

Physikalisches Anfängerpraktikum II für Lehramtsstudierende

Name: Marius Pfeiffer

Matrikel-Nr.: 4188573

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

Betreut durch: Selina Pohl

12.11.2024

Versuch 251: Statistik des radioaktiven Zerfalls

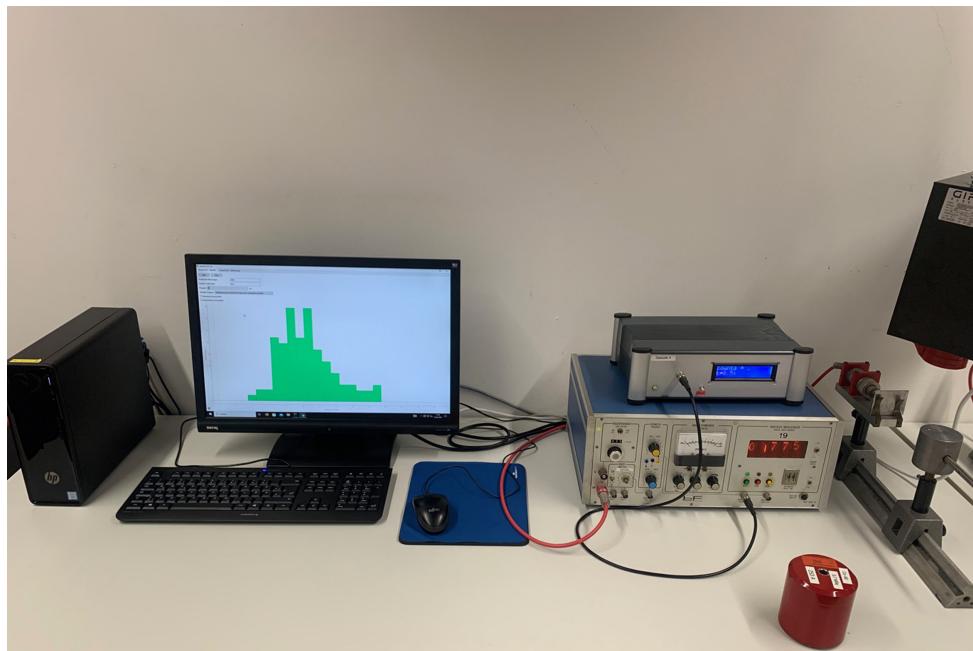


Abbildung 1: Versuchsaufbau

Inhaltsverzeichnis

1 Einleitung	2
1.1 Physikalische Grundlagen	2
1.2 Versuchsdurchführung	5
2 Messprotokoll	6
3 Auswertung	8
3.1 Zählrohrcharakteristik und Plateauanstieg	8
3.2 Statistik bei hoher mittlerer Ereigniszahl	11
3.3 Statistik bei kleiner Ereigniszahl	13
4 Zusammenfassung und Diskussion	15

1 Einleitung

Während es unmöglich ist, einen Zerfall eines einzelnen Atomkerns vorauszusagen, so ist es bei der Betrachtung einer großen Anzahl von Zerfallsprozessen möglich, statistische Aussagen und Voraussagen über die Gesamtheit der Zerfälle zu treffen. In diesem Versuch wird die Statistik von einer großen Anzahl an radioaktiven Zerfällen eines Präparats mit einem Geiger-Müller-Zählrohr aufgezeichnet und die Statistik der Zerfälle hingehend verschiedener Wahrscheinlichkeitsverteilungen untersucht.

1.1 Physikalische Grundlagen

Geiger-Müller-Zählrohr

Ein Geiger-Müller-Zählrohr kann verwendet werden, um verschiedene Arten von Strahlung nachzuweisen. Aufgebaut ist es, wie in Abbildung (3) zu sehen, aus einem zylinderförmigen Gehäuse, welches auf der Vorderseite mit einem Fenster versehen und auf der Rückseite geschlossen ist. Am Zählrohr liegt eine Spannung von mehreren 100 bis 1000 Volt an, wobei das Gehäuse selbst als Kathode dient während sich im Gehäuse ein Anodendraht befindet. Das Zählrohr ist mit einem Gas gefüllt, in welchem es bei Kontakt mit einem geladenen Teilchen, welches durch das Fenster in das Zählrohr eindringt, zu einer Ionisation und somit der Erzeugung freier Elektronen kommt. Diese verursachen über eine Gasentladung einen kurzen Stromfluss im Zählrohr, welcher durch eine Schaltung registriert werden kann.

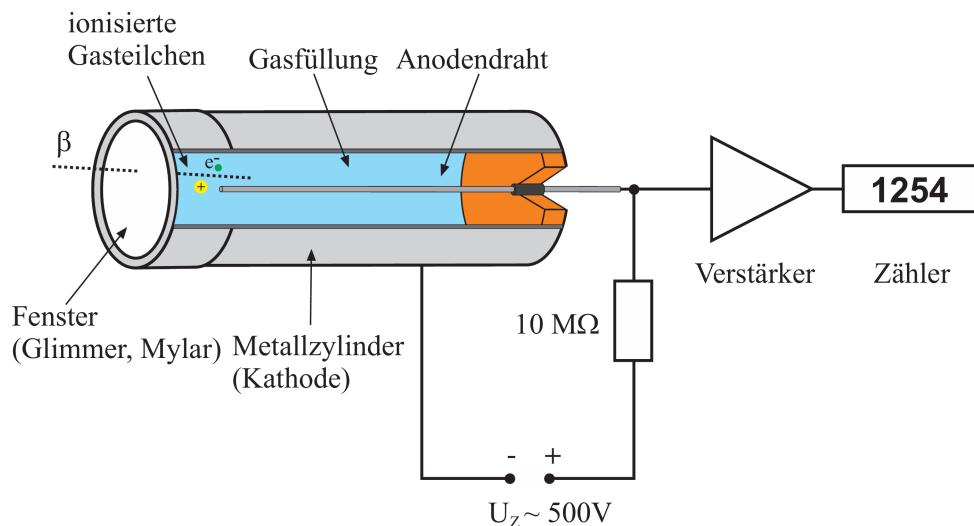


Abbildung 2: Schematische Darstellung des Geiger-Müller-Zählrohrs

Für den korrekten Betrieb des Zählrohrs ist die Wahl der anlegten Spannung von großer Bedeutung. Ab einer gewissen Einsatzspannung U_E befindet man sich im sogenannten Plateaubereich. In diesem wird das Zählrohr optimalerweise für das Detektieren von ionisierender Strahlung und Zählanwendungen, wie in diesem Versuch relevant, betrieben. Die Beschleunigung der durch die Ionisation primär erzeugten freien Elektronen ist in hier so hoch, dass diese eine Lawine an propagierenden Ionisationen im Füllgas entlang des Anodendrahts auslösen. Hierdurch wird erzielt, dass jedes eindringende Teilchen, unabhängig von seiner Energie, ein gleich großes Entladungssignal auslöst.

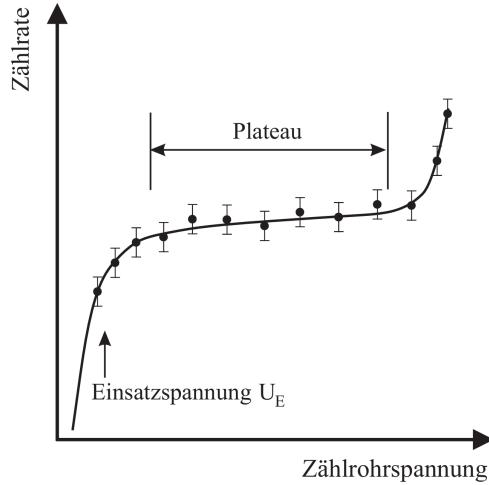


Abbildung 3: Plateaubereich des Geiger-Müller-Zählrohrs

Die Ermittlung des Plateaubereichs wird hier der erste Teil der Versuchsdurchführung sein.

Wahrscheinlichkeitsverteilungen

Wie im Einleitungstext kurz beschrieben, lassen sich radioaktiven Zerfälle, auch wenn diese im einzelnen komplett zufällig auftreten, in einer Gesamtheit sehr vieler Zerfälle in Form von Wahrscheinlichkeitsverteilungen erfassen, untersuchen und nutzen, um Vorhersagen zu tätigen. Die hierbei relevanten Verteilungen sind die Binomial-, die Poisson- und die Gauss-Verteilung.

Binomialverteilung. Die Binomialverteilung gibt die Wahrscheinlichkeit dafür an, dass ein Ereignis mit der Eintrittswahrscheinlichkeit p bei n voneinander unabhängigen Versuchen genau k mal eintritt. Die Verteilungsfunktion dieser ist gegeben durch

$$B(k; n, p) = \binom{n}{k} p^k (1-p)^{n-k}. \quad (1)$$

Hierbei sind $n, k \in \mathbb{N}$, es handelt sich also um eine diskrete Wahrscheinlichkeitsverteilung. Für die Binomialverteilung gilt

$$\text{Mittelwert: } \langle k \rangle = np, \quad (2)$$

$$\text{Varianz: } \sigma^2 = np(1-p). \quad (3)$$

Übertragen auf radioaktive Zerfälle beschreibt die Binomialverteilung die Wahrscheinlichkeit, dass von n Atomkernen innerhalb einer Zeit t genau k Kerne zerfallen sind. Hierbei geht die Zeit in die Beschreibung der Wahrscheinlichkeit

$$p(t) = 1 - e^{-\lambda t} \quad (4)$$

ein, da zu erwarten ist, dass nach einer längeren Zeit eine größere Anzahl an Zerfällen zu beobachten ist. λ ist die für ein Isotop charakteristische Zerfallskonstante.

