

Bachelorarbeit

***„Evaluation von Analysealgorithmen zur
Untergrundbestimmung bei Gammaspectren“***

Maximilian Gaag

Inhaltsverzeichnis

1 Einleitung.....	3
1.1 Gegenstand und Bedeutung.....	3
1.2 Problematik.....	4
1.3 Motivation.....	5
1.4 Ziele.....	5
1.5 Literatur.....	6
2 Grundlagen.....	7
2.1 Begriffserklärung und Abkürzungsverzeichnis.....	7
2.2 Theoretische Beschreibung der Algorithmen.....	7
2.3 Umsetzung der Algorithmen in MATLAB.....	9
3 Vorgehen inkl. grobe Beschreibung der Parameter.....	12
3.1 Vorgehen bei Snip.....	14
3.2 Vorgehen bei ParabEnv.....	15
4 Ergebnisse.....	17
4.1 Auswirkungen der Parameter.....	17
4.1.1 Snip.....	17
4.1.2 ParabEnv.....	21
4.2 Festlegung des Rahmens.....	30
4.2.1 Snip.....	30
4.2.2 ParabEnv.....	31
4.3 Stärken und Schwächen.....	33
4.3.1 Snip.....	33
4.3.2 ParabEnv.....	33
5 Diskussion der Ergebnisse.....	34
6 Zusammenfassung und Schlussfolgerung.....	36
7 Literaturverzeichnis.....	37
8 Anhang.....	38
8.1 Abbildungen.....	38
8.2 Code.....	45
9 Eidesstattliche Erklärung.....	49

1 Einleitung

1.1 Gegenstand und Bedeutung

Gammastrahlung ist hochenergetische elektromagnetische Strahlung, die zum Beispiel beim Zerfall radioaktiver Stoffe auftritt und besteht aus ungeladenen Photonen. Jeder dieser Gammastrahler strahlt Photonen mit einer für ihn spezifischen Energie aus, woran man ihn auch erkennen kann. So hat jeder Gammastrahler sozusagen seinen eigenen Fingerabdruck.

Wenn man wissen will, welche Gammastrahler in einer Probe vorhanden sind, kann auf die Gammaskopie zurückgegriffen werden. Diese misst, wie der Name schon sagt, das Spektrum, also die Energie und die Intensität, der Gammastrahlen. Aus dem daraus entstehenden Graphen kann dann anhand der Ausschläge abgelesen werden, welche Gammastrahler in der Probe enthalten sind, da sich diese über ihr Energiemaß bestimmen lassen.

Die beiden grundsätzlichen Bestandteile eines solchen Gammastrahlenspektrometers sind ein Detektor und eine Strahlungsquelle, meist eine radioaktive Probe. Die Probe wird so angebracht, dass sie auf den Detektor strahlt. Dieser registriert dann die Impulse, die durch Wechselwirkung zwischen den Gammastrahlen und den Elektronen des Detektors entstehen.

Es gibt zwei hauptsächlich auftretende Effekte. Der Wichtigste ist wohl der Photoeffekt. Hierbei trifft ein Photon auf ein Elektron und das Photon wird vollständig vom Elektron absorbiert, wodurch das Elektron aus seiner aktuellen Bindung, innerhalb des Atoms, gelöst wird. Dies ist der wichtigste Effekt, weil er genau die Information liefert, die gesucht wird. Das ist die exakte Energie, mit der ein radioaktives Nuklid abstrahlt. Den durch gehäuftes Vorkommen dieses Effektes auftretenden Ausschlag nennt man Photopeak. Er stellt die spezifische Energie von einem der, in der Probe vorhandenen, Gammastrahlern dar. Anhand dieses Peaks ist es letztendlich möglich die enthaltenen Gammastrahler zu identifizieren.

Der zweite Effekt nennt sich Compton-Effekt. Hierbei trifft ebenfalls ein Photon auf ein Elektron. Dieses Mal wird das Photon aber nicht absorbiert, sondern gibt lediglich einen Teil seiner Energie an das Elektron ab und wird dann in einem Winkel von 0° bis 180° gestreut. Es setzt seinen Weg also mit geringerer Energie und damit größerer Wellenlänge fort.

Am Ende sollte dann ein Spektrum herauskommen, das ungefähr so aussieht:

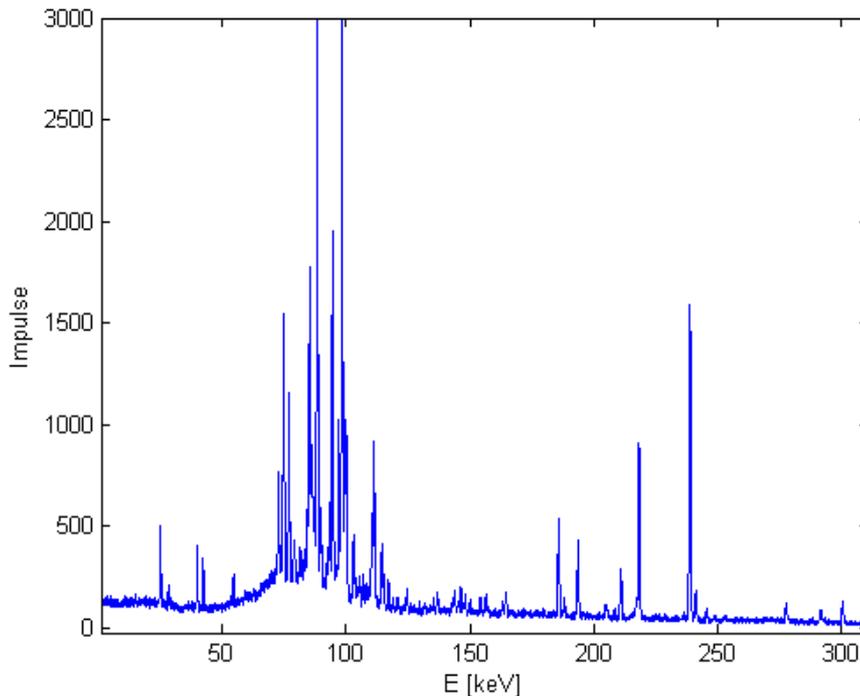


Abbildung 1: Beispielspektrum

1.2 Problematik

Leider ist das obige Spektrum im jetzigen Zustand noch lange nicht perfekt. Noch sind zu viele Störungen und Fehlinformationen enthalten. Zwar könnte damit gearbeitet werden, jedoch mit dem Risiko, dass eventuell Informationen verfälscht, oder sogar gar nicht ausgelesen werden. Perfekt wäre es, wenn das Spektrum komplett der x-Achse anliegen würde, vollkommen glatt, ohne Rauschen und nur die Photopeaks zu sehen wären. Ein kleiner Teil der Störungen, die das Signal beeinflussen, kann schon im obigen Teil erkannt werden. Es spielen nämlich viel mehr Effekte eine Rolle, als es im besten Fall sein sollten.

Einer der Störeffekte zum Beispiel, ist der bereits oben erwähnte Compton-Effekt. Dieser trägt nichts zum Photopeak bei, sondern liefert einen Impuls, dessen Energie bis zu einem gewissen Punkt, dieser „Punkt“ wird auch Compton-Kante genannt, unterhalb der spezifischen Energie des Photons liegt. Wenn die spezifische Energie eines Photons zum Beispiel 600keV betragen würde, dann könnte durch den Compton-Effekt eine Energie von 400keV geliefert werden. Es ist also möglich, dass angenommen wird es seien zwei unterschiedliche Gammastrahler in der Probe vorhanden. Glücklicherweise sind die Impulse, die durch den Compton-Effekt entstehen relativ gleich verteilt und erreichen kaum die Intensität eines Photopeaks. Dennoch zieht jeder Photopeak einen solchen Rattenschwanz aus schwächeren Impulsen, welche von 0keV bis zur Energie der Compton-Kante reichen, hinter sich her. Dies wird auch Compton-Kontinuum genannt. Da im niedrigeren Bereich des Energiespektrums immer mehr Compton-Kontinuen aufeinandertreffen, je nachdem wie viele Photopeaks, also unterschiedliche Gammastrahler, in der Probe vorhanden sind, bildet sich dort oft ein sichtbarer Hügel, oder eine Steigung. Umgekehrt flacht das Spektrum im höheren Bereich immer weiter ab, da

weniger Compton-Kontinuen vorhanden sind.

Das Compton-Kontinuum trägt einen Großteil zum Untergrund bei. Der Untergrund füllt fast den kompletten Bereich zwischen Gammaskpektrum und x-Achse aus. Ebenso spielt hier auch die Umgebungsstrahlung eine Rolle, welche eigentlich immer vorhanden ist und die man nur schwer ausschließen kann.

Ein weiterer Störfaktor ist das in der Elektrotechnik bekannte Rauschen, welches auch im obigen Spektrum sehr gut erkannt werden kann.

Dies sind nur einige Störungen, die eine Rolle spielen.

1.3 Motivation

Es werden somit sehr viele Störgrößen ungewollt mit betrachtet und diese Störgrößen müssen durch Werkzeuge, wie zum Beispiel Algorithmen, entfernt werden. Da ich mich in dieser Bachelorarbeit auf die Untersuchung von Algorithmen zur Bestimmung des Untergrundes beziehe, werde ich mich im Weiteren auf diese beschränken und die anderen Probleme größtenteils außer Acht lassen.

Nun gibt es natürlich bereits Algorithmen, mit denen sich der Untergrund von Gamma-spektren bestimmen lässt. Das Problem hierbei ist, dass diese meist nur bei bekannten Spektren funktionieren, auf die sie, mehr oder weniger, speziell zugeschnitten sind. Wenn diese Algorithmen eins zu eins auf ein unbekanntes Spektrum angewandt würden, kann es schnell vorkommen, dass die Untergrundbestimmung misslingt und weit vom angestrebten Ergebnis entfernt ist.

Einen einzelnen Algorithmus zu entwickeln, der für alle Spektren einsetzbar wäre, ist aber auch keine Möglichkeit. Der Aufwand und die Kosten wären noch viel zu hoch. Außerdem würde sich keine Firma finden, welche sich an solch ein Projekt wagen würde, da auch viel zu wenige Abnehmer dafür vorhanden sind.

Darum werde ich in dieser Arbeit kleinere, einfachere Algorithmen genauer untersuchen, um zu verstehen, wie sich diese auf unterschiedliche Spektren anwenden lassen. Das damit entstehende Ergebnis ist wahrscheinlich nicht perfekt, aber gut genug, um damit arbeiten zu können.

1.4 Ziele

Ziel dieser Arbeit ist es, zwei Algorithmen zur Untergrundbestimmung zu evaluieren und einen Überblick über folgende Themen zu geben:

- Einen Ausblick über die Auswirkungen der einzelnen Parameter der Algorithmen zu geben, um zum Beispiel schnell abschätzen zu können, welcher Parameter in einem konkreten Fall verändert werden muss.
- Einen groben Rahmen festzulegen, in dem die einzelnen Parameter gewählt werden können, sodass mit hoher Wahrscheinlichkeit ein gutes Ergebnis erreicht wird.
- Stärken und Schwächen der Algorithmen herausarbeiten, um zum Beispiel die Verwendung eines Algorithmus auszuschließen, der sich für den beabsichtigten Zweck nur schwer oder gar nicht einsetzen ließe.

- Einen schnellen Überblick über die Möglichkeiten geben, um mit den Algorithmen ohne großes Vorwissen zu arbeiten.

1.5 Literatur

Diese Bachelorarbeit basiert hauptsächlich auf den Papers von J. Kajfosz [1] und C.G. Ryan [2], welche in zwei Algorithmen umgesetzt wurden, die im restlichen Teil der Bachelorarbeit benutzt werden. Die Algorithmen werden außerdem im Weiteren mit Snip (Ryan) und ParabEnv (Kajfosz) bezeichnet.

2 Grundlagen

2.1 Begriffserklärung und Abkürzungsverzeichnis

Stützstelle: „Stützstelle ist ein Begriff aus der numerischen Mathematik. Damit werden die Argumente von Funktionen bezeichnet, die für die weitere Rechnung benutzt werden [...]“ [3], in diesem Fall der Berechnung der Untergrundkurve.“

ParabEnv Parabolic Envelope (Kajfosz [1])

SNIP Statistics-sensitive Non-linear Iterative Peak-Clipping (Ryan [2])

FWHM Full Width at Half Maximum

2.2 Theoretische Beschreibung der Algorithmen

Die wesentlichen Ideen in diesem Teil stammen aus den Arbeiten von Ryan und Kajfosz. Sie wurden nur übersetzt und dann hier nochmals aufgeführt.

Snip wurde von Ryan in drei Phasen unterteilt. Die erste Phase besteht aus der Anwendung eines Low Statistics Digital Filters und der Reduzierung der Anzahl an Impulsen, pro Kanal des Signals, mittels einer doppelten Anwendung des Logarithmus. Dadurch wird der Bereich, in dem sich die Anzahl der Impulse bewegen, von ca. 10^7 auf das Intervall 0-2,8 beschränkt, was die Dauer der Anwendung des Filters, sowie der Peak Clipping Schleife in Phase zwei sehr reduziert.

Der Low Statistics Digital Filter bietet eine Glättung, um Rauschen aus Bereichen niedriger Statistik (Anzahl der Impulse gering) zu entfernen, während eine Verbreiterung der Basis von Peaks und das Auffüllen von Minima zwischen Peaks vermieden wird [2,4]. Hierbei wird jeder Kanal i durchlaufen und für jedes i zwei Fenster, eines auf jeder Seite betrachtet. Die Fenster haben eine Breite von $f \times FWHM(E_i)$. In den Fenstern werden die y-Werte aufaddiert, was zu einer linken Summe L und einer rechten Summe R führt. Die Fensterbreite wird dann immer weiter reduziert, bis entweder die Summe

$S = L + y_i + R$ unter ein konstantes Minimum M fällt, oder zwei Bedingungen erfüllt sind:

1. S ist kleiner als $N = A\sqrt{y_i}$, A ist eine Konstante. Dies stellt sicher, dass die Glättung auf die Bereiche niedriger Statistik innerhalb des Spektrums beschränkt wird.
2. Die Neigung $(R+1)/(L+1)$ liegt zwischen $1/r$ und r , r ist eine Konstante. Dies verhindert, dass die Kanten von Peaks in die Bildung des Mittelwertes mit einbezogen werden.

Falls diese Bedingungen erfüllt werden, wird das Mittel $S/(2f \times FWHM + 1)$ als neuer Wert gesetzt und der nächste Kanal wird betrachtet. Gute Parameterbelegungen für f , A , M und r können zum Beispiel den Referenzen entnommen werden.

In Phase zwei läuft dann eine Peak Clipping Schleife durch. Hierbei wird jeder Kanalwert $z(x)$ ersetzt durch den niedrigeren Wert zwischen $z(x)$ und

$$\bar{z}(x, w) = (z(x+w) + z(x-w))/2$$

während die Filterbreite $w(x)$ auf das doppelte der FWHM des Kanals x festgelegt ist. In den letzten paar Schleifendurchläufen (meistens acht) wird die Filterbreite stufenweise nach jedem Durchlauf um den Faktor $\sqrt{2}$ reduziert. Da somit das Auftreten von Schwankungen im Untergrund verringert wird und der Untergrund nicht zusätzlich im Nachhinein noch einmal geglättet werden muss. Phase drei besteht dann nur noch aus der Rücktransformierung der Ergebnisse, also der Aufhebung der Logarithmierung aus Phase eins. Am Schluss steht dann eine glatte Untergrundkurve zur Verfügung.

Kajfosz geht in seiner Arbeit von zwei Annahmen aus:

- Da der Untergrund eine langsam veränderliche Struktur ist, kann angenommen werden, dass er sich aus Objekten zusammensetzt, die um ein vielfaches größer sind als die Peaks und
- die Amplituden der Peaks sind immer positiv. Somit können die Untergrundwerte nirgendwo größer sein, als die Werte des Spektrums.

Der Algorithmus generiert in jedem Kanal i an das Signal eine nach unten offene Parabel

$$y(j) = H(i) \left[1 - \frac{(i-j)^2}{x^2(u+vi)} \right],$$

wobei $H(i)$ die Amplitude definiert und die Parameter x , u und v eine Anpassung der Breite der Parabeln ermöglichen.

Da die Amplitude laut Annahme zwei nie den Wert des Spektrums überschreiten kann, ergibt sich dafür, nach der Korrektur des Spektrumwertes $S(j)$ im Bezug auf Schwankungen, folgende Formel:

$$H(i) = \min_j \frac{S(j)}{1 - \frac{(i-j)^2}{x^2(u+vi)}}.$$

Eine erste Annäherung des Untergrundes $B(i)$ erhält man über:

$$B(i) = \max_j \left[\left[H(j) \left[1 - \frac{(i-j)^2}{x^2(u+vi)} \right] \right] \right].$$

Man hat aber festgestellt, dass diese erste Annäherung unter Peaks zu hoch ist, weshalb es nötig war die Methode weiter zu verbessern. Dadurch, dass nicht nur nach unten offene Parabeln von unten, sondern auch nach oben offene Parabeln von oben, an das Spektrum angepasst werden, konnte die Form des Untergrundes stark verbessert werden. Die entstehende Linie $G(i)$ mit:

$$G(i) = \min_j \left[\left[S(j) \left[1 + 3 \frac{(i-j)^2}{y^2(u+vj)} \right] \right] \right],$$

y ist wieder ein Parameter zur Anpassung der Parabeln. Überall wo $G(i) < S(i)$ wird $S(i)$ dann durch $G(i)$ ersetzt:

$$S'(i) = \min[S(i), G(i)].$$

Das finale Ergebnis erhält man durch die erneute Berechnung von $H(i)$ und $B(i)$ anhand der obigen Gleichungen. Doch in diesem Durchlauf verwendet man bei der Berechnung von $H(i)$ nicht $S(i)$ sondern stattdessen $S'(i)$.

