



# Fortgeschrittenen-Praktikum II:

## Mößbauer-Effekt

Saskia Bondza  
Simon Stephan

durchgeführt vom 20.02.2017 bis 24.02.2017

---

# Übersicht

Mit Hilfe des Mößbauereffekts lässt sich eine sehr genaue und hochauflösende Aufnahme von Energiespektren durchführen. Der Mößbauereffekt beschreibt die rückstoßfreie Absorption und Emission von Photonen und wird in diesem Versuch zur Untersuchung der Emissionslinien des 14,4 keV-Zustands von  $^{57}\text{Fe}$  verwendet. Dabei wird die Lebensdauer des Zustands auf zwei unterschiedliche Arten bestimmt und die Hyperfeinaufspaltung von Natureisen vermessen.

Für die Lebensdauer des 14,4 keV-Zustands von  $^{57}\text{Fe}$  ergeben sich dabei folgende Ergebnisse:

$$\tau_1 = (110 \pm 70) \text{ ns}$$

$$\tau_2 = (162 \pm 12) \text{ ns}$$

$$\tau_{\text{lit}} = 141 \text{ ns.}$$

Aus der Hyperfeinaufspaltung des Natureisens wird das magnetische Moment des Kerns und das Magnetfeld am Kernort bestimmt. Diese ergeben sich zu:

$$\mu_a = (-0,157 \pm 0,004) \mu_N$$

$$\mu_{\text{lit}} = (-0,15531 \pm 0,00004) \mu_N$$

$$B = (32,9 \pm 0,3) \text{ T}$$

$$B_{\text{lit}} = (33,3 \pm 0,1) \text{ T} .$$

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Theoretische Grundlagen</b>	<b>2</b>
2.1	Wechselwirkung von Photonen mit Materie . . . . .	2
2.2	Nachweis der $\gamma$ -Strahlung mithilfe des Szintillationszählers . . . . .	2
2.3	Zerfallsschema von Cobalt . . . . .	3
2.4	Lebensdauer und Linienbreite . . . . .	4
2.4.1	Dopplerverbreiterung . . . . .	4
2.5	Resonanzabsorption . . . . .	5
2.6	Mößbauer-Effekt . . . . .	6
2.7	Gittermodelle . . . . .	7
2.7.1	Einsteinmodell . . . . .	7
2.7.2	Debyemodell . . . . .	8
2.8	Modifikationen des Übergangsspektrums . . . . .	10
2.8.1	Isomerieverschiebung . . . . .	10
2.8.2	Hyperfeinstrukturaufspaltung . . . . .	11
2.9	Geräte und Signalformen . . . . .	12
<b>3</b>	<b>Versuchsaufbau und -Durchführung</b>	<b>14</b>
3.1	Eichung des MCA und Setzen des Energiefensters . . . . .	14
3.2	Aufbau der Schaltung . . . . .	14
3.3	Untergrundmessungen . . . . .	16
3.4	Hauptmessungen . . . . .	16
<b>4</b>	<b>Auswertung</b>	<b>17</b>
4.1	Energieeichung des MCA . . . . .	17
4.2	Berechnung des Compton-Untergrunds . . . . .	19
4.3	Abschwächung durch Plexiglas . . . . .	21
4.4	Umrechnung der Messwerte . . . . .	21
4.5	Einlinienabsorber . . . . .	21
4.5.1	Isomerieverschiebung . . . . .	22
4.5.2	Linienbreite und Lebensdauer . . . . .	22
4.5.3	Debye-Waller-Faktor . . . . .	24
4.6	Sechslinienabsorber . . . . .	25
4.6.1	Hyperfeinaufspaltung . . . . .	27
4.6.2	Isomerieverschiebung . . . . .	27
4.6.3	Magnetisches Moment . . . . .	27
4.6.4	Magnetische Feldstärke . . . . .	28
<b>5</b>	<b>Zusammenfassung und Diskussion</b>	<b>29</b>
5.1	Eichung und Korrekturmessungen . . . . .	29
5.2	Einlinienabsorber . . . . .	29
5.3	Sechslinienabsorber . . . . .	30

<b>6</b>	<b>Anhang</b>	<b>31</b>
6.1	Laborbuch . . . . .	31
6.2	Spektren . . . . .	33
6.3	Sourcecode . . . . .	33
	<b>Literatur</b>	<b>55</b>

# 1 Einleitung

Der Mößbauereffekt beschreibt die rückstoßfreie Resonanzabsorption von Kernzuständen. Da bei diesem Prozess die Energieverluste eliminiert werden, die bisher häufig ein Hindernis darstellten und zugleich eine sehr hohe Auflösung und Genauigkeit erreicht werden kann, bot diese Entdeckung zahlreiche Anwendungen. Ein klassisches Beispiel ist hier die Spektroskopie. Es konnte mit Hilfe des Mößbauereffekts aber auch die Energieverschiebung von Photonen im Gravitationsfeld der Erde  $\frac{\Delta E}{E} = 10^{-16}$  pro Meter nachgewiesen werden, die von der Allgemeinen Relativitätstheorie vorhergesagt wird. Sogenannte MIMOS<sup>1</sup> werden in Marssonden zur Analyse von Marsgestein verwendet. Für die Entdeckung des nach ihm benannten Mößbauereffekts wurde Rudolf Mößbauer im Jahr 1961 der Nobelpreis verliehen.

In diesem Versuch sollen Absorptionslinien im Einlinien-Absorber Edelstahl und im Sechslinien-Absorber Eisen gemessen werden und deren Lebensdauer bestimmt werden. <sup>57</sup>Fe weist eine sehr schmale Linie bei einer Energie von 14,4 keV auf. Zur Messung dieses Übergangs wird der Absorber auf einem beweglichen Schlitten, der sich mit Geschwindigkeiten im Bereich von wenigen  $\frac{\text{mm}}{\text{s}}$  bewegt, von einer <sup>57</sup>Co-Probe bestrahlt, in der durch den Zerfall von Cobalt in einen angeregten Zustand von <sup>57</sup>Fe, der wiederum in den Grundzustand zerfällt, ein Photon der Energie 14,4 keV abgestrahlt wird. Ein Detektor bestehend aus Szintillator und Photomultiplier hinter dem Absorber nimmt die Zählrate in Abhängigkeit der Geschwindigkeit auf. Im so entstehenden Spektrum erwarten wir den typischen Mößbauerpeak zu sehen, aus dem die Lebensdauer des angeregten Zustands bestimmt werden kann, sowie aus der Verschiebung zur theoretischen Resonanzfrequenz die Isomerieverschiebung. Für Natureisen kann aus der Hyperfeinaufspaltung, die Grund für die sechs Absorptionslinien ist, das magnetische Feld bestimmt werden. Außerdem werden verschiedene Messungen zur Bestimmung des Untergrunds durchgeführt. Der in diesem Versuch untersuchte Übergang wurde bereits im Versuch „Kurze Halbwertszeiten“ mit der Methode der verzögerten Koinzidenzen gemessen.

---

<sup>1</sup>MI*niaturized* MO*essbauer* S*pektrometer*

## 2 Theoretische Grundlagen

### 2.1 Wechselwirkung von Photonen mit Materie

Um  $\gamma$ -Strahlung detektieren zu können, muss die  $\gamma$ -Strahlung mit Materie wechselwirken. Wir unterscheiden dabei grundsätzlich drei Prozesse: Den Photoeffekt, den Compton-Effekt und die Paarbildung. In Materie klingt die Intensität von  $\gamma$ -Strahlung exponentiell ab:

$$I = I_0 \cdot e^{-\mu x} \quad \text{mit } \mu = \mu_{\text{photo}} + \mu_{\text{Compton}} + \mu_{\text{Paar}} \quad (1)$$

Die einzelnen Prozesse werden im Folgenden erklärt.[1]

- **Photo-Effekt** Beim Photo-Effekt dringt ein Photon in das Atom ein und überträgt seine gesamte Energie an ein Elektron der inneren Schalen. Dabei wird Energie auf dieses Elektron übertragen, es wird aus der Atomhülle befreit und erhält kinetische Energie. Die hier entstandene Lücke wird über Abstrahlung eines  $\gamma$ -Quants oder eines Elektrons wieder gefüllt.
- **Compton-Effekt** Beim Compton-Effekt trifft ein einfallendes  $\gamma$ -Quant auf ein freies oder nur leicht gebundenes Elektron und überträgt einen Teil seiner Energie auf dieses. Dieser Prozess findet meist bei Energien zwischen 200 keV und 5 MeV statt.
- **Paarbildung und Paarvernichtung** Bei der Paarbildung entsteht durch die Wechselwirkung des  $\gamma$ -Quants mit dem elektromagnetischen Feld des Atomkerns oder eines Elektrons ein Teilchen-Antiteilchen-Paar, z.B. ein Elektronen-Positronen-Paar. Paarbildung ist für Energien über 1,022 MeV möglich. Die über diesen Grenzwert hinausgehende Energie wird auf die entstandenen Teilchen übertragen, der Impuls wird vom Kern aufgenommen. Da das Positron nicht lange alleine existieren kann, vereinigt es sich unter Abstrahlung von zwei  $\gamma$ -Quanten mit einer Energie von je 511 keV mit einem Elektron.

### 2.2 Nachweis der $\gamma$ -Strahlung mithilfe des Szintillationszählers

Die emittierten Photonen müssen nun detektiert und ihre Energie bestimmt werden. Dazu wird ein Energie-sensitiver Detektor, eine Kombination aus Szintillator und Photomultiplier, verwendet.

**Szintillator** Ein Szintillator detektiert Teilchen eines bestimmten Energiebereichs. Es gibt organische und anorganische Szintillatoren, wobei in diesem Versuch ein anorganischer NaI(Tl)-Szintillator verwendet werden. Dieser besteht aus einem mit Thallium dotierten NaI-Kristall, in welchem die eintreffenden Photonen ihre Energie durch den Photo- oder Compton-Effekt an Elektronen abgeben. Je höher die Energie der Photonen ist, desto mehr Elektronen werden erzeugt. Diese Elektronen werden nun angeregt und später unter Emission niederenergetischer Photonen wieder abgeregt. Die Dotierung mit Thallium verhindert, dass die emittierten Elektronen wieder absorbiert werden. Die Bandstruktur, auf der die Funktionsweise eines Szintillators beruht, ist in Abbildung 1 dargestellt.[2]

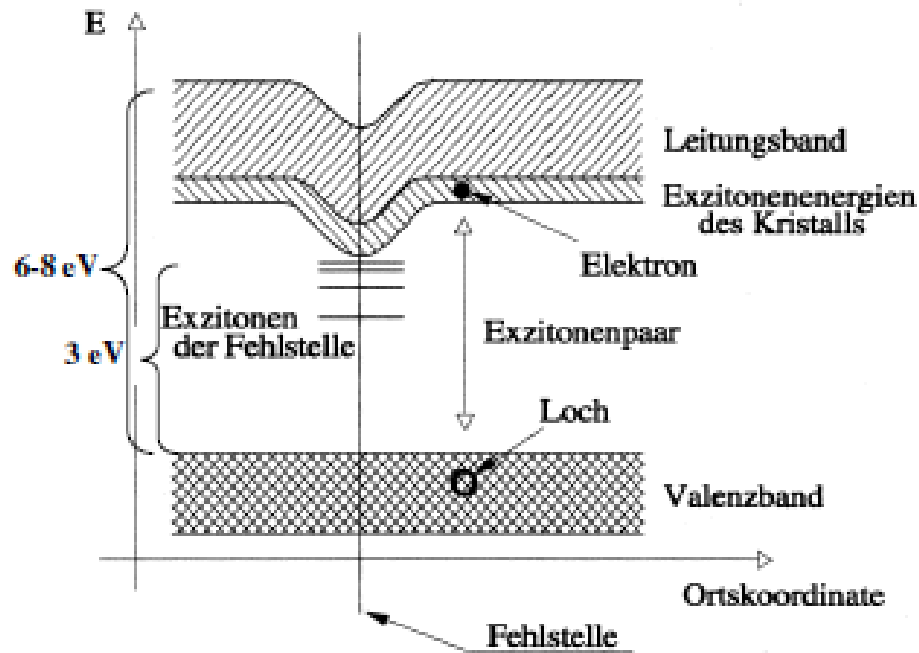


Abbildung 1: Funktionsweise eines Szintillators [2]

**Photomultiplier** Das Licht wird nun vom Szintillator über Lichtleiter zum Photomultiplier geleitet. Dieser wandelt die Lichtimpulse des Szintillators durch den Photoeffekt in elektrische Impulse um, welche proportional zur Lichtintensität sind und verstärkt diese durch Elektronenvervielfachung. Die Funktionsweise eines Photomultipliers ist in Abbildung 2 dargestellt.[3]

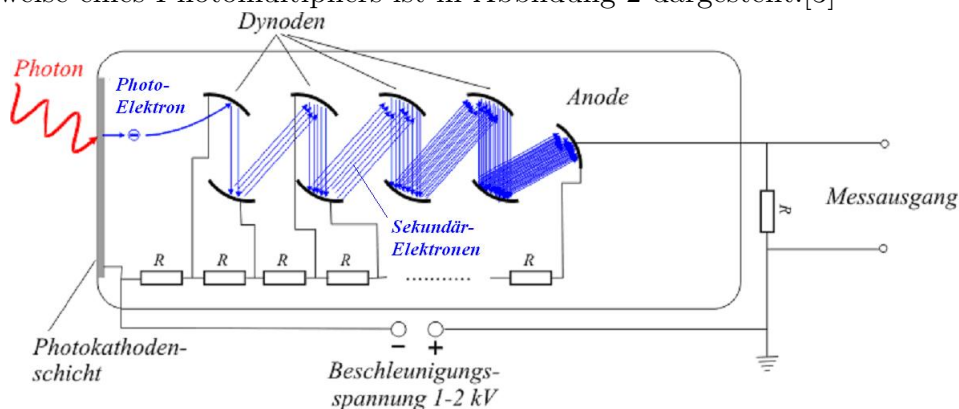
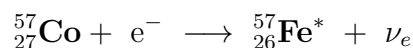
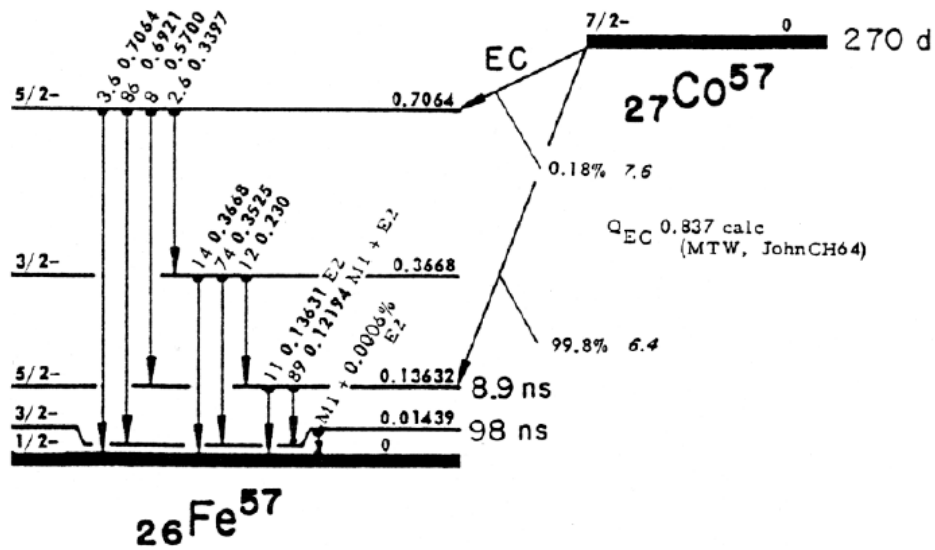


Abbildung 2: Funktionsweise eines Photomultipliers [3]

## 2.3 Zerfallsschema von Cobalt

Das in diesem Versuch verwendete  $^{57}\text{Co}$  zerfällt mit einer Wahrscheinlichkeit von 99,8% und einer Halbwertszeit von 270 Tagen über Elektroneneinfang in einen angeregten Zustand von  $^{57}\text{Fe}^*$  (siehe Abbildung 3):



Abbildung 3: Zerfallsschema von  $^{57}\text{Co}$  [4]

Der angeregte Zustand geht unter anderem über die Aussendung eines Photons mit einer Energie von 14,4 keV mit einer Halbwertszeit von 98 ns in den Grundzustand über. [4]

## 2.4 Lebensdauer und Linienbreite

Die mittlere Lebensdauer ist über den Erwartungswert der Zeit definiert, wie lange ein Zustand existiert:

$$\tau = \langle t \rangle = \int_0^\infty t \lambda e^{-\lambda t} dt = \frac{1}{\lambda}. \quad (2)$$

Grund dafür, dass Spektrallinien eine natürliche Breite haben und nicht als Delta-Distribution auftreten, ist die Heisenbergsche Unschärferelation:

$$\Delta E \cdot \Delta t \geq \frac{\hbar}{2}. \quad (3)$$

Die natürliche Zerfallsbreite weist ein Breit-Wigner Profil auf. Man erhält damit folgenden Zusammenhang zwischen Zerfallsbreite und Lebensdauer:

$$\Gamma = \frac{\hbar}{\tau}.$$

Die Lebensdauer des in diesem Versuch untersuchten 14,4 keV-Übergangs beträgt  $\tau = 141 \cdot \text{ns}$ . Es folgt eine natürliche Zerfallsbreite von  $\Gamma = 4,7 \cdot 10^{-9} \text{ eV}$ . Es handelt sich also um eine sehr scharfe Linie mit einer relativen Breite von  $\frac{\Gamma}{E_\gamma} = 3 \cdot 10^{-13}$ . [5]

### 2.4.1 Dopplerverbreiterung

Aus der thermischen Bewegung der Kerne folgt eine Dopplerverbreiterung der Linienbreite in beide Richtungen, der ein Gauß-Profil zu Grunde liegt. Diese ist wie folgt gegeben:



$$\Gamma_{\text{Dop}} = 2\sqrt{\ln 2} \cdot E_\gamma \cdot \sqrt{\frac{2kT}{Mc^2}} . \quad (4)$$

Diese ist nicht vernachlässigbar und spielt in dem in diesem Versuch zu betrachtenden Prozess eine wesentliche Rolle. [5]

## 2.5 Resonanzabsorption

Resonanzabsorption beschreibt den Prozess, bei dem ein Kern beim Übergang von einem angeregten Zustand  $E_a$  in den Grundzustand  $E_g$  ein Photon aussendet, das dann von einem zweiten Kern absorbiert wird und in diesem den gleichen Übergang anregt.

Auf Grund der Impulserhaltung muss bei der Aussendung eines Photons von einem ruhenden, freien Kern eben dieser Kern den gleichen Impuls  $p_K = -p_\gamma$  wie das ausgesandte Photon tragen. Die Rückstoßenergie, die auf den Kern übertragen wird, ist dann wie folgt gegeben:

$$E_R = \frac{p^2}{2M} = \frac{p_\gamma^2}{2M} = \frac{E_\gamma^2}{2Mc^2} , \quad (5)$$

mit der Photonenenergie  $E_\gamma = p_\gamma c$  und der Kernmasse  $M$ .

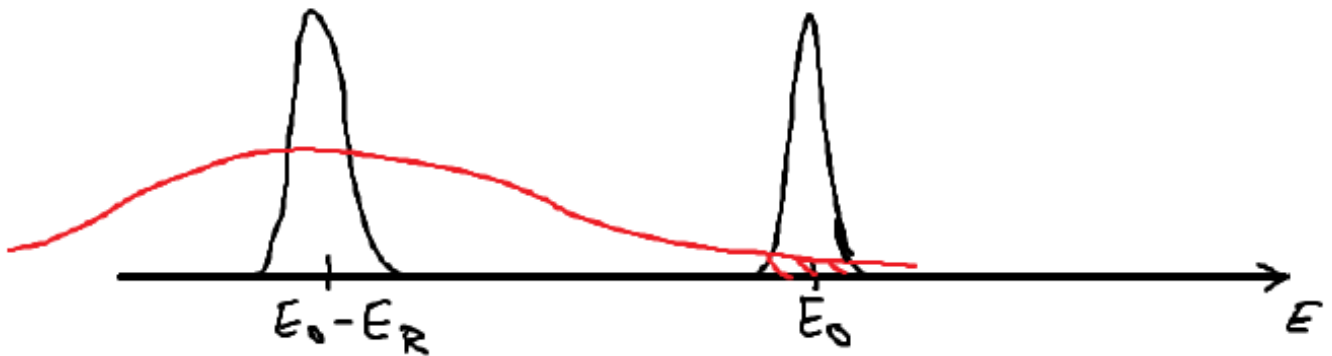
Aus Energieerhaltung folgt für die Energie des Photons somit:

$$E'_\gamma = E_0 - E_R . \quad (6)$$

mit der Übergangsenergie  $E_0$ . Bei der Reabsorption dieses Photons muss erneut die Rückstoßenergie  $E_R$  auf den Kern übertragen werden. Es trägt dann also noch folgende Energie  $E_\gamma$ :

$$E_\gamma = E_0 - 2E_R . \quad (7)$$

Die Rückstoßenergie liegt für freie Atome in einem Bereich von ca.  $10^{-3}$  eV. Die Verschiebung der Photonenenergie ist damit deutlich größer als die natürliche Linienbreite des Kernübergangs. Demnach kann bei freien Atomen theoretisch Resonanzabsorption nicht beobachtet werden. Allerdings liefert die Dopplerverbreiterung einen Beitrag, der in der Größenordnung der Rückstoßenergie liegt, sodass sich Emissions- und Absorptionsspektrum im Experiment überlappen und Resonanzabsorption stattfindet (siehe Abbildung 4).

