

Die approbierte Originalversion dieser Diplom-/  
Masterarbeit ist in der Hauptbibliothek der Tech-  
nischen Universität Wien aufgestellt und zugänglich.

<http://www.ub.tuwien.ac.at>



The approved original version of this diploma or  
master thesis is available at the main library of the  
Vienna University of Technology.

<http://www.ub.tuwien.ac.at/eng>



TECHNISCHE  
UNIVERSITÄT  
WIEN  
Vienna University of Technology

M A S T E R A R B E I T

„Modellunabhängige Datenanalyse von  
Ultrakleinwinkelstreuung mit polarisierten Neutronen“

Ausgeführt am  
A T O M I N S T I T U T  
der Technischen Universität Wien

unter Anleitung von  
Univ.Prof. Dr. Gerald Badurek  
und  
Dipl.-Ing. Dr. Erwin Jericha

durch  
**Wilfried Mach BSc.,**

Franz-Glaser-Gasse 36  
1170 Wien

Wien, 2013

Wilfried Mach

« **S**<sub>F</sub> I have seen further it is by standing on y<sup>e</sup> shoulders of giants. »

- Sir Isaac Newton

# Abstract

USANSPOL is a neutron scattering technique for polarised neutrons for non-destructive examination of magnetic materials which allows to access scattering structures of a few tenths up to a few tens of a  $\mu m$ . As an advanced development of USANS it is now possible to detect the different magnetic scattering of spin- $|\uparrow\rangle$ - and spin- $|\downarrow\rangle$ -neutrons on magnetic moments of electrons and nuclei. USANSPOL is done by exploiting the magnetic moment of the neutron. Due to the measurement setup, the three-dimensional scattering vector  $\vec{Q}$  of the neutron is projected on a one-dimensional neutron intensity distribution  $P(Q_y)$  which unfortunately cannot be measured directly since it is convolved with the instrument curve  $R(Q_y)$ . Hence, this convolution has to be reversed by deconvoluting the measurable neutron distribution  $I(Q_y)$  with  $R(Q_y)$  to get access to the instrument-independent function  $P(Q_y)$  which is the item of interest. This deconvolution is a well-known challenge in the scientific domain of scattering theory. For the case of USANSPOL a lot of approaches to this problem exist. Nevertheless, all of these approaches have more or less complicated model functions in common which will be fitted to the measured data points. A disadvantage of all these model-dependent approaches is the chance of missing effects which were not considered while creating these functions.

This thesis should be recognised as a *proof of principle* for a model-independent data analysis for USANSPOL because it has never been done before systematically. The thesis shows that model-independent data analysis for USANSPOL is possible and has some neat advantages. Furthermore, some preliminary predictions for sample characteristics are made.

The simultaneous development of a completely different model-independent way of analysing data has increased the credibility of this work [1].

# Zusammenfassung

USANSPOL ist eine effektive Möglichkeit zur zerstörungsfreien Probenanalyse mittels Neutronen mit einem Auflösungsvermögen von wenigen Zehntel bis zu einigen  $\mu m$ . Als Weiterentwicklung von USANS hat man bei dieser Messmethode nun auch die Möglichkeit die unterschiedliche magnetische Streuung von Spin- $|\uparrow\rangle$ - und Spin- $|\downarrow\rangle$ -Neutronen an Kernen und an den magnetischen Momenten der Atome zu analysieren, indem man das magnetische Dipolmoment des Neutrons ausnützt. Der prinzipiell dreidimensionale Streuvektor des Neutrons  $\vec{Q}$  wird bedingt durch den Messaufbau auf eine eindimensionale Neutronenintensitätskurve  $P(Q_y)$  abgebildet, welche jedoch noch mit der Gerätekurve  $R(Q_y)$  zur messbaren Kurve  $I(Q_y)$  gefaltet wird. Diese Faltung rückgängig zu machen um Zugriff auf die vom Messgerät unabhängige physikalische Neutronenverteilung zu bekommen, ist eine bekannte Schwierigkeit im Gebiet der Streutheorie. Im Bereich von USANSPOL führt dies zu den unterschiedlichsten Lösungsansätzen. Immer werden in diesen Ansätzen jedoch mehr oder weniger komplexe Modellfunktionen verwendet, welche an die Messdaten per Ausgleichsverfahren gelegt werden. Dadurch können jedoch Effekte übersehen werden, welche nicht in der Erstellung der Modellfunktionen bedacht wurden.

Diese Arbeit soll als *proof of principle* verstanden werden. Sie soll Methoden zur *modellunabhängigen* Datenanalyse zeigen, da dieser Lösungsansatz einerseits noch nie systematisch untersucht wurde, andererseits aber weitreichende Vorteile zu haben scheint. Diese Arbeit hat gezeigt, dass die modellunabhängige Datenanalyse für USANSPOL-Experimente möglich ist und gewisse Vorteile bietet. Weiters werden erste qualitative Vorhersagen von Messungen über Probeneigenschaften geliefert.

Durch die parallele, gleichzeitige Entwicklung eines komplett alternativen modellunabhängigen Lösungsansatzes in [1] wurde außerdem die Vertrauenswürdigkeit der Methodik gesteigert.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
1.1	Das Neutron . . . . .	7
1.1.1	Neutronen im Magnetfeld . . . . .	8
1.2	SANS, USANS und USANSPOL . . . . .	9
<b>2</b>	<b>Theoretische Grundlagen</b>	<b>12</b>
2.1	Mathematische Grundlagen zur USANS-Datenauswertung . . . . .	12
2.1.1	Die gemessene Neutronenintensität . . . . .	12
2.1.2	Faltungsintegrale . . . . .	14
2.1.3	Entfaltungen . . . . .	14
2.1.4	Größenverteilung der Domänen . . . . .	17
<b>3</b>	<b>Der Messaufbau</b>	<b>20</b>
3.1	Kristalle . . . . .	21
3.2	Magnetische Prismen . . . . .	23
3.3	Proben . . . . .	26
<b>4</b>	<b>Vorbereitung der Messdaten</b>	<b>27</b>
4.1	Grundlagen und Hintergründe . . . . .	27
4.2	Der Spline-Algorithmus . . . . .	28
4.2.1	Diskretisierung der Splines . . . . .	33
4.3	Qualitative Analyse des Spline-Algorithmus . . . . .	35
<b>5</b>	<b>Entfaltung der Messdaten</b>	<b>36</b>
5.1	Grundlagen zu Entfaltungen . . . . .	36
5.2	Trennungsalgorithmus der Rockingfunktion . . . . .	37
5.2.1	Der Trennungsalgorithmus im Detail . . . . .	39
5.3	Entfaltungsalgorithmus . . . . .	42
5.4	Der MyDeconv.m-Algorithmus . . . . .	42

5.4.1	Normieren der Rockingfunktionen . . . . .	44
5.5	Qualitative Analyse der Entfaltungsroutine . . . . .	45
<b>6</b>	<b>Abschätzung des ungestreuten Anteils</b>	<b>50</b>
6.1	Grundlagen zur Abschätzung des ungestreuten Anteils . . . . .	52
6.2	Qualitative Analyse der Prozedur mit Modelldaten . . . . .	53
6.3	Abschätzung der Streuwahrscheinlichkeit $p$ . . . . .	58
6.3.1	Conclusio . . . . .	61
<b>7</b>	<b>Zusammenfassung &amp; Ausblick</b>	<b>62</b>
<b>A</b>	<b>Quellcode</b>	<b>66</b>

# Kapitel 1

## Einleitung

### 1.1 Das Neutron

Das Neutron wurde im Jahre 1920 von Ernest Rutherford vorhergesagt, und 1932 von James Chadwick entdeckt. Die Masse des Neutrons beträgt  $1,674 \cdot 10^{-27} \text{ kg}$ , und es hat einen Spin von  $\frac{1}{2}\hbar$ , es ist somit ein Fermion. Obwohl es eine elektrische Ladung von  $q = 0$  hat, besitzt das Neutron ein magnetisches Moment  $\mu_n = -1,913\mu_B$ . Das magnetische Moment  $\mu_n$  ist ein erster Hinweis auf eine innere Struktur des Neutrons. Dieser Struktur wurde in den 1960er Jahren durch das Quarkmodell ein theoretischer Unterbau verliehen, welcher zwar nicht durch Messung freier Quarks experimentell bestätigt werden kann, da dies durch das Phänomen des Confinements ausgeschlossen ist. Es ließen sich jedoch mit dem Quarkmodell eine Vielzahl an Teilchen vorhersagen und experimentell bestätigen, was als Bestätigung des Modells angesehen werden kann.

Die elektrische Neutralität des Neutrons macht es zu einer idealen Sonde für Festkörper, da man anders als mit Elektronen und Ionen nicht die elektrische Ladungsverteilung innerhalb eines Festkörpers abbildet. Neutronen wechselwirken mit Atomkernen und mit den magnetischen Momenten von Elektronen. Während mit Elektronen oder Ionen nur die obersten Atom-schichten von Festkörpern untersucht werden können, ist durch die elektrische Ladungsfreiheit der Neutronen auch deren mittlere freie Weglänge in Festkörpern viel größer, wodurch auch Volumseigenschaften der Probe untersucht werden können.

Die oben erwähnte mittlere freie Weglänge von Neutronen in Festkörpern liegt in der Größenordnung Millimeter und ist indirekt proportional zum Gesamtwirkungsquerschnitt, welcher sich aus Streuquerschnitt und Absorp-

tionsquerschnitt zusammensetzt. Der genaue Wert hängt einerseits vom Material ab, und andererseits von der Neutronenenergie.

Die Neutronenenergie

$$E = \frac{p^2}{2m_n} \quad (1.1)$$

kann über die Gleichung von de-Broglie [17]

$$\lambda = \frac{h}{p} \quad (1.2)$$

auch in eine Wellenlänge umgerechnet werden:

$$\lambda = \sqrt{\frac{h^2}{2m_n E}}. \quad (1.3)$$

Diese Wellenlänge liegt für thermische Neutronen in der Größenordnung Ångstrom. Da auch der mittlere Atomabstand in Festkörpern in diesem Bereich liegt, ist Reflexion der Neutronen am Einkristall über die Bragg-Bedingung erst möglich (siehe Kapitel 3.1).

### 1.1.1 Neutronen im Magnetfeld

Beim Übergang von Neutronen von einem magnetfeldfreien Raum in ein zeitlich konstantes Magnetfeld, kommt es zur Aufspaltung der Energie des Neutrons, da sich die Spins der Neutronen parallel oder antiparallel zum Magnetfeld ausrichten. Dies ist die sogenannte Zeeman-Aufspaltung [17]:

$$\Delta E = \pm \mu B. \quad (1.4)$$

Diese Zustände nennt man Spin-Up  $|\uparrow\rangle$  und Spin-down  $|\downarrow\rangle$ .

Da das Magnetfeld ein zeitlich konstantes ist, muss auch der Hamiltonoperator des Systems zeitunabhängig sein. Die Änderung der potentiellen Energie aus Gleichung (1.4) muss somit durch eine Änderung der kinetischen Energie und wegen Gleichung (1.3) auch durch eine Änderung der Wellenlänge  $\lambda$  um den gleichen Betrag kompensiert werden [18]. Diese Änderung der Wellenlänge des Neutrons in einem magnetischen Prisma resultiert nun analog zur Brechung von Licht in einem optischen Prisma in eine Winkelaufspaltung der beiden Energiezustände (siehe zum Beispiel [15]).

## 1.2 SANS, USANS und USANSPOL

Wie eingangs erwähnt haben Neutronen herausragende Eigenschaften als Sonden zur Untersuchung von Festkörpern. Durch ihre elektrische Neutralität dringen sie weitaus tiefer in die Probe ein als zum Beispiel Elektronen und Ionen. Im Gegensatz zu Synchrotron- oder Röntgenstrahlung ermöglichen Neutronen auch die Unterscheidung unterschiedlicher Isotope, da die nukleare Streulängen für Neutronen für unterschiedliche Isotope sehr stark schwanken. Weiters lassen sich unter Zuhilfenahme des magnetischen Momentes des Neutrons im Gegensatz zu oben erwähnten Sonden zur Festkörperanalyse auch magnetische Strukturen auflösen.

**SANS** (*Small Angle Neutron Scattering*) ist eine Methode der Festkörperanalyse mittels Neutronen, welche Streuwinkel im Bereich von Grad auflösen kann, wobei eher der von der Neutronenwellenlänge und somit vom Messinstrument unabhängige Streuvektor angegeben wird, welcher mit dem Streuwinkel wie folgt zusammenhängt:

$$|\vec{q}| = \frac{4\pi}{\lambda} \sin\left(\frac{\theta}{2}\right).$$

$|\vec{q}|$  ist hier die Länge des Streuvektors,  $\lambda$  die Wellenlänge der verwendeten Neutronen, und  $\theta$  der Streuwinkel.

Im für SANS typischen experimentellen Aufbau, der *Pinhole-SANS-Kamera* (siehe Abbildung 1.1), wird eine Neutronenwellenlänge mittels Geschwindigkeitsselektors ( $\frac{\Delta\lambda}{\lambda} \sim 10\%$ ) ausgewählt, durch ein Blendensystem kollimiert und an der Probe gestreut. Die Detektion der Neutronen geschieht mittels eines positionssensitiven Detektors in bis zu 30 m Entfernung. Die so erhaltenen Informationen über die Probe lassen Strukturen wie zum Beispiel Ausscheidungen, Versetzungen, Poren, Texturen in einem Größenbereich von 1 nm bis einige 100 nm auflösen.

**USANS** (*Ultra Small Angle Neutron Scattering*) ist die Weiterentwicklung von SANS und verbessert die Auflösung weiter: hier sind Winkelauflösungen im Bereich von Bogensekunden möglich, wodurch Texturen im Bereich von 200 nm - in etwa 20  $\mu\text{m}$  aufgelöst werden können. Um dies zu realisieren werden im experimentellen Setup (siehe Abbildung 1.2) als Monochromator und Analysator spezielle Einkristalle verwendet (siehe Kapitel 3.1). Um eine Messkurve aufzunehmen wird der Analysatorkristall schrittweise um einen Winkel verdreht und der bei dem jeweiligen Winkel durch die Bragg-Bedingung (Gleichung (3.1)) selektierte Streuvektor  $\vec{q}$  aufgenommen. Dieses schrittweise Verdrehen des Analysatorkristalls geschieht mittels Piezo-

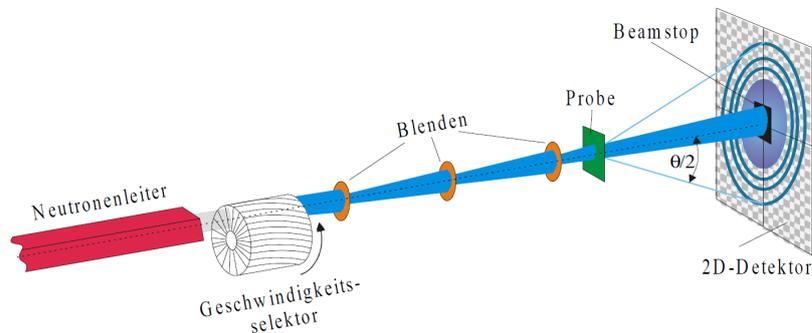


Abbildung 1.1: Der Typische experimentelle Aufbau eines SANS-Experimentes: die Pinhole-Kamera. [3]

versteller. In den 1990er-Jahren erfuhr die USANS-Technik durch Arbeiten von Agamalian [12] einen Aufschwung: er durchschnitt die lange Seite der Einkristalle und fügte in den Schnitt eine Cadmiumschicht ein (siehe Kapitel 3.1). Dies verkleinerte den Untergrund der Messungen um etwa eine Größenordnung, da Neutronenreflexion innerhalb des Kristalls nun unterbunden wird.

**USANSPOL** (*Ultra Small Angle Neutron Scattering with POLarized neutrons*) ist eine Weiterentwicklung von USANS, die zwar keine bessere Winkelauflösung ermöglicht, dafür aber das magnetische Moment des Neutrons ausnützt, um so auch magnetische Streuung an der Probe zu untersuchen, womit es möglich wird magnetische Domänen im Festkörper zu erforschen. Hierfür werden die zwei unterschiedlichen magnetischen Spinzustände  $|\uparrow\rangle$  und  $|\downarrow\rangle$  des Neutrons durch ein Magnetfeld aufgespalten. Hierfür wird in der USANS-Messaufbau durch magnetische Prismen im Strahlengang erweitert (siehe Abbildung 3.1).

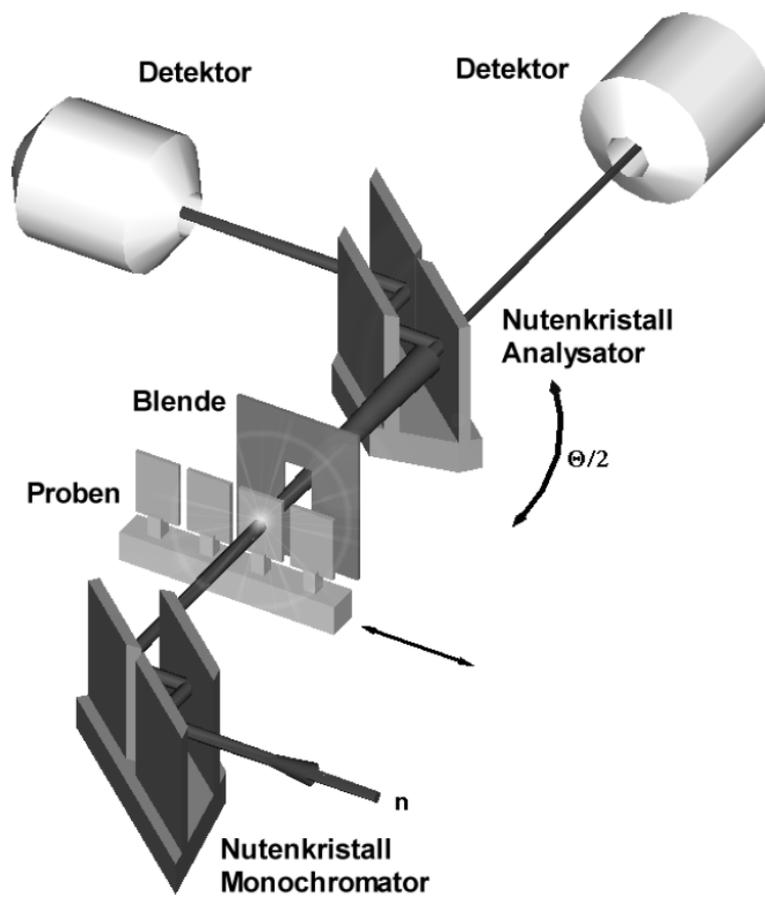


Abbildung 1.2: Der typische experimentelle Aufbau eines USANS-Experimentes: die Bonse-Hart-Kamera [3]

# Kapitel 2

## Theoretische Grundlagen

### 2.1 Mathematische Grundlagen zur USANS-Datenauswertung

Die Auswertung von Daten der USANS und USANSPOL-Experimente führt zu einigen Schwierigkeiten, die gelöst werden möchten.

Einerseits hat man - wie bei jeder Messung - die Schwierigkeit der Messungsgenauigkeit. Dieses Problem ist eigentlich nur durch eine ausreichende statistische Genauigkeit zu lösen, was darin resultiert, die Zählraten möglichst hoch zu halten. Um dies zu erreichen werden die Experimente unter anderem am ILL in Grenoble durchgeführt, wo derzeit der Forschungsreaktor mit dem höchsten Neutronenfluss steht. Es ist natürlich auch von notwendig die Neutronenverluste innerhalb der Messapparatur minimal zu halten. Wie man in Abbildung 3.4 erkennen kann, ist es stets von Vorteil, Kristalle zu verwenden, welche möglichst viele Reflexionen ermöglichen, da dies die Schultern der Reflexionskurven minimiert. Andererseits sind mehr als Dreifachreflexionen im Allgemeinen weit schwieriger zu justieren und manchmal auch nicht eindeutig, da zum Beispiel Mischungen aus Drei- und Fünffachstreuungen auftreten können. Wie man in Abbildung 3.5 erkennen kann, liegt ein guter Mittelweg zwischen beiden Extrema bei 3 Reflexionen pro Kristall.

Andererseits gibt es USANSPOL-spezifische Problemstellungen bei der Datenauswertung, welche in folgenden Kapiteln behandelt werden.

#### 2.1.1 Die gemessene Neutronenintensität

Bei Neutronenstreuexperimenten wie USANS und USANSPOL werden Neutronen an einem Target gestreut und die Winkelverteilung der gestreuten

Neutronen gemessen. Diese Neutronenverteilung ist prinzipiell eine zweidimensionale, da die Neutronen in x- und y-Richtung gestreut werden und lässt sich mathematisch wie folgt ausdrücken [19]:

$$I(\mathbf{Q}) \propto (N \langle b(\mathbf{Q}) \rangle V)^2 S(\mathbf{Q}), \quad (2.1)$$

mit  $N$  als Anzahl der Atome pro Einheitsvolumen,  $S(\mathbf{Q})$  als Streufunktion und

$$\langle b(\mathbf{Q}) \rangle = b_c + b_m \mathbf{P} \hat{\mathbf{M}}_{\perp}(\mathbf{Q}) \quad (2.2)$$

als mittlere Neutronenstrelänge.  $b_c$  ist der kohärente Anteil der Streuung, welche durch Kernstreuung entsteht,  $b_m$  ist der magnetische Streuanteil, welcher seinen Ursprung in der Wechselwirkung der Neutronen mit den Spins der Elektronen hat,  $\mathbf{P}$  ist die Polarisationsrichtung der Neutronen und  $\hat{\mathbf{M}}_{\perp}(\mathbf{Q})$  ist ein Einheitsvektor, welche in die Richtung der mittleren Magnetisierung der Probe zeigt.

Der Streuvektor  $\mathbf{Q}$  dieser Neutronen wird so definiert, dass die  $Q_x$ -Komponente in die Achse des ungestreuten Neutronenstrahls gelegt wird und die  $Q_y$ - und  $Q_z$ -Komponenten normal dazu:  $Q_y$  horizontal und  $Q_z$  vertikal. Im Vergleich zu den anderen Komponenten, jedoch ist die  $Q_x$ -Komponente vernachlässigbar klein und wird somit stets 0 gesetzt.

Diese zweidimensionale Neutronenverteilung ist jedoch nicht detektierbar, da die extrem hohe Auflösung der USANSPOL-Messapartur nur in einer Dimension gegeben ist. Diese Richtung wird von der Ausrichtung des Kristallgitters vorgegeben und ist im gegebenen Aufbau die  $Q_y$ -Achse. In  $Q_z$ -Richtung gestreute Neutronen werden auch detektiert, jedoch nicht winkelabhängig, sondern es werden alle Neutronen mit einer spezifischen  $Q_y$  und einer beliebigen  $Q_z$ -Komponente aufsummiert. Diese Summation funktioniert also so, als ob vor dem Detektor eine Schlitzblende in y-Richtung montiert wäre, daher der Name *Slit Height Smearing*.

Mathematisch lässt sich diese Verschmierung als Integration der  $Q_z$  Komponente von Gleichung (2.1) auffassen:

$$I(Q_y) \propto \int dQ_z (N \langle b(Q_y, Q_z) \rangle V)^2 S(Q_y, Q_z) \quad (2.3)$$

Die Intensität aus Gleichung (2.1) und das dazugehörige Schlitzverschmierung wurde in [5] durch mathematische Modellberechnungen bestimmt und ist in Abbildung 2.1 zu sehen. Abbildung 2.1.a zeigt die Intensitätsverteilung von in  $+z$ -Richtung polarisierten Neutronen, welche an einem auch in  $+z$ -Richtung magnetisiertem Target gestreut werden. In Abbildung 2.1.b und 2.1.c sind es auch  $|\uparrow\rangle$ -Neutronen, die jedoch an Targets gestreut werden,

welche in  $45^\circ$  respektive  $90^\circ$  zur x-Achse magnetisiert sind.

Wie eingangs erwähnt, ist die Messapparatur jedoch nur in der Richtung der y-Achse empfindlich, während die Intensitäten in z-Richtung aufsummiert werden. Dieses Slit Height Smearing wurde durchgeführt und in Abbildung 2.1.d für die drei Winkel  $0^\circ$  (rot),  $45^\circ$  (grün) und  $90^\circ$  (blau) geplottet. Dies sind die Kurven von Interesse, jedoch nicht die Kurven, welche gemessen werden: die Kurven aus 2.1.d müssen noch mit der Rockingkurve gefaltet werden, da das Messgerät diese mathematische Kurve verschmiert. Ergebnis dieser Faltung ist in Abbildung 2.1.e aufgetragen. Diese Kurvenschar entspricht der zu erwartenden Messkurve für  $0^\circ$  (rot),  $45^\circ$  (grün) und  $90^\circ$  (blau).

## 2.1.2 Faltungsintegrale

Im letzten Abschnitt wurde das Problem der Verschmierung der Messkurve durch die Gerätekurve schon angedeutet. Grund für diese Verschmierung liegt darin, dass die Messapparatur (ohne Probe) einen kollimierten Neutronenstrahl am Eingang der Apparatur nicht auf einen deltaförmigen Peak am Ausgang abbildet, sondern diesen verschmiert, wie es zum Beispiel in Abbildung 2.2 ersichtlich ist. Diese ohne Probe aufgenommene Messkurve ist für die Messapparatur eine charakteristische Kurve - sie wird Rockingkurve oder Leerkurve  $R(\theta)$  genannt - und ist ein Qualitätsmerkmal des Messgeräts: desto schmaler und spitzer die Kurve, desto besser das Messgerät.

Diese Rockingkurve ist also eine geräteabhängige Kurve, welche die tatsächliche physikalische Kurve  $P(\theta)$  gewissermaßen „überschattet“ oder verschmiert. Diese Verschmierung resultiert in einem Faltungsintegral:

$$I_{\uparrow\downarrow}(\theta) = R_{\uparrow\downarrow} * P_{\uparrow\downarrow} = \int d\theta' R_{\uparrow\downarrow}(\theta' - \theta) P_{\uparrow\downarrow}(\theta'), \quad (2.4)$$

und das Ergebnis dieses Integrals  $I_{\uparrow\downarrow}(\theta)$  ist die real gemessene Intensitätsverteilung  $I_{\uparrow\downarrow}(\theta)$ .

Durch Gleichung (2.4) erkennt man außerdem, dass die ideale Leerkurve eines Messgerätes die Dirac'sche Deltadistribution

$$R(\theta) = \delta(\theta' - \theta)$$

ist.

