

# Entwicklung eines Algorithmus zur Entfaltung und Signifikanzprüfung von Vielkanalspektren mit geringer Statistik

Development of an algorithm for deconvolution and significance testing of spectra with low counting rates

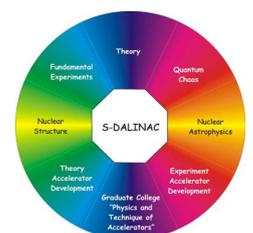
Bachelor Thesis von Jan Patrick Krol

September 2010



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Fachbereich Physik  
Institut für Kernphysik



Gefördert durch die DFG im Rahmen des SFB 634

---

Entwicklung eines Algorithmus zur Entfaltung und Signifikanzprüfung von Vielkanalspektren mit geringer Statistik

Development of an algorithm for deconvolution and significance testing of spectra with low counting rates

vorgelegte Bachelor Thesis von Jan Patrick Krol

1. Gutachten: Prof. Dr. Peter von Neumann-Cosel
2. Gutachten:

Eingegangen am:

---

## **1. Erklärung zur Bachelor Thesis**

---

Hiermit versichere ich die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 30. September 2010

---

Jan Patrick Krol

---

---

## 2. Inhaltsverzeichnis

---

1.....Erklärung zur Bachelor Thesis	i
2.....Inhaltsverzeichnis	ii
3.....Einführung	2
4.....Grundlagen charakteristischer Grenzen bei Vielkanalspektren	2
4.1. Charakteristische Grenzen	2
4.2. Untersuchung einzelner Linien	4
4.3. Anwendung der charakteristischen Grenzen im Algorithmus	7
5.....Technische Grundlagen der Implementierung	8
5.1. Wahl der Programmiersprache	8
5.2. Ausführungsebenen	8
5.2.1. Frontend	9
5.2.2. Backend	9
5.3. Funktionsumfang	10
5.3.1. Einlesen von Daten	10
5.3.2. Anpassen von Funktion im markierten Bereich	11
5.3.3. Dateiausgabe	11
6.....Bedienung	12
6.1. Allgemeine Bedienung	12
6.1.1. Anzeigebereich des Spektrums	12
6.1.2. Anzeigebereich der Fitparameter	12
6.1.3. Kontrollbereich	13
6.2. Datenbereich der Ausgleichsrechnung	13
6.3. Anpassen von Linien	13
7.....Ergebnisse	14
7.1. Untersuchung eines generierten Spektrums	14
7.2. Untersuchung des IAEA2008 Spektrums	17
7.3. Schätzung des Untergrundes in schwierigen Bereichen	18
7.3.1. Überlegungen zur Dynamisierung von Grenzen des Untersuchungsbereiches	19
7.4. Problemfelder in der Implementierung	20
7.5. Erweiterungsmöglichkeiten	20
8.....Zusammenfassung	21
9.....Ausblick	21
10. ..Anhang	22
10.1. Digitaler Anhang	22
10.2. Datenstruktur im Backend	22

---

10.3. Microsoft Public License (Ms-PL)	24
11. ..Abbildungsverzeichnis	25
12. ..Literaturverzeichnis	25

---

### 3. Einführung

---

In den letzten zwei Jahrhunderten wurden in der Physik große Fortschritte gemacht, ihre Umwelt und Phänomene zu beschreiben. Besonders durch die Erkenntnisse der Quantenphysik und der Entwicklung von Formalismen in ihr und deren Anwendung konnten mikroskopische Effekte und Phänomene nicht nur beschrieben, sondern auch verstanden werden. Auf der experimentellen Seite führte dies zur Entwicklung neuer Messgeräte, wie z.B. Detektoren, und erzeugte eine große Menge von Messdaten, die alle ausgewertet und bewertet werden. Daraus entsteht die Notwendigkeit Messergebnisse einer Signifikanzprüfung zu unterziehen, mit der Aussagen über die Qualität der Messung getroffen werden können. Das Konzept der charakteristischen Grenzen bietet hierzu einen modernen statistischen Ansatz, diese Problemstellung zu lösen. Diese Grenzen treffen innerhalb von Messungen eine Aussage, ob untersuchte Linien im vorliegenden Zusammenhang nur vermutet werden können oder mit einer gewissen Irrtumswahrscheinlichkeit angenommen werden dürfen. Mit der inzwischen weit fortgeschrittenen Verbreitung von Computern und deren konsequenten Einsatz können Auswertungen durch Anwendung dieser Bewertungsmethode unterstützt werden und Messungen als solche beschrieben und eingeordnet.

Die „Entwicklung eines Algorithmus zur Entfaltung und Signifikanzprüfung von Vielkanalspektren mit geringer Statistik“ soll aufbauend auf den DIN 1319 und DIN 25482, die Messunsicherheiten und charakteristische Grenzen beschreiben, diesen Ansatz implementieren und diesen dem Forscher bei der Auswertung seiner Messdaten zur Verfügung stellen. Eine allgemeine Anforderung an den zu entwickelnden Algorithmus ist daher, dass dieser robust ist und über eine grafische Oberfläche, die für den Benutzer intuitiv bzw. schnell erlernbar ist, einfach zu bedienen. Desweiteren soll dieser anpassbar sein, sowie modular erweitert werden können. Die Funktionalität, im Sinne dieser Forschungsarbeit, sind die Möglichkeiten zur Entfaltung von Vielkanalspektren von Gammastrahlungsmessungen und der Angabe von Erkennungs- und Nachweisgrenzen.

---

### 4. Grundlagen charakteristischer Grenzen bei Vielkanalspektren

---

Der im Rahmen dieser Arbeit entstandene Algorithmus ist letztendlich zur Auswertung von Elektronenstreupektren geplant, der Einfachheit halber wird der Fokus auf Gammaspektren gelegt. Die Besonderheit bei Gammaspektren liegt darin, dass sich die Linien als symmetrische Gaußverteilung mit einer Stufenfunktion im Untergrund darstellen lassen. Die Auswertung von Elektronenstreupektren wird nicht implementiert, sondern durch Bereitstellen entsprechender Schnittstellen vorbereitet.

Das typische Ziel bei der Auswertung eines Vielkanalspektrums ist die Nettofläche der Linie eines Radionuklids zu bestimmen. Diese Nettofläche ist daher die physikalische Messgröße, deren wahrer Wert geschätzt werden soll. Dies geschieht mit Hilfe der Ausgleichsrechnung, die der Linienform und dem Untergrund angemessene Modellfunktionen an die Kanalinhalt des Spektrums anpasst.

Wenn die Amplitude der Linie so klein ist, dass sie vom Rauschen des Untergrundes nicht mehr eindeutig trennbar ist, stellt sich die Frage, ob ein dann erhaltenes Messergebnis verwertbar ist oder nicht. Um eine Aussage darüber zu treffen, ob die erhaltene Nettofläche vom Untergrund signifikant verschieden ist, wird der Messwert mit den sogenannten charakteristischen Grenzen, Erkennungsgrenze (EKG) und Nachweisgrenze (NWG), verglichen. Im Folgenden sollen diese beiden Begriffe näher erklärt werden.

#### 4.1. Charakteristische Grenzen

Charakteristische Grenzen im Rahmen der Bayes-Statistik dienen dazu zu entscheiden, ob eine Messgröße von Null verschieden ist oder nicht. Im vorliegenden Fall handelt es sich bei der Messgröße um die Nettofläche einer Linie im Gammaspektrum. Die zentralen Fragen bei der Auswertung dieser Spektren lauten daher [1, S.1]:

- Wurde eine Linie anhand ihrer Nettofläche erkannt?
- Ab wann kann die Nettofläche als zuverlässig nachgewiesen angesehen werden?
- In welchem Wertebereich der Nettofläche ist ihr wahrer Wert enthalten?

Mithilfe von Hypothesentests, werden diese Fragen untersucht. Dabei werden drei Grenzen eingeführt: Die Erkennungsgrenze, die das Vorliegen eines physikalischen Effektes gekennzeichnet. Die Nachweisgrenze, die den kleinsten noch nachweisbaren Wert angibt. Sowie die Vertrauensgrenzen, die im Falle des Erkennens eines physikalischen Effektes einen Bereich angeben und in dem mit einer bestimmten Wahrscheinlichkeit der wahre Werte einer Messgröße enthalten ist.

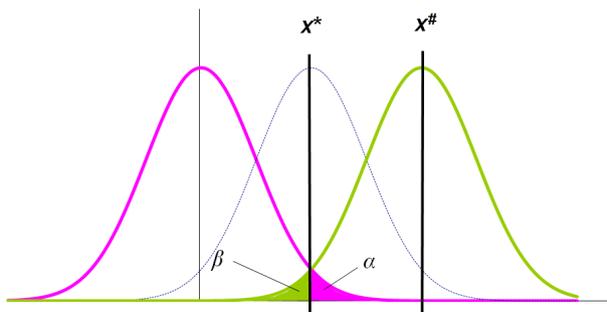


Abbildung 1:

Schematische Darstellung zur Festlegung der Erkennungsgrenze  $X^*$  und Nachweisgrenze  $X^\#$  mit den Irrtumswahrscheinlichkeiten  $\alpha$  und  $\beta$ . Die rosafarbene Kurve zeigt die Wahrscheinlichkeitsverteilung der Nullhypothese, die grüngefärbte die der Alternativhypothese.

Der Hypothesentest, aus dem sich die charakteristischen Grenzen ableiten lassen, testet statistisch die Nullhypothese, die Nettofläche sei null  $A_N = 0$ , gegen die Alternativhypothese,  $A_N > 0$ .

Das Ergebnis des Hypothesentests ist entweder die Annahme der Nullhypothese oder der Alternativhypothese. Bei dieser Entscheidung kann es zwei Arten von Fehlern geben. Eine Entscheidung für die Alternativhypothese, obwohl die Nullhypothese richtig wäre (Fehler 1. Art) oder eine Entscheidung für die Nullhypothese, obwohl die Alternativhypothese richtig wäre (Fehler 2. Art).

Die Erkennungsgrenze ist bei diesem statistischen Test als kritischer Wert zu einer bestimmten Irrtumswahrscheinlichkeit  $\alpha$  definiert. Wenn der Messwert die Erkennungsgrenze unterschreitet, wird die Nullhypothese  $A_N = 0$  angenommen, d.h. die Linie ist nicht signifikant vom Untergrund verschieden. Überschreitet der Messwert dagegen die Erkennungsgrenze, so wird in höchstens  $\alpha \cdot 100\%$  der Fälle auf  $A_N > 0$  entschieden, obwohl  $A_N = 0$  (Fehler 1. Art).

Der gegenteilige Fall, das irrtümliche Annehmen der Nullhypothese (Fehler 2. Art), führt zur Definition der Nachweisgrenze. Diese ist durch den kleinsten wahren Wert einer Messgröße definiert, für den der Fehler 2. Art gleich einer bestimmten Wahrscheinlichkeit  $\beta$  ist, bezogen auf die Erkennungsgrenze. Wenn die Erkennungsgrenze unterschritten wird, könnte der Messwert immer noch zur Verteilung um die Nachweisgrenze gehören. Dies führt jedoch zur Annahme der Nullhypothese  $A_N = 0$ , obwohl der Messwert zur Verteilung der Nachweisgrenze gehört und damit  $A_N > 0$ . Dieser Fehler 2. Art soll in maximal  $\beta \cdot 100\%$  auftreten.

*„Für die Beurteilung eines Messverfahrens in Bezug auf seine Eignung für den Zweck zieht man die Nachweisgrenze heran und vergleicht sie mit einem Richtwert, zur Beurteilung eines Messergebnisses jedoch die Erkennungsgrenze, die mit dem Messergebnis verglichen wird.“ [1, S.4]*

Die Vertrauensgrenzen geben hingegen ein Intervall oberhalb der Erkennungsgrenze an, in dem sich mit einer gewissen Wahrscheinlichkeit  $\gamma$  der wahre Wert der Messgröße befindet. Die Vertrauensgrenzen anzugeben erübrigt sich hier, da die Unsicherheiten der Messgröße als ausreichend angesehen werden.