Auch hier arbeitet der komplette Algorithmus wieder in mehreren Schritten. Zuerst wird das Spektrum optional geglättet, dann wird die Anzahl der Kanäle reduziert, um ähnlich wie bei

Snip die benötigte Rechenzeit zu verringern. Im Folgenden werden $G(i)$ und $S'(i)$ anhand der Formeln berechnet, im Anschluss ebenso $H(i)$ und $B(i)$ unter Verwendung von $S'(i)$. Zum Schluss wird die Anzahl der Kanäle wieder auf den Ursprung zurück transformiert.

2.3 Umsetzung der Algorithmen in MATLAB

Beide Algorithmen wurden in MATLAB, von der Firma *MathWorks*, implementiert. Der genaue Code ist im Anhang zu finden.

Snip wurde in MATLAB mit einigen Veränderungen umgesetzt. In der ersten Phase wurde der Low Statistics Digital Filter durch einen Savitzky-Golay-Filter ersetzt, mit dem eine erste Glättung des Signals erreicht wird, bevor der Algorithmus den Untergrund berechnet. Zusätzlich wird die Anzahl der Impulse pro Kanal nicht durch einen doppelten Logarithmus verringert, sondern der Logarithmus wird nur einmal angewandt und zwar am Anfang des Savitzky-Golay-Filters. Um trotzdem eine passende Reduzierung der Kanalwerte zu erreichen, wird vor der Anwendung des Filters obendrein die Wurzel des Signals gezogen. Die Entlogarithmierung passiert, nachdem der Savitzky-Golay-Filter durchgelaufen ist. Phase zwei wurde annähernd der Theorie implementiert. Die Peak Clipping Schleife wurde anhand der Formel von Ryan so eingebaut. Die Filterbreite wird jedoch nicht durch das doppelte der FWHM des Kanals x festgelegt, sondern durch die eingegebene Breite m . Dass die Filterbreite während der letzten acht Durchläufe nach jedem Durchlauf um den Faktor $\sqrt{2}$ reduziert wird, wurde beibehalten.

In Phase drei findet keine Entlogarithmierung statt, da diese bereits am Ende des Savitzky-Golay-Filters durchgeführt wurde. Stattdessen wird das Ergebnis quadriert, um das Ziehen der Wurzel aufzuheben.

Zuletzt wird das Ergebnis über einen Plot ausgegeben und angezeigt.

Die benötigten Eingabeparameter sind:

- E : Die Energie (d.h. die Abszisse),
- $sp1$: Das Signal,
- m : Die Breite des gleitenden Mittelwertfilters für den Savitzky-Golay-Filter und für die Peak Clipping Schleife,
- $niter$: Die Anzahl der Iterationen,
- E_{min} : Die linke Intervallgrenze,
- E_{max} : Die rechte Intervallgrenze.

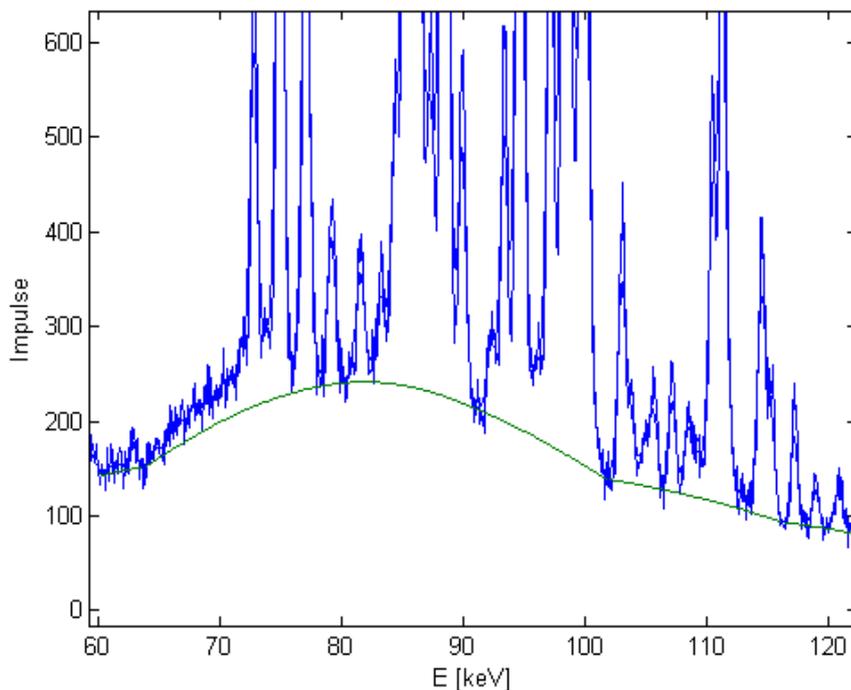


Abbildung 2: Untergrundbestimmung mittels Snip (Blau: Signal, Grün: Untergrund)

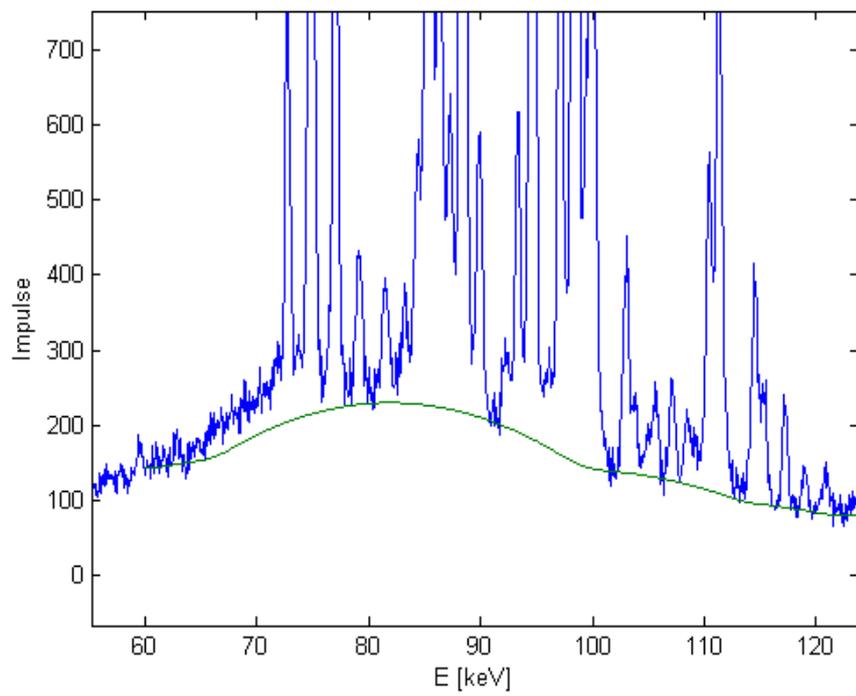
Im Gegensatz zu Snip wurde ParabEnv fast genauso umgesetzt, wie von Kajfosz beschrieben. Am Anfang wird wieder die Wurzel des Signals gezogen, um die Rechenzeit des Filters zu verringern. Der Filter ist derselbe Savitzky-Golay-Filter, der bei der Umsetzung von Snip verwendet wurde. Auch hier wird er zu Beginn eingesetzt, um das Signal zu glätten und das Rauschen zumindest ansatzweise zu entfernen. Im ersten Durchlauf wird die Amplitude $H(i)$ mithilfe von $S(i)$ berechnet. Es werden hier nur die nach unten offenen Parabeln betrachtet, die im ersten Schritt von unten an das Signal angepasst werden. Dies hat an sich keinen wirklichen Mehrwert im Algorithmus, da die Ergebnisse später nicht mehr verwendet werden.

Der zweite Durchlauf berechnet zuerst die Linie $G(i)$ und legt anschließend die Spektralpunkte $S'(i)$ fest. Mit den neuen Spektralpunkten $S'(i)$ wird die Amplitude $H(i)$ nochmals ermittelt. Im Anschluss muss noch die Annäherung des Untergrundes $B(i)$ errechnet werden.

Die Ergebnisse werden auch hier mittels Plotfunktion angezeigt.

Die benötigten Eingabeparameter sind:

- E : Die Energie (d.h. die Abszisse),
- f : sp1: Das Signal,
- E_{min} : Die linke Intervallgrenze,
- E_{max} : Die rechte Intervallgrenze,
- m : Die Breite des gleitenden Mittelwertfilters für den Savitzky-Golay-Filter,
- x, y, u, v : Parameter zur Kontrolle der Parabeln.



*Abbildung 3: Untergrundbestimmung mittels ParabEnv
(Blau: Signal, Grün: Untergrund)*

3 Vorgehen inkl. grobe Beschreibung der Parameter

Alle Tests wurden an mehreren zur Verfügung gestellten Spektren getestet. Zusätzlich zu den unterschiedlichen Spektren wurden die Methoden in drei verschiedenen, groben Bereichen, die innerhalb von Gammaspektren auftreten, angewandt. Die Bereiche, in die die Spektren aufgeteilt wurden, sind Bereiche mit hoher Aktivität, mittlerer Aktivität und niedriger Aktivität. Beispiele dafür sind in den Abbildungen 4 bis 6 zu sehen. Bei der Parameteranpassung wurde unterschieden zwischen einer Anpassung des ausgewählten Parameters nach oben und nach unten.

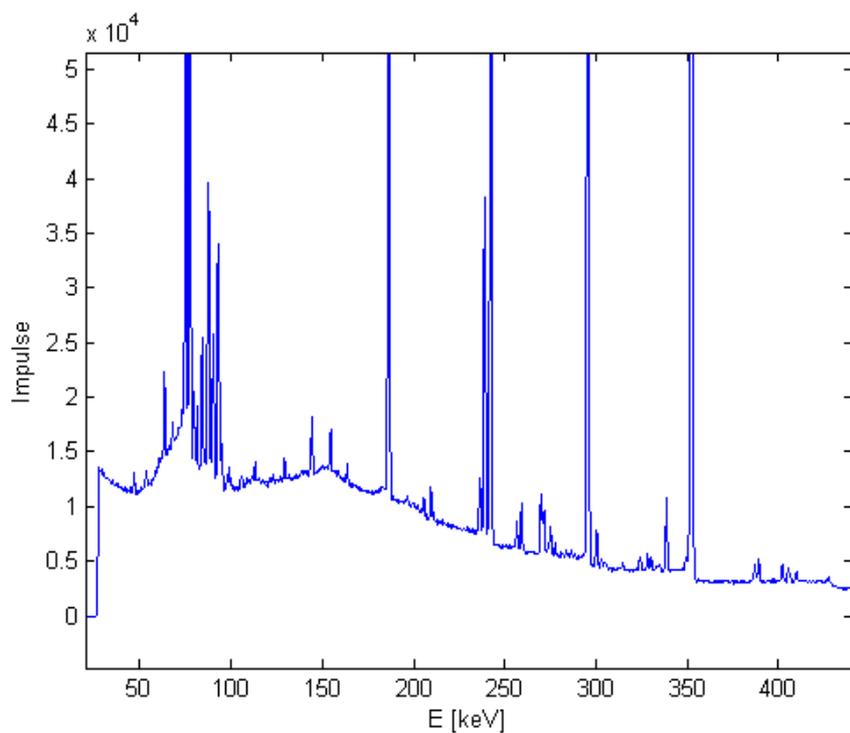


Abbildung 4: Bereich hoher Aktivität

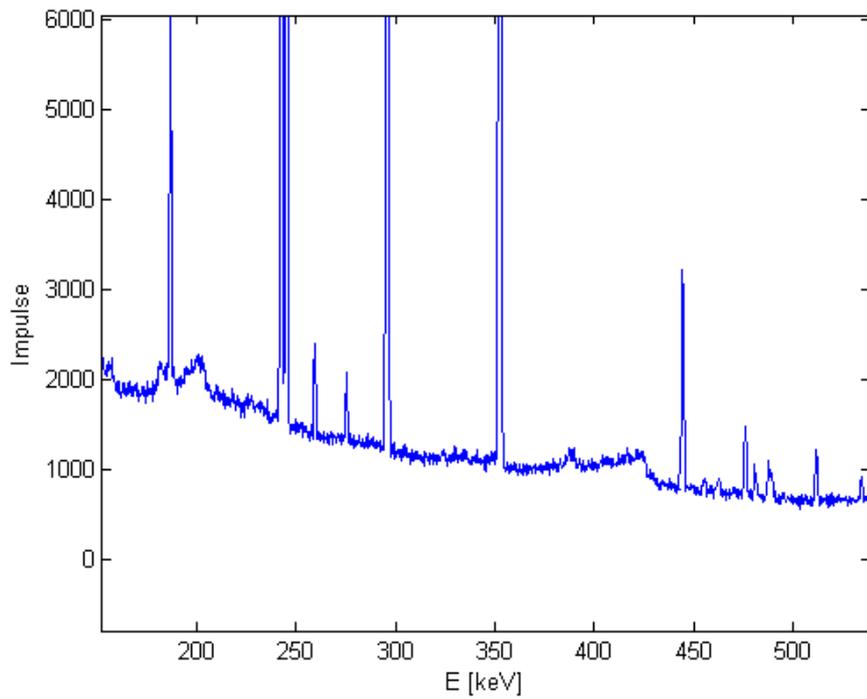


Abbildung 5: Bereich mittlerer Aktivität

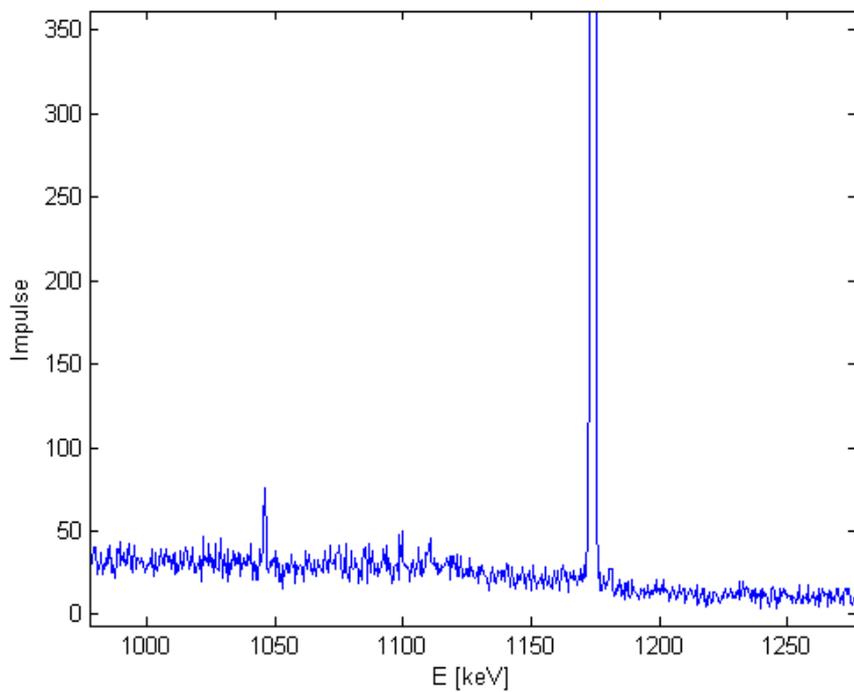


Abbildung 6: Bereich niedriger Aktivität

3.1 Vorgehen bei Snip

Um eine Arbeitsgrundlage zum Testen des Algorithmus, genauer gesagt den Auswirkungen der Eingabeparameter, zu schaffen, wurde zuerst mittels Trial and Error-Methode eine Parameterbelegung ermittelt, die den Untergrund gut bestimmt. Daraufhin wurden die Referenzkurve und eine zweite Kurve, bei der ein Parameter verändert wurde, zusammen geplottet, um die Auswirkungen der Parameterveränderung beobachten zu können.

Die relevanten Parameter, welche untersucht wurden, sind die Filterbreite m und die Anzahl der Iterationen $niter$. Bei größerer Filterbreite werden mehr Punkte gleichzeitig bei der Glättung und der Peak Clipping Schleife in Betracht gezogen. Es müsste bei einem größeren Wert der Untergrund stärker geglättet werden. Bei einem niedrigen Wert wird sich der Untergrund stärker dem Signal anpassen, da die einzelnen Spektralpunkte stärker ins Gewicht fallen. Unter Peaks wird der Untergrund dann nicht glatt durchlaufen, sondern stattdessen werden leichte Auswüchse in Richtung Peak zu vermuten sein. Eine Veränderung der Iterationen sollte ähnliche Auswirkungen haben, wie die entsprechende Anpassung der Filterbreite. Folglich müsste bei vielen Iterationen eine stärkere Glättung und ein flacherer Untergrund zu erkennen sein. Im Umkehrschluss sollte sich bei wenigen Iterationen eine stärkere Anpassung an das Signal abzeichnen.

Die Parameter E und $sp1$ sind durch das Spektrum gegeben und können nicht angepasst werden.

