt))
plt.savefig('Indium1.pdf')

print("A1=",popt[0], ", Standartfehler=", np.sqrt(pcov[0][0]))
print("l1=",popt[1], ", Standartfehler=", np.sqrt(pcov[1][1]))

```

```
l_0 = popt[1]
f_l_0 = np.sqrt(pcov[1][1])
```

A1= 729.6781327901697 , Standartfehler= 11.409087849049756
 l1= 0.00021440960466942893 , Standartfehler= 1.0282710258966877e-05



Berechnung der Charakteristika des Fits

```
In [16]: chi2=np.sum((fit_func(t,*popt)-N)**2/Fehler_N**2)
dof=len(N)-2 #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= 18.812875964287013
chi2_red= 0.817951128882044
wahrscheinlichkeit= 71.0 %
```

Addition des Fehlers

```
In [17]: y0=mw_unterg+fehler_unterg #Untergrund Indium+F
def fit_func(x, A1,l1):
    return A1*np.exp(-x*l1) + y0

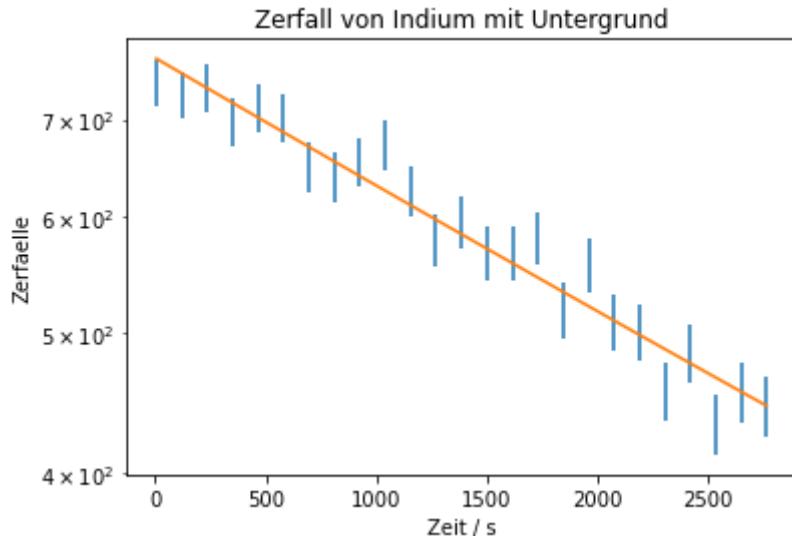
p0 = [680, 0.0002]
popt, pcov=curve_fit(fit_func,t[1:],N[1:], p0, sigma=Fehler_N[1:])

plt.errorbar(t,N, Fehler_N, linestyle='None')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Indium mit Untergrund')
plt.yscale('log')
plt.plot(t, fit_func(t, *popt))

print("A1_plu=",popt[0], ", Standartfehler=", np.sqrt(pcov[0][0]))
print("l1_plu=",popt[1], ", Standartfehler=", 10*np.sqrt(pcov[1][1]))

l_plu = popt[1]
f_l_plu = np.sqrt(pcov[1][1])
```

A1_plu= 726.9089456971919 , Standartfehler= 11.426241454973267
 l1_plu= 0.00021554278097788877 , Standartfehler= 0.00010346222633056648



Berechnung der Charakteristika des Fits

```
In [18]: chi2_=np.sum((fit_func(t,*popt)-N)**2/Fehler_N**2)
dof=len(N)-2 #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= 18.844437822979618
chi2_red= 0.8193233836078094
wahrscheinlichkeit= 71.0 %
```

Abzug des Fehlers

```
In [19]: y0=mw_unterg-fehler_unterg #Untergrund Indium+F
def fit_func(x, A1,l1):
    return A1*np.exp(-x*l1) + y0

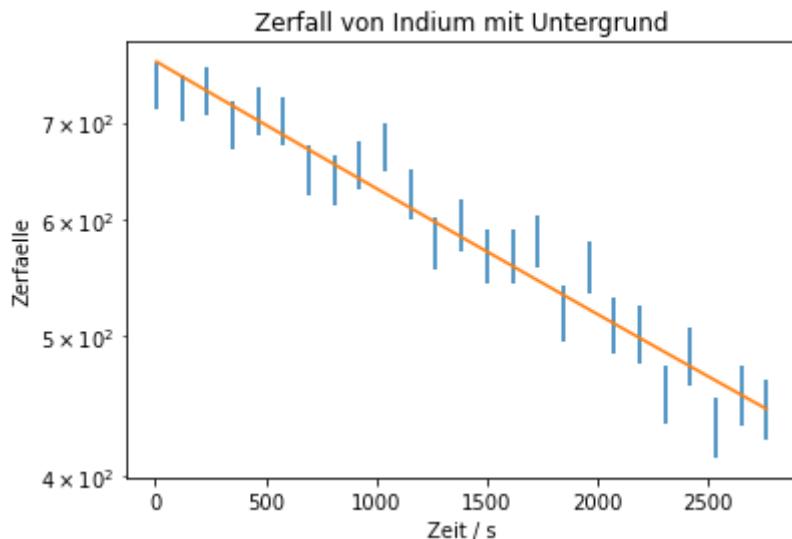
p0 = [680, 0.0002]
popt, pcov=curve_fit(fit_func,t[1:],N[1:], p0, sigma=Fehler_N[1:])

plt.errorbar(t,N, Fehler_N, linestyle='None')
plt.xlabel('Zeit / s')
plt.ylabel('Zerfaelle')
plt.title('Zerfall von Indium mit Untergrund')
plt.yscale('log')
plt.plot(t, fit_func(t, *popt))

print("A1_min=",popt[0], ", Standartfehler=", np.sqrt(pcov[0][0]))
print("l1_min=",popt[1], ", Standartfehler=", np.sqrt(pcov[1][1]))

l_min = popt[1]
f_l_min = np.sqrt(pcov[1][1])

A1_min= 732.4482372645722 , Standartfehler= 11.392160361196552
l1_min= 0.00021328821090894383 , Standartfehler= 1.022000020847493e-05
```



Berechnung der Charakteristika des Fits

```
In [20]: chi2_=np.sum((fit_func(t,*popt)-N)**2/Fehler_N**2)
dof=len(N)-2 #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= 18.78177834536769
chi2_red= 0.8165990584942474
wahrscheinlichkeit= 71.0 %
```

Berechnung des Fehlers

```
In [21]: d1_l_1 = abs(l_0-l_min)
d2_l_1 = abs(l_0-l_plu)

f_l_1 = np.sqrt(((d1_l_1+d2_l_1)/2)**2+(f_l_0)**2)

print('Zerfallsfehler:', f_l_1)
```

Zerfallsfehler: 1.0344317368430275e-05

```
In [22]: T = np.log(2)/l_0
f_T = f_l_0*np.log(2)/(l_0**2)

print('Halbwertszeit In=',T,'+-',f_T)
```

Halbwertszeit In= 3232.8177724529705 +- 155.04029553818003

```
In [23]: tau_In = 1/T
f_tau_In = f_T/(T**2)

print('tau_In=',tau_AG110,'+-',f_tau_AG110)
```

tau_In= 0.03579045824207794 +- 0.0030462548092759878