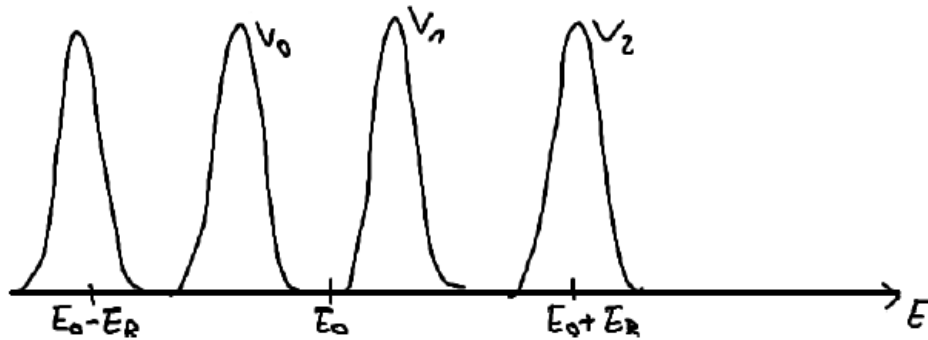


**Abbildung 4:** Dopplerverbreiterung der Emissionsenergie führt zu einem Überlapp mit dem Absorptionspeak und damit zu Resonanzabsorption bei Raumtemperatur [5]

Zudem kann (z.B. bei tieferen Temperaturen mit geringerer Dopplerverbreiterung) durch Bewegung der Quelle oder des Absorbers und daraus folgender Dopplerverschiebung die Verschiebung der Photonenenergie durch die Rückstoßenergie ausgeglichen werden. Bei einer Quellengeschwindigkeit  $v_Q$  ergibt sich für die gesamte Energieverschiebung <sup>2</sup>:

$$\Delta E = 2E_R - E_{\text{dop}} = \frac{E_\gamma^2}{Mc^2} - E_\gamma \frac{v_Q}{c} . \quad (8)$$

Bei passender Geschwindigkeit wird somit Resonanzabsorption beobachtet (siehe Abbildung 5). [5]



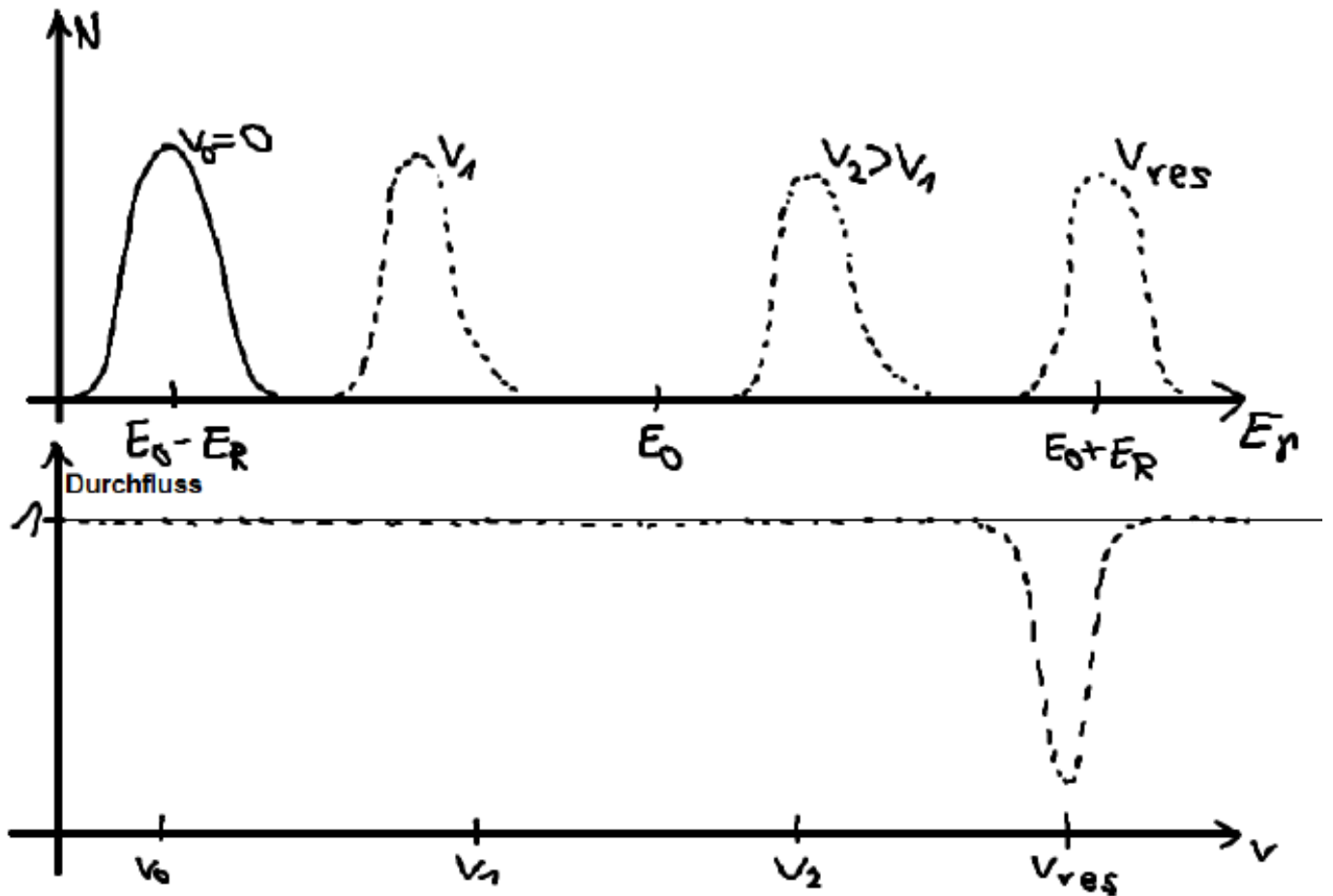
**Abbildung 5:** Die Energie der Photonen kann durch Geschwindigkeitsvariation der Quelle verändert werden. [5]

## 2.6 Mößbauer-Effekt

Beim Einbau eines Kerns in ein Kristallgitter wird der Impuls des Rückstoßes vom gesamten Kristall aufgenommen. Durch die vergleichsweise sehr große Masse des Kristalls ist die Rückstoßenergie nach 5 sehr gering. Beim Rückstoß eines Photons auf ein Mol Eisen beträgt die Rückstoßenergie beispielsweise nur etwa  $10^{-27}$  eV. Dieser Energieverlust ist deutlich kleiner als die natürliche Linienbreite, sodass trotz der Verschiebung die Photonenenergie nicht über den Bereich der natürlichen Linienbreite hinaus verschoben wird. Trifft das Photon dann erneut auf einen Atomkern, kann auch hier der Rückstoß vernachlässigt werden und es kommt zur Reabsorption. Auch hier kann analog zu freien Atomen die Photonenenergie durch gerichtete Bewegung verschoben werden. Bei Messung der Transmissionsrate hinter dem Absorber erwartet man dann einen Einbruch bei der relativen Resonanzgeschwindigkeit  $v_{\text{res}}$  (siehe Abbildung 6). Theoretisch liegt diese bei  $v_0 = 0$ , da, wie bereits erwähnt, hier die Rückstoßenergie vernachlässigt werden kann. In der Realität beobachtet man eine Verschiebung der Resonanzgeschwindigkeit auf Grund der Isomerie-Verschiebung (siehe Abschnitt 2.8.1). Diesen Effekt der rückstoßfreien Resonanzabsorption bezeichnet man als Mößbauer-Effekt. [5]

---

<sup>2</sup>Herleitung siehe [5]



**Abbildung 6:** Mößbauerspektrum, oben: Dopplerverschiebung der Energie als Ausgleich zum Rückstoßverlust; unten: Transmission hinter dem Absorber in Abhängigkeit der Geschwindigkeit [5]

## 2.7 Gittermodelle

Im Folgenden Abschnitt sollen die beiden wichtigsten Gittermodelle, das Einstein- und das Debye-Modell, sowie ihre jeweilige Gültigkeit diskutiert werden. [6]

### 2.7.1 Einsteinmodell

Das Einsteinmodell macht die Annahme, dass jedes Atom im Gitter als harmonischer Oszillator mit 3 Freiheitsgraden mit der Frequenz  $\omega_E$  beschrieben werden kann. Ein Gitter wird demnach als  $N$  Oszillatoren mit  $3N$  Freiheitsgraden und der einheitlichen Frequenz  $\omega_E$  betrachtet. Die Energie eines harmonischen Oszillators ist gegeben durch:

$$E_n = \left( n + \frac{1}{2} \right) \hbar \omega \quad . \quad (9)$$

Nach diesem Modell können durch Energieübertrag (z.B. Rückstoßenergie) Gitterschwingungen, sogenannte Phononen, der Energie  $\hbar \omega_E$  angeregt werden. Klassisch würde es immer zu Gitterschwingungen kommen. Durch die Quantisierung der Energie in der Quantenmechanik ist dies nicht mehr der Fall, da die Rückstoßenergie bei einem  $\gamma$ -Übergang von Kernzuständen  $E_R$  häufig kleiner als  $\hbar \omega_E$  ist. Man kann nun Wahrscheinlichkeiten dafür angeben, ob ein Übergang rückstoßfrei oder unter

Anregung eines Phonons stattfindet. Für  $^{57}\text{Fe}$  gilt:  $E_R \approx 0,2 \cdot 10^{-2} \text{ eV} < 10^{-2} \text{ eV} = \hbar\omega_E$ . Die Wahrscheinlichkeit für einen rückstoßfreien Übergang beträgt 0,8, für einen Übergang unter Anregung einer Gitterschwingung 0,2. Nach dem Einsteinmodell ist der Anteil der rückstoßfreien Übergänge durch folgenden Zusammenhang gegeben:

$$f_E = 1 - \frac{E_R}{\hbar\omega_E} \approx 0,9 \quad (10)$$

In der Realität sind die Phononen eines Körpers allerdings deutlich komplexer. Das Einstein-Modell bietet eine gute Näherung für hohe Temperaturen, versagt jedoch bei tiefen.

### 2.7.2 Debyemodell

Die Ursache der Unstimmigkeiten zwischen dem Einsteinmodell und den experimentellen Daten ist hauptsächlich die Annahme, dass eine einzelne Frequenz alle  $3N$  harmonischen Oszillatoren charakterisiert. Debye verbesserte Einsteins Theorie, indem er die Vibrationen eines Körpers als Ganzes betrachtete und von einem kontinuierlichen, elastischen Festkörper unter Berücksichtigung der Dispersionsrelation ausging. Er ordnete die Innere Energie des Festkörpers stationären, elastischen Schallwellen zu. Jede unabhängige Vibrations- (oder Normal-)Mode wird dabei als ein Freiheitsgrad behandelt. Anders ausgedrückt behandelt Debye einen Festkörper also als Phononengas. Vibrationswellen sind Materiewellen, jede mit ihrer eigenen de Broglie-Wellenlänge und assoziiertem Teilchen. Die 1. Brilliouin-Zone des Gitters lässt sich durch eine Kugel mit dem Radius  $k_D$  ersetzen, der dem Debye-Wellenvektor entspricht. Hieraus ergibt sich die Debye-Frequenz als Grenzfrequenz zu:

$$\omega_D = v \cdot k_D \quad (11)$$

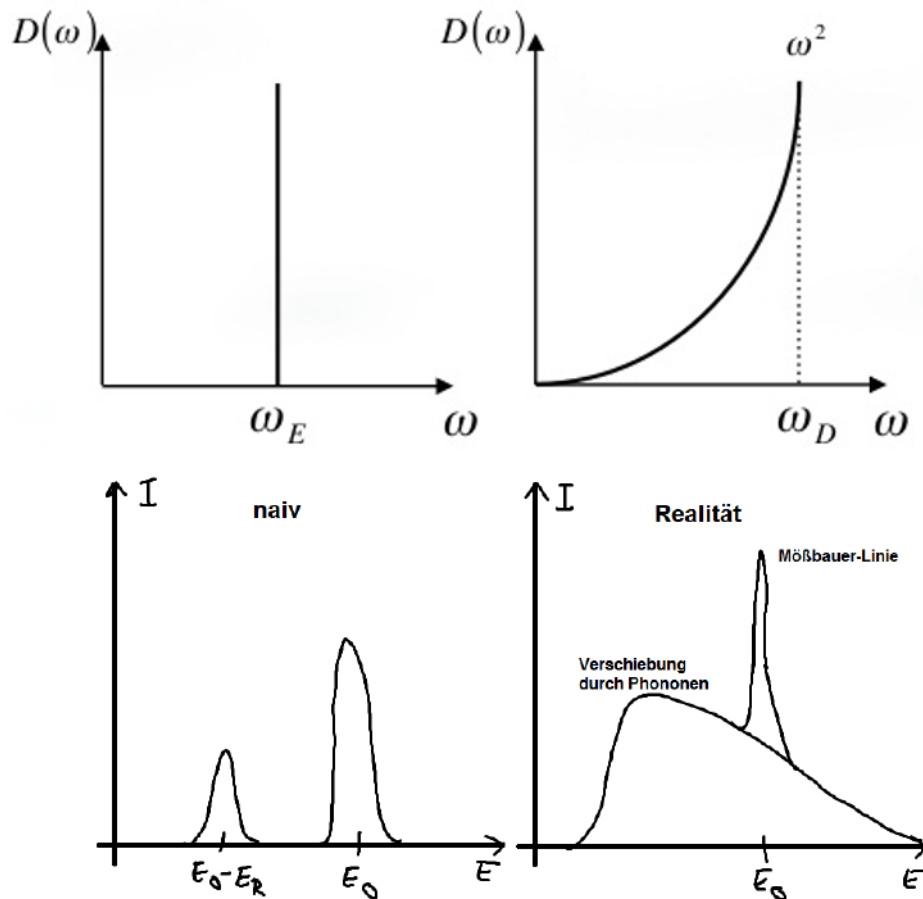
und die Debye-Temperatur zu:

$$\theta_D = \frac{\hbar\omega_D}{k} . \quad (12)$$

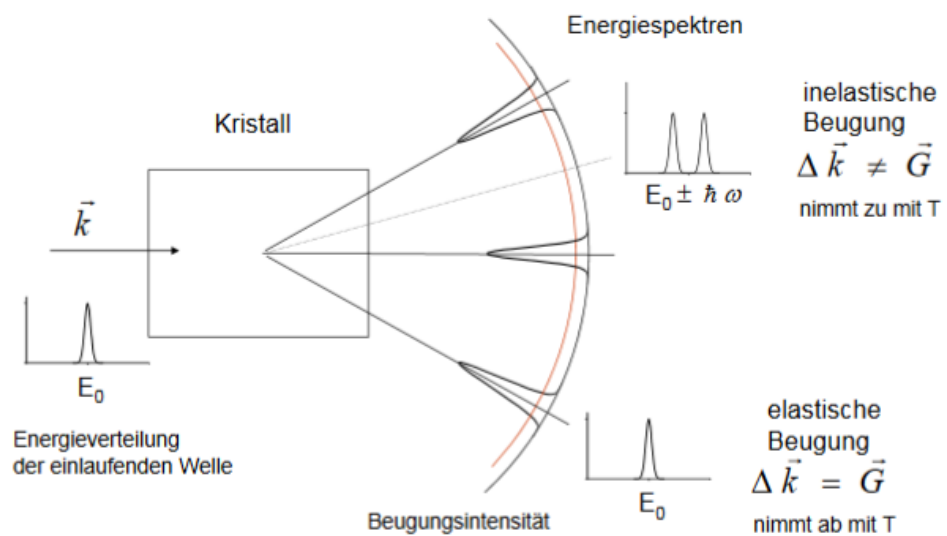
Die Debye-Temperatur gibt an, bei welcher Temperatur alle Schwingungsmoden angeregt sind. Das Debye-Modell hat nur innerhalb dieser Grenzen Gültigkeit.

In Abbildung 7 sind das Einsteinmodell und das Debyemodell im Vergleich dargestellt.

Auch im Debye-Modell kann der Bruchteil der rückstoßfreien Emissionen bestimmt werden, und zwar genauer als im Einstein-Modell. Dieser sogenannte Debye-Waller-Faktor ist temperaturabhängig. Das Prinzip hinter diesem physikalischen Prozess ist in Abbildung 8 dargestellt.



**Abbildung 7:** Vergleich zwischen Einstein- und Debye-Modell: oben: Frequenzverteilung bei Einstein (links) und Debye (rechts); unten: Energieverteilung im Festkörper beim Einstein-Modell (links) und in der Realität (durch Debye gut beschrieben, rechts). [5]



**Abbildung 8:** Anteil rückstoßfreier Emissionen im Debye-Modell. [5]

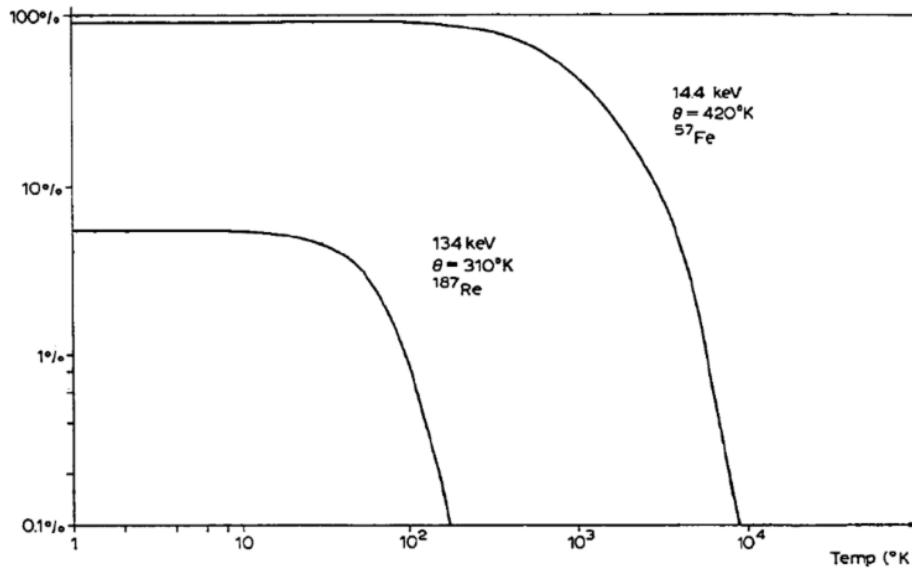
Der Debye-Waller-Faktor ergibt sich nach dem Debye-Modell zu

$$f = \exp \left[ -\frac{3E_R}{2k\theta_D} \left( 1 + \left( \frac{2T^2}{\theta_D} \int_0^{\theta_D/T} \frac{x dx}{e^x - 1} \right) \right) \right]. \quad (13)$$

Für Temperaturen  $T$ , für die  $T \ll \theta_D$  gilt, und Kerne in artemigen kubischen Gittern, kann folgende Näherung gemacht werden:

$$f \approx \exp \left[ -\frac{E_R}{k\theta_D} \left( \frac{3}{2} + \left( \frac{\pi T}{\theta_D} \right)^2 \right) \right]. \quad (14)$$

Die Debye-Temperatur von Eisen liegt bei ca. 470 K. Bei Raumtemperatur ergibt sich somit ein Debye-Waller-Faktor von  $f = 0.8$ , 80% der Übergänge finden also rückstoßfrei statt. Diese Eigenschaften machen Eisen zu einem guten Mößbauer-Kandidaten. Dies ist auch noch einmal in Abbildung 9 dargestellt. Man erkennt deutlich, dass der Debye-Waller-Faktor für Eisen bis über Raumtemperatur hinaus kaum absinkt, der Versuch kann also ungekühlt durchgeführt werden.



**Abbildung 9:** Temperaturabhängigkeit des Debye-Waller-Faktors für Eisen und Rhenium. [5]

## 2.8 Modifikationen des Übergangsspektrums

Im folgenden Abschnitt werden die Effekte, die zur Verschiebung und Aufspaltung der Energielevel beitragen, diskutiert.

### 2.8.1 Isomerieverschiebung

Die Änderung der chemischen Umgebung eines Atoms, also zum Beispiel der Einbau in ein Gitter, bewirkt eine effektive Änderung des Coulomb-Potentials des Kerns und dessen effektiven Radius. Auf Grund der Potentialabhängigkeit der Energie von Kernübergängen, verschiebt sich somit die Übergangsenergie. Dieser als Isomerieverschiebung bezeichnete Effekt tritt in diesem Versuch auf Grund der unterschiedlichen chemischen Umgebung von Quelle und Absorber auf und ist deshalb bei der Auswertung der Messergebnisse zu berücksichtigen.

### 2.8.2 Hyperfeinstrukturaufspaltung

Zwischen dem Magnetfeld der Elektronenhülle und dem magnetischen Kernspin findet eine Wechselwirkung statt, die zur Aufhebung der Entartung der Energieniveaus führt. Diese Hyperfeinstrukturaufspaltung ist gegeben durch:

$$E_{\text{mag}} = \frac{\mu_I m_I}{I} B \quad (15)$$

mit dem magnetischen Kernmoment  $\mu_I$ , dem Kernspin  $I$ , der magnetischen Quantenzahl  $m_I$  und dem Magnetfeld  $B$  am Kernort. Bei dem in diesem Versuch zu untersuchenden Übergang spaltet sich der Grundzustand in zwei Linien und der angeregte Zustand in vier Linien auf. Von diesen acht theoretischen Übergängen werden auf Grund der Auswahlregel  $\Delta m = 0, \pm 1$  nur sechs im Mößbauer-Spektrum beobachtet. Das durch die beiden beschriebenen Effekte resultierende Spektrum ist in Abbildung 10 dargestellt.