Die Intervallgrenzen sind frei wählbar, wurden aber auf jeweils einen der drei oben beschriebenen Bereiche festgelegt.

Da eine Vielzahl von Parametern verwendet werden kann, um den Untergrund zu bestimmen, wurden die zu betrachtenden Filterbreiten festgesetzt auf die Werte $m = 5, 10, 15$. Damit sollten genügend Breiten, von kleinen Filtern bis hin zu größeren Filtern, abgedeckt sein. Außerdem müssen die Parameter bei größer werdenden Filterbreiten immer weniger angepasst werden, um bei einem Selbstversuch mit breiterem Filter ein gutes Ergebnis zu erzielen.

Die Referenzkurven, die ermittelt wurden lauten (mit der MATLAB-Formel; siehe Abbildungen 24 bis 26 im Anhang):

- `snipbg(E,sp1,5,250,Emin,Emax);`
- `snipbg(E,sp1,10,80,Emin,Emax);`
- `snipbg(E,sp1,15,60,Emin,Emax);`

Der Rahmen, in dem die Parameter gewählt werden können, wurde so festgelegt, dass mit hoher Wahrscheinlichkeit eine gute Untergrundbestimmung gewährleistet werden kann. Dafür wurde die Anzahl der Iterationen in jedem Bereich für die drei Filterbreiten so lange verändert, bis das Ergebnis nicht mehr akzeptabel war. Somit konnten Ober- und Untergrenzen für die Iterationen bestimmt werden. Es sei hier angemerkt, dass die Grenzen absichtlich etwas großzügiger gewählt wurden. Da die Algorithmen damit zwar bei den zur Verfügung stehenden Spektren gute Ergebnisse lieferten, aber bei anderen Spektren vielleicht noch eine etwas größere Anpassung von Nöten sein könnte. Erkannte Stärken und Schwächen wurden nebenher notiert und gesammelt.

3.2 Vorgehen bei ParabEnv

Zu Beginn wurde auch hier eine Referenzkurve durch Trial and Error bestimmt, um die Auswirkung von Parameterveränderungen beobachten zu können.

Allerdings ist die Anzahl der relevanten Parameter bei diesem Algorithmus um einiges höher. Es müssen dieses Mal die Filterbreite m , sowie die vier Parameter x , y , u und v betrachtet werden.

Da die Filterbreite hier jedoch nur anfangs beim Savitzky-Golay-Filter Anwendung findet und nicht innerhalb des eigentlichen Algorithmus, hat eine Veränderung des Parameters nur leichte Auswirkungen auf den bestimmten Untergrund, siehe auch Abbildung 7.

Deswegen wurde bei diesem Algorithmus auf eine Unterscheidung der Filtergrößen verzichtet und hauptsächlich nur mit $m = 10$ gearbeitet.

x ist für die Breite der nach unten offenen Parabeln zuständig. Es ist also zu erwarten, dass dieser Parameter ähnliche Auswirkungen haben wird, wie eine Veränderung von m oder $niter$ bei Snip. Hohe Werte glätten den Untergrund von unten und niedrige Werte werden den Untergrund stärker an das Signal anpassen, weil eine feinere Differenzierung der Kurve möglich ist. Da y für die Breite der nach oben offenen Parabeln verantwortlich ist, lässt sich eine antagonistische Wirkung im Bezug auf x vermuten. Bei niedrigen Werten sollte der Untergrund also fülliger werden und sich stärker in Richtung Signalkurve ausbreiten. Wenn hohe Werte eingesetzt werden, müsste eine stärkere Glättung und allgemein ein flacherer Untergrund zu erkennen sein. Der Parameter u beeinflusst die Breite beider Parabeln, wenn auch nicht ganz so stark wie x und y selbst. Er dürfte sich demnach wieder ähnlich den beiden Parametern aus Snip verhalten. Hohe Werte führen zu einer Glättung, niedrige Werte zu einer Anpassung an das Signal. Am kompliziertesten ist der Parameter v . Er hat je nach Wert kaum bis sehr großen Einfluss auf den Untergrund, denn er bestimmt, mit welcher Wertung der eigentliche Kanalwert in die Berechnung der Parabeln mit einfließt. Deshalb wird er mit Werten <1 belegt. Er kontrolliert die „Steifheit“ der Parabeln. Damit kann vermutlich von einer sehr starken Glättung, bis zu keiner zusätzlichen Glättung alles herbeigeführt werden.

Die verwendete Referenzkurve für ParabEnv lautet (siehe Abbildung 27 im Anhang):

- `parabEnv(E,sp1,Emin, Emax,10,60,45,37,0.001);`

Eine Rahmenbestimmung ist nicht in derselben Art und Weise wie bei Snip möglich. Denn wenn alle Parameter bis auf einen fix wären und nur dieser Parameter verändert würde, wäre das Ergebnis schon nach kleinen Veränderungen nicht mehr zu gebrauchen. Dies hängt mit der starken Abhängigkeit der Parameter untereinander zusammen. Deswegen lässt sich idealerweise alles als ein Paket betrachten, welches gemeinsam angepasst wird. Darum wurde bei der Bestimmung des Rahmens auf einzelne Ober- und Untergrenzen der Parameter verzichtet und im Gegenzug Beispiele für niedrige und hohe Belegungen angeführt. Außerdem sind Relationen beschrieben, wie die Parameter im Verhältnis zueinander gewählt werden sollten, damit nachvollziehbar wird, wie zum Beispiel eine bestimmte Belegung von x durch geschickte Wahl der anderen Parameter aufgefangen werden kann.

Stärken und Schwächen wurden wieder nebenher notiert und gesammelt.

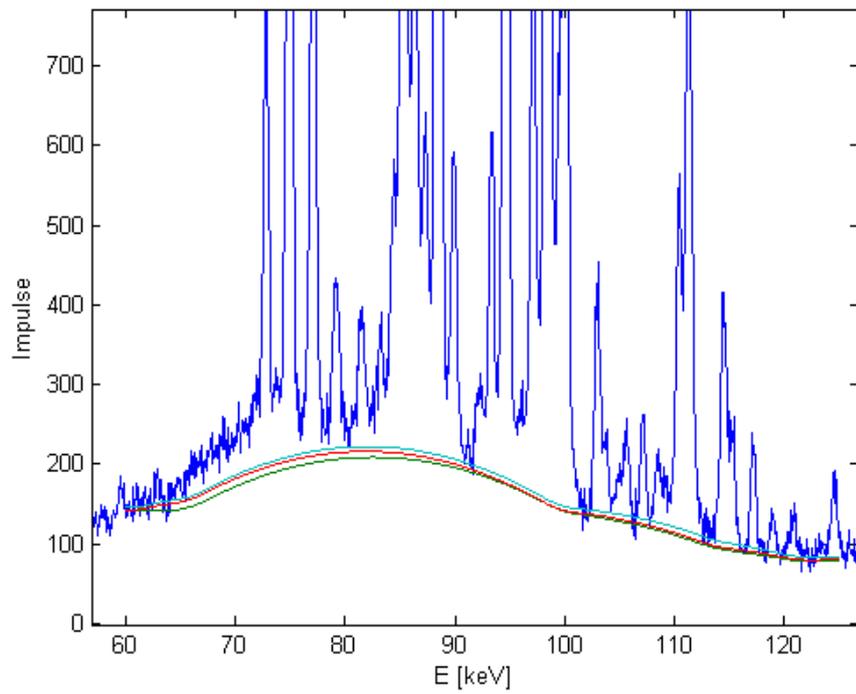


Abbildung 7: Filterbreiten $m = 5$ (grün), 10 (rot), 15 (blau) bei ParabEnv

4 Ergebnisse

4.1 Auswirkungen der Parameter

4.1.1 Snip

- *Breite des gleitenden Mittelwertfilters*

Beginnend mit einer Anpassung von m nach unten, also einer Verkleinerung der Filterbreite, sowohl für den Savitzky-Golay-Filter, als auch im Algorithmus selbst, lassen sich mehrere Auswirkungen erkennen, siehe auch Abbildung 8. Eine reine Verringerung der Filterbreite hat zur Folge, dass die Punkte im Bereich von Peaks stärker ins Gewicht fallen, weshalb eine höhere Anzahl an Iterationen vonnöten wäre, um dieselbe Glättung zu erreichen. Da aber die Anzahl der Iterationen gleich bleibt, hebt sich der Untergrund im Bereich von Peaks stärker ab, bzw. er passt sich generell stärker dem ursprünglichen Signal an. Die Erhebungen zwischen den Stützstellen neigen dabei nicht gegen eine der Flanken. Dies lässt sich besonders gut in den Bereichen zwischen den einzelnen Stützstellen erkennen. An den Stützstellen selbst ist eine deutliche Verschärfung der Winkel zu erkennen, was zu einem stark sprunghaften Aussehen der Kurve führt. An manchen Stellen kann es den Anschein haben, als würde die Kurve einigen zusätzlichen Stützstellen folgen, es kann aber auch lediglich ein Zufall sein, da diese Punkte zwischen zwei bereits vorhandenen Stützstellen liegen und nur durch das Aufblähen der Kurve zwischen diesen, die neuen Punkte durchläuft. Aber ab und zu können hierbei tatsächlich Stützstellen erkannt werden, die vorher nicht unbedingt offensichtlich waren. Erkennbar zum Beispiel zwischen 75 und 95keV in Abbildung 8.

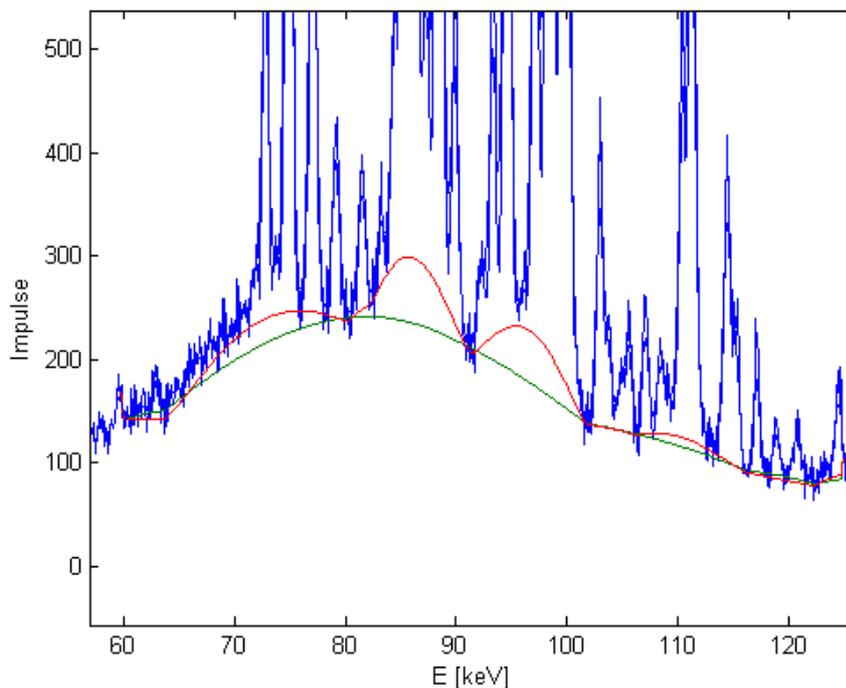


Abbildung 8: *Snip: Anpassung der Filterbreite nach unten (grün: Referenz, rot: Angepasste Kurve)*

Wenn die Filterbreite stattdessen vergrößert wird, also m nach oben angepasst wird, fallen die einzelnen Punkte im betrachteten Fenster nicht mehr so stark ins Gewicht. Dadurch werden Peaks und Wölbungen stärker geglättet und die Kurve erscheint flacher. Weil die Anzahl an Sprüngen, Beulen etc. in der Kurve verringert wird, wird eine höhere Feinheit erreicht. Allerdings passiert es auch schnell, dass der Untergrund zu stark geglättet wird. Hierdurch können nicht mehr alle Stützstellen erreicht werden und vor allem schnelle Änderungen im Signal, wie zum Beispiel im Bereich von 140 bis 160keV in Abbildung 9, werden nur noch schlecht, oder gar nicht mehr nachvollzogen. Dadurch, dass nicht mehr alle Stützstellen erreicht werden, sieht es so aus, als wären insgesamt weniger Stützstellen vorhanden, was aber nicht der Fall ist. Die Auswirkungen beschränken sich je nach Schwere der Anpassung nur auf Bereiche hoher Aktivität. In hochenergetischen Bereichen, die hauptsächlich aus Rauschen bestehen, oder Bereichen mittlerer Aktivität ohne grobe Veränderungen im Signal, ist nicht unbedingt ein Unterschied zu sehen. Denn dort ist der Untergrund eine sich so langsam verändernde Kurve, dass selbst bei hoher Filterbreite die Schwankungen im Signal noch ohne größere Probleme nachvollzogen werden können.

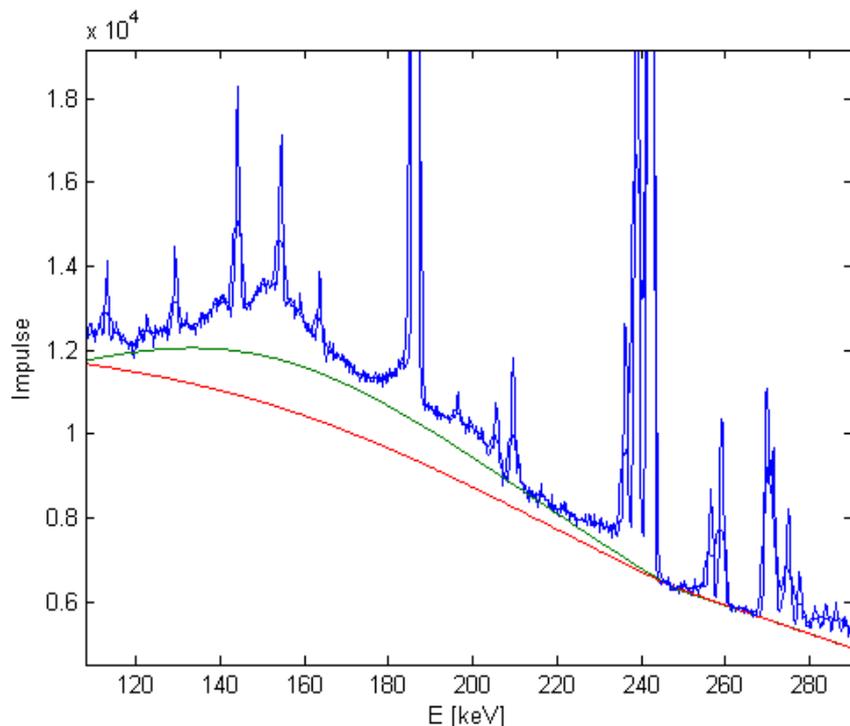


Abbildung 9: Snip: Anpassung der Filterbreite nach oben (grün: Referenz, rot: Angepasste Kurve)

- *Zahl der Iterationen*

Veränderungen an der Anzahl der Iterationen, die durchlaufen werden, haben ähnliche Auswirkungen wie die entsprechende Anpassung der Filterbreite. Eine Verringerung der Iterationen führt dazu, dass sowohl Filter, als auch Peak Clipping Schleife seltener abgearbeitet werden. Peaks werden somit nicht mehr so stark geglättet, weshalb sich auch hier der Untergrund zwischen den Stützstellen stärker heraushebt, bzw. sich dem Signal anpasst. An den Stützstellen treten ebenfalls durch den schnellen Richtungswechsel schärfere Winkel auf. Der Untergrund passt sich aber dadurch besser an Veränderungen im Signal an, was besonders in Bereichen hoher Aktivität wichtig sein kann, da sich das Signal hier oftmals schnell und stark verändert.