## 4.2. Untersuchung einzelner Linien

Zur Untersuchung von Linien im Spektrum wird eine Normalverteilung als mathematische Modellfunktion verwendet und an die Messergebnisse angepasst. Diese Modellfunktion ist eine technische Näherung, die neben der Lorentz-Funktion, die die  $\gamma$ -Linien selbst beschreibt, auch den Einfluss von Germaniumdetektoren und der Messelektronik beachtet [2],

$$Y(X) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{X-\mu}{\sigma}\right)^2}.$$

Das Anpassen von Modellfunktionen geschieht durch Ausgleichsrechnung nach der Methode der kleinsten Quadratsumme. Danach wird das Minimum der Summe über das Quadrat der Differenz jedes Datenpunkts  $X_i$  des Funktionswertes  $Y_i$  der Modellfunktion und des gemessenen Datenpunkts gebildet und anschließend durch Variation der Parameter  $s_i$  minimiert,

$$\min \underbrace{\sum_i (f(x_i, s_i) - y_i)^2}_{\chi^2}.$$

Diese Parameter  $s_i$  führen dazu, dass **alle** Modellwerte mit den Messwerten so verträglich wie möglich sind. Die Methode der kleinsten Quadratsumme gibt jedoch keine Auskunft darüber, ob eine Modellfunktion per se geeignet ist die Daten zu beschreiben. Dies muss über geeignete statistische Betrachtungen gesondert überprüft werden, hierzu eignet sich unter anderem das Bestimmtheitsmaß und das  $\chi_{red}^2$ , das gleichbedeutend mit dem Quotienten der Residuenquadratsumme und der Differenz aus Anzahl der Datenpunkte und Freiheitsgraden,  $\chi_{red}^2 = \frac{S_{red}}{n-p}$ , der Modellfunktion nach dem Fit ist.

Zur numerischen Berechnung des Minimierungsproblems wird der Marquard-Levenberg Algorithmus verwendet, der sich in den letzten Jahrzehnten zum Standardlösungsverfahren für nichtlineare Minimierungsaufgaben entwickelt hat und über welche schon umfangreiche wissenschaftliche Arbeiten vorliegen [3][4][5]. Grundlage des Marquard-Levenberg Algorithmus ist das Gauß-Newton Verfahren. Dieses überführt in jedem Iterationsschritt die Modellfunktion des nicht linearen Minimierungsproblems in eine quadratische Näherung, deren Minimum berechnet werden kann. Durch dieses Vorgehen ist eine Kenntnis der zweiten Ableitung nicht notwendig. Diese Linearisierung entspricht der Berechnung der ersten Glieder des Taylor-Polynoms, welche numerisch aus den Messdaten gewonnen werden können. Der Marquard-Levenberg Algorithmus ergänzt das Gauß-Newton Verfahren um eine Regularisierung der Schrittweite. Diese Regularisierung führt dazu, dass zu Beginn der Minimierung die Iterationsschritte groß gewählt werden, um eine schnellere Konvergenz zu erreichen, und bei Annäherung an das Minimum die Schrittweite reduziert wird, um den Suchbereich zu vergrößern. Lokal geht dabei der Marquard-Levenberg Algorithmus in das Gauß-Newton Verfahren über und konvergiert somit quadratisch.

Das Anpassen von Gaußfunktionen an die Messdaten geschieht in zwei Stufen, zuerst wird der Untergrund aus Messwerten im Randbereich um eine Linie herum bestimmt und danach die Gaußfunktion anhand der vom Untergrund bereinigten Daten (DIN 25482) [1, S.122ff.][6].

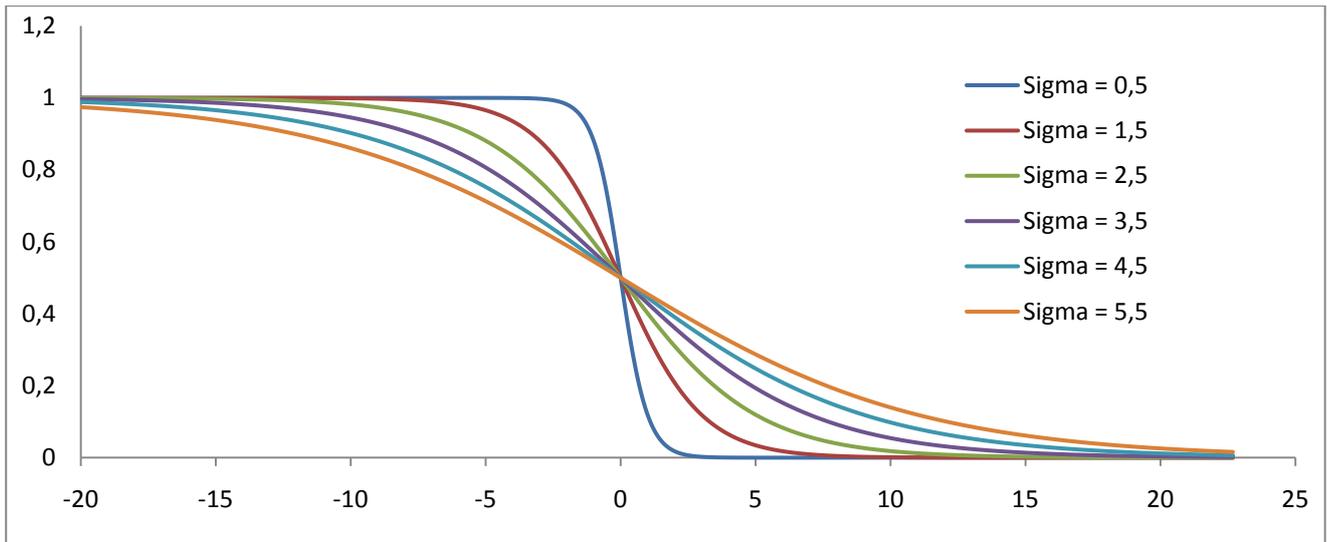


Abbildung 2:  
 Im Diagramm sind Stufenfunktionen für verschiedene Parameter  $\sigma$  dargestellt. Die übrigen Parameter sind fest mit  $\mu_{\text{Untergrund}} = 0, a_0 = 0, a_{\text{diff}} = 1$ .

Zur Beschreibung des Untergrundes wird als Modellfunktion eine Stufenfunktion, wie in Abbildung 2 dargestellt, die auch die Fermi-Verteilung beschreibt, verwendet

$$Y_{\text{Untergrund}}(X) = a_0 + a_{\text{diff}} \frac{1}{e^{\frac{X - \mu_{\text{Untergrund}}}{\sigma_{\text{Untergrund}}}} + 1} + k(X).$$

Der Parameter  $a_0$  kennzeichnet den Versatz in Y-Richtung der linken Flanke,  $a_{\text{diff}}$  die Differenz in Y-Richtung zur rechten Flanke,  $\mu_{\text{Untergrund}}$  die Lage und  $\sigma_{\text{Untergrund}}$  die Steigung der Kante. Die Funktion  $k(X)$  ist eine Nebenbedingung und dient dazu, den Wendepunkt innerhalb des Linienbereichs unter dem Maximum der Linie zu erzwingen. In  $k(X)$  wird das Verhältnis des Abstandes des Wendepunkts zum Hauptbereichsmittelpunkt und der halben Breite dieses Bereiches mit einer hohen, geraden natürlichen Zahl potenziert. Liegt der Wendepunkt außerhalb dieses Bereiches ist der zusätzliche Beitrag durch  $k(X)$  sehr groß, liegt er innerhalb wird er kleiner 1 und liegt er nahe dem Mittelpunkt des Hauptbereiches verschwindet dieser.

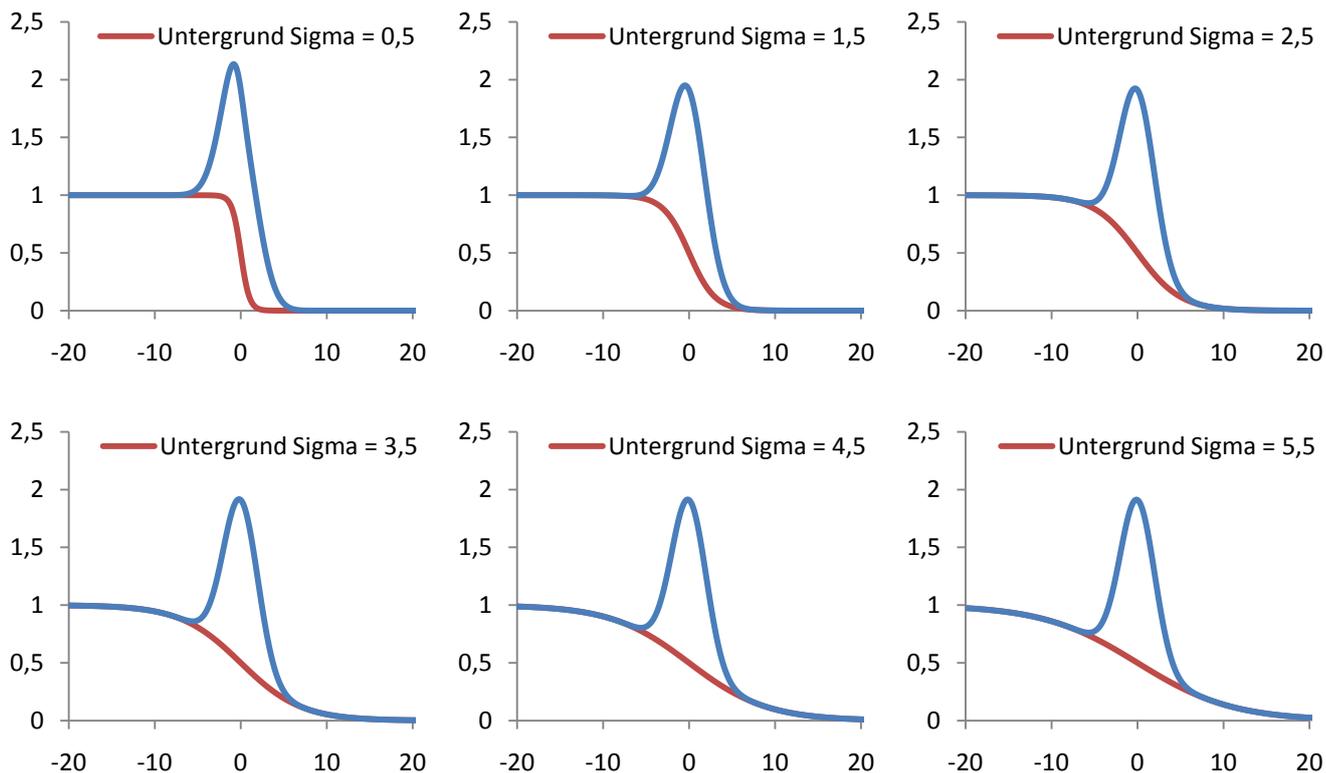


Abbildung 3:

Die Abbildung zeigt beispielhaft den Einfluss des Parameters  $\sigma$  der Modellfunktion des Untergrundes auf die Linienform. Für die Normalverteilung wurde  $Y(X) = 5 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{X}{\sigma}\right)^2}$  verwendet.

In den einzelnen Graphen in Abbildung 3 sind zur Verdeutlichung des Einflusses der gewählten Untergrundfunktion auf die Linienform der Normalverteilung jeweils die gleiche Verteilung zum Untergrund addiert. In den verschiedenen Diagrammen wird lediglich der Parameter  $\sigma_{\text{Untergrund}}$  variiert, die übrigen Parameter von  $Y_{\text{Untergrund}}(X)$  sind fest mit  $\mu_{\text{Untergrund}} = 0, a_0 = 0, a_{\text{diff}} = 1$ . Aufgrund der Symmetrie beider Funktionen können so die verschiedenen Randbereiche erfasst werden.

Obwohl durch die gesonderte Ausgleichsrechnung des Untergrundes ein erhöhter Rechenaufwand gegenüber der linearen Subtraktion nach dem Trapezverfahren [1, S.66] getätigt werden muss, können verschiedene Formen des Untergrundes in guter Näherung nachgebildet werden. Durch geeignete Wahl der Parameter können zum Beispiel ein über den untersuchten Bereich konstanter Untergrund, ein konstanter, jedoch links- und rechtsseitig der Gaußkurve verschiedener oder ein linear ansteigend bzw. abfallender Untergrund nachgebildet werden.