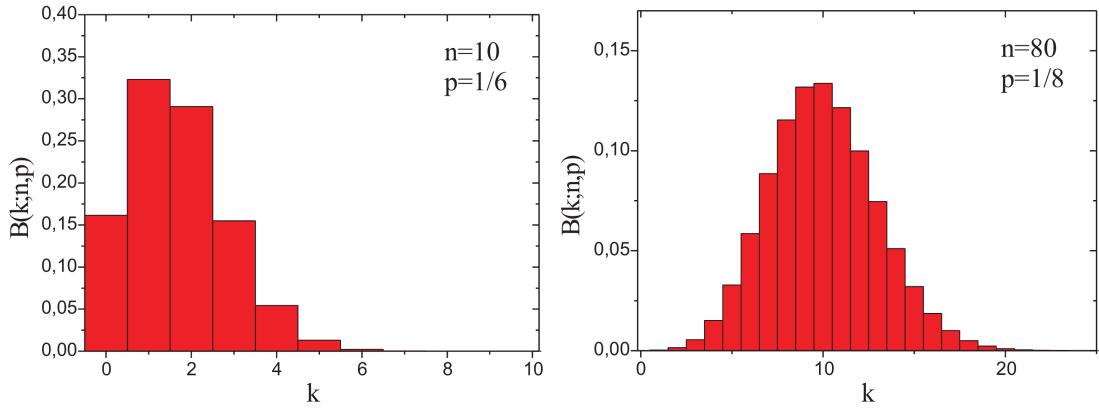


Abbildung 4: Beispiele für Binomialverteilungen

Poissonverteilung. In vielen Fällen ist die Zahl der betrachteten Atome sehr groß ($n \rightarrow \infty$) und die Zerfallswahrscheinlichkeit ($p \rightarrow 0$) sehr klein. Mit diesen Grenzüberlegungen kann, unter Beachtung, dass der Mittelwert endlich ist, von der Binomial- zu einer Poissonverteilung übergegangen werden. Die Wahrscheinlichkeitsverteilung diese ist gegeben durch

$$P(k; \mu) = \frac{\mu^k e^{-\mu}}{k!}. \quad (5)$$

Auch hier gilt, dass $k \in \mathbb{N}$. Weiter sind die statistischen Eigenschaften gegeben durch

$$\text{Mittelwert: } \langle k \rangle = \mu, \quad (6)$$

$$\text{Varianz: } \sigma^2 = \mu. \quad (7)$$

Aus der Tatsache, dass $\sigma = \sqrt{\mu}$ ergibt sich das \sqrt{N} -Gesetz, welches für die Fehlerbestimmung von gezählten Größen genutzt wird. Für $\mu < 1$ ist die Poissonverteilung in k monoton fallend, besitzt kein Maximum und der wahrscheinlichste Wert ist stets 0. Für $\mu > 1$ besitzt die Verteilung ein Maximum, dessen Breite durch σ^2 gegeben ist.

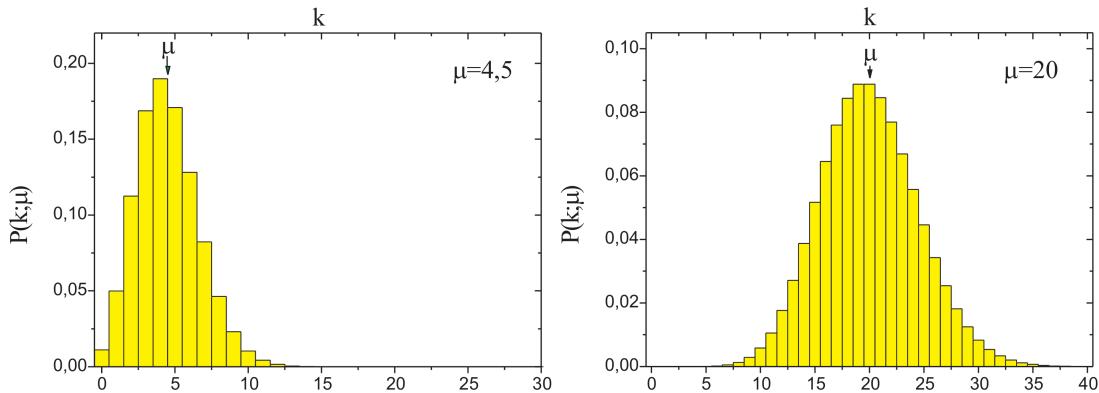


Abbildung 5: Beispiele für Poisson-Verteilungen

Gauß-Verteilung. Um das Maximum ist die Poissonverteilung für kleine Mittelwerte stark asymmetrisch. Mit steigendem μ nimmt sie eine zunehmend symmetrische Form an und kann für große Mittelwerte ($\mu > 30$) schließlich mit der Gauß-Verteilung approximiert werden. Die allgemeine Verteilungsfunktion der Gauß-Verteilung lautet

$$G(k; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\mu-k)^2}{2\sigma^2}}, \quad (8)$$

wobei nun $k \in \mathbb{R}$ gilt, es sich also um eine kontinuierliche Verteilung handelt. Mittelwert und Varianz sind hier bereits durch μ und σ^2 in der Verteilungsfunktion angegeben. Bekannterweise gibt μ die Position des Maximums an, während σ die Breite der Verteilung angibt.

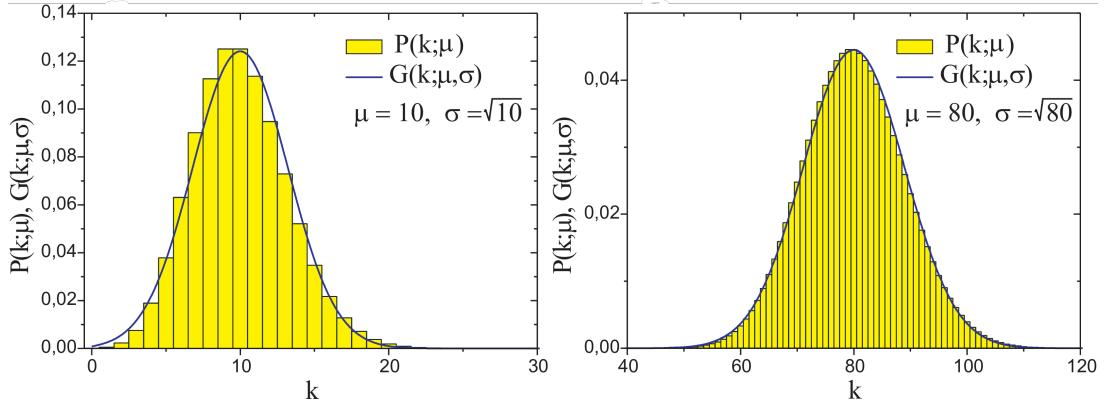


Abbildung 6: Beispiele für Gauß-Verteilungen

1.2 Versuchsdurchführung

Im ersten Teil des Versuchs werden wir uns mit der Kalibrierung des Zählrohrs beschäftigen und den Plateaubereich untersuchen. Im zweiten Teil nehmen wir Daten auf, um die Statistik der radioaktiven Zerfälle bei großer und kleiner mittlerer Ereigniszahl zu untersuchen. Als radioaktives Präparat wird im ganzen Versuch der β -Strahler Kobalt-60 (^{60}Co) verwendet.

Bestimmung der Zählrohrcharakteristik. Ab einer angelegten Einsatzspannung V_E beginnt das Zählrohr, eingehende Strahlung zu registrieren. Um den Plateaubereich zu bestimmen, wird die Spannung ausgehend von V_E um 150V in Schritten von 25V erhöht und jeweils die Anzahl der Ereignisse pro 30 Sekunden erfasst. Durch die Untersuchung der Zählraten lässt sich der Plateaubereich und die optimale Spannung U_0 für den weiteren Betrieb bestimmen.

Untersuchung des Plateauanstiegs. Innerhalb des Plateaubereichs sollte die Zählrate nicht von der Zählrohrspannung abhängig sein. Um dies zu verifizieren, wird die Probe direkt vor dem Zählrohr platziert und die Zählraten je zweimal für eine Minute und zweimal für drei Minuten bei U_0 und $U_0 + 100\text{V}$ gemessen.

Verifizierung der statistischen Natur des radioaktiven Zerfalls. Für diesen Versuchsteil wird das Präparat so platziert, dass ca. 140 bis 150 Zerfälle pro Sekunde registriert werden. In Intervallen von 500ms (entspricht der Torzeit) werden nun für mindestens 2000 Male die Anzahl der Zerfälle des Präparats bestimmt und in ein Histogramm eingetragen.

Vergleich der Poisson- und Gauß- Verteilung bei sehr kleinen Zählraten. Das Präparat wird nun so platziert, dass ca. 40 bis 50 Zerfälle pro Sekunde registriert werden. Bei einer Torzeit von 100ms werden 5000 Messungen durchgeführt und ebenfalls in ein Histogramm eingetragen.

2 Messprotokoll

Messprotokoll 251

Marius Pfeiffer

12.11.2024

Robert Gusch

Aufgabe 2

Spannung [V]	Gezählte Ereignisse
460 ± 10	1674
485 ± 10	1936
510 ± 10	2102
535 ± 10	2084
560 ± 10	2114
585 ± 10	2097
610 ± 10	2118

✓

Messzeit jeweils 30s

Aufgabe 3 mit $U_0 = 560 \text{ V}$

Messung über 1 min

$$N_1(U=U_0) = 10566$$

$$N_1(U=U_0 + 100 \text{ V}) = 10730$$

Messung über 3 min

$$N_3(U=U_0) = 31139$$

$$N_3(U=U_0 + 100 \text{ V}) = 31756$$

✓

Aufgabe 4

Erwartungswert: 71.05

Wurzel Erwartungswert: 8.43

Sigma für Gauskurve: 8.23

Aufgabe 5

Erwartungswert: 5.01

Wurzel Erwartungswert: 2.24

Sigma für Gauskurve: 2.25

s. Bell

3 Auswertung

3.1 Zählrohrcharakteristik und Plateauanstieg

Im ersten Teil des Versuchs befassten wir uns mit der Bestimmung der Zählrohrcharakteristik und der Untersuchung des Plateauanstiegs. Die gemessene Anzahlen an Ereignissen für die jeweilige Zählrohrspannung sind Abbildung (7) zu entnehmen.

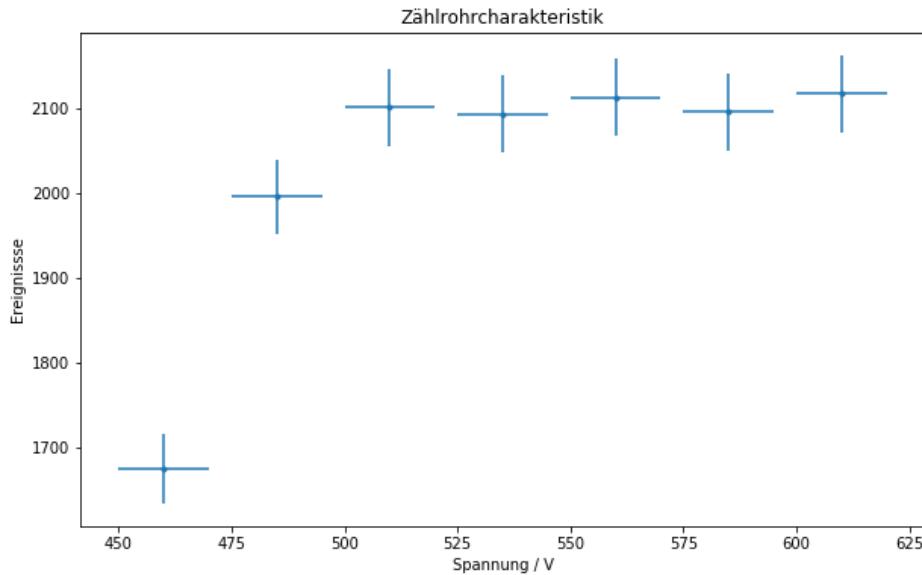


Abbildung 7: Messung der Zählrohrcharakteristik

Wie bereits in diesem Plot zu erkennen ist, stellt sich ab dem dritten Messpunkt ein plateauartiges Verhalten ein. Wir wählen diesen und die nachfolgenden vier Datenpunkte, um einen linearen Fit durchzuführen, zu sehen in Abbildung (8).