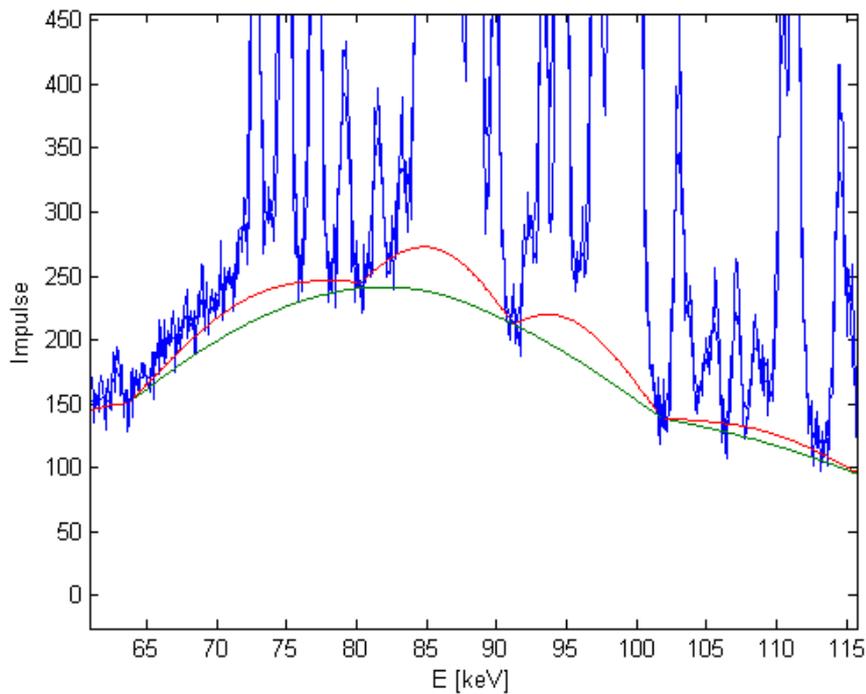
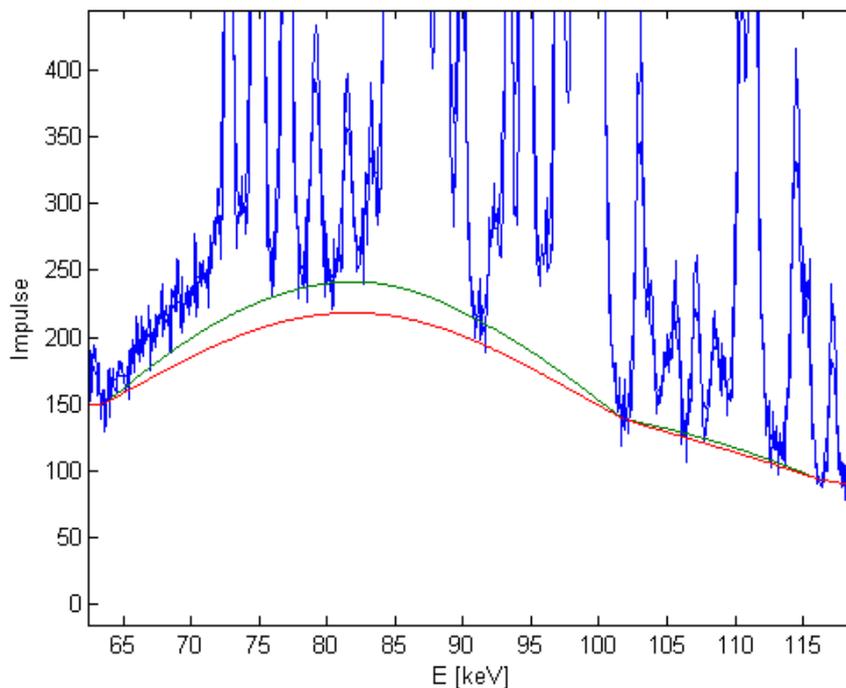


Abbildung 10: Snip: Anpassung der Iterationen nach unten (grün: Referenz, rot: Angepasste Kurve)

Eine hohe Anzahl an Iterationen hingegen führt zu einer stärkeren Glättung des Untergrundes und wiederum durch die geringere Anzahl an Sprüngen etc. zu einer feineren Kurve. Durch die stärkere Glättung sinkt die Kurve zwischen den Stützstellen ab und wird flacher. Obwohl dieselbe Anzahl an Stützstellen vorhanden ist, sieht es möglicherweise so aus, als würden manche nicht mehr durchlaufen, weil sie vom Untergrund nicht mehr erreicht werden können. Wenn der Wert zu hoch gewählt wird und zu stark geglättet wird, kann es dazu kommen, dass der komplette Untergrund glatt gebügelt wird. Oder falls es noch nicht ganz so weit ist, hat es eher den Anschein, als würden Geraden an das Signal angepasst werden, statt einer Kurve, weil nur knappe Verbindungen zwischen den einzelnen Stützpunkten gezogen werden. Wie bei einer hohen Filterbreite sind die Auswirkungen hauptsächlich in Bereichen hoher Aktivität zu erkennen. Langsame Veränderungen in den hohen Energiebereichen usw. können auch hier meist trotzdem noch nachvollzogen werden.



*Abbildung 11: Snip: Anpassung der Iterationen nach oben
(grün: Referenz, rot: Angepasste Kurve)*

4.1.2 ParabEnv

- *Breite des Mittelwertfilters*

Eine Anpassung der Filterbreite hat bei ParabEnv wie im Kapitel „Vorgehen“ erwähnt so geringe Auswirkungen, dass sie fast vernachlässigbar sind. Eine genaue Gegenüberstellung ist in Abbildung 7 zu sehen. Dennoch möchte ich kurz auf die Auswirkungen eingehen, da doch ein paar wesentliche Unterschiede im Vergleich zu Snip bestehen.

Eine Anpassung des Parameters nach unten hat im Gegensatz zu Snip keine schwächere, sondern eine stärkere Glättung, bzw. zumindest eine flachere, herabgesetzte Kurve zum Ergebnis. An manchen Stellen ließ sich außerdem so etwas wie eine Stauchung hin zu den Schnittstellen erkennen, dies könnte aber auch nur daran liegen, auf welcher Seite einer Flanke sich die Stelle befindet.

Im Umkehrschluss führt eine größere Filterbreite zu einer schwächeren Glättung und einer welligeren, höher gelegenen Kurve. Statt einer Stauchung lässt sich eine leichte Streckung des Untergrundes an wenigen Stellen sehen. Das könnte hier ebenso lediglich an den jeweiligen Flanken liegen.

Ansonsten sind die Unterschiede wirklich nur minimal, weshalb auch nur eine Filterbreite weiter betrachtet wurde, statt wie bei Snip gleich drei.

Das Augenmerk sollte demnach nicht auf diesen Parameter gelegt werden. Er kann aber dennoch beim Feinschliff der Untergrundkurve behilflich sein.

- x : Erster Parameter zur Kontrolle der Breite der Parabeln

X kommt sehr nahe an die Auswirkungen von m oder $niter$ bei Snip heran. Wenn ein niedriger Wert eingesetzt wird, sind die von unten angepassten Parabeln sehr schmal. Folglich kann der Untergrund einfacher dem eigentlichen Signal folgen und sich daran anpassen. Der Untergrund nähert sich stärker den Stützstellen und wölbt sich zwischen ihnen nach oben, falls dort ein Peak, oder eine Erhebung vorhanden ist. An diesen Stellen bilden sich die Erhebungen meist an der linken Flanke eines Peaks oder einer Erhöhung im Signal, um dann in Richtung Stützstelle wieder schnell abzufallen. Das hängt, wie Kajfosz in seiner Arbeit erwähnte, damit zusammen, dass x für die Form des Untergrundes in seinen konvexen Teilen zuständig ist. Wenn die Parabeln schmaler sind, fällt die Kurve in den konvexen Teilen also schneller ab, wodurch sich die Erhebungen in Richtung linker Flanke zu lehnen scheinen. Wie im Bereich zwischen 90 und 100keV in Abbildung 12 zu sehen ist, werden dadurch die Peaks auf dem Gefälle um einiges stärker angegriffen, darum lehnen sich die Erhebungen nicht nur hin zur linken Flanke, sondern sie werden generell im Bereich von Steigungen stärker ausgebildet. Man sollte den Parameter nicht zu klein wählen, oder zumindest mit der passenden Wahl anderer Parameter ausgleichen. Denn sonst kann ein enormes Ungleichgewicht innerhalb des Untergrundes auftreten, weil die Punkte des Signals in Steigungen mehr Einfluss auf den Untergrund nehmen. Dies sollte eventuell bei der Auswertung in Betracht gezogen werden.

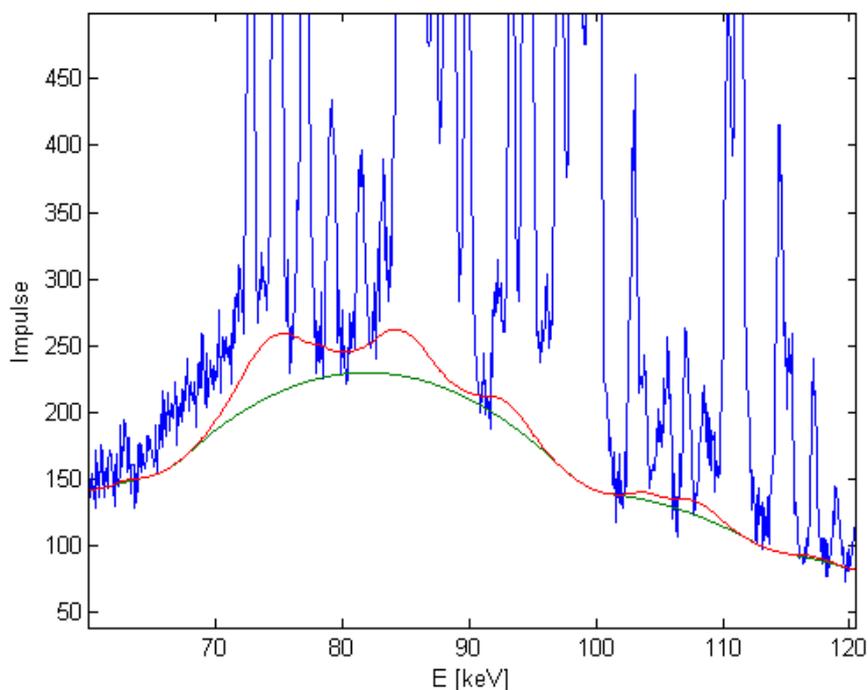
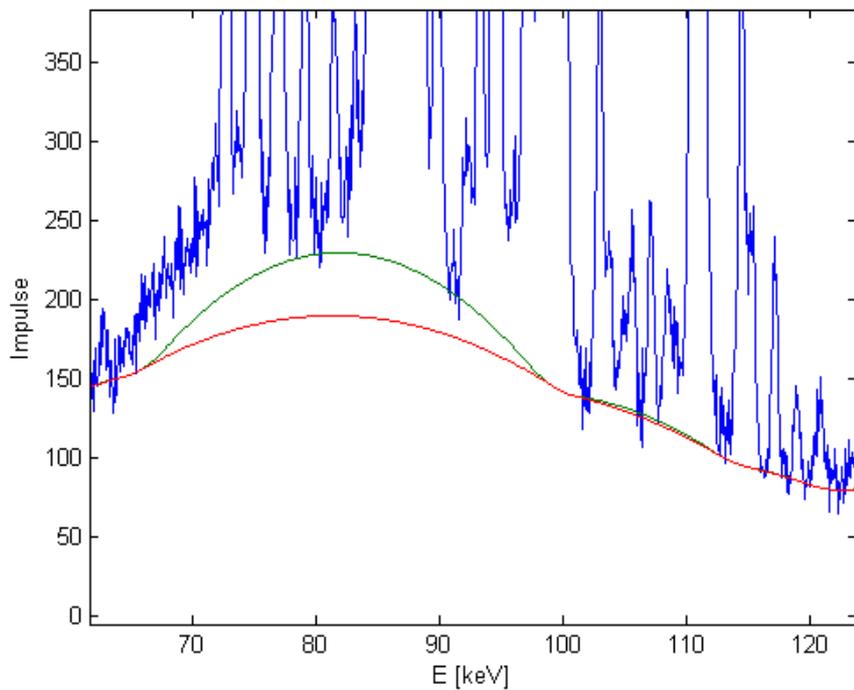


Abbildung 12: *ParabEnv: Anpassung von x nach unten (grün: Referenz, rot: Angepasste Kurve)*

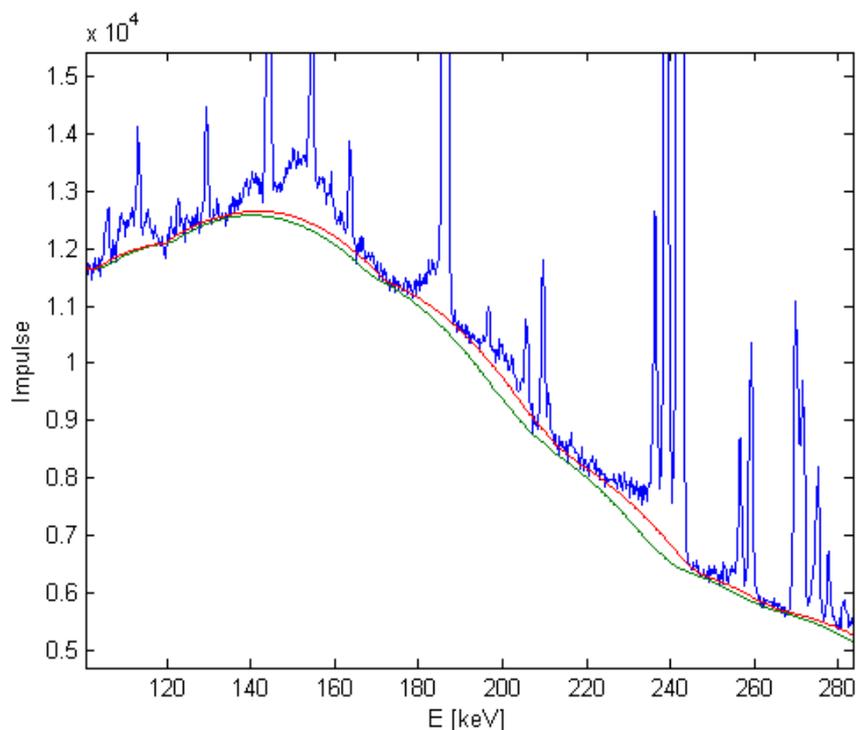
Breitere Parabeln bewirken einen flacheren, glatteren Untergrund. Im Vergleich zu Snip fällt die Glättung deutlich stärker aus. Erhebungen wie sie in Abbildung 13 zu sehen sind werden schnell fast komplett glatt gebügelt. Dafür ist das Ergebnis meist sehr fein und ohne größere Sprünge, selbst bei stark fluktuierendem Signal. Die breiteren Parabeln drücken die Kurve von unten flach. Auch hier sind die Auswirkungen in den konvexen Teilen am stärksten. Die Kurve verschiebt sich hierbei im Gegensatz zu den Auswirkungen bei kleinem x sich nicht in Richtung der linken Flanken und Steigungen, sondern ist im Bereich der rechten Flanke und Gefällen dominanter. Die Kurve flacht hier etwas langsamer ab.



*Abbildung 13: ParabEnv: Anpassung von x nach oben
(grün: Referenz, rot: Angepasste Kurve)*

- *y*: Zweiter Parameter zur Kontrolle der Breite der Parabeln

Da *y* der direkte Gegenspieler zu *x* ist, fehlt bei kleiner Parabelbreite das nötige Gegengewicht von oben an den Untergrund. Die Auswirkungen einer Anpassung nach unten sind ähnlich denen eines niedrigen *x*-Wertes, aber nicht ganz so stark. Der Untergrund wird zwischen den Stützstellen leicht aufgefüllt und weist allgemein eine schwächere Glättung auf. Die Sprünge im Untergrund, vor allem bei den Stützstellen treten stärker hervor und der Untergrund nähert sich mehr dem Signal. Im Gegensatz zu *x* behält die Kurve grob ihre Form und bildet unter Peaks nur minimal Erhöhungen aus. Da die Bereiche um die Wendepunkte an den Stützstellen aufgebläht werden, hat es dort oft den Anschein, als wären die Stützstellen verschoben, in Steigungen nach links und in Gefällen nach rechts. Das lässt sich zum Teil ähnlich wie bei *x* darauf zurückführen, dass *y* den Untergrund in seinen konkaven Teilen formt. Da bei kleinem Wert der „Druck“ auf diese Bereiche geringer ist, kann der Untergrund eher gewölbt werden. Dies führt dazu, dass im Bereich der Stützstellen die Untergrundkurve steiler wird und in Steigungen früher, als auch im Gefälle später auf die Stützstellen zu treffen scheint. Darum sieht es so aus, als seien sie verschoben.



*Abbildung 14: ParabEnv: Anpassung von *y* nach unten (grün: Referenz, rot: Angepasste Kurve)*

Die Erhöhung der Parabelbreite macht sich in diesem Fall etwas anders zwischen den Stützstellen bemerkbar. Statt wie bisher häufig den Untergrund in diesen Bereichen einfach nur platt zu drücken und zu glätten, wird er durch diesen Parameter nach unten gedrückt, sodass eher Kuhlen auftreten. Der Untergrund wird also von oben geglättet und zusätzlich noch etwas stärker nach unten gedrückt, als bei den meisten Parametern bisher. Wenn der Druck von oben durch breitere Parabeln zunimmt, ist es leicht verständlich, dass der Untergrund nachgibt und absinkt. An manchen Stellen wurde beobachtet, dass der Untergrund, trotz größerem Abstand zum Signal, den Stützstellen besser zu folgen scheint. Es könnte aber auch sein, dass dies nur den Anschein hat, da die Referenzkurve näher bzw. ggf. sogar im Signal verläuft und deswegen überdeckt wurde. Die Untergrundkurve ist durch ihren niedriger gelegenen Verlauf besser vom Signal zu unterscheiden.