Alternativ kann der Untergrund gleichzeitig mit der Modellfunktion der Messung ausgeglichen werden, dabei gibt es verschiedene Möglichkeiten die Ausgleichsfunktion zu gestalten: Zum einen können die bisherigen Funktionen von Untergrund und Modellfunktion des Effektes einfach addiert werden, hierdurch ergibt sich jedoch kein Vorteil gegenüber dem getrennten Ausgleichen; zum anderen könnte eine neue Funktion aufgestellt werden, in der als Randbedingung der Wendepunkt der Untergrundfunktion mit der Peakmitte zusammenfällt. Dadurch würde ein Freiheitsgrad wegfallen.

Bei beiden alternativen Verfahren würde jedoch die Kenntnis beziehungsweise Schätzung über die Zuordnung von Datenpunkten entweder zum Linienbereich oder Randbereich ignoriert und darüber hinaus das in DIN25482-5 beschriebene Verfahren nicht eingehalten.

### 4.3. Anwendung der charakteristischen Grenzen im Algorithmus

Die Messung der Impulszahl in einem bestimmten Kanal kann als Zufallsexperiment aufgefasst werden. Ein physikalisches Ereignis, das in diesem detektiert werden sollte, kann mit einer gewissen Wahrscheinlichkeit entweder gemessen werden oder nicht, dies entspricht im Wesentlichen einem Bernoulli Experiment. Eine Messung mehrerer Ereignisse entspricht damit einer Verkettung mehrerer Bernoulli Experimente, die durch eine Poisson-Verteilung beschrieben werden kann. Dies ermöglicht es Unsicherheiten in der Auswertung mit statistischen aus dieser Verteilung anzugeben.

Da die untergrundbereinigten Messdaten keiner normierten Normalverteilung entsprechen, wird die Modellfunktion um einen Proportionalitätsfaktor  $p$  erweitert,

$$Y(X) = p \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{X-\mu}{\sigma}\right)^2}.$$

Daraus folgt direkt, dass der Proportionalitätsfaktor den Flächeninhalt des Integrals über die Modellfunktion angibt, dieser Wert entspricht der Nettofläche. Um die Bruttofläche zu erhalten, muss die Modellfunktion des Untergrundes numerisch integriert werden und zu dieser addiert.

Die Flächen entsprechen den Summen der Impulse im betrachteten Kanalbereich, dabei bezeichnet  $A_0$  die Untergrundfläche,  $A_N$  die Nettofläche und  $A_B$  die Bruttofläche. Zur Berechnung der Nachweis- und Erkennungsgrenzen werden die aus dem Fit gewonnenen Linienflächen verwendet.

$$\begin{aligned} A_N &= A_B - A_0 \\ u^2(A_N) &= u^2(A_B) + u^2(A_0) = A_B + A_0 \end{aligned}$$

Die Unsicherheiten der Flächen werde angesetzt als statistische Unsicherheiten durch die Poissonverteilung,  $u(X) = \sqrt{X}$ . [1, S. 15]. In einer erweiterten Version des Algorithmus könnten die Unsicherheiten gemäß DIN 25482-12 noch genauer abgeschätzt werden.

Für die Verteilung um den wahren Wert  $\tilde{A}_N$  soll gelten:

$$\begin{aligned} \tilde{A}_N &= A_B - A_0 \Rightarrow A_B = \tilde{A}_N + A_0, \\ \tilde{u}^2(\tilde{A}_N) &= u^2(A_B) + u^2(A_0) = A_B + A_0 = \tilde{A}_N + 2A_0; \\ \Rightarrow \tilde{u}^2(0) &= 0 + 2A_0 \mid \tilde{A}_N := 0 \text{ für Erkennungsgrenze.} \end{aligned}$$

Mit dieser Vorbereitung ergibt sich für die Erkennungsgrenze  $A_N^*$ , wobei  $k_{1-\alpha}$  das  $(1 - \alpha)$ -Quantil der Gaußverteilung bezeichnet,

$$A_N^* = k_{1-\alpha} \cdot \tilde{u}(0) = k_{1-\alpha} \cdot \sqrt{2A_0}.$$

Aus der Erkennungsgrenze lässt sich die Nachweisgrenze ableiten, wobei  $k_{1-\beta}$  das  $(1 - \beta)$ -Quantil der Gaußverteilung bezeichnet:

$$\begin{aligned} A_N^\# &= A_N^* + k_{1-\beta} \cdot \tilde{u}(A_N^\#), \\ u^2(A_N^\#) &= A_N^\# + 2A_0 \mid \tilde{A}_N := A_N^\#. \end{aligned}$$

Bei der Bestimmung der Unsicherheit der Nachweisgrenze wird geschätzt, dass der wahre Wert gleich der Nachweisgrenze sei; dadurch wird die Unsicherheit überschätzt.

$$A_N^\# = A_N^* + k_{1-\beta} \cdot \sqrt{A_N^\# + 2A_0}.$$

Durch Gleichsetzen von  $k_{1-\alpha}$  und  $k_{1-\beta}$  und Substitution mit  $k$ , sowie Einsetzen der Gleichung der Erkennungsgrenze und Umformen ergibt sich für die Nachweisgrenze  $A_N^\# = k^2 + 2A_N^*$ . Im vorliegenden Algorithmus ist für  $\alpha = 0,05$  gewählt, daraus folgt für das  $(1 - \alpha)$ -Quantil  $k_{1-\alpha} = 1,65$ . Dies ist in Übereinstimmung mit den Empfehlungen der bereits genannten Normen.

---

## 5. Technische Grundlagen der Implementierung

---

### 5.1. Wahl der Programmiersprache

Für die Implementierung des Algorithmus wurde eine objektorientierte Programmiersprache gesucht, die auf möglichst vielen Computersystemen lauffähig ist und die Schnittstellen zur Programmierung graphischer Oberflächen zur Interaktion mit dem Benutzer bereitstellt. Favoriten hierbei sind Programmierumgebungen, die aus dem Quelltext einen Bytecode erzeugen, der auf der Zielmaschine interpretiert, sowie bei Bedarf fertig kompiliert und ausgeführt werden kann. Vorteil ist, dass dadurch eine Hardwareunabhängigkeit gesichert wird, ein großer Nachteil jedoch, dass durch die Interpretation und eventuelle Kompilierung zusätzliche Rechenzeit aufgewandt werden muss und es dadurch zu einer verlangsamten Ausführung gegenüber kompilierten Quelltextes kommt. Die wichtigsten Sprachen, die Bytecode erzeugen, sind die des Microsoft .Net Framework und Java.

Die Wahl der Programmiersprache zur Entwicklung des Algorithmus fiel auf Microsoft .Net C#. Mit dieser Programmiersprache entwickelte Algorithmen sind auf quasi allen Rechnern mit Microsoft Betriebssystem zusammen mit dem .Net Framework lauffähig. Die Möglichkeit, Java einzusetzen, wurde verworfen, da zwar mit dieser Sprache entwickelte Algorithmen auch auf Rechnern ohne Microsoft Betriebssystem funktionsfähig sind, jedoch Erweiterungsbibliotheken und externe Module hardware- bzw. betriebssystemspezifisch erstellt werden müssten.

Weiterhin sind für .Net Anwendungen viele Komponenten verfügbar, die bereits getestet und fehlerfrei sind. Zusätzlich ist mit dem Microsoft Visual Studio 2010 eine ausgereifte Programmierumgebung verfügbar, die neben der Programmierung auch ein ausgefeiltes Debugging erlaubt.

### 5.2. Ausführungsebenen

Der Algorithmus ist in drei gekapselte Bereiche aufgeteilt, der visuellen Oberfläche, der Ausführungsebene und der Zusatzbibliotheken (vgl. Abbildung 4). Durch Kapselung sind Daten des jeweiligen Bereiches nur in ihrem Gültigkeitsbereich verfügbar oder über definierte Schnittstellen untereinander. Durch diesen Aufbau des Algorithmus ist genau festgelegt, welche Daten in welchem Bereich zu finden sind. Diese Konstruktion der Datenstruktur legt die Basis für modulare Erweiterungen, so dass bei Programmierung neuer Module auf die bestehenden Daten zugegriffen werden kann und durch Programmierung weiterer Schnittstellen neue zur Verfügung gestellt werden können.

Berechnungen werden in der Ausführungsebene im Backend unter Berücksichtigung von Parametern, die über die visuelle Oberfläche im Frontend festgelegt werden, getätigt.

# Algorithmus

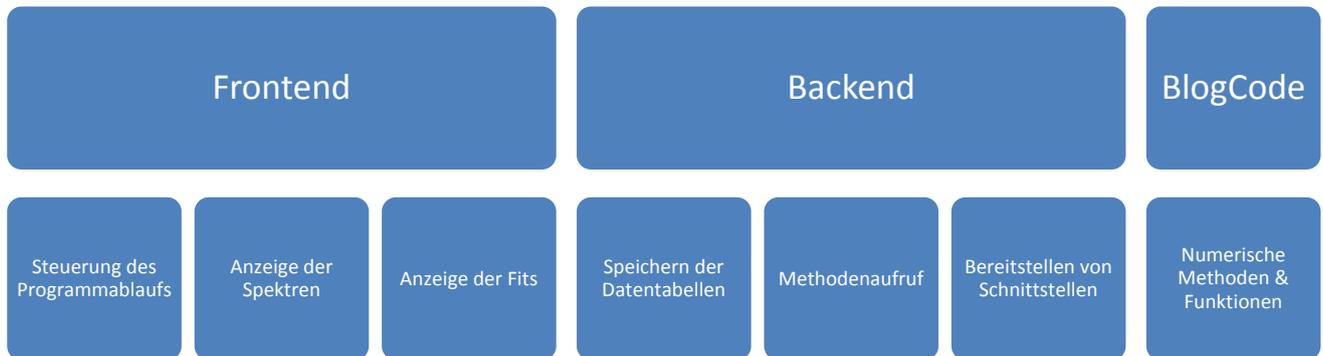


Abbildung 4:

Im Blockdiagramm wird die Softwarearchitektur des Algorithmus dargestellt. Unter der Aufteilung in Frontend, Backend und BlogCode sind die jeweils implementierten Funktionen zusammengefasst.

## 5.2.1. Frontend

Das Frontend ist die visuelle Schnittstelle, über die der Benutzer mit dem Algorithmus interagiert. Dazu gehört zum Einen, dass hier im Backend gespeicherte Daten und Ergebnisse von Berechnungen angezeigt werden, zum Anderen werden über diese Schnittstelle Eingaben des Benutzers an die Ausführungsebene weitergegeben und Methoden aufgerufen.

Um die allgemeine Forderung nach einem robusten Algorithmus im Frontend zu genügen, wurde, soweit möglich, auf bereits erprobte Komponenten zurückgegriffen. Zur Anzeige der Spektren wurde die im Microsoft.Net Framework 3.5 enthaltene MSChart-Komponente benutzt, die im Namespace `System.Windows.Forms.DataVisualization.Charting` implementiert ist. Darin sind bereits verschiedene Methoden zur Anzeige von Daten, die in den üblichen Datenstrukturen gespeichert sind, fehlerfrei programmiert und optimiert. Im Wesentlichen stellt diese Komponente Diagramme, wie sie auch in Microsoft Excel angezeigt werden, dar und bietet vergleichbare Möglichkeiten zur Formatierung der Ausgabe, mit dem Unterschied dass ein programmatischer Zugriff auf die Funktionen möglich ist.

Zur Anzeige der Tabelle im Frontend wurde ein `DataGridView` verwendet, welches eine direkte Datenanbindung an eine `DataTable` im Backend hat. Diese Komponente ist in der Standardschnittstelle `System.Windows.Forms` geschwindigkeitsoptimiert und fehlerfrei integriert.

Durch die Auswahl von erprobten Standardkomponenten ist sichergestellt, dass es nicht zu Programmabstürzen aufgrund fehlerhafter Programmierung im Frontend kommen wird.

## 5.2.2. Backend

Im Backend des Algorithmus werden die zur Ausführung benötigten Daten gespeichert, sowie die Datenoperationen aufgerufen beziehungsweise ausgeführt. Desweiteren werden von hier aus mathematische Methoden aus Klassenbibliotheken aufgerufen und die Ergebnisse in den bestehenden Datenstrukturen gespeichert.