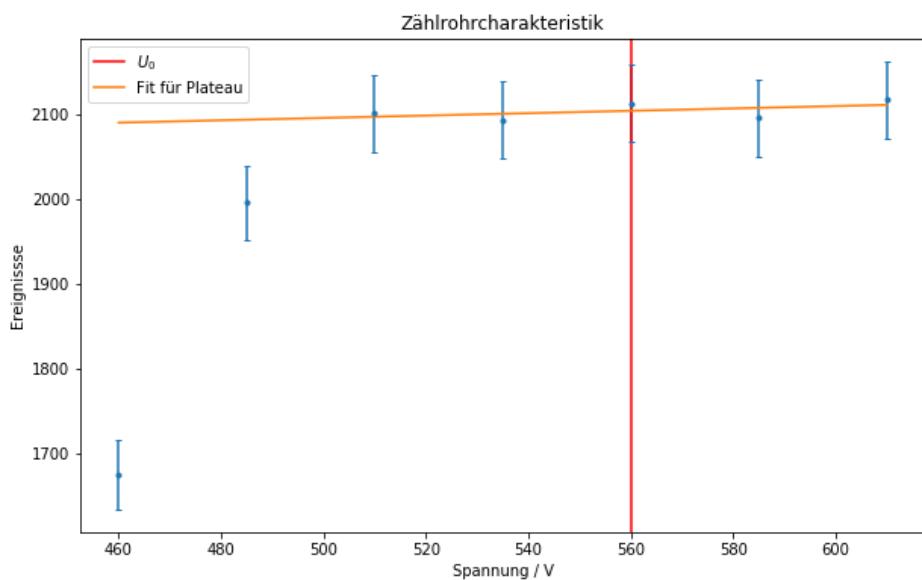


Abbildung 8: Messung der Zählrohrcharakteristik mit Fit an Plateaubereich

Die Steigung der optimierten Gerade geht gegen 0, was auf den Plateaubereich der Zählrohrcharakteristik hinweist. Die Spannung U_0 für den weiteren Versuchsverlauf ermitteln wir als Mittelwert über die fünf Datenpunkte zu $U_0 = 560\text{V}$.

Zur Untersuchung des Plateauanstieges nahmen wir bei U_0 , sowie $U_0 + 100\text{V}$ über eine und drei Minuten die folgenden Anzahlen an Ereignissen auf:

t [min]	U [V]	U_0	$U_0 + 100\text{V}$	Δ	Anstieg [%]	σ -Abw.
1 min		10566 ± 103	10730 ± 104	164 ± 146	1.55 ± 1.38	1.12
3 min		31199 ± 177	31756 ± 179	557 ± 251	1.79 ± 0.8	2.22

Tabelle 1: Anzahl N der Ereignisse zur Untersuchung des Plateauanstiegs

In der Tabelle entspricht der Wert Δ der Differenz

$$\Delta = N_{U_0+100} - N_{U_0} \quad (9)$$

mit dem Fehler

$$\sigma_\Delta = \sqrt{N_{U_0+100} + N_{U_0}}. \quad (10)$$

Der prozentuale Anstieg ist durch

$$p\% = \frac{N_{U_0+100} - N_{U_0}}{N_{U_0}} \quad (11)$$

mit dem Fehler

$$\Delta p\% = p\% \cdot \sqrt{\left(\frac{\Delta N_{U_0}}{N_{U_0}}\right)^2 + \left(\frac{\sigma_\Delta}{\Delta}\right)^2} \quad (12)$$

gegeben. Die σ -Abweichung ist durch das Verhältnis Δ/σ_Δ gegeben. Für beide Messzeiten ist der Anstieg der Anzahl an Ereignissen von U_0 zu $U_0 + 100\text{V}$ innerhalb des 3σ -Bereichs und gilt somit als nicht signifikant.

Nun möchten wir berechnen, wie lange wir hätten messen müssen, um einen prozentualen Fehler $\frac{\sigma_\Delta}{\Delta}$ von 1% zu erhalten. Hierzu Betrachten wir zu einer Anzahl N die Zählrate $n = \frac{N}{t}$ pro Sekunde, um folgende Formel herzuleiten:

$$\frac{\sigma_\Delta}{\Delta} = \frac{\sqrt{N_{U_0+100} + N_{U_0}}}{N_{U_0+100} - N_{U_0}} = \frac{\sqrt{tn_{U_0+100} + tn_{U_0}}}{tn_{U_0+100} - tn_{U_0}} = \frac{\sqrt{t}}{t} \frac{\sqrt{n_{U_0+100} + n_{U_0}}}{n_{U_0+100} - n_{U_0}} \quad (13)$$

$$= \frac{1}{\sqrt{t}} \frac{\sqrt{n_{U_0+100} + n_{U_0}}}{n_{U_0+100} - n_{U_0}}. \quad (14)$$

Die Zählraten n_x sind für alle t gleich, somit gilt

$$\frac{\sigma_\Delta}{\Delta} \sqrt{t} = \text{const.} \quad (15)$$

Mit dem relativen Fehler, berechnet aus den Werten in Tabelle 1 und der Messzeit t_{mess} von 1min bzw. 3min können wir die Formel wie folgt umstellen

$$\frac{\sigma_\Delta}{\Delta} \sqrt{t_{mess}} = 1\% \cdot \sqrt{t_{1\%}} \quad (16)$$

$$\Leftrightarrow t_{1\%} = \left(\frac{\sigma_\Delta}{\Delta} \frac{\sqrt{t_{mess}}}{1\%} \right)^2. \quad (17)$$

Setzen wir nun die Werte aus Tabelle 1, so erhalten wir

$$t_{1\%,1\text{min}} = \left(\frac{146}{164} \frac{\sqrt{1\text{min}}}{1\%} \right)^2 = 7917.91\text{min} = 5.50\text{d} \quad (18)$$

aus dem Plateauanstieg über 1min und

(19)

$$t_{1\%,3\text{min}} = \left(\frac{251}{557} \frac{\sqrt{3\text{min}}}{1\%} \right)^2 = 6087.53\text{min} = 4.23\text{d} \quad (20)$$

aus dem Plateauanstieg über 3min.

Abschließend zu diesem Versuchsteil möchten wir nun die maximal mögliche prozentuale Variation von U_0 zu $U_0 + 100\text{V}$ berechnen, wenn wir Anzahlen in einem Vertrauensbereich von 68% und 95% zulassen. Dies entspricht etwa den 1σ - bzw. 2σ -Bereichen.

Den maximalen Anstieg, wenn wir eine Abweichung von 1σ zulassen erhalten wir durch

$$\Delta_{\max,1\sigma} = (U_{0+100} + \Delta U_{0+100}) - (U_0 - \Delta U_0) \quad (21)$$

$$= U_{0+100} + \Delta U_{0+100} - U_0 + \Delta U_0, \quad (22)$$

lassen also zu, dass U_0 um ein σ niedriger und U_{0+100} ein σ höher ist. Analog gilt für den 2σ -Bereich

$$\Delta_{\max,2\sigma} = (U_{0+100} + 2 \cdot \Delta U_{0+100}) - (U_0 - 2 \cdot \Delta U_0) \quad (23)$$

$$= U_{0+100} + 2 \cdot \Delta U_{0+100} - U_0 + 2 \cdot \Delta U_0. \quad (24)$$

Die prozentuale Variation rechnen wir, wie oben, wieder ausgehend von U_0 , diesmal aber mit der eingerechneten Abweichung.

$$p_{\%,\max,1\sigma} = \frac{\Delta_{\max,1\sigma}}{U_0 - \Delta U_0} \quad (25)$$

$$p_{\%,\max,2\sigma} = \frac{\Delta_{\max,2\sigma}}{U_0 - 2 \cdot \Delta U_0}. \quad (26)$$

In diese Formeln können wir nun erneut die Werte aus Tabelle 1 einsetzen und erhalten die in Tabelle 2 aufgezeigten Resultate.

Messzeit	Abweichung im Vertrauensbereich	
	$1\sigma \approx 68\%$	$2\sigma \approx 95\%$
1min	3.54 %	5.57 %
3min	2.94 %	4.11 %

Tabelle 2: Mögliche prozentuale Variation der Zählrate im Vertrauensbereich von 68% bzw. 95%

3.2 Statistik bei hoher mittlerer Ereigniszahl

Plot 9 zeigt die jeweiligen Häufigkeiten, wie oft eine Anzahl an Zerfällen pro Zeiteinheit aufgetreten ist, als Histogramm. Es wurden hierfür 2000 Messungen für Zeitintervalle von 500ms von einem Computer aufgenommen.