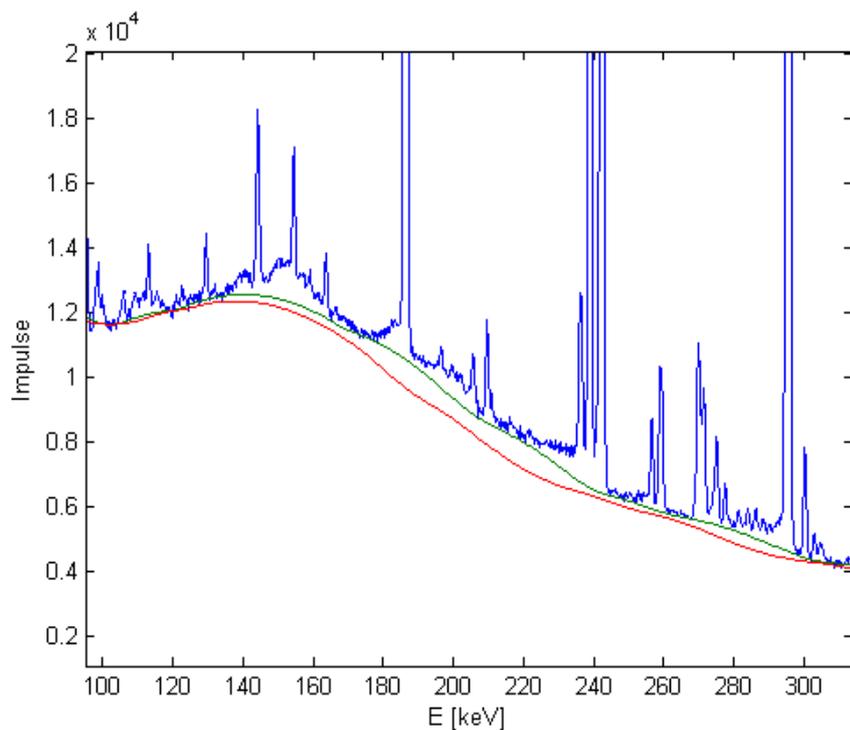


Abbildung 15: ParabEnv: Anpassung von y nach oben (grün: Referenz, rot: Angepasste Kurve)

- *u*: Dritter Parameter zur Kontrolle der Breite der Parabeln

U ist beinahe äquivalent zu *x* und *y*. Die Parameter haben fast den gleichen Effekt, nur ist *u* etwas schwächer. Bei niedrigen Werten passt sich der Untergrund stärker dem Signal an und wird ebenfalls zwischen den Stützstellen aufgebläht, allerdings sind die Erhebungen eher voll und rund und nicht so spitz wie bei dem anderen Beiden, vergleiche auch stellvertretend Abbildungen 12 und 16. Die Sprünge und Winkel an den Stützstellen sind nicht ganz so stark ausgeprägt, wodurch im Gegensatz zu *x* eine feinere Kurve entsteht. Die Unterschiede zu *x* und *y* zeichnen sich deshalb ab, da *u* den Untergrund nicht in eine gewisse Richtung beeinflusst. Die Anderen Parameter haben beide dazu geneigt die Kurve in eine bestimmte Richtung zu drängen. Da *u* Einfluss auf beide Parabelarten hat, gleicht es sich aus.

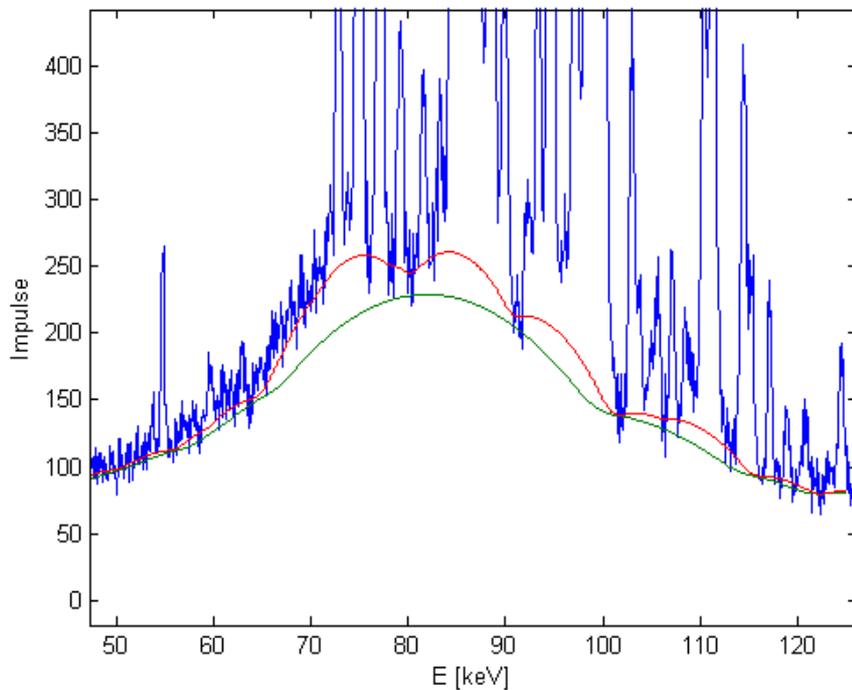


Abbildung 16: ParabEnv: Anpassung von u nach unten (grün: Referenz, rot: Angepasste Kurve)

Höhere Werte von u haben eine stärkere Glättung zur Folge, wodurch die Kurve flacher und feiner wird, ähnlich einer Mischung aus x und y . Dies ist besonders schnell in Bereichen mit Erhöhungen im Untergrund zu sehen, wie in Abbildung 17. In Bereichen, wo die Erhebungen nicht so markant hervortreten, ist die Glättung nicht ganz so stark, wie ebenfalls in der Abbildung erkannt werden kann. Knicks und ähnliches werden aufgeweicht. Die Proportionen bleiben weitestgehend erhalten und auch die Stützstellen werden fast an derselben Stelle durchlaufen. Es bilden sich keine Kuhlen und der Untergrund flacht auch nicht in Bereichen langsamer ab, wie bei den anderen Parametern. Dadurch zeigt sich wieder, dass sich die Untergrundkurve durch den Parameter gleichmäßiger beeinflussen lässt, als mit x und y .

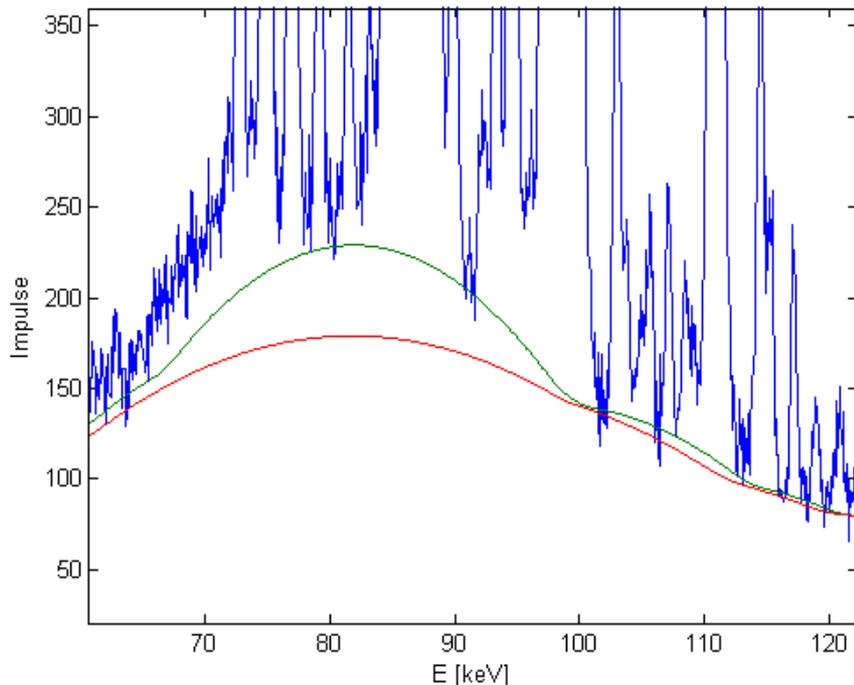


Abbildung 17: ParabEnv: Anpassung von u nach oben (grün: Referenz, rot: Angepasste Kurve)

- v : *Vierter Parameter zur Kontrolle der Steifheit der Parabeln*

Dies ist der wohl am schwierigsten zu wählende Parameter. Es sind hauptsächlich Werte zwischen null und eins zulässig, allerdings können schon vermeintlich kleine Werte dazu führen, dass der Untergrund nahezu als Gerade dargestellt wird, wie es sich in Abbildung 19 zeigt. Werte ab 0,5 sollten schon nur mit Bedacht gewählt werden. Stattdessen kann man sich auch dazu entscheiden den Parameter gar nicht zu belegen und ihm den Wert null zuzuweisen. Die Auswirkungen können häufig von den anderen Parametern kompensiert werden, sodass v nicht unbedingt benötigt wird.

Wenn dieser jedoch geschickt gewählt wird, kann eine zusätzliche Glättung über den kompletten Untergrund erreicht werden, welche anders kaum rekonstruierbar ist. Weil er für die Steifheit der Parabeln zuständig ist, ist das Ergebnis der Glättung ähnlich dessen eines hohen x -Wertes. Die Kurve fällt zu höheren Bereichen des Intervalls langsamer ab. Allerdings ist der Effekt viel stärker als bei x . Darum degeneriert die Kurve auch so schnell zu einer Geraden.

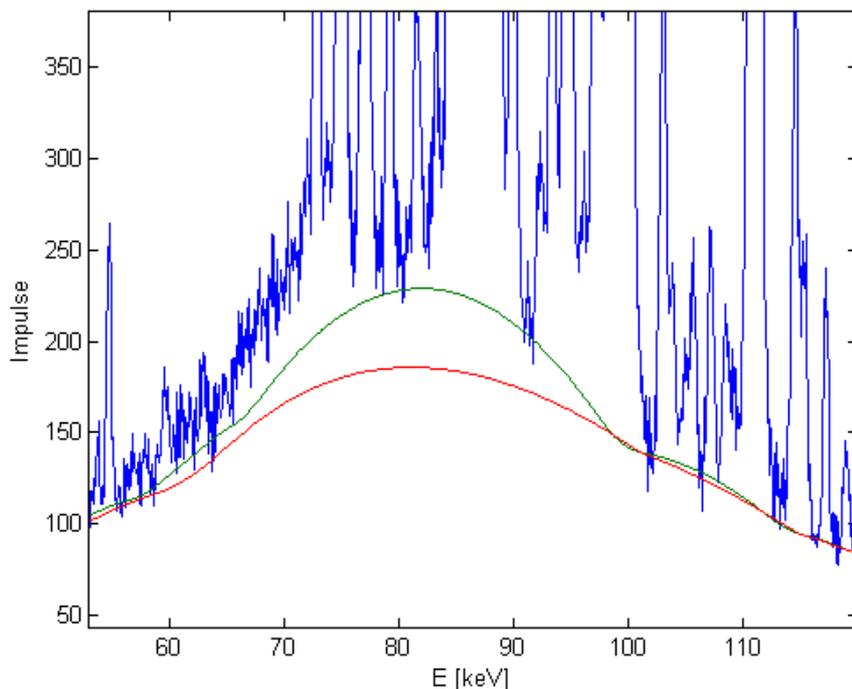


Abbildung 18: *ParabEnv: Anpassung von v (grün: Referenz, rot: Angepasste Kurve)*

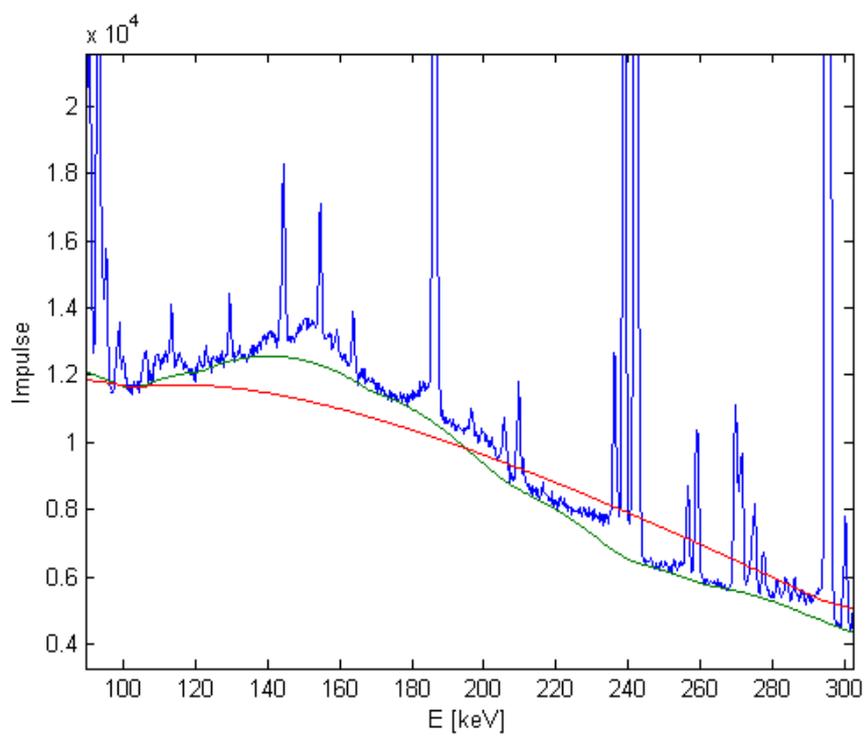


Abbildung 19: ParabEnv: Auswirkung von zu hohem Parameter v (grün: Referenz, rot: Angepasste Kurve)

4.2 Festlegung des Rahmens

4.2.1 Snip

Die in der Tabelle genannten Abbildungen sind im Anhang aufgeführt.

Aktivität im Bereich	Filterbreite m	Niter-Untergrenze	Niter-Obergrenze	Abbildung
Hohe Aktivität	5	80	500	Abbildung 28
	10	50	200	Abbildung 29
	15	20	120	Abbildung 30
Mittlere Aktivität	5	60	300	Abbildung 31
	10	30	150	Abbildung 32
	15	20	70	Abbildung 33
Niedrige Aktivität	5	30	-	Abbildung 34
	10	15	-	Abbildung 35
	15	10	-	Abbildung 36

Im Bereich niedriger Aktivität macht es kaum Sinn eine Obergrenze festzulegen, da selbst mit Werten um die Tausend gearbeitet werden kann, siehe hierfür auch die gelisteten Abbildungen. Die Kurven sind dort meist so flach, dass der Untergrund auch mit Geraden beschrieben werden könnte und deshalb die Steifheit der Kurve eine verschwindend geringe Rolle spielt.

4.2.2 ParabEnv

Wie im Vorgehen erklärt, ist es schwierig eine Rahmenbelegung für diesen Algorithmus festzulegen. Stattdessen werden mehrere Beispiele aufgeführt, an denen die Relationen zwischen den einzelnen Parametern erkennbar sind, sodass klar wird, wie sie in Abhängigkeit voneinander gewählt werden können.

Zwischen x und y hat sich ein Verhältnis von ungefähr 1:0,75 herauskristallisiert. U sollte je nach Wahl der x und y Parameter gewählt werden, da er hauptsächlich als Ausgleich dient. Je höher die beiden anderen Parameter desto niedriger u und umgekehrt. Der letzte Parameter v muss nicht unbedingt belegt werden, kann aber den letzten Schliff liefern.

Dennoch sollte, möchte man den Parameter verwenden, niedrig angefangen werden, da bei den meisten höheren Belegungen der Untergrund fast glatt geschliffen wird. Von einer höheren Belegung als 0,5 kann nur abgeraten werden.

Im Folgenden sind ein paar Beispiele aufgeführt, um das Ganze zu veranschaulichen.