Für das Vorhalten der Daten ist in der Klasse „Daten“ ein `DataSet` implementiert, in dem in den drei Tabellen „Daten“, „Regression“ und „ShowData“ vom Typ `DataTable` alle für die Ausführung des Algorithmus benötigten Daten gespeichert werden. Das Initialisieren des Backend geschieht über die Erzeugung einer neuen Klasse vom Typ „Daten“ im Frontend. Nach dem Aufruf der darin enthaltenen Funktion „InitData“ ist das `DataSet` initialisiert und alle Tabellen erzeugt. Als nächstes müssen in die Tabelle „Daten“ Messdaten aus der einzulesenden Datei gespeichert werden, so dass diese allen

---

Methoden und Funktionen zur Verfügung stehen. Aus diesem Grund sollten alle Erweiterungen, sowie der Aufruf von neuen Modulen, in dieser Klasse implementiert werden.

Im Backend wurde, soweit möglich, auf die Einführung eigener Datentypen verzichtet. Stattdessen werden die aus der unter der Microsoft Public License (Ms-PL) veröffentlichten Klassenbibliothek BlogCode [7], die numerischen Berechnungsmethoden enthält, verwendet. Obwohl diese Bibliothek nicht in dem Umfang, wie zum Beispiel Microsoft Produkte, getestet ist, scheint dieser nach ausführlichen Tests, die typische Operationen abdecken, ausreichend robust. Datentypen, die Matrizen oder Parameter einer zu fittenden Funktion enthalten, haben weder Auffälligkeiten noch Fehler gezeigt. Einzige Auffälligkeit während der Tests war der Marquardt-Levenberg Algorithmus selbst, der bei dem Versuch Modellfunktionen an wenigen Datenpunkten anzupassen, mit einer Fehlermeldung abgebrochen worden ist. Nach einer Untersuchung stellte sich jedoch heraus, dass nicht der Algorithmus selbst fehlerhaft ist, sondern die Routine zur Lösung eines linearen Gleichungssystems mithilfe der Singulärwertzerlegung. Eine Auswirkung auf die Funktionsfähigkeit des Algorithmus als Ganzes ist dadurch nicht gegeben, durch Veränderung des Bereiches der für den Ausgleich verwendeten Datenpunkte kann diese Fehlermeldung umgangen werden. Ein Test der im Backend implementierten Methoden verlief zufriedenstellend und zeigte keine Fehlfunktionen auf. Somit kann auch das Backend und die Klassenbibliothek BlogCode als ausreichend robust angesehen werden.

### 5.3. Funktionsumfang

Der Algorithmus in seiner derzeitigen Ausbaustufe hat alle notwendigen Datenstrukturen, auch für Erweiterungen, implementiert, so dass weitere Funktionen direkt ergänzt werden können. Der Benutzer kann Spektren aus Textdateien einlesen, wobei der Algorithmus zeilenweise die Impulse im jeweiligen Kanal verarbeitet. Im angezeigten Spektrum können daraufhin Markierungen gesetzt und entfernt werden, sowie in den ausgewählten Bereichen Fits durchgeführt. Die Ergebnisse werden über die Laufzeit gespeichert und tabellarisch angezeigt. Aus diesem Speicher können einzelne Fits auch wieder gelöscht, beziehungsweise die Anzeige als Ganzes in einer CSV-Datei zur Weiterverarbeitung und Dokumentation gespeichert werden. Die Anzeige des Spektrums enthält eine logarithmische und lineare Skalierung.

In der Ausführungsebene sind die mathematischen Funktionen in C# programmiert. Diese enthält neben Definitionen für algorithmusspezifische Datentypen wie Matrizen, Parameter oder Tabellen den Fitalgorithmus, Levenberg-Marquardt, selbst.

#### 5.3.1. Einlesen von Daten

Das Einlesen der Daten aus Textdateien wird mit der Funktion `ReadData(int abZeile, string FileName)` im Backend realisiert. Beim Aufruf dieser aus dem Frontend müssen als Parameter die Zeile, ab der Daten eingelesen werden sollen, und der vollständige Pfad zur Datei angegeben werden.

Das Eingabeformat ist derzeit auf Dateien beschränkt, in deren Zeilen Impulswerte gespeichert sind. Ab der angegebenen Zeile werden die eingelesenen Werte als ganze Zahlen ab Kanal null fortlaufend zeilenweise eingelesen und in `DataTable("Daten")` gespeichert. Zur Anzeige auf dem Frontend muss dort noch die Prozedur `GraphenZeichnen()` aufgerufen werden. Um andere Datenformate, wie zum Beispiel mehrspaltige CSV-Dateien, einzulesen, muss lediglich diese Funktion um eine Abfrage des Formats und die Einleseroutine selbst erweitert werden.

---

### 5.3.2. Anpassen von Funktion im markierten Bereich

Um eine Funktion in einem markierten Bereich auszugleichen, müssen mehrere Schritte nacheinander durchgeführt werden. Deshalb werden diese im Backend in einer Methode zusammengefasst und über das Frontend aufgerufen.

1. Einlesen der zum Ausgleich benötigten Daten aus `DataTable("Daten")`
2. Initialisieren der Ausgleichsparameter
3. Festlegen der Variable
4. Initialisieren der Modellfunktion
5. Erzeugen eines `LevenbergMarquardt` Objekts mit dem Datenmaterial aus 1., den Parametern aus 2. und 3., sowie der Modellfunktion
6. Ausführen der Iterationsschritte

Die nötigen Schritte werden beispielhaft an der bereits implementierten Methode `StartRegression(bool FitWithBGD, int Steps)` beschrieben, die Parameter geben an, ob der Untergrund mitbrachtet werden soll und wie viele Iterationsschritte der Marquardt-Levenberg Algorithmus durchgeführt.

Als erstes wird geprüft, ob vier Markierungen gesetzt wurden. Dies ist Voraussetzung um den Randbereich und den Datenbereich zu kennen. Im ersten Schritt wird ein zwei dimensionales Datenfeld erzeugt, indem die Kanäle mit den dazugehörigen Impulsen gespeichert sind. Mit den Daten aus dem Datenfeld werden Startparameter gewonnen, mit denen die auszugleichenden Parameter der Untergrundfunktionen bestimmt werden. Nachdem ein mathematisches Modell der Untergrundfunktion zugewiesen wurde und alle Parameter zu einem Parameterfeld zusammengefasst, wird daraus ein `LevenbergMarquard` Objekt erstellt und die festgelegten Iterationsschritte ausgeführt. Zum Abschluss des Ausgleichs werden die Unsicherheiten bestimmt.

Im zweiten Schritt wird ein Datenfeld erzeugt, indem die Kanäle und zugehörigen Impulse aus der zentralen Tabelle ausgelesen werden und bei Beachtung des Untergrundes hiervon der Untergrund anhand der vorher bestimmten Funktion abgezogen wird. Mit diesem Datenfeld werden die Startwerte der Parameter für die Gaußverteilung geschätzt. Nach dem Initialisieren der Ausgleichsparameter, der Variablen und der Funktion wird wiederum ein `LevenbergMarquard` Objekt erstellt und Iterationsschritte durchgeführt und abgeschlossen.

Zum Abschluss werden alle relevanten Daten dieses Vorgangs (Parameter, Funktionen, Untersuchungsbereich) mit der Funktion `AddRegression(...)` der zentralen Tabelle der durchgeführten Fits hinzugefügt und mit dem Aufruf `ShowGridData()` die Einträge in der Datentabelle, die auch im Frontend angezeigt wird, vorgenommen.

Um Anpassungen an der bereits implementierten Ausgleichsroutine vorzunehmen, musste diese Methode im Quelltext vom Backend geändert werden. Falls davon unabhängig weitere Modellfunktionen implementiert werden sollen, reicht es aus, analog hierzu eine Methode zu implementieren, die die gewünschte Modellfunktion enthält, und diese vom Frontend aus aufzurufen.

### 5.3.3. Dateiausgabe

Für die Dateiausgabe ist eine Funktion implementiert, die die angezeigten Werte aus der Tabelle im Frontend in eine CSV-Datei speichert. Dazu wird zeilenweise durch die Spalten iteriert und diese Werte kommasepariert in eine ASCII Datei geschrieben.

## 6. Bedienung

### 6.1. Allgemeine Bedienung

Die Oberfläche des Programms ist in drei Bereiche aufgeteilt, einen Anzeigebereich des Spektrums, der Fit-Parameter und der Kontrollbereich.

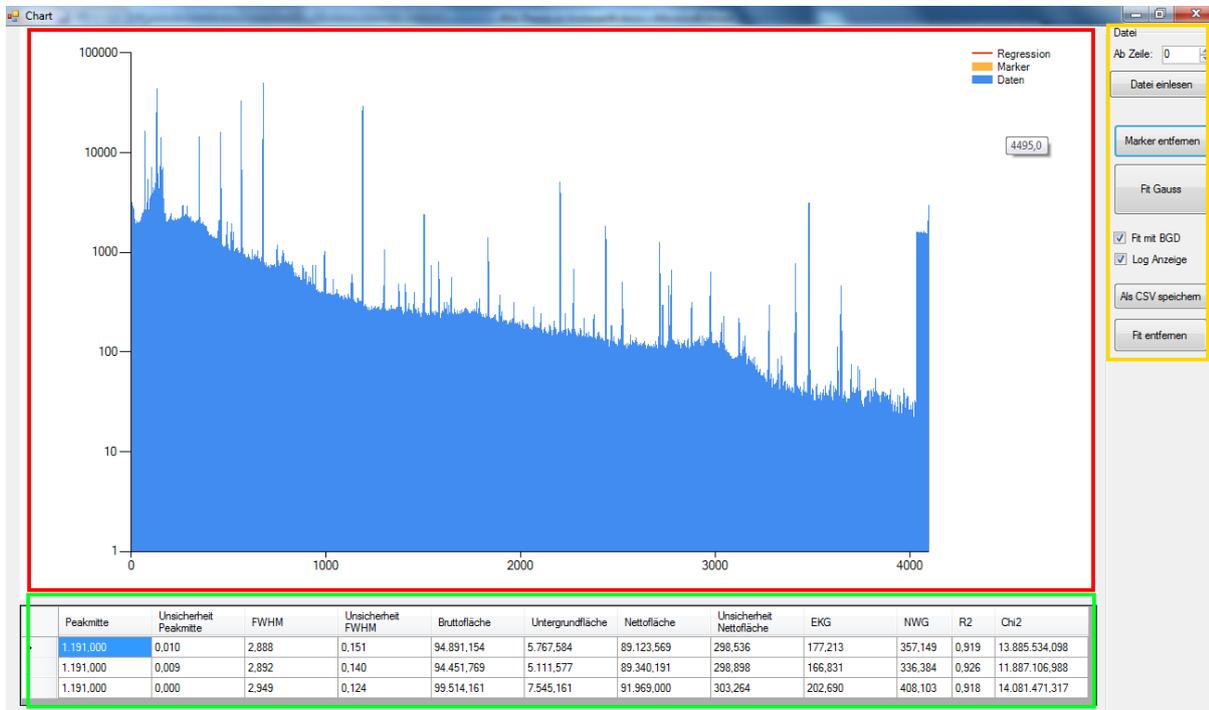


Abbildung 5:

Die Abbildung zeigt die Programmoberfläche mit farbigen Markierungen zur Visualisierung der zwei Anzeigebereiche für Spektrum und Fitparameter, sowie dem Kontrollbereich.

#### 6.1.1. Anzeigebereich des Spektrums

Nachdem ein Spektrum eingelesen wurde, wird es in diesem Bereich angezeigt. Auf der positiven x-Achse befinden sich die Kanäle, auf der y-Achse sind die Impulse aufgetragen. Das gemessene Spektrum wird durch ein blau eingefärbtes Balkendiagramm dargestellt. Mit der linken Maustaste können, durch Gedrückthalten und Aufziehen, Bereiche ausgewählt werden, die vergrößert dargestellt werden sollen. Mit der rechten Maustaste werden Markierungen gesetzt und beim Klicken auf bereits gesetzte wieder entfernt. Markierungen sind gelb eingefärbt und erscheinen im Balkendiagramm der Messdaten. Nachdem ein Fit ausgeführt oder zur Anzeige ausgewählt wurde, wird dieser rot eingefärbt über die Messdaten gelegt.