## 2.1.3 Entfaltungen

Natürlich möchte man aber absolute Messdaten analysieren, also Messdaten, welche von der jeweils verwendeten Messapparatur unabhängig sind und somit die physikalischen Vorgänge in der Probe abbilden. Dazu muss nun die

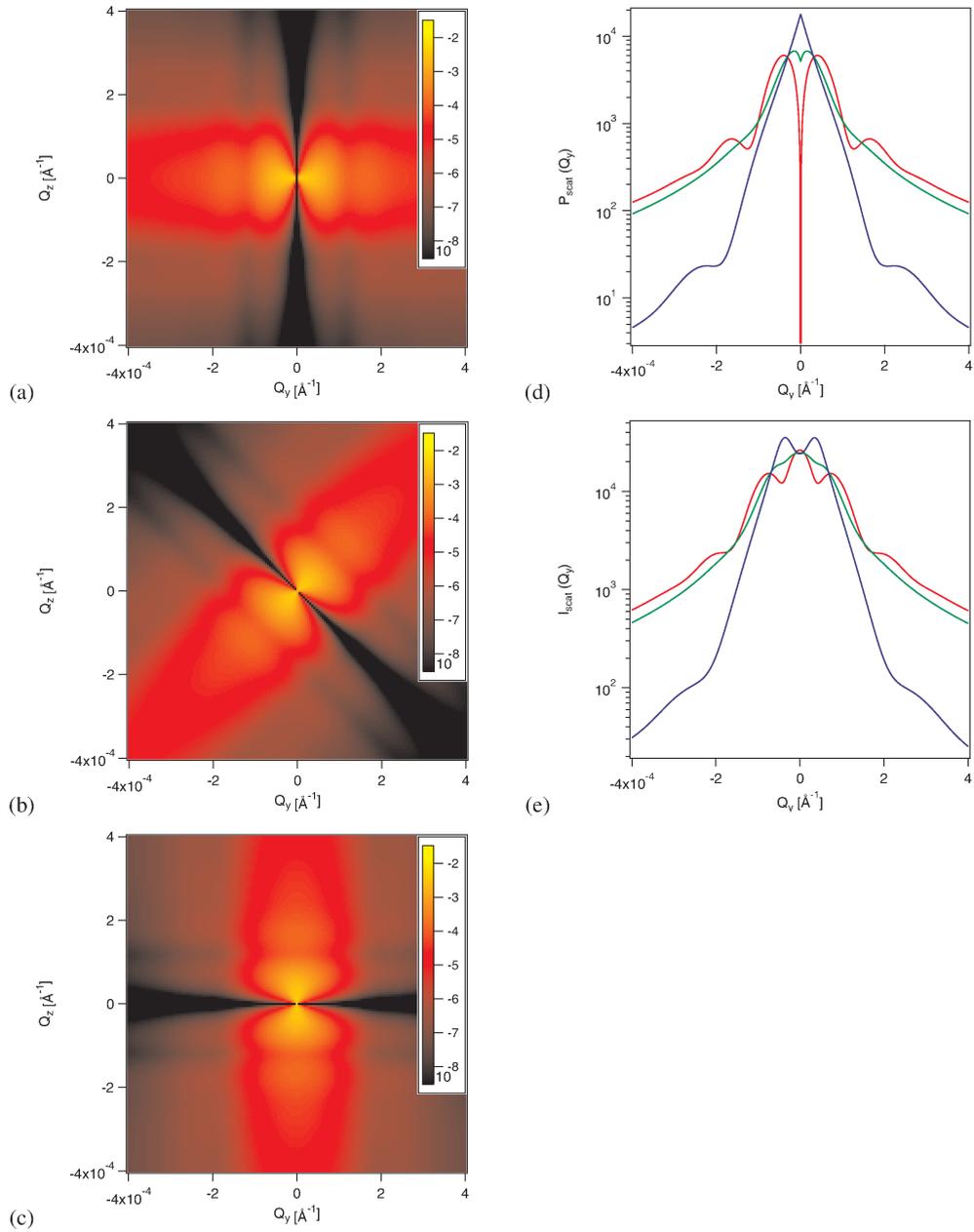


Abbildung 2.1: Während die Abbildungen (a-c) zweidimensionale Intensitätsverteilungen einer einzigen Spinkomponente darstellen, zeigt Bild (d) das dazupassende *Slit height smearing*: (a) rot, (b) grün, (c) blau. In Abbildung (e) ist die Verschmierung der physikalischen Kurven aus (d) mit einer Rockingfunktion zu erkennen. Für Details siehe Kapitel 2.1.1. Abbildung aus [5].

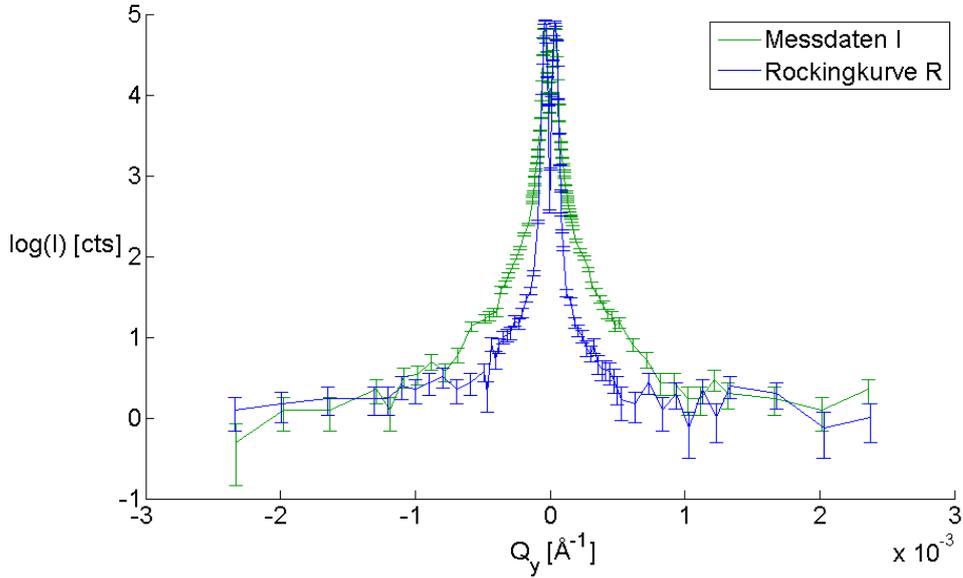


Abbildung 2.2: Rockingkurve und Messkurve

Faltung in Gleichung (2.4) gewissermaßen rückgängig gemacht werden, um  $P_{\uparrow\downarrow}(\theta)$  zu erhalten. Dies erfolgt mit der Umkehroperation der Faltung, der Entfaltung, welche aber wesentlich aufwändiger anzuwenden ist und auch weit empfindlicher auf Fehler in den Daten ist. Außerdem muss die Entfaltung keine eindeutige Lösung haben und auch nicht positiv definite Funktionen sind als Lösungen möglich, da es sich aber um Kurven handelt, die Messintensitäten beschreiben, sind diese vom physikalischen Standpunkt her auch ausgeschlossen.

Dieses Verhalten lässt sich wie folgt erklären: durch das Faltungstheorem lässt sich die Faltung in Gleichung (2.4) durch Fouriertransformation der einzelnen Komponenten umwandeln in eine Multiplikation:

$$\mathcal{F}(f * g) = \sqrt{2\pi} \mathcal{F}(f) \cdot \mathcal{F}(g). \quad (2.5)$$

Dadurch kann man die Faltung in Gleichung (2.4) anschreiben als:

$$\hat{I}_{\uparrow\downarrow}(\theta) = \hat{R}(\theta)_{\uparrow\downarrow} \cdot \hat{P}(\theta)_{\uparrow\downarrow}, \quad (2.6)$$

wobei die jeweiligen Fouriertransformierten durch ein Dach gekennzeichnet wurden.

Nun ist es jedoch so, dass jede Messung mit einem gewissen Fehler, respektive Rauschen behaftet ist, die Faltung aus Gleichung (2.4) also viel mehr aussieht

wie

$$I_{\uparrow\downarrow}(\theta) = R(\theta)_{\uparrow\downarrow} \cdot P(\theta)_{\uparrow\downarrow} + F(\theta), \quad (2.7)$$

mit einem nicht weiter bekannten Fehler  $F(\theta)$ .

Wie oben bereits erwähnt, ist  $P(\theta)$  die gesuchte Funktion, welche man durch Entfaltung, respektive durch Fouriertransformation finden möchte. Dazu muss man also die Komponenten von Gleichung (2.7) fouriertransformieren und  $P(\theta)$  explizit ausdrücken:

$$P(\hat{\theta})_{\uparrow\downarrow} = \frac{\hat{I}_{\uparrow\downarrow}(\theta)}{\hat{R}(\theta)_{\uparrow\downarrow}} - \frac{\hat{F}(\theta)}{\hat{R}(\theta)_{\uparrow\downarrow}}. \quad (2.8)$$

Nun ist es möglich die Ursache zu erkennen, warum Entfaltungen meist aufwendiger sind als Faltungen: einerseits ist es möglich, dass  $\hat{R}(\theta)_{\uparrow\downarrow}$  0 werden kann, andererseits geht  $\hat{R}(\theta)_{\uparrow\downarrow}$  für hohe Frequenzen gegen 0, gerade dort hat der Rauschanteil  $\hat{F}(\theta)$  aber Werte ungleich 0, welche somit mit  $\frac{1}{\hat{R}(\theta)_{\uparrow\downarrow}}$  verstärkt werden.

#### 2.1.4 Größenverteilung der Domänen

All diese Überlegungen gelten für eine ideale Probe mit Streuzentren der exakt gleichen Ausdehnung. Dies ist natürlich eine unnatürliche Annahme, da die Streuzentren - für magnetische Streuung sind dies die magnetischen Domänen im Festkörper - nicht alle genau gleich groß sind. Um die Modellrechnung aus Integral (2.1) noch weiter an die Realität in der Probe heranzuführen, müssen die Streuer im Modell also einer bestimmten Größenverteilung folgen. Deshalb wird Gleichung (2.1) noch mit einer Verteilungsfunktion  $P(\mathbf{a})$  versehen, welche die Größenverteilung der Streuer beschreibt. Es muss über die Streuteile aller möglichen Größen der Streuer integriert werden. So kommt man zu dem Integral

$$I(\mathbf{Q}) \propto \langle |\rho(\mathbf{Q})|^2 \rangle = \int (N \langle b(\mathbf{Q}) \rangle V(\mathbf{a}))^2 S(\mathbf{Q}, \mathbf{a}) \cdot P(\mathbf{a}) d^n a, \quad (2.9)$$

mit  $\mathbf{a}$  als Satz von Parametern.

Dies geschieht in einer ersten Näherung durch die Annahme, dass die Streuer quaderförmig sind und das Produkt aus deren Länge in Flugrichtung (also in x-Richtung) und der Komponente des Streuvektors in diese Richtung gegen 0 geht. Außerdem nimmt man an, dass Länge  $a$  und Breite  $b$  in z- respektive in y-Richtung normalverteilt um  $a$  und  $b$  mit gewissen Varianzen  $\sigma_a$  und  $\sigma_b$  sind. Aus diesen Annahmen folgt für  $P(\mathbf{a})$  eine Normalverteilung in 2 Dimensionen

mit den Parametern  $a$  und  $b$ :

$$P(\mathbf{a}) = \frac{1}{\sqrt{2\pi\sigma_a^2}} \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{1}{2}\left(\frac{a-a_0}{\sigma_a}\right)^2\right) \exp\left(-\frac{1}{2}\left(\frac{b-b_0}{\sigma_b}\right)^2\right). \quad (2.10)$$

$P(\mathbf{a})$  in Gleichung (2.1) eingesetzt, umformen wie es in Tobias Rechbergers Diplomarbeit ([1]) en détail erklärt wird und integrieren über alle möglichen Größen für  $a$  und  $b$  führt auf

$$\begin{aligned} I(\mathbf{Q}) &= \int_0^\infty \int_0^\infty n^2 \cdot (b_c + b_m (P_z Q_y - P_y Q_z)) \cdot \left(\frac{1}{Q_y^2 + Q_z^2} (Q_y M_z - Q_z M_y)\right)^2 \\ &\quad \cdot \operatorname{sinc}\left(\frac{b_0 Q_y}{2}\right) \operatorname{sinc}\left(\frac{a_0 Q_z}{2}\right) \\ &\quad \cdot \frac{1}{\sqrt{2\pi\sigma_a^2}} \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{1}{2}\left(\frac{a-a_0}{\sigma_a}\right)^2\right) \exp\left(-\frac{1}{2}\left(\frac{b-b_0}{\sigma_b}\right)^2\right) da db. \end{aligned} \quad (2.11)$$

Führt man nun auch noch das *slit height smearing* (siehe Kapitel 2.1.1) durch, erweitert sich obiges Integral um eine Integration in  $Q_z$

$$\begin{aligned} I(Q_y) &= \int_{-\infty}^\infty \int_0^\infty \int_0^\infty n^2 (b_c + b_m (P_z Q_y - P_y Q_z)) \left(\frac{1}{Q_y^2 + Q_z^2} (Q_y M_z - Q_z M_y)\right)^2 \\ &\quad \cdot \operatorname{sinc}\left(\frac{b_0 Q_y}{2}\right) \operatorname{sinc}\left(\frac{a_0 Q_z}{2}\right) \\ &\quad \cdot \frac{1}{\sqrt{2\pi\sigma_a^2}} \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{1}{2}\left(\frac{a-a_0}{\sigma_a}\right)^2\right) \exp\left(-\frac{1}{2}\left(\frac{b-b_0}{\sigma_b}\right)^2\right) da db Q_z. \end{aligned} \quad (2.12)$$

Die Intensitätsverteilung in Gleichung (2.11) stellt ein theoretisches Modell für die Streuung der Neutronen in  $y$ - und  $z$ -Richtung dar. Diese zweidimensionale Verteilung ergibt Plots, wie sie zum Beispiel in Abbildung 2.1 dargestellt sind: man erkennt deutlich wie die Vorzugsrichtung der Streuung der polarisierten Neutronen vom Winkel zwischen Spineinstellung der Neutronen  $\mathbf{P}$  und der Richtung der Magnetisierung der Probe  $\mathbf{M}$  abhängt.

Leider kann USANSPOL jedoch diese zweidimensionalen Neutronenintensitäten nicht auflösen, da die hohe Messempfindlichkeit nur in einer Richtung gegeben ist - in vorliegendem Fall ist dies die  $Q_y$ -Richtung. Aus diesem Grunde muss - wie in Kapitel 2.1.1 beschrieben - noch das *slit height smearing* durchgeführt werden, also eine Integration über  $Q_z$ . Dies führt nun auf Gleichung 2.12 und somit zu prinzipiell detektierbaren Intensitätsverteilungen wie sie zum Beispiel in Abbildung 2.1.d abgebildet sind.

Diese beiden Integrale und deren grafische Darstellungen bieten einem nun die Möglichkeit, die gemessenen Intensitätskurven mit den theoretisch vorhergesagten Intensitätskurven zu vergleichen und daraus gewisse Parameter zu bestimmen. So ist es zum Beispiel möglich aus dem Vergleich der beiden Kurven die mittlere Größe  $a_0$  und  $b_0$  der Streuzentren zu bestimmen, oder deren Standardabweichung  $\sigma_a$  und  $\sigma_b$ . Weiters ist es natürlich auch von Interesse herauszufinden, ob die Annahme, dass die Streuzentren quaderförmig und deren Abmessungen in x- und y-Richtung gaußverteilt sind, eine hinreichend gute Näherung ist, oder ob vielleicht nicht eine andere Näherung die Physik in der Probe besser abzubilden vermag.

All diese Fragen sind von sehr spannender Natur, jedoch kann im Zuge dieser Diplomarbeit leider nicht weiter darauf eingegangen werden, da noch spannendere Fragen zu beantworten versucht werden.

# Kapitel 3

## Der Messaufbau

IN Abbildung 3.1 ist der Messaufbau eines typischen USANSPOL-Experimentes schematisch dargestellt. Mit diesem klassischen Bonse-Hart-Setup wird einerseits am Institut Laue Langevin (ILL) in Grenoble gearbeitet, als auch am Atominstitut der TU Wien. Da es am ILL einen weit höheren Neutronenfluss gibt, werden dort die Messungen an den Proben durchgeführt, um statistisch signifikantere Messergebnisse zu erlangen. Dies liegt in der Tatsache begründet, dass bei den Messungen von Neutronen die Zählrate immer durch das  $\sqrt{n}$ -Gesetz der Poisson-Statistik einer gewissen statistischen Schwankung unterliegt. Wie bei jedem simplen Abzählen von Ereignissen liegt nämlich der wahre Messwert mit einer Wahrscheinlichkeit von 68,3% innerhalb des Intervalls

$$n \pm \sqrt{n}.$$

Möchte man also eine Messgenauigkeit von  $\pm 1\%$  erzielen, muss man im Schnitt 10 000 Neutronen messen!

Zusätzlich spielt noch das Verhältnis zwischen Peakintensität und Untergrund eine bedeutende Rolle. Dieses *Untergrundverhältnis* hat in Wien eine viel geringe Dynamik, es liegen nur in etwa 2 Größenordnungen zwischen Peakintensität und Untergrund, während es am ILL in etwa 5 Größenordnungen sind. Dadurch lassen sich gewisse Streuphänomene in Wien nicht auflösen sondern gehen im Untergrund unter.

Der in Abbildung 3.1 gezeigte Messaufbau besteht aus Monochromator- und Analysatorkristallen, magnetischen Prismen, der Probe und 3 orthogonal zueinander stehenden Helmholtzspulenpaaren. Diese Komponenten möchten im folgenden kurz beschrieben werden, um einen Überblick über den experimentellen Messaufbau zu verschaffen. Für Details wird aber auf einschlägige Literatur verwiesen (zum Beispiel [21] [12] [16]).

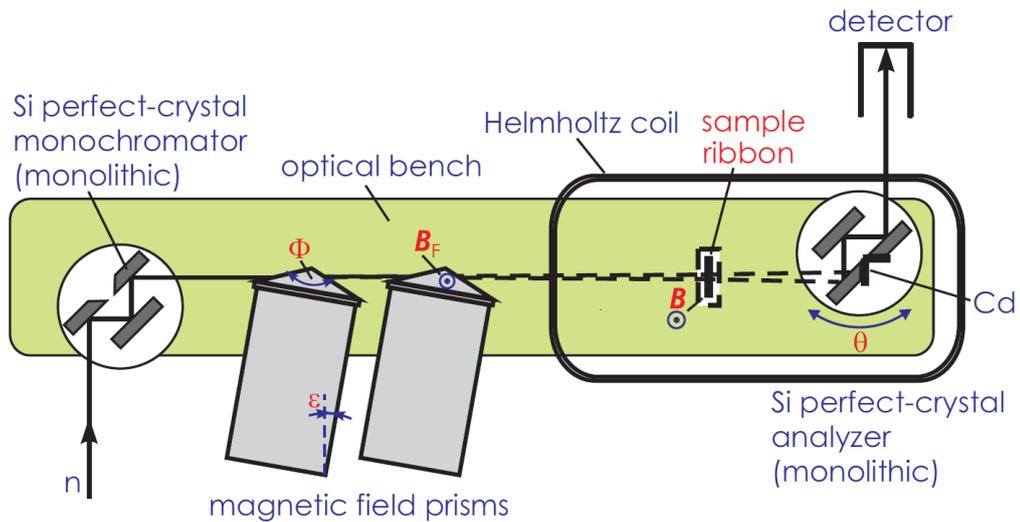


Abbildung 3.1: Schema einer Bonse-Hart-Kamera zur Ultrakleinwinkelmes-  
sung. [4]

### 3.1 Kristalle

Als Monochromator- und Analysatorkristalle fungieren zwei Siliziumeinkri-  
stalle. Durch die Bragg-Bedingung [17]

$$n\lambda = 2d \sin(\theta) \quad (3.1)$$

( $n$  ist die Beugungsordnung,  $\lambda$  die Wellenlänge der für das Experiment  
verwendeten Neutronen,  $d$  ist der Gitterebenenabstand der Atome im Kristall  
und  $\theta$  der Braggwinkel) wird im Monochromator durch gezieltes Ausrich-  
ten des Kristalls eine Neutronenwellenlänge ausgewählt, welche die Bragg-  
Bedingung erfüllt.

Mit dem Analysator am anderen Ende des Aufbaus ist es möglich extrem  
kleine Änderungen des Einfallswinkels über die Bragg-Bedingung (etwa im  
Bereich  $\mu rad$ ) zu detektieren: der Perfektkristall wird mittels Piezoverstel-  
ler über den zu untersuchenden Winkelbereich gedreht. Da die Wellenlänge  
der eintreffenden Neutronen schon vom Monochromator definiert wurde, ist  
der Analysator durch Gleichung (3.1) nun empfindlich auf den Winkel  $\theta$ , da  
nur Neutronen mit dem Einfallswinkel  $\theta$  die Braggbedingung erfüllen, den  
Strahlgang des Kristalls passieren und somit im Detektor detektiert werden  
können.

Wie in [13] nachzulesen ist, hat die Reflektivitätskurve der Kristalle im Ideal-  
fall die Form einer Rechtecksfunktion. Dies ergibt sich daher, dass Neutronen

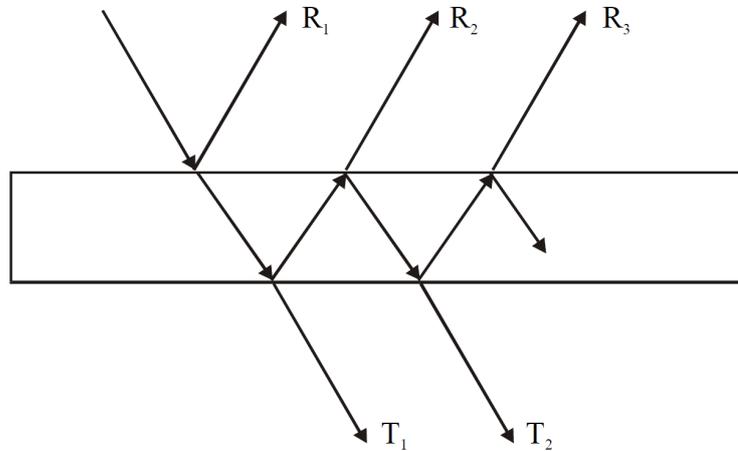


Abbildung 3.2: Erklärung zu Mehrfachreflexionen innerhalb des Kristalls. Diese sind wesentlich für die Schultern der Reflektivitätskurve der Kristalle verantwortlich. [11]

nicht nur (wie durch die Bragg-Bedingung (3.1) suggeriert) bei einem einzigen Winkel reflektiert werden, sondern in einem kleinen Winkelbereich, dem sogenannten Darwinplateau vollständig und in den angrenzenden Winkelbereichen mit einer gewissen noch signifikanten Wahrscheinlichkeit. Um diesen Effekt zu minimieren werden die Neutronen Mehrfachreflexionen an einem Kristall unterworfen, die die Gesamtwahrscheinlichkeit für Reflexion in den Schulterbereichen reduzieren.

Die reale Reflektivitätskurve eines Einkristalls hat somit die Form einer Rechteckfunktion mit abfallenden Schultern, so wie in Abbildung 3.4 dargestellt. Diese werden durch Mehrfachreflexion der Neutronen innerhalb der Kristalle hervorgerufen. Um diesen Effekt zu unterdrücken und um somit die Rockingkurve zu optimieren wurde die ursprünglich längere Seite der Kristalle in der Mitte durchgeschnitten um Mehrfachreflexion zu vermeiden. Zusätzlich wurde in den so entstandenen Schlitzen eine Schicht Cadmium eingefügt (siehe Abbildung 3.3), um eine Propagation der Neutronen innerhalb der Kristallplatten zu verhindern, was die Intensität der Flanken weiter verringerte.

Um die Auswirkung von Monochromator und Analysator auf die Rockingkurve des gesamten Setups abschätzen zu können, ist es notwendig die Faltung der Reflektivitätskurven der beiden Kristalle zu berechnen. Dies ist eine Faltung von zwei Rechteckfunktionen mit Schultern. Zwei Rechteckfunktio-

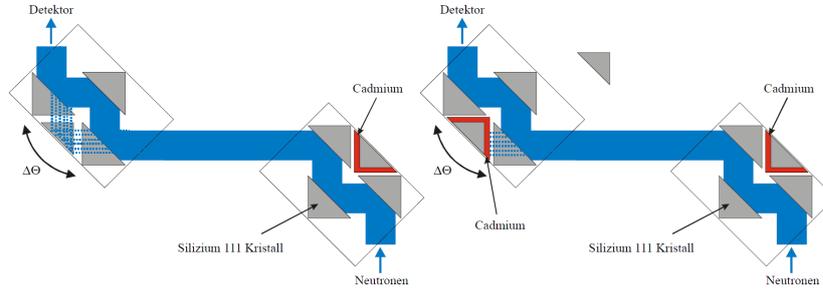


Abbildung 3.3: Durch Aufschneiden der langen Seite der Kristalle und einfügen einer Cd-Schicht werden Mehrfachreflexionen innerhalb des Kristalls weitgehend unterdrückt. [11]

nen miteinander gefaltet ergeben eine Dreiecksfunktion:

$$\text{rect}(\theta) * \text{rect}(\theta) = \int d\theta' \text{rect}(\theta' - \theta) \text{rect}(\theta) = \Lambda(\theta - \theta_i).$$

Das Ergebnis muss somit eine Dreiecksfunktion mit Schultern sein (siehe Abb. 3.5). Dieses Wissen über die Reflektivitätskurve der beiden Kristalle ist notwendig für etwaige Modellrechnungen für Rockingfits, wie es zum Beispiel in Formel (4.1). In diesem Modell entspricht der erste Summand  $A_t \Lambda(\frac{\theta - \theta_i}{\tau})$  genau der oben besprochenen Dreiecksfunktion.

## 3.2 Magnetische Prismen

Abbildung 3.6 zeigt die zur magnetischen Separation der Spinzustände der Neutronen in den Strahlengang eingeführten magnetischen Prismen. Um den aufspaltenden Effekt des Magnetfeldes zu verstärken, werden immer zwei Prismen in den Strahl gestellt. Diese erzeugen in ihrem 1 cm hohen Luftspalt ein weitgehend homogenes Magnetfeld von 0.9 T, nur an den Rändern ist das Magnetfeld inhomogen. Diese Randeffekte können jedoch vernachlässigt werden, wenn die beiden Magnete weit genug voneinander entfernt im Strahlengang stehen, sodass der Feldüberlapp möglichst gering ist. Weil dieser Überlapp der Felder nicht zu vermeiden ist, spalten 2 magnetische Prismen die Spinzustände nicht doppelt so weit auf wie ein einzelnes, die Aufspaltung fällt geringer aus [21].