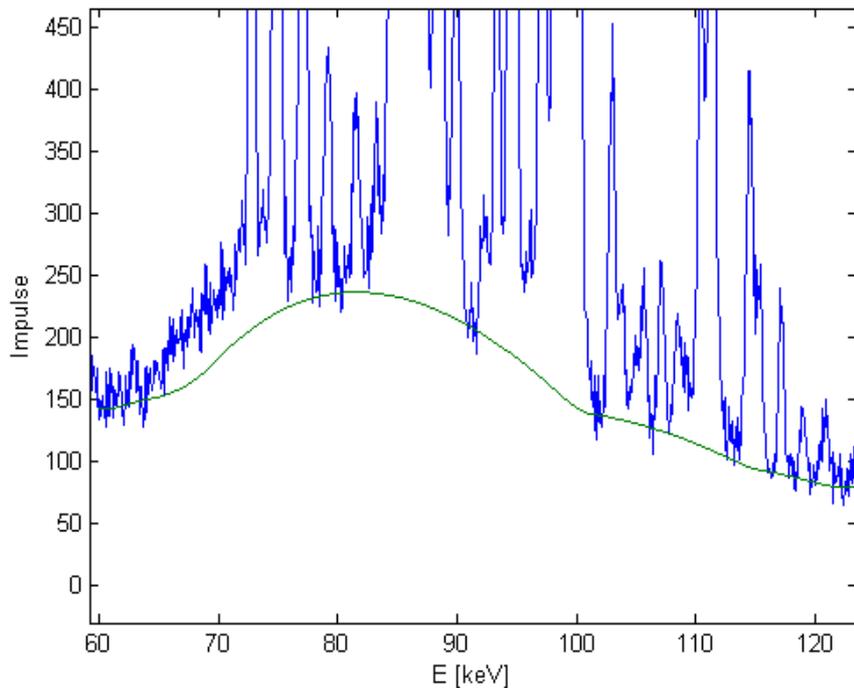


Abbildung 20: In MATLAB:
`parabEnv(E,sp1,Emin,Emax,15,12,400,0.5)`

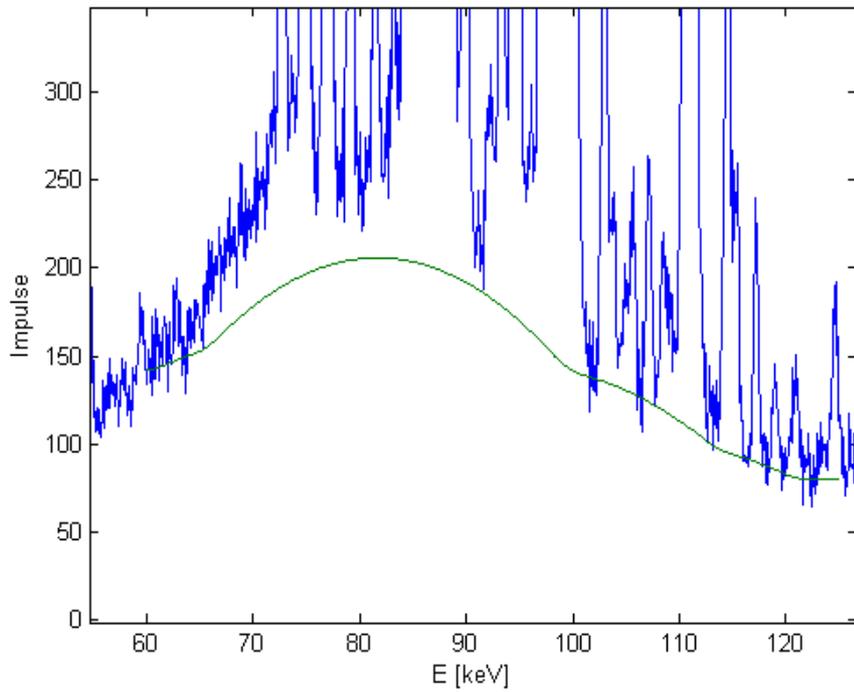


Abbildung 21: In MATLAB:
`parabEnv(E,sp1,Emin,Emax,30,20,200,0)`

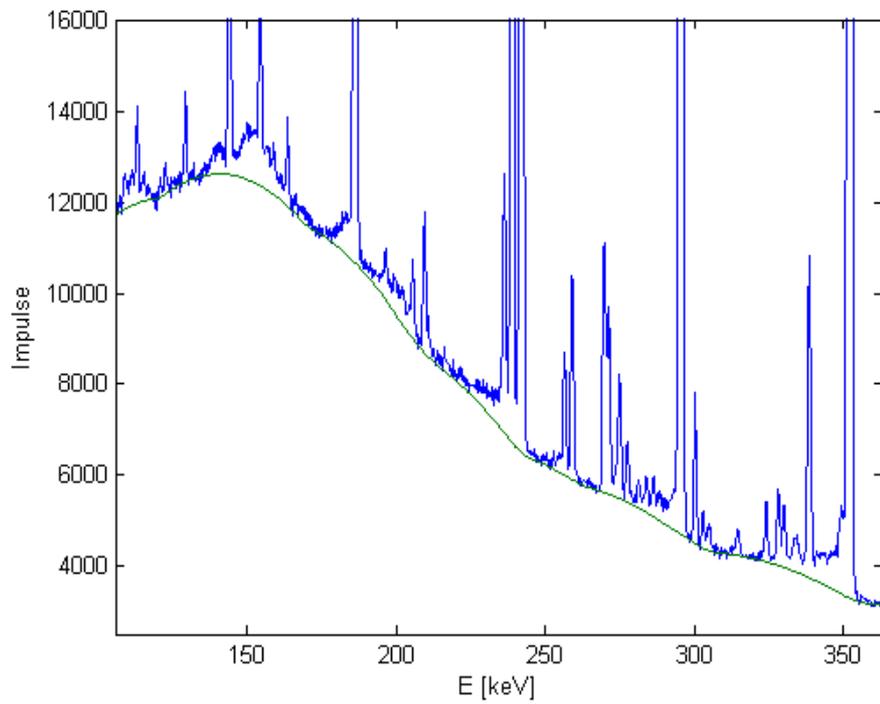


Abbildung 22: In MATLAB:
`parabEnv(E,sp1,Emin,Emax,120,80,8,0.002)`

4.3 Stärken und Schwächen

4.3.1 Snip

Eine klare Stärke von Snip ist seine Einfachheit. Hier wird nur mit zwei selbstbestimmten Parametern gearbeitet. Somit ist das Ganze sehr überschaubar und eignet sich sehr gut zum Einstieg, da auch ohne Vorwissen schnell Ergebnisse erzielt werden. Trotzdem liefert er zumeist relativ schöne Untergrundkurven.

Eine Schwäche ist leider, dass sich schnelle Veränderungen im Spektrum bzw. im Untergrund nur schlecht nachverfolgen lassen, da sich die Kurve nur schlecht an diese Stellen anpassen lässt. Ein Lösungsansatz wäre hier eine geringe Filterbreite gekoppelt mit einer niedrigen Anzahl an Iterationen. Das hat aber wiederum Auswirkungen auf einen Großteil der anderen Bereiche des Intervalls. Vor allem unter Peaks hebt sich die Kurve dadurch an.

Generell sehen viele Untergründe so aus, als wären lediglich die jeweiligen Stützstellen grob miteinander verbunden, wodurch das Erscheinungsbild leiden kann.

Deshalb lässt sich Snip am Besten in Bereichen mittlerer und niedriger Aktivität anwenden. Durch die geringe Anzahl an Parametern und den klaren Auswirkungen der jeweiligen Anpassungen, lässt sich ein Ergebnis relativ gut abschätzen. Anders als bei ParabEnv weißt die Kurve auch deutlich weniger Schwankungen und Sprünge auf, was ästhetischer wirkt.

In Bereichen hoher Aktivität sollte zuerst der andere Algorithmus ausprobiert werden, da Snip einfach das gewisse Maß an Anpassungsfähigkeit fehlt. Um dort trotzdem gute Ergebnisse erzielen zu können, müssen die Intervalle, die Filterbreite und die Iterationen sehr klein gewählt werden. Gegebenenfalls wäre es nötig, den betrachteten Bereich in mehrere solcher Intervalle aufzuteilen, damit der Untergrund gut nachvollzogen werden kann.

4.3.2 ParabEnv

Die Anzahl und Komplexität der Parameter ist ein großer Nachteil gegenüber Snip. Ohne sich über einen längeren Zeitraum mit dem Algorithmus auseinandergesetzt zu haben, ist es fast unmöglich, diesen effizient anzuwenden. Auch wenn man sich schon etwas mit dem Algorithmus auskennt ist es schwierig das Ergebnis vorherzusagen. Zumindest solange man nicht gewissen Richtlinien folgt, wie sie zum Beispiel im Rahmen beschrieben wurden. Denn selbst kleinste Veränderungen der Parameter können schon große Auswirkungen haben, vor allem wenn die Relation untereinander nicht mehr stimmt. Die hohe Anzahl an Parametern ist gleichzeitig aber auch eine Stärke. ParabEnv kann viel besser als Snip an den Untergrund und das Spektrum angepasst werden. Es ist meistens möglich selbst kleinen, oder auch gravierenden Veränderungen schnell zu folgen, ohne dabei Abstriche in anderen Bereichen des Intervalls machen zu müssen. Leider ist das Ergebnis häufig wellig, oder weist Sprünge auf, was etwas stören kann. Auf was ebenfalls geachtet werden sollte, ist eine Übergewichtung verschiedener Regionen im Untergrund. Da fast jeder Parameter die Untergrundkurve in die eine oder andere Richtung beeinflusst, kann ein Ungleichgewicht zu einer groben Fehleinschätzung des Untergrundes führen. Dennoch lässt sich der Algorithmus sehr gut in allen Bereichen eines Spektrums anwenden. Man muss vielleicht an einigen Stellen Abzüge bei der Optik machen, hat dafür aber meist eine sehr gut angepasste Untergrundkurve zum Ergebnis.

5 Diskussion der Ergebnisse

Die Ergebnisse sind durchaus positiv zu bewerten. Es wurden im Grunde alle gesteckten Ziele erreicht.

Die einzelnen Parameter und ihre Auswirkungen wurden betrachtet und festgehalten. Zusammen mit der allgemeinen Beschreibung der Algorithmen, sowohl in ihrer Theorie, als auch ihrer praktischen Umsetzung, lässt sich ein genaues Bild der Funktionsweise und der Anwendungsgebiete der Algorithmen erstellen. Damit lassen sich vorgegebene Spektren analysieren und die wichtigsten Parameteranpassung abschätzen.

Um darauf aufbauend dann die Untergrundbestimmung durchzuführen, wurden für alle möglichen vorkommenden Bereiche und gewünschten Fenstergrößen Rahmenbelegungen definiert, innerhalb derer optimistisch mit einem akzeptablen Ergebnis gerechnet werden kann. Falls das Ergebnis nicht den Erwartungen entspricht, kann das Wissen um die Parameterauswirkungen benutzt und die Belegung angepasst werden, um beim nächsten Versuch ein besseres Ergebnis zu erzielen.

Um von vornherein einen Algorithmus zu empfehlen, können die Stärken und Schwächen der Algorithmen verglichen werden, damit pro Intervall und Spektrum der passende Algorithmus ausgewählt und angewandt werden kann.

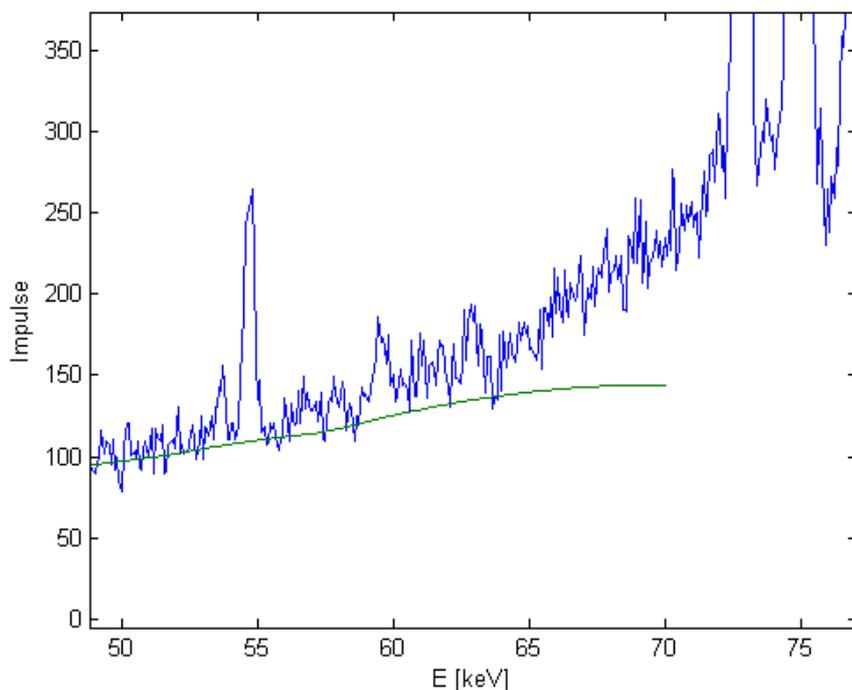


Abbildung 23: Schlechte Wahl des Intervalls für ParabEnv

Es gibt aber auch einige Stellen, die besonders zu beachten sind. Hier sollte in Zukunft angeknüpft und an Verbesserungen gearbeitet werden.

Zum Beispiel lässt die Ästhetik beider durch die Algorithmen erzeugter Kurven etwas zu wünschen übrig. Die Ergebnisse von ParabEnv sind oft sehr wellig und weisen Knicke auf, während Snip häufig zu gerade und abgehackte Kurven liefert. Auch die Intervallgrenzen sind bei beiden Algorithmen noch Problemzonen. Wie zum Beispiel in Abbildung 23 erkannt werden kann, kommt es bei ParabEnv, wenn die Intervallgrenze in eine Steigung gelegt wird dazu, dass die Kurve plötzlich den Verlauf des Signals verlässt und in einiger Entfernung ausläuft.

Bei Snip tritt ein weiteres Problem auf. Hier verfällt die Kurve an den Intervallgrenzen meist in ein sehr sprunghaftes Verhalten. Da das Intervall generell immer etwas größer gewählt werden sollte, als der gewünschte zu untersuchende Bereich und die Grenzen meist nur geringe Relevanz haben, sind das vorerst alles nur optische Verbesserungen. Diese könnten durch Überarbeitung der Implementierung der Algorithmen erreicht werden. Obwohl durch die Anzahl an Testspektren eine hohe Abdeckung erreicht wurde, könnten die Ergebnisse am Besten in einer Testumgebung auf ihre Zuverlässigkeit und ihre Gültigkeit überprüft werden. Dadurch wäre sichergestellt, dass zum Beispiel die Rahmenbelegungen zum Großteil passable Ergebnisse liefern und Anpassungen wären einfach umzusetzen. Denn die Ergebnisse basieren bisher nur auf meiner persönlichen Einschätzung und wurden mehr oder weniger willkürlich festgelegt. Es fehlt also das gewisse Expertenwissen, um sie endgültig zu bestätigen, oder gar abzulehnen. Ein guter Test wäre es die Ergebnisse mit anderen Algorithmen, wie sie zum Beispiel in der Einleitung erwähnt wurden, abzugleichen. Da diese Algorithmen bereits zahlreich getestet wurden und schon Anwendung in ihrem speziellen Bereich finden, ließe sich damit gut ein gewisser Mehrwert der beiden untersuchten Algorithmen feststellen, oder aber auch widerlegen.

Dennoch sollten die Algorithmen so wie sie zum jetzigen Zeitpunkt sind, Anwendung finden können. Die nötigen Mittel dazu stehen zur Verfügung.

6 Zusammenfassung und Schlussfolgerung

Der erste Schritt bestand darin die Arbeit von Kajfosz und Ryan, in ihren wichtigsten Teilen, noch einmal aufzuführen. Im Anschluss daran wurde die praktische Umsetzung beschrieben und die Unterschiede zur Theorie aufgezeigt.

Des Weiteren wurden die Auswirkungen einzelner Parameter mithilfe von Referenzkurven verfolgt und analysiert. Für die Ermittlung eines passenden Rahmens der Parameter, wurden Gammaspektren in drei grobe Bereiche eingeteilt. In diesen Bereichen wurden dann Rahmenbelegungen für die Filterbreiten $m = 5$, 10 und 15 bei Snip und $m = 10$ bei ParabEnv ermittelt. Zuletzt wurden Stärken und Schwächen beider Algorithmen zusammengetragen.

Die Ergebnisse sind zufriedenstellend und liefern einen guten Einstiegspunkt in ein Gebiet, das bisher nur schwach besetzt ist. Es sind nicht viele Algorithmen frei verfügbar, oder so vielseitig einsetzbar, ohne vermutlich große Änderungen daran vorzunehmen. Die Ergebnisse mögen zwar in einem groben Rahmen gehalten sein, zeigen aber jetzt schon, was die Algorithmen leisten können und dass man mit ihnen gut arbeiten kann. Eine fachliche Untersuchung der Ergebnisse anzuschließen würde den Kontext verbessern, in dem die Algorithmen derzeit zu anderen Produkten stehen und ob sie möglicherweise konkurrenzfähig sind.

Zusammenfassend lässt sich sagen, dass beide Algorithmen ihren Platz haben. Es gibt jetzt schon genug Anwendungsgebiete, in denen mit ihnen gearbeitet werden kann.

Als Beispiel sei ein Projekt genannt, das bereits umgesetzt wurde:

Es handelt sich um eine App für mobile Endgeräte, in der Gammaspektren mithilfe einer Nuklidatenbank analysiert werden können. Allerdings fehlte dort bisher das Verständnis dafür, wie die Algorithmen genau arbeiten und wie die Parameter zu belegen sind.

Dadurch wurde ein schneller Einstieg verhindert. Wenn man die Ergebnisse dieser Arbeit mit solchen Projekten verknüpfen könnte, wären Analysen vielseitiger möglich und einsetzbar. Es sind folglich genug Ansatzpunkte für weiterführende Projekte vorhanden.