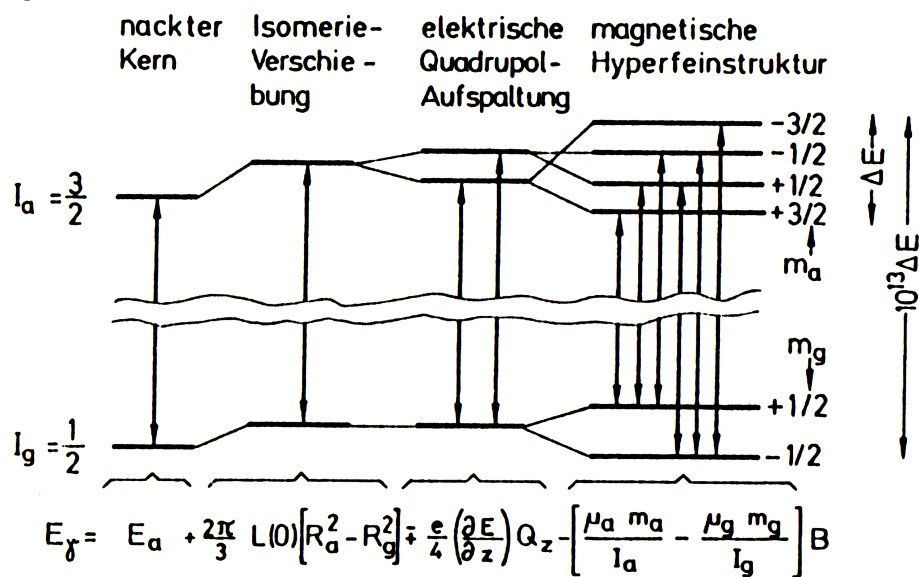


Abbildung 10: Hyperfeinaufspaltung von Eisen-57. [7]

## 2.9 Geräte und Signalformen

In diesem Abschnitt werden die wesentlichen Komponenten des Aufbaus diskutiert. [2]

**Detektor** Der Szintillator und der Photomultiplier bilden zusammen den Detektor.

**Main Amplifier (MA)** Der Hauptverstärker verstärkt das Spannungssignal rauscharm und erzeugt einen möglichst kurzen Puls, dessen Dauer amplitudenunabhängig ist. Die Pulshöhe ist dabei annähernd proportional zur deponierten Ladungsmenge im Szintillator. Der Hauptverstärker hat zwei verschiedene Ausgänge, die entweder ein bipolares Signal (zeitsensible Messungen) oder ein unipolares Signal (Aufnahme der Spektren) liefern (siehe Abbildung 11).

**Multi Channel Analyzer** Der Multi Channel Analyzer ordnet jeden eingehenden Puls abhängig von der Pulshöhe (also von der im Szintillator deponierten Ladungsmenge) einem Channel zu. Das so erhaltene Energiespektrum kann als Histogramm dargestellt werden.

**Single Channel Analyzer** Ein Single Channel Analyzer selektiert aus eingehenden Signalen, indem nur für Energien innerhalb eines einstellbaren Energiefensters ein Ausgangspuls erzeugt wird. Für die Messung der Einlinien- und Sechslinienspektren sollen hier die Energiefenster dem 14,4 keV-Peak angepasst werden (siehe Abschnitt 3.1).

**Linear Gate und Delay Unit** Liegt am enable-Eingang des Gates ein Signal an, so wird in dieser Zeit das Signal des Eingangs direkt an den Ausgang weitergeleitet. Mit einem Schlitzschraubenzieher kann die Dauer des Gatesignals eingestellt werden. Mit der Delay-Unit können Signale um einige Mikrosekunden verzögert werden und so z.B. unterschiedlich lange Signalwege ausgleichen. Die erwarteten Signalformen nach den einzelnen Einheiten sind in Abbildung 11 dargestellt. Die hier beschriebenen Signalformen werden während des Aufbaus des Versuches mit Hilfe eines Oszilloskops überprüft (siehe Kapitel 3).



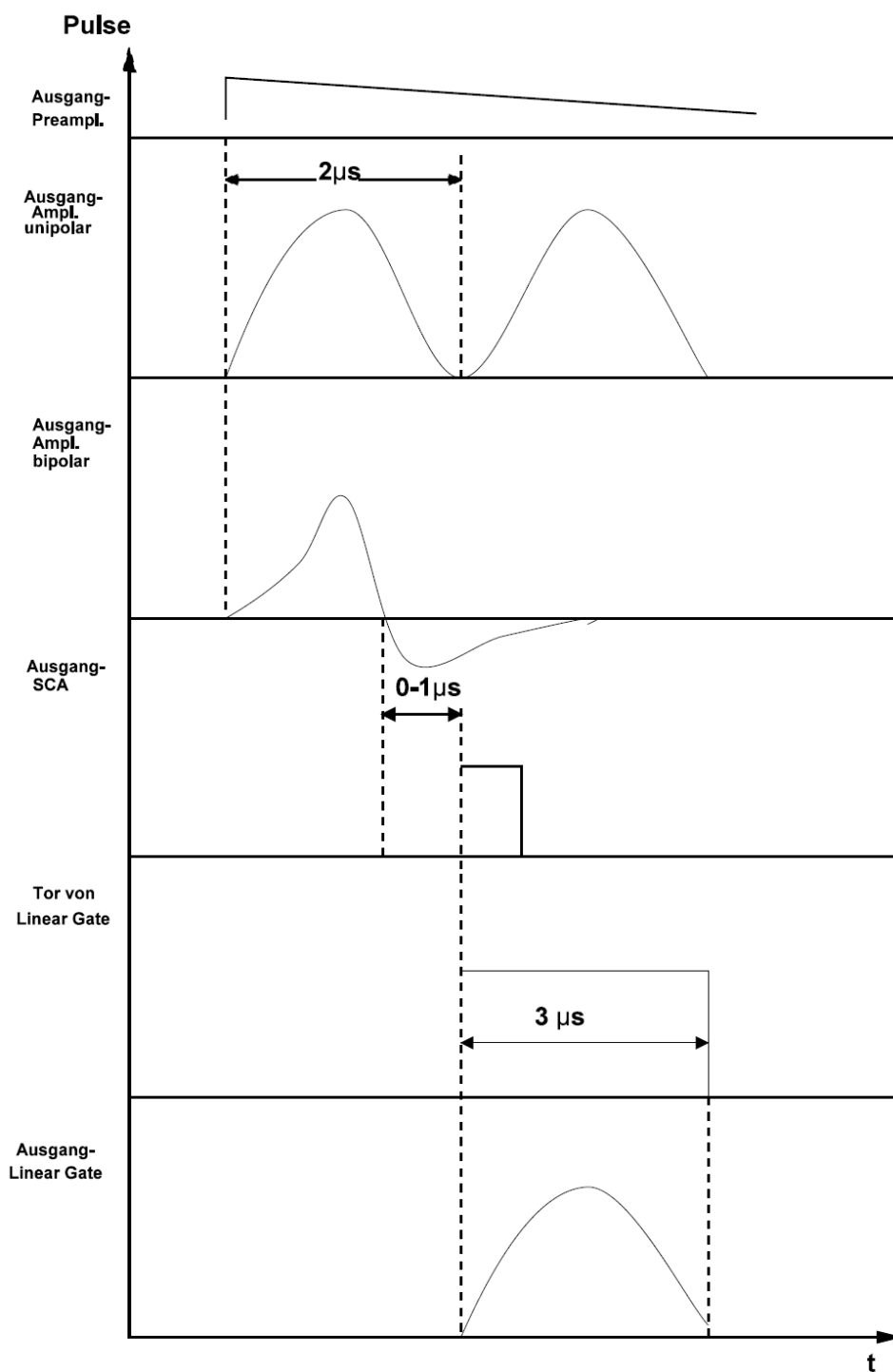


Abbildung 11: Erwartete Signalformen nach den einzelnen Einheiten. [4]

### 3 Versuchsaufbau und -Durchführung

#### 3.1 Eichung des MCA und Setzen des Energiefensters

Zu Beginn des Versuchs werden mit Hilfe verschiedener Metalle, deren Emissionsspektren bekannt sind, Eichmessungen durchgeführt. Dabei werden die Emissionsspektren aufgenommen und die bekannten Linien in diesen Spektren identifiziert. Anschließend wird daraus ein Umrechnungsfaktor zwischen Channels und Energie berechnet, und damit der zu beobachtende 14,4 keV-Peak von  $^{57}\text{Fe}$  identifiziert. Nun wird das Energiefenster des SCAs auf den 14,4 keV-Peak gesetzt, um nur Photonen mit dieser Energie zu detektieren. Zur Aufnahme der Eichspektren wird die in Abbildung 12 beschriebene Schaltung verwendet.

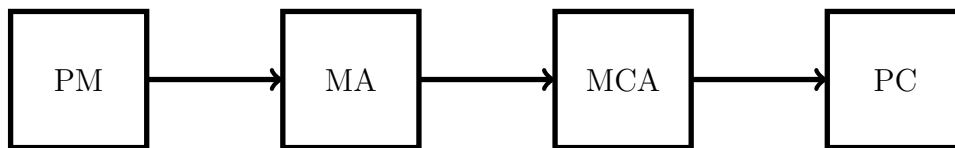


Abbildung 12: Schematische Darstellung des Versuchsaufbaus für die Eichmessung

#### 3.2 Aufbau der Schaltung

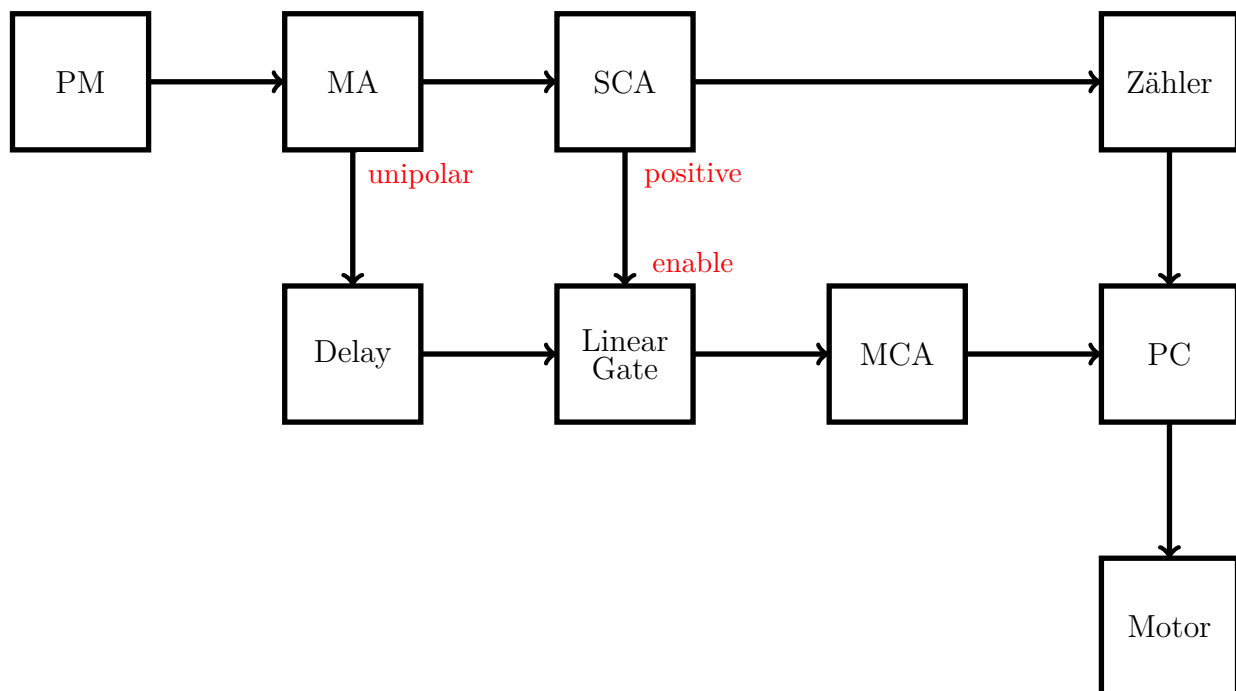
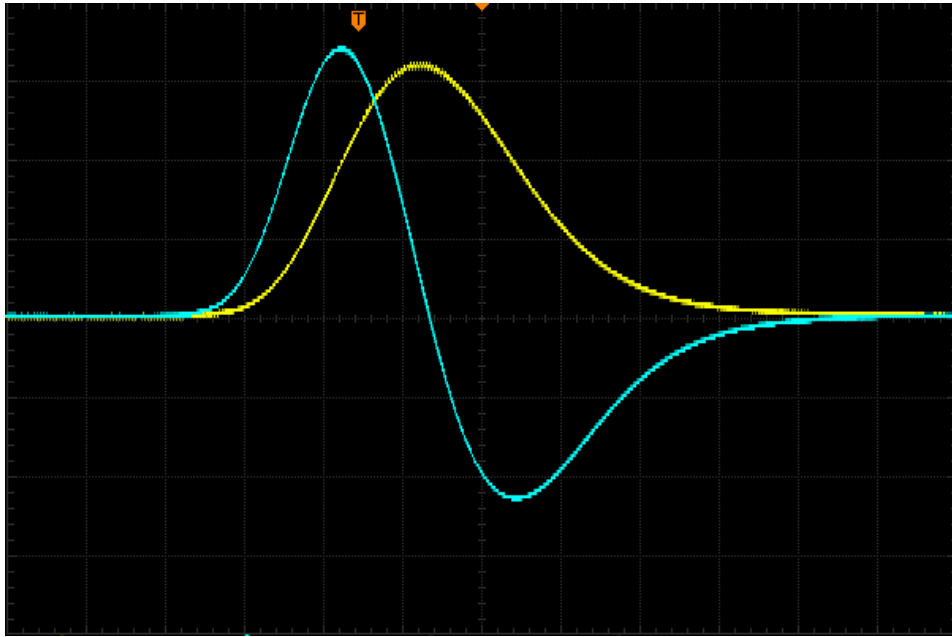


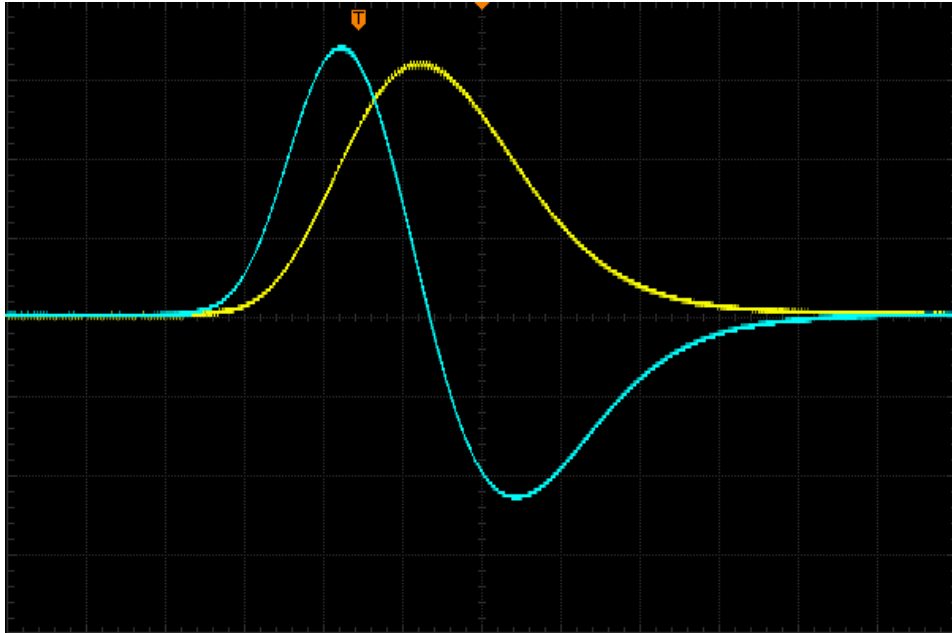
Abbildung 13: Schematische Darstellung des Versuchsaufbaus. Hierbei bezeichnet PM den Photomultiplier mit dem Szintillationszähler, MA den Hauptverstärker und SCA den Single Channel Analyzer.

Die Schaltung für den Rest des Versuchs wird gemäß Abbildung 13 aufgebaut. Der Photomultiplier wird an den Hauptverstärker angeschlossen. Dieser liefert sowohl an den SCA als auch an das Linear Gate ein unipolares Signal (siehe Abbildung 14), welches über eine Delay-Einheit verzögert werden kann. Im SCA kann nun ein Energiefenster gesetzt werden, mit welchem der Durchlass des Linear

Gates gesteuert werden kann (siehe Abbildung 15). Gleichzeitig ist der SCA an einen Zähler angeschlossen, welcher mit dem Programm zur Motorsteuerung auf dem PC verbunden ist. Ebenso ist das Linear Gate über einen MCA an den PC angeschlossen.



**Abbildung 14:** Unipolarer (gelb) und bipolarer (blau) Ausgang des Hauptverstärkers.



**Abbildung 15:** Signalausgänge des SCA (gelb) und des Linear Gates (blau).

### 3.3 Untergrundmessungen

Die Messungen der Transmission werden durch zwei Untergrundfaktoren beeinflusst. Dies sind die Compton-Streuung von höherenergetischen Photonen und die Abschwächung durch das Plexiglas der Absorberhaltungen. Um diese zu bestimmen wird die Transmission durch Plexiglas gemessen. Zur Messung des Compton-Untergrunds wird die Zählrate der Quelle mit Abschirmung durch Aluminiumplatten für verschiedene Plattendicken gemessen und daraus der Compton-Untergrund zur Dicke  $d_0 = 0$  extrapoliert.

### 3.4 Hauptmessungen

Nun werden die beiden Hauptmessungen durchgeführt. In der ersten Hauptmessung wird das Transmissionsspektrum in Abhängigkeit der Geschwindigkeit von Edelstahl (Einlinienabsorber) aufgenommen. Dazu wird die Edelstahlprobe mit Hilfe eines Motors in Geschwindigkeiten von  $-6 \text{ mms}^{-1}$  bis  $6 \text{ mms}^{-1}$  in Schritten von  $0.05 \text{ mms}^{-1}$  bewegt. Dabei wird an jedem Messpunkt eine Messzeit von 600 s eingestellt.

Anschließend wird das Transmissionsspektrum von Natureisen (Sechslinienabsorber) aufgenommen. Hierbei werden mit dem Motor Geschwindigkeiten von  $-8 \text{ mms}^{-1}$  bis  $8 \text{ mms}^{-1}$  in Abständen von  $0,1 \text{ mms}^{-1}$  abgefahren. Auch hier wird an jedem Messpunkt 600 s lang gemessen.

## 4 Auswertung

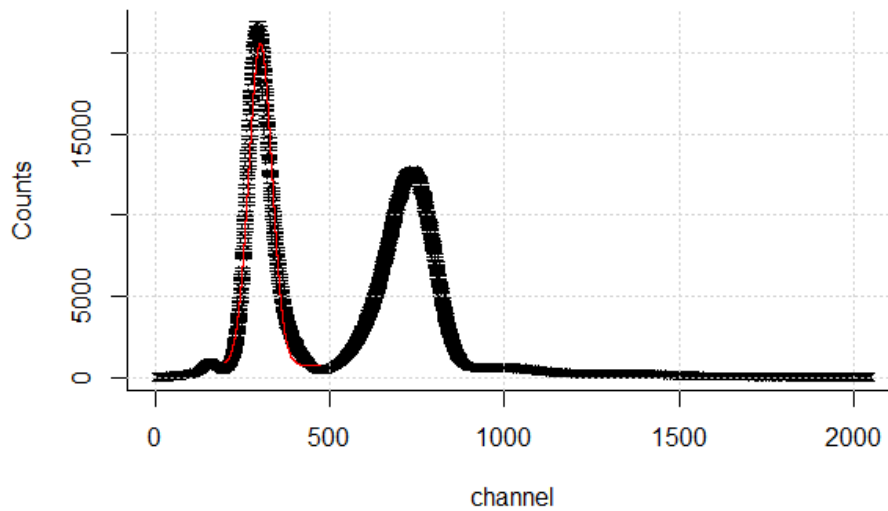
Zur Auswertung wurde die Programmiersprache R verwendet. Die Skripte zur Auswertung sind im Anhang unter Abschnitt 6.3 zu finden.

### 4.1 Energieeichung des MCA

Zur Identifizierung des 14,4 keV-Peaks von Eisen wird eine Energieeichung an Hand der bekannten Peaks verschiedener Elemente, wie in 3 beschrieben, durchgeführt. An die beobachteten Peaks werden Gaußfunktionen folgender Form gefittet:

$$r_{\text{Gauss}}(x) = C + \frac{N}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right). \quad (16)$$

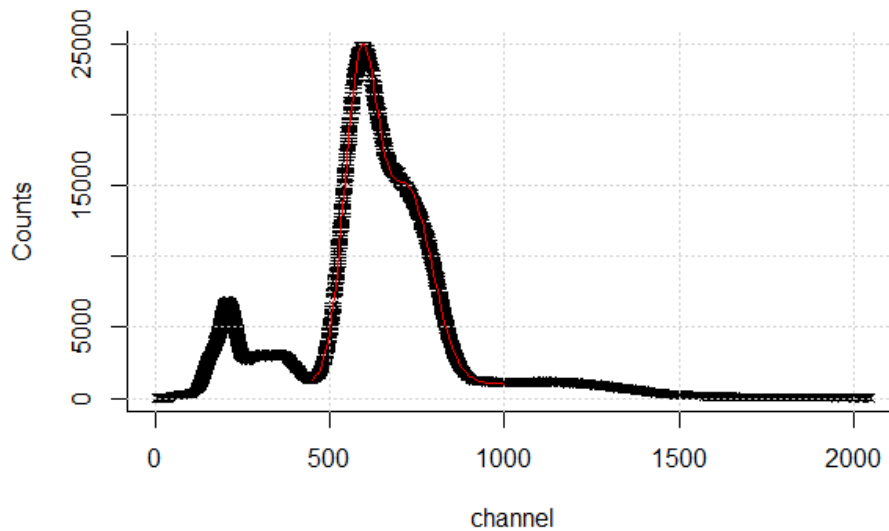
Ein beispielhafter Fit ist in Abbildung 16 für die  $K_{\alpha}$ -Linien von Silber dargestellt.



**Abbildung 16:** Beispielhafte Auswertung der  $K_{\alpha}$ -Linie von Silber

Bei einigen Spektren überlappen sich die Peaks, sodass hier doppelte Gaußfits gefittet wurden. Dies ist beispielhaft in Abbildung 17 für Terbium dargestellt. Die Spektren der restlichen Elemente mit zugehörigen Fits sowie der Code in R sind im Anhang unter Abschnitt 6.2 und 6.3 zu finden.