Analog zu einem doppelbrechenden Kalkspatkristall, welcher für die beiden Polarisationsrichtungen der einfallenden Lichtwelle einen unterschiedlichen Brechungsindex  $n$  aufweist, werden im Prisma die beiden Spinzustände der Neutronen durch ihre unterschiedliche Wechselwirkung mit dem magnetischen Feld getrennt, indem der Wellenvektor  $k$  der  $\uparrow$  /  $\downarrow$ -Neutronen um

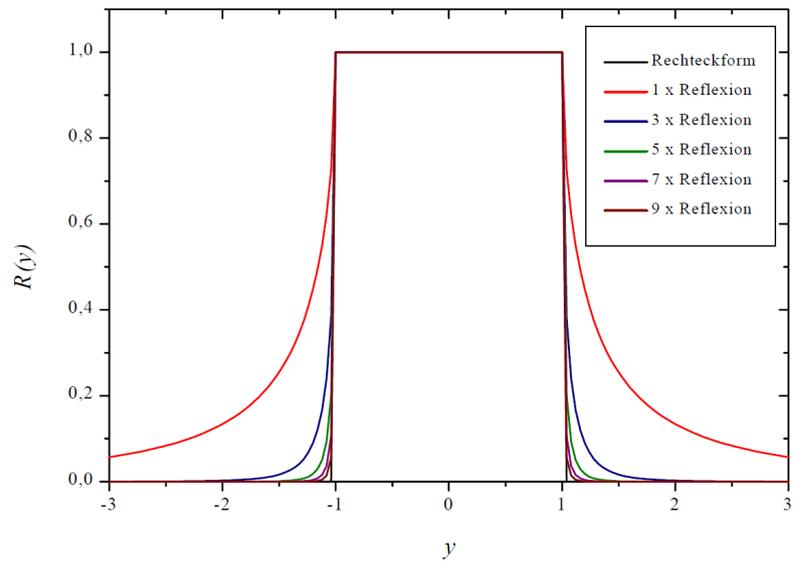


Abbildung 3.4: Rockingkurve eines Perfekt-Kristalls verglichen mit einer Rechtecksfunktion: man erkennt, dass Mehrfachreflexionen dieser bedeutend näher kommen. [11]

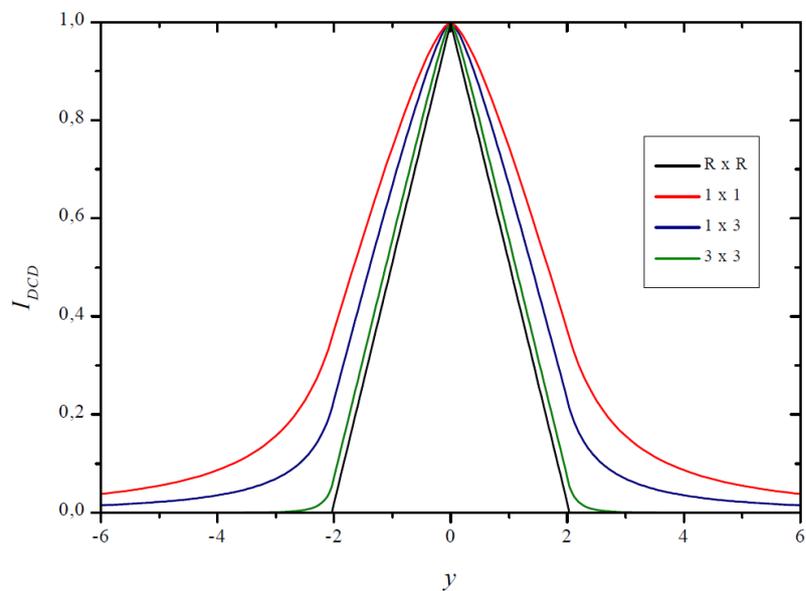


Abbildung 3.5: Die mit sich selbst gefaltete Rockingkurve aus Abb. 3.4. Wieder erkennt man, dass Mehrfachreflexionen der idealisierten Dreiecksfunktion mehr entsprechen. [11]

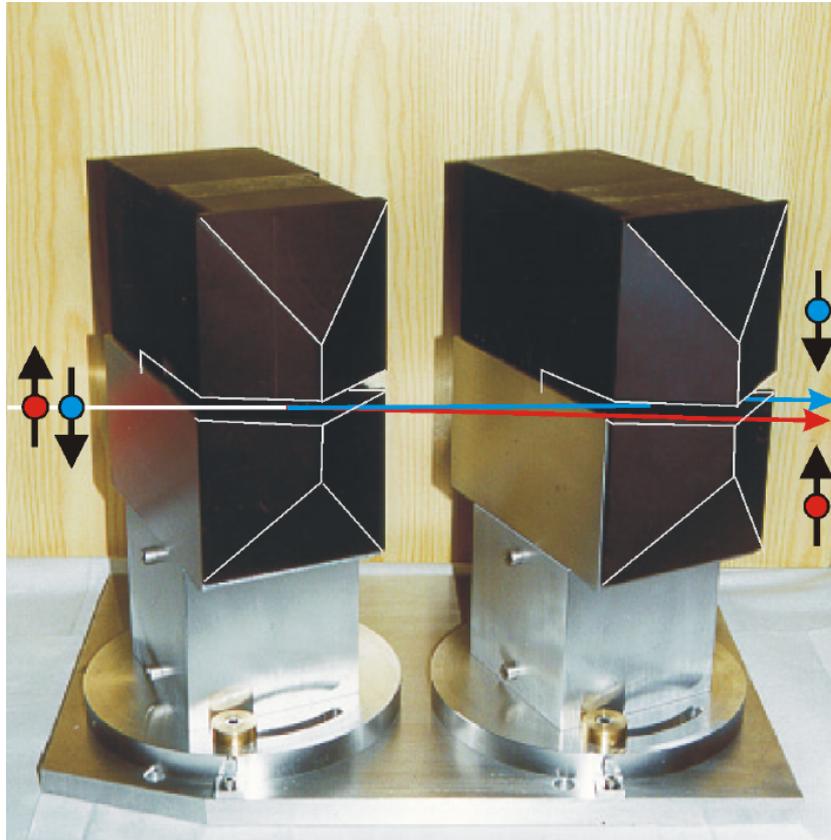


Abbildung 3.6: Magnetische Prismen welche zur Spinaufspaltung genutzt werden. [4]

$\Delta k \propto \pm \mu B$  aufgespalten wird. [14]

Dieser Umstand kann analog zum doppelbrechenden Kalkspat als unterschiedlicher Brechungsindex des Magnetfeldes für  $\uparrow$ - respektive  $\downarrow$ -Neutronen beschrieben werden (siehe auch Kapitel 1.1.1):

$$n_{\uparrow\downarrow} = \frac{k}{k_0} \simeq 1 \pm \frac{m\mu B}{\hbar^2 k_0^2}.$$

### 3.3 Proben

Um reproduzierbare Messergebnisse zu erzeugen ist es unerlässlich, dass die zu untersuchenden Proben in einer streng definierten Umgebung gemessen werden. Aus diesem Grunde wurde unter anderem von T. Rechberger eine Probenhalterung weiterentwickelt, um dies zu gewährleisten [24]. Diese Probenumgebung besteht aus 3 orthogonal stehenden Helmholtz-Spulenpaaren, mit denen man einerseits unerwünschte externe Felder eliminieren kann, um somit einen magnetfeldfreien Raum für die Probe zu erzeugen. Alternativ kann man aber auch ein Magnetfeld  $\vec{B}$  in allen 3 Raumdimensionen beliebig definieren, welches für die äußere Magnetisierung  $\vec{M}$  der Probe verantwortlich ist. Außerdem wird die Probe in eine spezielle Halterung eingespannt, welche es ermöglicht sie unter Zugkraft zu setzen. Somit kann untersucht werden, wie sich die Streukurve unter Zugkraft der Probe verhält [7].

Die Proben selbst sind aus dem Probenset der " $\alpha$ -Proben", und sind metallische Bänder aus  $Fe_{78}Mo_2Bo_{20}$ , deren Streukurve man messen möchte, um auf magnetische Domänen rückschließen zu können und wie sich diese unter äußere Magnetisierung und/oder mechanische Zugkräfte verhalten.

Für geplottete Messdaten in dieser Arbeiten wurde stets der Messdatensatz „A0B0S0\_22Sep1142\_Qcvt.dat“ für  $I(Q_y)$ , und der Messdatensatz „prock16Sep0404\_Qcvt.dat“ für die Rockingkurve  $R(Q_y)$  verwendet, welche beide am S18 in Grenoble erstellt wurden. Nur in Kapitel 6 werden auch andere Messdaten und daher auch andere Proben untersucht, darauf wird an der jeweiligen Stelle jedoch hingewiesen.

# Kapitel 4

## Vorbereitung der Messdaten

### 4.1 Grundlagen und Hintergründe

BIS jetzt wurden zur Datenauswertung der USANSPOL-Messdaten stets einfache theoretische Modelle verwendet, welche an die Messdaten gefittet wurden. So wird zum Beispiel in [7] ein modellabhängiger Fit für die Instrumentenkurven vorgeschlagen, welcher rein empirisch motiviert ist:

$$R(\theta) = I_0 \sum_{i=1}^2 \left[ A_t \Lambda\left(\frac{\theta - \theta_i}{\tau}\right) + A_0 \mathcal{N}(\theta_i, \sigma_0^2) + \sum_j A_j \mathcal{N}_a(\theta_i, \sigma_{aj}^2, \sigma_{bj}^2) \right]. \quad (4.1)$$

$\Lambda$  ist hier die Dreiecksfunktion, zentriert an  $\theta_i$  mit Breite  $\tau$ ,  $\mathcal{N}(\theta_i, \sigma_0^2)$  ist eine Gaußfunktion zentriert an  $\theta_i$  und einer Breite von  $\sigma_0^2$ , und  $\mathcal{N}_a(\theta_i, \sigma_a^2, \sigma_b^2)$  ist eine asymmetrische Gaußfunktion welche an  $\theta_i$  zentriert ist und eine Breite von  $\sigma_a^2$  links vom Peak respektive  $\sigma_b^2$  rechts vom Peak hat.

Die Dreiecksfunktion  $\Lambda$  wird dadurch motiviert, dass die Rockingkurve die Faltung der Reflektivitäten von Monochromator- und Analysatorkristall ist, welche im Großen und Ganzen Rechteckfunktionen sind [10]. Zwei Rechteckfunktionen miteinander gefaltet ergeben eine Dreiecksfunktion. Die symmetrische Gaußfunktion soll den Rand der Rockingkurve modellieren. Die asymmetrische Gaußfunktion soll asymmetrische Beugungseffekte in das Modell einbringen.

Es ist jedoch von Vorteil einen modellunabhängigen Fit zu verwenden, da man so Effekte welche das Modell nicht berücksichtigt einfließen lassen kann und so nicht so leicht übersieht.

Außerdem ist es notwendig, dass die an non-äquidistanten Stützstellen gemessenen Daten auf äquidistante Messdaten ummodelliert werden, um dis-

krete Entfaltungsalgorithmen auf die Messdaten anwenden zu können, da diese stets gleichgroße Schrittweiten benötigen. Die Winkel der Messdaten sind nicht äquidistant, da der für die Winkelabtastung verwendete Piezoversteller der verwendeten USANSPOL-Anlage keine exakt äquidistanten Schrittweiten abfahren kann.

Da die Messdaten gemäß Poisson-Statistik fehlerbehaftet sind (siehe Abb. 2.2), kann kein gewöhnlicher Spline (für weiterführende Informationen über Splines siehe zum Beispiel [23]) durch die gesamten Daten gelegt werden. Ausreißer bringen unphysikalische Schwingungen ein, vor allem in den Randbereichen wo die Zählrate zu gering ist. Andererseits gehen Splines immer genau durch den Messpunkt. Dies ist bei Messdaten aber nicht unbedingt erwünscht, da sie eine gewisse Unschärfe haben. Genau diese Unschärfe nützt der selbst geschriebene Spline-Algorithmus aus, indem der Spline nicht genau durch den Messpunkt durchgehen muss, sondern gemäß der statistischen  $\sqrt{n}$ -Messungenauigkeit auch daran vorbeilaufen darf.

Dies hat den Vorteil, dass der Spline sehr viel glatter verläuft. Unphysikalische Schwingungen wie in Abbildung 4.4 werden unterdrückt.

Aus Gründen zur

- **Diskretisierung**
- **Fehleroptimierung**
- **Modellunabhängigkeit**

wurde eine Prozedur erdacht, welche einen Spline gemäß der statistischen Messfehler der Messpunkte und selbige legt und diesen dann wiederum an äquidistanten Stützstellen diskretisiert. Dies ermöglicht eine modellunabhängige Diskretisierung der fehleroptimierten Messdaten.

Dieser Spline-Algorithmus wird nun im Detail beschrieben.

## 4.2 Der Spline-Algorithmus

Der Spline-Algorithmus ist ein iterativer Monte-Carlo-Algorithmus, welcher unter Einbeziehung des statistischen Fehlers des Messpunktes einen Spline an die Messpunkte fittet, sodass der Betrag der Krümmung der gesamten Kurve minimal wird.

Dazu werden zuerst die Messpunkte mit Geraden verbunden. Von diesem

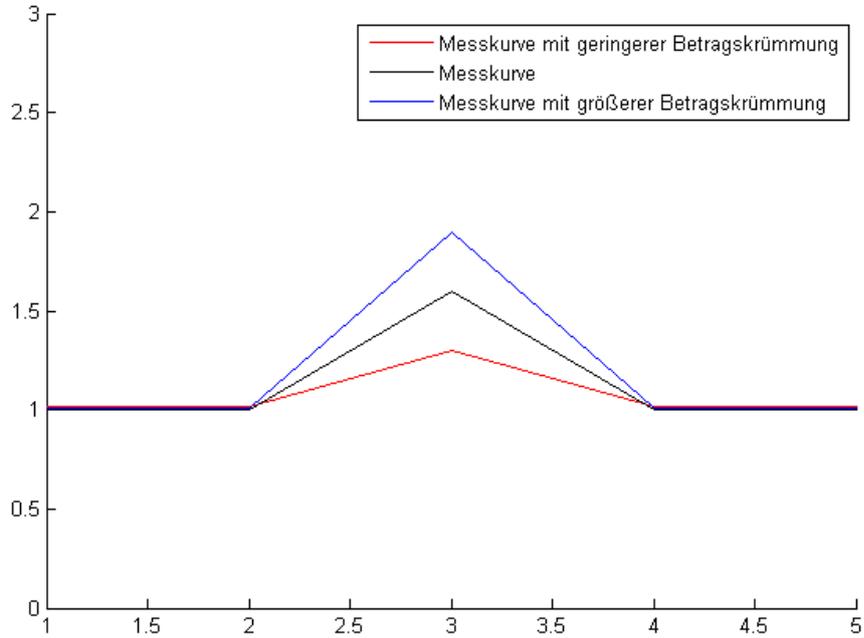


Abbildung 4.1: Polygonzüge mit größerer Krümmung  $\mathcal{C}$  (blau) werden vom Algorithmus verworfen, Polygonzüge mit kleinerer Krümmung  $\mathcal{C}$  (rot) werden verwendet. Dies glättet den Polygonzug.

Polygonzug wird für jede Gerade der Differenzenquotient berechnet. Für die  $i$ -te Gerade lautet der Differenzenquotient  $k_i$  somit:

$$k_i = \frac{x_{i+1} - x_i}{\theta_{i+1} - \theta_i}.$$

wobei  $x_i$  die gemessenen Neutronenintensitäten sind und  $\theta_i$  die Winkel an den jeweils gemessenen Stellen. Die  $\theta$ -Werte im Nenner sind notwendig, da die Messpunkte noch nicht äquidistant sind.

Daraufhin wird der Betrag der Differenz zweier Nachbarsteigungen berechnet:

$$c_i = |k_i - k_{i-1}|.$$

Für die beiden Randpunkte, welche keine äußeren Nachbarn besitzen, wird die Krümmung so definiert, als ob es äußere Nachbarsteigungen mit  $k = 0$  gibt:

$$c_1 = |k_1|$$

und

$$c_n = |-k_n|,$$

wobei  $c_n$  der letzte Messpunkt ist.

Diese Randbedingungen bewirken, dass die Krümmung in den äußeren Bereichen der Kurven minimal wird, was die Realität der Experimente widerspiegelt.

$c_i$  ist ein Maß für die Krümmung des Polynoms im  $i$ -ten Messpunkt. Der Betrag der Gesamtkrümmung der Kurve berechnet sich einfach aus der Summe der Beträge der Einzelkrümmungen. Der Betrag Gesamtkrümmung

$$\mathcal{C} = \sum_i c_i.$$

soll nun mittels Monte-Carlo-Algorithmus minimiert werden

$$\min \mathcal{C} = \min \sum_i c_i.$$

Dazu wird eine Stützstelle ausgewählt und ein alternativer Punkt mittels Zufallsgenerator bestimmt. Ist die Krümmung  $\mathcal{C}$  des alternativen Polygonzuges geringer als des alten Polygonzuges (siehe Abb. 4.1), dann wird der alte verworfen und mit dem neuen Polygonzug weitergearbeitet. Nun wird eine neue Stützstelle ausgewählt und das Prozedere beginnt von vorne.

Das wichtige an dieser Prozedur ist jedoch die Wahl der Zufallsverteilung gemäß der die alternativen Punkte ausgewählt werden: es handelt sich hierbei um eine Gaußverteilung mit einem Erwartungswert  $\mu$  gleich dem Messwert und einer Standardabweichung  $\sigma$  gleich der Größe des statistischen Fehlers des jeweiligen Messpunktes. Hierdurch wird sichergestellt, dass der Polygonzug wie gewünscht leicht von den Messpunkten abweicht um Messfehler wie Ausreißer, Rauschen etc. zu unterdrücken. Da diese Abweichungen der alternativen Punkte um die Messpunkte per Gaußverteilungsfunktion ausgewählt wurden, ist die Wahrscheinlichkeit geringer, dass ein Punkt ausgewählt wird, welcher vom tatsächlichen Messpunkt weit entfernt ist.

Mit dieser Prozedur entsteht somit ein Polygonzug, welcher über die Bedingung der Krümmungsminimierung geglättet wurde und somit von Ausreißern und statistischen Schwankungen so gut wie möglich befreit wurde. Da die alternativen Punkte per Gaußverteilung um den tatsächlichen Messpunkt ausgewählt wurden, ist auch sichergestellt, dass sich die Punkte gemäß Fehlerbalken im Mittel um den Messpunkt gaußverteilt ansiedeln.

Dies ist in Abbildung 4.2 erkennbar: aufgetragen sind die Beträge der Abstände zwischen den tatsächlichen Messpunkten und den vom Algorithmus ermittelten Punkten, wobei diese Residuen auf die positive Seite einer Standardnormalverteilung mit  $\mu = 0$  und  $\sigma = 1$  durch die Formel

$$z_i = \left| \frac{x_i - \mu_i}{\sigma_i} \right|$$

abgebildet werden, mit  $z_i$  als dem jeweiligen Residuum,  $x_i$  als dem Wert des durch den Algorithmus ermittelten Punktes,  $\mu_i$  als dem Wert des tatsächlichen Messpunktes und  $\sigma_i$  als die Größe des statistischen Fehlers, gegeben durch das  $\sqrt{n}$ -Gesetz der Poisson-Verteilung. Es sind rund 73% der neuen Punkte innerhalb des Fehlerbalkens des Messpunktes. Zum Vergleich: Der Fehlerbalken ist jener Bereich um einen Messpunkt in dem der wahre Wert der Messung mit einer Wahrscheinlichkeit von 68,3% liegt. Die beiden Ausreißer auf der rechten Seite lassen sich dadurch erklären, dass die charakteristische Form der Messkurven mit den Doppelpeaks in der Mitte 3 Punkte hat (die beiden Hochpunkte und der Tiefpunkt), an denen die Krümmung extrem groß ist und die alternativen Punkte dadurch weit entfernt von den tatsächlichen Punkten sind.

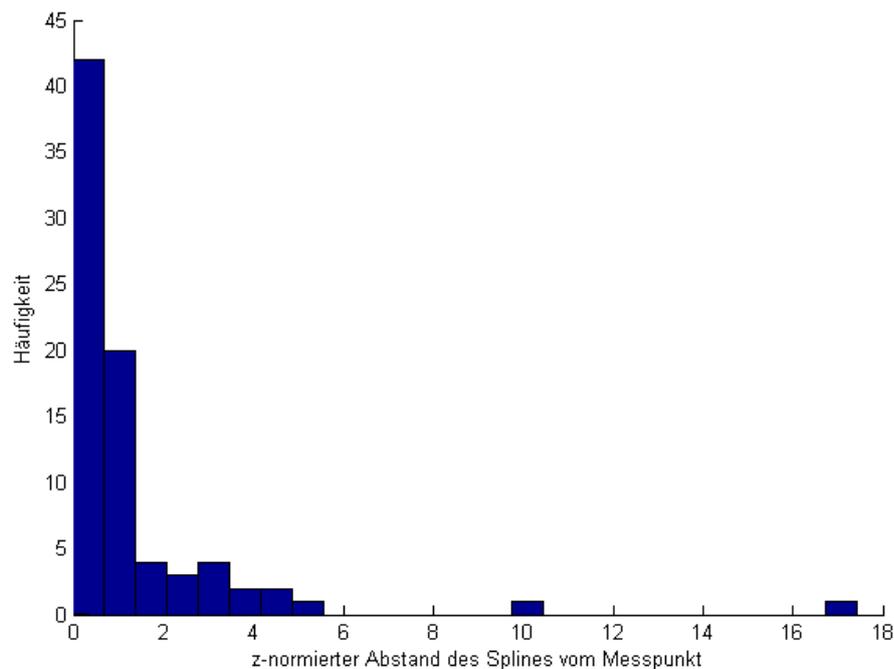


Abbildung 4.2: Verteilung der Beträge der Residuen zwischen tatsächlichen Messpunkten und Punkte durch welche der Spline verläuft. Normiert auf eine Standardnormalverteilung.

Dass Abbildung 4.2 nur annähernd gaußverteilt ist, lässt sich dadurch erklären, dass zwar der Zufallsgenerator die Punkte nach einer Gaußverteilung erzeugt, die Bedingung der Krümmungsminimierung diese Gaußverteilung - aber abhängig von der Kurvenform - nicht unbedingt erhalten muss. Tests mit einfacheren Funktionen wie zum Beispiel eine Sinusfunktion über eine ganze Periode, oder nur das erste Drittel der Messdaten (um den Doppelpeak in der Mitte auszublenden) ergaben jedoch zufriedenstellende Ergebnisse: die Residuen der alternativen Punkte waren vielversprechend (siehe Abb. 4.3).

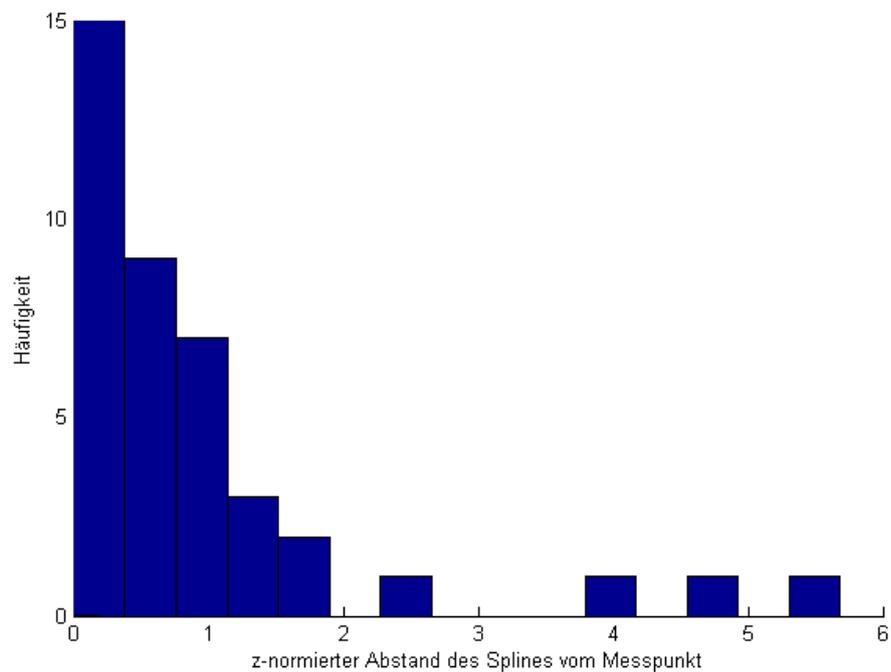


Abbildung 4.3: Residuenverteilung der ersten 40 Messpunkte einer Datenkurve. Da in diesem Bereich der Doppelpeak ausgeschlossen ist, ist die Verteilung einer Gaußverteilung sehr viel näher als Abb. 4.2

Anschließend wird an die Punkte des neuen, geglätteten Polygonzuges ein Spline gelegt. Dieser ist nun natürlich glatter und weist weniger Schwingungen und Ausreißer auf, als sein ursprüngliches Pendant.

Die Prozedur setzt sich zusammenfassend also aus folgenden Schritten zusammen:

1. Verbinden der Messpunkte mit Geraden zur Bildung eines Polygonzuges. Berechnung des Betrages der Gesamtkrümmung  $\mathcal{C}$  des Polygonzuges.
2. Auswahl einer Stützstelle  $\theta_i$
3. Erzeugung eines alternativen Punktes  $x'(\theta_i)$  an der Stützstelle  $\theta_i$  gemäß einer Gaußverteilung mit  $\mu = x(\theta_i)$  und  $\sigma$  gleich dem statistischen Fehler des Messpunktes an  $x(\theta_i)$ .
4. Berechnung des Betrages der Gesamtkrümmung  $\mathcal{C}'$  des alternativen Polygonzuges
5. Ist  $\mathcal{C}' < \mathcal{C}$  so wird der alternative Punkt  $x'(\theta_i)$  gespeichert und weiterverwendet. Ist  $\mathcal{C}' \geq \mathcal{C}$  wird  $x'(\theta_i)$  verworfen.
6. Weiter bei Punkt 3.
7. Abbruchbedingung: Wenn die Schleife eine gewisse Anzahl an Iterationsschritten durchlaufen hat, wird die Prozedur abgebrochen.

### 4.2.1 Diskretisierung der Splines

Wie am Anfang des Kapitels beschrieben, ist die Aufgabe des Spline-Algorithmus die *Fehleroptimierung*, *Modellunabhängigkeit* und *Diskretisierung* im weiteren Umgang mit den Messdaten. Im vorherigen Abschnitt wurden die Messdaten mit einem Spline-Algorithmus modellunabhängig so gefittet, dass die Auswirkung von Ausreißern minimiert wurde, um Schwankungen in den Datenkurven zu minimieren.

Nun wird dieser so erhaltene Spline mit dem selbst erstellten Programm *discretizer.m* diskretisiert, da der Trennungsalgorithmus (siehe Kapitel 5.2) und der Entfaltungsalgorithmus (siehe Kapitel 2.1.3) äquidistante und diskretisierte Messdaten verlangen.

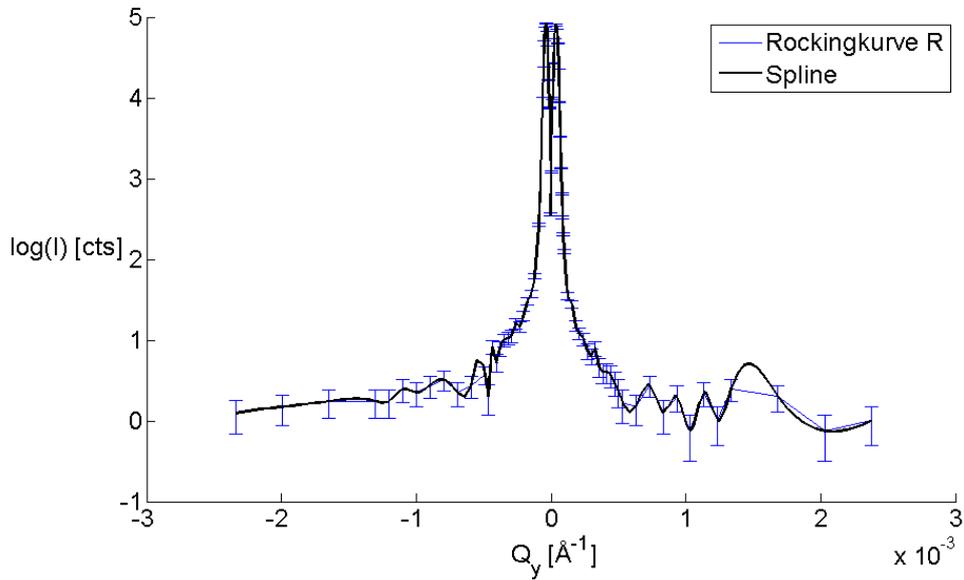


Abbildung 4.4: Ein durch die Messdaten der Rockingkurve gelegter Spline. Man sieht, dass unphysikalische Schwingungen diese Methode unbrauchbar machen.