7 Literaturverzeichnis

- [1] J. Kajfosz
Nonpolynomial Approximation of Background in X-Ray Spectra
Nuclear Instruments and Methods in Physics Research B22 (1987) 78-81

- [2] C.G. Ryan
SNIP, a statistics-sensitive Background treatment for the quantitative Analysis of PIXE Spectra in Geoscience Applications
Nuclear Instruments and Methods in Physics Research B34 (1988) 396-402

- [3] Wikipedia
Stützstelle
<https://de.wikipedia.org/wiki/St%C3%BCtzstelle>
Letzter Zugriff: 29.09.2016

- [4] Persönliche Kommunikation mit Oliver Kalthoff. Heilbronn, über den gesamten Zeitraum der Arbeit

8 Anhang

8.1 Abbildungen

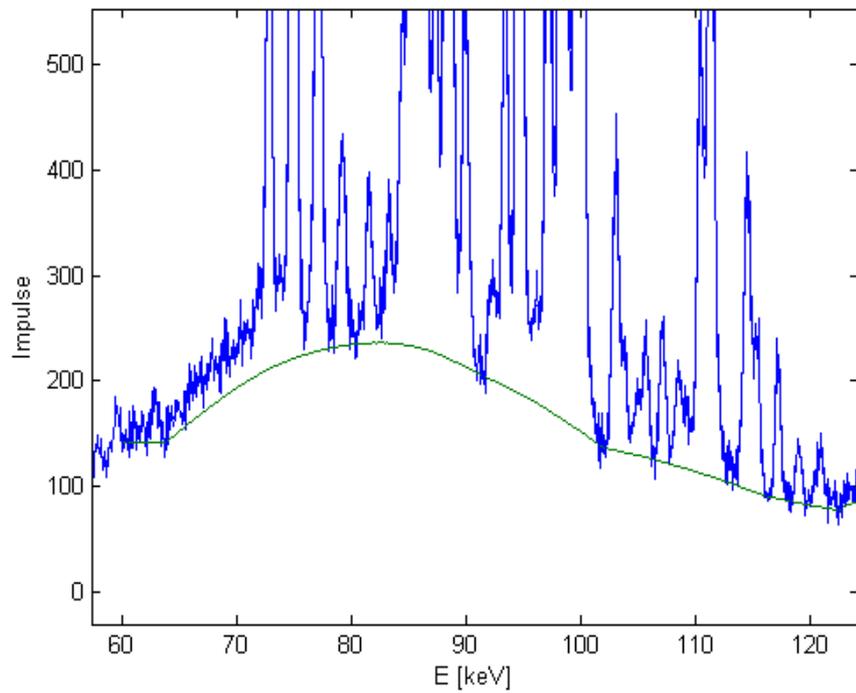


Abbildung 24: Snip-Referenzkurve:
 $snipbg(E,sp1,5,250,E_{min},E_{max})$

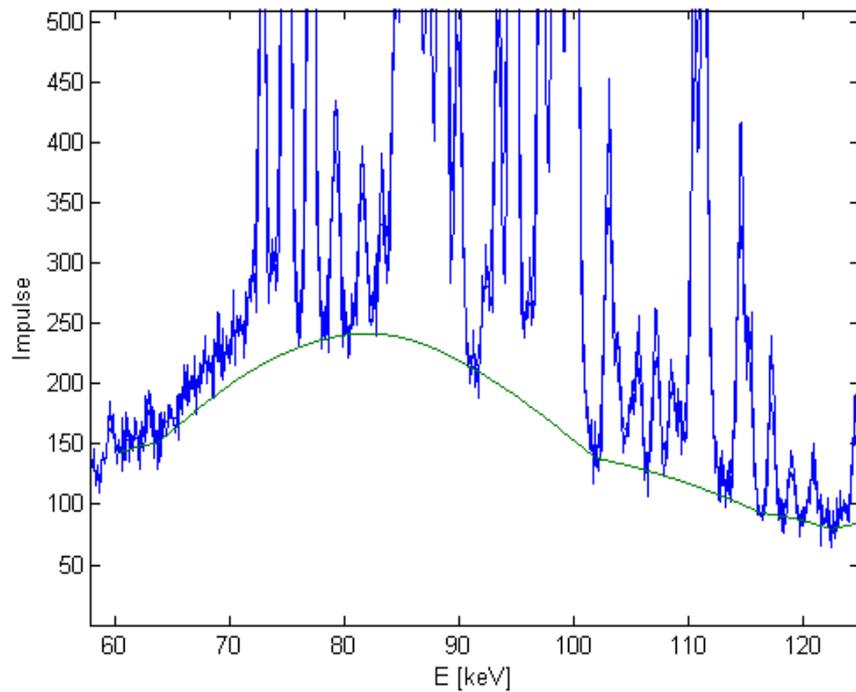


Abbildung 25: Snip-Referenzkurve:
 $snipbg(E,sp1,10,80,E_{min},E_{max})$

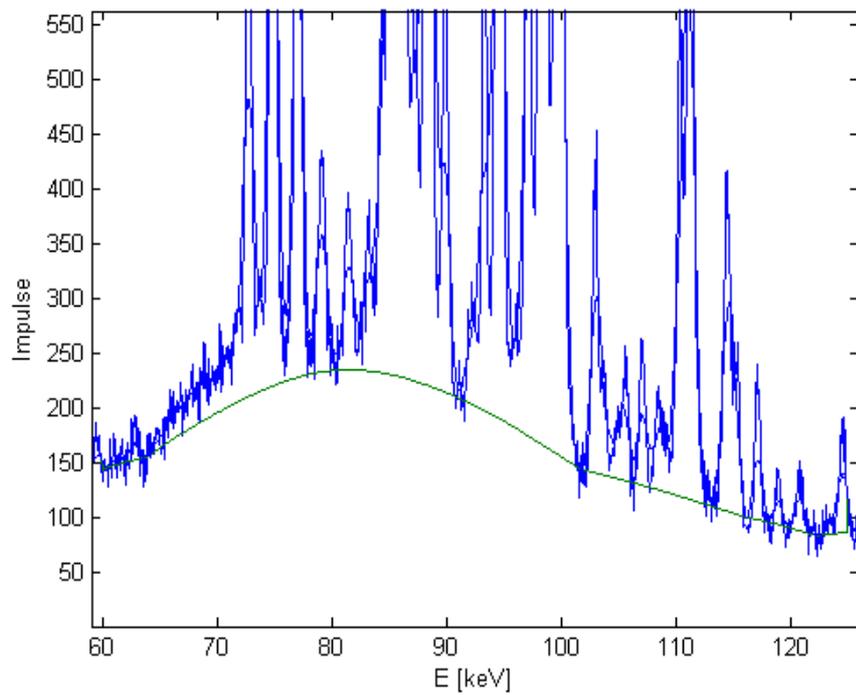


Abbildung 26: Snip-Referenzkurve:
 $snipbg(E,sp1,15,60,Emin,Emax)$

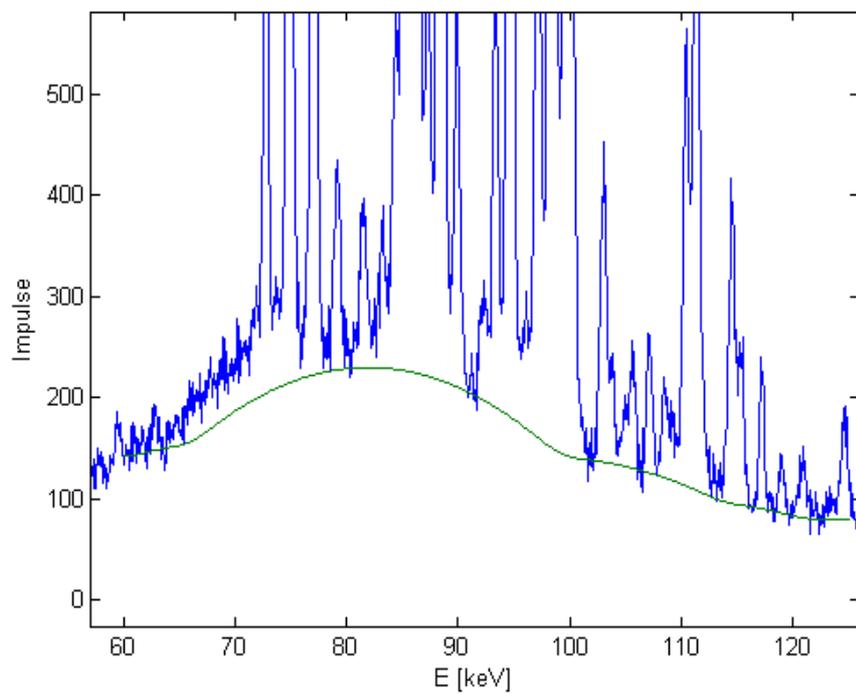


Abbildung 27: ParabEnv-Referenzkurve:
 $parabEnv(E,sp1,Emin,Emax,10,60,45,37,0.001)$