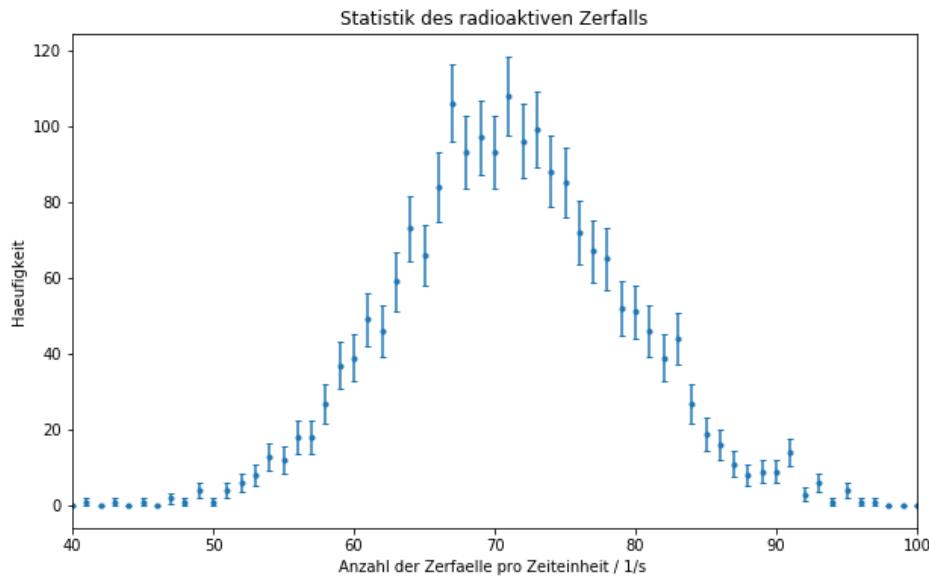


Abbildung 9: Verteilung der Zählraten bei hoher mittlerer Ereigniszahl

Bereits in diesem Plot erkennen wir, dass die Form der Verteilung einer Poisson- oder Gaußverteilung ähnelt. Um dies quantitativ zu untersuchen, werden wir nun an das Histogramm eine Gauß- und eine Poisson-Verteilungsfunktion fitten. Die verwendete Fitmethode verlangt, dass die Fehler gaussverteilt sind, was wir mit einer Einschränkung des Fit-Bereichs auf Klassen mit einer Häufigkeit größer als zehn erreichen. Die Definition der Verteilungsfunktionen ist den Formeln 5 und 8 des Theorieteils dieser Auswertung zu entnehmen. Als Startparameter für die Fits wählen wir die Werte aus Tabelle 3. μ und σ wurden vom Computer bei der Aufzeichnung ermittelt, den Startwert für die Fläche A unter der Kurve entnehmen wir der Praktikumsanleitung.

Parameter	Wert
μ	71.05
σ	8.23
A	2000

Tabelle 3: Startparameter für Gauß- und Poisson-Fit

Plot 10 zeigt das Ergebnis des Fits, in Tabelle 4 sind die optimierten Parameter zu finden.

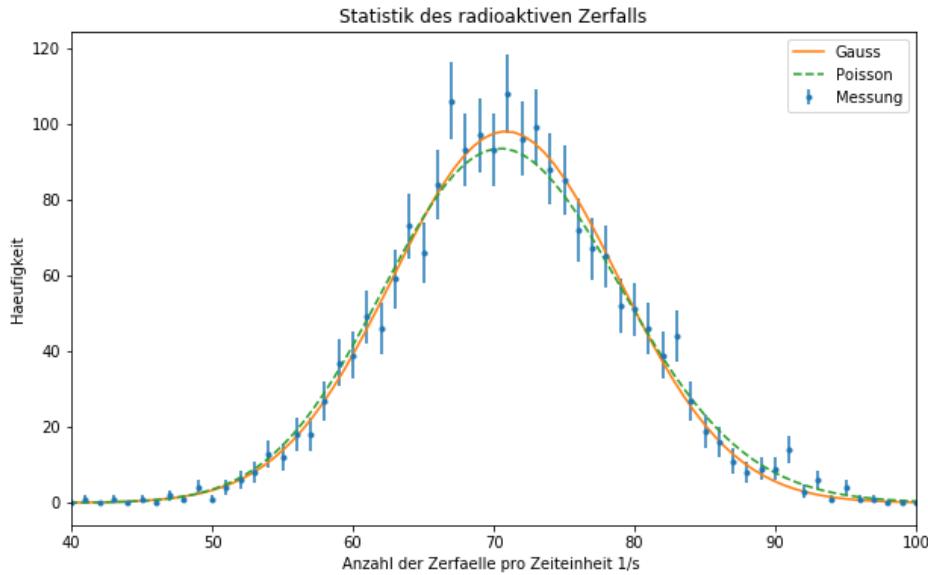


Abbildung 10: Verteilung der Zählraten bei hoher mittlerer Ereigniszahl mit Gauß- und Poisson-Fit

Parameter	Optimierter Wert	
	Gauß	Poisson
μ	70.86 ± 0.20	70.96 ± 0.21
σ	8.00 ± 0.18	-
A	1965 ± 50	1972 ± 50

Tabelle 4: Optimierte Werte Gauß- und Poisson-Fit bei hoher mittlerer Ereigniszahl

Zur Beurteilung der Güte des Fits möchten wir nun noch die χ^2 -Summe berechnen. Hierfür gilt die Formel

$$\chi^2 = \sum_i^N \left(\frac{f(b_i) - h_i}{\Delta h_i} \right)^2 \quad (27)$$

mit der Anzahl der Messungen N , der optimierten Funktion f , der Klasse b_i , sowie der gemessenen Häufigkeit h_i und Fehler Δh_i für die Klasse b_i des Histogramms. Die reduzierte χ^2 -Summe berechnen wir nach $\chi_{\text{red}}^2 = \chi^2/\text{dof}$. Den Freiheitsgrad dof bestimmen wir aus der Differenz der Anzahl der Messwerte und der Zahl der Fitparameter.

Für den hier durchgeführten Fit der Gaußverteilung erhalten wir die folgenden Werte

$$\chi^2 = 18.5, \quad \chi_{\text{red}}^2 = 0.5, \quad (28)$$

für den Fit der Poisson-Verteilung die Werte

$$\chi^2 = 24.6, \quad \chi_{\text{red}}^2 = 0.7. \quad (29)$$

Betrachten wir die Fläche unter der χ^2 -Verteilung im Intervall $[\chi_2, \infty]$, so erhalten wir die Wahrscheinlichkeit dafür, dass wir bei einer Wiederholungsmessung einen gleich großen

oder größeren χ^2 -Wert erhalten. Die Fitwahrscheinlichkeiten im betrachteten Fall sind

$$\mathbb{P}(\chi^2 \geq 18.5) = 99.0\% \quad (30)$$

für den Gauß-Fit und

$$\mathbb{P}(\chi^2 \geq 24.6) = 90.0\% \quad (31)$$

für den Poisson-Fit.

Die Resultate werden im entsprechenden, abschließenden Kapitel diskutiert werden. Nun betrachten wir erst noch die Daten bei kleiner Ereigniszahl.

3.3 Statistik bei kleiner Ereigniszahl

Für Intervalle von 100ms wurden in diesem Versuchsteil 5000 Messungen vom Computer durchgeführt. Die Daten sind im Histogramm 11 zu sehen.

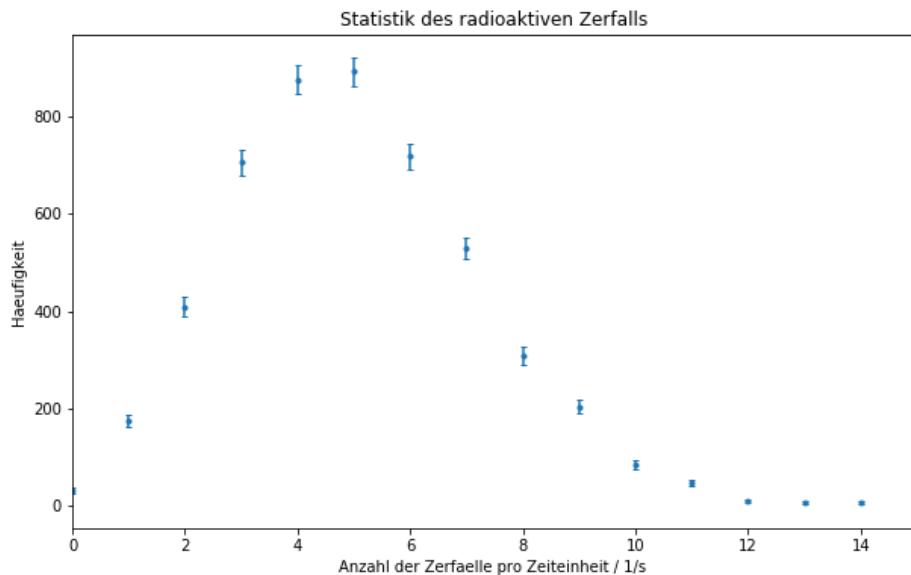


Abbildung 11: Verteilung der Zählraten bei kleiner Ereigniszahl

Bereits hier ist zu erkennen, dass sich die Form der Verteilung eher der einer Poisson-Verteilung annähert. Die Auswertung dieses Versuchsteils läuft analog zur Vorherigen ab. Mit den Startparametern in Tabelle 5 führen wir Fits für eine Gauß- und eine Poisson-Verteilungsfunktion durch.

Parameter	Wert
μ	5.01
σ	2.25
A	5000

Tabelle 5: Startparameter für Gauß- und Poisson-Fit bei kleiner Ereigniszahl

Die Resultate der Fits sind in Abbildung (12) grafisch dargestellt, die Werte der Parameter können Tabelle 6 entnommen werden. Es ist zu beachten, dass wir aus Darstellungsgründen im Plot eine logarithmische Skala für die y -Achse verwenden.

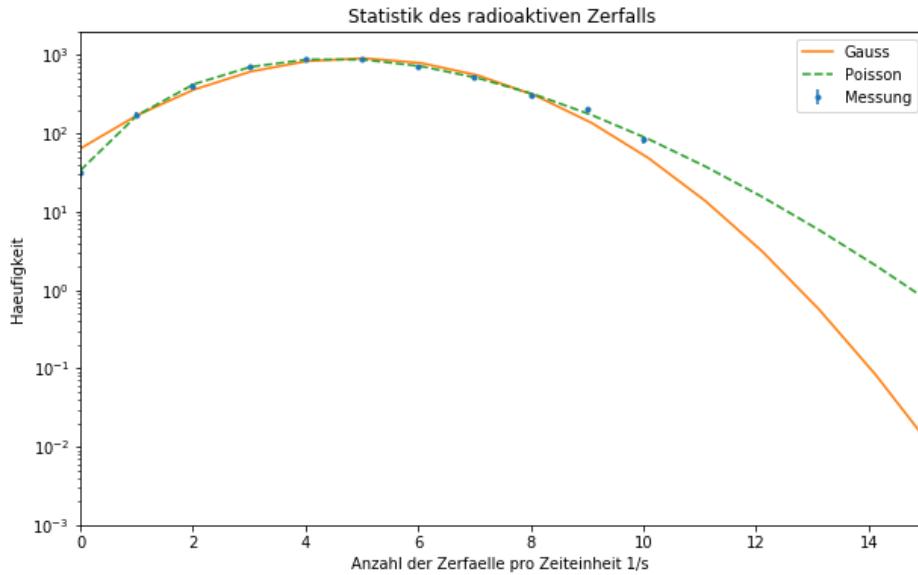


Abbildung 12: Verteilung der Zählraten bei kleiner Ereigniszahl mit Gauß- und Poisson-Fit

Parameter	Optimierter Wert	
	Gauß	Poisson
μ	4.92 ± 0.04	5.00 ± 0.04
σ	2.134 ± 0.026	-
A	4880 ± 71	4992 ± 72

Tabelle 6: Optimierte Werte Gauß- und Poisson-Fit bei kleiner Ereigniszahl

Wie zuvor berechnen wir nun auch noch die χ^2 -Werte, sowie die Fitwahrscheinlichkeiten. Für den Gauß-Fit gilt

$$\chi^2 = 94.3, \quad \chi^2_{\text{red}} = 11.9, \quad \mathbb{P}(\chi^2 \geq 94.3) = 0.0\% \quad (32)$$

und für den Poisson-Fit

$$\chi^2 = 5.14, \quad \chi^2_{\text{red}} = 0.57, \quad \mathbb{P}(\chi^2 \geq 94.3) = 82.0\%. \quad (33)$$

Auch diese Ergebnisse werden wir im folgenden Kapitel diskutieren.