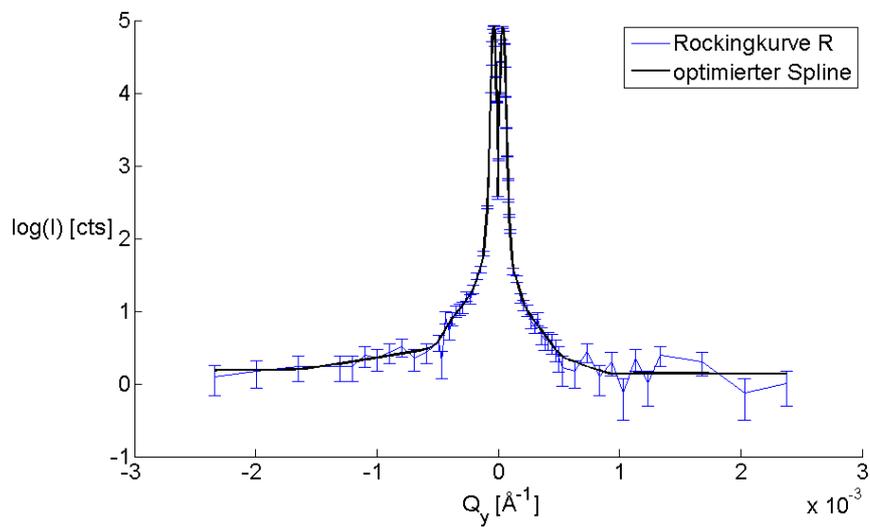


Abbildung 4.5: Ein durch die alternativen Punkte des Spline-Algorithmus gelegter Spline. Man sieht, dass die unphysikalischen Schwingungen nun komplett ausgelöscht wurden. In 73% der Fälle verläuft der Spline durch die Fehlerbalken der jeweiligen Messpunkte ( $\pm\sigma \cong 68,3\%$ ).

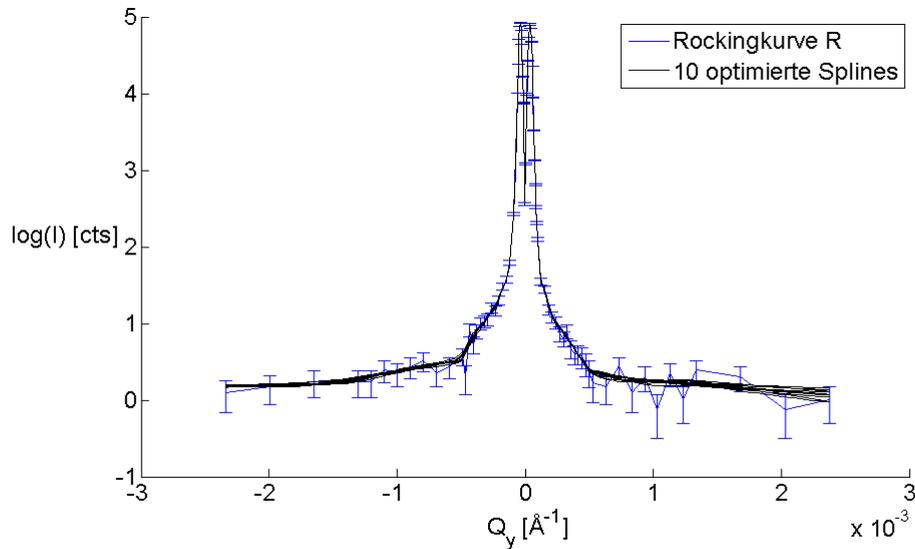


Abbildung 4.6: 10 durch die alternativen Punkte des Spline-Algorithmus gelegte Splines zeigen, dass die entstandenen Splines nur wenig von der Monte-Carlo-Methode abhängen. Schwankungen entstehen quasi nur am Rand im Bereich von kleinen Messraten und daraus resultierenden großen Messungenauigkeiten.

### 4.3 Qualitative Analyse des Spline-Algorithmus

Dadurch, dass in vorgestelltem Algorithmus eine Zufallsvariable verwendet wird, stellt sich natürlich die Frage, ob die erhaltenen Ergebnisse auch reproduzierbar sind, oder ob bei wiederholter Anwendung der Prozedur an dasselbe Set von Messdaten ein anderes Ergebnis und somit auch ein anderer Spline entsteht.

Um dies zu überprüfen wurden nacheinander Splines mit denselben Messdaten erzeugt und übereinander gezeichnet (siehe Abb. 4.6). Man erkennt, dass der Spline-Algorithmus in den Randbereichen unterschiedliche Ergebnisse liefert, da hier die Fehlerbalken zu groß sind. Im Bereich des Doppelpeaks - und dies ist der physikalische interessante Bereich - aber, liefert der Spline-Algorithmus bei öfterem Anwenden auf dieselben Messdaten exzellente übereinstimmende Ergebnisse, da sich die unterschiedlichen Ergebnisse nur minimalst voneinander unterscheiden.

# Kapitel 5

## Entfaltung der Messdaten

### 5.1 Grundlagen zu Entfaltungen

Das USANSPOL-Experiment hat in seiner Genauigkeit natürlich nur eine gewisse Bandbreite. Die Kurve  $P(\theta)$  - in welcher die physikalischen Informationen stecken und somit von Interesse ist - wird vom Messgeräteaufbau gewissermaßen "verschmiert". Diesen Vorgang des Verschmierens kann man mittels Faltungsintegral beschreiben. Hierbei wird die physikalische Kurve  $P(\theta)$ , in der die physikalische Information steckt, mit der Leer-, oder Rockingkurve,  $R(\theta)$  des Messgerätes gefaltet. Das Ergebnis ist die experimentell gemessene Streukurve  $I(\theta)$  [6] [8]:

$$I_{\uparrow\downarrow}(\theta) = R_{\uparrow\downarrow} * P_{\uparrow\downarrow} = \int d\theta' R_{\uparrow\downarrow}(\theta' - \theta) P_{\uparrow\downarrow}(\theta'). \quad (5.1)$$

Man erkennt, dass eine theoretische, exakt arbeitende Messapparatur eine Dirac'sche Deltadistribution als Rockingkurve haben muss:

$$I_{\uparrow\downarrow}(\theta) = \int d\theta' \delta(\theta' - \theta) P_{\uparrow\downarrow}(\theta' - \theta) = P_{\uparrow\downarrow}(\theta).$$

Daraus folgt, dass die physikalische Kurve gleich der Messkurve ist:

$$I_{\uparrow\downarrow}(\theta) = P_{\uparrow\downarrow}(\theta).$$

In der Realität ist die Rockingkurve keine Deltadistribution, sondern sieht vielmehr so aus wie in Abbildung 4.5.

Um die Messdaten physikalisch interpretieren zu können, ist es notwendig die Faltung der physikalischen Kurve mit der Rockingkurve in Gleichung (5.1) zu invertieren, um sich so  $P_{\uparrow\downarrow}(\theta)$  ausrechnen zu können. Man würde also

gerne den Umkehroperator der Faltung, genannt Entfaltung oder Dekonvolution (in dieser Arbeit zum besseren Verständnis und zur Vereinfachung lapidar mit " $*^{-1}$ " bezeichnet), auf Gleichung (5.1) anwenden

$$I_{\uparrow\downarrow}(\theta) = R_{\uparrow\downarrow}(\theta) * P_{\uparrow\downarrow}(\theta)$$

um so die vom Messgerät unabhängige Streukurve zu erhalten

$$P_{\uparrow\downarrow} = I_{\uparrow\downarrow}(\theta) *^{-1} R_{\uparrow\downarrow}(\theta)$$

welche es dann zu interpretieren gilt.

Die Entfaltung ist zwar im Prinzip möglich, aber leider im Allgemeinen nicht brauchbar, da sie extrem empfindlich auf Fehler und Rauschen in den Eingangsdaten ist. Die gemessenen Daten haben gemäß Poisson-Statistik eine Messunschärfe proportional zu  $\sqrt{n}$ , was somit die simple Entfaltung unbrauchbar macht.

Um die Rockingkurve dennoch aus den Daten herausfalten zu können, gibt es in der Literatur mehrere Möglichkeiten. In der Diplomarbeit von Tobias Rechberger zum Beispiel werden lineare Gleichungssysteme mittels Stabilisierungsalgorithmen gelöst, um so eine Näherungslösung der Entfaltung berechnen zu können [1]. Im Gegensatz dazu wurde für diese Arbeit ein iterativer Algorithmus geschrieben, welcher versucht die Messkurve aus normierten und verschobenen Rockingkurven zu modellieren, um somit eine Näherungslösung des Entfaltungsproblems zu erhalten.

Bevor dieser in Kapitel 5.3 beschrieben wird, muss aber noch die gemessene Gesamtrockingkurve  $R(\theta)$  in ihre  $\uparrow$ - und  $\downarrow$ -Anteile aufgeteilt werden.

## 5.2 Trennungsalgorithmus der Rockingfunktion

Gemessene Rockingkurven, wie zum Beispiel in Abb. 5.1, sind tatsächlich eine Summe aus zwei Rockingkurven der Up- und Down-Neutronen. Es gilt also folgende Gleichung:

$$R_{\uparrow}(\theta) + R_{\downarrow}(\theta) = R(\theta). \quad (5.2)$$

Während die Strahldivergenz des einfallenden polarisierten, aber unaufgespaltenen Neutronenstrahls ungefähr  $0.5^\circ$  ist, spalten die magnetischen Prismen die Anteile der  $\uparrow$ - und  $\downarrow$ -Neutronen nur um etwa  $22 \mu rad$  auf. Es ist also ersichtlich, dass die beiden Neutronenspinanteile durch das magnetische Prisma nur unwesentlich voneinander getrennt werden. Die Messapparatur,

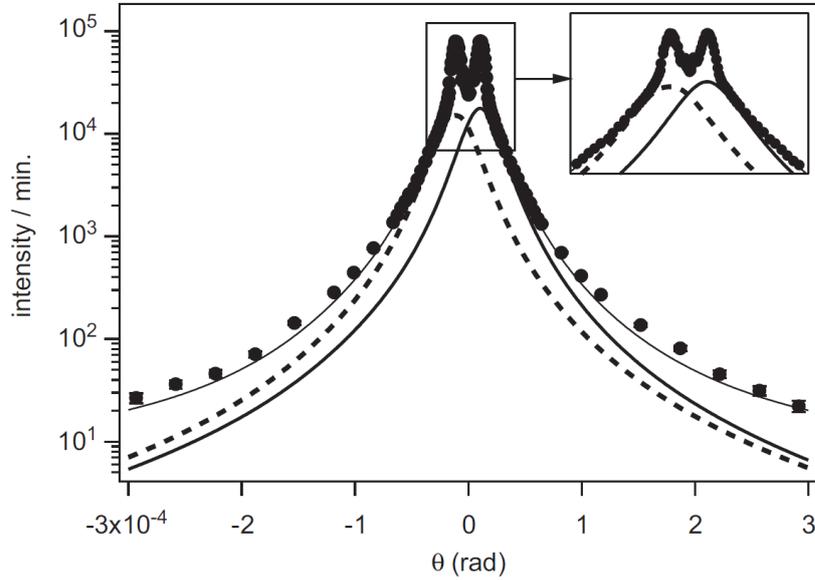


Abbildung 5.1: Die strichlierte Linie ist die mit der Messkurve gefaltete Streukurve der Spin-up Neutronen, die durchgezogene Linie die der Spin-down Neutronen. Die schwarzen Punkte sind Messpunkte, die Linie durch die Messpunkte eine Fittingkurve. Die Summe aus Spin-up und Spin-down Kurve ergibt die gefittete Kurve. [6]

insbesondere die Blenden, Magnetprismen und die Kristalle, welche für die Verschmierung maßgeblich beteiligt sind, wirkt also für die beiden Spinkomponenten des Strahles in sehr guter Näherung gleich. Durch diese Überlegung ist es möglich die Rockingfunktionen  $R_{\uparrow}(\theta)$  und  $R_{\downarrow}(\theta)$  in Gleichung (5.2) als gleich und um den Winkel  $\Delta$  verschoben anzusetzen:

$$r\left(\theta + \frac{\Delta}{2}\right) + r\left(\theta - \frac{\Delta}{2}\right) = R(\theta). \quad (5.3)$$

$r(\theta)$  ist die gesuchte Auflösungsfunktion des Messgerätes,  $\Delta$  ist der Winkel, um welchen die beiden Auflösungsfunktionen für  $\uparrow$ - und  $\downarrow$ -Neutronen durch das magnetische Prisma gegeneinander verschoben sind und  $R(\theta)$  ist - wie gewohnt - die gemessene Gesamtrockingkurve. Die gemessene Gesamtrockingfunktion  $R(\theta)$  besteht somit in guter Näherung aus zwei äquivalenten Rockingfunktionen für Spin  $\uparrow$ - und Spin  $\downarrow$ -Komponenten.

Alternativ kann man sich den Sachverhalt der Rockingkurven aber auch wie folgt vorstellen: Die gemessene Rockingkurve  $R(\theta)$  kann man aus zwei Faktoren zusammengesetzt ansehen, welche in der Messapparatur eine ein-

deutig getrennte Ursache haben. So bewirken die Blenden, Monochromator-, Analysatorkristall und die magnetischen Prismen eine Verschmierung der zu messenden Kurve, welche aber für beide Spinkomponenten gleich sind. Dieser Anteil der Rockingkurve sieht aus wie in Abbildung 5.2 dargestellt. Dem gegenüber steht der aufspaltende Effekt den die magnetischen Prismen auf die Neutronen mit unterschiedlichen Spinkomponenten haben.

Man kann die gemessene Rockingkurve somit als Singlerockingfunktion  $r(\theta)$  ansehen, welche um  $+\frac{\Delta}{2}$  beziehungsweise um  $-\frac{\Delta}{2}$  verschoben und aufaddiert wurde.

Diese Singlerockingfunktion  $\vec{r}(\theta)$  wurde mit einem Algorithmus gefunden, welcher in folgendem Abschnitt erklärt werden möchte.

### 5.2.1 Der Trennungsalgorithmus im Detail

Da die diskreten Messdaten schon in äquidistanten Winkelschritten vorliegen, ist es möglich einen iterativen Algorithmus zu entwickeln, welcher die Funktion  $r(\theta)$  aus Gleichung (5.3) berechnet, sodass sie addiert mit sich selbst - nur um  $\Delta$  verschoben -  $R(\theta)$  ergibt.

Da es sich um äquidistante Messdaten handelt, können  $R(\theta)$  und  $r(\theta)$  als Vektor aufgefasst werden.  $\vec{R}(\theta)$  ist somit ein Vektor der Dimension gleich der Anzahl der Messpunkte, zum Beispiel  $n = 500$ , und  $\vec{r}(\theta)$  ist ein Vektor gleich der Anzahl der Datenpunkte weniger der Verschiebung  $\Delta$  zwischen  $\uparrow$ -Peak und  $\downarrow$ -Peak.  $\Delta$  ist zum Beispiel 8, also ist die Dimension von  $\vec{r}(\theta)$  gleich 492. Gleichung (5.3) lässt sich somit leicht in ein lineares Gleichungssystem der Form

$$\mathbf{A}\vec{x} = \vec{b} \tag{5.4}$$

bringen, wobei  $x$  der gesuchte Vektor der Daten der Rockingkurve ist, und  $\vec{b}$  ist der Vektor  $\vec{R}$  der gemessenen Daten.  $\mathbf{A}$  muss somit eine Matrix sein, welche einen  $(n - \Delta)$ -dimensionalen Vektor auf einen  $n$ -dimensionalen Vektor abbildet, also eine  $n \times (n - \Delta)$ -Matrix.

Gleichung (5.3) lässt sich in Matrix-Vektor-Schreibweise unter Verwendung

der Matrizen  $\mathbf{B}$  und  $\mathbf{C}$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 \\ 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 \end{pmatrix}$$

also wie folgt darstellen:

$$\mathbf{B}\vec{r} + \mathbf{C}\vec{r} = \vec{R}. \quad (5.5)$$

Mit

$$(\mathbf{B} + \mathbf{C}) = \mathbf{A}$$

gilt somit

$$\mathbf{A}\vec{r} = \vec{R}. \quad (5.6)$$

Die Matrix  $\mathbf{B}$  bildet die  $i$ -te Vektorkomponente eines  $(n-\Delta)$ -dimensionalen Vektors auf die  $i$ -te Komponente eines  $n$ -dimensionalen Vektors ab. Die übrigen  $\Delta$  Komponenten bekommen den Eintrag 0. Die Matrix  $\mathbf{C}$  bildet die Komponenten des  $(n-\Delta)$ -dimensionalen Vektors  $\vec{r}$  auch auf einen  $(n-\Delta)$ -dimensionalen Vektor ab, nur um  $\Delta$  verschoben.

Berechnet man nun die Matrix  $\mathbf{A}$  und bringt die erweiterte Koeffizientenmatrix der Gleichung (5.6) auf eine obere Dreiecksform, so erhält man folgendes Ergebnis:

$$(\mathbf{A}|\mathbf{b}) = \left( \begin{array}{cccc|cccc} 1 & 0 & 0 & \cdots & 0 & R_1 & & & \\ 0 & 1 & 0 & \cdots & 0 & R_2 & & & \\ 0 & 0 & 1 & \cdots & 0 & R_3 & & & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & & & \\ 0 & 0 & 0 & \cdots & 1 & R_\Delta & & & \\ 0 & 0 & 0 & \cdots & 0 & R_\Delta - R_1 & & & \\ 0 & 0 & 0 & \cdots & 0 & R_{\Delta+1} - R_2 & & & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & & & \\ 0 & 0 & 0 & \cdots & 0 & R_n - R_{n-\Delta-1} & & & \\ 0 & 0 & 0 & \cdots & 0 & R_n - R_{n-\Delta} & & & \end{array} \right).$$

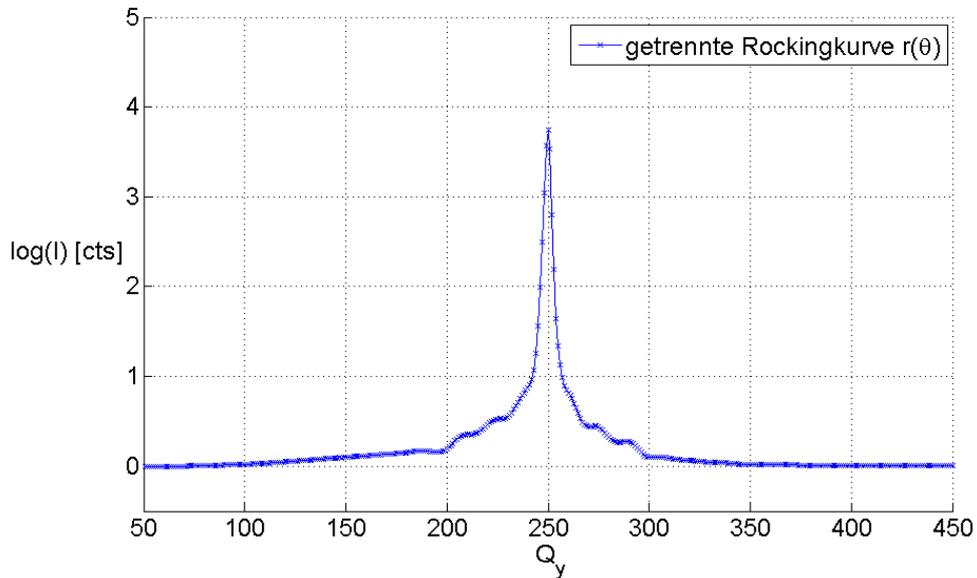


Abbildung 5.2: Aus  $R(\theta)$  separierte Singlerockingfunktion  $r(\theta)$ .

Es lässt sich leicht ablesen, dass somit das lineare Gleichungssystem (5.6) ohne einschränkende Bedingungen für die Inhomogenität  $\vec{R}$ , also der Rockingkurve, keine Lösung hat, da der Rang der Matrix  $\mathbf{A}$  stets größer ist als der Rang der erweiterten Koeffizientenmatrix  $(\mathbf{A}|\mathbf{b})$ .

Natürlich lassen sich aber Lösungen finden, welche Gleichung (5.5) nur näherungsweise erfüllen indem entweder die Summe nicht exakt  $\vec{R}$  ergibt, oder die Vektoren  $\vec{r}$  nicht exakt gleich sind. Dazu wurde eine Methode entwickelt, welche einen Vektor  $\vec{r}$  findet, der um  $\Delta$  verschoben bis auf einen vernachlässigbaren Fehler den Vektor der Messdaten der Rockingkurve  $\vec{R}$  ergibt.

Diese Näherungslösung wird iterativ berechnet, indem in jedem Iterationsschritt der Wert der Rockingkurve  $R(\theta_i)$  mit der Summe von  $r(\theta_i)$  und  $r(\theta_{i+\Delta})$  verglichen wird. Ist besagte Summe kleiner als  $R(\theta_i)$  wird an  $r(\theta_i)$  und  $r(\theta_{i+\Delta})$  jeweils ein Inkrement  $dy$  addiert, ansonsten subtrahiert. Danach wird  $i$  um 1 erhöht und die Prozedur beginnt von vorne. Wurde die Laufzeit lange genug gewählt, so nähert sich  $r(\theta)$  sehr gut einer Funktion an, welche Gleichung (5.5) löst (siehe Abb. 5.2).

## 5.3 Entfaltungsalgorithmus

Da es mit der in Matlab eingebauten Entfaltungsfunktion *deconv.m* nicht gelang die Rockingkurve aus der Messkurve herauszufalten, wurde es notwendig eine eigene Entfaltungsfunktion zu schreiben, welche an das spezielle Problem angepasst ist. Die Grundidee hinter diesem Algorithmus wurde der Dissertation von Martin Hainbuchner entnommen [3].

Wegen der Diskretheit der modellierten Messdaten wird aus dem Integral

$$I(\theta) = P(\theta) * R(\theta) = \int_{-\infty}^{\infty} P(\theta) R(\theta' - \theta) d\theta,$$

welches eine Faltung der physikalischen Kurve  $P(\theta)$  mit der Gerätekurve  $R(\theta)$  darstellt, eine Summe über alle Messpunkte:

$$I(\theta_i) = \sum_j P(\theta_j) R(\theta'_j - \theta_i) \Delta\theta.$$

Da man die Messkurve  $I(\theta)$  und die Rockingkurve  $R(\theta)$  kennt und die physikalische Kurve  $P(\theta)$  sucht, möchte man die Faltung durch deren Umkehroperation, die Entfaltung, rückgängig machen. Dies würde theoretisch exakt funktionieren. Da aber die Messdaten mit einer gewissen Messungenauigkeit behaftet sind und die Dekonvolution extrem empfindlich auf Fehler in den Daten ist, ist eine einfache Rückfaltung leider keine Option. In der Dissertation von Martin Hainbuchner wird ein iterativer Algorithmus zur Entfaltung beschrieben, welcher im Zuge diese Diplomarbeit in abgewandelter Form verwendet wurde.

## 5.4 Der MyDeconv.m-Algorithmus

Grundlage dieser Prozedur ist die Idee, dass sich die gemessene Intensität in jedem Punkt der Streukurve  $I(\theta)$  aus einer Summe von entsprechend verschobenen und normierten Rockingkurven zusammensetzt. Der Algorithmus berechnet somit, wie viele normierte Rockingkurven für jeden  $\theta$ -Wert notwendig sind, um aus der Summe die Streukurve zu modellieren. Die Anzahl dieser normierten Rockingkurven an der jeweiligen Stelle  $\theta$  ergibt die Entfaltung  $P(\theta)$ .

Der Algorithmus setzt sich aus folgenden Schritten zusammen:

1. Normieren der Rockingkurve, sodass gilt:

$$\sum R(\theta) = 1.$$

2. Eine Pivotstelle wird ausgewählt: wenn die Summenfunktion an dieser Stelle unter der Messkurve liegt, wird eine speziell normierte Rockingkurve addiert, sodass an dieser Pivotstelle die Summenfunktion gleich der Messkurve ist.

Liegt die Summenfunktion über der Messkurve, wird der Normierungsfaktor für die speziell normierte Rockingkurve berechnet. Nur wenn dieser abgezogen vom derzeitigen Wert von  $P(\theta)$  größer als 0 ist, wird die Subtraktion durchgeführt. Dies stellt sicher, dass die Entfaltungsfunktion  $P(\theta)$  nur positive Werte annimmt, da negative Werte unphysikalisch wären.

3. Der Normierungsfaktor der Rockingkurve wird in einem Array unter den jeweiligen Streuwinkeln aufsummiert. Die so entstehende Kurve ergibt die gesuchte physikalische Kurve  $P(\theta)$ .
4. Die nächste Pivotstelle wird ausgewählt.
5. Zurück zu Punkt 2.
6. Abbruchbedingung: Es werden die Differenzen zwischen den Punkten der Messkurve und der somit verschobenen Rockingkurven modellierten Approximation der Messkurve berechnet und addiert. Ist diese Summe kleiner als eine vorgegebene Schranke, wird der Algorithmus abgebrochen.

Die so in Punkt 3 aufsummierte Kurve ergibt eine Approximation der gesuchten Entfaltung  $P(\theta)$ .

Eine Möglichkeit zu testen, ob die so gefundene Entfaltung prinzipiell stimmen kann, ist es die Integrale der einzelnen Funktionen zu berechnen. Aus Gleichung (5.1) folgt, dass sich die Flächen der einzelnen Funktionen wie folgt verhalten müssen:

$$\int I(\theta)d\theta = \int P(\theta) * R(\theta)d\theta = \int P\theta d\theta \int R(\theta)d\theta. \quad (5.7)$$

Da die Rockingkurve in dieser Arbeit bei Entfaltungen auf 1 normiert wurde, folgt daraus, dass

$$\int I(\theta)d\theta = \int P(\theta)d\theta \quad (5.8)$$

immer gelten muss. Dass somit die Flächen unter den Kurven der Entfaltung und der Messkurve immer gleich sein müssen ist eine notwendige (aber keine hinreichende!) Bedingung und somit ein nützliches Indiz um zu erkennen, ob der selbst entwickelte Entfaltungsalgorithmus prinzipiell richtige Lösungen ausgibt.

Weiters wurde der Algorithmus mit Modellmessdaten ausführlich auf Vertrauenswürdigkeit überprüft.

### 5.4.1 Normieren der Rockingfunktionen

Im Zuge dieser Arbeit stellte sich die Frage nach der richtigen Normierung der Funktionen  $I(Q_y)$ ,  $P(Q_y)$ ,  $R(Q_y)$  und  $r(Q_y)$ . Diese sind natürlich prinzipiell frei wählbar, jedoch müssen sie immer konsistent durchgeführt werden, sodass die entstandenen Kurven und deren Intensitäten vergleichbar bleiben. Hierfür haben sich im Zuge dieser Arbeit gewisse „Normen zur Normierung“ herauskristallisiert, welche garantieren sollen, dass Normierungen stets konsistent durchgeführt werden.

So wurden  $I(Q_y)$  und  $P(Q_y)$  stets nicht normiert, sondern ihre Intensitäten beibehalten, da es sich hierbei um gemessene Kurven und somit um Neutronen pro Minute handelt.