#### 6.1.2. Anzeigebereich der Fitparameter

In diesem Bereich werden bereits berechnete Fits mit allen relevanten Parametern tabellarisch angezeigt. Durch Anklicken des Zeilenkopfes auf der linken Seite wird im Anzeigebereich des Spektrums die zu diesem Fit gehörige Markierung und Fitfunktion gezeichnet. Die angezeigten Parameter sind zum einen die Parameter aus der Modellfunktion der Verteilung mit ihren berechneten Unsicherheiten und die sich aus dem markierten Bereich ergebenden Flächeninhalte und der daraus berechneten Nachweis- und Erkennungsgrenzen, als auch das Bestimmtheitsmaß und  $\chi_{red}^2$ .

---

### 6.1.3. Kontrollbereich

Von hier aus können alle wesentlichen Funktionen des Algorithmus aufgerufen werden. Dazu gehört das Einlesen von Dateien und Speichern der Ergebnisse, das Ausführen eines Fits und Entfernen eines solchen aus dem Anzeigebereich, das Entfernen von gesetzten Markierungen und das Umschalten der Anzeige auf logarithmische Darstellung.

### 6.2. Datenbereich der Ausgleichsrechnung

Bevor der Algorithmus Untersuchung an Messdaten vornehmen kann, muss der Benutzer den zu untersuchenden Datenbereich angeben. Dieser teilt sich in zwei gesonderte Bereiche auf, zum einen den Randbereich aus dem der Untergrund geschätzt wird, zum anderen der Bereich, in dem die Linie vorliegt. Der Algorithmus berechnet aus den Randbereichen die Parameter der Ausgleichsfunktion des Untergrundes unter der Bedingung des Wendepunkts innerhalb des Hauptbereichs, subtrahiert die entsprechenden Funktionswerte aus dem zu untersuchenden Bereich und berechnet die Ausgleichsfunktion der Verteilung. Alle Bereiche zusammen müssen ein zusammenhängendes Intervall ergeben. Dieses wird über vier Markierungen definiert, jeweils eine für Anfang und Ende des Intervalls und zwei dazwischenliegende für Beginn und Ende des Hauptbereiches. Wichtig ist, dass möglichst wenige Messdaten aus der Glockenkurve dem Randbereich zugeordnet werden, um das Ergebnis der Anpassung nicht zu verfälschen, wobei zu beachten ist, dass die Bereichsauswahl eine durch den Benutzer vorgenommene Schätzung ist, die sich auf die Berechnung der charakteristischen Grenzen auswirken kann. Die Größe der Randbereiche sollte sich grob am Hauptbereich orientieren, da aus dem Betrag des gesamten Intervalls die Startparameter für den Fit bestimmt werden. Der Hauptbereich sollte die  $3\sigma$ -Umgebung der Glockenverteilung umfassen und die gleiche Breite haben wie die beiden Randbereiche zusammen.

### 6.3. Anpassen von Linien

Im ersten Schritt müssen vier Markierungen zur Beschreibung der Randbereiche und des Linienbereichs gesetzt werden. Die beiden Markierungen um die Linie herum sollten den wesentlichen Teil der Linie selbst begrenzen, dies entspricht in etwa der  $3\sigma$ -Umgebung, in welcher 99,7% der erwarteten Messwerte liegen. Der Randbereich links und rechts sollte so begrenzt werden, dass daraus der Verlauf des Untergrundes erkennbar ist. Für den Algorithmus ist es wichtig, ausreichend Datenpunkte für die Bestimmung des Untergrundes zugewiesen zu bekommen, da bei einer mangelhaften Anpassung aufgrund zu weniger Datenpunkte sich die Unsicherheit durch den Untergrundabzug direkt in der Qualität der vom Untergrund bereinigten Daten niederschlägt. Je genauer der Untergrund bestimmt wurde, desto genauer sind die Daten zur Bestimmung der Linie selbst.

Nachdem ein Fit durchgeführt wurde und die Gesamtfunktion, bestehend aus Modellfunktion der Linie und des Untergrundes, angezeigt, kann durch Versetzungen einzelner Markierung, aus denen sich die Startbedingungen ableiten, Einfluss auf das Fitergebnis genommen werden. Um verschiedene Anpassungen miteinander vergleichen zu können, wird jeweils das Bestimmtheitsmaß und  $\chi_{red}^2$  („Chi2“), also die durch das Modell nicht erklärten Abweichungen der Datenwerte von den Funktionswerten der Modellfunktion, dargestellt. Bei  $\chi_{red}^2$  ist zu beachten, dass dieses direkt nur im Bezug zu einem Ausgleich derselben Linie eine Aussage besitzt. Jedoch können damit als Grundlage bei Bedarf weitere statistische Tests durchgeführt werden. Auf diese Problematik, auch dass  $\chi_{red}^2$  betragsmäßig groß sein kann, wird in [3] näher eingegangen und Interpretationsmöglichkeiten aufgezeigt.

---

## 7. Ergebnisse

---

Um zu einer Bewertung der Leistungsfähigkeit des Algorithmus zu gelangen wird dieser beispielhaft bei der Auswertung bekannter Vielkanalspektren eingesetzt. Es soll untersucht werden, ob dieser bereits in der jetzigen Entwicklungsstufe zuverlässig bei Auswertungen genutzt werden kann. Dazu wird zum Einen ein selbst generiertes Spektrum analysiert, bei dem die Lage, die Halbwertsbreite und die Nettofläche bekannt ist, sowie zum Anderen ein reales Spektrum aus einem Leistungstest zur Bestimmung von natürlichen Radionukliden in Phosphorgips [8]. Für diese Untersuchung wurde die Anzahl der Iterationsschritte im Marquart-Levenberg Algorithmus auf 10000 festgesetzt, der verwendete Rechner verfügte über einen 1,3 GHz Intel Core 2 Duo Prozessor (SU7300) und 4 GB Arbeitsspeicher.

### 7.1. Untersuchung eines generierten Spektrums

Das untersuchte Spektrum wurde in Microsoft Excel erzeugt und zeichnet sich durch die sieben bekannte Linien aus Tabelle 1 mit unterschiedlicher Höhe und einem Untergrundrauschen aus, welches so gewählt wurde, dass es einige Linien verdeckt.

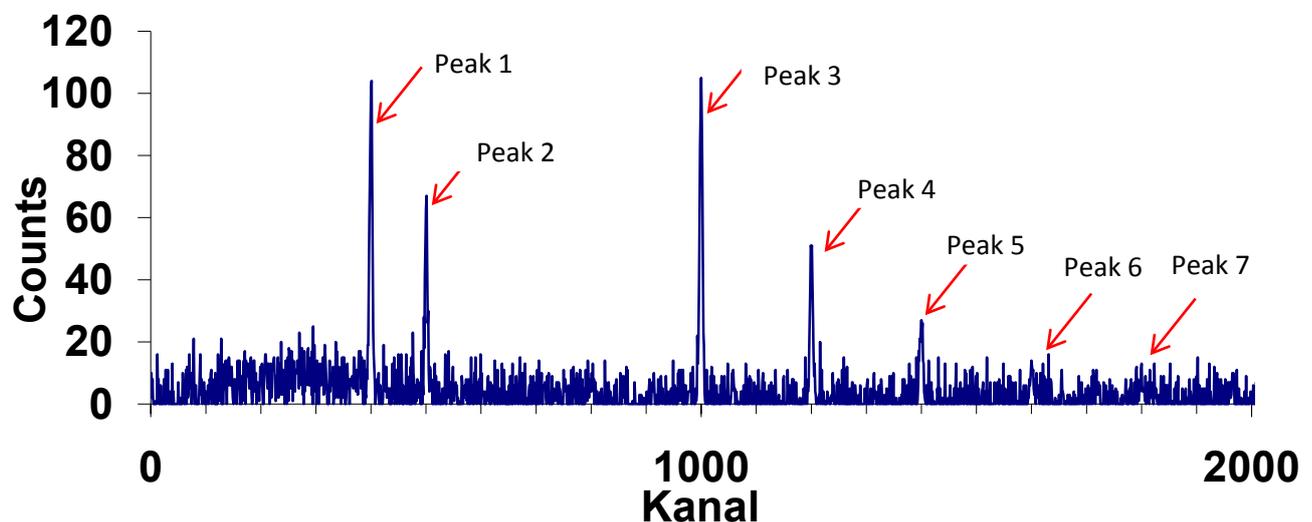


Abbildung 6:

Das für die Untersuchung erzeugte Spektrum enthält definierte Linien, an denen nach Entfaltung das Verhalten des Algorithmus bewertet werden kann.

Um das Verschwinden von Linien im Untergrund zu verdeutlichen wird als Peak zu Untergrundverhältnis (PzU) der Quotient  $\frac{\text{Amplitude}}{3\sigma_{\text{Untergrund}}}$  eingeführt, wobei die Amplitude die Differenz des Linienmaximums und des durchschnittlichen Untergrunds ist. Da das Untergrundrauschen normalverteilt um dessen Mittelwert ist, ist zu erwarten, dass alle Ereignisse, die sich in der  $3\sigma$  – Umgebung befinden mit 99,7% Wahrscheinlichkeit dem Untergrund zuzuordnen sind. Dies entspricht i.w.S. auch einer Signifikanzprüfung, bei der überprüft wird, ob die zum Ausgleich verwendeten Daten überhaupt einer Linie zugeordnet werden können oder diese mit großer Wahrscheinlichkeit dem Untergrund angehören [2, S.64].

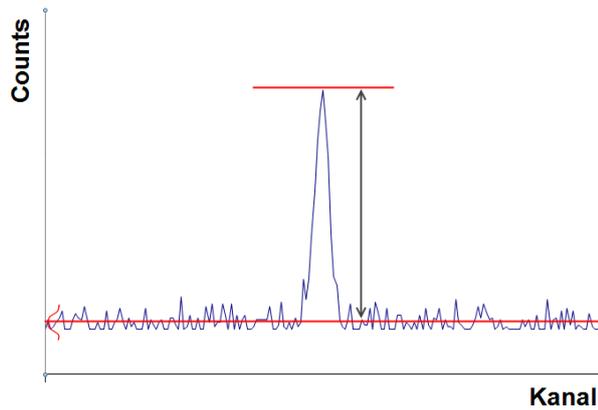


Abbildung 7:  
Schematische Darstellung des Peak zu Untergrundverhältnis (PzU) [2]

Somit sollten alle Anpassungen an Linien, die sich nicht in diesem Bereich befindet, also deren  $PzU \gg 1$ , mit hohem Bestimmtheitsmaß durchgeführt werden, diese Vermutung wird überprüft. Der Mittelwert und die Standardabweichung des Rauschens wurden aus je 50 Kanälen ober- und unterhalb der jeweiligen Linie bestimmt.

Tabelle 1:  
In der Tabelle sind die im generierten Spektrum enthaltenen Linien inklusive deren Parameter aufgeführt.

	HWB	Lage	Amplitude	Nettofläche	$\frac{\text{Amplitude}}{3\sigma_{\text{Untergrund}}}$
Peak 1	5,4	400	100	570	7,00
Peak 2	5,5	500	50	290	3,48
Peak 3	5,9	1000	100	627	8,03
Peak 4	6,1	1200	50	324	3,96
Peak 5	6,3	1400	25	167	2,19
Peak 6	6,4	1600	12	82	1,02
Peak 7	6,6	1800	6	42	0,62

Um zu einer differenzierten Aussage zu kommen und auch den Einfluss des Benutzers auf die Güte der Anpassung zu berücksichtigen, wurde an den jeweiligen Stellen im Spektrum für verschiedene, sinnvoll gesetzte Grenzen gefittet und das erhaltene Ergebnis gemittelt und mit den Sollwerten verglichen.

Tabelle 2:  
Vergleich der Soll und Ist Werte der Lage und HWB im künstlich generiertem Spektrum mit Angabe der relativen Abweichung.