4 Zusammenfassung und Diskussion

In Versuch 251 untersuchten wir die statistische Natur des radioaktiven Zerfalls. Hierzu nahmen wir Zerfälle eines ^{60}Co Präparates mit einem Geiger-Müller-Zählrohr auf. Der Versuch begann mit der Kalibrierung des Geiger-Müller-Zählrohrs. Hierbei ermittelten wir die Zählrohrspannung U_0 , welche wir für den weiteren Betrieb des Zählrohrs über den Versuch verwendeten. Außerdem führten wir einige Messungen zur Beurteilung des Verhaltens der Zählraten im Plateaubereich des Zählrohrs durch. Hierzu betrachteten wir die Zahl registrierter Zerfälle einmal für die Zählrohrspannung U_0 , sowie $U_0 + 100\text{V}$ jeweils über eine Minute und drei Minuten.

Mit einem prozentualen Anstieg von unter 2% für beide betrachteten Zeiträume waren die Plateauanstiege, wie erwartet sehr gering. Für die Messung über eine Minute lag die Abweichung bei ca. 1.12σ , für die Messung über drei Minuten bei etwa 2.22σ . Beide Werte sind somit innerhalb des 3σ -Bereichs und können als nicht signifikant gewertet werden.

Um eine Genauigkeit von 1% zu erreichen, müssten für nach den Daten der Messung über 1min ca. 5.5d und nach den Daten über 3min ca. 4.23d Tage messen. Der relative Fehler bei unseren Messzeiten entspricht nach σ_Δ/Δ für 1min 89.02% bzw. 45.06% für 3min. Wir sehen zum einen, dass dieser bereits bei einer Verlängerung der Messung um 2min stark absinkt. Allerdings sorgt die $\frac{1}{\sqrt{t}}$ -Abhängigkeit von der Messzeit dafür, dass wir uns bei weiterer Verlängerung der Messzeit nur sehr langsam asymptotisch der 0 annähern. Dies erklärt die sehr hohen Werte im Bereich mehrerer Tage. Dass sich die beiden berechneten Zeiten unterscheiden kann daran liegen, dass die Werte der Messung über 3min bereits eine höhere Genauigkeit haben, von welcher aus es weniger lange dauern würde, die Genauigkeit von 1% über eine längere Messung zu erreichen.

Ließen wir für die Zählraten Werte in einem Vertrauensbereich von 68% (95%) zu, so würden wir eine prozentuale Veränderung dieser von 3.54% (5.57%) über 1min bzw. 2.94% (4.11%) über 3min erhalten. Obwohl wir hiermit bereits sehr große Fehlerbereiche ausschöpfen, ist der Plateauanstieg mit einem Maximum von gerade einmal 5.57% immer noch sehr gering. Dies bestätigt noch einmal die Genauigkeit im Plateaubereich des Zählrohrs, welche für die nun folgenden Versuchsteile benötigt wird.

Nachfolgend betrachteten wir einmal das statistische Verhalten bei einer hohen mittleren Ereigniszahl, für 2000 Messungen in Intervallen von 500ms, und einmal bei kleiner mittlerer Ereigniszahl, für 5000 Messungen in Intervallen von 100ms. An die aufgenommenen Daten passten wir jeweils eine Gauß- und eine Poissonverteilungsfunktion an und berechneten die χ^2 - und reduzierte χ^2 -Summe, sowie die Fitwahrscheinlichkeiten zur Beurteilung der Güte der Fits.

Für den Fit an die Daten der hohen mittleren Ereigniszahl erhielten wir die Werte in Tabelle 7.

Variable	Wert	
	Gauß	Poisson
χ^2	18.5	24.6
χ^2_{red}	0.5	0.7
\mathbb{P}_{Fit}	99.0%	90.0%

Tabelle 7: Güte der Fits an Daten für hohe mittlere Ereigniszahl

Da die χ^2 -Summe die Abweichung des Modells von den gemessenen Daten angibt, ist es ein Ziel, diese möglichst gering zu halten. Dies ist für beide Fit-Funktionen der Fall. Allerdings kann ein zu kleiner χ^2 -Wert darauf hinweisen, Unsicherheiten überschätzt zu haben oder, dass das Modell überangepasst ist. Der reduzierte χ^2 geht optimalerweise gegen 1. Dies deutet darauf hin, dass die Poisson-Verteilung in diesem Fall die richtige Wahl sein könnte. Auch in Anbetracht der nahezu 100%-igen Fitwahrscheinlichkeit für die Gaußverteilung, könnte die Vermutung auftreten, dass hier eine unrealistische Überanpassung der Fall ist. Dahingegen spricht die 90.0%-ige Fitwahrscheinlichkeit der Poissonverteilung erneut eher für die Verwendung dieses Modells. Allgemein ist die Wahl des Modells hier nicht ganz eindeutig möglich, da sich die Wahrscheinlichkeitsverteilung, wie im Theorienteil erklärt, in diesem Fall sowohl durch eine Poissonverteilung, als auch, angenähert, durch eine Gaußverteilung interpretieren lässt.

Nun Betrachten wir noch die Resultate für die Fits an die Daten bei kleiner mittlerer Ereigniszahl, in Tabelle 8.

Variable	Wert	
	Gauß	Poisson
χ^2	94.3	5.14
χ^2_{red}	11.9	0.57
\mathbb{P}_{Fit}	0.0%	82.0%

Tabelle 8: Güte der Fits an Daten für kleine Ereigniszahl

Die Beurteilung der Werte ist in diesem Beispiel etwas deutlicher. Wir sehen einen sehr großen Wert für die χ^2 -Summe im Fall der Gaußverteilung, sowie einen Wert von χ^2_{red} weit über 1, was auf einen schlechten Fit hinweist. Auch die Fitwahrscheinlichkeiten von 0% deutet darauf hin, dass dieses Modell die Daten quasi gar nicht beschreiben kann. Die entsprechenden Werte für die Poissonverteilung bewegen sich in einem sehr guten Bereich. Mit 0.57 ist der Wert von χ^2_{red} zwar unter 1, hier aber in einem akzeptablen Bereich. Die Fitwahrscheinlichkeiten von 89% weist darauf hin, dass das Modell die Daten gut beschreiben kann. Dies ist, wie im Theorienteil eingeführt, für eine kleine mittlere Ereigniszahl auch so zu erwarten.

Python Code, Hauptprogramm

Versuch_251-RG_MP

December 12, 2024

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

      from scipy.optimize import curve_fit
      from scipy.special import gamma
      from scipy.stats import chi2
```

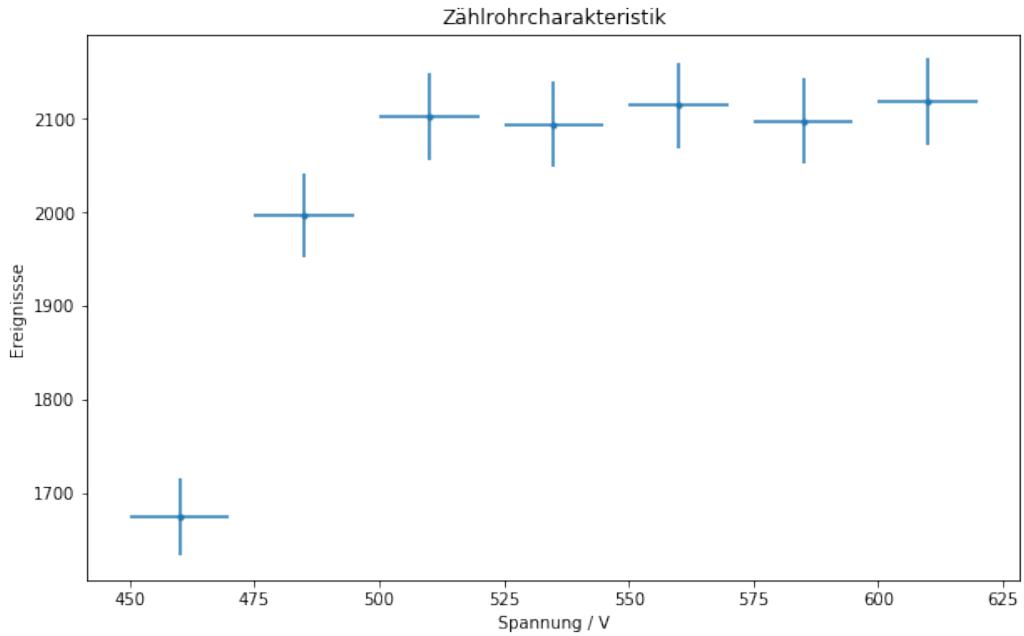
```
[2]: # Aufgabe 2

#Zählrohrspannung:
U=np.arange(460, 615, 25)
print(U)
dU=10#Ablesefehler der Spannung

#gezählte Ereignisse
N=np.array([1674, 1996, 2102, 2094, 2114, 2097, 2118])
assert(len(U) == len(N))
Fehler_N=np.sqrt(N)
print(Fehler_N)
```

```
[460 485 510 535 560 585 610]
[40.91454509 44.67661581 45.84757355 45.76024475 45.97825573 45.79301257
 46.021734 ]
```