Demgegenüber stehen aber die Rockingkurven  $R(Q_y)$  und  $r(Q_y)$ , welche stets auf 1 normiert wurden, falls sie in Faltungen oder Entfaltungen verwendet wurden. Dies hat den einfachen Grund, dass über Gleichung (5.8) somit automatisch die gemessene Intensität  $I(Q_y)$  und die physikalische Kurve  $P(Q_y)$  im selben Intensitätsbereich sein müssen, man kann also sofort erkennen, ob man durch einen Rechenfehler Neutronen „verloren“ hat.

Für diesen Sachverhalt der Normierung von  $R(Q_y)$  auf 1 gibt es jedoch eine alternative Erklärung, welche sehr zum Verständnis beitragen kann. Man kann sich die gemessene Rockingkurve als Wahrscheinlichkeitsverteilung des Neutrons im Teilchenbild vorstellen. Diese Wahrscheinlichkeitsverteilung  $R(Q_y)$  beschreibt wie groß die Wahrscheinlichkeit ist, dass das Neutron, welches die Messapparatur passiert, durch selbige einen zusätzlichen Streuvektoranteil  $Q_y$  bekommt. Somit hat ein jedes Neutron, welches die Messapparatur passiert, eine gewisse Unschärfe in der Messrichtung (y-Achse), welche durch die Rockingkurve als Wahrscheinlichkeitsverteilung beschrieben werden kann. Diese Wahrscheinlichkeitsverteilung muss natürlich immer auf 1 normiert sein, da die Wahrscheinlichkeit, das Neutron irgendwo im Bereich von  $-\infty$  bis  $+\infty$  zu detektieren 1 sein muss. Außerdem muss auch die Teilchenzahlerhaltung gelten.

Aus diesem Grund wird - wie in Kapitel 6 beschrieben - die Rockingfunktion  $r(Q_y)$  nicht normiert wenn man den ungestreuten Anteil  $p$  abschätzen

möchte, da es sich in diesem Fall nicht um eine Wahrscheinlichkeitsverteilung handelt welche aus einer Messungenauigkeit stammt, sondern, um tatsächliche Neutronenintensitäten.

Außerdem gibt es noch eine andere Normierungsvariante, in welcher man der Einfachheit halber alle Funktionen auf 1 normiert. Dies ist zwar der einfachere Weg, man verliert dadurch aber die Aussagekraft der Intensitäten der Kurve. Aus diesem Grunde wurde diese Art der Normierung in dieser Arbeit nicht angewandt.

Zusammenfassend lässt sich also sagen, dass nur die Rockingfunktionen  $R(Q_y)$  und  $r(Q_y)$  auf 1 normiert werden, aber auch nur wenn sie als Wahrscheinlichkeitsverteilung, also im Zuge einer Faltung oder Entfaltung verwendet werden, und nicht als Funktion, welche den ungestreuten Anteil beschreiben soll. Alle anderen Funktionen werden keiner Normierung unterzogen, was zur Folge hat, dass Intensitäten stets physikalisch bleiben.

## 5.5 Qualitative Analyse der Entfaltungsroutine

Zur Überprüfung der selbst entwickelten Entfaltungsprozedur *MyDeconv.m* wurden Modellmessdaten generiert, welche aus einfachen Funktionen aufgebaut sind, von welchen man das Verhalten unter Faltungen und Entfaltungen bereits gut kennt. Hierfür bieten sich Gaußfunktionen sehr an, da diese gefaltet mit Gaußfunktionen wiederum Gaußfunktionen ergeben. Aus diesem Grunde wurde  $R(\theta)$  und  $P(\theta)$  jeweils aus 3 Gaußfunktionen nach dem Schema

$$A(\theta) = \sum_{i=1}^3 C_i \exp\left(-\left(\frac{\theta - \mu_i}{\sigma_i}\right)^2\right)$$

zusammengesetzt, sodass eine gewisse subjektive Ähnlichkeit mit den Messkurven (Doppelpeak, welcher in der Mitte scharf getrennt ist; Untergrund, welcher nach außen gegen 0 geht) besteht. Die Parameter der so willkürlich gefundenen Kurven für  $P(\theta)$  und  $R(\theta)$  können aus Tabelle 5.1 entnommen werden.

Um die Entfaltungsprozedur zu testen, wurde nun  $P(\theta)$  einmal mit  $R(\theta)$  gefaltet und einmal mit  $R(\theta)$  mit der Funktion *MyDeconv.m* entfaltet, wodurch ein  $P'(\theta)$  berechnet wird, welches theoretisch gleich  $P(\theta)$  sein muss.

Die Ergebnisse dieses Prozederes sind in Abbildung 5.3 abgebildet. Sie sprechen absolut für die Funktion *MyDeconv.m*:

Abbildung 5.3 zeigt im ersten Plot die modellierten Funktionen  $R(\theta)$  und  $P(\theta)$ , im zweiten Plot ist die von der Spinaufspaltung getrennte Rockingfunk-

Modellfunktion $P(\theta)$ :		
C	$\mu$	$\sigma$
0.8	$+6 \cdot 10^{-5}$	$7 \cdot 10^{-5}$
0.8	$-6 \cdot 10^{-5}$	$7 \cdot 10^{-5}$
0.2925	0	$6 \cdot 10^{-4}$

Modellfunktion $R(\theta)$ :		
C	$\mu$	$\sigma$
0.8	$+6 \cdot 10^{-5}$	$4 \cdot 10^{-5}$
0.8	$-6 \cdot 10^{-5}$	$4 \cdot 10^{-5}$
0.2925	0	$3 \cdot 10^{-4}$

Tabelle 5.1: Parameter einer willkürlich zusammengesetzten Rocking- und Messfunktion aus jeweils 3 Gaußfunktionen.

tion  $r(\theta)$  dargestellt, im dritten Plot wurde  $P(\theta)$  mit eben dieser Rockingfunktion gefaltet, um  $I(\theta)$  zu erhalten und im letzten Plot ist die Entfaltung von  $I(\theta)$  mit  $r(\theta)$  mittels *MyDeconv.m*.

**Nota bene:** Für einen Vergleich von  $P(\theta)$  mit  $P'(\theta)$  wurden diese Kurven im untersten Plot in Abbildung 5.3 übereinander geplottet. Da *MyDeconv.M* aber so gut funktioniert, sind diese Funktionen deckungsgleich.

Da nun gezeigt wurde, dass *MyDeconv.m* problemlos mit Gaußfunktionen umgehen kann, wurde im nächsten Schritt  $P(\theta)$  und  $R(\theta)$  besser an die Messdaten angepasst und zusätzlich mit einem Rauschsignal überlagert. Hierfür wird das *Matlab*-interne *Curve Fitting Tool* zum Fitten von Kurven verwendet, welches auf eine Summe von 5 Gaußfunktionen

$$A(\theta) = \sum_{i=1}^5 C_i \exp \left( - \left( \frac{\theta - \mu_i}{\sigma_i} \right)^2 \right) \quad (5.9)$$

einen Least-Square-Fit anwendet, um die Parameter zu berechnen, welche in Tabelle 5.2 gefunden werden können.

Die so gefundenen Funktionen  $P(\theta)$  und  $R(\theta)$  durchlaufen nun dasselbe Prozedere wie oben beschrieben um  $P'(\theta)$  zu berechnen. Das Ergebnis dieser Prozedur mit verrauschten Gaußfunktionen kann als sehr positiv beschrieben werden, wie man in Abbildung 5.4 rechts unten erkennt: die ursprüngliche Kurve  $P(\theta)$  wird durch Falten und Entfalten mittels *MyDeconv.m* wieder sehr gut in sich selbst übergeführt.

Modellfunktion  $P(\theta)$ :

C	$\mu$	$\sigma$
1.559	$+3.816 \cdot 10^{-5}$	$3.34 \cdot 10^{-5}$
1.624	$-3.768 \cdot 10^{-5}$	$2.76 \cdot 10^{-5}$
0.2925	$-8.425 \cdot 10^{-5}$	$2.763 \cdot 10^{-5}$
1.603	$+3.888 \cdot 10^{-6}$	$2.145 \cdot 10^{-4}$
1.655	$-2.413 \cdot 10^{-5}$	$8.547 \cdot 10^{-4}$

Modellfunktion  $R(\theta)$ :

C	$\mu$	$\sigma$
3.028	$-3.883 \cdot 10^{-5}$	$3.006 \cdot 10^{-5}$
3.231	$+4.011 \cdot 10^{-5}$	$3.697 \cdot 10^{-5}$
0.01726	$+1.261 \cdot 10^{-4}$	$5.709 \cdot 10^{-6}$
0.3801	$-8.437 \cdot 10^{-5}$	$4.676 \cdot 10^{-5}$
1.804	$-3.301 \cdot 10^{-5}$	$3.733 \cdot 10^{-4}$

Tabelle 5.2: Parameter einer aus jeweils 5 Gaußfunktionen zusammengesetzten Rocking- und Messfunktion, welche per Least-Square-Verfahren an Messdaten gefittet wurden.

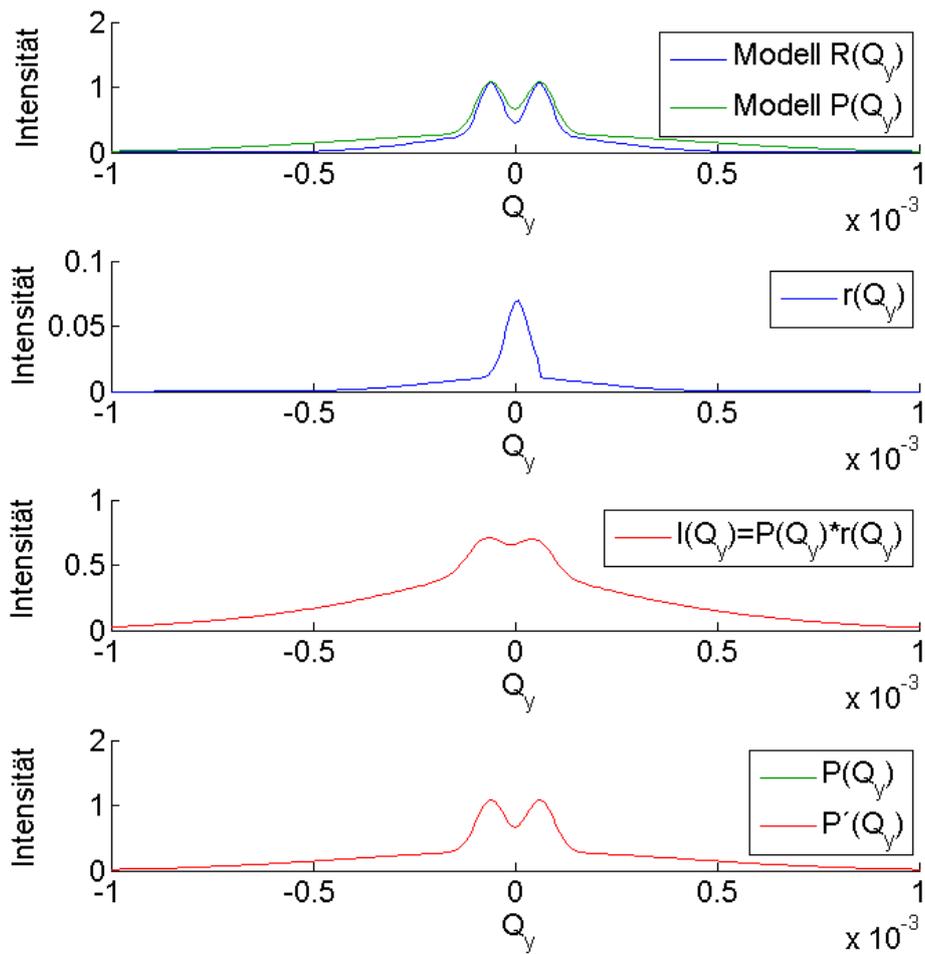


Abbildung 5.3: Modellkurven aus Gaußfunktionen  $P(\theta)$ ,  $R(\theta)$  (1. Teilbild) werden gefaltet (3. Teilbild), entfaltet ( $P'(\theta)$ , 4. Teilbild), und verglichen (auch 4. Teilbild). Man erkennt, dass  $P(\theta)$  und  $P'(\theta)$  deckungsgleich sind. Für Details siehe Text.

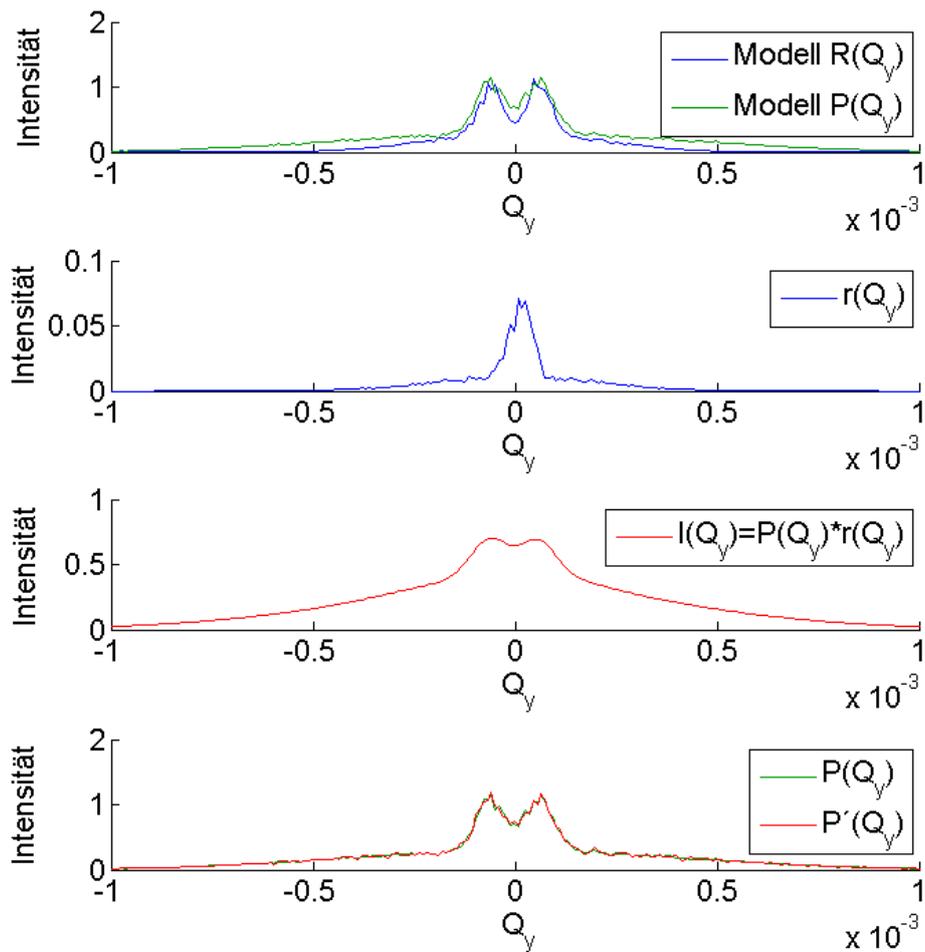


Abbildung 5.4: Modellkurven aus Gaußfunktionen mit zufälligem Störterm  $P(\theta)$ ,  $R(\theta)$  (1. Teilbild) werden gefaltet (3. Teilbild), entfaltet ( $P'(\theta)$ ), 4. Teilbild) und verglichen (auch 4. Teilbild). Man erkennt, dass  $P(\theta)$  und  $P'(\theta)$  beinahe deckungsgleich sind. Für Details siehe Text.

## Kapitel 6

# Abschätzung des ungestreuten Anteils

UM Mehrfachstreuungen in der Probe vernachlässigen zu können, ist die Dicke der Probe so dimensioniert, dass der überwiegende Anteil der Neutronen  $p$  nur maximal ein einziges mal gestreut wird. Dies hat aber zu Folge, dass ein Anteil  $(1 - p)$  von Neutronen überhaupt nicht gestreut wird. Dieser Anteil an der Messkurve  $I(\theta)$  wird *ungestreuter Anteil* genannt. Dadurch ergibt sich für  $I(\theta)$  aus Gleichung (2.4)

$$I(\theta) = p \cdot P(\theta) * r(\theta) + q \cdot R(\theta) \quad (6.1)$$

mit  $p$  als der Streuwahrscheinlichkeit für Neutronen,  $q = 1 - p$  ist die dazugehörige Gegenwahrscheinlichkeit.  $r(\theta)$  ist die Rockingfunktion des experimentellen Aufbaus ohne magnetische Aufspaltung, so wie sie in dieser Arbeit in Abschnitt 5.2 definiert wurde.

Es wird also auf den physikalischen Anteil  $p \cdot P(\theta) * r(\theta)$  ein ungestreuter Anteil  $q \cdot R(\theta)$  aufaddiert, welcher die Form einer speziell normierten Rockingkurve hat. Nun ist es jedoch leider so, dass man den Faktor  $p$  nicht kennt. Es hat sich jedoch herausgestellt, dass die Kenntnis dieses Faktors sehr wichtig ist um die Entfaltung der Messkurve  $I(\theta)$  mit der Rockingkurve durchzuführen, da in Gleichung (6.1) der ungestreute Anteil die Effekte der physikalischen Streukurve  $P(\theta)$  überschatten würde, so wie dies in Abbildung 6.1 zu erkennen ist: Es wurde für Demonstrationszwecke vernachlässigt den ungestreuten Anteil abzuziehen, es gilt also  $p = 1$ . Die beiden hohen Peaks haben ihre Ursachen im zweiten Term in Gleichung (6.1). Genau diese Peaks machen es leider unmöglich die Daten zu interpretieren.

Wie bereits oben erwähnt liegt das daran, dass das Experiment so konzipiert werden muss, dass der Faktor  $q$  relativ hoch ist. Leider gibt es keine zuverlässigen Daten über  $p$  und auch nur sehr geringe Erfahrungswerte. So wurde in Simulationen in [20] eine grobe Abschätzung von 10 – 20% für diesen Faktor bestimmt. Dieser Wert gilt aber natürlich nur für die verwendete Probe. Ohne Abzug des ungestreuten Anteils ist die Entfaltung der Messdaten leider nicht verwendbar.

Weiters kommt hinzu, dass die Streuwahrscheinlichkeit für  $|\uparrow\rangle$ - und  $|\downarrow\rangle$ -Neutronen unterschiedlich sein kann - beziehungsweise genau dieser Unterschied der Streuwahrscheinlichkeiten genau einer der Punkte ist, der die Evolution von USANS zu UNSANSPOL motiviert und somit von hohem Interesse ist. Dies bedeutet also, dass Gleichung (6.1) im Prinzip nur für UNSANS gilt. Möchte man jedoch die Neutronenspins miteinbeziehen, muss die Gleichung adaptiert werden, indem die Gesamtrockingkurve  $R(\theta)$  durch die beiden Einzelrockingkurven für  $|\uparrow\rangle$ - und  $|\downarrow\rangle$ -Neutronen substituiert wird:

$$I(\theta) = p \cdot P(\theta) * r(\theta) + q_{\uparrow} \cdot r_{\uparrow}(\theta) + q_{\downarrow} \cdot r_{\downarrow}(\theta) \quad (6.2)$$

mit

$$p + \frac{q_{\uparrow}}{2} + \frac{q_{\downarrow}}{2} = 1$$

als Normierung zur Teilchenzahlerhaltung, wobei  $p$  die Streuwahrscheinlichkeit ist, und  $q_{\uparrow} = 1 - p_{\uparrow}$  und  $q_{\downarrow} = 1 - p_{\downarrow}$  die Vorfaktoren für den ungestreuten Anteil für die jeweilige Spinkomponente sind. Da

$$\int R(\theta) d\theta = \frac{1}{2} \int r_{\uparrow}(\theta) d\theta$$

und

$$\int R(\theta) d\theta = \frac{1}{2} \int r_{\downarrow}(\theta) d\theta$$

gilt, die beiden Singlerockingkurven also jeweils die halbe Intensität der Doppelpeak-Rockingkurve haben, folgt daraus, und aus Gleichung (6.2), dass

$$p = \frac{p_{\uparrow}}{2} + \frac{p_{\downarrow}}{2}$$

für die Gesamtstreuwahrscheinlichkeit gelten muss.

Formt man Gleichung (6.2) um, so erhält man somit folgende Gleichung:

$$P(\theta, p_{\uparrow, \downarrow}) = \frac{1}{p} [I(\theta) - (1 - p_{\uparrow})r_{\uparrow}(\theta) - (1 - p_{\downarrow})r_{\downarrow}(\theta)] *^{-1} r(\theta). \quad (6.3)$$

Diese Gleichung beschreibt nun die physikalische Kurve  $P(\theta, p_{\uparrow}, p_{\downarrow})$ , in Abhängigkeit der Streuwahrscheinlichkeiten für beide Spinkomponenten  $p_{\uparrow}$  und  $p_{\downarrow}$ , welche man problematischerweise aber leider nicht kennt.

Es wurde jedoch im Zuge dieser Arbeit eine Methode gefunden, welche zumindest eine Abschätzung dieses Faktors ermöglicht.

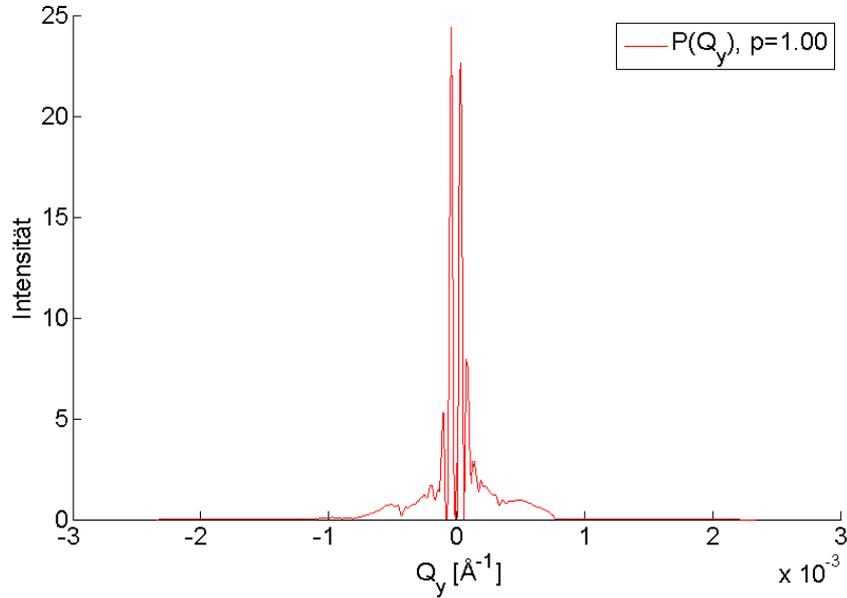


Abbildung 6.1: Zu Demonstrationszwecken der Auswirkung der Vernachlässigung des ungestreuten Anteils wurde  $p_{\uparrow\downarrow} = 1$  gesetzt. Physikalische Effekte von Interesse werden dadurch überschattet.

## 6.1 Grundlagen zur Abschätzung des ungestreuten Anteils

Man erkennt aus Gleichung (6.3), dass die Lösung der Entfaltung vom ungestreuten Anteil abhängt. Das Problem ist nun wie im oberen Abschnitt beschrieben, dass man  $p_{\uparrow\downarrow}$  nicht kennt, somit also die Entfaltung nicht durchführen kann.

Um die Faktoren  $p_{\uparrow\downarrow}$  aber zumindest grob abschätzen zu können, kann man  $p_{\uparrow\downarrow}$  jeweils von 0 bis 1 mit einer Schrittweite von  $\Delta p_{\uparrow\downarrow}$  abrastern und die Entfaltung aus Gleichung (6.3) mit den jeweiligen Werten für  $p_{\uparrow\downarrow}$  durchführen. Wählt man bei dieser Vorgangsweise für  $\Delta p_{\uparrow\downarrow}$  eine Schrittweite von beispielsweise jeweils 0,1 entstehen so eine Reihe aus 100 Kurven für  $P(\theta, p_{\uparrow\downarrow})$ , welche es nun zu analysieren gilt. Aus dieser Analyse lassen sich Werte für  $p_{\uparrow\downarrow}$  abschätzen.

Um herauszufinden auf welche Merkmale man in dieser Kurvenschar achten muss, um den richtigen Wert für  $p_{\uparrow\downarrow}$  zu erkennen und um diese Prozedur auf Vertrauenswürdigkeit und Funktionsfähigkeit zu testen wurde wieder ein Satz von Modelldaten erstellt und oben beschriebene Prozedur darauf angewandt.

## 6.2 Qualitative Analyse der Prozedur mit Modelldaten

Um die oben beschriebene Prozedur auf ihre Funktionsweise zu testen wurden wie auch in Abschnitt 5.5 Modelldaten  $P(\theta)$  und  $R(\theta)$  erzeugt, welche aus Gaußfunktionen bestehen und der Algorithmus auf sie angewandt. In diesem Abschnitt wurde beschrieben, dass der Entfaltungsalgorithmus, der auch hier wieder die Grundlage bilden wird, für aus Gaußfunktionen zusammengesetzte Mess- und Rockingkurve ausgezeichnet funktioniert.

Um einen vertrauenswürdigen Test für diese Prozedur zu erstellen, ist es notwendig, dass die Modellfunktionen  $P(\theta)$  und  $R(\theta)$  jene Eigenschaften besitzen, welche die realen Messdaten auch haben. Diese Eigenschaften gilt es nun zu beschreiben, um brauchbare Modellfunktionen erstellen zu können.

Während für die Modellrockingkurve  $R(\theta)$  zum Beispiel die Werte aus Tabelle 5.1 verwendet werden können, soll die Modellfunktion für  $P(\theta)$  vier Peaks besitzen, da man aus theoretischen Herleitungen erwartet, dass für magnetische Streuung der Neutronen jeweils zwei Peaks pro Spineinstellung entstehen. Weiters ist es notwendig, dass die Dimensionen von  $P(\theta)$  und  $R(\theta)$  so aufeinander abgestimmt sind, dass beim Übereinanderlegen der Kurven die Peaks der Rockingkurve genau über den äußeren Minima von  $P(\theta)$  liegen, welche die Minima der Streukurven der einzelnen Spinkomponenten sind. Diese Eigenschaften führen auf eine willkürliche Funktion für  $P(\theta)$ , welche aus einer Summe von 5 Gaußfunktionen nach Gleichung (5.9) zusammengestellt wird (siehe Abbildung 6.2), deren Koeffizienten in Tabelle 6.1 wiedergegeben sind.

Um das Prozedere zur Abschätzung des ungestreuten Anteils mit Modellfunktionen zu testen, wird auf die Modellfunktion  $P(\theta)$  eine Rockingfunktion multipliziert mit dem Faktor  $k \in ]0, 1]$  aufaddiert. Um diesen Test einfach zu halten wird ein ungestreuter Anteil  $kR(\theta)$  aufaddiert, welcher nicht zwischen den Spins unterscheidet und nicht zwei unterschiedliche Anteile für  $|\uparrow\rangle$  und  $|\downarrow\rangle$  hat:

$$P'(\theta) = P(\theta) + kR(\theta), \quad (6.4)$$

mit

$$k := 0.3,$$

Diese Gleichung simuliert somit einen „unbekannten“ ungestreuten Anteil. Diesen Faktor  $k$  gilt es mittels der oben vorgestellten grafischen Methode herauszufinden.