	Lage Soll	Lage Ist	relative Abw.	HWB Soll	HWB Ist	relative Abw.
Peak 1	400,000	399,086	-0,002	5,360	5,133	-0,042
Peak 2	500,000	499,209	-0,002	5,450	5,391	-0,011
Peak 3	1000,000	998,674	-0,001	5,900	6,003	0,017
Peak 4	1200,000	1199,211	-0,001	6,080	6,789	0,117
Peak 5	1400,000	1399,124	-0,001	6,260	4,835	-0,228
Peak 6	1600,000	1599,462	0,000	6,440	5,171	-0,197
Peak 7*	1800,000	-	-	6,620	-	-

Tabelle 3:

Vergleich der Soll und Ist Werte der Nettofläche im künstlich generiertem Spektrum, sowie Angabe der EKG, NWG und relative Abweichung

	Nettofläche Soll	Nettofläche Ist	relative Abw.	EKG(Nettofläche)	NWG(Nettofläche)
Peak 1	570,000	547,701	-0,039	32,689	68,101
Peak 2	290,000	298,532	0,029	27,182	57,087
Peak 3	627,000	663,531	0,058	8,737	20,196
Peak 4	324,000	364,226	0,124	21,604	45,931
Peak 5	167,000	196,181	0,175	13,848	30,418
Peak 6	82,000	69,378	-0,154	17,162	37,046
Peak 7*	42,000	-	-	-	-

Aus Tabelle 2 wird direkt sichtbar, dass die Halbwertsbreite nach der Anpassung einen relativen Fehler aufweist, der abhängig vom gewählten Untersuchungsbereich bis zu 25% betragen kann. Die relative Abweichung der Peaklage hingegen befindet sich im Promillebereich und ist somit als ausreichend genau anzusehen. Zur Berechnung der charakteristischen Grenzen sind bei dieser Untersuchung besonders die in Tabelle 3 aufgeführten Nettoflächen interessant, da diese direkt verwendet werden, während hingegen der Untergrund erst numerisch integriert werden muss. Bis auf Peak 7, der vollständig im Rauschen verschwindet, konnte eine nur geringe Abweichung von bis zu 10,5% (Peak 4) festgestellt werden.

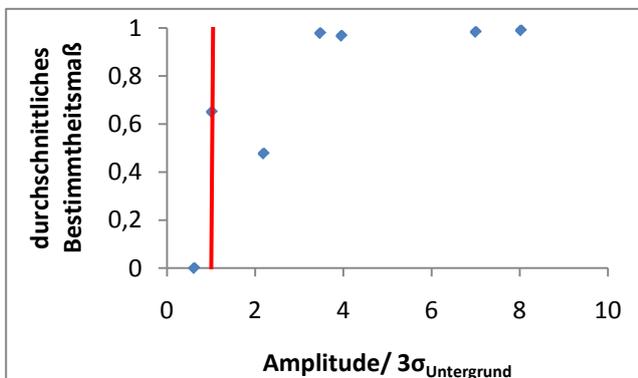


Abbildung 8

Auftrag des Bestimmtheitsmaßes über dem Quotienten  $\frac{\text{Amplitude}}{3\sigma_{\text{Untergrund}}}$  mit den Daten aus Tabelle 4.

Tabelle 4:

Zeigt die im erzeugten Untersuchungsspektrum enthaltenen Linien mit dem jeweiligen  $\frac{\text{Amplitude}}{3\sigma_{\text{Untergrund}}}$  und durchschnittlichen Bestimmtheitsmaß nach dem Ausgleich.

	Maximum	Amplitude $\frac{\text{Amplitude}}{3\sigma_{\text{Untergrund}}}$	$\overline{R^2}$
Peak 1	100	7,0	0,983
Peak 2	50	3,5	0,978
Peak 3	100	8,0	0,990
Peak 4	50	3,9	0,967
Peak 5	25	2,2	0,477
Peak 6	12	1,0	0,651
Peak 7	6	0,6	0,000

Der eingangs vermutete Zusammenhang des Verhältnisses der Amplitude und der Standardabweichung des Untergrundes in Bezug auf die Auswertbarkeit scheint bestätigt. In Tabelle 4 und Abbildung 8 ist deutlich zu erkennen, dass bei Amplituden, deren Abstand zum Untergrund groß gegenüber der Standardabweichung desselben ist, der Ausgleich sehr gut gelingt und das durchschnittliche Bestimmtheitsmaß  $\overline{R^2} > 90\%$ . Daraus lässt sich folgern, dass Linien, die sich so vom Untergrund abheben, ausgleichen lassen und die auf dieser Berechnung aufbauenden charakteristischen Grenzen aussagekräftig sind. Im Umkehrschluss bedeutet dies jedoch, dass charakteristische Grenzen, die auf einem Ausgleich mit einem Bestimmtheitsmaß von  $R^2 \leq 90\%$  basieren genauer hinterfragt werden

müssen. Besonders muss der Benutzer selbst entscheiden, ob die berechnete Ausgleichsfunktion ausreichend scheint, um als Grundlage für weitere Berechnungen zu dienen.

## 7.2. Untersuchung des IAEA2008 Spektrums

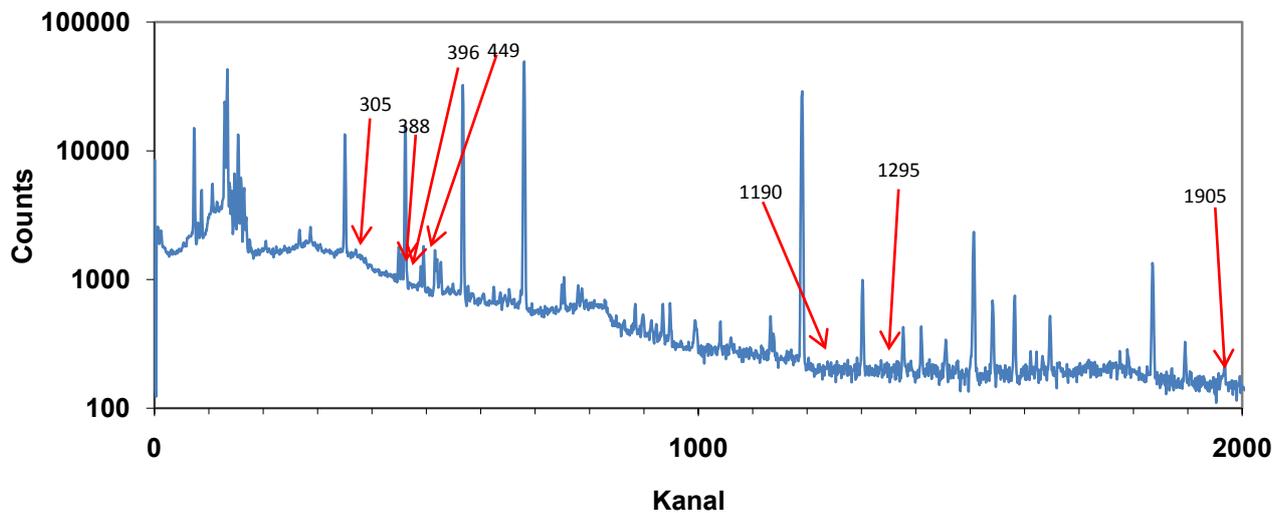


Abbildung 9:  
Reales Spektrum, bei dem mit Germanium-Detektoren  $\gamma$ -Zerfälle aufgenommen wurden.

Das IAEA-2008-03 Phosphogypsum Spektrum ist am Zentrum für Strahlenschutz und Radioökologie in Hannover mit dem Germanium-Detektor „Liese“ im Rahmen der Diplom-Arbeit[9] von Jonny H. Birkhan aufgenommen und mit der Software GammaW der Firma Dr. Westmeier GmbH ausgewertet worden. Um eine Aussage zur Anwendbarkeit der Ergebnisse des Algorithmus zu erhalten, werden mit diesem am selben Spektrum einige Linien einschließlich derer charakteristischen Grenzen untersucht und mit den Ergebnissen von GammaW verglichen.

Tabelle 5:  
Kennzahlen ausgewählter Linien aus dem IAEA08 Spektrum.

Spektrum		Gamma-Katalog des BfS		$A_{netto}$
Kanal	E (keV)	Nuklid	E (keV)	
305	163,4	U-235	163,33	514,4
388	205,2	U-235	205,31	670,2
396	209,2	Ac-228	209,25	301,1
449	235,9	Th-227	235,97	1970,7
1190	609,0	Bi-214	609,31	89448,7
1295	661,8	Cs-137	661,66	296,4
1905	968,9	Ac-228	968,97	242,9

Während der Untersuchung fällt direkt auf, dass das gezielte Arbeiten in einem realen Spektrum mit dem derzeitigen Entwicklungsstand eine Herausforderung darstellt. In diesen Vielkanalspektren ist nicht davon auszugehen, dass die Messdaten für eine Anpassung so vorliegen, dass sie nach einer einfachen Untergrundsubtraktion isoliert betrachtet werden können. Vielmehr kommt es häufig vor, dass im Randbereich um den Untersuchungsbereich herum, bereits weitere Linien oder Linienausläufer zu finden sind. Dies führt direkt zur Problematik, dass bei Linien, die nicht signifikant sind, oftmals nicht genügend Datenpunkte vorliegen.

Ein Lösungsansatz an dieser Stelle ist eine Modellfunktion auszugleichen, die mehrere Linien entfaltet oder einen trivialen Ansatz für die Bestimmung des Untergrundes. Durch Annahme einer konstanten oder linearen Funktion, die den Untergrund beschreibt, kann in einem schweren Umfeld dennoch ein Ergebnis erzielt werden. Vor allem könnte der Untergrund aus wenigen Datenpunkten geschätzt werden und müsste nicht aufwändig und mit zusätzlichen Unsicherheiten behaftet, die sich in der Modellfunktion der Linie fortsetzen, bestimmt werden.

Der vorliegende Algorithmus leidet darunter, dass die integrierten Routinen zur Berechnung von Ausgleichsfunktionen relativ viele Datenpunkte benötigen, um ein Ergebnis zu erzielen, wenn die Modellfunktion nicht gut passt. Dies führt besonders in Bereichen, in denen mehrere Linien dicht beieinander liegen und somit kein geeignet breiter Randbereich vorliegt, zu nicht ausreichend genauen Ausgleichsfunktionen. Dadurch sind Nachweis- und Erkennungsgrenzen nicht mehr vertrauenswürdig anzugeben.

Tabelle 6:

Auswertung der Entfaltung mittels des Algorithmus und Vergleich der Ergebnisse mit denen aus GammaW auf Übereinstimmung.

Peak	$A_{\text{Untergrund}}$	$A_N$	EKG	NWG	EKG	NWG	Relative Abw. $A_N$
305,08	22564,88	645,48	350,52	703,77	>	<	0,25
388,30	12051,51	291,68	256,16	515,05	>	<	-0,56
396,00	-	-	-	-	-	-	-
449,00	-	-	-	-	-	-	-
1191,00	6100,15	91969,00	182,25	367,22	>	>	0,03
1295,00	-	-	-	-	-	-	-
1904,00	2295,00	224,09	111,79	226,30	>	<	-0,08

In Tabelle 6 sind die Ergebnisse aus der Untersuchung des IAEA2008 Spektrum zusammengefasst, für einige Linien konnte keine Ausgleichsfunktion bestimmt werden. Obwohl im Algorithmus noch kein zufriedenstellendes Verfahren für die Analyse von dicht beieinander liegenden Linien implementiert ist, konnten die charakteristischen Grenzen im Falle einer akzeptablen Ausgleichsfunktion berechnet werden. Die relative Abweichung der bestimmten Nettoflächen der ausgeglichenen Linien gegenüber den mit GammaW berechneten scheint noch groß ( $> 10\%$ ), jedoch konnte trotzdem bei Vorliegen einer Ausgleichsfunktion die untersuchte Linie erkannt werden. Dies legt es nahe, dass mit entsprechenden Anpassungen und einer dadurch gegebener Verbesserung ein konkurrenzfähiger Algorithmus programmiert wurde.

### 7.3. Schätzung des Untergrundes in schwierigen Bereichen

Falls nicht ausreichend viele Datenpunkte zur Schätzung des Untergrundes vorliegen, bzw. dieser nicht isoliert vorliegt, müsste der Untergrund geschätzt werden. Dazu könnte aus den Funktionswerten an den Grenzen des Innenbereichs eine (konstante oder lineare) Untergrundfunktion bestimmt werden, die auf den gesamten Bereich interpoliert wird. Dieses Vorgehen ist durch die Interpolation natürlich mit großen Unsicherheiten behaftet, was sich auf Grund der Schätzung jedoch nicht vermeiden lässt, ohne ausgeprägte Randbereiche zu betrachten. In [6] wird im Anhang C die Verwendung geeigneter Untergrundfunktionen diskutiert. Dazu wird der Randbereich rechts und links der Verteilung in jeweils zwei gleichgroße Bereiche geteilt und hierin ein Polynom  $n$ -ten Grades eingepasst, welches durch weitere Annahmen ohne Ausgleichsrechnung direkt angegeben wird. Der Grad des Polynoms wird dazu in Abhängigkeit der Umgebung der zu untersuchenden Linie gewählt. Allerdings geht auch dieses Vorgehen nicht auf die Problematik ein, keine ausreichend breite Randbereiche zur Verfügung zu haben.

Ein anderer Ansatz wäre zwar konträr zur Norm, würde aber ein Ausgleich unabhängig von Randbereichen ermöglichen. Grundsätzliche Idee ist, wie bereits am Ende von Kapitel 4.2 angedeutet, die Untergrundfunktion gleichzeitig mit der Verteilungsfunktion auszugleichen. Durch Gleichsetzen des Wendepunktes der Untergrundfunktion mit dem Maximum der mit einem Proportionalitätsfaktor versehenen Normalverteilung (vgl. Kapitel 4.3) könnte eine zusätzliche Randbedingung geschaffen werden, die für  $\gamma$ -Spektroskopie gültig ist [1, S.122]. Die für die Berechnung der charakteristischen Grenzen nötigen Linienflächen ließen sich trotzdem ableiten, der Flächeninhalt der Nettolinie entspricht hier wieder dem Proportionalitätsfaktor der Verteilung, die Bruttolinie würde numerisch integriert und für den Nulleffekt müsste lediglich die Differenz gebildet werden. Bei diesem Vorgehen wäre die Untersuchung einer Linie im Vielkanalspektrum unabhängig von benachbarten, solange diese zwar beliebig nahe zueinander, aber nicht ineinander verschoben wären. Hierfür müsste mit einer auf diesen Fall angepassten Modellfunktion zur Entfaltung gearbeitet werden.

### 7.3.1. Überlegungen zur Dynamisierung von Grenzen des Untersuchungsbereiches

Häufig können Begrenzungen des Untersuchungsbereiches und der Randbereiche nur grob geschätzt werden und sind somit mitverantwortlich für den Erfolg einer Anpassung in diesem Bereich. Ein alternativer Ansatz zur Entfaltung wäre nicht die Grenzen, sondern die Lage anzugeben. Meistens weiß der Benutzer, welche Linie er mit welchem Modell auswerten möchte und kann sie durch sein Vorwissen im Spektrum lokalisieren. Aus dem Wissen um die Form der Linie, z.B. normalverteilt bei  $\gamma$ -Strahlung, und der Form des Untergrundes, im Fall eines  $\gamma$ -Spektrums eine Stufenfunktion zzgl. evtl. Konstante, kann eine Modellfunktion aufgestellt werden, die in einem um den Startwert wachsenden Bereich ausgeglichen wird, solange bis durch weiteres Vergrößern die Güte der Anpassung schlechter wird.

Der Ablauf dieser Berechnungsroutine könnte so aussehen:

1. Setzen der Startparameter (Lage wird durch Benutzer festgelegt; übrige Standardwerte)
2. Festsetzen des Startintervalls  $I_0$  auf Startwert der Lage und die beiden umliegenden Datenpunkte
3. Ausgleichen und Berechnung der Güte  $\chi_0^2$
4. Erweiterung des Intervalls  $I_0$  um einen Datenpunkt nach rechts, Ausgleichen und Berechnung der Güte  $\chi_R^2$  zum Intervall  $I_R$
5. Erweiterung des Intervalls  $I_0$  um einen Datenpunkt nach links, Ausgleichen und Berechnung der Güte  $\chi_L^2$  zum Intervall  $I_L$
6. Erweiterung des Intervalls  $I_0$  um je einen Datenpunkt nach links und rechts, Ausgleichen und Berechnung der Güte  $\chi_{LR}^2$  zum Intervall  $I_{LR}$
7. Falls im Intervall  $I_R, I_L$  oder  $I_{LR}$  die Güte besser ist als im Intervall  $I_0$ , setze  $I_0$  gleich diesem und wiederhole ab Schritt 4
8. Falls keine Verbesserung durch Vergrößerung erreicht wird, Suche beenden
9. Berechnen der Linienflächen und charakteristischen Grenzen

Dieses Verfahren kann nur funktionieren, wenn sichergestellt ist, dass der Ausgleich der Modellfunktion an die Datenpunkte immer konvergiert und im Startintervall nur Ereignisse des Untergrundes und einer, nämlich der gesuchten, Linie enthalten sind.

---

Falls sich die  $3\sigma$ -Umgebung (99,7%) zweier Linien in zwei verschiedenen Intervallen überschneiden, sollte dieses Verfahren die Intervalle mit den dazwischenliegenden Datenpunkten zu einem großen verschmelzen und ab da die Summe der beiden Modellfunktionen ausgeglichen.

Mit diesem Vorgehen würde die Unsicherheit, durch die Wahl der Bereichsgrenzen durch den Benutzer nicht die bestmögliche Ausgleichsfunktion der Linie zu finden, beseitigt und könnte gerade in Spektren mit niedriger Signifikanz das Überprüfen der charakteristischen Grenzen erleichtern.

#### **7.4. Problemfelder in der Implementierung**

In der durchgeführten Testphase ergab sich nicht nur ein Verbesserungspotenzial im Umgang mit dem Untergrund, sondern auch in den Bereichen Rechengeschwindigkeit und Rechengenauigkeit. Die Implementierung des Marquardt-Levenberg Algorithmus in C# weist diesbezüglich deutliche Nachteile gegenüber einer vollständig kompilierten Version in C++, wie sie z.B. in den Numerical Recipes [4] oder der GNU Scientific Library [10] enthalten ist, auf. Besonders die Ausführungsgeschwindigkeit fällt dem Benutzer deutlich auf. Dies ist vor allem auf den vorkompilierten Bytecode zurückzuführen, der wesentlich langsamer ist als eine kompilierte C++ Klassenbibliothek. Aus diesem Grund wurde der Marquardt-Levenberg Algorithmus so implementiert, dass die Anzahl der Iterationsschritte fest vorgegeben sind. Eine Überprüfung, ob der Algorithmus konvergiert, oder eine Mindestgenauigkeit erreicht hat, ist nicht enthalten, da die Ausführungsdauer unverhältnismäßig lang wäre und das Programm nicht mehr sinnvoll zu benutzen. Daraus folgt, dass die Angabe der Unsicherheiten bei schwierigen Fits nur unzureichende Ergebnisse liefert, da der Marquardt-Levenberg Algorithmus innerhalb der vorgegebenen Iterationsschritte nicht zwingend konvergiert. Die Unsicherheiten werden dadurch überschätzt.

Diese Schwierigkeiten ließen sich beheben, indem der Fitalgorithmus in einer C++ Klassenbibliothek implementiert würde und somit über eine Schnittstelle aufgerufen werden könnte, die ähnlich der jetzigen ist. Alternativ kann auch ein entsprechend schneller Prozessor verwendet werden, der mit der implementierten Optimierung für mehrere Prozessorkerne den Marquardt-Levenberg Algorithmus zügig ausführt. Eine genaue Betrachtung auf verschiedenen Prozessorsystemen, die eine genaue Prognose der Mindestanforderung an die Rechnerkomponenten stellt, wurde nicht durchgeführt. Vermutlich würde bereits ein moderner Mehrkernprozessor mit mind. 2 GHz ausreichen um eine deutliche Verbesserung gegenüber dem Testsystem (Intel Core 2 Duo 1,3GHz) zu erhalten.

#### **7.5. Erweiterungsmöglichkeiten**

Durch die Art des Aufbaus des Algorithmus inklusive seiner Datenstruktur sind Erweiterungen problemlos möglich und können modulweise vorgenommen werden. Neben den im vorangegangenen Abschnitt aufgezeigten Optimierungspotentialen gibt es eine Reihe von Erweiterungen, die sinnvoll scheinen. Dazu gehört eine Erweiterung der Dateiformate zum Einlesen von Messdaten. Diese könnte zum Einen eine automatisierte Erkennung des Datenformats enthalten, zum Anderen auch eine Auswahl von Spalten in tabellarischen Datentypen, wie zum Beispiel csv-Dateien und auch die Möglichkeit statt Kanalwerten Energiewerte einzulesen. Zusätzlich könnte noch die Möglichkeit ein Vielkanalspektrum einzulesen, welches ohne die Probe aufgenommen wurde und vom Spektrum mit der Probe abgezogen würde.

Zur Optimierung der Ausgleichsroutine selbst könnte entweder ein Simplexalgorithmus implementiert werden, der geeignete Startwerte sucht, oder dem Benutzer die Möglichkeit gegeben werden Startwerte selbst anzugeben. Letzteres kann jedoch nur sinnvoll sein, wenn der Benutzer das Ergebnis seiner Eingabe direkt angezeigt bekommt. Dies sollte jedoch nicht mehr nötig sein, wenn das Geschwindigkeitsproblem des Marquardt-Levenberg Algorithmus gelöst ist und so der Algorithmus bis zum Erreichen von Konvergenz oder einer Mindestgenauigkeit laufen kann. Der Marquardt-Levenberg Algorithmus ist robust genug, um auch mit durchschnittlichen Startwerten zu konvergieren.

Zur automatisierten Auswertung könnte ein noch zu entwickelndes Modul Linien in einem Spektrum selbstständig erkennen und die dazugehörigen statistischen Berechnungen durchführen und in einer

---

Zusammenfassung exportieren. Dessen Erweiterung ermöglichte dann die Auswertung ganzer Meßreihen nacheinander.

---

## 8. Zusammenfassung

---

Ziel dieser Arbeit war es einen Algorithmus zu entwickeln, mit dem Vielkanalspektren mit geringer Statistik entfaltet und einer Signifikanzprüfung unterzogen werden. Aufgrund des Umfangs der Thematik wurde der Fokus auf  $\gamma$ -Spektren gelegt. Die zentrale Anforderungen waren, dass die Benutzung anwenderfreundlich über eine visuelle Oberfläche geschehen sollte, und Erweiterungen, im Besonderen die Entfaltung von  $(e, e')$ -Spektren, jederzeit problemlos möglich.

Deshalb wurde mit der Programmiersprache Microsoft C# eine Softwarearchitektur aufgebaut, die diese Aspekte mit einer Trennung des Algorithmus in Frontend, Backend und mathematische Methoden beachtet. Dieses ermöglicht es zukünftige Anpassung, Ergänzung und Erweiterungen problemlos einzuarbeiten, besonders da durch das Verwenden von Schnittstellen bereits existierende Methoden konsequent genutzt werden können.

Zur Entfaltung der  $\gamma$ -Vielkanalspektren wurde der Marquardt-Levenberg Algorithmus zur Lösung der Minimierungsaufgabe nach der Methode der kleinsten Quadratsumme zum Ausgleichen entsprechender Modellfunktionen implementiert und zur Signifikanzprüfung eine Berechnung von charakteristischen Grenzen.

Damit steht nach Abschluss dieser Arbeit ein Algorithmus bereit, mit dem prinzipiell bereits Linien ausgewertet werden können und Nachweiss- und Erkennungsgrenzen angegeben werden. Die Verlässlichkeit dieser ist lediglich abhängig von der Güte der Ausgleichsrechnung und der Modellfunktion. Diese bietet zugleich neben der Ausführungsgeschwindigkeit Raum für Verbesserung, vor allem in realen Spektren mit dicht beieinander liegenden Linien.

---

## 9. Ausblick

---

Das moderne Konzept der charakteristischen Grenzen bietet eine gute Möglichkeit, Messergebnisse auf Signifikanz zu testen. Um diese voll auszuschöpfen sollten auch die Vorgehen bei der Entfaltung selbst überprüft werden, denn die Grundannahmen, dass zu untersuchende Linien ausreichend isoliert zur Schätzung des Untergrundes auftreten, lassen sich an realen Spektren nur schwer nachvollziehen.

Beim Ausgleichen der Modellfunktionen der Linien ist der Algorithmus auf eine gute Aufbereitung der Messdaten im untersuchten Bereich angewiesen, d.h. um eine Linie möglichst genau bewerten zu können, ist es erforderlich, den Untergrund genau zu bestimmen oder zu schätzen. Obwohl hierzu die DIN 25482 bzw. ISO 11929 Handlungsanweisungen geben, sind diese Verfahren nicht als alleinige Lösung zu sehen. Das übliche Vorgehen, Startparameter und Untergrund aus der Linie und deren Umgebung in einem festgelegten Bereich zu schätzen, kann nicht abschließend sein. Generell sollten Startparameter möglichst unabhängig vom gewählten Datenbereich sein und sich lediglich an den Modelldaten orientieren, damit also nicht an den Grenzen des untersuchten Bereichs. Ein alternatives Vorgehen wurde mit der Überlegung zur Dynamisierung von Grenzen des Untersuchungsbereiches angedacht.

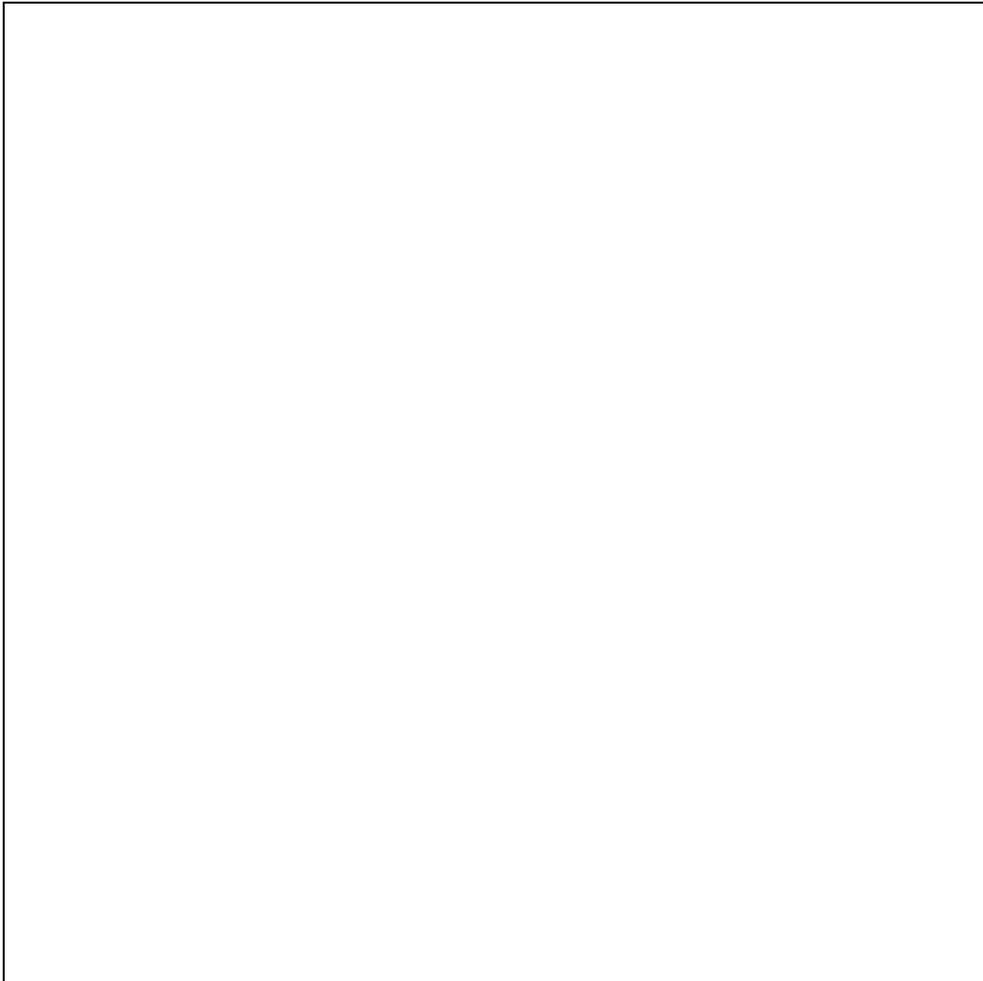
---

## 10. Anhang

---

### 10.1. Digitaler Anhang

Auf der DVD befinden sich drei Komponenten, die Microsoft Visual Studio 2010 Projektmappe mit allen zugehörigen Dateien, eine kompilierte Version des Algorithmus zum direkten Ausführen und eine Installationsroutine, die zusätzlich zum Algorithmus das benötigte Microsoft .Net 4 Framework inkl. deutschem Sprachpaket enthält.



### 10.2. Datenstruktur im Backend

Nach dem Initialisieren des Backend stehen folgende `DataTable` im `DataSet` bereit, um die während der Laufzeit benötigten Daten vorzuhalten.

Tabelle 7 zeigt die in der `DataTable` Daten definierten Spalten mit den Datentypen.

DataTable „Daten“		
DataColumn „Kanal“	Int32	Nummer des Kanals
DataColumn „Counts“	Int32	Anzahl der Impulse im Kanal
DataColumn „Energie“	Double	Energie des Kanals

Tabelle 8 zeigt die in der `DataTable` Regression enthaltenen Spalten mit den Datentypen

<code>DataTable</code> „Regression“		
DataColumn „ID“	<code>Int</code>	Index der Datenzeile
DataColumn „Name“	<code>String</code>	Name der Funktion
DataColumn „Funktion“	<code>Funktion</code>	Gefittete Modellfunktion
DataColumn „RegressionParameter“	<code>Parameter[]</code>	Array der Fitparameter der Modellfunktion
DataColumn „ObservationParameter“	<code>Parameter[]</code>	Array der Variablenparameter der Modellfunktion
DataColumn „UFunktion“	<code>Funktion</code>	Modellfunktion des Untergrundes
DataColumn „URegressionParameter“	<code>Parameter[]</code>	Array der Fitparameter der Modellfunktion des Untergrundes
DataColumn „UObservationParameter“	<code>Parameter[]</code>	Array der Variablenparameter der Modellfunktion des Untergrundes
DataColumn „R2“	<code>Double</code>	Bestimmtheitsmaß des Fits
DataColumn „Chi2“	<code>Double</code>	Residuenquadrat des Fits
DataColumn „Marker“	<code>SortedDictionary&lt;Int,Int&gt;</code>	Markierungen des Datenbereichs
DataColumn „Regression“	<code>Dictionary&lt;Int,Double&gt;</code>	Funktionswerte der ausgeglichenen Funktion

Tabelle 9 zeigt die in der `DataTable` ShowData definierten Spalten mit den zugehörigen Datentypen.

<code>DataTable</code> „ShowData“		
DataColumn „ID“	<code>Int</code>	Indes der Datenzeile
DataColumn „Peakmitte“	<code>Double</code>	~ der Modellfunktion
DataColumn „Unsicherheit Peakmitte“	<code>Double</code>	
DataColumn „FWHM“	<code>Double</code>	
DataColumn „Unsicherheit FWHM“	<code>Double</code>	
DataColumn „Bruttofläche“	<code>Double</code>	
DataColumn „Untergrundfläche“	<code>Double</code>	
DataColumn „Nettofläche“	<code>Double</code>	
DataColumn „Unsicherheit Nettofläche“	<code>Double</code>	
DataColumn „EKG“	<code>Double</code>	Erkennungsgrenze
DataColumn „NWG“	<code>Double</code>	Nachweisgrenze
DataColumn „R2“	<code>Double</code>	Bestimmtheitsmaß
DataColumn „Chi2“	<code>Double</code>	Residuenquadrat

---

### 10.3. Microsoft Public License (Ms-PL)

Der Quelltext der Klasse NumericalMethods in BlogCode unterliegt der Microsoft Public License (Ms-PL).

*This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.*

#### 1. Definitions

*The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law.*

*A "contribution" is the original software, or any additions or changes to the software.*

*A "contributor" is any person that distributes its contribution under this license.*

*"Licensed patents" are a contributor's patent claims that read directly on its contribution.*

#### 2. Grant of Rights

*(A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.*

*(B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.*

#### 3. Conditions and Limitations

*(A) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.*

*(B) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.*

*(C) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.*

*(D) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.*

*(E) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.*

---

## 11. Abbildungsverzeichnis

---

Abbildung 1: Schematische Darstellung zur Festlegung der Erkennungsgrenze $X^*$ und Nachweisgrenze $X^\#$ mit den Irrtumswahrscheinlichkeiten $\alpha$ und $\beta$ . Die rosagefärbte Kurve zeigt die Wahrscheinlichkeitsverteilung der Nullhypothese, die grüingefärbte die der Alternativhypothese. .3	3
Abbildung 2: Im Diagramm sind Stufenfunktionen für verschiedene Parameter $\sigma$ dargestellt. Die übrigen Parameter sind fest mit $\mu_{\text{Untergrund}} = 0, a_0 = 0, a_{\text{diff}} = 1$ .....5	5
Abbildung 3: Die Abbildung zeigt beispielhaft den Einfluss des Parameters $\sigma$ der Modellfunktion des Untergrundes auf die Linienform. Für die Normalverteilung wurde $YX = 51\sigma 2\pi e - 12X 22$ verwendet. ....6	6
Abbildung 4: Im Blockdiagramm wird die Softwarearchitektur des Algorithmus dargestellt. Unter der Aufteilung in Frontend, Backend und BlogCode sind die jeweils implementierten Funktionen zusammengefasst. ....9	9
Abbildung 5: Die Abbildung zeigt die Programmoberfläche mit farbigen Markierungen zur Visualisierung der zwei Anzeigebereiche für Spektrum und Fitparameter, sowie dem Kontrollbereich.....12	12
Abbildung 6: Das für die Untersuchung erzeugte Spektrum enthält definierte Linien, an denen nach Entfaltung das Verhalten des Algorithmus bewertet werden kann. ....14	14
Abbildung 7: Schematische Darstellung des Peak zu Untergrundverhältnis (PzU) [2] .....15	15
Abbildung 8 Auftrag des Bestimmtheitsmaßes über dem Quotienten Amplitude $3\sigma$ Untergrund mit den Daten aus Tabelle 4.....16	16
Abbildung 9: Reales Spektrum, bei dem mit Germanium-Detektoren $\gamma$ -Zerfälle aufgenommen wurden. ....17	17

---

## 12. Literaturverzeichnis

---

1. Michel, R.; Kirchoff, K. *Nachweis-, Erkennungs- und Vertrauensgrenzen bei Kernstrahlungsmessungen*; TÜV-Verlag GmbH: Köln, 1999.
2. Gilmore, G.; Hemingway, J. D. *Practical Gamma-Ray Spectrometry*, 1st ed.; John Wiley & Sons: Chichester, 1995.
3. J.Wolberg. *Data Analysis Using the Method of Least Squares - Extracting the Most Information from Experiments*; Springer-Verlag: Heidelberg, 2006.
4. Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes - The Art of Scientific Computing*, 3rd ed.; Cambridge University Press: Cambridge, 2007.
5. Weise, K.; Wöger, W. *Meßunsicherheit und Meßdatenauswertung*; Wiley-VCH Verlag GmbH: Weinheim, 1999.
6. Weise, K.; Hübel, K.; Michel, R.; Rose, E.; Schläger, M.; Schrammel, D.; Täschner, M. *Nachweisgrenze und Erkennungsgrenze bei Kernstrahlungsmessungen: Spezielle Anwendungen* -

---

Vorschlag für eine Norm; TÜV-Verlag Köln GmbH: Köln, 2004.

7. Guidry, T. F. Trent Guidry's blog about Numerical Methods, Mathematical Modeling, Numerical Simulations, C#, Silverlight, and WPF. <http://www.trentguidry.net/post/2010/01/17/Latest-version.aspx> (accessed Juli 28., 2010).
8. IAEA - International Atomic Energy Agency. *The IAEA-2008-03 proficiency test on the determination of natural radionuclides in phosphogypsum and spiked water*; Wien, 2008.
9. Birkhan, J. H. *Gammaskpektrometrie an Bodenproben von als Grünland genutzten Auenflächen der Zwickauer und Vereinigten Mulde*; Zentrum für Strahlenschutz und Radioökologie der Leibniz Universität Hannover, 2009; Diplomarbeit, nicht veröffentlicht.
10. Free Software Foundation, Inc. GSL - GNU Scientific Library. <http://www.gnu.org/software/gsl/> (accessed Juli 28, 2010).
11. Birkhan, J. H. *Persönliche Kommunikation*; 2010.