```
[3]: plt.figure(figsize=(10,6))
plt.errorbar(U, N, yerr=Fehler_N,xerr=dU, fmt=".")
plt.xlabel('Spannung / V')
plt.ylabel('Ereignisse')
plt.title('Zählrohrcharakteristik')
plt.savefig("bestimmungU0.png", format="png")
```



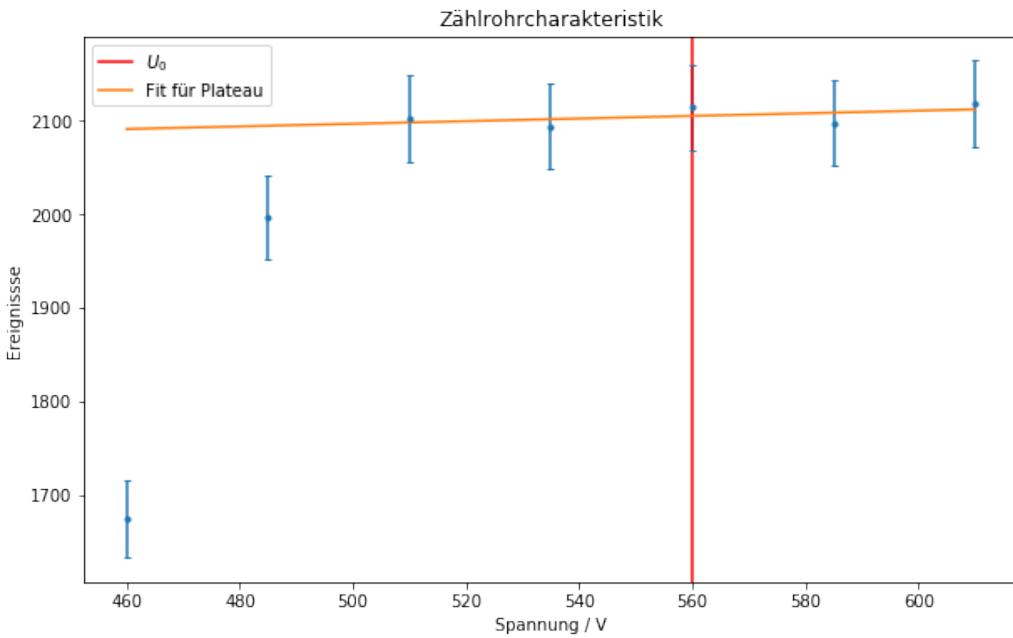
```
[4]: def linear(x,a,b):
    return a*x+b

popt, pcov = curve_fit(linear, U[2:], N[2:])
```

```
[5]: U0 = np.mean(U[[2,-1]])
U0
```

```
[5]: 560.0
```

```
[6]: plt.figure(figsize=(10,6))
plt.axvline(x=U0, color="red", label="$U_0$")
plt.errorbar(U, N, Fehler_N, fmt=". ", capsize=2)
plt.xlabel('Spannung / V')
plt.ylabel('Ereignisse')
plt.title('Zählrohrcharakteristik')
plt.plot(U[0:], linear(U[0:],*popt), label="Fit für Plateau")
plt.legend()
plt.savefig("bestimmungU0_fit_mean.png", format="png")
```



```
[7]: # Aufgabe 3
# U_0 = 560
N_1 = np.array([10566, 10730]) # U0 , U0 +100, t=1min
N_3 = np.array([31199, 31756]) # U0 , U0 +100, t=3min
err_N_1 = np.sqrt(N_1)
err_N_3 = np.sqrt(N_3)
A_1 = N_1[1]-N_1[0]
err_A_1 = np.sqrt(N_1[1]+N_1[0])
A_3 = N_3[1]-N_3[0]
err_A_3 = np.sqrt(N_3[1]+N_3[0])

print(err_N_1, err_N_3)

print("1min: slope =", A_1 , "+-", np.round(err_A_1))
print("3min: slope =", A_3 , "+-", np.round(err_A_3))
print("sigma 1min: ", np.round(A_1/err_A_1,2))
print("sigma 3min: ", np.round(A_3/err_A_3,2))
```

```
[102.79105019 103.5857133 ] [176.63238661 178.20213242]
1min: slope = 164 +- 146.0
3min: slope = 557 +- 251.0
sigma 1min:  1.12
sigma 3min:  2.22
```

```
[8]: print(f"Messung 1 Minute:\n3sigma-Bereich: [{-3 * err_A_1}, {3 * err_A_1}],\u2194{A_1} entspricht {A_1 / err_A_1} sigma => Nicht signifikant")
print(f"Messung 3 Minuten:\n3sigma-Bereich: [{-3 * err_A_3}, {3 * err_A_3}],\u2194{A_3} entspricht {A_3 / err_A_3} sigma => Nicht signifikant")
```

Messung 1 Minute:
3sigma-Bereich: [-437.7944723269128, 437.7944723269128], 164 entspricht
1.123815011608028 sigma => Nicht signifikant
Messung 3 Minuten:
3sigma-Bereich: [-752.7250494038311, 752.7250494038311], 557 entspricht
2.2199340932302647 sigma => Nicht signifikant

```
[9]: ratio_1 = (N_1[1]-N_1[0])/N_1[0]
err_ratio_1 = ratio_1 * np.sqrt( (err_N_1[0]/N_1[0])**2 + (err_A_1/A_1)**2)
ratio_3 = (N_3[1]-N_3[0])/N_3[0]
err_ratio_3 = ratio_3 * np.sqrt( (err_N_3[0]/N_3[0])**2 + (err_A_3/A_3)**2)
print("1min Anstieg =", np.round(ratio_1*100,2), "+-", np.
      round(err_ratio_1*100, 2), "%")
print("3min Anstieg =", np.round(ratio_3*100,2), "+-", np.
      round(err_ratio_3*100, 2), "%")
```

1min Anstieg = 1.55 +- 1.38 %
3min Anstieg = 1.79 +- 0.8 %

```
[10]: # Aufgabe (b)

#  $t_{\text{1}} \approx (\frac{\sigma_{\Delta}}{\Delta} \frac{\sqrt{t_{\text{mess}}}}{1} \%) \approx$ 
t1p1 = (((err_A_1 / A_1)*(np.sqrt(1) / 0.01))**2)
t1p3 = (((err_A_3 / A_3)*(np.sqrt(3) / 0.01))**2)

print(t1p1, t1p1/(60*24), t1p3, t1p3/(60*24))
```

7917.906008328377 5.498545839116928 6087.529693891035 4.2274511763132185

```
[11]: # Aufgabe = (c)

ratio_1_1sig = (N_1[1] + 1 * err_N_1[1] - (N_1[0] - 1 * err_N_1[0]))/(N_1[0] - 1 * err_N_1[0])

ratio_1_2sig = (N_1[1] + 2 * err_N_1[1] - (N_1[0] - 2 * err_N_1[0]))/(N_1[0] - 2 * err_N_1[0])

print(ratio_1_1sig * 100, ratio_1_2sig * 100)

ratio_3_1sig = (N_3[1] + 1 * err_N_3[1] - (N_3[0] - 1 * err_N_3[0]))/(N_3[0] - 1 * err_N_3[0])
```

```

ratio_3_2sig = (N_3[1] + 2 * err_N_3[1] - (N_3[0] - 2 * err_N_3[0]))/(N_3[0] - 2 * err_N_3[0])

print(ratio_3_1sig * 100, ratio_3_2sig * 100)

```

3.539800889683748 5.566894430107933
2.939280877583519 4.106464082491329

Aufgabe 4

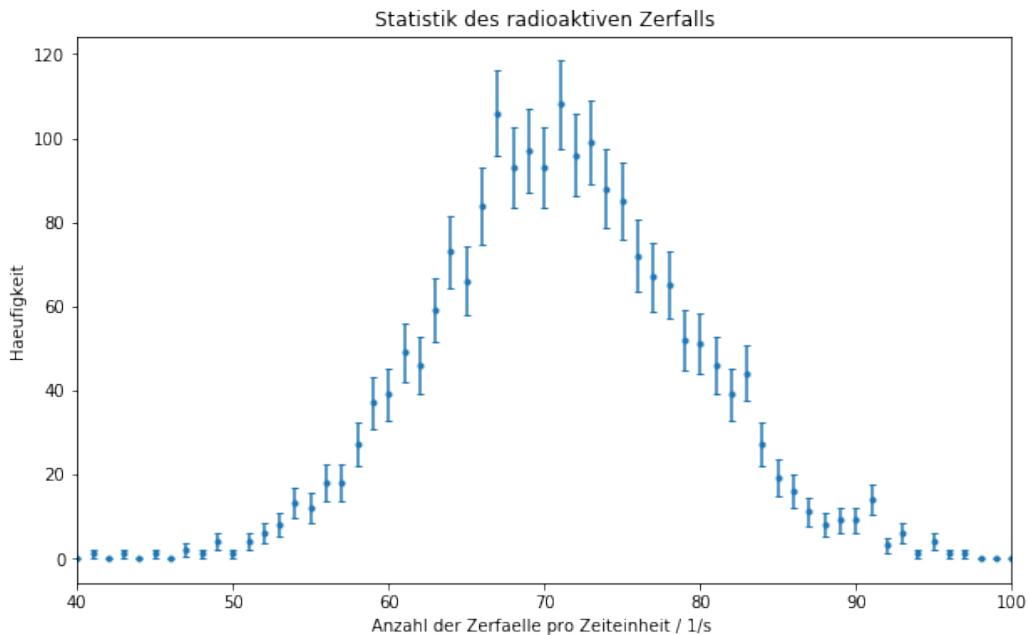
[12]: # Aufgabe 4

```

anzahl, haeufigkeit=np.loadtxt("data_aufgabe2.txt", unpack=True,
                               delimiter = ",",
                               skiprows = 4, usecols = (0,1))
fehler=np.sqrt(haeufigkeit)

```

[13]: plt.figure(figsize=(10,6))
plt.errorbar(anzahl, haeufigkeit, fehler, fmt=".",
 capsize=2)
plt.xlim((40, 100))
plt.xlabel('Anzahl der Zerfaelle pro Zeiteinheit / 1/s')
plt.ylabel('Haeufigkeit')
plt.title('Statistik des radioaktiven Zerfalls')
plt.savefig("aufgabe4_data.png", format="png")



```
[14]: def gaussian(x, A, mu, sig): #A: Flaeche der Gaussfunktion
       return A/(np.sqrt(2 * np.pi)*sig)*np.exp(-(x-mu)**2 / (2 * sig**2))
```

```
[15]: zerf_gr10 = np.where(haeufigkeit > 10)
      zerf_gr10_min = 21
      zerf_gr10_max = 58

      zerf_gr10
```

```
[15]: (array([21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,
       38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
       58]),)
```