Modellfunktion  $P(\theta)$ :

C	$\mu$	$\sigma$
0.83	$-12 \cdot 10^{-5}$	$4 \cdot 10^{-5}$
0.83	$+12 \cdot 10^{-5}$	$4 \cdot 10^{-5}$
0.3	$+0 \cdot 10^{-5}$	$6 \cdot 10^{-4}$
0.8	$-4 \cdot 10^{-6}$	$4 \cdot 10^{-5}$
0.8	$+4 \cdot 10^{-5}$	$47 \cdot 10^{-5}$

Modellfunktion  $R(\theta)$ :

C	$\mu$	$\sigma$
3.0	$-8 \cdot 10^{-5}$	$3.006 \cdot 10^{-5}$
3.0	$+8 \cdot 10^{-5}$	$3.697 \cdot 10^{-5}$
1.8	0	$3.733 \cdot 10^{-4}$

Tabelle 6.1: Parameter einer aus Gaußfunktionen zusammengesetzten Modellfunktion  $P(\theta)$  und  $R(\theta)$  (Siehe auch Abbildung 6.2).

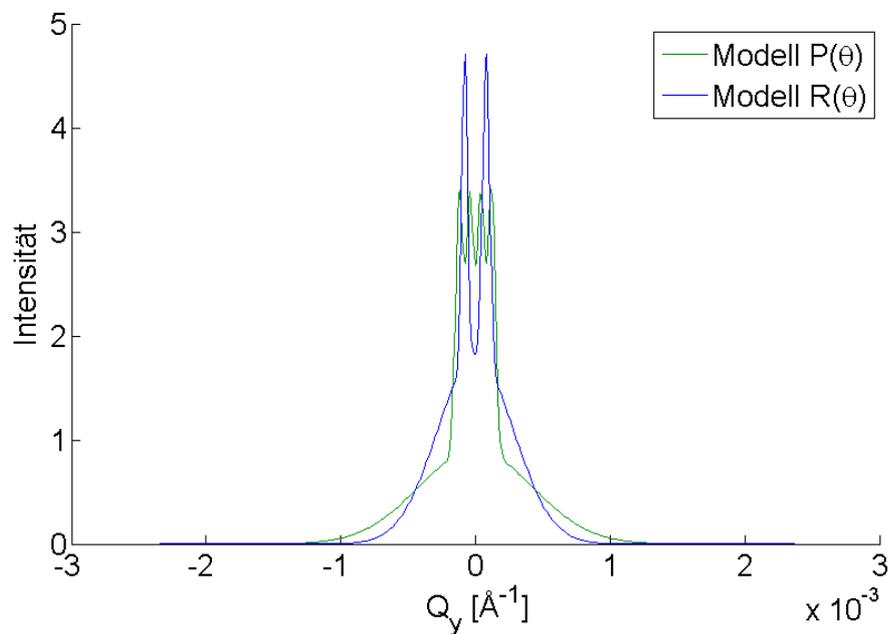


Abbildung 6.2: Modellfunktionen für  $P(\theta)$  und  $R(\theta)$  mit den Parametern aus Tabelle 6.1.

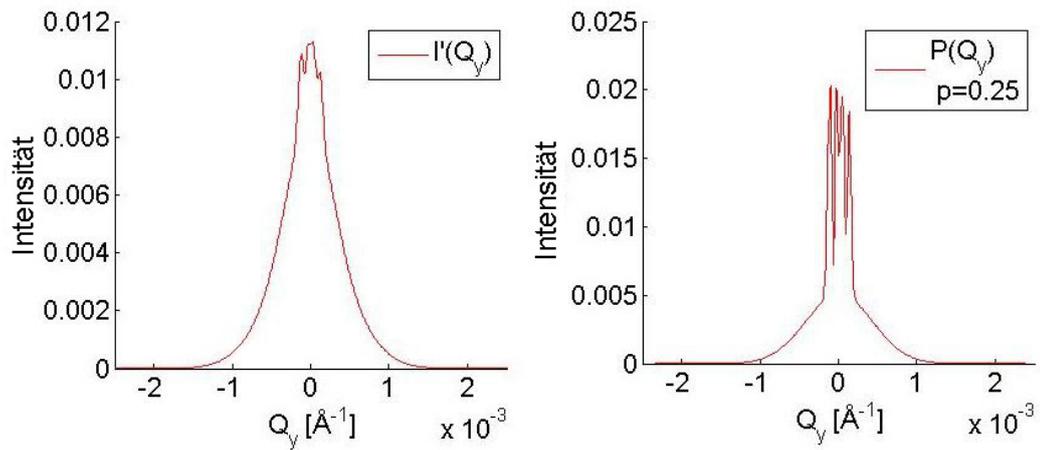


Abbildung 6.3: Darstellung von  $I'$  und  $P(Q_y)$  für  $k = 0.3$  und  $p = 0.25$ .

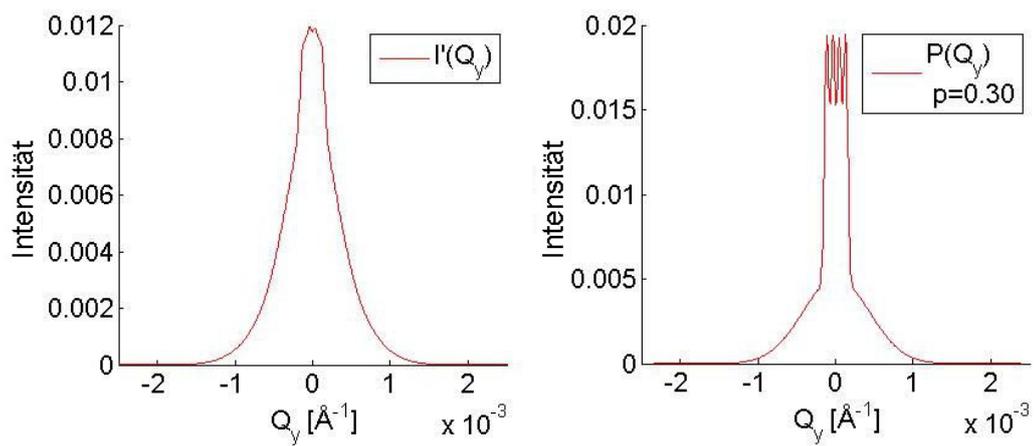


Abbildung 6.4: Darstellung von  $I'$  und  $P(Q_y)$  für  $k = 0.3$  und  $p = 0.30$ .

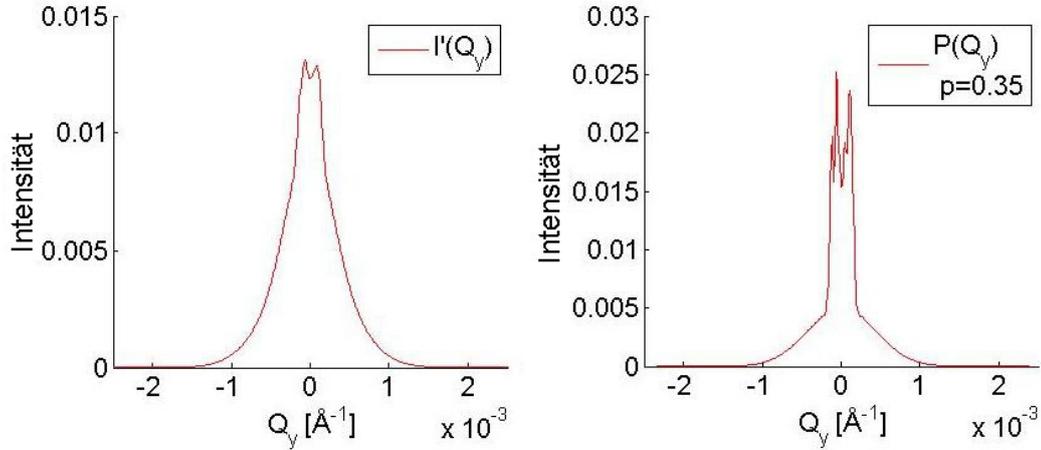


Abbildung 6.5: Darstellung von  $I'$  und  $P(Q_y)$  für  $k = 0.3$  und  $p = 0.35$ .

Man wählt also eine Schrittweite  $\Delta p$ , mit welcher Gleichung (6.3) abgerastert werden soll, in diesem Fall wird  $\Delta p = 0.1$  gesetzt, es entstehen somit zehn Plots. Diese Plots gilt es nun auf Eigenschaften zu analysieren, welche auf das wahre  $p$  der Probe hindeuten. Dazu sind in den Abbildungen 6.3, 6.4 und 6.5 jeweils auf der linken Seite die Funktion

$$I'(Q_y) = I(Q_y) - (1 - p)R(Q_y),$$

und  $P(Q_y)$  jeweils auf der rechten Seite für Werte für  $p$  von 0.25, 0.30 und 0.35 abgedruckt.  $I'(Q_y)$  ist hier die gemessene Neutronenintensität abzüglich des vermuteten ungestreuten Anteils und  $P(Q_y)$  ist die Entfaltung von  $I'(Q_y)$  mit der Singlerockingfunktion  $r(Q_y)$ . Es lässt sich nun durch Vergleichen der Abbildungen feststellen, dass die Prozedur - für Modellfunktionen - extrem empfindlich auf die Wahl von  $p$  ist.

So ist erkennbar, dass in Abbildung 6.3 - also der Fall, in welchem die Streuwahrscheinlichkeit als zu gering eingeschätzt wird, indem 5 Prozentpunkte zu viel ungestreuter Anteil abgezogen wird - einerseits die Kurve nicht sehr glatt ist, und andererseits haben die vier Peaks in der Entfaltung nicht die selbe Intensität. Dies sind aber Eigenschaften, die man sich zumindest näherungsweise von der Entfaltung erwartet, was daran liegt, dass die beiden linken Peaks Intensitäten von gestreuten Neutronen mit Spin  $|\uparrow\rangle$ , und die beiden rechten Peaks Intensitäten von gestreuten Neutronen mit Spin  $|\downarrow\rangle$  darstellen, und Streuung der Neutronen um  $Q_{\uparrow\downarrow y} = 0$  immer symmetrisch sein muss.

Dies lässt sich anschaulich mit theoretischen zweidimensionalen Streudarstellungen aus Abbildung 2.1 erklären, welche in Kapitel 2.1.1 eingehend erklärt wurden. Man kann erkennen, dass die Symmetrie des in Abbildung

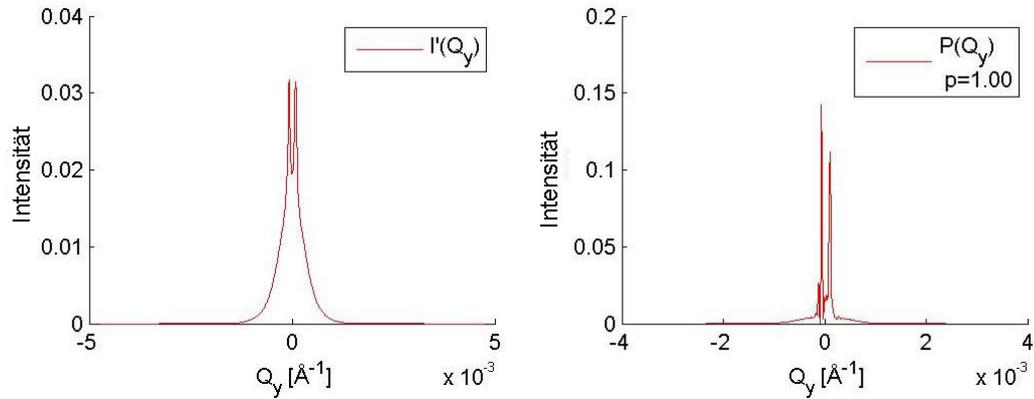


Abbildung 6.6: Darstellung von  $I'$  und  $P(Q_y)$  für  $k = 0.3$  und  $p = 1.00$ . Man erkennt deutlich, dass der nichtabgezogene ungestreute Anteil zwei Peaks erzeugt.

2.1.d durchgeführten slit height smearing unabhängig von der jeweiligen Drehung der Probe (entspricht den Abbildungen (a-c)) ist. Dies bedeutet, dass die Streuwahrscheinlichkeit für jede Spinkomponente um den jeweiligen Wert für  $Q_{\uparrow\downarrow y} = 0$  symmetrisch sein muss. Bei der Suche nach einer Näherung für  $p$  gilt es also stets darauf zu achten, dass die beiden linken Peaks und die rechten beiden Peaks in der Intensitätsverteilung jeweils gleich hoch sein müssen, da sie der symmetrischen Streuung der beiden Neutronenspinkomponenten um den Wert von  $Q_{\uparrow\downarrow y} = 0$  entsprechen.

Daraus lässt sich somit aus Abbildung 6.3 und 6.5 schließen, dass hier der Streufaktor  $p$  zu gering beziehungsweise als zu hoch eingeschätzt wurde. Schätzt man  $p$  zu hoch ein, so führt das unter anderem dazu, dass in den Funktionen von  $P(Q_y)$  zwei Spitzen auftauchen, welche dem ungestreuten Anteil entsprechen, den man fälschlicherweise nicht abgezogen hat (siehe Abbildung 6.6). Schätzt man umgekehrt  $p$  zu klein ein, so zieht man zu viel des ungestreuten Anteils ab. Dies resultiert darin, dass die Funktion  $P(Q_y)$  an den Stellen, für welche der Streuvektor  $\vec{q}$  Spin- $|\uparrow\rangle$  respektive Spin- $|\downarrow\rangle$  0 ist mehr oder weniger starke Einbrüche zu erkennen sind (siehe Abbildung 6.7).

Auch ein Blick auf die Funktion  $I'(Q_y)$ , welche in den drei Abbildungen jeweils auf der linken Seite für die dazugehörigen Werte von  $p$  dargestellt ist, lässt wichtige Schlüsse auf den Faktor  $p$  zu. Die Funktion  $I'(Q_y)$  ist in Abbildung 6.4 bei weitem am glattesten, da hier der richtige Anteil des ungestreuten Anteils abgezogen wurde. In den Abbildungen 6.3 und 6.5 sind immer noch leichte Spitzen oder Einbrüche zu erkennen.

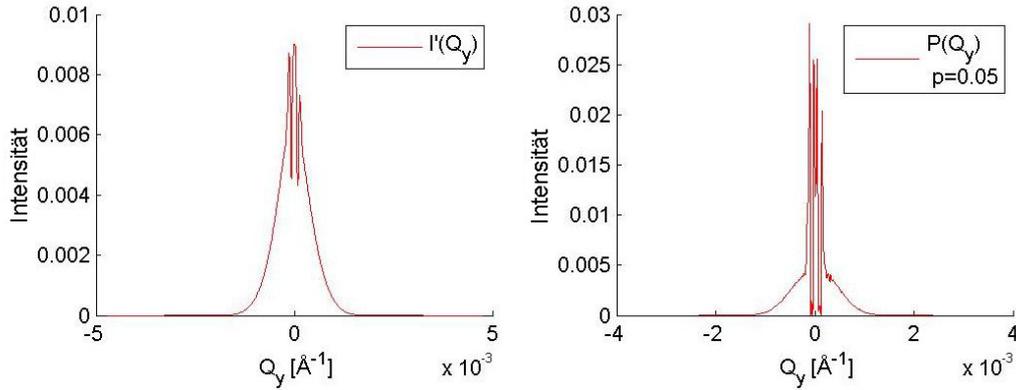


Abbildung 6.7: Darstellung von  $I'$  und  $P(Q_y)$  für  $k = 0.3$  und  $p = 0.05$ .

Sieht man sich nun zum Vergleich Abbildung 6.4 an, so untermauert dies die oben beschriebenen „weichen“ Eigenschaften, welche zur Findung des Streufaktors  $p$  beitragen.

Über diese Analyse mittels Modellfunktionen wurden so nun Eigenschaften gefunden, welche die Funktion  $I'(Q_y)$  und  $P(Q_y)$  besitzen muss, sodass man Schlüsse auf den Parameter  $p$  machen kann. Mit Modellfunktionen war dies sehr einfach, und dadurch lässt sich hier auch  $p$  auf mindestens ein Hundertstel genau abschätzen. Verwendet man jedoch anstatt Modellfunktionen reale Messkurven verringert sich natürlich die Genauigkeit der Abschätzung. Diese Abschätzung ist jedoch immer noch ein ausgezeichneter Anhaltspunkt für weitere Arbeiten.

### 6.3 Abschätzung der Streuwahrscheinlichkeit $p$

Wie mittels Gleichung (6.2) beschrieben, besteht die gemessene Intensitätskurve  $I(\theta)$  aus der physikalischen Streukurve  $P(\theta)$  und dem ungestreuten Anteil der  $|\uparrow\rangle$ - und  $|\downarrow\rangle$ -Neutronen  $(1 - p_\uparrow)r_\uparrow(\theta)$  respektive  $(1 - p_\downarrow)r_\downarrow(\theta)$ . Gleichung (6.4) ist also wie beschrieben nur ein einfaches Modell für eine erste Näherung gewesen. Tatsächlich muss die Gesamtrockingkurve  $I(\theta)$  in ihre Einzelkomponenten  $r_\downarrow(\theta)$  und  $r_\uparrow(\theta)$  aufgespalten und somit voneinander getrennt aufaddiert werden. Hiermit ergeben somit die beiden Faktoren  $p_{\downarrow\uparrow}$ . Diese rastert man nun analog zu oben beschriebenem Beispiel mit den Modellfunktionen ab, um so auf Intensitätsverteilungen zu stoßen, bei welchen die vier mittleren Peaks ungefähr gleich hoch sind. Dies wurde für verschie-

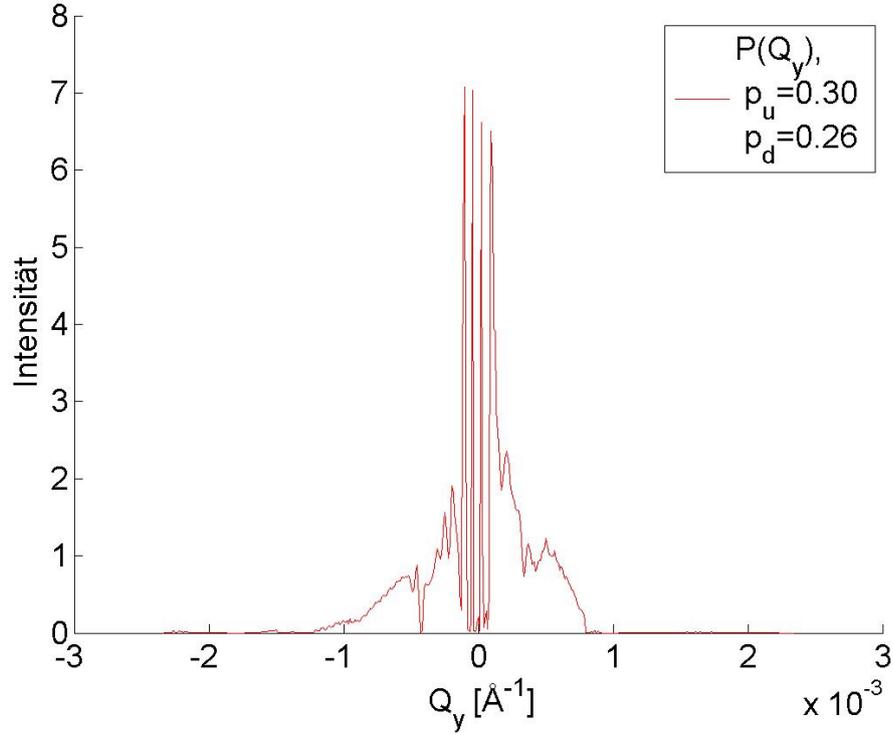


Abbildung 6.8: Entfaltung einer Messkurve mit korrekt abgezogenem ungestreuten Anteil für  $|\uparrow\rangle$ - und  $|\downarrow\rangle$ -Neutronen. Der ungestreute Anteil der Spin- $|\uparrow\rangle$ -Neutronen ist 30%, jener der Spin- $|\downarrow\rangle$ -Neutronen 26%.

dene Messdaten durchgeführt, wobei hier die  $\alpha$ -Probe mit einem Winkel von  $0^\circ$  und einem externen Magnetfeld von  $0\text{ G}$  exemplarisch erklärt werden soll.

Durch systematisches Absuchen der Wertebereiche von  $p_{\downarrow\uparrow}$  nach gleich hohen Intensitäten für  $|\uparrow\rangle$ - und  $|\downarrow\rangle$ -Neutronen wurde eine Abschätzung für  $p_{\downarrow\uparrow}$  gefunden (siehe Abbildung 6.8). Diese Abschätzung lautet

$$p_{\uparrow} = 0.30$$

$$p_{\downarrow} = 0.26.$$

Zum Vergleich ist in Abbildung 6.9 eine Entfaltung mit schlecht justiertem ungestreuten Anteil abgedruckt. Man kann erkennen, dass die beiden inneren Peaks wesentlich höher sind als die äußeren. Dies ist physikalisch jedoch nicht möglich, wie in Abschnitt 6.2 ausführlich beschrieben wurde.

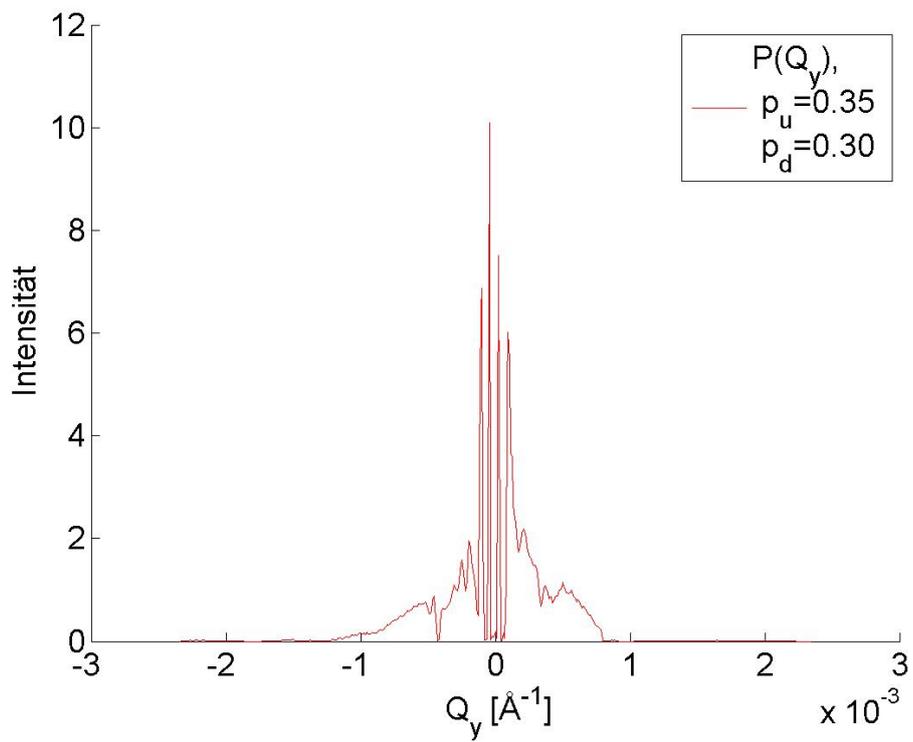


Abbildung 6.9: Eine Entfaltung einer Messkurve mit nicht korrekt abgezogenem ungestreuten Anteil von  $p_{\uparrow} = 0.35$  und  $p_{\downarrow} = 0.30$  für  $|\uparrow\rangle$ - und  $|\downarrow\rangle$ -Neutronen. Man erkennt deutlich, dass die beiden inneren Peaks um einiges höher sind als die äußeren. Dies ist physikalisch jedoch nicht möglich.

### 6.3.1 Conclusio

Wie erwartet, hat sich bei der Anwendung der oben beschriebenen Prozedur auf die Messdaten sehr schnell herausgestellt, dass die in den Modelldaten erzielte Genauigkeit nicht zu erreichen ist. Jedoch ist eine Abschätzung im selben Genauigkeitsbereich wie in [20] schon jetzt möglich. Würde man jedoch alle Algorithmen optimieren, so ließe sich die Genauigkeit sicher um einiges steigern. Dies war aus zeitlichen Gründen bis jetzt leider nicht möglich. So hat sich gezeigt, dass die Abschätzung der Faktoren  $p_{\downarrow\uparrow}$  sehr von den getrennten Single-Rockingfunktionen  $r_{\downarrow\uparrow}$  abhängt. Nur leider ist die Hypothese, dass die Rockingfunktionen der einzelnen Spinkomponenten die gleiche Form haben müssen weder experimentell noch theoretisch untermauert. Wie in Kapitel 5.2 beschrieben, ist es so, dass der Trennungsalgorithmus für die Rockingfunktion keine eindeutige exakte Lösung besitzt. Man muss sich somit auf Näherungslösungen beschränken, von denen noch nicht vollkommen klar ist, auf welche Eigenschaften dieser Näherungslösungen mehr Augenmerk gelegt werden soll. So kann man einerseits zwei Funktionen  $r_{\uparrow}$  und  $r_{\downarrow}$  finden, welche exakt gleich - nur zueinander verschoben - sind, nur ergibt deren Summe leider nicht genau die Gesamtrockingfunktion  $R(\theta)$ . Andererseits kann man natürlich auch nach zwei Funktionen  $r_{\uparrow}$  und  $r_{\downarrow}$  suchen, deren Summe exakt  $R(\theta)$  ergeben soll, nur können sie dann in vielen Fällen nicht genau die gleiche Form haben. Weiters hängt das erhaltene Ergebnis auch von der Differenz

$$\Delta(\theta) = R(\theta) - r_{\uparrow}(\theta) - r_{\downarrow}(\theta)$$

ab, welche den gemachten Fehler beschreibt. Durch die Funktion  $\Delta(\theta)$  lässt sich sehr gut erkennen, wie der gemachte Fehler verteilt ist, da zum Beispiel Fehler an den Schultern für starkes Rauschen sorgen, und somit die Methode unbrauchbar wird. Die Qualität der Abschätzung der Faktoren  $p_{\downarrow\uparrow}$  hängt also sehr von der Qualität der Trennung der Rockingkurve ab. Aus diesem Grunde können die Zahlenwerte für  $p_{\downarrow\uparrow}$  nur als grobe Schätzung, als erste Näherung angesehen werden.

Weiters muss erwähnt werden, dass die Ergebnisse in [1] für  $p_{\downarrow\uparrow}$  der Probe von den Ergebnissen in dieser Arbeit doch signifikant abweichen. Es wurde jedoch ein Test mit einem gemeinsamen ungestreuten Anteil für beide Spinkomponenten wie in Gleichung (6.4) beschrieben durchgeführt, welcher für beide Prozeduren sehr ähnliche Ergebnisse lieferte. Die Diskrepanz dürfte also mit ziemlicher Sicherheit auf die Trennung der Rockingfunktionen für die Analyse des ungestreuten Anteiles zurückzuführen sein.