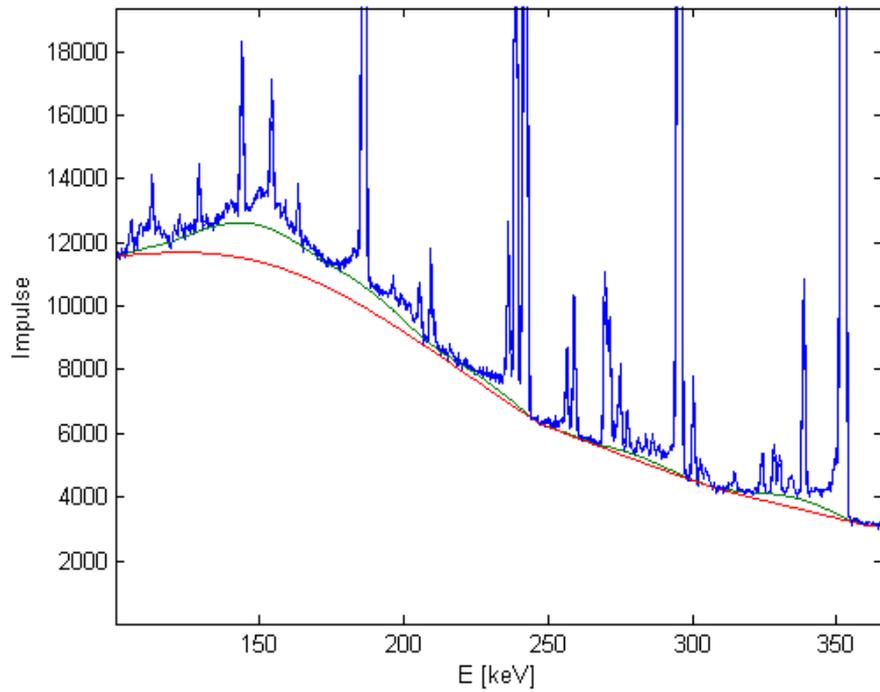


Abbildung 28: Rahmen bei hoher Aktivität, Filterbreite $m=5$

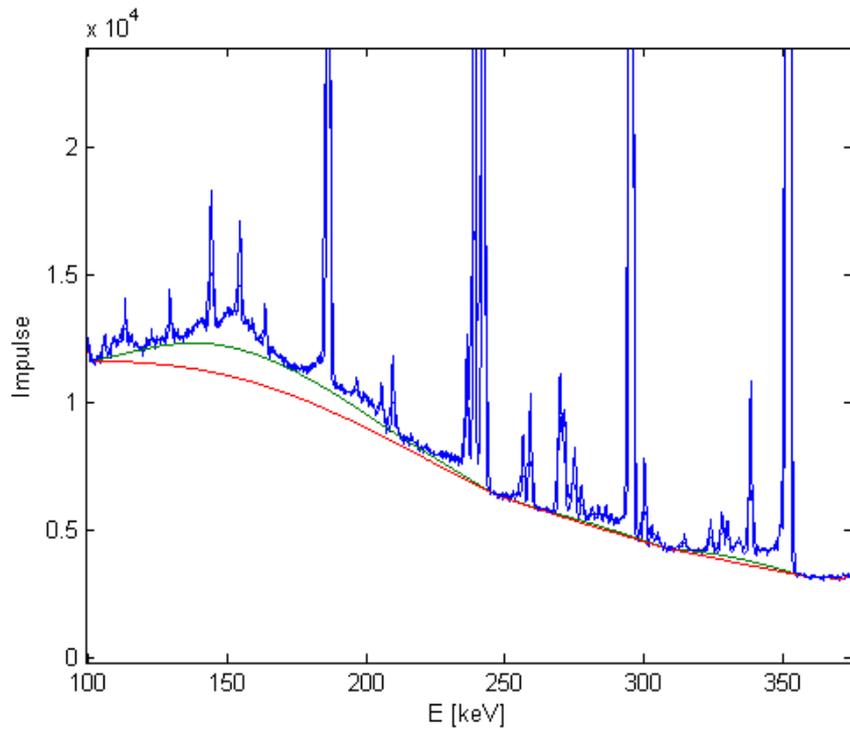


Abbildung 29: Rahmen bei hoher Aktivität, Filterbreite $m=10$

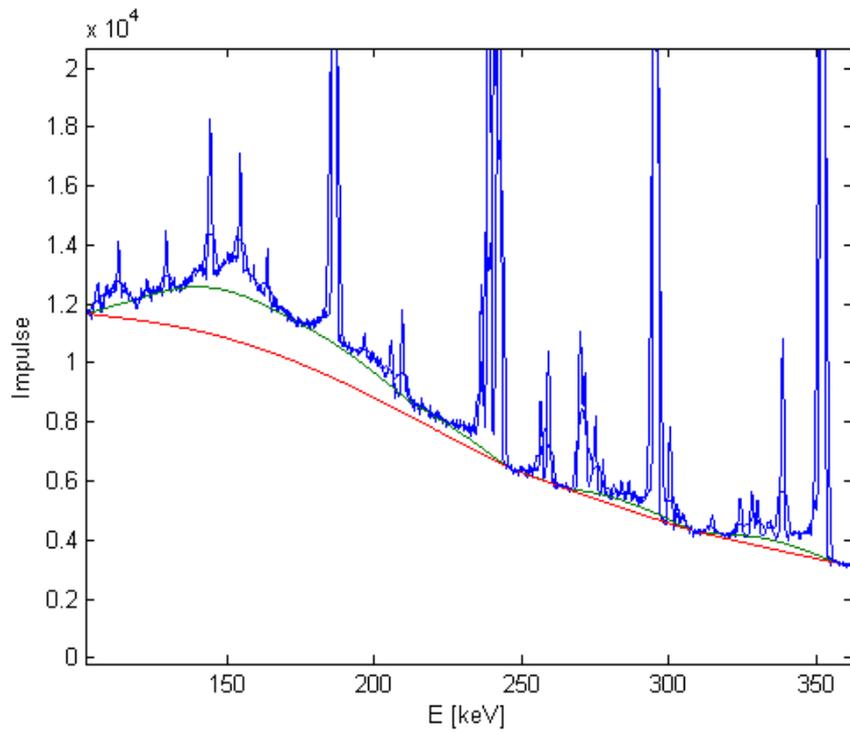


Abbildung 30: Rahmen bei hoher Aktivität, Filterbreite $m=15$

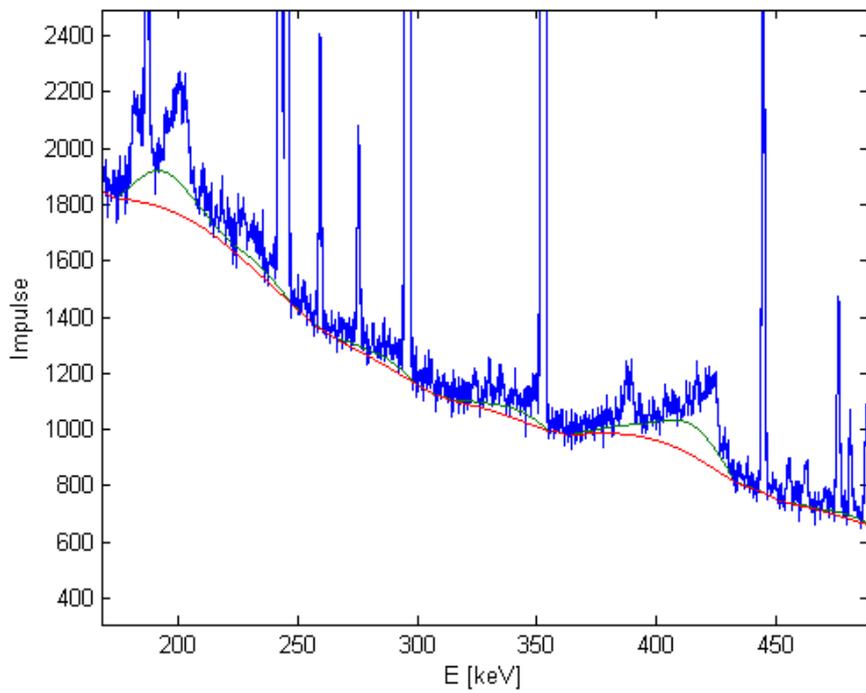


Abbildung 31: Rahmen bei mittlerer Aktivität, Filterbreite $m=5$

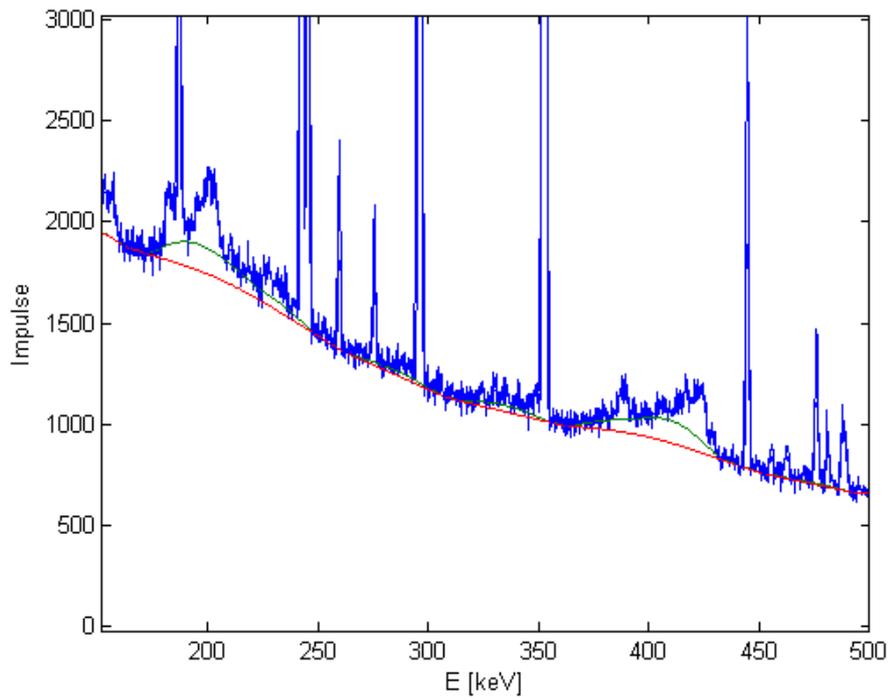


Abbildung 32: Rahmen bei mittlerer Aktivität, Filterbreite $m=10$

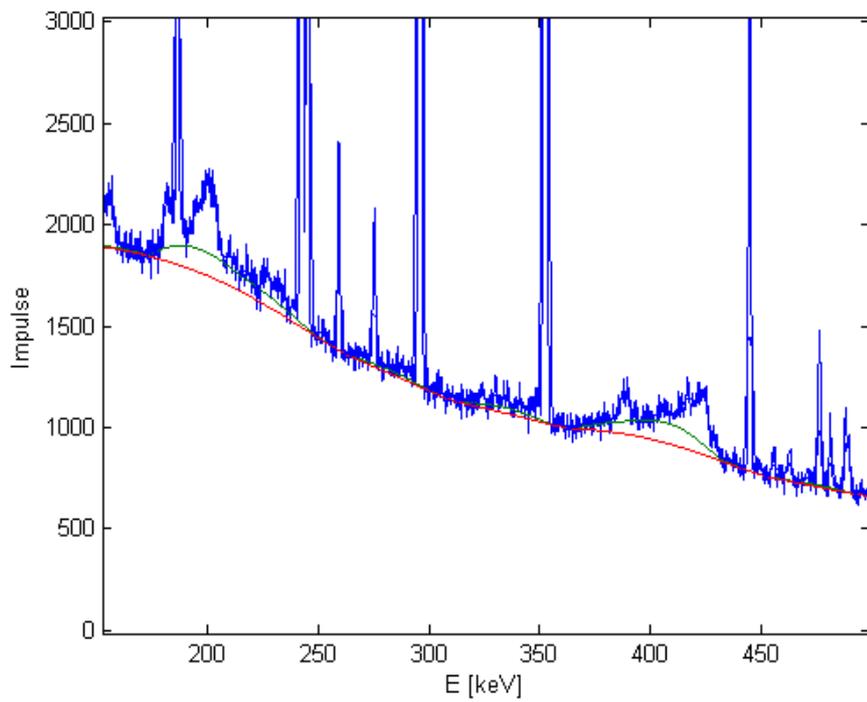


Abbildung 33: Rahmen bei mittlerer Aktivität, Filterbreite $m=15$

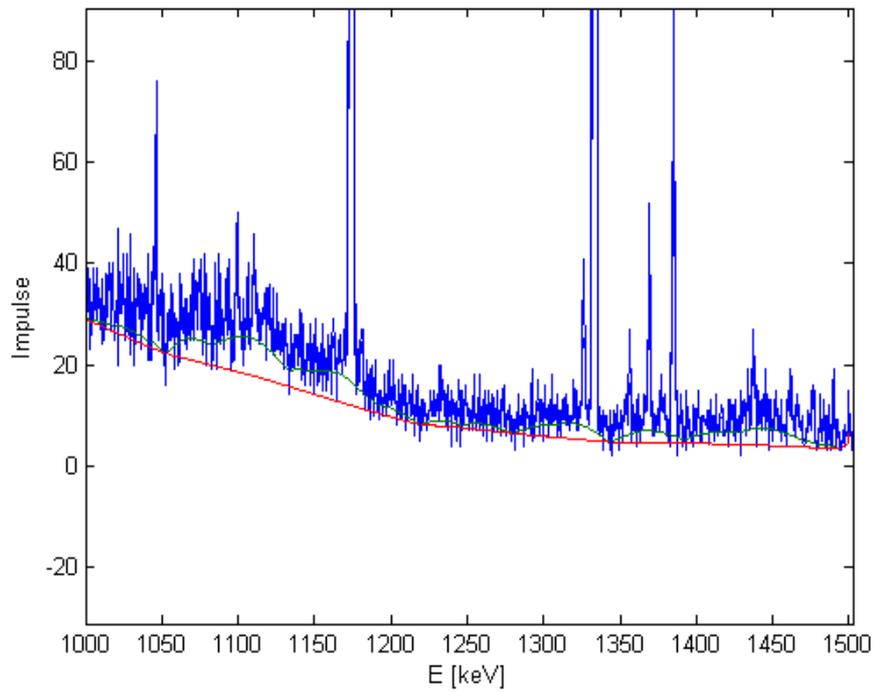


Abbildung 34: Rahmen bei niedriger Aktivität, Filterbreite $m=5$

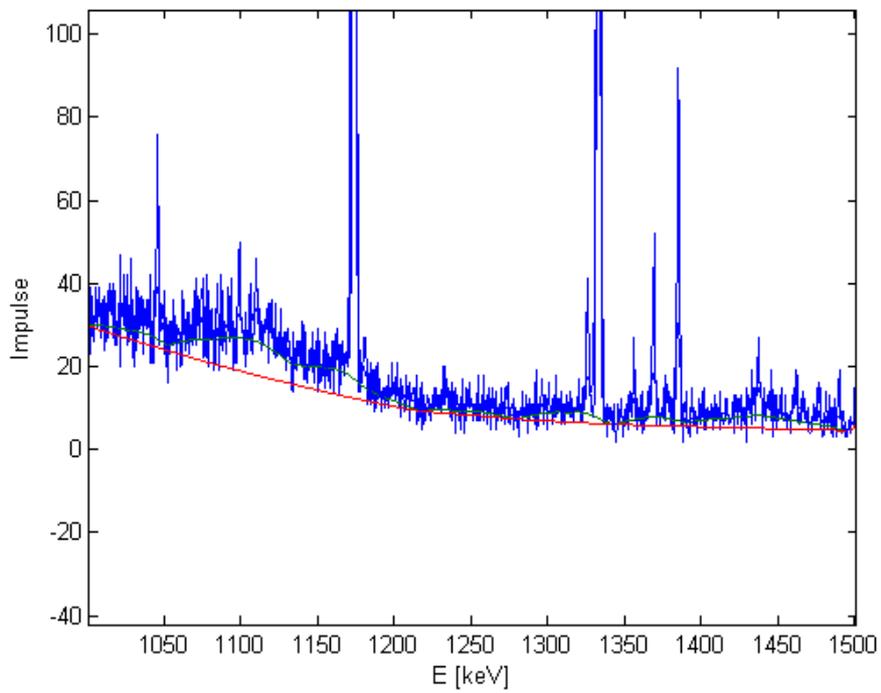


Abbildung 35: Rahmen bei niedriger Aktivität, Filterbreite $m=10$

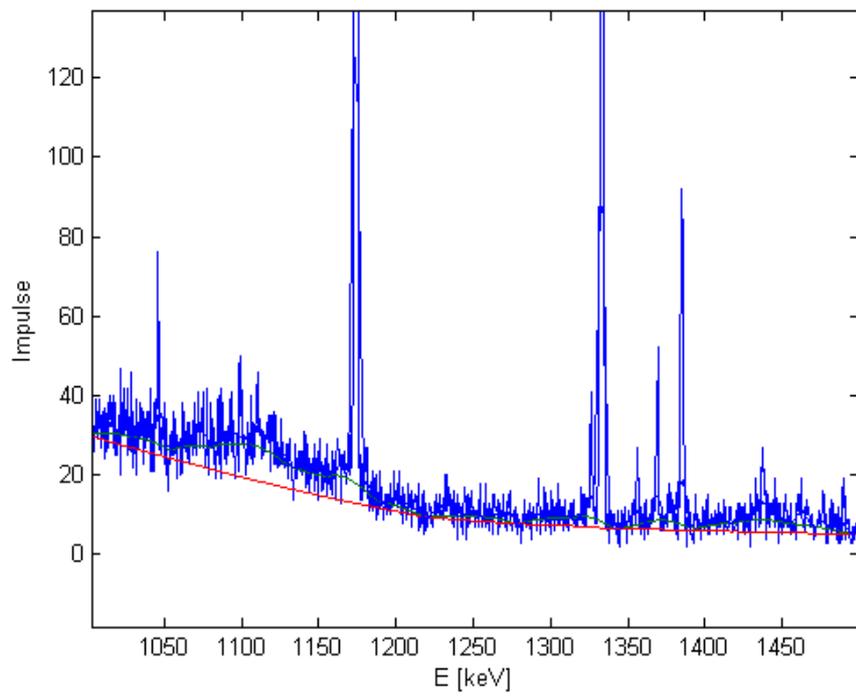


Abbildung 36: Rahmen bei niedriger Aktivität, Filterbreite $m=15$

8.2 Code

SNIP:

```
function splb = snipbg(E, spl, m, niter, Emin, Emax)

% Input
% E:      Energie (d.h. Abszisse)
% spl:    Signal
% m:      Breite des gleitenden Mittelwertfilters
% niter:  Zahl der Iterationen
% Emin:   linke Intervallgrenze
% Emax:   rechte Intervallgrenze
%
% Output
% splb:   Untergrund von spl

nreduc = 8; % Konstante
dE = (E(end) - E(1))/(length(E) - 1);

% Wurzel, anschl. Glättung durch Savitzky-Golay-Filter, Filterb. m, Grad p

spl = sqrt(max(spl, 0));
p = 0; % NUR Glättung
[spls,~,~,~] = sgderiv1D(E,spl,p,m,-1,1); % -1 und 1 sind Dummy-Variable

% Extrahiere Teil des Spektrums zwischen Emin und Emax

idx = find(E >= Emin - m*dE & E <= Emax + m*dE);
len = length(idx);
wrk = spls(idx); % Spektrum im Intervall [Emin;Emax]

redfac = 1;
for n = 1:niter
    if (n > niter - nreduc)
        redfac = redfac/sqrt(2);
    end
    iw = round(redfac*m);
    for i = m+1:len-m
        i1 = i - iw;
        i2 = i + iw;
        wrk(i) = min(wrk(i), (wrk(i1) + wrk(i2))/2);
    end
end

spl = spl.^2; % Quadrat des Spektrums
spls = spls.^2; % Quadrat der Glättung
splb = wrk.^2; % Quadrat des geglätteten Untergrundes, s.o.

disp(' ')
disp(' Emin Emax Emin< Emax>')
disp([Emin Emax Emin-m*dE Emax+m*dE])

plot(E, spl, E(idx), splb) % Plotte Ergebnis falls gewünscht
xlabel('E [keV]')
ylabel('Impulse')
hold on % Plotte, falls nötig, die Glättung wrks
plot(E(idx), spls(idx))
hold off
```

end

ParabEnv:

```
function yb = parabEnv(E,f,Emin,Emax,m,x,y,u,v)

% Input
% E:   Abszisse
% f:   Signal
% Emin: Abszisse, linke Grenze
% Emax: Abszisse, rechte Grenze
% m:   Breite des Mittelwertfilters in sgderiv
% x,y, Parameter zur Kontrolle der Breite der Parabeln ...
% u,v: ... aus NIM B 22 (1987) 78-81 (Kajfosz)
%
% Output
% yb:  Untergrund

% Glätte Spektrum und extrahiere Bereich zwischen Emin und Emax

f = sqrt(max(f, 0));
p = 0; % NUR Glättung
[s,~,~,~] = sgderiv1D(E,f,p,m,-1,1); % -1 und 1 sind Dummy-Variable
f = f.^2;
s = s.^2;

idx = find(E >= Emin & E <= Emax);
s = s(idx);
len = length(s);
k = 1:len; % Neue Indizes zw. Emin und Emax

% Erster Durchlauf

for i = 1:len
    wrk = s(k)./(1 - (i - k).^2/(x^2*(u + v*i)));
    jdx = find(wrk > 0); % Minimumsuche nur bei ...
    h(i) = min(wrk(jdx)); % ... positivem Argument in Gl. (2)
end

% Zweiter (verbesserter) Durchlauf

for i = 1:len % Nach oben offene Parabeln
    g(i) = min(s(k).*(1 + 3*(i - k).^2/(y^2*(u + v*i))));
end

sp = min(s, g); % s' im Artikel, S. 79

for i = 1:len
    wrk = sp(k)./(1 - (i - k).^2/(x^2*(u + v*i)));
    jdx = find(wrk > 0); % s.o.
    h(i) = min(wrk(jdx));
end

for i = 1:len % Berechne Untergrund
    yb(i) = max(h(k).*(1 - (i - k).^2/(x^2*(u + v*i))));
end

% Plote Ergebnisse falls nötig
```

```

plot(E,f,E(idx),yb)
xlabel('E [keV]')
ylabel('Impulse')

```

```
end
```

SGDeriv:

```

function [ff,df,ddf,t0hilo] = sgderiv1D(t,f,p,m,thmax,thmin)

% Input
% t:      Abszisse
% f:      Signale, für die die 1. und 2. Ableitungen gesucht sind, ARRAY
% p:      Grad des Polynoms, nur p = 0,1,2,3 erlaubt
% m:      Fensterbreite, m = 5,7,9,...
% thmax:  Schwelle für Maxima (mittels zweiter Ableitung)
% thmin:  Schwelle für Minima (mittels zweiter Ableitung)
%
% Output
% ff:     Geglättetes Signal
% df:     Erste (geglättete) Ableitung von f
% ddf:    Zweite (geglättete) Ableitung von f
% t0:     Lage d. Hoch- UND Tiefpunkte von f aus lin. Interpol., siehe Code
%
% Beispiele
%
% [ff,df,ddf,t0hilo] = sgderiv1D(E,sp1,0,11,eps,0.7); Nur Glättung
% [ff,df,ddf,t0hilo] = sgderiv1D(E,sp1,2,13,-0.1,5); UPu-Spektrum

f = log(f); % Logarithmierung wg SG-Filter, gaußförm. Apex

% Berechnung der ersten und zweiten Ableitungen df und ddf nach SG

i = -(m - 1)/2:(m - 1)/2; % Neuindizierung im gleitenden Fenster

switch p
case 0
a0 = 1/m*ones(1,m); % Nur Glättung, 1. u. 2. Ableitungen Null
a1 = zeros(1,m);
a2 = zeros(1,m);
case 1
a0 = 1/m*ones(1,m); % Glättung, nur erste Ableitung ungleich Null
a1 = 12*i/(m*(m^2 - 1));
a2 = zeros(1,m);
case 2
a0 = 3/4*(3*m^2 - 7 - 20*i.^2)/(m*(m^2 - 4));
a1 = 12*i/(m*(m^2 - 1));
a2 = 15*(12*m*i.^2 - m*(m^2 - 1))/(m*(m^2 - 1)*(m^2 - 4));
case 3
a0 = 3/4*(3*m^2 - 7 - 20*i.^2)/(m*(m^2 - 4));
a1 = 15*(5*(3*m^4 - 18*m^2 + 31)*i - 28*(3*m^2 - 7)*i.^3)/...
(m*(m^2 - 1)*(3*m^4 - 39*m^2 + 108));
a2 = 15*(12*m*i.^2 - m*(m^2 - 1))/(m*(m^2 - 1)*(m^2 - 4));
otherwise
warning('Polynomgrad p nicht vorgesehen')
end

len = length(f);

```

```

ff = zeros(1,len); % Initialisierung der Glättung ff, der
df = zeros(1,len); % 1. Ableitung df und der
ddf = zeros(1,len); % 2. Ableitung ddf

ff = conv(f,a0,'same'); % Glättung
df = conv(f,a1,'same'); % 1. Ableitung
ddf = conv(f,a2,'same'); % 2. Ableitung

% Nullstellenberechnung der ersten Ableitung

t0 = []; % Init. Nullstellenvektor
hilo = []; % Init. Vektor der Extremstellen
t0hi = []; % lokale Maxima
t0lo = []; % lokale Minima

k = 1;
for j = 1:len-1
    t1 = t(j); % Gerade zwischen (t1,y1) u. (t2,y2)
    t2 = t(j+1);
    y1 = df(j);
    y2 = df(j+1);
    a = y1 - (y1-y2)/(t1-t2)*t1; % Steigung der Geraden
    b = (y1-y2)/(t1-t2); % y-Achsenabschnitt
    if (-a/b >= t1 && -a/b <= t2)
        t0(k) = -a/b; % Nullst. liegt zw. t1 und t2
        k = k + 1;
    end
end
hilo = interp1(t',ddf',t0'); % lok. Extrema hinr. Krit.

hidx = find(hilo <= thmax); % lokale Maxima oberhalb v. thmax
t0hi = t0(hidx);

lidx = find(hilo >= thmin); % lokale Minima unterhalb v. thmin
t0lo = t0(lidx);

f = exp(f); % Daten wurden zuvor logarithmiert
ff = exp(ff);
df = exp(df);
ddf = exp(ddf);
t0hilo = cat(2,t0hi,t0lo); % Abszissen der Hoch- UND Tiefpunkte

% Plote Ergebnisse, falls nötig

% plot(t,f,t,ff)
% hold on
% plot(t0hi,interp1(t',f',t0hi'),'r^')
% plot(t0lo,interp1(t',f',t0lo'),'bv')
% hold off
% xlabel('E [keV]')
% ylabel('Impulse')

end

```

9 Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die aus fremden Quellen direkt oder indirekt übernommenen Gedanken als solche kenntlich gemacht habe. Die Arbeit habe ich bisher keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher auch nicht veröffentlicht.

(Ort, Datum)

(Unterschrift)