```
[16]: popt_g, pcov_g=curve_fit(gaussian, anzahl[zerf_gr10_min:zerf_gr10_max], ↴
      ↴haeufigkeit[zerf_gr10_min:zerf_gr10_max], p0 = [2000, 71.05, 8.23], ↴
      ↴sigma=fehler[zerf_gr10_min:zerf_gr10_max], absolute_sigma = True)
      # print(pcov_g)

      print(f"A = {popt_g[0]:.2f} +- {np.sqrt(pcov_g[0][0]):.2f}")
      print(f"mu = {popt_g[1]:.2f} +- {np.sqrt(pcov_g[1][1]):.2f}")
      print(f"sig = {popt_g[2]:.2f} +- {np.sqrt(pcov_g[2][2]):.2f}")
```

A = 1965.84 +- 45.20
 mu = 70.86 +- 0.20
 sig = 8.00 +- 0.17

```
[17]: print(fehler[zerf_gr10_min:zerf_gr10_max])
```

```
[ 3.60555128  3.46410162  4.24264069  4.24264069  5.19615242  6.08276253
  6.244998     7.          6.78232998  7.68114575  8.54400375  8.1240384
  9.16515139 10.29563014  9.64365076  9.8488578   9.64365076 10.39230485
  9.79795897  9.94987437  9.38083152  9.21954446  8.48528137  8.18535277
  8.06225775  7.21110255  7.14142843  6.78232998  6.244998   6.63324958
  5.19615242  4.35889894  4.          3.31662479  2.82842712  3.
  3.          ]]
```

```
[18]: def poisson(x, A_p, mu_p):
       return A_p*np.exp(-mu_p)*mu_p**x/gamma(x+1)

      popt_p, pcov_p = curve_fit(poisson, anzahl[zerf_gr10_min:zerf_gr10_max], ↴
      ↴haeufigkeit[zerf_gr10_min:zerf_gr10_max], p0=[2000, 71.05], sigma = ↴
      ↴fehler[zerf_gr10_min:zerf_gr10_max], absolute_sigma=True)
      #print(pcov_g)

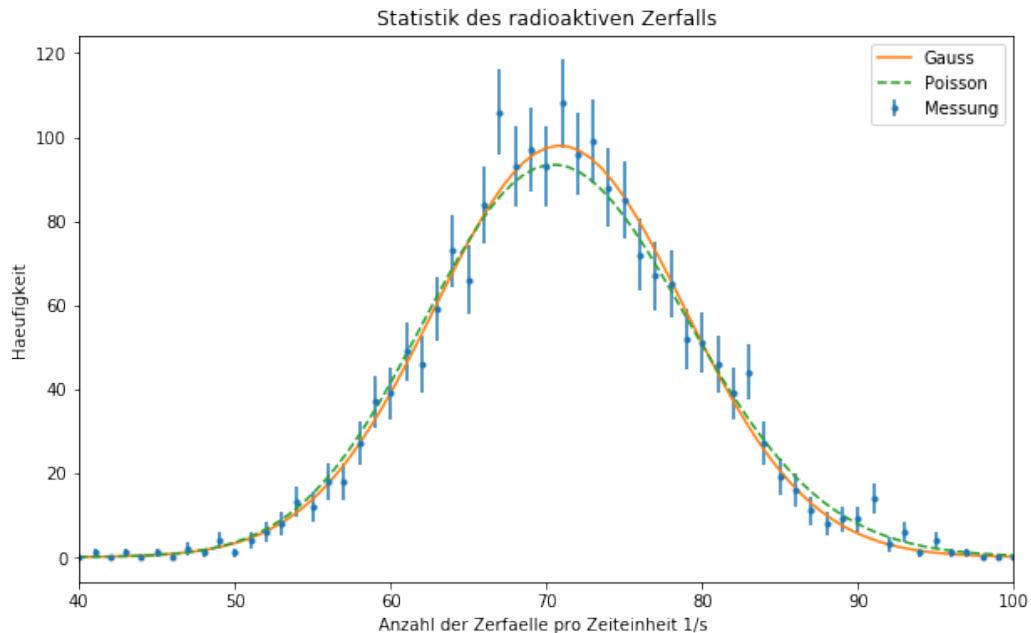
      print(f"A = {popt_p[0]:.2f} +- {np.sqrt(pcov_p[0][0]):.2f}")
      print(f"mu = {popt_p[1]:.2f} +- {np.sqrt(pcov_p[1][1]):.2f}")
```

A = 1972.29 +- 45.05

```
mu = 70.96 +- 0.20
```

```
[19]: plt.figure(figsize=(10,6))
plt.errorbar(anzahl,haeufigkeit,fehler, fmt=". ",label='Messung')
plt.xlabel('Anzahl der Zerfaelle pro Zeiteinheit 1/s ')
plt.ylabel('Haeufigkeit')
plt.title('Statistik des radioaktiven Zerfalls')
x=np.linspace(min(anzahl), max(anzahl), int(max(anzahl)))

plt.plot(x, gaussian(x,*popt_g), label='Gauss')
plt.plot(x, poisson(x,*popt_p), label='Poisson',linestyle='dashed')
plt.legend()
plt.xlim((40, 100))
# plt.yscale('log')
plt.savefig("aufgabe4_gauss_poisson_fit.png", format="png")
```



```
[20]: print("Gaussfit:")
print("A=", popt_g[0], ", Standardfehler=", np.sqrt(pcov_g[0][0]))
print("mu=", popt_g[1], ", Standardfehler=", np.sqrt(pcov_g[1][1]))
print("sig=", popt_g[2], ", Standardfehler=", np.sqrt(pcov_g[2][2]))
print("Poissonfit:")
print("A=", popt_p[0], ", Standardfehler=", np.sqrt(pcov_p[0][0]))
print("mu=", popt_p[1], ", Standardfehler=", np.sqrt(pcov_p[1][1]))
```

Gaussfit:

```

A= 1965.8373349113892 , Standardfehler= 45.20203874142463
mu= 70.85823505791575 , Standardfehler= 0.19655842820679315
sig= 8.004791336335561 , Standardfehler= 0.17160470029907776
Poissonfit:
A= 1972.288272915696 , Standardfehler= 45.05407969780351
mu= 70.96489066870816 , Standardfehler= 0.20287774389457452

```

```
[21]: #Gauss:
x1 = ((gaussian(anzahl[zerf_gr10_min:
    ↪zerf_gr10_max],*popt_g)-haeufigkeit[zerf_gr10_min:zerf_gr10_max])**2)
x2 = fehler[zerf_gr10_min:zerf_gr10_max]**2
chi2_div = x1/x2
chi2_g= np.sum(chi2_div)
dof_g=len(anzahl[zerf_gr10_min:zerf_gr10_max])-3 #dof: degrees of freedom, ↪Freiheitsgrad
chi2_red_g=chi2_g/dof_g
print("chi2_g=", chi2_g)
print("chi2_red_g=",chi2_red_g)
```

```

chi2_g= 18.49944005335566
chi2_red_g= 0.5441011780398723

```

```
[22]: #Poisson:
x11 = (poisson(anzahl[zerf_gr10_min:
    ↪zerf_gr10_max],*popt_p)-haeufigkeit[zerf_gr10_min:zerf_gr10_max])**2
x22 = fehler[zerf_gr10_min:zerf_gr10_max]**2
chi2_div = x11/x22
chi2_p = np.sum(chi2_div)
dof_p = len(anzahl[zerf_gr10_min:zerf_gr10_max])-2 #poisson hat nur 2 Parameter
chi2_red_p=chi2_p/dof_p
print("chi2_p=", chi2_p)
print("chi2_red_p=",chi2_red_p)
```

```

chi2_p= 24.617953649112792
chi2_red_p= 0.7033701042603655

```

```
[23]: prob_g = round(1-chi2.cdf(chi2_g, dof_g), 2)*100
prob_p = round(1-chi2.cdf(chi2_p, dof_p), 2)*100
print("Wahrscheinlichkeit Gauss=", prob_g, "%")
print("Wahrscheinlichkeit Poisson=", prob_p, "%")
```

```

Wahrscheinlichkeit Gauss= 99.0 %
Wahrscheinlichkeit Poisson= 90.0 %

```

```
[24]: # Aufgabe 5
anzahl, haeufigkeit=np.loadtxt("data_aufgabe3.txt", unpack=True,
                                delimiter = ", ", skiprows = 4, usecols = (0,1) )
```

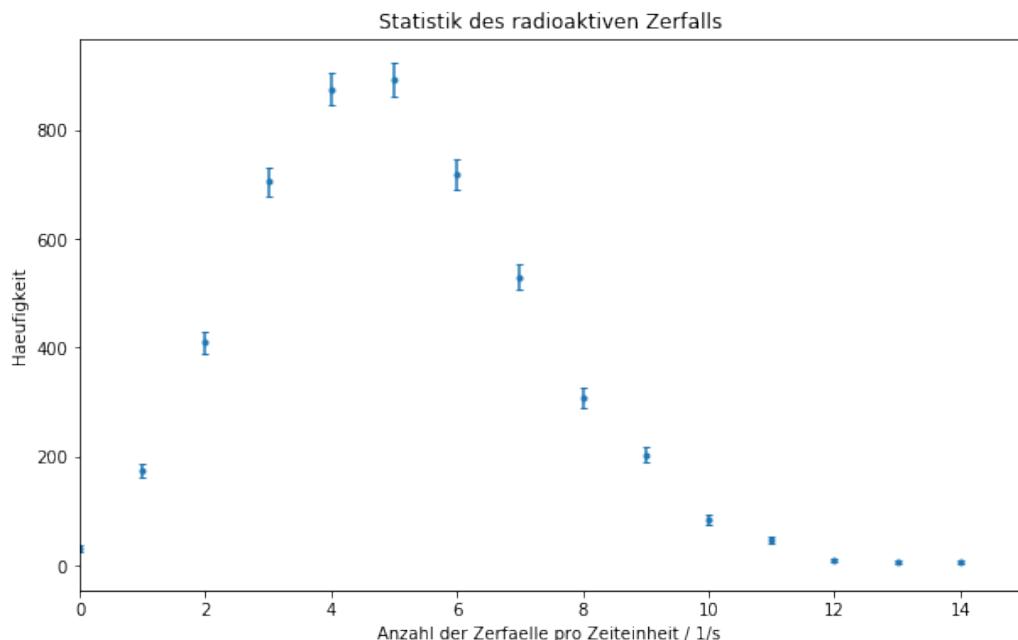
```

fehler=np.sqrt(haeufigkeit)