# Kapitel 7

## Zusammenfassung & Ausblick

Ein sehr wesentlicher und zeitintensiver Punkt des USANSPOL-Experimentes ist die Auswertung und Analyse der gemessenen Daten. Eine große Schwierigkeit dabei ist die Faltung der physikalischen Streukurve  $P(\theta)$  mit der Gerätekurve  $R(\theta)$  zur gemessenen Neutronenintensitätsverteilung  $I(\theta)$

$$I(\theta) = P(\theta) * R(\theta).$$

Dies führt dazu, dass  $I(\theta)$  einerseits nicht von der Messapparatur unabhängig ist und andererseits ist die physikalische Streukurve gewissermaßen „verschmiert“, sodass eine direkte Interpretation von  $I(\theta)$  nicht möglich ist. Aus diesem Grunde wurden in der Vergangenheit mehrere Lösungsansätze entwickelt, um aus  $I(\theta)$  und  $R(\theta)$  die gesuchte Streukurve  $P(\theta)$  zu erhalten. Eine Möglichkeit hierfür ist zum Beispiel durch einen *educated guess* eine passende Streukurve  $P(\theta)$  zu erstellen, welche man mit der bekannten Gerätekurve faltet, und so mit der gemessenen Intensitätskurve vergleichen kann [2]. Durch diesen Vergleich kann man rückschließen, ob die anfangs nur „geratene“ Funktion  $P(\theta)$  der tatsächlichen Funktion nahe kommt, oder nicht. Dieser inverse Weg der Extraktion von  $P(\theta)$  zur Datenanalyse wird gewählt, da man die Faltung zwar theoretisch umkehren kann, dieser Umkehroperator der Faltung - die *Entfaltung* - aber nur schwer zu handhaben ist, da sie einerseits sehr empfindlich auf Messfehler in den Daten ist, andererseits aber nicht zwingend eine eindeutig Lösung liefern muss. Genau dieser direkte Weg der Datenanalyse, also die Entfaltung der Messdaten mit der Rockingkurve  $R(\theta)$  wurde im Zuge dieser Diplomarbeit aber gewählt, um herauszufinden ob die Entfaltung von Messdaten mit Messfehlern prinzipiell möglich ist. Dieser *proof of principle* kann als absolut erfolgreich angesehen werden.

Ein weiterer Vorteil dieser Vorgehensweise ist die modellunabhängige Datenanalyse. Bisherige Lösungsansätze haben stets mehr oder weniger komplizierte Modellfunktionen verwendet, welche zum Beispiel aus Summen von Gauß-, Sinus Cardinalis-, Cosinus-, Rechtecks-, Dreiecksfunktionen et cetera bestehen. Der entscheidende Vorteil der Modellunabhängigkeit in dieser Arbeit ist, dass somit keine physikalischen Effekte übersehen werden können, welche in den verwendeten Modellen nicht bedacht wurden.

Um diese direkte, modellunabhängige Entfaltungsmethode zur Messdatenanalyse zu bewerkstelligen war es notwendig eine Reihe von Prozeduren zu programmieren. Da die Messdaten als Zählvorgang der Poisson-Statistik unterworfen sind und dadurch einem gewissen statistischen Fehler unterliegen, eine Standardentfaltungsroutine jedoch höchst sensibel auf Fehler ist, war es notwendig ein Programm zu schreiben, welches diese statistischen Schwankungen und Ausreißer gewissermaßen glättet. Dies geschieht, indem ein Spline so an die Messpunkte gelegt wird, sodass die Krümmung der Kurve minimiert wird. Dies geschieht, indem die Fehlertoleranz (also die Fehlerbalken) der Messung mit in Betracht gezogen wird. Dadurch werden statistische Schwankungen der Messdaten oder Ausreißer hervorragend geglättet, die als physikalischen Effekte identifizierten Peaks, Schultern, Flanken etc. aber bleiben weitgehendst erhalten. Während der Arbeit mit der eigenen Entfaltungsroutine, wurde schnell klar, dass es wichtig ist, den ungestreuten Anteil der Neutronen von  $I(\theta)$  abzuziehen, da dieser Anteil ansonsten alle anderen physikalischen Effekte überschattet. Da man den Anteil der ungestreuten Neutronen  $p_{\uparrow\downarrow}$  jedoch nicht kennt, war es notwendig diesen zur korrekten Entfaltung abzuschätzen. Hier bot sich als Möglichkeit an, dass die Entfaltung nur dann gewisse Eigenschaften aufweist, wenn der korrekte Anteil von  $p_{\uparrow\downarrow}$  abgezogen wird. Somit wurde eine Möglichkeit gefunden um  $p_{\uparrow\downarrow}$  abschätzen zu können. Diese so bereinigte und optimierte Messkurve kann nun durch den ebenso selbst verfassten Entfaltungsalgorithmus mit der Gerätekurve entfaltet werden, um  $P(\theta)$  zu erhalten. Weiters konnten qualitative Aussagen über die Messproben getroffen werden. So stellte sich zum Beispiel heraus, dass die Größenverteilung der Streuer der „ $\alpha$ -Probe“ sehr schmal ist, also die Größenunterschiede der Streuer nur gering ist, da Simulationen zeigen, dass bei einer breiteren Größenverteilung die Maxima und Minima der Funktion  $P(\theta)$  mehr verschwimmen müssten.

Zusammenfassend lässt sich also sagen, dass der eingeschlagene Weg der Modellunabhängigkeit sehr vielversprechend wirkt, da einerseits glaubwürdige Aussagen über die Proben gemacht werden konnten und andererseits gleichzeitig während des Entstehens dieser Arbeit in T. Rechbergers Diplomarbeit [1] eine ebenfalls modellunabhängige Methode entwickelt wurde, es wurden jedoch komplett unterschiedliche Lösungen für oben genannten Probleme ge-

funden. Die Übereinstimmungen der Ergebnisse aus dieser Arbeit mit jenen T. Rechbergers steigern die Vertrauenswürdigkeit der Methodik.

Jedoch ist natürlich auch zu erwähnen, dass diese Arbeit nur die Machbarkeit einer modellunabhängigen Datenanalyse aufzeigen soll, es somit einige Punkte gibt, an denen man noch wesentliche Optimierungsarbeit leisten kann. So ist es zum Beispiel so, dass in den 3 Hauptprogrammen, welche verwendet werden manche Punkte zwar funktionieren und gute Ergebnisse liefern, jedoch verbessert werden können, um Laufzeit, Flexibilität und Genauigkeit zu optimieren. So kann man im Spline-Algorithmus einerseits die Randbedingungen für den Spline noch versuchen zu optimieren, andererseits ist es sicher auch noch möglich die Abweichungen des Splines von den Messpunkten um einiges feiner einstellen. Letzter Punkt könnte durch die Beschaffenheit des Codes unter anderem gelöst werden, indem man sich eine bessere Abbruchbedingung für die Iterationsschleife überlegt - diese läuft bis jetzt einfach nur eine gewisse Anzahl an Durchläufen. Weiters liefert auch der Trennungsalgorithmus zur Trennung der Doppelpeak-Rockingkurve in zwei Singlepeak-Rockingkurven zwar gute Ergebnisse, jedoch könnte auch dieser Code optimiert werden, um seine Flexibilität zu erweitern. Da diese Ungenauigkeit leider nicht gleichmäßig im Wertebereich beider Funktionen verteilt ist, sondern sich ziemlich genau in der Mitte akkumuliert, entsteht so abhängig von der Größe des gemachten Fehlers ein mehr oder weniger leichtes Rauschen beim Abziehen des ungestreuten Untergrundes, da bei der Entfaltung (siehe Gleichung (6.3))  $r_{\uparrow}(\theta)$ ,  $r_{\downarrow}(\theta)$  und  $R(\theta)$  gleichzeitig auftauchen und entfaltet werden.

Die Routine für die Entfaltung ist jener Teil des Codes, welcher die meiste Laufzeit in Anspruch nimmt. Hier könnte eine Optimierung der Geschwindigkeit vorgenommen werden.

Zur Analyse der Daten wurde es notwendig, die Doppelpeak-Rockingkurve mittels Trennungsalgorithmus in zwei SinglePeak-Rockingkurven für die beiden Spinkomponenten zu trennen. Die gemachte Annahme dabei ist, dass die verschmierenden Eigenschaften der Messapparatur auf Spin- $|\uparrow\rangle$ - oder Spin- $|\downarrow\rangle$ -Neutronen gleich wirkt. Somit sollten auch die jeweiligen  $|\uparrow\rangle$ - oder  $|\downarrow\rangle$ -Rockingkurven exakt gleich aussehen. Diese Annahme wurde jedoch nie durch Experimente untermauert und ist eine reine Annahme. Somit wäre es natürlich von Interesse diese Singlepeak-Rockingkurven auch wirklich zu messen. Dies könnte entweder dadurch geschehen, dass vor die Apparatur ein Superspiegel oder ein Neutronenresonator (siehe [18]) in den Strahlengang eingeführt wird, der nur mehr eine Spinkomponente passieren lässt. Oder man verstärkt das magnetische Feld der magnetischen Prismen so, dass die Aufspaltung vergrößert wird, und die Spinkomponenten dadurch auch besser voneinander getrennt sichtbar werden.

Weiters wäre natürlich eine bessere Statistik wünschenswert. Bei einer Messdauer von ungefähr 3 Stunden pro Kurve ist dies durch eine verlängerte Messdauer jedoch nur mehr mit unverhältnismäßigem Aufwand möglich. Bei der Messdatenauswertung hat sich jedoch herausgestellt, dass die Datenpunkte im Zentrum sehr dicht liegen, an den Außenbereichen der Kurve jedoch nur verhältnismäßig wenige Punkte gemessen wurden, und diese durch ihre niedrigere Zählrate natürlich auch eine größere Ungenauigkeit haben. Hier wäre eine Umverteilung der Messzeit vom Zentrum der Kurve nach außen hin wünschenswert, da die Schultern der Kurven eine nicht zu vernachlässigende Auswirkung auf die Ergebnisse haben.

# Anhang A

## Quellcode

Die im Zuge dieser Diplomarbeit entstandenen MATLAB-Routinen zur Datenanalyse sind auf den folgenden Seiten zu finden. Die Routine *Gaussinator.m* erzeugt gaußförmige Rockingkurven und Messkurven mit optionalem Rauschterm, mit welchen man die anderen Routinen auf Funktionsfähigkeit überprüfen kann. *Austrofred.m* ist das selbstgeschriebene Fittingtool, mit welchem ein Spline an die Messpunkte gelegt werden kann, wobei der gegebene Messfehler ein Maß für den maximal erlaubten Abstand zwischen Messpunkt und Spline ist.

*discretizer.m* ist eine einfache Routine, welche aus den Messpunkten mit beliebigem Abstand mittels der vorher erzeugten Splines äquidistante Punkte erstellt. *TrenntR* ist der Trennungsalgorithmus für die Rockingkurve. Hierbei wird versucht aus einer Doppelrockingkurve  $R(\theta)$  zwei exakt gleiche Singlerockingfunktionen  $r(\theta)$  zu erzeugen. Dieses Problem ist jedoch nur näherungsweise lösbar. *MyDeconv.m* ist der selbstgeschriebene Entfaltungsalgorithmus, welcher versucht die physikalische Streukurve aus  $I(\theta)$  und  $R(\theta)$  zu berechnen.

*MyMain.m* ist das Hauptprogramm, welches alle voran erklärten Subroutinen nacheinander aufruft. *MySuperMain* wiederum ruft *MyMain.m* mit unterschiedlichen Werten für die geschätzte Streuwahrscheinlichkeit  $p_{\uparrow\downarrow}$  auf, um somit eine Abschätzung für diesen Parameter zu finden.

Listing A.1: MySuperMain.m

```

5  time1=cputime;

7  headerlinesIn = 0;
8  delimiterIn = '\t';
9  clc
10 %A0B0:
11 datafileA0B0 = importdata('C:\Users\MINT\Desktop\Messdaten_Grenoble\
    A0B0S0_22Sep1142_Qcvt.dat', delimiterIn, headerlinesIn);
12 %A0B60
13 datafileA0B60 = importdata('C:\Users\MINT\Desktop\Messdaten_Grenoble\neu
    \A0B60S0_22Sep0552_Qcvt.dat', delimiterIn, headerlinesIn);
14 %A30B0:
15 datafileA30B0 = importdata('C:\Users\MINT\Desktop\Messdaten_Grenoble\neu
    \A30B0S0_23Sep0650_Qcvt.dat', delimiterIn, headerlinesIn);
16 %A45B0
17 datafileA45B0 = importdata('C:\Users\MINT\Desktop\Messdaten_Grenoble\neu
    \A45B0S0_23Sep1209_Qcvt.dat', delimiterIn, headerlinesIn);
18 %A60B0
19 datafileA60B0 = importdata('C:\Users\MINT\Desktop\Messdaten_Grenoble\neu
    \A60B0S0_23Sep1702_Qcvt.dat', delimiterIn, headerlinesIn);
20 %A90B0:
21 datafileA90B0 = importdata('C:\Users\MINT\Desktop\Messdaten_Grenoble\neu
    \A90B0S0_22Sep1634_Qcvt.dat', delimiterIn, headerlinesIn);
22 %A90B60
23 datafileA90B60 = importdata('C:\Users\MINT\Desktop\Messdaten_Grenoble\
    neu\A90B60S0_19Sep0459_Qcvt.dat', delimiterIn, headerlinesIn);

26 rockingfile4 = importdata('C:\Users\MINT\Desktop\Messdaten_Grenoble\
    prock16Sep0404_Qcvt.dat', delimiterIn, headerlinesIn);

29 repetitions=100000;
30 steps=500;
31 zugaben=1;

33 %XXX

35 datafile=datafileA60B0;
36 rockingfile=rockingfile4;
37 %XXX

39 lengthD=size(datafile,1); %Length of arrays: 80 oder 99 (?)
40 lengthR=size(rockingfile,1); %Length of arrays: 80 oder 99 (?)

```

```

41 DAngle=datafile (1:lengthD ,1);
42 RAngle=rockingfile (1:lengthR ,1);

44 newDCtsArray=zeros (lengthD , zugaben+1);
45 newRCtsArray=zeros (lengthR , zugaben+1);
46 newDCtsArray (: ,1)=datafile (1:lengthD ,1);
47 newRCtsArray (: ,1)=rockingfile (1:lengthR ,1);

49 %OPTIMIEREN
50 % [ AustrofredD]=Austrofred (datafile , repetitions , zugaben);
51 % [ AustrofredR]=Austrofred (rockingfile , repetitions , zugaben);
52 %
53 % AustrofredDBackup=AustrofredD;
54 % AustrofredRBackup=AustrofredR;

56 AustrofredD=AustrofredDBackup;
57 AustrofredR=AustrofredRBackup;

59 p_down=0;
60   for p_up=0.0:0.1:0.4
61   for p_down=0.0:0.1:0.4
62       MyMain(p_up,p_down,AustrofredD ,AustrofredR ,datafile ,rockingfile)
63           ;
64   end
65   end

66 time2=cputime;
67 time=time2-time1

```

## Listing A.2: MyMain.m

```

5  function []=MyMain(p_up, p_down, AustrofredD , AustrofredR , datafile ,
      rockingfile )
7  time1=cputime;

10 repetitions=100000;
11 steps=500;
12 zugaben=1;

14 AngleStepR= ( abs(rockingfile(1,1)) + abs(rockingfile(length(rockingfile)
      ),1)) ) /steps;
15 AngleStepD= ( abs(datafile(1,1)) + abs(datafile(length(datafile),1)) ) /
      steps;

19 lengthD=length(AustrofredD); %Length of arrays: 80 oder 99 (?)
20 lengthR=length(AustrofredR); %Length of arrays: 80 oder 99 (?)
21 % DAngle=datafile(1:lengthD,1);
22 % RAngle=rockingfile(1:lengthR,1);

24 newDCtsArray=zeros(lengthD , zugaben+1);
25 newRCtsArray=zeros(lengthR , zugaben+1);
26 newDCtsArray(:,1)=datafile(1:lengthD,1);
27 newRCtsArray(:,1)=rockingfile(1:lengthR,1);

29 % OPTIMIEREN
30 % [AustrofredD]=Austrofred(datafile , repetitions , zugaben);
31 % [AustrofredR]=Austrofred(rockingfile , repetitions , zugaben);
32 %
33 % %SICHERUNG ERSTELLEN
34 % AustrofredDBackup=AustrofredD;
35 % AustrofredRBackup=AustrofredR;
36 %
37 % %AUS SICHERUNG WIEDERHERSTELLEN
38 % AustrofredD=AustrofredDBackup;
39 % AustrofredR=AustrofredRBackup;

43 for i=1:zugaben
44     newDCtsArray(:,i+1)=AustrofredD(:,i);
45     newRCtsArray(:,i+1)=AustrofredR(:,i);

```

```

46 end

48 %DISKRETISIEREN
49 [discreteData, discreteRock]= discretizer (newDCtsArray, newRCtsArray, steps)
    ;

51 %Gausstestfunktionen statt Messdaten erstellen
52 % discreteData=Gaussinator(0);
53 % discreteRock=Gaussinator(1);

57 % TRENNUNG von up und down- Rockingkurve:
58 [RockU, RockD]=TrenntR(discreteRock);

61 % Definition der x-Achsen
62 for i=1:length(discreteData)
63     RockU(i,1)=rockingfile(1,1)+(i-1)*AngleStepR;
64     RockD(i,1)=rockingfile(1,1)+(i-1)*AngleStepD;
65     RockC(i,1)=rockingfile(1,1)+(i-1)*AngleStepD;
66     discreteRock(i,1)=rockingfile(1,1)+(i-1)*AngleStepR;
67     discreteData(i,1)=datafile(1,1)+(i-1)*AngleStepD;

69     %     RockU(i,1)=i;
70     %     RockD(i,1)=i;
71     %     RockC(i,1)=i;
72     %     discreteRock(i,1)=i;
73     %     discreteData(i,1)=i;

75 end

78 %Shift von RockU, sodass Maximum zentral liegt
79 [C, RockUmax]=max(RockU(:,2));
80 [C, RockDmax]=max(RockD(:,2));
81 % RockU(:,2)=circshift(RockU(:,2),-1);
82 RockC(:,2)=circshift(RockU(:,2),length(RockU(:,2))/2-RockUmax);
83 % RockD(:,2)=circshift(RockU(:,2),(length(RockU(:,2))/2-RockUmax)*2-2);
84 % RockU(:,2)=circshift(RockU(:,2),-1);
85 % RockD(:,2)=circshift(RockD(:,2),1);

87 testarray(:,2)=discreteRock(:,2)-RockU(:,2)-RockD(:,2);

91 % %NORMIERUNG von RockC (für Entfaltung):
92 normC=sum(RockC(:,2));
93 RockC(:,2)=RockC(:,2)/normC;

```

```

95 [C, discreteUmax]=max(discreteData(1:250,2))
96 [C, discreteDmax]=max(discreteData(250:500,2))
97 [C,Umax]=max(RockU(:,2))
98 [C,Dmax]=max(RockD(:,2))
99 [C,Cmax]=max(RockC(:,2))

103 % clf
104 %     hold on
105 %     set(gca,'fontsize',16,'FontName','arial');
106 %     xlabel('Q_y [A^-1]');
107 %     ylabel('Intensität','rot',90);
108 % %     axis([100 400 -1 5]),
109 % %     plot(discretePhysics(:,1),discretePhysics(:,2),'Color',[1 0
110 %           0],'LineWidth',0.8);
110 % plot(discreteRock(:,1),discreteRock(:,2),'Color',[0 0.6 0])
111 % % plot(discreteData(:,1),discreteData(:,2),'Color',[0 0 1]);
112 % plot(RockU(:,1),RockU(:,2)+RockD(:,2),'Color',[0 0 0])
113 % plot(RockU(:,1),RockU(:,2),'Color',[1 0 0])
114 % plot(RockD(:,1),RockD(:,2),'Color',[0 0 1])
115 % plot(RockU(:,1),testarray(:,2))
116 % plot(RockC(:,1),RockC(:,2),'Color',[0 0.6 0])
117 %     legend(sprintf('P(Q_y), \n pu=%.2f \n pd=%.2f',p_up,p_down));
118 %     hold off
119 % pause(1)

121 %FALTUNG
122 % discreteDataConv(:,2)=conv(discreteData(:,2),RockC(:,2));
123 % for i=1:length(discreteDataConv)
124 %     discreteDataConv(i,1)=rockingfile(1,1)-250*AngleStepD+(i-1)*
125 %     AngleStepD;
125 % end

129 %nur zum Testen: bauen einer Messintensitt I mit ungestreuten Anteil, "
130 %man nicht kennt", und rasifnden möchte:

132 % k_up=0; %Streuwahrscheinlichkeit up
133 % k=0.3;
134 % for i=1:length(RockU)
135 %     discreteData(i,2)=discreteData(i,2)+(1-k)*discreteRock(i,2);
136 %     discreteDataConv(i+250,2)=discreteDataConv(i+250,2)+(1-k)*
137 %     discreteRock(i,2);
137 % end
138 %

```

```

139 % k_down=0.4; %Streuwahrscheinlichkeit up
140 % for i=1:length(RockD)
141 % % discreteData(i,2)=discreteData(i,2)+(1-k)*discreteRock(i,2);
142 % discreteDataConv(i+250,2)=discreteDataConv(i+250,2)+(1-k_down)*
    RockD(i,2);
143 % end

148 %Num möchte man rausfinden wieviel Rocking dazuaddiert wurde (also das k
149 %finden), indem man immer ein bisschen mehr R abzieht, und schaut ob Kurve
150 %dadurch smoother wird.
151 discreteDataReal=discreteData;
152 for i=1:length(RockU)
153     %für Gauss, weil Rocking scheinbar kürzer als data:
154 % discreteDataConv(i+250,2)=discreteDataConv(i+250,2)-(1-p_up)*
    discreteRock(i,2);
155 % für Messdaten, weil Arrays gleich lang sind.
156 % discreteData(i,2)=discreteData(i,2)-(1-p_up)*discreteRock(i,2);
157 % discreteData(i,2)=discreteData(i,2)-(1-p_up)*RockU(i,2)-(1-p_up)*RockD
    (i,2);
158 discreteData(i,2)=discreteData(i,2)-(1-p_up)*RockU(i,2)-(1-p_down)*RockD
    (i,2);
159 % discreteDataConv(i+250,2)=discreteDataConv(i+250,2)-(1-p_up)*RockU
    (i,2);
160 end

163 % [discretePhysics]=MyDeconv(discreteData(:,2),RockC(:,2));
164 % [discretePhysics]=MyDeconv(discreteData(:,2),discreteRock(:,2));
165 [discretePhysics]=MyDeconv(discreteData(:,2),RockC(:,2));
166 for i=1:length(discretePhysics)
167 % discretePhysics(i,1)=i;
168 discretePhysics(i,1)=rockingfile(1,1)+(i-1)*AngleStepD;
169 end

171 % recU=conv(discretePhysics(:,2),RockU(:,2));
172 % recD=conv(discretePhysicsD(:,2),RockD(:,2));
173 % reconvU=zeros(length(recU),2);
174 % reconvD=zeros(length(recD),2);
175 % for i=1:length(reconvU)
176 % reconvU(i,1)=i;
177 % reconvD(i,1)=i;
178 % end
179 % reconvU(:,2)=recU;
180 % reconvD(:,2)=recD;

```

```

183 clf
184 hold on
185 subplot(2,2,1),
186     hold on
187     set(gca,'fontsize',16,'FontName','arial');
188     xlabel('Q_y [A^-1]');
189     ylabel('Intensität','rot',90);
190 %     plot(discreteData1(:,1),discreteData1(:,2),'Color',[1 0 0]);
191     plot(discreteRock(:,1),discreteRock(:,2),'Color',[0 0 1]);
192     plot(discreteData(:,1),discreteData(:,2),'Color',[1 0 0]);
193     legend('R(Q_y)','P(Q_y)');
194     hold off

196 subplot(2,2,2),
197     hold on
198     xlabel('Q_y [A^-1]');
199     set(gca,'fontsize',16,'FontName','arial');
200     ylabel('Intensität','rot',90);
201     plot(RockU(:,1),RockU(:,2),'Color',[1 0 0],'LineWidth',1.3);
202     plot(RockD(:,1),RockD(:,2),'Color',[0 0 1],'LineWidth',1.3);
203     legend('r(Q_y)');
204     hold off
205 subplot(2,2,3),
206     hold on
207     set(gca,'fontsize',16,'FontName','arial');
208 %     xlim([-2.5e-3 2.5e-3])
209     xlabel('Q_y [A^-1]');
210     ylabel('Intensität','rot',90);
211     plot(discreteData(:,1),discreteData(:,2),'Color',[1 0 0]);
212 %     legend('Faltung I(\theta)');
213     legend('I''(Q_y)');
214 %     legend('I(Q_y)-qR(Q_y)');
215 % legend('I(Q_y)=P(Q_y)*r(Q_y)');
216     hold off

218 subplot(2,2,4),
219     hold on
220     set(gca,'fontsize',16,'FontName','arial');
221 %     xlim([-2.5e-3 2.5e-3])
222 %     axis([0 500 0 20]),
223     xlabel('Q_y [A^-1]');
224     ylabel('Intensität','rot',90);
225     plot(discretePhysics(:,1),discretePhysics(:,2),'Color',[1 0 0],
226         'LineWidth',0.8);
226 %     plot(discreteData1(:,1),discreteData1(:,2),'Color',[0 0 1]);
227 %     legend(sprintf('P(Q_y) \n p=%2f',p));
228     legend(sprintf('P(Q_y) \n pu=%2f \n pd=%2f',p_up,p_down));
229 % legend('P''(Q_y)', 'P(Q_y)');
230 legend('P''(Q_y)');

```

```

231     hold off
232 hold off

234 saveas(gcf, sprintf('4_4Peak_A45B0_pu%.2fpd%.2f.jpg', p_up, p_down));
235 % saveas(gcf, sprintf('4_4Peak_k=03_p%.2f.jpg', p_up));
236 clf
237     hold on
238     set(gca, 'fontSize', 18, 'FontName', 'arial');
239     xlabel('Q_y [Å-1]');
240     ylabel('Intensität', 'rot', 90);
241 %     axis([0 500 0 20]),
242     plot(discretePhysics(:,1), discretePhysics(:,2), 'Color', [1 0 0], '
        LineWidth', 0.8);
243 %     plot(discreteData(:,1), discreteData(:,2), 'Color', [0 0 1]);
244 % plot(RockU(:,1), RockU(:,2), 'Color', [0 0 1])
245 % plot(RockD(:,1), RockD(:,2), 'Color', [1 0 0])
246 % plot(RockC(:,1), RockC(:,2), 'Color', [0 0.6 0])
247     legend(sprintf('P(Q_y), \n p_u=%.2f \n p_d=%.2f', p_up, p_down));
248 %     legend(sprintf('P(Q_y), p=%.2f', p_up));

250     hold off
251 saveas(gcf, sprintf('4Peak_A45B0_pu%.2fpd%.2f.jpg', p_up, p_down));
252 % saveas(gcf, sprintf('4Peak_k=03_p%.2f.jpg', p_up));

256 time15=cputime;
257 time=time15-time1

```

### Listing A.3: Spline-Algorithmus

```

1 function [newCtsArray]=Austrofred(datafile , repetitions , zugaben)
2 global z;

4 %Check if File hat 3 Spalten
5 if size(datafile ,2)~= 3
6     error('Eingelesene Datei hat keine 3 Spalten!');
7 end;
8 length=size(datafile ,1); %Length of arrays: 80 oder 99 (?)

11 zero=zeros(length ,1);
12 Angle=datafile(1:length ,1);
13 Counts=datafile(1:length ,2);
14 Err=datafile(1:length ,3);
15 newCtsArray=zeros(length , zugaben);

17 %for LOG10:
18 ErrU=zero;
19 ErrL=zero;
20 for i=1:length
21     ErrU(i)=log10(Counts(i)+Err(i))-log10(Counts(i));
22     ErrL(i)=log10(Counts(i))-log10(Counts(i)-Err(i));
23     Err(i)=log10(Counts(i)+Err(i))-log10(Counts(i));
24 end;
25 Counts=log10(datafile(1:length ,2));

29 for k=1:zugaben
30     newCts=Counts;
31     roughnessold=1000000000;
32     counter=0;
33     gradient=zeros((length) ,1);           %"1. Ableitung"
34     localroughness=zeros(length ,1);       %"2. Ableitung"
35     oldgradient=zeros((length) ,1);

37     %Einmal Am Anfang, um Positiven oder Negativen Ast der
        Gaussverteilung
38     %auswählen zu können
39     %Steigungen berechnen:
40     for i=1:(length-1)
41         gradient(i)=(newCts(i+1)-newCts(i))/(Angle(i+1)-Angle(i));
42     end
43     %ABS() im Nenner, weil erster und letzter Winkel unterschiedliche
44     %Vorzeichen haben!
45     gradient(length)=(newCts(1)-newCts(length))/abs(Angle(1)-Angle(
        length));
46     oldgradient=gradient;

```

```

49 %Krümmungen berechnen:
50 roughness=0;
51 for i=2:(length-1)
52     localroughness(i)=gradient(i)-gradient(i-1);
53     roughness=roughness+abs(localroughness(i));
54 end
55 %Krümmung an den Randpunkten wird über periodische Randbedingungen
56 %berechnet. Dadurch hat der 1. Punkt eine Steigung auf der linken
    Seite!
57 %     localroughness(1)=gradient(1)-gradient(length);
58 %     localroughness(length)=gradient(length)-gradient(length-1);

60 %Testen neuer RB:
61     localroughness(1)=gradient(1);
62     localroughness(length)=-gradient(length-1);
63     roughness=roughness+abs(localroughness(1))+abs(localroughness(
        length));

68 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
69     for j=1:repetitions

71         myrandomint=randi(length);

73         %geht Cts von links nach rechts durch indem mod() verwendet wird
            . wenn
74         %mod() 0,1,length ergibt wird eine randomzahl verwendet.
75         %scheint aber auch nicht besser zu sein.
76         %     myrandomint=mod(j,25);
77         %     if myrandomint==0 || myrandomint==1 || myrandomint ==
            length
78         %     myrandomint=randi(length);
79         %     while myrandomint==1 || myrandomint == length
80         %         myrandomint=randi(length);
81         %     end
82         %     end

84         oldCts=newCts(myrandomint);

86         if localroughness(myrandomint)>0
87             newCts(myrandomint)=newCts(myrandomint)+abs(random('norm',0,
                Err(myrandomint)^2));
88         %     newCts(myrandomint)=random('unif',newCts(myrandomint)-
            Err(myrandomint)^2,newCts(myrandomint)+Err(myrandomint)^2);
89         elseif localroughness(myrandomint)<=0

```

```

90         newCts(myrandomint)=newCts(myrandomint)-abs(random('norm',0,
91             Err(myrandomint)^2));

92     end

93
94     %Steigungen berechnen:
95     for i=1:(length-1)
96         gradient(i)=(newCts(i+1)-newCts(i))/(Angle(i+1)-Angle(i));
97     end
98     gradient(length)=(newCts(1)-newCts(length))/abs(Angle(1)-Angle
99         (2));
100    % Abstand zwsichen 1. und letztem Punkt bei periodieschen RB ist
101    % nicht wirklich definiert...

102    %Krümmungen berechnen:
103    roughness=0;
104    for i=2:(length-1)
105        localroughness(i)=gradient(i)-gradient(i-1);
106        roughness=roughness+abs(localroughness(i));
107    end
108    %alte RB:
109    %    localroughness(1)=gradient(1)-gradient(length);
110    %    localroughness(length)=gradient(length)-gradient(length-1);
111    %Testen neuer RB:
112    localroughness(1)=gradient(1);
113    localroughness(length)=-gradient(length-1);

114
115    roughness=roughness+abs(localroughness(1))+abs(localroughness(
116        length));

117
118    roughnessarray(j)=roughness;
119    % 2. und 3. Bedingung machen Steigung

120
121
122    %Hier werden Gradienten am Rand immer mit alten Gradienten
123    %verglichen. Neue Gradienten müssen links immer größer, rechts
124    %immer kleiner als alte sein. Dadurch bewegens ich gradienten in
125    %richtigeRichtung. Sobald Gradienten größer/kleiner als 0 sind,
126    %wird
127    %oldgradient 0 gesetzt, und Gradienten können jeden Wert größer
128    % (links) respektive kleiner (rechts) als 0 annehmen.

129
130    if (oldgradient(1)>0)
131        oldgradient(1)=0;
132    end
133    if (oldgradient(length-1)<0)
134    %        oldgradient(length-1)=0;
135    end

```

```

137 %         if (roughness<roughnessold) && (gradient(1)>=oldgradient(1))
&& (gradient(length-1)<=oldgradient(length-1))
138         if (roughness<roughnessold) && (gradient(1)>=oldgradient(1))

140             roughnessold=roughness;
141             %newCts(myrandomint)=oldCts;
142             counter=counter+1;
143         else
144             newCts(myrandomint)=oldCts;
145         end
146 %%%%%%%%%%
147 end %repetitionschleife
148 %%%%%%%%%%

151 %check how many newCts are inside Errorbar or not
152 insidebar=0;
153 inside2sigma=0;
154 for i=2:(length-1)
155     if (~((newCts(i)< (Counts(i)- ErrL(i))) || (newCts(i) > (Counts(
i)+ ErrU(i))))&&(j==repetitions))
156         insidebar=insidebar+1;
157     end
158     if (~((newCts(i)< (Counts(i)- 2*ErrL(i))) || (newCts(i) > (
Counts(i)+ 2*ErrU(i))))&&(j==repetitions))
159         inside2sigma=inside2sigma+1;
160     end
161 end
162 insidebar=insidebar/length;
163 inside2sigma=inside2sigma/length;

165     %z=(x-mu)/sigma:
166     z=zero;
167     for i=1:length
168         z(i)=(newCts(i)-Counts(i))/Err(i);
169     end
170     z=abs(z);

178 % end

181 %Array in dem alle Splines der Zugaben gespeichert sind

```

```

182 % splineArray(k)=spline(Angle,newCts);
184 %Array (eigentlich MAtrix!) in dem alle newCts der Zugaben
    gespeichert
185 %sind

189 % %ENTLOGARITHMIEREN
190     for i=1:length
191         newCtsArray(i,k)=10^newCts(i);
192     end

196 %%%%%%%%%%%
197 end %Zugabenschleife
198 %%%%%%%%%%%

204 % counter=counter/repetitions
205 insidebar
206 inside2sigma
207 % counter

209 % clf
210 % hold on
211 % hist(z,10)
212 % hold off

```

Listing A.4: discretizer.m

```

3 function [discreteData , discreteRock]= discretizer (newDCtsArray ,
    newRCtsArray , steps)
4 %Diese Funktion nimmt optimierteDaten aus Austrofred , und
5 % :: diskretisiert die Winkel und Daten
6 % :: zieht den Untergrund von der Funktion ab

9 zugaben=size (newRCtsArray ,2) -1;
10 lengthR=length (newRCtsArray) ;
11 lengthD=length (newDCtsArray) ;

13 %linkester und rechtester Punkt werden definiert :
14 %die beiden inneren Punkte werden genommen , da dort garantiert auch die
15 %andere Funktion schon Messdaten hat .
16 if newRCtsArray (1 ,1)<=newDCtsArray (1 ,1)
17     lowestAngle=newDCtsArray (1) ;
18 else
19     lowestAngle=newRCtsArray (1 ,1) ;
20 end

22 if newRCtsArray (lengthR ,1)<=newDCtsArray (lengthD ,1)
23     highestAngle=newRCtsArray (lengthR ,1) ;
24 else
25     highestAngle=newDCtsArray (lengthD ,1) ;
26 end

29 % mit Diskretisierung beginnen : zuerst werden die Winkel diskretisiert .
30 discreteRock=zeros ( steps , zugaben+1) ;
31 discreteData=zeros ( steps , zugaben+1) ;
32 xlength=highestAngle -lowestAngle ;
33 stepsize=xlength /steps ;

35 for i =1:steps
36     discreteData ( i ,1)=lowestAngle+stepsize *( i -1) ;
37     discreteRock ( i ,1)=lowestAngle+stepsize *( i -1) ;
38 end

40 for j =2:( zugaben+1)
41     %LOGARITHMIEREN
42     for i =1:lengthR
43         newRCtsArray ( i , j)=log10 (newRCtsArray ( i , j) ) ;
44     end
45     for i =1:lengthD
46         newDCtsArray ( i , j)=log10 (newDCtsArray ( i , j) ) ;
47     end

```

```

49     %Spline benötigt logarithmierte Daten, da Messpunkte sonst zu steil
50     %sind, und er zum Schwingen anfängt.
51     for i=1:steps
52         discreteRock(i,j)=spline(newRCtsArray(:,1),newRCtsArray(:,j),
53             discreteRock(i,1));
54         discreteData(i,j)=spline(newDCtsArray(:,1),newDCtsArray(:,j),
55             discreteData(i,1));
56     end
57 end

58 %Untergrund berechnen, und von Kurve abziehen:
59 %Untergrund ist einfach der kleinste Wert der jeweiligen Kurve.
60 backgroundR=zeros(zugaben+1,1);
61 backgroundD=zeros(zugaben+1,1);
62 for i=2:zugaben+1
63     backgroundR(i)=min(discreteRock(:,i));
64     backgroundD(i)=min(discreteData(:,i));
65 end

66 for k=2:zugaben+1
67     for i=1:steps
68         discreteRock(i,k)=discreteRock(i,k)-backgroundR(k);
69         discreteData(i,k)=discreteData(i,k)-backgroundD(k);
70     end
71 end
72 end

```

```

4 %Entfaltungsalgorithmus a la Diss Hainbucher
5 function [discretePhysics]=MyDeconv(discreteData1 ,discreteRock1)
6 discreteRock=zeros(length(discreteRock1),2);    discreteRock(:,2)=
    discreteRock1;
7 discreteData=zeros(length(discreteData1),2);    discreteData(:,2)=
    discreteData1;

10 %UNTERGRUND wird links und rechts abgezogen, sodass keine Stufen an den
11 %Rändern entstehen. Dadurch werden aber alle Messunkte um Untergrund
    nach
12 %unten verschoben. Um diesen selbst verursachten Fehler zumindest
    teilweise
13 %auszubessern, werden Messdaten wiederum mit Faktor >1 multipliziert,
14 %sodass sie wieder um etwas erhöht werden.
15 % maxL=max(discreteData1(1:250));
16 % maxR=max(discreteData1(251:length(discreteData1)));
17 maxL=discreteData1(250);
18 maxR=discreteData1(251);
19 undergroundL=min(discreteData1(1:250));
20 undergroundR=min(discreteData1(251:length(discreteData1)));
21 Pl=maxL/(maxL-undergroundL);
22 Pr=maxR/(maxR-undergroundR);
23 for i=1:250
24     discreteData1(i)=(discreteData1(i)-undergroundL)*Pl;
25 end
26 for i=251:length(discreteData1)
27     discreteData1(i)=(discreteData1(i)-undergroundR)*Pr;
28 end

31 %PADDING:
32 %Falls Data und Rocking gleich lang sind
33 %(so wie es bei Messdaten der Fall ist)
34 if ( (length(discreteRock1)) == (length(discreteData1)) )
35     [C,RockMax]=max(discreteRock(:,2));
36     discreteData=zeros((length(discreteData1)*2-1),2);
37     for i=1:(length(discreteData1))
38         discreteData(i+RockMax,2)=discreteData1(i);
39     end

41     padding=1;
42 else
43     padding=2;
44 end

```

```

48 %NORMIERUNG:
49 norm=sum(discreteRock(:,2));
50 discreteRock(:,2)=discreteRock(:,2)/norm;

53 % Array für Vergleich von Messkurve zu geschätzter Faltung
54 discretePhysics=zeros(length(discreteRock),2);
55 newData=zeros(length(discreteData),2);
56 % X-Arrays werden gefüllt:
57 for i=1:length(newData)
58     newData(i,1)=i;
59     discreteData(i,1)=i;
60 end
61 for i=1:length(discretePhysics)
62     discretePhysics(i,1)=i;
63 end
64 for i=1:length(discreteRock)
65     discreteRock(i,1)=i;
66 end

71 lim=1;
72 upperlim=lim;
73 lowerlim=lim;
74 relaxer=1;

76 dummy=true;
77 counterU=0;
78 counterL=0;
79 i=1;
80 residuum=1000;
81 residuummin= 100000000;
82 residuumUmin=100000000;
83 residuumLmin=100000000;
84 [C,RockMax]=max(discreteRock(:,2));

88 delta=0.1;
89 for i=1:50000
90 % while (residuum < residuummin+delta)
91 % while (residuum > 10)

93     if mod(i,50000)==0
94         relaxer=relaxer*0.9;

```

```

95     end

97     %     pivot=randi( length( discreteRock ) );
98     pivot=mod( i , length( discreteRock ) );
99     if pivot ==0
100         pivot=length( discreteRock );
101     end

105     if ((newData( pivot+RockMax-1,2) ) < discreteData( pivot+RockMax-1,2) *
        lowerlim )
106         gap = discreteData( pivot+RockMax-1,2) - (newData( pivot+RockMax
            -1,2) );
107         for j=1:length( discreteRock )
108             newData( j+pivot -1,2)=newData( j+pivot -1,2)+relaxer*gap*(
                discreteRock( j ,2) );
109         end
110         discretePhysics( pivot ,2) = discretePhysics( pivot ,2) + relaxer*
            gap;
111         counterL=counterL+1;
112     %     end

114     else
115         gap = newData( pivot+RockMax-1,2) - discreteData( pivot+RockMax
            -1,2) ;
116         if ( discretePhysics( pivot ,2) -relaxer*gap >0 )
117             %     if ((newData( pivot+RockMax-1,2) > (discreteData( pivot+
                RockMax-1,2) )/upperlim ) ) && (discretePhysics( pivot ,2) -relaxer*
                gap >0 )
118             %     gap = newData( pivot+RockMax-1,2) - discreteData( pivot+
                RockMax-1,2) ;
119             for j=1:length( discreteRock )
120                 newData( j+pivot -1,2)=newData( j+pivot -1,2)-relaxer*gap*(
                    discreteRock( j ,2) );
121             end
122             discretePhysics( pivot ,2) = discretePhysics( pivot ,2) -relaxer
                *gap ;
123             counterU=counterU+1;
124         end
125     end

131     residuum=0;
132     residuumU=0;
133     residuumL=0;

```

```

134     for j=1:length(newData)
135         residuum=residuum+abs(newData(j,2)-discreteData(j,2));
136     end
137     if residuum<residuummin
138         residuummin=residuum;
139     end

141     if (mod(i,10000)==0) || (residuum > residuummin+delta)

143         for j=1:length(newData)
144             if newData(j,2) > discreteData(j,2)
145                 residuumU=residuumU+ abs(newData(j,2)-discreteData(j,2))
146                 ;
147             else
148                 residuumL=residuumL + abs(newData(j,2)-discreteData(j,2)
149                 );
150             end
151         end

152         if residuumU<residuumUmin
153             residuumUmin=residuumU;
154         end
155         if residuumL<residuumLmin
156             residuumLmin=residuumL;
157         end

158         clc
159         residuumU
160         residuumL
161         residuum
162         residuummin
163         counterL
164         counterU
165         i
166     end
167 end
168
169     IntData=sum(discreteData(:,2))
170     IntRock=sum(discreteRock(:,2))
171     IntPh=sum(discretePhysics(:,2))
172     IntnewData=sum(newData(:,2))

176 [myconv]=conv(discretePhysics(:,2), discreteRock(:,2));
177 reconv=zeros(length(myconv),2);
178 reconv(:,2)=myconv;
179 for i=1:length(reconv) [caption=MyDeconv.m]
180     reconv(i,1)=i;

```

```

181 end

184 % clf
185 % hold on
186 %     subplot(2,2,1),
187 %         line(discreteData(:,1),discreteData(:,2),'Color',[1 0 1],'
           LineWidth',1)
188 %         line(newData(:,1),newData(:,2),'LineWidth',1)
189 % %         axis([-Inf Inf -0.2 2]),
190 %         legend('dData', 'newData');
191 %     subplot(2,2,2),
192 %         line(discreteData(:,1),discreteData(:,2),'Color',[1 0 1])
193 % %         line(discreteRock(:,1)+RockMax,discreteRock1)
194 % %         line(reconv(:,1),reconv(:,2))
195 %         line(discreteRock(:,1),discreteRock1)
196 % %         axis([-Inf Inf -0.2 70]),
197 %         legend('dData','dRock');
198 %     subplot(2,2,3),
199 %         line(newData(:,1),newData(:,2))
200 % %         axis([-Inf Inf -0.2 2]),
201 %         legend('nData');
202 %     subplot(2,2,4),
203 %     grid on
204 %
205 %         line(discretePhysics(:,1),discretePhysics(:,2))
206 % %         axis([-Inf Inf -0.2 70]),
207 %         legend('MyDeconv');
208 % hold off

```

### Listing A.5: Gaussinator.b

```

1  % Funktion , welche ein Array einer bestimmten Länge erstellen soll , in
    welchem
2  % sich eine DoppelGaussfunktion befindet .
3  % Zusatz: Verrauschter Gauss
4  %Damit soll man die Codekette Austrofred+Trennung+MyDeconv testen , und
5  %schauen was rauskommt: weil wenn das gut ist , dann man Ergebnissen
6  %wahrscheinlich vertrauen .

9  function [GaussArray]=Gaussinator(RD)

11 headerlinesIn = 0;
12 delimiterIn = '\t';
13 datafile = importdata('C:\Users\MINT\Desktop\Messdaten_Grenoble\
    A0B0S0_22Sep1142_Qcvt.dat', delimiterIn , headerlinesIn );

15 rockingfile = importdata('C:\Users\MINT\Desktop\Messdaten_Grenoble\
    prock16Sep0404_Qcvt.dat', delimiterIn , headerlinesIn );

17 % Verschmrswitch:
18 smearing=0;
19 % RD=1;

22 % if RD ==0
23 % file=datafile;
24 % %Data:
25 %an Messdaten gefittete Funktion:
26 %      C1 =      1.559;      mu1 =  3.816e-005;      sigma1 =
    3.34e-005;
27 %      C2 =      1.624;      mu2 = -3.768e-005;      sigma2 =
    2.76e-005;
28 %      C3 =      0.2925;      mu3 = -8.425e-005;      sigma3 =
    2.763e-005;
29 %      C4 =      1.603;      mu4 =  3.888e-006;      sigma4 =
    0.0002145;
30 %      C5 =      1.655;      mu5 = -2.413e-005;      sigma5 =
    0.0008547;
31 % elseif RD ==1
32 %      file=rockingfile;
33 % %Rocking:
34 %      C1 =      3.028;      mu1 = -3.883e-005;      sigma1 =
    3.006e-005;
35 %      C2 =      3.231;      mu2 =  4.011e-005;      sigma2 =
    3.697e-005;
36 %      C3 =      0.01726;      mu3 =  0.0001261;      sigma3 =
    5.709e-006;

```

```

37 %      C4 =      0.3801;      mu4 = -8.437e-005;      sigma4 =
      4.676e-005;
38 %      C5 =      1.804;      mu5 = -3.301e-005;      sigma5 =
      0.0003733;
39 % else
40 %      error('na hoppala!');
41 % end

45 % RD=1;

48 % if RD ==0
49 % file=datafile;
50 % %Data:
51 % %willkürliche , einfache Funktion mit Ähnlichkeit zu Messdaten:
52 %      C1 =      0.8;      mu1 = 6e-005;      sigma1 = 5e-005;
53 %      C2 =      0.8;      mu2 = -6e-005;      sigma2 = 5e-005;
54 %      C3 =      0.2925;      mu3 = 0;      sigma3 = 6e-004;
55 %      C4 =      0;      mu4 = 3.888e-006;      sigma4 =
      0.0002145;
56 %      C5 =      0;      mu5 = -2.413e-005;      sigma5 =
      0.0008547;
57 % elseif RD ==1
58 %      file=rockingfile;
59 % %Rocking:
60 %      C1 =      0.8;      mu1 = 6e-005;      sigma1 = 4e-005;
61 %      C2 =      0.8;      mu2 = -6e-005;      sigma2 = 4e-005;
62 %      C3 =      0.2925;      mu3 = 0;      sigma3 = 3e-004;
63 %      C4 =      0;      mu4 = -8.437e-005;      sigma4 = 4.676e
      -005;
64 %      C5 =      0;      mu5 = -3.301e-005;      sigma5 =
      0.0003733;
65 % else
66 %      error('na hoppala!');
67 % end

71 % %Funktion für P mit 4 Peaks , wie man es theoretisch erwarten würde:
72 if RD ==0
73 file=datafile;
74 %Data:
75      C1 =      2.5;      mu1 = -12e-005;      sigma1 = 4e-005;
76      C2 =      2.5;      mu2 = 12e-005;      sigma2 = 4e-005;
77      C3 =      0.9;      mu3 = 0;      sigma3 = 6e-004;
78      C4 =      2.4;      mu4 = -4e-005;      sigma4 = 4e-005;
79      C5 =      2.4;      mu5 = 4e-005;      sigma5 = 4e-005;

```

```

80 elseif RD ==1
81     file=rockingfile;
82 %Rocking:
83     C1 =      3.0;      mu1 = -8.0e-005;      sigma1 = 3.0e-005;
84     C2 =      3.0;      mu2 =  8.0e-005;      sigma2 = 3.7e-005;
85     C3 =      0.0;      mu3 =  0.0001;      sigma3 = 5.7e-006;
86     C4 =      0.0;      mu4 = -8.4e-005;      sigma4 = 4.7e-005;
87     C5 =      1.8;      mu5 = 0;      sigma5 = 0.0003;
88 else
89     error('na hoppala!');
90 end

93 % points=length(file);
94 points=500;
95 miny=file(1,1);
96 maxy=file(length(file),1);
97 stepsize=(abs(miny)+maxy)/points;
98 GaussArray=zeros(points,2);

100 for i=1:points
101     GaussArray(i,1)=miny+(i-1)*stepsize;
102     x=miny+(i-1)*stepsize;
103 % GaussArray(:,1)=file(:,1);
104 % GaussArray(:,3)=file(:,3);
105     x=GaussArray(i,1);
106     GaussArray(i,2)=Gauss(C1,mu1,sigma1,x)+Gauss(C2,mu2,sigma2,x)+Gauss(
        C3,mu3,sigma3,x)+Gauss(C4,mu4,sigma4,x)+Gauss(C5,mu5,sigma5,x);
107     if smearing == 1
108         GaussArray(i,2)=GaussArray(i,2)+random('norm',0,(GaussArray(i,2)
            )/16);
109     end
110 end

112 %Achtung: Normierung auf 1 ergibt keinen Sinn, da Intensitäten!
113 % %NORMIERUNG:
114 % norm=sum(GaussArray(:,2))*(GaussArray(2,1)-GaussArray(1,1))
115 % GaussArray(:,2)=GaussArray(:,2)/norm;
116 % ppp=sum(GaussArray(:,2))*(GaussArray(2,1)-GaussArray(1,1))

120 function [x]=Gauss(C,mu,sigma,i)
121 x=C*exp(-((i-mu)/sigma)^2);

```

# Literaturverzeichnis

- [1] Tobias Rechberger: *Untersuchung von Ultrakleinwinkelstreuung an magnetischen Mikrostrukturen mit polarisierten Neutronen*. Diplomarbeit am Atominstytut der TU Wien, (2013)
- [2] A. Zdarzil: *Simulation von magnetischer Ultrakleinwinkelneutronenstreuung mittels Streudatensynthese*. Diplomarbeit am Atominstytut der TU Wien, (2013)
- [3] Martin Hainbuchner: *Ultra-Kleinwinkelstreuung von Neutronen an strukturierten Materialien*. Dissertationschrift. Atominstytut der TU Wien (2000)
- [4] Martin Trinker: *Neutronen-Kleinwinkelstreuung an mikro- und nanostrukturierten Materialien* Dissertationschrift. Atominstytut der TU Wien (2006)
- [5] E. Jericha et al.: *Towards a modelling of USANSPOL intensities from magnetic ribbons*. Physics Procedia **42** (2013) 58-65
- [6] E. Jericha et al.: *Characterisation of novel magnetic materials using USANSPOL technique*. Physica B **406** (2011) 2401-2404
- [7] E. Jericha et. al.: *Experimental and methodic progress in ultra-small-angle polarised neutron scattering on novel magnetic materials*. Journal of Physics: Conference Series **340** (2012) 012007
- [8] Christian Lang, Norbert Pucker: *Mathematische Methoden der Physik*. Springer Spektrum (2005) ISBN 3-8274-1558-6
- [9] Alice Redermeier: *Realisierung einer modernen "Bonse-Hart Kamera für Ultra-Kleinwinkelstreuung thermischer Neutronen* Diplomarbeit. Atominstytut der TU Wien (2011)

- [10] Rauch H. und Petraschek D: *Neutron Diffraction* (Topics in Current Physics vol. 6) (Berlin: Springer) chap. Dynamical Neutron Diffraction and Its Application, p. 303-351 (1978)
- [11] M. Villa et al.: *Optimization of a crystal design for Bonse-Hart camera* J. Appl. Cryst. (2003) **36**, 769-773
- [12] M. Agamalian et al.: *Optimization of a Bonse-Hart Ultra-Small-Angle Neutron Scattering Facility by Elimination of the Rocking-Curve-Wings.* Journal of Applied Crystallography **30** (1997) 345-352
- [13] W.H. Zachariasen: *A general Theory of X-Ray diffraction in crystals.* Acta Cryst. (1967) **23** 558-564
- [14] E. Jericha et al.: *Ultra-small-angle scattering with polarized neutrons.* Physica B **397** (2007) 88-90
- [15] W. Demtröder: *Experimentalphysik 1: Mechanik und Wärme.* Springer Verlag ISBN 3-540-26034-X
- [16] W. Demtröder: *Experimentalphysik 3: Elektrizität und Optik.* Springer Verlag ISBN 978-3-540-68210-3
- [17] W. Demtröder: *Experimentalphysik 3: Atome, Moleküle und Festkörper.* Springer Verlag ISBN 3-540-66790-3
- [18] C. Gösselsberger: *Entwicklung eines Wanderwellen-Neutronenspinresonators.* Dissertation. Atominstitut der TU Wien. (2012)
- [19] V.F. Seirs: *Neutron optics: An introduction to the theory of neutron optical phenomena and their applications* Oxford University Press (1989). ISBN 0195046013.
- [20] G. Badurek et al.: *Amorphous soft-magnetic ribbons studies by ultra-small-angle polarized neutron scattering.* Journal of Physics: Conference Series **211** (2011) 012027
- [21] G. Badurek et al.: *Permanent magnetic field-prism polarizer for perfect crystal neutron interferometers.* Physica B **283** (2000) 289-392
- [22] P. Debye et. al.: *Scattering by an Inhomogeneous Solid. II. The correlation Function and Its Application.* J. Applied Physics. **28** (1957) 679-682

- [23] W. Dahmen et al.: *Numerik für Ingenieure und Naturwissenschaftler*. Springer Verlag: ISBN 978-3-540-25544-4 (2006)
- [24] T. Rechberger: "Verbesserung einer USANS Probenumgebung und deren Ansteuerung". Projektarbeit am Atominstitut der TU Wien. (2012)