Material	$C$ / Counts	$N$ / $\cdot 10^6$ (Counts $\cdot$ Channels)	$\mu$ / Channels	$\sigma$ / Channels	$\frac{\chi^2}{\text{ndf}}$
Ba	$2369 \pm 4$	$2,714 \pm 0,011$	$418,56 \pm 0,12$	$38,4 \pm 1,4$	1,84
Ag	$733 \pm 4$	$1,64 \pm 0,02$	$301,7 \pm 0,4$	$32,8 \pm 0,4$	208,9
Mo	$682 \pm 24$	$0,545 \pm 0,005$	$237,8 \pm 0,2$	$26,5 \pm 0,2$	23,4
Rb	$204 \pm 11$	$0,246 \pm 0,003$	$178,96 \pm 0,16$	$21,86 \pm 0,18$	$8 \pm 5$
Tb	$992 \pm 11$	$2,541 \pm 0,018$	$588,5 \pm 0,3$	$47,11 \pm 0,17$	9,39

**Tabelle 1:** Fitdaten der Gaußfits zur Eichung**Abbildung 17:** Doppel-Gaußfit zur Auswertung der  $K_{\alpha}$ -Linie von Terbium

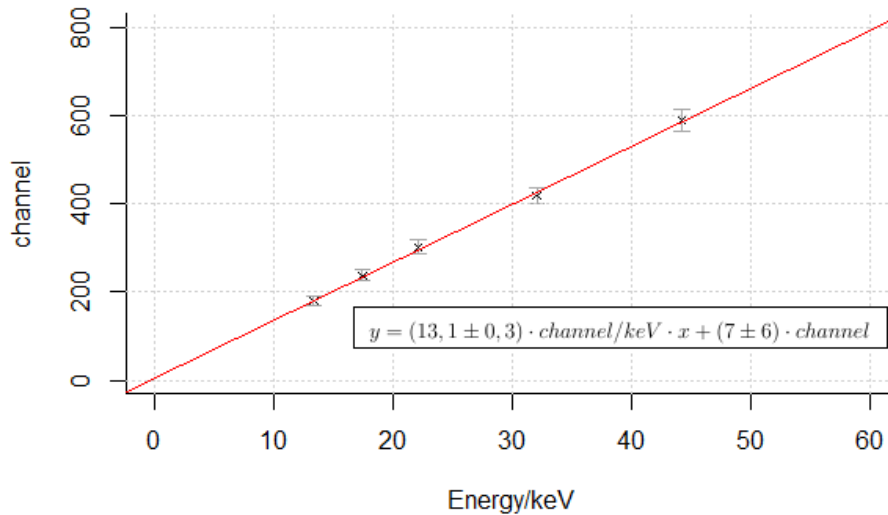
Die Fitparameter der einzelnen Peaks, die für die Energieeichung verwendet wurden, sind in 1 aufgestellt.

Relevant für die Energieeichung ist dabei der Mittelwert, der die Energie des Peaks, die der Anleitung entnommen wird, dem Kanal zuordnet. Als Fehler wird sinnvoller Weise die halbe Standardabweichung gewählt und nicht der Fehler auf den Fitparameter, da dieser nicht dem statistischen Fehler entspricht. Die Werte, die direkt als Punkte für die Eichgerade verwendet wurden, sind in Tabelle 2 dargestellt.

Material	$\mu$ / Channels	Energie / keV
Ba	$418,56 \pm 19$	32,06
Ag	$301,7 \pm 16$	22,10
Mo	$237,8 \pm 13$	17,44
Rb	$178,96 \pm 11$	13,37
Tb	$588,5 \pm 24$	44,2

**Tabelle 2:** Fitdaten der Gaußfits zur Eichung

In Abbildung 18 sind die einzelnen Kanäle gegen die bekannten Energien aufgetragen, wobei die halbe Standardabweichung als y-Fehler aufgetragen wurden. Innerhalb dieses Fehlers liegen alle Punkte auf der Eichgeraden, sodass sich nun leicht die erwartete Position des Eisenpeaks ablesen lässt.



**Abbildung 18:** Linearer Fit zur Energiekalibrierung des MCA

Die Position des 14,4 keV-Eisenpeak ergibt sich an Hand von 18 damit zu  $195 \pm 7$ , wobei sich der Fehler mit Gauß'scher Fehlerfortpflanzung aus den Fehlern auf die Fitparameter ergibt.

## 4.2 Berechnung des Compton-Untergrunds

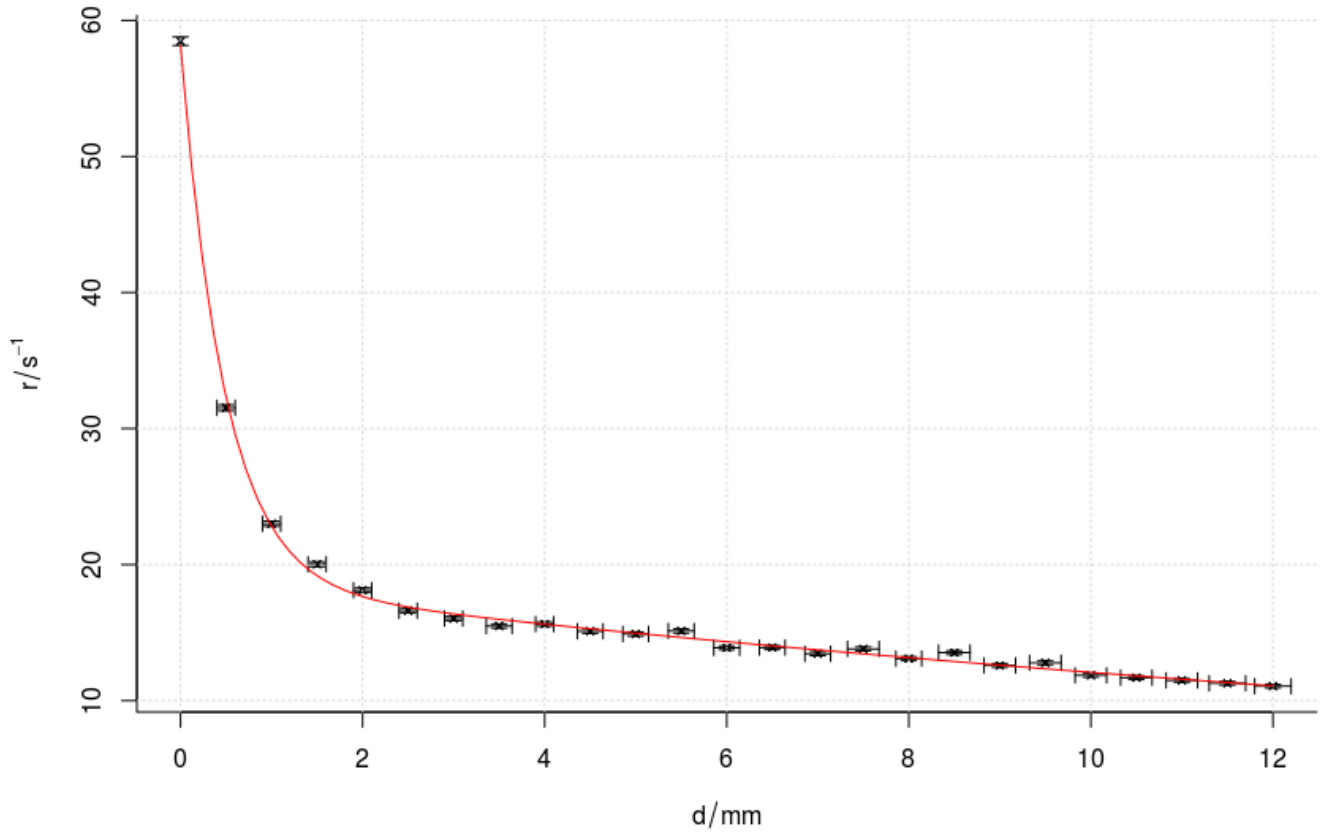
Ein Effekt, welcher in die Messungen mit einfließt, ist der Compton-Untergrund. Dieser entsteht durch Photonen aus Zerfällen höherer Energien, welche einen Teil ihrer Energie durch Compton-Streuung an Elektronen abgeben und anschließend Energien innerhalb des 14.4 keV-Energiefensters haben. Um diesen Effekt auszugleichen wird aus Messungen mit Aluminiumplatten die Compton-Zählrate bestimmt. Hierzu wird die Zählrate der Quelle mit Abschirmung durch Aluminiumplatten für verschiedene Plattendicken gemessen. Die daraus resultierende Kurve besteht zum einen aus dem Beitrag des Compton-Untergrunds und zum anderen aus dem Beitrag der exponentiellen Abschwächung durch Absorption in den Aluminiumplatten.

Um die einzelnen Beiträge zu bestimmen, wird ein doppelter Exponentialfit

$$r(d) = A \cdot e^{-\lambda d} + B \cdot e^{-\mu d} \quad (17)$$

an die Messdaten gelegt, wobei  $A \cdot e^{-\lambda d}$  den Beitrag des Compton-Untergrunds und  $B \cdot e^{-\mu d}$  den Beitrag der Materialabschwächung darstellt. Aus dem Beitrag des Compton-Untergrundes wird nun die Compton-Zählrate für die Dicke  $d = 0$  extrapoliert:

$$r_{\text{Untergrund}} = r(0) = A \cdot e^{-\lambda 0} = A. \quad (18)$$



**Abbildung 19:** Messung der Transmission durch Aluminium in Abhängigkeit der Schichtdicke zur Bestimmung des Compton-Untergrunds

Wir erhalten nun bei der Messung die in Abbildung 19 dargestellten Messdaten. Die  $y$ -Fehler wurden aus den Poissonfehlern der Zählmessung berechnet:  $s_r = r \cdot \frac{s_{\text{Counts}}}{\text{Counts}}$ . Für die Messung der Schichtdicken wurde ein Fehler von  $s_{d,i} = 0,1 \text{ mm}$  für eine Aluminiumplatte und damit  $s = \sqrt{m} \cdot s_{d,i}$  mit der Anzahl  $m$  der jeweils verwendeten Aluminiumplatten. Bei der Durchführung des Fits an Gleichung 18 erhalten wir folgende Parameter:

$$\begin{aligned} A &= 18,5 \pm 0,3 \text{ s}^{-1} , \\ B &= 39,6 \pm 0,8 \text{ s}^{-1} , \\ \lambda &= 0,043 \pm 0,002 \text{ mm}^{-1} , \\ \mu &= 2,06 \pm 0,07 \text{ mm}^{-1} \end{aligned}$$

$$\text{mit } \frac{\chi^2}{\text{ndf}} = 5,7 .$$

Daraus erhalten wir nun als Ergebnis für den Compton-Untergrund:

$$r_{\text{Untergrund}} = (18,5 \pm 0,3) \text{ s}^{-1} . \quad (19)$$

Für die Berechnung des Compton-Untergrunds wurde das in Anhang 6.3 angehängte Skript `Compton-Untergrund.R` verwendet.



### 4.3 Abschwächung durch Plexiglas

Die Absorberproben liegen in Plexiglashalterungen vor. Da das Plexiglas die  $\gamma$ -Strahlung abschwächt, muss zur Korrektur der Messdaten die Abschwächung durch das Plexiglas berechnet werden. Hierzu liegt eine leere Plexiglashalterung vor, mit welcher eine Messung der Zählraten der Quelle durchgeführt wird. Diese Messung wird ebenso ohne Plexiglashalterung durchgeführt. Aus diesen beiden Messungen wird nun das Verhältnis zwischen der Zählrate mit und ohne Plexiglas und daraus der Korrekturfaktor  $k_{\text{Plexi}}$  bestimmt. Dazu wurde das Skript `Plexiglas.R` aus Anhang 6.3 verwendet.

Dabei wurde folgendes Ergebnis erzielt:

$$k_{\text{Plexi}} = \frac{r_{\text{leer}}}{r_{\text{plexi}}} = 1,243 \pm 0,010 . \quad (20)$$

### 4.4 Umrechnung der Messwerte

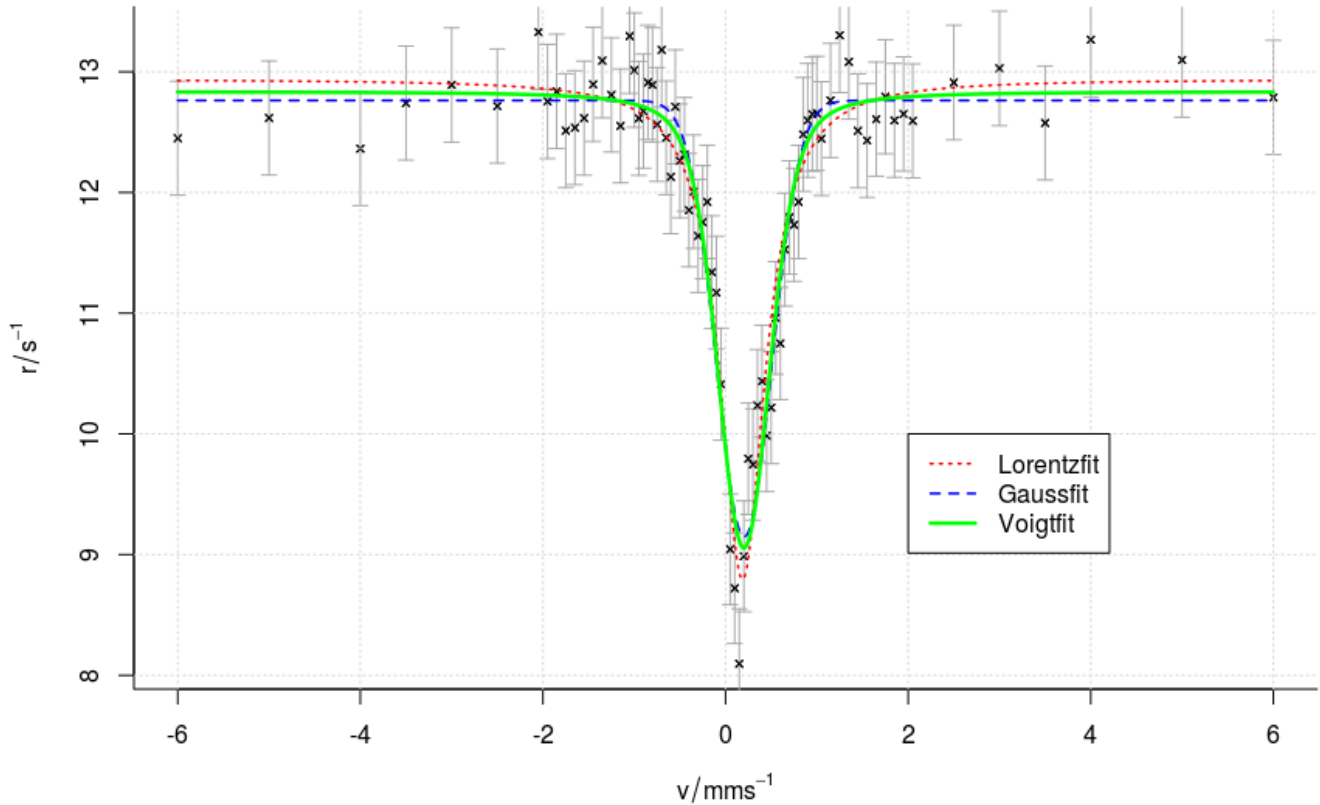
Die Geschwindigkeiten werden mit Hilfe der folgenden Formel in Energien umgerechnet:

$$E = E_0 \frac{v}{c} . \quad (21)$$

Aus den gemessenen Zählraten werden mit Hilfe des Abschwächungsfaktors und des Compton-Untergrunds die tatsächlichen Transmissionsraten bestimmt:

$$r = k_{\text{Plexi}} \cdot r_{\text{Mess}} - r_{\text{Untergrund}} . \quad (22)$$

### 4.5 Einlinienabsorber



**Abbildung 20:** Messung der Transmission durch die Edelstahlprobe (Einlinienabsorber). Hier ist die Zählrate  $r$  über die Geschwindigkeit  $v$  aufgetragen.

Bei den Messungen für den Einlinienabsorber wurden die in Abbildung 20 aufgetragenen Daten aufgenommen. An die Daten wurde zunächst eine Gaußfunktion

$$r_{\text{Gauss}}(v) = C + \frac{N}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(v - \mu)^2}{2\sigma^2}\right), \quad (23)$$

eine Lorentzfunktion

$$r_{\text{Lorentz}}(v) = C + \frac{N}{2\pi} \frac{\gamma}{(v - \mu)^2 + \frac{1}{4}\gamma^2} \quad (24)$$

und anschließend eine Voigtfunktion

$$r_{\text{Voigt}}(v) = (r_{\text{Gauss}} * r_{\text{Lorentz}})(v) \quad (25)$$

gefittet. Dabei wurden folgende Fitparameter ermittelt:

<b>r<sub>Gauss</sub>:</b>	<b>r<sub>Lorentz</sub>:</b>	<b>r<sub>Voigt</sub>:</b>
$C = (12,76 \pm 0,05) \text{ s}^{-1}$	$C = (12,94 \pm 0,06) \text{ s}^{-1}$	$C = (12,84 \pm 0,08) \text{ s}^{-1}$
$N = -(2,75 \pm 0,13) \text{ mms}^{-2}$	$N = -(3,9 \pm 0,2) \text{ mms}^{-2}$	$N = -(3,2 \pm 0,4) \text{ mms}^{-2}$
$\mu = (0,204 \pm 0,013) \text{ mms}^{-1}$	$\omega = (0,186 \pm 0,012) \text{ mms}^{-1}$	$\mu = (0,201 \pm 0,013) \text{ mms}^{-1}$
$\sigma = (0,303 \pm 0,014) \text{ mms}^{-1}$	$\gamma = (0,60 \pm 0,04) \text{ mms}^{-1}$	$\sigma = (0,22 \pm 0,05) \text{ mms}^{-1}$
		$\gamma = (0,13 \pm 0,08) \text{ mms}^{-1}$
$\frac{\chi^2}{\text{ndf}} = 0,55$	$\frac{\chi^2}{\text{ndf}} = 0,56$	$\frac{\chi^2}{\text{ndf}} = 0,55$

Zur Auswertung des Einlinienabsorbers wurden die Skripte `Einlinien.R`, `Absorberdicke.R` und `Debye-Waller.R` verwendet (siehe Anhang 6.3).

#### 4.5.1 Isomerieverschiebung

Die in Abschnitt 2.8.1 beschriebene Isomerieverschiebung kann aus der Position des Peaks bestimmt werden. Die Isomerieverschiebung wird durch die Abweichung des Peaks von der Energie  $E_0$ , also der Geschwindigkeit  $v_0 = 0$  bestimmt. Aus dem Voigt-Fit in Abschnitt 4.5 wird die Position  $\mu$  des Peaks abgelesen und mit Hilfe von Gleichung 21 in Energie umgerechnet. Daraus erhalten wir für die Isomerieverschiebung:

$$E_{\text{iso}} = E_0 \frac{\mu}{c} = (9,7 \pm 0,9) \text{ neV} \quad (26)$$

#### 4.5.2 Linienbreite und Lebensdauer

Die gemessene Linienbreite ergibt sich aus der zu messenden natürlichen Linienbreite und der relativen Linienverbreiterung durch den endlich dicken Absorber. Die natürliche Linienbreite lässt sich durch eine Lorentz-Funktion, die relative Linienverbreiterung durch die Faltung einer Gauß-Funktion an der Lorentz-Funktion beschreiben (siehe Abschnitt 2.4). Die Faltung der Lorentz- und der Gauß-Funktion ergibt eine Voigt-Funktion, welche an die Daten gefittet wurde.

**Berechnung aus dem Lorentz-Anteil des Voigt-Fits** Zunächst wird zum Berechnen der natürlichen Linienbreite  $\Gamma$  der Lorentz-Anteil des Voigt-Fits verwendet. Dieser ergibt sich aus dem Fit-Parameter  $\gamma$  des Voigt-Fits. Die natürliche Linienbreite  $\Gamma$  berechnet sich nun mit Hilfe der Energieumrechnung in Gleichung 21 folgendermaßen:

$$\Gamma = E_0 \frac{\mu}{c} = (6 \pm 4) \text{ neV} . \quad (27)$$

Daraus berechnen sich die Lebensdauer und die Halbwertszeit des 14,4 keV-Zustand folgendermaßen:

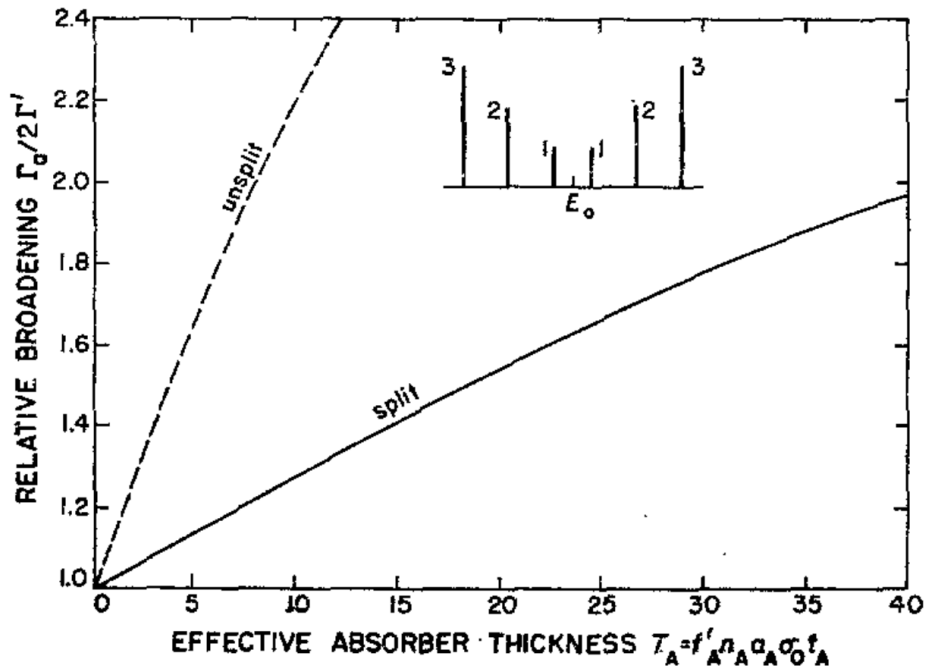
$$\tau = (110 \pm 70) \text{ ns} \quad (28)$$

$$T_{\frac{1}{2}} = (80 \pm 50) \text{ ns} \quad (29)$$

**Berechnung aus der Linienbreite mit Hilfe der effektiven Absorberdicke** Als zweite Möglichkeit zur Berechnung der natürlichen Linienbreite lässt sich die relative Linienverbreiterung aus der effektiven Absorberdicke berechnen. Anschließend kann die natürliche Linienbreite aus der gesamten Linienbreite und der relativen Linienverbreiterung berechnet werden. Die effektive Absorberdicke lässt sich auf folgende Art und Weise berechnen [4]:

$$T_A = f_A n_A \beta \sigma_0 d_A = 6,1 \pm 0,7 . \quad (30)$$

Dabei ist  $f_A$  der Debye-Waller-Faktor,  $n_A$  die Teilchendichte,  $d_A$  die Dicke des Absorbers,  $\beta$  der Anteil von  $^{57}\text{Fe}$  im Isotopengemisch und  $\sigma_0 = 2\pi\lambda^2 \cdot \frac{2I^*+1}{2I+1} \cdot \frac{1}{1+\alpha}$  der Wirkungsquerschnitt.



**Abbildung 21:** Relative Linienverbreiterung in Abhängigkeit der effektiven Absorberdicke (Quelle: [8])

Aus dem Fit erhalten wir

$$\gamma_{\text{Lorentz}} = 0,60 \pm 0,04 \text{ mms}^{-1} \quad (31)$$

und mit Hilfe von Gleichung 21 erhalten wir

$$\Gamma_0 = 29 \pm 2 \text{ neV} . \quad (32)$$

Mit Hilfe von Abbildung 21 kann nun die relative Verbreiterung

$$W = \frac{\Gamma_0}{2\Gamma} = 1.78 \pm 0.06 \quad (33)$$

graphisch bestimmt werden. Da zusätzlich zur Absorberdicke auch die Quelledicke berücksichtigt werden muss, wird die relative Verbreiterung folgendermaßen berechnet::

$$\Gamma = \frac{\Gamma_0}{4W} = (4.1 \pm 0.3) \text{ neV} \quad (34)$$

$$\tau = (162 \pm 12) \text{ ns} \quad (35)$$

$$T_{\frac{1}{2}} = (112 \pm 9) \text{ ns} \quad (36)$$

### 4.5.3 Debye-Waller-Faktor

Der Debye-Waller-Faktor der Quelle berechnet sich mit folgender Gleichung:

$$f_Q = \frac{Z_\infty - Z_\mu}{Z_\infty} \cdot \frac{1}{1 - \exp\left(-\frac{T_A}{2}\right) \cdot J_0\left(i\frac{T_A}{2}\right)} , \quad (37)$$

wobei  $J_0$  eine Besselfunktion bezeichnet. Außerdem gilt:

$$Z_\infty = r_{\text{Voigt}}(\infty) = (12,84 \pm 0,08) \text{ s}^{-1} , \quad (38)$$

$$Z_\mu = r_{\text{Voigt}}(\mu) = (9,06 \pm 0,10) \text{ s}^{-1} . \quad (39)$$

Damit gilt:

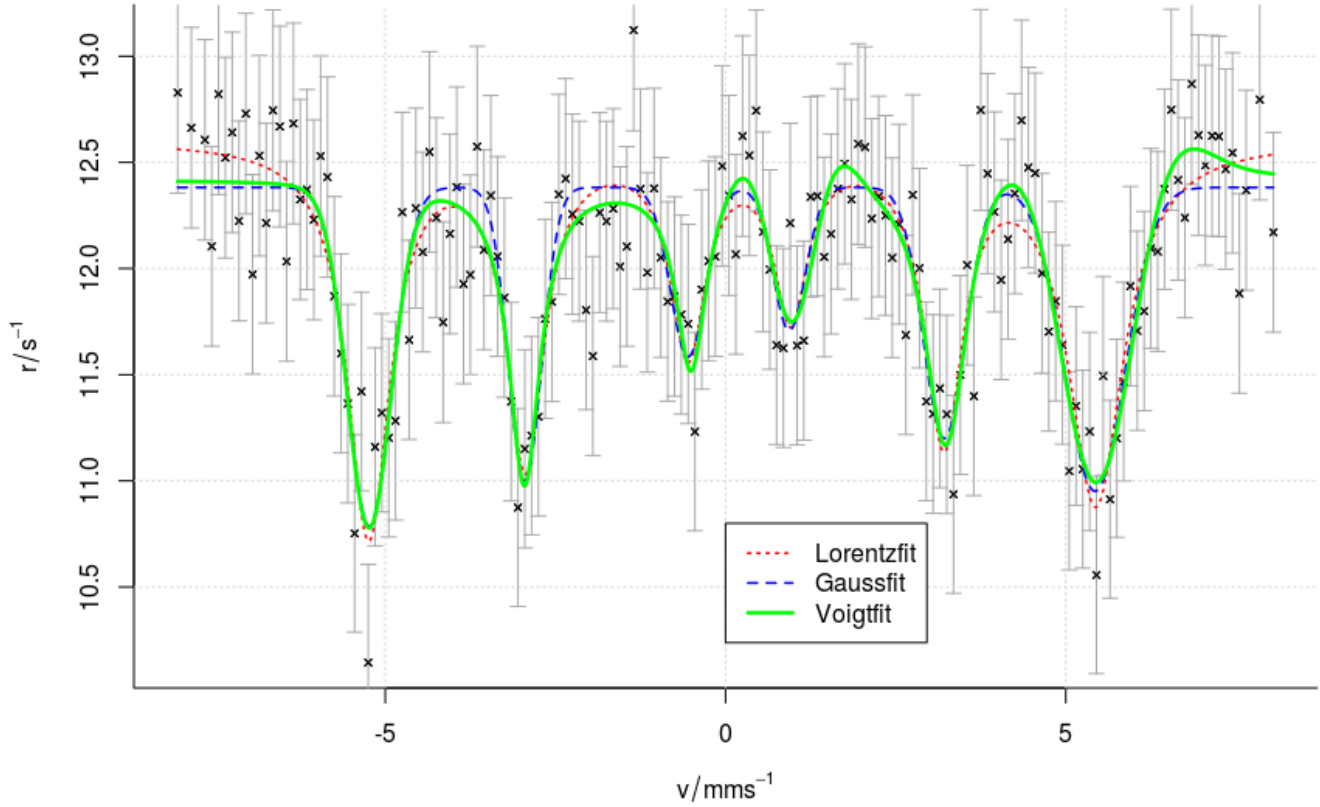
$$f_Q = 0,388 \pm 0,014 . \quad (40)$$

Der Fehler berechnet sich dabei mit Hilfe von Gauß'scher Fehlerrechnung folgendermaßen:

$$s_{f_Q} = \sqrt{\left(\frac{\partial f_Q}{\partial Z_\infty} \cdot s_{Z_\infty}\right)^2 + \left(\frac{\partial f_Q}{\partial Z_\mu} \cdot s_{Z_\mu}\right)^2 + \left(\frac{\partial f_Q}{\partial T_A} \cdot s_{T_A}\right)^2} \quad (41)$$

$$= f_Q \sqrt{\frac{\left(\frac{Z_\mu}{Z_\infty^2}\right)^2 s_{Z_\infty}^2 + \left(\frac{1}{Z_\infty}\right)^2 s_{Z_\mu}^2}{\frac{Z_\infty - Z_\mu}{Z_\infty}} + \left(\frac{\frac{1}{2} \exp\left(-\frac{T_A}{2}\right) \left(iJ_{-1}\left(i\frac{T_A}{2}\right) - J_0\left(i\frac{T_A}{2}\right)\right)}{1 - \exp\left(-\frac{T_A}{2}\right) \cdot J_0\left(i\frac{T_A}{2}\right)}\right)^2 s_{T_A}^2} . \quad (42)$$

## 4.6 Sechslinienabsorber



**Abbildung 22:** Messung der Transmission durch die Eisenprobe (Sechslinienabsorber). Hier ist die Zählrate  $r$  über die Geschwindigkeit  $v$  aufgetragen.

Die für den Sechslinienabsorber erhaltenen Daten sind in Abbildung 22 aufgetragen. An diese Daten wurde eine sechsfache Gaußfunktion

$$r_{\text{Gauss}}^{n=6}(v) = C + \sum_{i=1}^n \frac{N_i}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(v - \mu_i)^2}{2\sigma_i^2}\right), \quad (43)$$

eine sechsfache Lorentzfunktion

$$r_{\text{Lorentz}}^{n=6}(v) = C + \sum_{i=1}^n \frac{N_i}{2\pi} \frac{\gamma_i}{(v - \mu_i)^2 + \frac{1}{4}\gamma_i^2} \quad (44)$$

und anschließend eine sechsfache Voigtfunktion

$$r_{\text{Voigt}}^{n=6}(v) = \sum_{i=1}^n (r_{\text{Gauss},i} * r_{\text{Lorentz},i})(v) \quad (45)$$

gefittet. Die hierbei erhaltenen Fitparameter sind in den Tabellen 3 bis 5 aufgeführt.

Die Auswertung des Sechslinienabsorbers wurde mit dem Skript `Sechslinien.R` in Anhang 6.3 durchgeführt.

$i$	$N_i / \text{mms}^{-2}$	$\mu_i / \text{mms}^{-1}$	$\sigma_i / \text{mms}^{-1}$
1	$-1,60 \pm 0,15$	$-5,24 \pm 0,03$	$0,33 \pm 0,04$
2	$-1,38 \pm 0,17$	$-2,95 \pm 0,03$	$0,24 \pm 0,04$
3	$-0,80 \pm 0,17$	$-0,55 \pm 0,06$	$0,26 \pm 0,06$
4	$-0,67 \pm 0,18$	$0,94 \pm 0,07$	$0,24 \pm 0,07$
5	$-1,18 \pm 0,15$	$3,21 \pm 0,05$	$0,31 \pm 0,05$
6	$-1,43 \pm 0,13$	$5,43 \pm 0,05$	$0,44 \pm 0,05$

$$C = (12,38 \pm 0,04) \text{ s}^{-1}$$

$$\frac{\chi^2}{\text{ndf}} = 0,38$$

**Tabelle 3:** Fitdaten des sechsfachen Gaußfits aus Abbildung 22 (siehe Gleichung 43)

$i$	$N_i / \text{mms}^{-2}$	$\mu_i / \text{mms}^{-1}$	$\gamma_i / \text{mms}^{-1}$
1	$-2,2 \pm 0,3$	$-5,24 \pm 0,03$	$0,77 \pm 0,12$
2	$-1,6 \pm 0,3$	$-2,94 \pm 0,04$	$0,68 \pm 0,13$
3	$-1,0 \pm 0,2$	$-0,55 \pm 0,06$	$0,6 \pm 0,2$
4	$-0,7 \pm 0,2$	$0,96 \pm 0,08$	$0,6 \pm 0,2$
5	$-1,5 \pm 0,3$	$3,21 \pm 0,04$	$0,68 \pm 0,14$
6	$-2,6 \pm 0,4$	$5,44 \pm 0,04$	$0,99 \pm 0,15$

$$D = (12,61 \pm 0,06) \text{ s}^{-1}$$

$$\frac{\chi^2}{\text{ndf}} = 0,39$$

**Tabelle 4:** Fitdaten des sechsfachen Lorentzfits aus Abbildung 22 (siehe Gleichung 44)

$i$	$N_i / \text{mms}^{-2}$	$\mu_i / \text{mms}^{-1}$	$\sigma_i / \text{mms}^{-1}$	$\gamma_i / \text{mms}^{-1}$
1	$-1,3 \pm 0,4$	$-5,24 \pm 0,03$	$0,31 \pm 0,12$	$0,0 \pm 0,2$
2	$-1,3 \pm 0,3$	$-2,96 \pm 0,04$	$0,0 \pm 1,5$	$0,28 \pm 0,15$
3	$-0,7 \pm 0,2$	$-0,51 \pm 0,07$	$0 \pm 60$	$0,3 \pm 0,2$
4	$-0,16 \pm 0,19$	$0,98 \pm 0,07$	$0,5 \pm 0,2$	$-0,6 \pm 0,8$
5	$-1,5 \pm 0,4$	$3,24 \pm 0,05$	$0,1 \pm 0,4$	$0,3 \pm 0,2$
6	$-1,1 \pm 0,4$	$5,45 \pm 0,04$	$0,69 \pm 0,16$	$0,6 \pm 0,4$

$$C = (12,41 \pm 0,07) \text{ s}^{-1}$$

$$\frac{\chi^2}{\text{ndf}} = 0,37$$

**Tabelle 5:** Fitdaten des sechsfachen Voigtfits aus Abbildung 22 (siehe Gleichung 45)

### 4.6.1 Hyperfeinaufspaltung

Wie in Abschnitt 2.8.2 beschrieben, erlaubt die Hyperfeinstrukturaufspaltung sechs Übergänge vom angeregten Zustand in den Grundzustand. Diese können wie in Tabelle 6 beschrieben den Peaks zugeordnet werden.

$i$	1	2	3	4	5	6
$m_a$	$\frac{3}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{3}{2}$
$m_g$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$

**Tabelle 6:** Magnetische Quantenzahlen der Hyperfeinaufspaltung von  $^{57}\text{Fe}$ . Die Quantenzahl des Grundzustands ist  $m_g$  und  $m_a$  die Quantenzahl des angeregten Zustands.

Aus den magnetischen Momenten  $\mu_a$  und  $\mu_g$  lässt sich nun für jeden Peak mit Gleichung 15 die theoretische Aufspaltungsenergie  $E_i$  berechnen:

$$E_1 = (\mu_a - \mu_g)B = -E_6 \quad (46)$$

$$E_2 = \left(\frac{1}{3}\mu_a - \mu_g\right)B = -E_5 \quad (47)$$

$$E_3 = \left(-\frac{1}{3}\mu_a - \mu_g\right)B = -E_4 \quad (48)$$

### 4.6.2 Isomerieverschiebung

Aus den Gleichungen 46 bis 48 lässt sich die in Abschnitt 2.8.1 beschriebene Isomerieverschiebung bestimmen. Aus dem Voigt-Fit in Abschnitt 4.5 wird die Position  $\mu$  der Peaks abgelesen und mit Hilfe von Gleichung 21 in Energien umgerechnet. Anschließend wird daraus der Nullpunkt für jedes Energiepaar und danach der Mittelwert aus den drei erhaltenen Werten berechnet. Daraus erhalten wir für die Isomerieverschiebung:

$$E_{\text{iso}} = (6,7 \pm 0,9) \text{ neV} . \quad (49)$$

Die  $E_i$  berechnen sich nun durch

$$E_i = E_{i,\text{mess}} - E_{\text{iso}} . \quad (50)$$

### 4.6.3 Magnetisches Moment

Das magnetische Moment des Grundzustands ist gegeben durch [9]:

$$\mu_g = (0.090604 \pm 0.000009) \mu_N , \quad (51)$$

mit dem Kernmagneton  $\mu_N = \frac{e\hbar}{2m_{\text{Proton}}}$ .

Aus den Gleichungen 46 bis 48 ergeben sich die folgenden Verhältnisse:

$$\frac{E_2}{E_3} = \frac{E_5}{E_4} = \frac{\frac{1}{3}\mu_a - \mu_g}{-\frac{1}{3}\mu_a - \mu_g} \quad (52)$$

$$\frac{E_1}{E_3} = \frac{E_6}{E_4} = \frac{\mu_a - \mu_g}{-\frac{1}{3}\mu_a - \mu_g} \quad (53)$$

$$\frac{E_1}{E_2} = \frac{E_6}{E_5} = \frac{\mu_a - \mu_g}{\frac{1}{3}\mu_a - \mu_g} . \quad (54)$$

Daraus ergeben sich folgende Gleichungen für das magnetische Moment des angeregten Zustands:

$$\mu_a = 3\mu_g \left( \frac{1 - \frac{E_2}{E_3}}{1 + \frac{E_2}{E_3}} \right) = 3\mu_g \left( \frac{1 - \frac{E_5}{E_4}}{1 + \frac{E_5}{E_4}} \right) \quad (55)$$

$$\mu_a = \mu_g \left( \frac{1 - \frac{E_1}{E_3}}{1 + \frac{1}{3} \frac{E_1}{E_3}} \right) = \mu_g \left( \frac{1 - \frac{E_6}{E_4}}{1 + \frac{1}{3} \frac{E_6}{E_4}} \right) \quad (56)$$

$$\mu_a = \mu_g \left( \frac{-1 + \frac{E_1}{E_2}}{-1 + \frac{1}{3} \frac{E_1}{E_2}} \right) = \mu_g \left( \frac{-1 + \frac{E_6}{E_5}}{-1 + \frac{1}{3} \frac{E_6}{E_5}} \right) . \quad (57)$$

Aus dem Mittelwert der einzelnen Ergebnisse ergibt sich als Ergebnis für das magnetische Moment:

$$\mu_a = (-0,157 \pm 0,004) \mu_N . \quad (58)$$

#### 4.6.4 Magnetische Feldstärke

Aus den Gleichungen 46 bis 48 ergeben sich folgende Formeln für die magnetische Feldstärke am Kernort:

$$B_1 = \frac{E_1}{\mu_a - \mu_g} = (33,1 \pm 0,6) \text{ T} \quad B_6 = \frac{E_6}{\mu_g - \mu_a} = (32,7 \pm 0,6) \text{ T} \quad (59)$$

$$B_2 = \frac{E_2}{\frac{1}{3}\mu_a - \mu_g} = (33,1 \pm 0,5) \text{ T} \quad B_5 = \frac{E_5}{\mu_g - \frac{1}{3}\mu_a} = (33,1 \pm 0,7) \text{ T} \quad (60)$$

$$B_3 = \frac{E_3}{\frac{1}{3}\mu_a + \mu_g} = (26 \pm 3) \text{ T} \quad B_4 = \frac{E_4}{\mu_g + \frac{1}{3}\mu_a} = (33 \pm 3) \text{ T} . \quad (61)$$

Aus dem Mittelwert ergibt sich hiermit folgendes Magnetfeld am Kernort:

$$B = (32,9 \pm 0,3) \text{ T} . \quad (62)$$



## 5 Zusammenfassung und Diskussion

### 5.1 Eichung und Korrekturmessungen

Zur Energieeichung des MCA wurden bekannte Peaks von Barium, Silber, Molybdän, Rubidium und Terbium vermessen. Daraus ergibt sich der Umrechnungsfaktor

$$b = (0,0763 \pm 0,0017) \frac{\text{keV}}{\text{Channel}} , \quad (63)$$

womit sich die erwartete Position des 14,4 keV-Peaks zu

$$\text{Pos}_{14,4} = 195 \pm 7 \quad (64)$$

ergibt.

Des Weiteren wurde der Einfluss des Compton-Untergrundes mit Hilfe der Messung von Transmission durch Aluminiumplatten unterschiedlicher Dicke sowie die Abschwächung durch das Plexiglas der Absorberhalterung untersucht. Dabei wurde folgende Korrekturformel für die Zählrate ermittelt:

$$r = k_{\text{Plexi}} \cdot r_{\text{Mess}} - r_{\text{Untergrund}} , \quad (65)$$

mit

$$k_{\text{Plexi}} = 1,243 \pm 0,010 , \quad (66)$$

$$r_{\text{Untergrund}} = (18,5 \pm 0,3) \text{ s}^{-1} . \quad (67)$$

### 5.2 Einlinienabsorber

Für den Einlinienabsorber wurde eine Isomerieverschiebung von

$$E_{\text{iso}} = (9,7 \pm 0,9) \text{ neV} \quad (68)$$

gemessen.

Die Linienbreite und Lebensdauer wurden einerseits durch den Lorentzanteil der Voigtfunktion und andererseits mit Hilfe der effektiven Absorberdicke  $T_A = 6,1 \pm 0,7$  berechnet und sind in Tabelle 7 aufgetragen im Vergleich zu den Literaturwerten aufgetragen.

	$\Gamma / \text{neV}$	$\tau / \text{ns}$	$T_{\frac{1}{2}} / \text{ns}$	$\frac{\chi^2}{\text{ndf}}$
Voigt-Fit	$6 \pm 4$	$110 \pm 70$	$80 \pm 50$	0,55
effektive Absorberdicke	$4,1 \pm 0,3$	$162 \pm 12$	$112 \pm 9$	0,56
Literaturwerte [4]	4,7	141	98	

**Tabelle 7:** Messergebnisse für den Einlinienabsorber

Dabei stellen wir fest, dass die aus der Lorentzbreite des Voigtfit bestimmten Werte innerhalb eines  $\sigma$  mit den Literaturwerten übereinstimmen, der Fehler allerdings über 50% des Wertes beträgt. Diese hohen Fehler liegen darin begründet, dass der Lorentzanteil des Voigtfits mit einem hohen Fehler

bestimmt wurde. Die Werte, die wir durch Berechnung der effektiven Absorberdicke erhalten haben, sind deutlich genauer und stimmen innerhalb von  $2\sigma$  mit den Literaturwerten überein.

Der Debye-Waller-Faktor der Quelle berechnet sich aus gegebenen Werten aus der Literatur und den Messwerten zu

$$f_Q = 0,388 \pm 0,014 . \quad (69)$$

Der dem im Labor ausliegenden Datenblatt entnommene Wert für den Debye-Waller-Faktor beträgt 0,77. Die starke Abweichung kann in der Temperaturabhängigkeit des Debye-Waller-Faktors begründet liegen. Da im Datenblatt keine Temperatur angegeben war, gilt dieser Wert möglicherweise nicht für die Temperatur, bei welcher der Versuch durchgeführt wurde. Des Weiteren wurde möglicherweise der Untergrund zu niedrig bestimmt.

### 5.3 Sechslinienabsorber

Für den Sechslinienabsorber wurde ebenfalls die Isomerieverschiebung gemessen. Außerdem wurden das magnetische Moment des angeregten Zustands sowie das Magnetfeld am Kernort berechnet. Die Ergebnisse sind in Tabelle 8 zu finden.

Größe	Messwert	Literaturwert
$E_{\text{iso}} / \text{neV}$	$6,7 \pm 0,9$	$5,3^3$
$\mu_a / \mu_N$	$-0,157 \pm 0,004$	$-0,15531 \pm 0,00004$ [9]
$B / \text{T}$	$32,9 \pm 0,3$	$33,3 \pm 0,1$ [9]

**Tabelle 8:** Messergebnisse für den Sechslinienabsorber

Hierbei fällt auf, dass der Wert für die Isomerieverschiebung nur innerhalb von  $2\sigma$  mit dem Literaturwert übereinstimmt, die Ergebnisse für das magnetische Moment und das Magnetfeld am Kernort jedoch innerhalb eines  $\sigma$  mit dem jeweiligen Literaturwert übereinstimmen.

Dies lässt vermuten, dass das Spektrum möglicherweise zusätzlich zur Isomerieverschiebung verschoben ist. Die Ergebnisse sind jedoch insgesamt zufriedenstellend.

---

<sup>3</sup>Dieser Wert wurde dem Datenblatt aus dem Labor entnommen

## 6 Anhang

### 6.1 Laborbuch

Laborbuch Möpbauer

a) Beobachtung d. Signals am Oszilloskop

Bild 1: unipol. / bipolar Ausgang MA

Bild 2: Nach SCA / Gate

b) Eichung  $j < 320$  s Schaltung ①

c) Einstellung SCA: e.l.: 0,7 u.l.: 1,3  
Schaltung ②

d) Aufbau eig. Schaltung mit Hilfe d. Oszilloskops: Bild 3

Amplifier: Gain: 0,6 Coarse Gain: 50

Shaping Time 2  $\mu$ s Delay 3  $\mu$ s

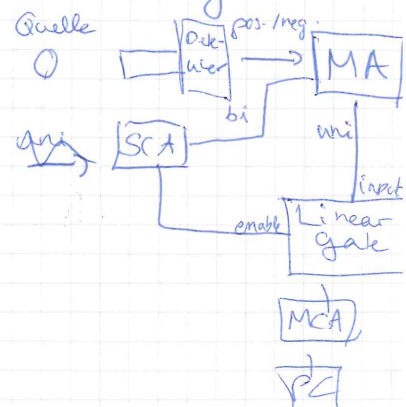
e) Messung Abschirmung Aluminium  $j < 600$  s  
0-12 mm in 0,5 mm Schritten

f) Messung Abschirmung Plexiglas 600 s

Schaltung ①



Schaltung ②



g) Hauptmessung 1-Linien-Absorber  
- 6 mm/s - + 6 mm/s 0,05-Schritte je 600s

h) Hauptmessung 6-Linien-Absorber  
- 8 mm/s - + 8 mm/s 0,1-Schritte je 600s

## 6.2 Spektren

## 6.3 Sourcecode

### Eichung.R

```
setwd("C:\\Users\\Saskia\\Documents\\Physik\\FP\\fp2-moessbauer\\scripts")
source("readFiles.R")
source("functions.R")
source("gausfit1.R")
source("supergaus.R")

par(mar=c(5,5,1,1))
Tb = readTKA("eichung/TbEichung")[[1]]
Rb = readTKA("eichung/RbEichung")[[1]]
Mo = readTKA("eichung/MoEichung")[[1]]
Cu = readTKA("eichung/CuEichung")[[1]]
Ag = readTKA("eichung/AgEichung")[[1]]
Ba = readTKA("eichung/BaEichung")[[1]]

Tb_times=Tb[1:2]
Rb_times=Rb[1:2]
Mo_times=Mo[1:2]
Cu_times=Cu[1:2]
Ag_times=Ag[1:2]
Ba_times=Ba[1:2]

Tb[1:2]=0
Rb[1:2]=0
Mo[1:2]=0
Cu[1:2]=0
Ag[1:2]=0
Ba[1:2]=0

x=c(1:2048)

drawCI(x, Tb, sqrt(Tb), scol="black", xlab = "channel", ylab = "Counts")
bereich=c(450, 1000)
fit5=supergausfit(data.frame(x=x, y=Tb, sy=sqrt(Tb)), Bereich, mu=593, sig0=55, N0=22500, N01=2500000,
  sig01=80, mu01=740, weighted=TRUE)
plotsupergaus(fit5, Bereich, col="red")
printsupergausdata(fit5)
drawCI(x, Rb, sqrt(Rb), scol="black", xlab = "channel", ylab = "Counts")
fit4=gausfit(data.frame(x=x, y=Rb, sy=sqrt(Rb)), c(100,218), weighted=TRUE)
plotgaus(fit4, c(100,218), col="red")
drawCI(x, Mo, sqrt(Mo), scol="black", xlab = "channel", ylab = "Counts")
fit3=gausfit(data.frame(x=x, y=Mo, sy=sqrt(Mo)), c(135,290), weighted=TRUE)
plotgaus(fit3, c(135,290), col="red")

#drawCI(x, Cu, sqrt(Cu), scol="black")

drawCI(x, Ag, sqrt(Ag), scol="black", xlab = "channel", ylab = "Counts")
fit2=gausfit(data.frame(x=x, y=Ag, sy=sqrt(Ag)), c(200,475), weighted=TRUE)
plotgaus(fit2, c(200,475), col="red")

drawCI(x, Ba, sqrt(Ba), scol="black", xlab = "channel", ylab = "Counts")
fit1=gausfit(data.frame(x=x, y=Ba, sy=sqrt(Ba)), c(300,550), weighted=TRUE)
plotgaus(fit1, c(300,550), col="red")
```

### LinFitEichung.R

```
library("Hmisc")
source("functions.R")

x <- c(13.37, 17.44, 22.10, 32.06, 44.23)
y1 <- c(178.96, 237.79, 301.68, 418.56, 588.46)

erry1 <- c(11, 13, 16, 19, 24)
```

```
drawCllim(x, y1, erry1, xlab = "Energy/keV", ylab = "channel")
```

```
grid()
```

```
text(x=40, y=100, "y=(13.1+-0.3)*channel/keV*x+(7+-6)*channel", cex=0.8)
```

```
fm1 <- lm(y1 ~ x, weights=1/erry1^2)
```

```
summary(fm1)
```

```
abline(fm1, col = "red")
```

```
*****
```

## Compton-Untergrund.R

```
source("readFiles.R")
```

```
source("functions.R")
```

```
source("expfit.R")
```

```
par(mar=c(5,5,1,1))
```

```
alu_names = c("0-0", "0-5", "1-0", "1-5", "2-0", "2-5", "3-0", "3-5", "4-0", "4-5", "5-0", "5-5", "6-0", "6-5", "7-0", "7-5", "8-0", "8-5", "9-0", "9-5", "10-0", "10-5", "11-0", "11-5", "12-0")
```

```
n = length(alu_names)
```

```
alu = array(dim=c(2048,n))
```

```
#alu[,1] = readTKA("aluminium/Alu_0-0mm")[[1]]
```

```
m=c(0,1,1,1,1,1,1,2,1,2,2,2,2,2,2,3,2,3,3,3,3,3,4,4)
```

```
s_di=0.1
```

```
s_d=sqrt(m)*s_di
```

```
for(i in c(1:n)){
```

```
  alu[,i] = readTKA(paste("aluminium/Alu_", alu_names[i], "mm", sep=""))[[1]]
```

```
}
```

```
alu_times=alu[1:2,]
```

```
alu[1:2,]=0
```

```
alu_rates=alu/alu_times[1,]
```

```
alu_sum2=c()
```

```
alu_sum=c()
```

```
for(i in c(1:n)){
```

```
  alu_sum[i]=sum(alu_rates[,i])
```

```
  alu_sum2[i]=sum(alu[,i])
```

```
}
```

```
alu_err=alu_sum/sqrt(alu_sum2)
```

```
x=c(1:n)*0.5-0.5
```

```
bereich=c(0,12)
```

```
drawCllx(x,y=alu_sum,sy=alu_err,sx=s_d,scol="black",xlab=expression(d/mm),ylab=expression(r/s^-1))
```

```
fit=dexpfit(data.frame(x=x,y=alu_sum,sy=alu_err),bereich,weighted=TRUE)
```

```
plotdexp(fit,c(0,12))
```

```
printdexpdata(fit)
```

```
C<-fit["C","Estimate"]
```

```
sC<-C*sqrt((fit["C","Std. Error"]/fit["C","Estimate"])^2)
```

```
untergrund=roundfunc(c(C,sC))
```

```
cat("\nUntergrund:\nr_U= (")
```

```
cat(paste(untergrund[1], " +- ", untergrund[2], ") 1/s", sep=""))
```

## Plexiglas.R

```
source("readFiles.R")
```

```
source("functions.R")
```

```
source("expfit.R")
```

```
par(mar=c(5,5,1,1))
```

```

empty = readTKA("plexiglas/leer")[[1]]
plexi = readTKA("plexiglas/Plexi")[[1]]

empty_times=empty[1:2]
plexi_times=plexi[1:2]

empty[1:2]=0
plexi[1:2]=0

rate_e=sum(empty)/empty_times[1]
rate_p=sum(plexi)/plexi_times[1]
s_e=rate_e/sqrt(sum(empty))
s_p=rate_p/sqrt(sum(plexi))

ratio=roundfunc(c(rate_p/rate_e,rate_p/rate_e*sqrt((s_e/rate_e)^2+(s_p/rate_p)^2))
ratio2=roundfunc(c(rate_e/rate_p,rate_e/rate_p*sqrt((s_e/rate_e)^2+(s_p/rate_p)^2))

cat(paste("Verhaeltnis r_p/r_e=",ratio[1]," +- ",ratio[2],sep=""))
cat(paste("\nnnnnnnnk_p=r_e/r_p=",ratio2[1]," +- ",ratio2[2],sep=""))

```

## Absorberdicke.R

```

fA=0.8
beta=0.022
dA=25*10^-6#n
f=0.7
sf=0.05
Na=6.022*10^23#1/mol
mmol=55.85#g/mol
rho=7.874*10^6#g/m^3
alpha=9.1#Conny#8.44#Moritz
salpha=0.8
I=1/2
Istern=3/2

c=2.998*10^8#m/s
E0=14.412*10^3#eV
hquer=6.582*10^(-16)#eVs
lquer=c*hquer/E0
sigma=2*pi*lquer^2*(2*Istern+1)/(2*I+1)*1/(1+alpha)
nA=f*Na/mmol*rho
TA=c()
TA[1]=fA*nA*beta*sigma*dA
TA[2]=TA[1]*sqrt((sf/f)^2+(salpha/alpha)^2)

```

## Debye-Waller.R

```

library(Bessel)
source('functions.R')

Zinf=c(12.84,0.08)
Zmu=c(9.06,0.10)
Z=c()
Z[1]=(Zinf[1]-Zmu[1])/Zinf[1]
Z[2]=Zmu[1]/Zinf[1]*sqrt((Zinf[2]/Zinf[1])^2+(Zmu[2]/Zmu[1])^2)
i=complex(imaginary=1)
y=c()
y[1]=1/Re(1-exp(-TA[1]/2)*BesselJ(i*TA[1]/2,0))
y[2]=Re(TA[2]/2*exp(-TA[1]/2)*(-BesselJ(i*TA[1]/2,0)+i*BesselJ(i*TA[1]/2,-1))/(1-exp(-TA[1]/2)*
BesselJ(i*TA[1]/2,0))^2)
fQ=multiply(Z,y)
#fQ[2]=fQ[1]*sqrt((Z[2]/Z[1])^2)

```

## Einlinien.R

```

library(RcppFaddeeva)#install.packages('RcppFaddeeva')

source("readFiles.R")
source("functions.R")
source("expfit.R")
source("lorentzfit.R")
source("gausfit1.R")

```

```

source("voigtfit.R")

par(mar=c(5,5,1,1))

data = readTXT(paste("einlinien/einlinien2",sep=""))

data[[2]]=data[[2]]*10^-3

rate=data[[3]]/data[[2]]
sdata=sqrt(data[[3]])
srate=rate/sdata

untergrund=18.5
suntergrund=0.3
rate=rate-untergrund
srate=sqrt(srate^2+suntergrund^2)

k=1.243
sk=0.010
srate=rate*k*sqrt((srate/rate)^2+(sk/k)^2)
rate=rate*k

x=data[[1]]

drawCI(x,rate,srate,xlab=expression(v / mms^-1),ylab=expression(r / s^-1))
fit=lorentzfit(data.frame(x=x,y=rate,sy=srate),neg=TRUE,weighted=TRUE)
plotlorentz(fit,c(-6,6),lwd=1.5,lty=3)
fit2=gausfit(data.frame(x=x,y=rate,sy=srate),c(-6,6),weighted=TRUE,N0=-5)
plotgaus(fit2,c(-6,6),col="blue",lwd=1.5,lty=2)
A0=fit2['N','Estimate']
C0=fit2['C','Estimate']
mu0=fit2['mu','Estimate']
sigma0=fit2['sig','Estimate']
gamma0=fit2['omega','Estimate']
fit3=voigtfit(data.frame(x=x,y=rate,sy=srate),A0,C0,mu0,sigma0,gamma0,weighted=TRUE)
plotvoigt(fit3,c(-6,6),col="green",lwd=2.5,lty=1)

legend(2,10,c("Lorentzfit","Gaussfit","Voigtfit"),col=c("red","blue","green"),lty=c(3,2,1),lwd=c(1.5,1.5,2.5))

sigma=fit3['sigma','Estimate']
ssigma=fit3['sigma','Std. Error']
gamma=fit3['gamma','Estimate']
sgamma=fit3['gamma','Std. Error']

fg=2*sigma*sqrt(2*log(2,exp(1)))
fl=gamma

fv=0.5346*fl+sqrt(0.2166*fl^2+fg^2)

sfg=2*ssigma*sqrt(2*log(2,exp(1)))
sfl=sgamma
sfv=fv*sqrt(0.02^2+(sfg/fg)^2+(sfl/fl)^2)

sigmaG=fit2['sig','Estimate']
ssigmaG=fit2['sig','Std. Error']
gammaL=fit2['tau','Estimate']
sgammaL=fit2['tau','Std. Error']

fG=2*sigmaG*sqrt(2*log(2,exp(1)))
fL=gammaL

sfG=2*ssigmaG*sqrt(2*log(2,exp(1)))
sfL=sgammaL
fv2=0.5346*fL+sqrt(0.2166*fL^2+fG^2)
sfv2=fv2*sqrt(0.02^2+(sfG/fG)^2+(sfL/fL)^2)

GAMMA0=roundfunc(toenergy(c(fL,sfL)))

W=c(1.78,0.06)
GAMMA=c()
GAMMA[1]=GAMMA0[1]/4/W[1]
GAMMA[2]=GAMMA[1]*sqrt((GAMMA0[2]/GAMMA0[1])^2+(W[2]/W[1])^2)

```



```

#GAMMA=roundfunc(toenergy(c(fl, sfl)))

hquer=6.582*10^(-16)#eV
tau=c()
tau[1]=hquer/GAMMA[1]
tau[2]=tau[1]*GAMMA[2]/GAMMA[1]

Thalb=tau*log(2,exp(1))

#drawClim(x,rate,srate,xlab=expression(v / mms^-1),ylab=expression(r / s^-1),xlim=c(-1,1),ylim=c
(8.5,13))
#plotlorentz(fit,c(-6,6),lwd=1.5,lty=3)
#plotgaus(fit2,c(-6,6),col="blue",lwd=1.5,lty=2)
#plotvoigt(fit3,c(-6,6),col="green",lwd=2.5,lty=1)

mu=c(fit3['mu','Estimate'],fit3['mu','Std. Error'])
isomerie=toenergy(mu)

```

## Sechslinien.R

```

library(RcppFaddeeva)#install.packages('RcppFaddeeva')

source("readFiles.R")
source("functions.R")
source("expfit.R")
source("lorentzfit.R")
source("gausfit1.R")
source("voigtfit.R")

par(mar=c(5,5,1,1))

data = readTXT(paste("sechslinien/sechslinien0",sep=""))

data[[2]]=data[[2]]*10^-3
rate=data[[3]]/data[[2]]
sdata=sqrt(data[[3]])
srate=rate/sdata

untergrund=18.5
suntergrund=0.3
rate=rate-untergrund
srate=sqrt(srate^2+suntergrund^2)

k=1.243
sk=0.010
srate=rate*k*sqrt((srate/rate)^2+(sk/k)^2)
rate=rate*k

x=data[[1]]

drawCI(x,rate,srate,xlab=expression(v / mms^-1),ylab=expression(r / s^-1))
fit=sixlorentz(data.frame(x=x,y=rate,sy=srate),neg=TRUE,weighted=TRUE
,-5.23,-2.97,-0.51,0.92,3.24,5.43)
plotsixlorentz(fit,c(-8.05,8.05),lwd=1.5,lty=3)
fit2=sixgaus(data.frame(x=x,y=rate,sy=srate),c(-8.05,8.05),weighted=TRUE,-5,-3,-0.5,1,3,5)
plotsix(fit2,c(-8.05,8.05),col="blue",lwd=1.5,lty=2)
A01=fit2['N1','Estimate']
A02=fit2['N2','Estimate']
A03=fit2['N3','Estimate']
A04=fit2['N4','Estimate']
A05=fit2['N5','Estimate']
A06=fit2['N6','Estimate']
C0=28.6#fit2['C','Estimate']
mu01=fit2['mu1','Estimate']
mu02=fit2['mu2','Estimate']
mu03=fit2['mu3','Estimate']
mu04=fit2['mu4','Estimate']
mu05=fit2['mu5','Estimate']
mu06=fit2['mu6','Estimate']
sigma01=fit2['sig1','Estimate']
sigma02=fit2['sig2','Estimate']
sigma03=fit2['sig3','Estimate']

```

```

sigma04=fit2 [ 'sig4', 'Estimate' ]
sigma05=fit2 [ 'sig5', 'Estimate' ]
sigma06=fit2 [ 'sig6', 'Estimate' ]
gamma=0.1
gamma01=fit [ 'tau1', 'Estimate' ] #gamma#fit [ 'omega1', 'Estimate' ]
gamma02=fit [ 'tau2', 'Estimate' ] #gamma#fit [ 'omega2', 'Estimate' ]
gamma03=fit [ 'tau3', 'Estimate' ] #gamma#fit [ 'omega3', 'Estimate' ]
gamma04=fit [ 'tau4', 'Estimate' ] #gamma#fit [ 'omega4', 'Estimate' ]
gamma05=fit [ 'tau5', 'Estimate' ] #gamma#fit [ 'omega5', 'Estimate' ]
gamma06=fit [ 'tau6', 'Estimate' ] #gamma#fit [ 'omega6', 'Estimate' ]
#gamma01=gamma#fit [ 'omega1', 'Estimate' ]
#gamma02=gamma#fit [ 'omega2', 'Estimate' ]
#gamma03=gamma#fit [ 'omega3', 'Estimate' ]
#gamma04=gamma#fit [ 'omega4', 'Estimate' ]
#gamma05=gamma#fit [ 'omega5', 'Estimate' ]
#gamma06=gamma#fit [ 'omega6', 'Estimate' ]
fit3=sixvoigt (data.frame(x=x,y=rate,sy=srate),A01,A02,A03,A04,A05,A06,C0,mu01,mu02,mu03,mu04,mu05,
mu06,sigma01,sigma02,sigma03,sigma04,sigma05,sigma06,gamma01,gamma02,gamma03,gamma04,gamma05,
gamma06,weighted=TRUE)
plotsixvoigt (fit3,c(-8.05,8.05),col="green",lwd=2.5,lty=1)

legend(0,10.8,c("Lorentzfit","Gaussfit","Voigtfit"),col=c("red","blue","green"),lty=c(3,2,1),lwd=c
(1.5,1.5,2.5))

mu1=c(fit3 [ 'mu1', 'Estimate' ], fit3 [ 'mu1', 'Std. Error' ])
mu2=c(fit3 [ 'mu2', 'Estimate' ], fit3 [ 'mu2', 'Std. Error' ])
mu3=c(fit3 [ 'mu3', 'Estimate' ], fit3 [ 'mu3', 'Std. Error' ])
mu4=c(fit3 [ 'mu4', 'Estimate' ], fit3 [ 'mu4', 'Std. Error' ])
mu5=c(fit3 [ 'mu5', 'Estimate' ], fit3 [ 'mu5', 'Std. Error' ])
mu6=c(fit3 [ 'mu6', 'Estimate' ], fit3 [ 'mu6', 'Std. Error' ])
isomerie1=c()
isomerie2=c()
isomerie3=c()
isomerie1[1]=toenergy((mu4[1]+mu3[1])/2)
isomerie2[1]=toenergy((mu5[1]+mu2[1])/2)
isomerie3[1]=toenergy((mu6[1]+mu1[1])/2)
isomerie1[2]=toenergy(sqrt(mu4[2]^2+mu3[2]^2)/2)
isomerie2[2]=toenergy(sqrt(mu5[2]^2+mu2[2]^2)/2)
isomerie3[2]=toenergy(sqrt(mu6[2]^2+mu1[2]^2)/2)

isomerie=c()
isomerie[1]=(isomerie1[1]/isomerie1[2]^2+isomerie2[1]/isomerie2[2]^2+isomerie3[1]/isomerie3[2]^2)/
(1/isomerie1[2]^2+1/isomerie2[2]^2+1/isomerie3[2]^2)
isomerie[2]=sqrt(1/(1/isomerie1[2]^2+1/isomerie2[2]^2+1/isomerie3[2]^2))

E1=roundfunc(substract(toenergy(mu1),isomerie))
E2=roundfunc(substract(toenergy(mu2),isomerie))
E3=roundfunc(substract(toenergy(mu3),isomerie))
E4=roundfunc(substract(toenergy(mu4),isomerie))
E5=roundfunc(substract(toenergy(mu5),isomerie))
E6=roundfunc(substract(toenergy(mu6),isomerie))

E23=divide(E2,E3)
E13=divide(E1,E3)
E12=divide(E1,E2)
E54=divide(E5,E4)
E64=divide(E6,E4)
E65=divide(E6,E5)

mug=c(0.090604,0.000009)#muN
one=c(1,0)
mua23=multiply(3*one,multiply(mug,divide(substract(one,E23),add(one,E23))))
mua54=multiply(3*one,multiply(mug,divide(substract(one,E54),add(one,E54))))

mua12=multiply(one,multiply(mug,divide(substract(E12,one),substract(E12/3,one))))
mua65=multiply(one,multiply(mug,divide(substract(E65,one),substract(E65/3,one))))

mua13=multiply(one,multiply(mug,divide(substract(one,E13),add(one,1/3*E13))))
mua64=multiply(one,multiply(mug,divide(substract(one,E64),add(one,1/3*E64))))

mua1=roundfunc(mean(mua23,mua54))
mua2=roundfunc(mean(mua13,mua64))

```

```

mua3=roundfunc(mean(mua12,mua65))

mua=roundfunc(mean3(mua1,mua2,mua3))

hquer=6.582*10^(-16)
e=1.602*10^(-19)
mproton=1.672*10^(-27)
mun=e*hquer/(2*mproton)

B1=divide(E1,multiply(subtract(mua,mug),mun*one))
B6=divide(E6,multiply(subtract(mug,mua),mun*one))
B2=divide(E2,multiply(subtract(1/3*mua,mug),mun*one))
B5=divide(E5,multiply(subtract(mug,1/3*mua),mun*one))
B3=subtract(c(0,0),divide(E3,multiply(add(1/3*mua,mug),mun*one)))
B4=divide(E4,multiply(add(mug,1/3*mua),mun*one))

B16=roundfunc(mean(B1,B6))
B25=roundfunc(mean(B2,B5))
B34=roundfunc(mean(B3,B4))

B=roundfunc(mean3(B16,B25,B34))

```

## Expfit.R

```

expfit <- function(input, bereich, weighted=FALSE){ #— Fitten der Exponentialfunktion

  theexponential <- y ~ A + C *exp(lambda*x)

  #daten=input[bereich[1]:bereich[2],]
  daten=subset(input,x>=bereich[1] & x <= bereich[2])

  #— Startwerte
  ymin=min(daten$y)
  xmin=daten$x[which.min(daten$y)]
  ymax=max(daten$y)
  xmax=daten$x[which.max(daten$y)]
  lambda_est=(log(ymin)-log(ymax))/(xmin-xmax)
  C_est=ymin*exp(-lambda_est*xmin)
  A_est=0
  err=daten$y
  startvalues=list(C=C_est,A=A_est,lambda=lambda_est)
  #— Startwerte

  #— Durchführen des Fits
  nlc<-nls.control(maxiter=5000)
  if(weighted)
    fit = nls(theexponential, daten, weights=1/err^2, start=startvalues, control=nlc)
  else
    fit = nls(theexponential, daten, start=startvalues, control=nlc)

  chiquadratndf=sum(summary(fit)[[2]]^2)/summary(fit)[[4]][[2]]
  fitdata=rbind(summary(fit)$parameters,c(chiquadratndf,0,0,0))

  return(fitdata)
}

dexpfit <- function(input, bereich, weighted=FALSE){ #— Fitten der Exponentialfunktion

  theexponential <- y ~ C *exp(lambda*x) + D*exp(mu*x)

  #daten=input[bereich[1]:bereich[2],]
  daten=subset(input,x>=bereich[1] & x <= bereich[2])

  #— Startwerte
  ymin=min(daten$y)
  xmin=daten$x[which.min(daten$y)]
  ymax=max(daten$y)
  xmax=daten$x[which.max(daten$y)]
  lambda_est=(log(ymin)-log(ymax))/(xmin-xmax)
  C_est=ymin*exp(-lambda_est*xmin)
  D_est=C_est
  mu_est=-1

```

```

A_est=0
err=daten$sy
startvalues=list(C=C_est, lambda=lambda_est, D=D_est, mu=mu_est)
#---Startwerte

#plot (function(x){A_est + C_est *exp(lambda_est*x) + D_est*exp(mu_est*x)}, bereich[1], bereich[2],
      add=TRUE, col="green")

#---Durchführen des Fits
nls<-nls.control(maxiter=5000)
if(weighted)
  fit = nls(theexponential, daten, weights=1/err^2, start=startvalues, control=nls)
else
  fit = nls(theexponential, daten, start=startvalues, control=nls)

chiquadratndf=sum(summary(fit)[[2]]^2)/summary(fit)[[4]][[2]]
fitdata=rbind(summary(fit)$parameters, c(chiquadratndf, 0, 0, 0))

return(fitdata)
}

plotexp <- function(fitdata, bereich){ #--- Plotten der gefitteten Exponentialfunktion in vorhandenen
  Graph

  lambda<-fitdata["lambda", "Estimate"]
  C<-fitdata["C", "Estimate"]
  A<-fitdata["A", "Estimate"]

  plot (function(x){A + C *exp(lambda*x)}, bereich[1], bereich[2], add=TRUE, col="red")
}

plotdexp <- function(fitdata, bereich){ #--- Plotten der gefitteten Exponentialfunktion in
  vorhandenen Graph

  lambda<-fitdata["lambda", "Estimate"]
  mu<-fitdata["mu", "Estimate"]
  C<-fitdata["C", "Estimate"]
  D<-fitdata["D", "Estimate"]
  #A<-fitdata["A", "Estimate"]

  plot (function(x){C *exp(lambda*x) + D*exp(mu*x)}, bereich[1], bereich[2], add=TRUE, col="red")
}

printexpdata <- function(fitdata, title="", factor=1, error=0){ #--- Ausgabe der Exponentialfit-Daten

  lambda<-fitdata["lambda", "Estimate"]/factor
  slambda<-lambda*sqrt((fitdata["lambda", "Std. Error"]/fitdata["lambda", "Estimate"])^2+(error/factor
    )^2)

  cat(title)
  cat("\n")

  cat(" Zerfallskonstante====")
  cat(lambda)
  cat("+")
  cat(slambda)
  cat("\n")

  tau=1/lambda
  stau=1/lambda*slambda/lambda

  cat(" Mittlere Lebensdauer====")
  cat(tau)
  cat("+")
  cat(stau)
  cat("\n")
}

```

```

T=log(2)*tau
sT=log(2)*stau

cat("  Halbwertszeit=====")
cat(T)
cat("+-")
cat(sT)
cat("\n")
}

printexpdata <- function(fitdata , title="", factor=1,error=0){ #--- Ausgabe der Exponentialfit-Daten

  lambda<-fitdata["lambda","Estimate"]/factor
  slambda<-lambda*sqrt((fitdata["lambda","Std. Error"]/fitdata["lambda","Estimate"])^2+(error/factor)^2)
  C<-fitdata["C","Estimate"]/factor
  sC<-C*sqrt((fitdata["C","Std. Error"]/fitdata["C","Estimate"])^2+(error/factor)^2)
  D<-fitdata["D","Estimate"]/factor
  sD<-D*sqrt((fitdata["D","Std. Error"]/fitdata["D","Estimate"])^2+(error/factor)^2)

  cat(title)
  cat("\n")
  cat("  Erste Funktion:")
  cat("\n")

  cat("  Faktor=====")
  cat(C)
  cat("+-")
  cat(sC)
  cat("\n")

  cat("  Zerfallskonstante=====")
  cat(lambda)
  cat("+-")
  cat(slamba)
  cat("\n")

  tau=1/lambda
  stau=1/lambda*slambda/lambda

  cat("  Mittlere Lebensdauer=====")
  cat(tau)
  cat("+-")
  cat(stau)
  cat("\n")

  T=log(2)*tau
  sT=log(2)*stau

  cat("  Halbwertszeit=====")
  cat(T)
  cat("+-")
  cat(sT)
  cat("\n")

  cat("\n")
  cat("  Zweite Funktion:")
  cat("\n")

  cat("  Faktor=====")
  cat(D)
  cat("+-")
  cat(sD)
  cat("\n")

  lambda<-fitdata["mu","Estimate"]/factor
  slambda<-lambda*sqrt((fitdata["mu","Std. Error"]/fitdata["mu","Estimate"])^2+(error/factor)^2)

  cat("  Zerfallskonstante=====")
  cat(lambda)

```

```

cat("+-")
cat(slambda)
cat("\n")

tau=1/lambda
stau=1/lambda*slambda/lambda

cat("░Mittlere░Lebensdauer░=")
cat(tau)
cat("+-")
cat(stau)
cat("\n")

T=log(2)*tau
sT=log(2)*stau

cat("░Halbwertszeit░░░░░░░░░░=")
cat(T)
cat("+-")
cat(sT)
cat("\n")

}

functions.R

drawCI <- function(x,y,sy,xlab="x",ylab="y",col="black",scol="darkgrey"){
  plot(x,y,pch=4,cex=0.6,bty="l",col=col,xlab=xlab,ylab=ylab)
  arrows(x,y,x,y-sy,cex=0.6,pch=4,bty="l",col=scol,length=0.05,angle=90)
  arrows(x,y,x,y+sy,cex=0.6,pch=4,bty="l",col=scol,length=0.05,angle=90)
  points(x,y,cex=0.6,pch=4,col=col)
  grid()
}

drawCILim <- function(x,y,sy,xlab="x",ylab="y",col="black",scol="darkgrey",xlim=c(0,60),ylim=c
(0,800)){
  plot(x,y,pch=4,cex=0.6,bty="l",col=col,xlab=xlab,ylab=ylab,xlim=xlim,ylim=ylim)
  arrows(x,y,x,y-sy,cex=0.6,pch=4,bty="l",col=scol,length=0.05,angle=90)
  arrows(x,y,x,y+sy,cex=0.6,pch=4,bty="l",col=scol,length=0.05,angle=90)
  points(x,y,cex=0.6,pch=4,col=col)
  grid()
}

drawCIx <- function(x,y,sy,sx,xlab="x",ylab="y",col="black",scol="darkgrey"){
  plot(x,y,pch=4,cex=0.6,bty="l",col=col,xlab=xlab,ylab=ylab)
  arrows(x,y,x,y-sy,cex=0.6,pch=4,bty="l",col=scol,length=0.05,angle=90)
  arrows(x,y,x,y+sy,cex=0.6,pch=4,bty="l",col=scol,length=0.05,angle=90)
  arrows(x,y,x-sx,y,cex=0.6,pch=4,bty="l",col=scol,length=0.05,angle=90)
  arrows(x,y,x+sx,y,cex=0.6,pch=4,bty="l",col=scol,length=0.05,angle=90)
  points(x,y,cex=0.6,pch=4,col=col)
  grid()
}

draw <- function(x,y,xlab="x",ylab="y",col="black",scol="darkgrey"){
  plot(x,y,pch=4,cex=0.6,bty="l",col=col,xlab=xlab,ylab=ylab)
  grid()
}

roundfunc <- function(vals){
  x=vals[1]
  xerr=vals[2]
  #print(vals)
  n=0
  for(i in -20:20){
    a=round(xerr,i)*10^i
    #print(a)
    if(a==1){
      n=i+1
      return(c(round(x,n),round(xerr,n)))
    }
    if(a==2){
      if(xerr*10^i<1.95){
        n=i+1

```

```

    } else {
      n=i
    }
    return(c(round(x,n),round(xerr,n)))
  }
  if(a>2){
    n=i
    return(c(round(x,n),round(xerr,n)))
  }
}
return( vals )

}

toenergy <- function( val ){
  c=2.998*10^8*10^3#mm
  E0=14.412*10^3

  return(E0*val/c)
}

add <- function( vec1 , vec2 ){
  res=c()
  res[1]=vec1[1]+vec2[1]
  res[2]=sqrt( vec1[2]^2+vec2[2]^2 )
  return( res )
}
subtract <- function( vec1 , vec2 ){
  res=c()
  res[1]=vec1[1]-vec2[1]
  res[2]=sqrt( vec1[2]^2+vec2[2]^2 )
  return( res )
}
divide <- function( vec1 , vec2 ){
  res=c()
  res[1]=vec1[1]/vec2[1]
  res[2]=abs( res[1] ) *sqrt( ( vec1[2]/vec1[1] )^2+(vec2[2]/vec2[1] )^2 )
  return( res )
}
multiply <- function( vec1 , vec2 ){
  res=c()
  res[1]=vec1[1]*vec2[1]
  res[2]=abs( res[1] ) *sqrt( ( vec1[2]/vec1[1] )^2+(vec2[2]/vec2[1] )^2 )
  return( res )
}

mean <- function( vec1 , vec2 ){
  res=c()
  res[1]=( vec1[1]/vec1[2]^2+vec2[1]/vec2[2]^2 )/(1/vec1[2]^2+1/vec2[2]^2)
  res[2]=sqrt( 1/(1/vec1[2]^2+1/vec2[2]^2) )

  return( res )
}

mean3 <- function( vec1 , vec2 , vec3 ){
  res=c()
  res[1]=( vec1[1]/vec1[2]^2+vec2[1]/vec2[2]^2+vec3[1]/vec3[2]^2 )/(1/vec1[2]^2+1/vec2[2]^2+1/vec3[2]^2)
  res[2]=sqrt( 1/(1/vec1[2]^2+1/vec2[2]^2+1/vec3[2]^2) )

  return( res )
}

```

## Gausfit.R

```

library( "minpack.lm" )
source( "functions.R" )

gausfit <- function( input , bereich , weighted=FALSE, sig0=0,N0=0 ){ #—— Fitten der Exponentialfunktion

  thegaussian <- y ~ C + N/(sqrt(2*pi)*sig)*exp(-(x-mu)^2/(2*sig^2))

  daten=subset( input , x>=bereich[1] & x<= bereich[2] )

```

```

ymin=min(daten$y)
if(N0==0){
  ymax=max(daten$y)
} else {
  ymax=N0
}
mu0 =daten$x[which.max(daten$y)]
if(sig0==0)
{
  #sig0=(daten$x[bereich[2]]-daten$x[bereich[1]])/3
  sig0=(bereich[2]-bereich[1])/6
}
N0=(sqrt(2*pi)*sig0)*(ymax-ymin)
#cat(paste("\nC=",ymin,"nN=",N0,"nmu=",mu0,"nsigma=",sig0,sep=" "))
#plot(function(x){ymin + N0/(sqrt(2*pi)*sig0)*exp(-(x-mu0)^2/(2*sig0^2))},bereich[1],bereich[2],
      add=TRUE,col="yellow")
if(weighted)
  err=daten$y
else
  err=1

fit = nlsLM(the gaussian, daten, weights=1/err^2, start=list(C=ymin,N=ymax,mu=mu0,sig=sig0))

chiquadratndf=sum(summary(fit)[[2]]^2)/summary(fit)[[4]][[2]]
fitdata=rbind(summary(fit)$parameters,c(chiquadratndf,0,0,0))

return(fitdata)
}

sixgaus <- function(input, bereich, weighted=FALSE, mu01, mu02, mu03, mu04, mu05, mu06){
  the gaussian <- y ~ C + N1*exp(-(x-mu1)^2/(2*sig1^2)) + N2*exp(-(x-mu2)^2/(2*sig2^2)) + N3*exp(-(x-
    mu3)^2/(2*sig3^2)) + N4*exp(-(x-mu4)^2/(2*sig4^2)) + N5*exp(-(x-mu5)^2/(2*sig5^2)) + N6*exp(-(
    x-mu6)^2/(2*sig6^2))
  daten=subset(input,x>=bereich[1] & x<= bereich[2])
  ymin=min(daten$y)
  ymax=max(daten$y)
  sig0=(bereich[2]-bereich[1])/36
  N0=(sqrt(2*pi)*sig0)*(ymax-ymin)

  if(weighted)
    err=daten$y
  else
    err=daten$y/daten$y

  fit = nlsLM(the gaussian, daten, weights=1/err^2, start=list(C=ymin,N1=ymax,N2=ymax,N3=ymax,N4=ymax,N5
    =ymax,N6=ymax,mu1=mu01,mu2=mu02,mu3=mu03,mu4=mu04,mu5=mu05,mu6=mu06,sig1=sig0,sig2=sig0,sig3=
    sig0,sig4=sig0,sig5=sig0,sig6=sig0))

  chiquadratndf=sum(summary(fit)[[2]]^2)/summary(fit)[[4]][[2]]
  fitdata=rbind(summary(fit)$parameters,c(chiquadratndf,0,0,0))

  return(fitdata)
}

plotgaus <- function(fitdata, bereich, col="red",lwd=1,lty=1){ #—— Plotten der gefitteten Gauß
  funktion in vorhandenen Graph

  N<-fitdata["N","Estimate"]
  C<-fitdata["C","Estimate"]
  mu<-fitdata["mu","Estimate"]
  sig<-fitdata["sig","Estimate"]

  plot(function(x){C + N/(sqrt(2*pi)*sig)*exp(-(x-mu)^2/(2*sig^2))},bereich[1],bereich[2],add=TRUE,
    col=col,n=10000,lwd=lwd,lty=lty)
}

plotsix <- function(fitdata, bereich, col="red",lwd=1,lty=1){ #—— Plotten der gefitteten Gauß
  funktion in vorhandenen Graph

  C<-fitdata["C","Estimate"]
  N1<-fitdata["N1","Estimate"]

```



```

mul<-fitdata["mul","Estimate"]
sig1<-fitdata["sig1","Estimate"]
N2<-fitdata["N2","Estimate"]
mu2<-fitdata["mu2","Estimate"]
sig2<-fitdata["sig2","Estimate"]
N3<-fitdata["N3","Estimate"]
mu3<-fitdata["mu3","Estimate"]
sig3<-fitdata["sig3","Estimate"]
N4<-fitdata["N4","Estimate"]
mu4<-fitdata["mu4","Estimate"]
sig4<-fitdata["sig4","Estimate"]
N5<-fitdata["N5","Estimate"]
mu5<-fitdata["mu5","Estimate"]
sig5<-fitdata["sig5","Estimate"]
N6<-fitdata["N6","Estimate"]
mu6<-fitdata["mu6","Estimate"]
sig6<-fitdata["sig6","Estimate"]

plot(function(x){C + N1*exp(-(x-mu1)^2/(2*sig1^2)) + N2*exp(-(x-mu2)^2/(2*sig2^2)) + N3*exp(-(x-
mu3)^2/(2*sig3^2)) + N4*exp(-(x-mu4)^2/(2*sig4^2)) + N5*exp(-(x-mu5)^2/(2*sig5^2)) + N6*exp(-(
x-mu6)^2/(2*sig6^2))},bereich[1],bereich[2],add=TRUE,col=col,n=10000,lwd=lwd,lty=lty)

}

printfitdata <- function(fitdata,title=""){ #--- Ausgabe der Gauß fit-Daten

mu<-fitdata["mu","Estimate"]
smu<-fitdata["mu","Std. Error"]
A<-fitdata["N","Estimate"]
sA<-fitdata["N","Std. Error"]
sig<-fitdata["sig","Estimate"]
ssig<-fitdata["sig","Std. Error"]

cat(paste("\\text{",title,"",sep=""))
cat("\\\\\\\\n")

cat("\\\\mu=\\")
cat(mu)
cat("\\\\pm")
cat(smu)
cat("\\\\\\\\n")

cat("\\\\sigma=\\")
cat(sig)
cat("\\\\pm")
cat(ssig)
cat("\\\\\\\\n")

cat("\\A=\\")
cat(A)
cat("\\\\pm")
cat(sA)
cat("\\\\\\\\n")

}

```

## Lorentzfit.R

```

library("minpack.lm")

lorentzfit<-function(input,neg=FALSE,weighted=FALSE){

#lorentz <- y~D+C/(2*(2-(2*(x-omega)*tau)^2/(1+(2*(x-omega)*tau)^2))
lorentz <- y~D + C*1/(2*pi)*tau/((x-omega)^2+(1/2*tau)^2)

D0=max(input$y)
C0=min(input$y)-max(input$y)
omega0=input$x[which.max(input$y)]
tau0=10^(0.18)

if(neg){

```

```

C0=max(input$y)-min(input$y)
D0=max(input$y)
omega0=input$x[which.min(input$y)]

}

#print(input)
#print(c(D0, C0, omega0, tau0))

#plot(function(x){D0+C0*((2*(x-omega0)*tau0)^2/(1+(2*(x-omega0)*tau0)^2))}, -6, 6, add=TRUE, col="
green ")
if(weighted)
err=input$y
else
err=1*input$y/input$y

try({
fit=nlsLM(lorentz, input, weights=1/err^2, start=list(D=D0, C=C0, omega=omega0, tau=tau0))
#chiquadratndf=sum(residuals(fit)^2/abs(fitted(fit)))/summary(fit)$df[2]
chiquadratndf=sum(summary(fit)[[2]]^2)/summary(fit)[[4]][[2]]
fitdata=rbind(summary(fit)$parameters, c(chiquadratndf, 0, 0, 0))
return(fitdata)
})

return(NULL)
}

sixlorentz<-function(input, neg=FALSE, weighted=FALSE, omega01, omega02, omega03, omega04, omega05, omega06)
{
lorentz <- y-D + C1*1/(2*pi)*tau1/((x-omega1)^2+(1/2*tau1)^2) + C2*1/(2*pi)*tau2/((x-omega2)
^2+(1/2*tau2)^2) + C3*1/(2*pi)*tau3/((x-omega3)^2+(1/2*tau3)^2) + C4*1/(2*pi)*tau4/((x-
omega4)^2+(1/2*tau4)^2) + C5*1/(2*pi)*tau5/((x-omega5)^2+(1/2*tau5)^2) + C6*1/(2*pi)*tau6/((
x-omega6)^2+(1/2*tau6)^2)
##C1/2*(2-(2*(x-omega1)*tau1)^2/(1+(2*(x-omega1)*tau1)^2))+C2/2*(2-(2*(x-omega2)*tau2)^2/(1+(2*(x-
omega2)*tau2)^2))+C3/2*(2-(2*(x-omega3)*tau3)^2/(1+(2*(x-omega3)*tau3)^2))+C4/2*(2-(2*(x-
omega4)*tau4)^2/(1+(2*(x-omega4)*tau4)^2))+C5/2*(2-(2*(x-omega5)*tau5)^2/(1+(2*(x-omega5)*tau5
)^2))+C6/2*(2-(2*(x-omega6)*tau6)^2/(1+(2*(x-omega6)*tau6)^2))

D0=max(input$y)
C0=min(input$y)-max(input$y)
omega0=input$x[which.max(input$y)]
tau0=10^(0.18)

if(neg){
C0=max(input$y)-min(input$y)
D0=max(input$y)-0.0045
omega0=input$x[which.min(input$y)]
}

#print(input)
#print(c(D0, C0, omega0, tau0))

#plot(function(x){D0+C0*((2*(x-omega0)*tau0)^2/(1+(2*(x-omega0)*tau0)^2))}, -6, 6, add=TRUE, col="
green ")
if(weighted)
err=input$y
else
err=1*input$y/input$y

try({
fit=nlsLM(lorentz, input, weights=1/err^2, start=list(D=D0, C1=C0, C2=C0, C3=C0, C4=C0, C5=C0, C6=C0,
omega1=omega01, omega2=omega02, omega3=omega03, omega4=omega04, omega5=omega05, omega6=omega06,
tau1=tau0, tau2=tau0, tau3=tau0, tau4=tau0, tau5=tau0, tau6=tau0))
#chiquadratndf=sum(residuals(fit)^2/abs(fitted(fit)))/summary(fit)$df[2]
chiquadratndf=sum(summary(fit)[[2]]^2)/summary(fit)[[4]][[2]]
fitdata=rbind(summary(fit)$parameters, c(chiquadratndf, 0, 0, 0))
return(fitdata)
})
}

```

```

    return(NULL)
}
dispersionsfit<-function(input){

  disp <- y~D+C*((2*(x-omega)*tau)/(1+(2*(x-omega)*tau)^2)^2)

  D0=(min(input$y)+max(input$y))/2
  C0=max(input$y)-min(input$y)
  omega0=(input$x[which.max(input$y)]+input$x[which.min(input$y)])/2
  tau0=10^(-7)

  #print(c(D0,C0,omega0,tau0))

  #plot(function(x){D0+C0*((2*(x-omega0)*tau0)/(1+(2*(x-omega0)*tau0)^2)^2)},-6*10^(7),6*10^(7),add=
    TRUE,col="green")

  try({
    fit=nls(disp,input,start=list(D=D0,C=C0,omega=omega0,tau=tau0))
    chiquadratndf=sum(residuals(fit)^2/abs(fitted(fit)))/summary(fit)$df[2]
    fitdata=rbind(summary(fit)$parameters,c(chiquadratndf,0,0,0))
    return(fitdata)
  })

  return(NULL)
}

plotlorentz<-function(fitdata,bereich,lwd=1,lty=1){
  D<-fitdata["D","Estimate"]
  C<-fitdata["C","Estimate"]
  omega<-fitdata["omega","Estimate"]
  tau<-fitdata["tau","Estimate"]

  try({plot(function(x){D + C*1/(2*pi)*tau/((x-omega)^2+(1/2*tau)^2)},-6,6,add=TRUE,col="red",n
    =10000,lwd=lwd,lty=lty)})
}

plotsixlorentz<-function(fitdata,bereich,lwd=1,lty=1){
  D<-fitdata["D","Estimate"]
  C1<-fitdata["C1","Estimate"]
  omega1<-fitdata["omega1","Estimate"]
  tau1<-fitdata["tau1","Estimate"]
  C2<-fitdata["C2","Estimate"]
  omega2<-fitdata["omega2","Estimate"]
  tau2<-fitdata["tau2","Estimate"]
  C3<-fitdata["C3","Estimate"]
  omega3<-fitdata["omega3","Estimate"]
  tau3<-fitdata["tau3","Estimate"]
  C4<-fitdata["C4","Estimate"]
  omega4<-fitdata["omega4","Estimate"]
  tau4<-fitdata["tau4","Estimate"]
  C5<-fitdata["C5","Estimate"]
  omega5<-fitdata["omega5","Estimate"]
  tau5<-fitdata["tau5","Estimate"]
  C6<-fitdata["C6","Estimate"]
  omega6<-fitdata["omega6","Estimate"]
  tau6<-fitdata["tau6","Estimate"]

  try({plot(function(x){D + C1*1/(2*pi)*tau1/((x-omega1)^2+(1/2*tau1)^2) + C2*1/(2*pi)*tau2/((x-
    omega2)^2+(1/2*tau2)^2) + C3*1/(2*pi)*tau3/((x-omega3)^2+(1/2*tau3)^2) + C4*1/(2*pi)*tau4/((
    x-omega4)^2+(1/2*tau4)^2) + C5*1/(2*pi)*tau5/((x-omega5)^2+(1/2*tau5)^2) + C6*1/(2*pi)*tau6/
    ((x-omega6)^2+(1/2*tau6)^2)},bereich[1],bereich[2],add=TRUE,col="red",n=10000,lwd=lwd,lty=lty)
  })
}

plotdisp<-function(fitdata,bereich){
  D<-fitdata["D","Estimate"]
  C<-fitdata["C","Estimate"]
  omega<-fitdata["omega","Estimate"]
  tau<-fitdata["tau","Estimate"]

```

```

    try({plot(function(x){D+C*((2*(x-omega)*tau)/(1+(2*(x-omega)*tau)^2)^2)},-6*10^(7),6*10^(7),add=
      TRUE,col="red")})
}

getlorentzvalue<-function(fitdata,x){
  D<-fitdata["D","Estimate"]
  C<-fitdata["C","Estimate"]
  omega<-fitdata["omega","Estimate"]
  tau<-fitdata["tau","Estimate"]

  return(D+C/2*(2-(2*(x-omega)*tau)^2/(1+(2*(x-omega)*tau)^2)))
}

getdispvalue<-function(fitdata,x){
  D<-fitdata["D","Estimate"]
  C<-fitdata["C","Estimate"]
  omega<-fitdata["omega","Estimate"]
  tau<-fitdata["tau","Estimate"]

  return(D+C*((2*(x-omega)*tau)/(1+(2*(x-omega)*tau)^2)^2))
}

getlorentzformula<-function(fitdata){
  D0<-fitdata["D","Estimate"]
  C0<-fitdata["C","Estimate"]
  omega0<-fitdata["omega","Estimate"]
  tau0<-fitdata["tau","Estimate"]
  sD<-fitdata["D","Std. Error"]
  sC<-fitdata["C","Std. Error"]
  somega<-fitdata["omega","Std. Error"]
  stau<-fitdata["tau","Std. Error"]

  D=roundfunc(c(D0,sD))
  C=roundfunc(c(C0,sC))
  omega=roundfunc(c(omega0, somega))
  tau=roundfunc(c(tau0, stau))

  txt=paste("(",D[1],"+",D[2],")\u00d7V\u00d7(C[1]/2,"+",C[2]/2,")\u00d7(2-(2(omega_L-(",omega[1],"+",
    omega[2],")Hz)*((",tau[1],"+",tau[2],")s)^2/(1+(2(omega_L-(",omega[1],"+",omega[2],")Hz)*((",
    tau[1],"+",tau[2],")s)^2))",sep="")

  return(txt)
}

getdispformula<-function(fitdata){
  D0<-fitdata["D","Estimate"]
  C0<-fitdata["C","Estimate"]
  omega0<-fitdata["omega","Estimate"]
  tau0<-fitdata["tau","Estimate"]
  sD<-fitdata["D","Std. Error"]
  sC<-fitdata["C","Std. Error"]
  somega<-fitdata["omega","Std. Error"]
  stau<-fitdata["tau","Std. Error"]

  D=roundfunc(c(D0,sD))
  C=roundfunc(c(C0,sC))
  omega=roundfunc(c(omega0, somega))
  tau=roundfunc(c(tau0, stau))

  txt=paste("(",D[1],"+",D[2],")\u00d7V\u00d7(C[1],"+",C[2],")\u00d7(2-(2(omega_L-(",omega[1],"+",omega
    [2],")Hz)*((",tau[1],"+",tau[2],")s)/(1+(2(omega_L-(",omega[1],"+",omega[2],")Hz)*((",tau[1],"+
    "+",tau[2],")s)^2)^2)",sep="")

  return(txt)
}

```

## ReadFiles.R

```

readCSV <- function(name){
  daten=read.csv(paste("../data/",name,"_HM1508.csv",sep=""))

  return(data.frame(x=daten[[1]],y1=daten[[2]],y2=daten[[3]]))
}

```

```

}

readTKA <- function(filename){

  return (read.table(paste("../data/", filename, ".TKA", sep="")))

}

readTXT <- function(filename){

  return (read.table(paste("../data/", filename, ".txt", sep=""), dec=","))

}

```

## Supergaus.R

```

library("minpack.lm")
source("functions.R")

supergausfit <- function(input, bereich, weighted=FALSE, mu=0, sig0=0, N0=0, N01=2000000, sig01=80, mu01=750){ #—— Fitten der Exponentialfunktion

  thegaussian <- y ~ C + N/(sqrt(2*pi)*sig)*exp(-(x-mu)^2/(2*sig^2)) + N1/(sqrt(2*pi)*sig1)*exp(-(x-mu1)^2/(2*sig1^2))

  daten=subset(input, x>=bereich[1] & x<= bereich[2])
  ymin=min(daten$y)
  if(N0==0){
    ymax=max(daten$y)
  } else {
    ymax=N0
  }
  if(mu==0){
    mu0 =daten$x[which.max(daten$y)]
  } else {
    mu0=mu
  }
  if(sig0==0)
  {
    #sig0=(daten$x[bereich[2]]-daten$x[bereich[1]])/3
    sig0=(bereich[2]-bereich[1])/6
  }
  N0=(sqrt(2*pi)*sig0)*(ymax-ymin)
  # cat(paste("\nC=",ymin," \nN=",N0," \nmu=",mu0," \nsigma=",sig0, sep=""))
  # cat(paste("\nN1=",N0," \nmu=",mu0," \nsigma=",sig0, sep=""))
  #plot (function(x){ymin + N0/(sqrt(2*pi)*sig0)*exp(-(x-mu0)^2/(2*sig0^2)) + N01/(sqrt(2*pi)*sig01)*exp(-(x-mu01)^2/(2*sig01^2))}, bereich[1], bereich[2], add=TRUE, col="green")
  if(weighted)
  {
    err=daten$y
    fit = nlsLM(thegaussian, daten, weights=1/err^2, start=list(C=ymin, N=N0, mu=mu0, sig=sig0, N1=N0, mu1=mu0, sig1=sig0))
  }
  else
  {
    fit = nlsLM(thegaussian, daten, start=list(C=ymin, N=ymax, mu=mu0, sig=sig0, N1=N0, mu1=mu0, sig1=sig0))

    #chiquadratndf=sum(residuals(fit)^2/abs(fitted(fit)))/summary(fit)$df[2]
    chiquadratndf=sum(summary(fit)[[2]]^2)/summary(fit)[[4]][[2]]
    fitdata=rbind(summary(fit)$parameters, c(chiquadratndf, 0, 0, 0))
    return(fitdata)
  }
}

plotsupergaus <- function(fitdata, bereich, col="red"){ #—— Plotten der gefitteten Gauß funktion in vorhandenen Graph

  N<-fitdata["N", "Estimate"]
  C<-fitdata["C", "Estimate"]
  mu<-fitdata["mu", "Estimate"]
  sig<-fitdata["sig", "Estimate"]

  N1<-fitdata["N1", "Estimate"]

```

```

mul<-fitdata["mul","Estimate"]
sig1<-fitdata["sig1","Estimate"]

plot(function(x){C + N/(sqrt(2*pi)*sig)*exp(-(x-mu)^2/(2*sig^2)) + N1/(sqrt(2*pi)*sig1)*exp(-(x-
mul)^2/(2*sig1^2))},bereich[1],bereich[2],add=TRUE,col=col)

}

printsupergausdata <- function(fitdata,title=""){ #--- Ausgabe der Gauß fit-Daten

mu<-fitdata["mu","Estimate"]
smu<-fitdata["mu","Std. Error"]
mul<-fitdata["mul","Estimate"]
smul<-fitdata["mul","Std. Error"]
A<-fitdata["N","Estimate"]
sA<-fitdata["N","Std. Error"]
C<-fitdata["C","Estimate"]
sC<-fitdata["C","Std. Error"]
sig1<-fitdata["sig1","Estimate"]
ssig1<-fitdata["sig1","Std. Error"]
sig<-fitdata["sig","Estimate"]
ssig<-fitdata["sig","Std. Error"]
A1<-fitdata["N1","Estimate"]
sA1<-fitdata["N1","Std. Error"]

cat(paste("\\text{",title,"}",sep=""))
cat("\\\\n")

cat("\\mu=\\")
cat(mu)
cat("\\pm\\")
cat(smu)
cat("\\\\n")

cat("\\sigma=\\")
cat(sig)
cat("\\pm\\")
cat(ssig)
cat("\\\\n")

cat("\\A=\\")
cat(A)
cat("\\pm\\")
cat(sA)
cat("\\\\n")

cat("\\mul=\\")
cat(mul)
cat("\\pm\\")
cat(smul)
cat("\\\\n")

cat("\\sigma1=\\")
cat(sig1)
cat("\\pm\\")
cat(ssig1)
cat("\\\\n")

cat("\\A1=\\")
cat(A1)
cat("\\pm\\")
cat(sA1)
cat("\\\\n")

cat("\\C=\\")
cat(C)
cat("\\pm\\")
cat(sC)
cat("\\\\n")

}

```

## Voigtfit.R

```

library("minpack.lm")
library(RcppFaddeeva)#install.packages('RcppFaddeeva')

voigtfit <- function(input,A0,C0,mu0,sigma0,gamma0,weighted=FALSE) {

  y <- input$y
  x <- input$x

  thevoigt <- y ~ A * Voigt(x,mu,sigma,gamma) + C

  if(weighted)
    err=input$y
  else
    err=1*input$y/input$y

  try({
    fit=nlsLM(thevoigt,input,weights=1/err^2,start=list(A=A0,C=C0,mu=mu0,sigma=sigma0,gamma=gamma0),
      control=nls.control(maxiter=100))
    chiquadratndf=sum(summary(fit)[[2]]^2)/summary(fit)[[4]][[2]]
    fitdata=rbind(summary(fit)$parameters,c(chiquadratndf,0,0,0))
    return(fitdata)
  })
}

sixvoigt <- function(input,A01,A02,A03,A04,A05,A06,C0,mu01,mu02,mu03,mu04,mu05,mu06,sigma01,sigma02,
  sigma03,sigma04,sigma05,sigma06,gamma01,gamma02,gamma03,gamma04,gamma05,gamma06,weighted=FALSE)
{

  y <- input$y
  x <- input$x

  thevoigt <- y ~ A1 * Voigt(x,mu1,sigma1,gamma1) + A2 * Voigt(x,mu2,sigma2,gamma2) + A3 * Voigt(x,
    mu3,sigma3,gamma3) + A4 * Voigt(x,mu4,sigma4,gamma4) + A5 * Voigt(x,mu5,sigma5,gamma5) + A6 *
    Voigt(x,mu6,sigma6,gamma6) + C

  if(weighted)
    err=input$y
  else
    err=1*input$y/input$y
  print(C0)
  plot(function(x){A01 * Voigt(x,mu01,sigma01,gamma01) + A02 * Voigt(x,mu02,sigma02,gamma02) + A03 *
    Voigt(x,mu03,sigma03,gamma03) + A04 * Voigt(x,mu04,sigma04,gamma04) + A05 * Voigt(x,mu05,
    sigma05,gamma05) + A06 * Voigt(x,mu06,sigma06,gamma06) + C0},-8.05,8.05,add=TRUE,col="yellow",
    n=10000)

  try({
    fit=nlsLM(thevoigt,input,weights=1/err^2,start=list(A1=A01,A2=A02,A3=A03,A4=A04,A5=A05,A6=A06,C=
      C0,mu1=mu01,mu2=mu02,mu3=mu03,mu4=mu04,mu5=mu05,mu6=mu06,sigma1=sigma01,sigma2=sigma02,
      sigma3=sigma03,sigma4=sigma04,sigma5=sigma05,sigma6=sigma06,gamma1=gamma01,gamma2=gamma02,
      gamma3=gamma03,gamma4=gamma04,gamma5=gamma05,gamma6=gamma06), control = list(maxiter = 100))
    chiquadratndf=sum(summary(fit)[[2]]^2)/summary(fit)[[4]][[2]]
    fitdata=rbind(summary(fit)$parameters,c(chiquadratndf,0,0,0))
    return(fitdata)
  })
}

plotvoigt<-function(fitdata,bereich,col="red",lwd=1,lty=1){
  A<-fitdata["A","Estimate"]
  C<-fitdata["C","Estimate"]
  mu<-fitdata["mu","Estimate"]
  sigma<-fitdata["sigma","Estimate"]
  gamma<-fitdata["gamma","Estimate"]

  try({plot(function(x){A * Voigt(x,mu,sigma,gamma) + C},bereich[1],bereich[2],add=TRUE,col=col,n
    =10000,lwd=lwd,lty=lty)})
  #print(optimize(function(x){A * Voigt(x,mu,sigma,gamma) + C},interval=c(-6,6)))
  #print(optimize(function(x){A * Voigt(x,mu,sigma,gamma) + C},interval=c(-6,6),maximum=TRUE))
  print(A * Voigt(mu,mu,sigma,gamma) + C)
}

```

```

plotsixvoigt<-function(fitdata ,bereich ,col="red" ,lwd=1,lty=1){
  C<-fitdata["C", "Estimate"]
  A1<-fitdata["A1", "Estimate"]
  A2<-fitdata["A2", "Estimate"]
  A3<-fitdata["A3", "Estimate"]
  A4<-fitdata["A4", "Estimate"]
  A5<-fitdata["A5", "Estimate"]
  A6<-fitdata["A6", "Estimate"]
  mu1<-fitdata["mu1", "Estimate"]
  mu2<-fitdata["mu2", "Estimate"]
  mu3<-fitdata["mu3", "Estimate"]
  mu4<-fitdata["mu4", "Estimate"]
  mu5<-fitdata["mu5", "Estimate"]
  mu6<-fitdata["mu6", "Estimate"]
  sigma1<-fitdata["sigma1", "Estimate"]
  sigma2<-fitdata["sigma2", "Estimate"]
  sigma3<-fitdata["sigma3", "Estimate"]
  sigma4<-fitdata["sigma4", "Estimate"]
  sigma5<-fitdata["sigma5", "Estimate"]
  sigma6<-fitdata["sigma6", "Estimate"]
  gamma1<-fitdata["gamma1", "Estimate"]
  gamma2<-fitdata["gamma2", "Estimate"]
  gamma3<-fitdata["gamma3", "Estimate"]
  gamma4<-fitdata["gamma4", "Estimate"]
  gamma5<-fitdata["gamma5", "Estimate"]
  gamma6<-fitdata["gamma6", "Estimate"]

  try({plot(function(x){A1 * Voigt(x,mu1,sigma1,gamma1) + A2 * Voigt(x,mu2,sigma2,gamma2) + A3 *
    Voigt(x,mu3,sigma3,gamma3) + A4 * Voigt(x,mu4,sigma4,gamma4) + A5 * Voigt(x,mu5,sigma5,gamma5)
    + A6 * Voigt(x,mu6,sigma6,gamma6) + C},bereich[1] ,bereich[2] ,add=TRUE,col=col ,n=10000,lwd=lwd
    ,lty=lty)})
}

```



## Abbildungsverzeichnis

1	Funktionsweise eines Szintillators . . . . .	3
2	Funktionsweise eines Photomultipliers . . . . .	3
3	Zerfallsschema von $^{57}\text{Co}$ . . . . .	4
4	Dopplerverbreiterung . . . . .	5
5	Resonanzgeschwindigkeit . . . . .	6
6	Moessbauerspektrum . . . . .	7
7	Vergleich zwischen Einstein- und Debye-Modell . . . . .	9
8	Debye-Waller-Faktor . . . . .	9
9	Temperaturabhängigkeit des Debye-Waller-Faktors . . . . .	10
10	Hyperfeinaufspaltung von Eisen-57 . . . . .	11
11	Erwartete Signalformen nach den einzelnen Einheiten . . . . .	13
12	Schematische Darstellung des Versuchsaufbaus für die Eichmessung . . . . .	14
13	Schematische Darstellung des Versuchsaufbaus . . . . .	14
14	Signalausgänge des Hauptverstärkers . . . . .	15
15	Signalausgänge des SCA und des Linear Gates . . . . .	15
16	Beispielhafte Auswertung der $K_\alpha$ -Linie von Silber . . . . .	17
17	Doppel-Gaußfit zur Auswertung der $K_\alpha$ -Linie von Terbium . . . . .	18
18	Linearer Fit zur Energiekalibrierung des MCA . . . . .	19
19	Messung zur Bestimmung des Compton-Untergrunds . . . . .	20
20	Messung der Transmission durch die Edelstahlprobe (Einlinienabsorber) . . . . .	21
21	Relative Linienverbreiterung . . . . .	23
22	Messung der Transmission durch die Eisenprobe (Sechslinienabsorber) . . . . .	25

## Tabellenverzeichnis

1	Fitdaten der Gaußfits zur Eichung . . . . .	18
2	Fitdaten der Gaußfits zur Eichung . . . . .	18
3	Fitdaten des sechsfachen Gaußfits . . . . .	26
4	Fitdaten des sechsfachen Lorentzfits . . . . .	26
5	Fitdaten des sechsfachen Voigtfits . . . . .	26
6	Magnetische Quantenzahlen der Hyperfeinaufspaltung von $^{57}\text{Fe}$ . . . . .	27
7	Messergebnisse für den Einlinienabsorber . . . . .	29
8	Messergebnisse für den Sechslinienabsorber . . . . .	30

## Literatur

- [1] W. Demtröder. *Experimentalphysik 4*. Experimentalphysik / Wolfgang Demtröder. Springer Berlin Heidelberg, 2009. ISBN: 9783642015984.  
URL: <https://books.google.de/books?id=50xTorK8Zx8C>.
- [2] T. Kotyk. “Versuche zur Radioaktivität im Physikalischen Fortgeschrittenen Praktikum an der Albert-Ludwigs-Universität Freiburg”. Staatsexamensarbeit. Institut für Mathematik und Physik, Albert-Ludwigs-Universität Freiburg, 2011.
- [3] DESY. *Praktikum im DESY Zeuthen*. 2017. URL: [https://www-zeuthen.desy.de/exps/physik\\_begreifen/chris/Photomultiplier.html](https://www-zeuthen.desy.de/exps/physik_begreifen/chris/Photomultiplier.html).
- [4] A. Zwerger, S. Winkelmann, M. Köhli. *Versuchsanleitung Fortgeschrittenen-Praktikum Teil I - Der Mößbauer-Effekt*. 2012.
- [5] P. K. Jakobs. *Experimentalphysik V - Kern- und Teilchenphysik*. 2015.
- [6] G. Morris. *Lecture Notes - Statistical Physics*. 2013.  
URL: <http://chate.phys.unsw.edu.au/~garymorriss/PHYS3020.html>.
- [7] T. Mayer-Kuckuk. *Kernphysik*. 6. Aufl. Stuttgart: Teubner-Verlag, 1994.
- [8] H. F. S. Margulies P. Debrunner.  
“Transmission and Line Broadening in the Mössbauer Effect. II”.  
In: *Nuclear Instruments and Methods* (21 1963), S. 217–231.
- [9] A. W. G. Schatz. *Nukleare Festkörperphysik*. 3. Aufl.  
Universität Konstanz, Hahn-Meitner-Institut Berlin, 1997.