[25]: plt.figure(figsize=(10,6))
plt.errorbar(anzahl, haeufigkeit, fehler, fmt=". ", capsize=2)
plt.xlim((0, 15))

plt.xlabel('Anzahl der Zerfaelle pro Zeiteinheit / 1/s ')
plt.ylabel('Haeufigkeit')
plt.title('Statistik des radioaktiven Zerfalls')
plt.savefig("aufgabe5_data.png", format="png")

```



```

[26]: zerf_5_gr10 = np.where(haeufigkeit > 10)
zerf_5_gr10_min = 0
zerf_5_gr10_max = 11

```

```

[27]: popt_g, pcov_g=curve_fit(gaussian, anzahl[zerf_5_gr10_min:zerf_5_gr10_max], haeufigkeit[zerf_5_gr10_min:zerf_5_gr10_max], p0 = [5000, 5.01, 2.25], sigma=fehler[zerf_5_gr10_min:zerf_5_gr10_max], absolute_sigma = True)
print(pcov_g)

```

```

[[4.94368575e+03 1.57823182e-02 1.04571080e-01]
 [1.57823182e-02 1.04433066e-03 1.34655224e-04]
 [1.04571080e-01 1.34655224e-04 6.34671949e-04]]

```

```
[28]: print(fehler[zerf_5_gr10_min:zerf_5_gr10_max])

[ 5.65685425 13.19090596 20.22374842 26.55183609 29.58039892 29.86636905
 26.79552201 23.          17.54992877 14.24780685  9.16515139]

[29]: def poisson(x, A_p, mu_p):
    return A_p*np.exp(-mu_p)*mu_p**x/gamma(x+1)

popt_p, pcov_p = curve_fit(poisson, anzahl[zerf_5_gr10_min:zerf_5_gr10_max], ↴
    ↪haeufigkeit[zerf_5_gr10_min:zerf_5_gr10_max], p0=[5000, 5.01], sigma = ↴
    ↪fehler[zerf_5_gr10_min:zerf_5_gr10_max], absolute_sigma=True)
print(pcov_g)

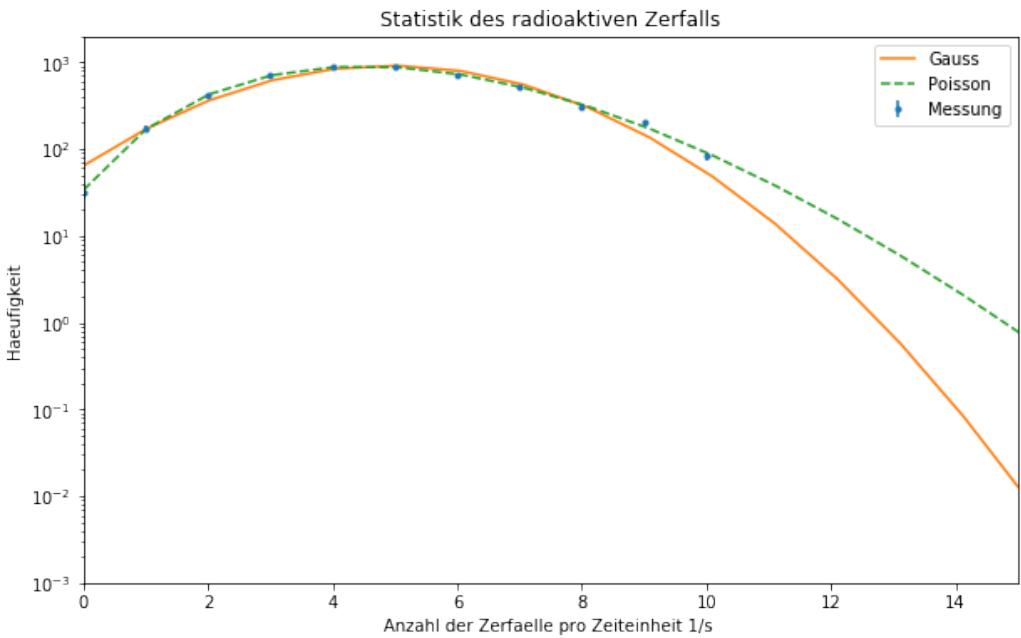
[[4.94368575e+03 1.57823182e-02 1.04571080e-01]
 [1.57823182e-02 1.04433066e-03 1.34655224e-04]
 [1.04571080e-01 1.34655224e-04 6.34671949e-04]]
```

```
[30]: fig = plt.figure(figsize=(10,6))
plt.errorbar(anzahl[zerf_5_gr10_min:
    ↪zerf_5_gr10_max], haeufigkeit[zerf_5_gr10_min:
    ↪zerf_5_gr10_max], fehler[zerf_5_gr10_min:zerf_5_gr10_max], fmt=".",
    ↪", label='Messung')
plt.xlabel('Anzahl der Zerfaelle pro Zeiteinheit 1/s ')
plt.ylabel('Haeufigkeit')
plt.title('Statistik des radioaktiven Zerfalls')
x=np.linspace(min(anzahl), max(anzahl), int(max(anzahl)))

plt.plot(x, gaussian(x,*popt_g), label='Gauss')
plt.plot(x, poisson(x,*popt_p), label='Poisson', linestyle='dashed')
plt.legend()

plt.yscale('log')

plt.semilogy()
plt.xlim((0,15))
plt.ylim((1e-3,2e3))
plt.savefig("aufgabe5_gauss_poisson_fit.png", format="png")
```



```
[31]: print("Gaussfit:")
print("A=", popt_g[0], ", Standardfehler=", np.sqrt(pcov_g[0][0]))
print("mu=", popt_g[1], ", Standardfehler=", np.sqrt(pcov_g[1][1]))
print("sig=", popt_g[2], ", Standardfehler=", np.sqrt(pcov_g[2][2]))
print("Poissonfit:")
print("A=", popt_p[0], ", Standardfehler=", np.sqrt(pcov_p[0][0]))
print("mu=", popt_p[1], ", Standardfehler=", np.sqrt(pcov_p[1][1]))
```

Gaussfit:
A= 4880.366788352602 , Standardfehler= 70.31134863922186
mu= 4.919135221373067 , Standardfehler= 0.03231610533346779
sig= 2.134058560699586 , Standardfehler= 0.02519269633950696
Poissonfit:
A= 4992.297029938469 , Standardfehler= 71.21313047216086
mu= 5.000742744202284 , Standardfehler= 0.03377435196371074

```
[32]: #Gauss:
x1 = ((gaussian(anzahl[zerf_5_gr10_min:
    ↪zerf_5_gr10_max],*popt_g)-haeufigkeit[zerf_5_gr10_min:zerf_5_gr10_max])**2)
x2 = fehler[zerf_5_gr10_min:zerf_5_gr10_max]**2
chi2_div = x1/x2
chi2_g= np.sum(chi2_div)
dof_g=len(anzahl[zerf_5_gr10_min:zerf_5_gr10_max])-3 #dof: degrees of freedom, ↪Freiheitsgrad
chi2_red_g=chi2_g/dof_g
```

```
print("chi2_g=", chi2_g)
print("chi2_red_g=", chi2_red_g)
```

```
chi2_g= 94.25083765206813
chi2_red_g= 11.781354706508516
```

```
[33]: #Poisson:
x11 = (poisson(anzahl[zerf_5_gr10_min:
    ↪zerf_5_gr10_max],*popt_p)-haeufigkeit[zerf_5_gr10_min:zerf_5_gr10_max])**2
x22 = fehler[zerf_5_gr10_min:zerf_5_gr10_max]**2
chi2_div = x11/x22
chi2_p = np.sum(chi2_div)
dof_p = len(anzahl[zerf_5_gr10_min:zerf_5_gr10_max])-2 #poisson hat nur 2
    ↪Parameter
chi2_red_p=chi2_p/dof_p
print("chi2_p=", chi2_p)
print("chi2_red_p=", chi2_red_p)
```

```
chi2_p= 5.141080127786679
chi2_red_p= 0.5712311253096309
```

```
[34]: prob_g = round(1-chi2.cdf(chi2_g, dof_g), 2)*100
prob_p = round(1-chi2.cdf(chi2_p, dof_p), 2)*100
print("Wahrscheinlichkeit Gauss=", prob_g, "%")
print("Wahrscheinlichkeit Poisson=", prob_p, "%")
```

```
Wahrscheinlichkeit Gauss= 0.0 %
Wahrscheinlichkeit Poisson= 82.0 %
```

```
[ ]:
```

Python Code, Bibliothek

lib

December 7, 2024

```
[32]: def floatfmt(v, prec, exp):
        return f"{v/10**exp}:0={prec}f}{f'e{exp}' if exp != 0 else ''}"
```

```
[36]: class ValErr:
    val: float = 0
    err: float = 0
    err_set = False

    def __init__(self, val, err=0):
        self.val = val
        if err != 0:
            self.err_set = True
            self.err = err

    def getTuple(self):
        return (self.val, self.err)

    @classmethod
    def fromTuple(self, tup):
        return ValErr(tup[0], tup[1])

    @classmethod
    def fromFit(self, popt, pcov, i):
        return ValErr(popt[i], np.sqrt(pcov[i][i]))

    @classmethod
    def fromFitAll(self, popt, pcov):
        for i in range(0, len(popt)):
            yield ValErr(popt[i], np.sqrt(pcov[i][i]))

    def strfmt(self, prec=2):
        if self.err != 0:
            return f'{self.val:.{prec}e} ± {self.err:.{prec}e}'
        else:
            return f'{self.val:.{prec}e}'

    def strfmtf(self, prec, exp):
```

```

if self.err != 0:
    return fr"{{floatfmt(self.val, prec, exp)} ± {floatfmt(self.err,_
→prec, exp)}}"
else:
    return f"{{floatfmt(self.val, prec, exp)}}"

def strltx(self, prec=2):
    if self.err != 0:
        return fr"{{self.val:.{prec}e} \pm {self.err:.{prec}e}}"
    else:
        return f"{{self.val}}"

def __repr__(self):
    return f"ValErr({self.val}, {self.err})"

def __mul__(self, num):
    return ValErr(self.val * num, self.err * num)

```

[34]: def spacearound(dat, add):
 return np.linspace(dat[0] - add, dat[len(dat)-1] + add)

[35]: def div_with_err(a, a_err, b, b_err):
 err = (1 / b) * np.sqrt(a_err**2 + (a * b_err / b)**2)
 return (a / b, err)

[]: