

Friedrich-Alexander Universität Erlangen-Nürnberg
Lehrstuhl für Elektronische Bauelemente
Prof. Dr. rer. nat. L. Frey



PROJEKTARBEIT

RECHNERUNTERSTÜTZTE AUSWERTUNG VON DIODENKENNLINIEN

Bearbeiter: Leander Sturm
Betreuer: Jochen Kaiser
Stefan Donath
Beginn: 09. November 2009
Abgabe: 22. Oktober 2010

Erlangen, den 20. Oktober 2010

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Leander Sturm

Zusammenfassung

Mittelpunkt der vorliegenden Arbeit ist die Entwicklung einer Anwendung zur rechnergestützten Auswertung von Diodenkennlinien, welche unabhängig vom Betriebssystem einsetzbar ist und ohne zusätzliche Software verwendet werden kann. Die Applikation soll es dem Anwender ermöglichen, bei geringem Zeitaufwand, die Eigenschaften einer Diode aus deren Kennlinie zu extrahieren, ohne dass hierfür tiefere Vorkenntnisse vonnöten sind. Im Zuge dieser Arbeit werden zunächst die theoretischen Grundlagen einer Halbleiterdiode erörtert und anschließend ein parametrisiertes Modell einer realen Halbleiterdiode erarbeitet.

Das erarbeitete Modell stellt die Grundlage für die rechnergestützte Auswertung von gemessenen Diodenkennlinien dar. Über die Parameter wird das mathematische Modell in die einzelnen Messpunkte eingepasst. Dieser Prozess – der als Kurvenanpassung bezeichnet wird – stellt ein weiteres zentrales Theoriekapitel dieser Arbeit dar. In diesem Zuge werden mit dem Levenberg-Marquardt- und Nelder-Mead-Algorithmus zwei Algorithmen vorgestellt, die sich für die Lösung des Problems der Kurvenanpassung nutzen lassen und in der entwickelten Applikation implementiert wurden.

Des Weiteren werden die Implementierungen der beiden Algorithmen auf ihre praktische Tauglichkeit geprüft und miteinander verglichen. Zusätzlich werden verschiedene Problemfelder der Kurvenanpassung, sowohl im Allgemeinen als auch im Speziellen für die Diodenkennlinie, erörtert und Lösungsansätze präsentiert. Abschließend liefert die Arbeit einen Überblick über die Struktur der entwickelten Applikation.

Abstract

This thesis focuses on the development of an application for computer aided evaluation of diode characteristics. The application is platform independent and usable without additional software. Furthermore, it assists the user with the quick extraction of the properties of a diode without requiring deeper knowledge of the physics of this semiconductor device.

The thesis discusses at first the principles of a semiconductor diode and then proceeds with the development of a parametrized model of a real semiconductor diode. This parametrized model is the fundament for the computer aided evaluation of measured diode characteristics. With the help of the parameters the mathematical model is fitted to the single measuring points. This process, which is called curve fitting, is another central theoretical chapter of this thesis. In this regard two different algorithms are presented, namely the Levenberg-Marquardt and Nelder-Mead-Algorithm. Both algorithms have been implemented in the developed application.

Furthermore, both implementations of the algorithms are discussed concerning their behavior in practice. Additionally, different problems, concerning the curve fitting process in general and the semiconductor diode in particular, are discussed and appropriate solutions are presented. Finally, the thesis gives an overview of the structure of the developed application.

Inhaltsverzeichnis

1. Einleitung	7
2. Halbleitergrundlagen	8
2.1. Bändermodell	8
2.2. Intrinsischer Halbleiter	10
2.3. Dotierter Halbleiter	10
2.4. Halbleiterdiode	12
2.4.1. Verhalten im thermodynamischem Gleichgewicht	12
2.4.2. Verhalten bei angelegter Spannung	13
2.4.3. Ideale Diode	14
2.4.4. Reale Diode	15
3. Kurvenanpassung	17
3.1. Optimierung	17
3.2. Kleinste-Quadrate-Problem	18
3.3. Nelder-Mead (Simplex-Downhill)	19
3.4. Liniensuche	23
3.4.1. Gradientenverfahren	23
3.4.2. Newton-Verfahren	23
3.4.3. Gauß-Newton-Verfahren	24
3.4.4. Levenberg-Marquardt-Verfahren	25
3.5. Problemfelder und Lösungen bei der praktischen Umsetzung	26
3.5.1. Reale Diode als implizite Funktion	26
3.5.2. Ableitung der Zielfunktion	27
3.5.3. Auffinden des globalen Minimums	28
3.5.4. Begrenzung der Parameter	29
3.5.5. Logarithmierte Kurvenanpassung	29
4. Praktische Auswertung von Diodenkennlinien	31
4.1. Vergleich von linearer und logarithmischer Kurvenanpassung	32
4.2. Vergleich von Levenberg-Marquardt- und Nelder-Mead-Algorithmus	36
4.3. Einfluss der Startparameter	39
4.4. Auswirkungen von Modellabweichungen	41
5. Fazit und Ausblick	43

A. Software-Aufbau und Funktionsweise der Applikation	45
A.1. UML-Modellierung	45
A.2. Erweiterbarkeit	46

1. Einleitung

Die Halbleitertechnik ist über die vergangenen Jahrzehnte hinweg zu einem der wichtigsten Bereiche in der Elektrotechnik herangewachsen. Halbleiter-Bauelemente und darauf basierende integrierte Schaltungen (ICs) sind mittlerweile fester Bestandteil der heutigen Welt und in einem Großteil technischer Geräten anzutreffen. Im Hinblick auf die voranschreitende Miniaturisierung wird der Halbleitertechnik auch zukünftig ein hoher Stellenwert zukommen.

Das einfachste Bauelement der Halbleitertechnik ist hierbei die Halbleiterdiode. Sie hat die Funktion elektrischen Strom lediglich in eine Betriebsrichtung passieren zu lassen und in die andere Betriebsrichtung isolierend zu wirken. Um diese Funktion zu Erfüllen, besteht die Halbleiterdiode im Wesentlichen schlicht aus zwei Gebieten mit unterschiedlicher Dotierung. Trotz seiner Simplizität ist dieses Bauelement im Halbleiterbereich von zentraler Bedeutung, da es die Basis für eine Vielzahl verschiedener anderer Bauteile darstellt. Aufgrund der tragenden Bedeutung bei gleichzeitig einfachem Aufbau spielt die Halbleiterdiode dementsprechend bei der Analyse von Fertigungsprozessen eine elementare Rolle.

Um mittels produzierter Halbleiterdioden Rückschlüsse auf den Fertigungsprozess zu ziehen, müssen deren wesentlichen charakteristischen Eigenschaften bestimmt werden. Hierfür muss die Halbleiterdiode zunächst in verschiedenen Betriebspunkten vermessen und die daraus gewonnene Kennlinie analysiert werden. Diesbezüglich ist in der Praxis eine Auswertung per Hand ebenso anzutreffen wie eine rechnergestützte Analyse der Halbleiterdiode. Durch die steigende Rechenleistung aktueller Computersysteme ist jedoch – gerade bei einer größeren Menge an Kennlinienauswertungen – eine rechnergestützte Auswertung wünschenswert. Ein weiterer Vorteil dieser Auswertungsart liegt in der höheren Genauigkeit.

Grundlage für die rechnergestützte Auswertung ist ein parametrisiertes mathematisches Modell einer realen Halbleiterdiode. Das Modell kann hierbei über seine Parameter in die gemessene Diodenkennlinie eingepasst werden, wobei die Parameter den spezifischen Eigenschaften der vermessenen Diode entsprechen. Ziel dieser Arbeit war es eine systemunabhängige Software-Applikation zu entwickeln, welche den Anwender bei der Auswertung von Diodenkennlinien unterstützt. Neben einem Modell für Dioden aus den Halbleitermaterialien Silicium und Germanium beinhaltet dies auch verschiedene Algorithmen, welche die Kurvenanpassung automatisiert vornehmen.

2. Halbleitergrundlagen

Dieses Kapitel soll zur Vermittlung der, für diese Arbeit relevanten, physikalischen Grundlagen eines Halbleiters im Allgemeinen, und der Halbleiterdioden im Speziellen, dienen. Um spätere Zusammenhänge erklären zu können, wird zunächst das Bändermodell betrachtet. Darauf aufbauend widmet sich dieses Kapitel dem reinen, intrinsischen Halbleiter, sowie dem dotierten Halbleiter, bei welchem Fremdatome in die ursprüngliche Kristallstruktur eingebracht werden. Der dotierte Halbleiter stellt die Basis für die Halbleiterdiode dar, die aus zwei verschiedenen Bereichen mit unterschiedlicher Dotierung besteht. In diesem Zuge wird abschließend die Strom-Spannungs-Beziehung der Halbleiterdiode erörtert.

2.1. Bändermodell

Grundsätzlich handelt es sich bei einem Halbleiter – wie der Name bereits andeutet – um einen Festkörper, dessen Leitfähigkeit sich in einem Bereich zwischen der eines Leiters und der eines Isolators bewegt. Die Leitfähigkeit der Halbleiter ist stark temperaturabhängig und nimmt mit steigender Temperatur zu. Abbildung 2.1 veranschaulicht die Leitfähigkeit von undotiertem Germanium und Silicium im direkten Vergleich mit dem Leiter Silber und dem Isolator Schwefel.

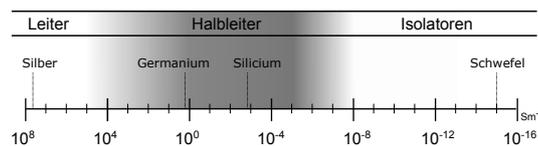


Abbildung 2.1.: Leitfähigkeit verschiedener Materialien [1]

Aus Materialsicht lassen sich Halbleiter mit den Elementhalbleitern und den Verbindungshalbleitern in zwei verschiedene Gruppen unterteilen. Während die Elementhalbleiter aus einem einzigen chemischen Element bestehen, kommen bei Verbindungshalbleitern mehrere Elemente zum Tragen. Die für diese Arbeit relevanten Halbleiter Germanium und Silicium gehören der Gruppe der Elementhalbleiter an.

Bei der Beschreibung kristalliner Festkörper spielt die Energie-Impuls-Beziehung der

Ladungsträger eine elementare Rolle. Im Falle eines Kristalls ergibt die grafische Darstellung dieser Beziehung eine Bandstruktur, welche aus der Lösung der Schrödinger-Gleichung für ein angenähertes Ein-Elektronen-Problem gewonnen werden kann [2]. Jeder Halbleiter verfügt hierbei über eine Bandlücke E_g , in welcher keine Energiezustände definiert sind. Unterhalb der Bandlücke finden sich die Valenzbänder und oberhalb die Leitungsbander wieder. Rechnerisch ergibt sich die Bandlücke aus der Differenz zwischen dem Maximum des höchsten Valenzbandes E_v und dem Minimum des niedrigsten Leitungsbandes E_c .

$$E_g = E_c - E_v \quad (2.1)$$

Da die Größe der Bandlücke maßgeblich die elektrischen Eigenschaften eines Halbleiters bestimmt, stellt sie eine der wesentlichsten charakteristischen Merkmale dar. Das Fermi-Niveau E_f ist in diesem Zusammenhang als jenes Energie-Niveau definiert, welches über eine Besetzungswahrscheinlichkeit von 0,5 verfügt. Im intrinsischen Halbleiter (vgl. Abschnitt 2.2) liegt das Fermi-Niveau E_{fi} nahezu in der Mitte der Bandlücke [2]. Abbildung 2.2 zeigt exemplarisch das Bandstrukturdiagramm von Silicium mit den Energieniveaus E_g , E_c und E_v . Zudem lässt sich aus der Bandstruktur erkennen, ob ein direkter oder indirekter Halbleiter vorliegt. Im Falle von Silicium und Germanium handelt es sich diesbezüglich um indirekte Halbleiter. Anders als bei den direkten Halbleitern, wie beispielsweise Galliumarsenid, tritt bei den indirekten Halbleitern das Energiemaximum des Valenzbandes und das Energieminimum des Leitungsbandes bei unterschiedlichen Wellenvektoren auf (vgl. Abb. 2.2).

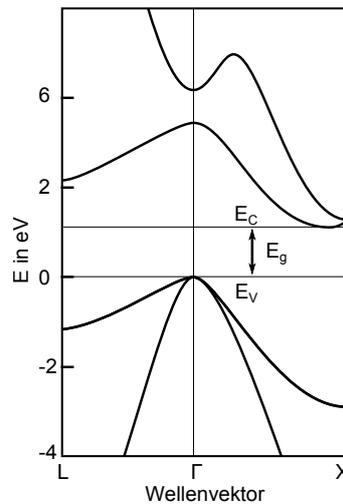


Abbildung 2.2.: Bandstruktur von Silicium im Impulsraum nach Sze et al. [2]

Bei der Darstellung im Ortsraum wird die Energie nicht über den Impuls, sondern über den Ort aufgetragen. Der Verlauf der einzelnen Energiebänder ist dort nicht mehr erkennbar, wodurch sich die Darstellungen erheblich vereinfachen (vgl. Abb. 2.3). Die Energieniveaus E_v und E_c , sowie die daraus resultierende Bandlücke E_g , entsprechen

jenen aus der Darstellung im Impulsraum (vgl. Abb. 2.2). Da für die nachfolgenden Betrachtungen der exakte Verlauf der einzelnen Bänder nicht relevant ist, werden sich die weiteren Abbildungen auf eine Darstellung im Ortsraum beschränken.

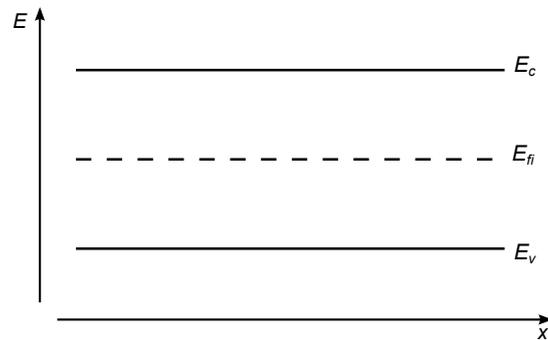


Abbildung 2.3.: Banddiagramm intrinsischer Halbleiter im Ortsraum

2.2. Intrinsischer Halbleiter

Betrachtet man den intrinsischen Halbleiter, bei welchem der Halbleiter in Reinform ohne Dotierung vorliegt, so ist dieser bei einer Temperatur am absoluten Nullpunkt nicht leitend. Bei der Nullpunkttemperatur ist bei einem Halbleiter das Valenzband voll besetzt, während sich im Leitungsband keine Elektronen wiederfinden. In diesem Zustand kann keines der beiden Bänder zur elektrischen Leitung beitragen, der Halbleiter verhält sich dementsprechend wie ein Isolator.

Die Bandlücke eines Halbleiters ist jedoch deutlich kleiner als bei einem Isolator, wodurch bereits bei geringer Energiezufuhr Elektronen vom Valenzband in das Leitungsband gehoben werden können (vgl. Abb. 2.4). Sowohl das im Valenzband entstandene Defektelektron (Loch), als auch das ins Leitungsband gehobene Elektron, können nun zur elektrischen Leitung beitragen. Je nach Größe des Bandabstandes sind Halbleiter dementsprechend teilweise bei Raumtemperatur bereits elektrisch leitend.

2.3. Dotierter Halbleiter

Die Dotierung bezeichnet das Einbringen Fremdatomen in die Kristallstruktur um die elektrischen Eigenschaften des Halbleiters gezielt zu beeinflussen. Fremdatome mit einem Valenzelektron mehr im Vergleich zum intrinsischen Halbleiter, werden als Donatoren bezeichnet - Störstellen mit einem Valenzelektron weniger hingegen als Akzeptoren. Ein mit Donatoren dotierter Halbleiter verfügt über einen Überschuss an Elektronen und stellt

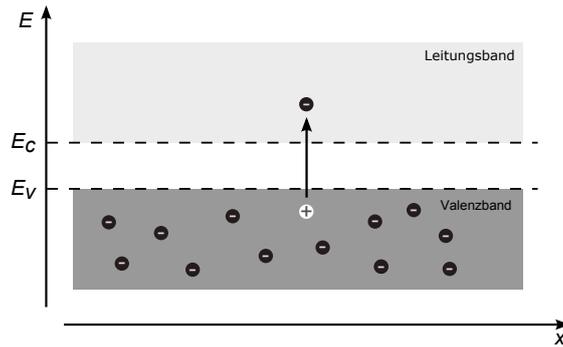


Abbildung 2.4.: Intrinsic Halbleiter bei $T > 0$

einen n-Halbleiter dar. Im Falle des p-Halbleiters kommen Akzeptoren für die Dotierung zum Einsatz, wodurch ein Überschuss an Löchern entsteht. Die Ladungsträger mit höherer Konzentration werden in diesem Zusammenhang als Majoritäten und jene mit niedrigerer Konzentration als Minoritäten bezeichnet.

Durch das Einbringen der Störstellen werden zusätzliche Energieniveaus (E_a für Akzeptoren, E_d für Donatoren) in den Halbleiter eingeführt (vgl. Abb. 2.5), die in der Bandlücke des intrinsischen Halbleiters liegen. Aufgrund des geringen energetischen Unterschieds zwischen dem Niveau der Störstellen und des Leitungs- bzw. Valenzbandes können diese bereits bei deutlich geringerer Energiezufuhr zur elektrischen Leitung beitragen, als dies beim intrinsischen Halbleiter der Fall ist.

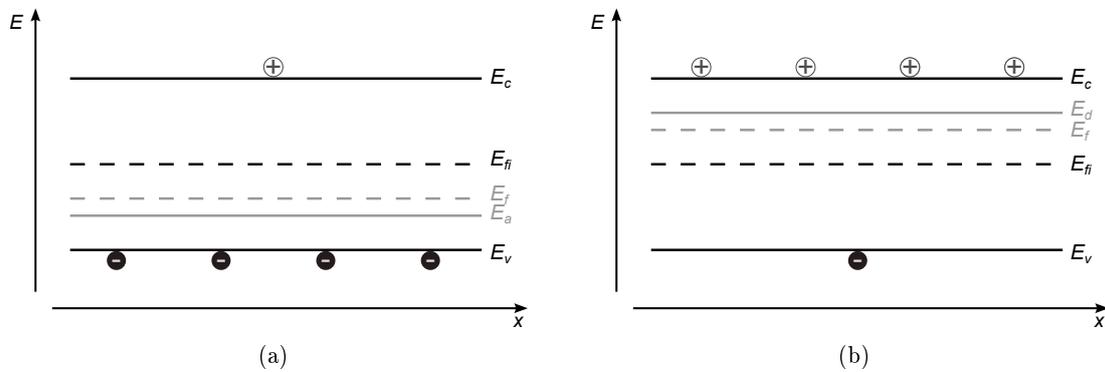


Abbildung 2.5.: Bändermodell dotierter Halbleiter a) n-Halbleiter b) p-Halbleiter

2.4. Halbleiterdiode

Nach der Betrachtung der für dieses Thema relevanten Halbleitergrundlagen soll im Folgenden die für diese Arbeit zentrale Thematik der Halbleiterdiode betrachtet werden. Zunächst wird die grundlegende Funktionsweise der Halbleiterdiode erarbeitet. Darauf basierend wird mit der idealen Diode nach Shockley das einfachste mathematische Modell einer Diode vorgestellt. Abschließend wird ein Modell für eine reale Halbleiterdiode erarbeitet.

2.4.1. Verhalten im thermodynamischem Gleichgewicht

Der PN-Übergang ist der wesentliche Bestandteil einer Halbleiter-Diode und vieler anderer Bauteile in der Halbleitertechnik. Der PN-Übergang besteht hierbei aus einem p-dotierten und einer n-dotierten Schicht die aneinander angrenzen.

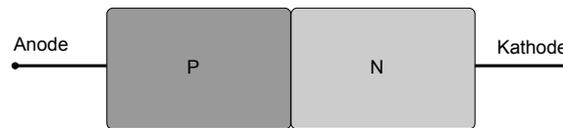


Abbildung 2.6.: PN-Übergang

Aufgrund des Konzentrationsgefälles diffundieren die Löcher aus dem p-Gebiet in das n-Gebiet und die Elektronen aus dem n-Gebiet umgekehrt in das p-Gebiet. Die Ladungsträgerbewegung wird als Diffusionsstrom bezeichnet. Die diffundierenden Majoritätsladungsträger rekombinieren anschließend mit den Majoritätsladungsträgern des jeweils gegenüberliegenden Gebiets. Es bildet sich eine Zone ohne frei bewegliche Ladungsträger, welche sowohl in das p-Gebiet als auch in das n-Gebiet hineinragt. Dieser Bereich wird als Raumladungszone bezeichnet.

Als Folge der Raumladungszone erzeugen die ortsfesten Atomrümpfe ein elektrisches Feld, welches der Diffusion entgegen wirkt. Die Ladungsträgerbewegung aufgrund dieses elektrischen Feldes wird als Driftstrom bezeichnet. Nach einer gewissen Zeit, stellt sich ein Gleichgewicht zwischen Diffusions- und Driftstrom ein. Zwischen dem n- und p-Gebiet stellt sich eine Potentialdifferenz, die Diffusionsspannung, ein. Im Bändermodell gleichen sich die Fermi-Niveaus der beiden Gebiete an. Das Leitungsband und das Valenzband erfahren eine Krümmung im Übergangsbereich. Der Potentialunterschied in Folge der Krümmung der Energiebänder wird hierbei als Potentialwall bezeichnet.

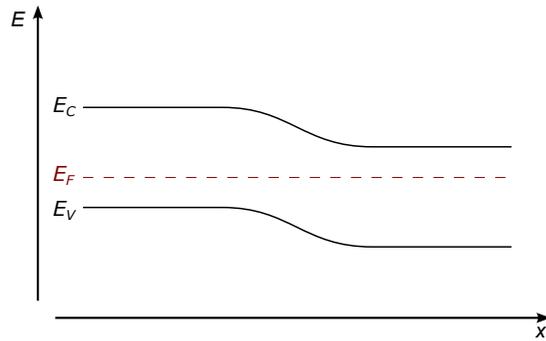


Abbildung 2.7.: PN-Übergang im thermodynamischen Gleichgewicht

2.4.2. Verhalten bei angelegter Spannung

Je nach Polung kann durch das Anlegen einer Spannung U an den PN-Übergang das elektrische Feld der Sperrschicht verstärkt oder abgeschwächt werden. Beim Betrieb in Durchlassrichtung wird das interne Feld und somit die Raumladungszone abgebaut. Wird das interne elektrische Feld durch Anlegen einer Spannung vollständig kompensiert, können die von der Quelle nachgelieferten Ladungsträger fortlaufend im Grenzbereich rekombinieren. Es ergibt sich ein signifikanter Stromfluss.

Im Bändermodell resultiert das Anlegen einer Spannung in Durchlassrichtung in einer Reduzierung des Potentialwalls, während dieser sich beim Anlegen einer Sperrspannung vergrößert.

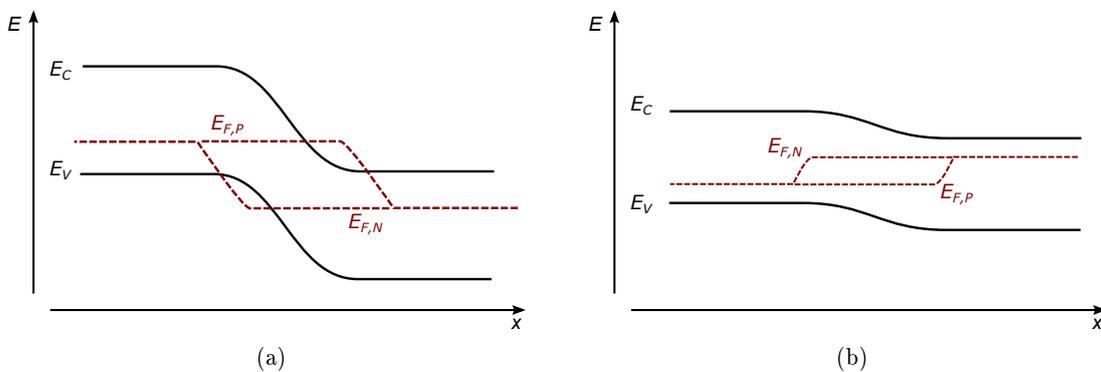


Abbildung 2.8.: Bändermodelle bei angelegter Spannung. a) $U < 0$ b) $U > 0$

Insofern keine Hochinjektion [2] vorliegt, resultiert der Stromfluss in Durchflussrichtung, mit dem Diffusionsstrom und dem Generation-/Rekombinationsstrom, aus zwei

verschiedenen Teilströmen. Der Diffusionsstrom

$$I_{Diff} = I_{Diff,0} \left[\exp \left(\frac{qU}{n_1 kT} - 1 \right) \right] \quad (2.2)$$

ergibt sich durch die Ladungsträgerbewegung der Löcher aus dem p-Gebiet in das n-Gebiet (Löcherstrom I_p) und dem entsprechenden, umgekehrten Vorgang im p-Gebiet (Elektronenstrom I_n). Nicht beinhaltet sind in diesem Fall die Stromanteile, welche durch die Rekombination in der Raumladungszone entstehen. $I_{Diff,0}$ bezeichnet diesbezüglich den Diffusions-Sättigungssperrestrom, n_1 den Diffusions-Idealitätsfaktor, k die Boltzmann-Konstante und T die Temperatur.

Die Rekombination von Elektronen aus dem n-Gebiet mit Löchern aus dem p-Gebiet innerhalb der Raumladungszone wird als Generation-/Rekombinationsstrom

$$I_{GR} = I_{GR,0} \left[\exp \left(\frac{qU}{n_2 kT} - 1 \right) \right] \quad (2.3)$$

bezeichnet. $I_{GR,0}$ ist hierbei der Generations-/Rekombinations-Sättigungssperrestrom und n_2 der Idealitätsfaktor des Generations-/Rekombinationsstroms. In der Theorie nehmen die beiden Idealitätsfaktoren n_1 und n_2 die Werte 1 und 2 an [2].

Bei dem Betrieb in Sperrrichtung fließt bis zum Durchbruch nur ein kleiner, durch die Minoritätsladungsträger hervorgerufener, Sperrstrom. Ab der sogenannten Durchbruchspannung steigt jedoch auch in Sperrrichtung der Stromfluss signifikant an. Zurückzuführen ist der Stromanstieg im Wesentlichen auf thermische Instabilität, den Tunneleffekt und die Avalanche-Multiplikation [2].

2.4.3. Ideale Diode

Betrachtet werden soll in diesem Abschnitt die ideale Diode nach Shockley [3]. Die idealisierte Strom-Spannungsbeziehung basiert hierbei auf den folgenden vier Annahmen [2]:

- Abrupter PN-Übergang, d.h. der Spannungsabfall findet ausschließlich über der Raumladungszone statt
- Gültigkeit der Boltzmann-Näherung für die Ladungsträgerdichten in der Raumladungszone
- Geringe Injektion, d.h. die Minoritätsladungsträgerkonzentration ist klein im Vergleich zur Konzentration der Majoritätsladungsträger
- Keine Generation-Rekombinations-Vorgänge in der Raumladungszone

Demzufolge enthält das Modell der idealen Diode nach Shockley lediglich den Diffusionsstrom, wie er in Gleichung 2.2 angegeben ist. Abbildung 2.9 zeigt die Spannungs-Strom-Kennlinie einer idealen Diode.

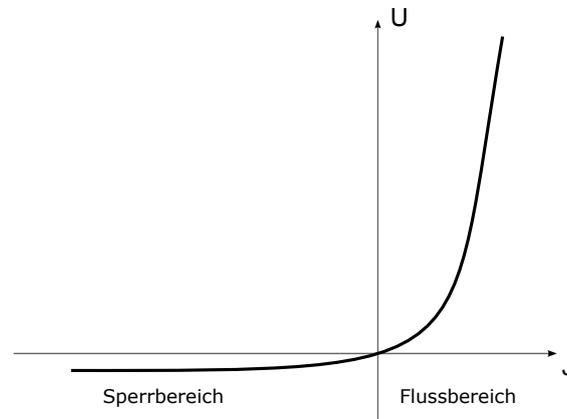


Abbildung 2.9.: Kennlinie der idealen Diode

2.4.4. Reale Diode

Das Modell der idealen Diode nach Shockley beschreibt qualitativ den Verlauf einer Diode. Durch die im Vorfeld getroffenen Vereinfachungen weicht die real gemessene Kennlinie einer Diode jedoch teils deutlich von der idealisierten Kennlinie ab. Für die Beschreibung einer realen Diode müssen eine Vielzahl von zusätzlichen Effekten berücksichtigt werden.

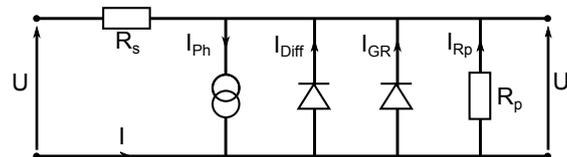


Abbildung 2.10.: Ersatzschaltbild für den Vorwärtsbereich ($U > 0$) einer realen Diode

Die nachfolgenden Betrachtungen beziehen sich auf das Ersatzschaltbild in Abbildung 2.10. Zusätzlich zu dem Diffusionsstrom I_{Diff} berücksichtigt dieses die folgenden Mechanismen:

Generation Rekombination in der Raumladungszone Neben dem Diffusionsstrom I_{Diff} über der neutralen Zone des PN-Übergangs tritt durch die Generation und Rekombination von Ladungsträgern in der Raumladungszone ein zusätzlicher Strom nach Gl. 2.3 auf. Im Ersatzschaltbild wird dieser als zusätzliche Diode modelliert.

Parallelwiderstand Der Parallelwiderstand wird durch Ströme senkrecht zum PN-Übergang (meist Oberflächenströme) hervorgerufen. Der Widerstand R_p liegt parallel zu den beiden Dioden im Ersatzschaltbild und äußert sich durch den Stromanteil I_{rp} .

$$I_{Rp} = \frac{U - IR_s}{R_p} \quad (2.4)$$

Serienwiderstand Über den Serienwiderstand R_s wird der ohmsche Widerstand der Bahngebiete sowie der Kontaktwiderstand der Metallkontakte berücksichtigt. Da sich der Einfluss des Serienwiderstandes mit zunehmendem Strom verstärkt, macht sich dieser insbesondere bei höheren Strömen bemerkbar. Durch den Serienwiderstand ergibt sich die Spannung über den parallelen Bauelementen im Ersatzschaltbild zu:

$$U_i = U - IR_s. \quad (2.5)$$

Photostrom Der Photostrom tritt bei Beleuchtung als zusätzlicher Stromanteil auf. Daraus ergibt sich eine Verschiebung der Kennlinie in negativer Richtung auf der Strom-Achse im Kennliniendiagramm.

Aus der Kombination der aufgeführten Mechanismen und Effekte resultiert die Gleichung einer realen Diode, wie sie für diese Arbeit Verwendung findet:

$$I_{real} = I_{Diff,0} \left[\exp \left(\frac{q(U - IR_s)}{n_1 kT} - 1 \right) \right] + \\ + I_{GR,0} \left[\exp \left(\frac{q(U - IR_s)}{n_2 kT} - 1 \right) \right] + \frac{U - IR_s}{R_p} - I_{ph}. \quad (2.6)$$

3. Kurvenanpassung

Nach der Betrachtung der wesentlichen Halbleitergrundlagen einer Halbleiterdiode, beschäftigt sich dieses Kapitel mit der Auswertung real gemessener Diodenkennlinien. Ziel dieses Vorgehens ist es, die wesentlichen Kennwerte einer Diode aus deren Kennlinie zu extrahieren, um eine Aussage über Qualität und Beschaffenheit der Diode treffen zu können. Verwendung findet hierbei die sogenannte Kurvenanpassung (engl. curve fitting). Bei dieser wird ein gegebenes mathematisches Modell über seine Parameter an eine gemessene Kennlinie derart angepasst, dass Modell und Kennlinie bestmöglich übereinstimmen. Für die vorliegende Arbeit kam als mathematisches Modell die reale Diode nach Gleichung 2.6 zum Einsatz. Die Parameter, über welche das Modell der realen Diode an die gemessene Kennlinie angepasst wird, sind die Sperrströme $I_{Diff,0}$ und $I_{GR,0}$, der Serienwiderstand R_s , der Parallelwiderstand R_p , der Photostrom I_{ph} , sowie die beiden Idealitätsfaktoren n_1 und n_2 .

Da es sich bei der Kurvenanpassung um ein Optimierungsproblem handelt, wird zunächst dieses grundlegende Problem behandelt. Mit der Methode der kleinsten Quadrate wird anschließend eine Vorgehensweise zur Verknüpfung von Messwerten und mathematischem Modell vorgestellt, welche die Grundlage für den eigentlichen Optimierungsprozess schafft. Darauf basierend werden die für diese Arbeit implementierten Optimierungsalgorithmen nach Nelder-Mead sowie Marquardt-Levenberg vorgestellt. Die Basis für den letzteren Algorithmus bilden das Gradienten-, Newton und Gauß-Newton-Verfahren, die zusätzlich für das bessere Verständnis erläutert werden.

3.1. Optimierung

Aus mathematischer Sicht bezeichnet die Optimierung allgemein die Minimierung oder Maximierung einer Zielfunktion. Da jedes Maximierungsproblem, durch Transformation der Zielfunktion, in ein Minimierungsproblem überführt werden kann, wird im Folgenden ausschließlich Letzteres behandelt. Zudem beschränken sich die Betrachtungen auf die nichtlineare Optimierung, welcher eine nichtlineare Zielfunktion oder Nebenbedingung zu Grunde liegen, da dies für die vorliegende Diodengleichung (vgl. Gl. 2.6) mit deren exponentiellen Anteilen der Fall ist. Mathematisch kann das Optimierungsproblem wie folgt definiert werden:

$$\min f(x). \tag{3.1}$$

Hierbei bezeichnet f die zu minimierende Zielfunktion und x den Vektor mit Variablen, auch als Parameterraum der Zielfunktion bekannt.

Die zu optimierende Zielfunktion durchläuft im Allgemeinen eine Vielzahl von lokalen Minima, wie dies auch für die Zielfunktion bei der Diodenauswertung der Fall ist. Jenes lokale Minimum mit dem niedrigsten Funktionswert wird in diesem Zusammenhang als globales Minimum bezeichnet. Das Auffinden dieses globalen Minimums stellt ein zentrales Problem der Optimierung dar. Während sich beispielsweise über den Gradienten der Zielfunktion verhältnismäßig einfach herausfinden lässt, ob sich ein Betrachtungspunkt in einem lokalem Minimum befindet, lässt sich in der Regel nur schwierig feststellen, ob es sich um globales Minimum handelt (vgl. Abschnitt 3.5.3).

3.2. Kleinste-Quadrate-Problem

Grundlage für die Optimierung ist, wie in Gleichung 3.1 beschrieben, eine zu minimierende Zielfunktion. Bei der Kurvenanpassung muss die Zielfunktion eine Verknüpfung zwischen n realen Messwerten $(t_1, y_1), \dots, (t_n, y_n)$ und dem parametrisierten Modell (im vorliegenden Fall Gl. 2.6) herstellen. Auf diese Weise kann durch eine Minimierung der Zielfunktion jener Parametersatz gefunden werden, dessen zugehörige Modellkurve m möglichst gut mit den Messwerten übereinstimmt. Eine der gängigsten Herangehensweisen für die Verknüpfung von Modellkurve und Messwerten ist die Methode der kleinsten Quadrate (engl. least squares, LSQ). Diese ist als

$$\text{LSQ} = \frac{1}{2} \sum_{i=1}^n r_i^2 \quad (3.2)$$

definiert [4], wobei r_i das Residuum (vgl. Abb. 3.1) bezeichnet, welches als Abweichung zwischen Modellwert mit Parametervektor x und dem Messwert definiert ist:

$$r_i = y_i - m(x, t_i) \quad (3.3)$$

Im vorliegenden Fall einer Diodenkennlinie entspricht y_i hierbei dem elektrischen Strom eines Messpunktes in der gemessenen Kennlinie. Bei $m(x, t_i)$ handelt es sich dementsprechend um die Lösung der realen Diodengleichung (vgl. 2.6) mit dem Vektor der Diode-Parameter x und der zum Messpunkt zugehörigen Spannung t_i . Mittels der Methode der kleinsten Quadrate (vgl. 3.2) werden die Differenzen zwischen Modell- und Messwerten miteinander zu einer Zielfunktion verknüpft. Diese Zielfunktion wird anschließend über einen Optimierungsalgorithmus minimiert um schlussendlich eine möglichst gute Übereinstimmung zwischen Modellkurve und gemessener Kennlinie zu erreichen. Im Zuge dieser Arbeit wurden mit dem Marquardt-Levenberg- und dem Nelder-Mead-Algorithmus zwei lokale Optimierungsmethoden umgesetzt.

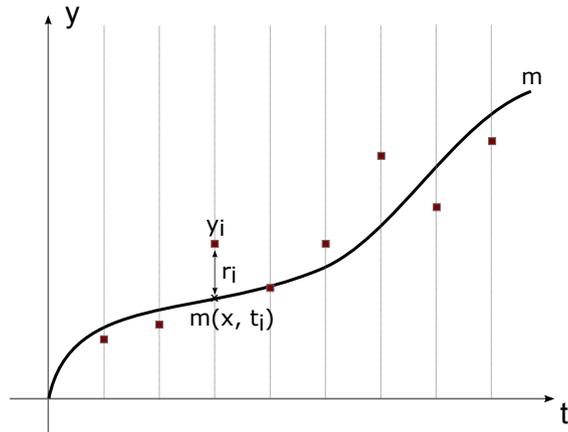


Abbildung 3.1.: Beispielhafte Beziehung zwischen Residuum r_i , Messwert y_i und Modellwert $m(x, t_i)$ bzw. Modellkurve m bei der Methode der kleinsten Quadrate

3.3. Nelder-Mead (Simplex-Downhill)

Der Simplex-Downhill-Algorithmus wurde 1965 von John Nelder und Roger Mead als Technik zur Minimierung einer Zielfunktion im multidimensionalen Parameterraum vorgestellt. Anders als der in Kapitel 3.4.4 behandelte Marquardt-Levenberg-Algorithmus kommt diese Methode ohne jegliche Ableitung der Zielfunktion aus. Der Name *Simplex-Downhill* ist an die grundlegende Funktionsweise, welche auf einem Simplex basiert, angelehnt. Ein n -Simplex bezeichnet in der Mathematik einen n -dimensionalen Körper mit $n + 1$ Eckpunkten [4]. Dementsprechend ist ein 2-Simplex ein Dreieck in einem zwei-dimensionalen Raum, und ein 3-Simplex ein Tetraeder, welcher sich in drei Dimensionen ausbreitet und über 4 Eckpunkte verfügt (vgl. Abbildung 3.2).

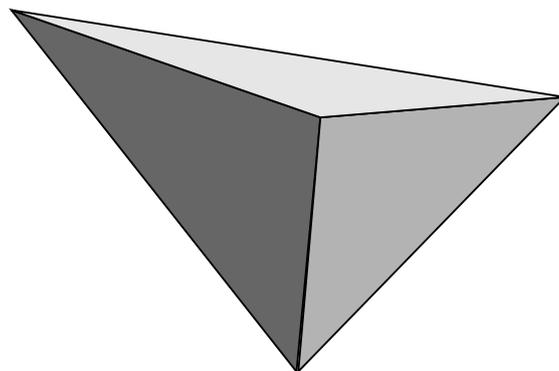


Abbildung 3.2.: Beispielhaftes 3-Simplex (Tetraeder)

Für die Optimierung einer Zielfunktion mit m Parametern wird bei dem Optimie-

rungsalgorithmus von Nelder-Mead ein entsprechendes m -Simplex herangezogen. Jeder Eckpunkt des Simplex symbolisiert hierbei einen Parametersatz, welcher aus m einzelnen Parametern besteht. Das gesamte Simplex besteht somit aus $m + 1$ Parametersätzen. Da der Algorithmus iterativ eine Anfangslösung verbessert, muss dieser initial mit einem Start-Simplex angestoßen werden. Voraussetzung für einen fehlerfreien Ablauf des Algorithmus ist, dass es sich um ein nicht degeneriertes Simplex handelt.

Ein n -Simplex wird diesbezüglich als degeneriert (oder singulär) bezeichnet, falls es sich in weniger als n Dimensionen ausbreitet [4]. Bei einem 2-Simplex wäre dies beispielsweise der Fall, falls alle drei Eckpunkte auf einer Linie liegen, bei einem 3-Simplex, falls die vier Eckpunkte in einer Ebene liegen. Sei S ein n -Simplex mit den Eckpunkten $x_1 \dots x_{n+1}$, dann lässt sich seine Singularität über die Singularität der Matrix

$$V(S) = (x_2 - x_1, x_3 - x_1, \dots, x_{n+1} - x_1) \quad (3.4)$$

überprüfen [4].

Bei einer Zielfunktion mit m Parametern wird die Simplex-Downhill-Methode mit einem nicht degenerierten m -Simplex (bestehend aus $m + 1$ Parametersätzen) gestartet. In jeder Iteration werden die zu den Eckpunkten (Parametersätzen) zugehörigen Funktionswerte berechnet. Aus dieser Menge wird der Eckpunkt mit dem niedrigsten Funktionswert f_l (bester Eckpunkt x_l), sowie die beiden mit den höchsten Funktionswerten f_h und f_{h2} (schlechtester x_h und zweitschlechtester Eckpunkt x_{h2}) bestimmt. Der Schwerpunkt des Simplex unter Nichtberücksichtigung des schlechtesten Eckpunktes x_h , ist als

$$\bar{x} = \frac{1}{m} \sum_{j \neq h} x_j \quad (3.5)$$

definiert [5]. Punkte entlang der Linie zwischen \bar{x} und dem schlechtesten Punkt x_h werden durch

$$\bar{x}(t) = \bar{x} + t(x_h - \bar{x}) \quad (3.6)$$

angegeben [4], wobei der Faktor t die Entfernung zum Schwerpunkt \bar{x} regelt. Anhand dieser Informationen wird in jedem Iterationsschritt ein neues Simplex gebildet. Hierbei wird grundsätzlich versucht, den schlechtesten Eckpunkt x_h (vgl. Abb. 3.3) durch einen besseren zu ersetzen. Grundlage für die Entscheidung, welcher Mechanismus zum Tragen kommt, ist der Reflektionspunkt x_r . Dieser ergibt sich entsprechend Gleichung 3.6 zu:

$$x_r = \bar{x}(-1). \quad (3.7)$$

Die nachfolgenden Mechanismen (vgl. Abb. 3.3) kommen bei der Neubildung des Simplex zum Tragen:

Reflektion Bedingung: Der reflektierte Punkt ist weder der beste noch der schlechteste Punkt im Simplex. Die Reflektion entspricht der Spiegelung des schlechtesten Punktes am Mittelpunkt der verbleibenden Punkte.

$$x_r = \bar{x}(-1) \quad (3.8)$$

Reflektion mit Expansion Bedingung: Der Reflektionspunkt ist besser als der beste Punkt im Simplex. Es wird eine Reflektion mit dem Faktor Gamma – hier 2 – getätigt um den Schritt in die gleiche Richtung zu vergrößern. Der Expansionsfaktor x_e ergibt sich zu:

$$x_e = \bar{x}(-2). \quad (3.9)$$

Kontraktion Bedingung: Der Reflektionspunkt ist schlechter, als der schlechteste Eckpunkt des Simplex. Der schlechteste Punkt wird näher an den Mittelpunkt der Punkte x_{h2} und x_l herangerückt. Für die innere Kontraktion (Kontraktionspunkt x_{ic}) wird ein negativer, für die äußere Kontraktion (Kontraktionspunkt x_{oc}) ein positiver Faktor verwendet.

$$x_{ic} = \bar{x} \left(-\frac{1}{2} \right) \quad (3.10)$$

$$x_{oc} = \bar{x} \left(\frac{1}{2} \right) \quad (3.11)$$

Komprimierung Bedingung: Die Kontraktion war nicht erfolgreich, d.h. der Funktionswert der zu minimierenden Zielfunktion hat sich bei keiner der vorherigen Maßnahmen verkleinert. Das Simplex wird um den besten Punkt x_l zusammengezogen. Alle Eckpunkte werden entsprechend der Vorschrift

$$x_i = \frac{1}{2}(x_l + \bar{x}) \quad \forall i \neq l \quad (3.12)$$

neu berechnet und zugewiesen.

Eine Veranschaulichung der zum Tragen kommenden Mechanismen für den Fall einer Optimierung mit zwei Parametern (2-Simplex) liefert Abbildung 3.3. Informationen über die allgemeine Implementierung liefert Algorithmus 1. Dieser beschreibt einen Iterationsschritt des Downhill-Simplex-Verfahren.

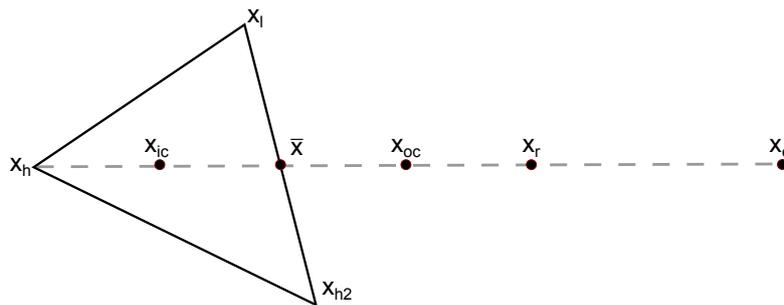


Abbildung 3.3.: Mechanismen Simplex Downhill

Algorithm 1 Iterationsschritt Nelder-Mead-Algorithmus - Pseudocode

```
1:  $x_r \leftarrow \bar{x}(-1)$  ▷ Berechnung des Reflektionspunktes  $x_r$ 
2:  $f_r \leftarrow f(x_r)$ 
3: if  $f_l \leq f_r < f_{h2}$  then ▷  $x_r$  weder bester noch schlechtester Punkt im neuen Simplex
4:    $x_h \leftarrow x_r$ 
5: else if  $f_r < f_l$  then ▷  $x_r$  ist besser als der aktuell beste Punkt
6:    $x_e \leftarrow \bar{x}(-2)$ 
7:    $f_e \leftarrow f(x_e)$  ▷ Expansion
8:   if  $f_e < f_r$  then
9:      $x_h \leftarrow x_e$ 
10:  else
11:     $x_h \leftarrow x_r$ 
12:  end if
13: else if  $f_r \geq f_{h2}$  then ▷  $x_r$  ist weiterhin schlechtester Punkt im neuen Simplex
14:    $success \leftarrow false$ 
15:   if  $f_{h2} \leq f_r < f_h$  then
16:      $x_{ic} \leftarrow \bar{x}(-1/2)$ 
17:      $f_{ic} \leftarrow f(x_{ic})$  ▷ Innere Kontraktion
18:     if  $f_{ic} \leq f_r$  then
19:        $x_h \leftarrow x_{ic}$ 
20:        $success \leftarrow true$ 
21:     end if
22:   else
23:      $x_{oc} \leftarrow \bar{x}(1/2)$ 
24:      $f_{oc} \leftarrow f(x_{oc})$  ▷ Äußere Kontraktion
25:     if  $f_{ic} < f_h$  then
26:        $x_h \leftarrow x_{oc}$ 
27:        $success \leftarrow true$ 
28:     end if
29:   end if
30:   if  $success \equiv false$  then ▷ äußere bzw. innere Kontraktion war nicht erfolgreich
31:     for  $i = 1$  to  $n + 1$  do
32:        $x_i \leftarrow (1/2)(x_1 + x_i)$  ▷ Dreieck um den besten Punkt zusammenziehen
33:     end for
34:   end if
35: end if
```

3.4. Liniensuche

Die Liniensuche (engl. line search) stellt eine iterative Strategie für den Optimierungsprozess dar. Ausgehend vom aktuellen Punkt x_k wird in jeder Iteration zunächst eine Suchrichtung und anschließend eine Schrittlänge berechnet um den nächsten Punkt x_{k+1} zu bestimmen. Die Suchrichtung p_k gibt an, in welcher Richtung sich der nächste Iterationspunkt x_{k+1} befindet, während die Schrittlänge α_k darüber entscheidet, wie weit sich der Algorithmus entlang der zuvor berechneten Richtung p_k bewegt (vgl. Abb. 3.4).

$$x_{k+1} = x_k + \alpha_k p_k. \quad (3.13)$$

Die Berechnung der Suchrichtung p_k und der Schrittweite α_k stellt dementsprechend die zentrale Aufgabe der Algorithmen der Liniensuche dar. Hierfür haben sich unterschiedliche Herangehensweisen etabliert. Für die Berechnung der Richtung p_k werden im Folgenden mit dem Gradienten- und Newton-Verfahren, bzw. dem darauf basierenden Gauß-Newton-Verfahren verschiedene Algorithmen betrachtet.

Abschließend wird der Levenberg-Marquardt-Algorithmus erläutert, welcher im Programmteil dieser Arbeit implementiert wurde. Der Levenberg-Marquardt-Algorithmus kombiniert das Gradienten- mit dem Gauß-Newton-Verfahren und bringt zudem eine adaptiv arbeitende Schrittweitenregulierung mit. Aufgrund von Letzterem wird im Folgenden auf eine Erläuterung separater Algorithmen zur Schrittweitensteuerung verzichtet. Stattdessen wird auf die Übersicht von Nocedal und Wright [4][Kapitel 3] verwiesen.

3.4.1. Gradientenverfahren

Das Gradientenverfahren ist auch als Methode des steilsten Abstiegs (engl. steepest descent) bekannt. Das Verfahren nutzt den Umstand, dass sich der Funktionswert einer differenzierbaren, realen Funktion am schnellsten in Richtung des negativen Gradienten verkleinert. Dementsprechend setzt dieses Verfahren für die Bestimmung der Richtung p_k (vgl. Gl. 3.13) auf den negativen Gradienten der Funktion im jeweiligen Iterationspunkt.

$$p_{k,G} = -\nabla f \quad (3.14)$$

Wie in Abbildung 3.4 dargestellt, steht die Richtung p_k senkrecht auf den Niveaulinien der Funktion, wodurch sich der für das Gradientenverfahren typische Zick-Zack-Kurs einstellt. Für die Schrittweitenregelung bieten sich neben einer konstanten Schrittweite auch adaptive Verfahren [4] an.

3.4.2. Newton-Verfahren

Eine weiteres zentrales Verfahren zur Bestimmung der Suchrichtung ist die Newton-Methode. Ausgangspunkt ist die Taylor-Approximation zweiter Ordnung der Zielfunktion

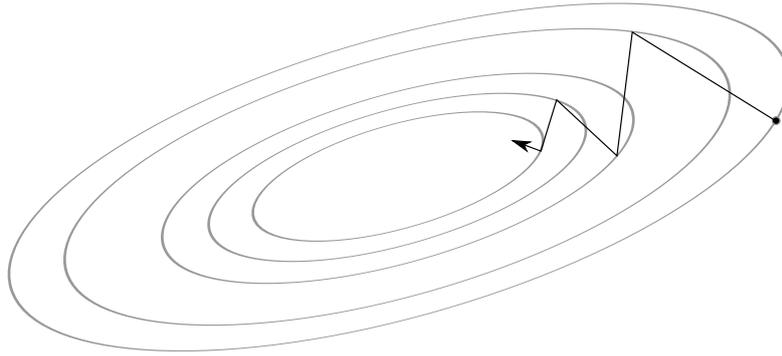


Abbildung 3.4.: Zick-Zack-Kurs des Gradientenverfahrens

um einen Entwicklungspunkt x_k . Dann ist

$$f(x_k + p) \approx f_k + p^T \nabla f_k + \frac{1}{2} p^T \nabla^2 f_k p, \quad (3.15)$$

wobei f_k dem Funktionswert und ∇f_k dem Vektor mit den Gradientenwerten der Zielfunktion im Punkt x_k entspricht. Die Matrix $\nabla^2 f_k$ stellt die Hesse-Matrix dar, welche die partiellen zweiten Ableitungen der Zielfunktion nach deren Parametern zusammenfasst. Die Newton-Richtung wird über die Newton-Gleichung definiert und ergibt sich durch Nullsetzen der ersten Ableitung von Gleichung 3.15 zu:

$$\nabla^2 f_k p_{k,N} = -\nabla f_k. \quad (3.16)$$

Die Newton-Richtung führt zuverlässig zum Minimum, solange die Differenz zwischen Taylor-Approximation und tatsächlicher Funktion klein bleibt und die Matrix $\nabla^2 f_k$ positiv definit ist. Trifft Letzteres nicht zu, kann der Fall eintreten, dass keine Newton-Richtung berechnet werden kann, da $(\nabla^2 f_k)^{-1}$ unter Umständen nicht existiert.

In der Nähe eines lokalen Minimums ist $\nabla^2 f_k$ positiv definit [4]. In dieser Umgebung konvergiert die Newton-Methode mit quadratischer Geschwindigkeit, insofern eine Schrittweite von $\alpha_k = 1$ genutzt wird. Wesentlicher Nachteil der Newton-Methode ist die Voraussetzung der zweiten Ableitung. Zudem konvergiert der Algorithmus nur in der Nähe der Lösung zuverlässig, solange $\nabla^2 f_k$ positiv definit ist.

3.4.3. Gauß-Newton-Verfahren

Das Gauß-Newton-Verfahren kann als Erweiterung des Newton-Verfahrens zum Lösen allgemeiner Optimierungsprobleme angesehen werden. Im Gegensatz zum Newton-Verfahren ist das Gauß-Newton-Verfahren speziell auf das Problem der kleinsten Quadrate mit dem Residuenvektor

$$r(x) = (r_1(x), r_2(x), \dots, r_m(x))^T \quad (3.17)$$

mit m Elementen angepasst (vgl. Kapitel 3.2). Mit dem Residuum-Vektor r lässt sich die Jacobi-Matrix als $m \times n$ Matrix der ersten partiellen Ableitungen nach den n Parametern x_i definieren:

$$J(x) = \left[\frac{\partial r_j}{\partial x_i} \right]_{k=1, \dots, m; i=1, \dots, n} = \begin{bmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix}. \quad (3.18)$$

Anstelle der expliziten Lösung der Hesse-Matrix, wie dies für das Newton-Verfahren der Fall ist, wird beim Gauß-Newton-Verfahren folgende Vereinfachung angenommen:

$$\nabla^2 f_k \approx J_k^T J_k. \quad (3.19)$$

Aus Gleichung 3.16 ergibt sich durch die Approximation der Hesse-Matrix die Gauß-Newton-Suchrichtung durch Lösung der Gleichung

$$J_k^T J_k p_{k,GN} = -J_k^T r_k. \quad (3.20)$$

Ein erheblicher Vorteil gegenüber dem Newton-Verfahren ist die Tatsache, dass mit der Jacobi-Matrix lediglich die ersten partiellen Ableitungen der Zielfunktion benötigt werden. Nahe eines lokalen Minimums stimmt die Approximation aus Gleichung 3.19 gut mit der tatsächlichen Hesse-Matrix überein [4], weshalb die Konvergenzgeschwindigkeit in diesem Bereich gleich jener des Newton-Verfahrens ist. Bei größerer Entfernung zum lokalen Minimum ist jedoch die Konvergenz nicht garantiert.

3.4.4. Levenberg-Marquardt-Verfahren

Der Levenberg-Marquardt-Algorithmus kombiniert das Gauß-Newton- mit dem Gradientenverfahren. Das Verfahren nutzt die gleiche Näherung für die Hesse-Matrix, wie das Gauß-Newton-Verfahren (vgl. Gl. 3.19), benötigt deshalb ebenfalls nur die ersten Ableitungen der Zielfunktion. Die Richtung $p_{k,LM}$ wird hierbei über eine Modifizierung von Gleichung 3.20 definiert [6].

$$(J_k^T J_k + \lambda \text{diag}(J_k^T J_k)) p_{k,LM} = -J_k^T r_k \quad (3.21)$$

Die Variable λ bezeichnet den positiv definierten Dämpfungsfaktor. Durch diesen wird garantiert, dass die Koeffizienten-Matrix immer positiv definit ist, wodurch gesichert ist, dass $p_{k,LM}$ grundsätzlich in eine absteigende Richtung zeigt. Zum Anderen wird über den Dämpfungsfaktor die Interpolation zwischen dem Gauß-Newton-Verfahren und dem Gradientenverfahren bewerkstelligt. Für große Werte von λ gilt:

$$\lim_{\lambda \rightarrow \infty} p_{k,LM} = -\frac{1}{\lambda} \nabla f_k = -\frac{1}{\lambda} p_{k,G}. \quad (3.22)$$

Das Levenberg-Marquardt- wird zum Gradientenverfahren. Nimmt λ hingegen kleine Werte an, geht das Verfahren in das Gauß-Newton-Verfahren mit

$$\lim_{\lambda \rightarrow 0} p_{k,LM} = p_{k,GN} \quad (3.23)$$

über. Neben der Richtung reguliert der Dämpfungsfaktor dementsprechend auch die Schrittweite. Ein separater Schrittweitenalgorithmus ist nicht erforderlich. Der Dämpfungsfaktor kann in jedem Iterationsschritt angepasst werden. Für die Anpassung haben sich verschiedene Varianten herauskristallisiert. Eine gängige Variante erhöht λ um einen Faktor γ falls, die neue Lösung besser ist als die aktuelle. Ist die neue Lösung schlechter, wird λ hingegen um den Faktor η verringert.

Als Abbruchkriterium wird von Press et al. [6] vorgeschlagen, die Änderung des Funktionswertes der aktuellen Lösung zu beobachten. Fällt die Differenz zwischen aktuellem und vorangegangenem Lösungswert unter einen vordefinierten, kleinen Wert ϵ , so ist das Abbruchkriterium erfüllt. Im Rahmen dieser Arbeit wurde eine abgeänderte Variante dieses Kriteriums implementiert. So werden nicht die Unterschiede zwischen den beiden aktuellsten Lösungen, sondern die Differenzen über mehrere Iterationen hinweg betrachtet. In Tests erwies sich hier der Wert von 5 Iterationen als praktikabel.

Da das Levenberg-Marquardt-Verfahren lokal in das Gauß-Newton-Verfahren übergeht, erbt es auch dessen lokale Konvergenzeigenschaften. Dementsprechend konvergiert das Verfahren nahe dem Optimum mit quadratischer Geschwindigkeit. Eine Konvergenz ist allerdings auch hier nicht garantiert. Dadurch, dass das Verfahren weit entfernt vom Minimum auf das Gradientenverfahren setzt, und erst nahe der Lösung in das Gauß-Newton-Verfahren übergeht, ist es jedoch deutlich robuster.

3.5. Problemfelder und Lösungen bei der praktischen Umsetzung

Bei der Umsetzung der theoretischen Grundlagen ergaben sich im Laufe der Arbeit eine Reihe von Schwierigkeiten, auf welche im Folgenden eingegangen wird.

3.5.1. Reale Diode als implizite Funktion

Ein zentrales Problem bei der praktischen Umsetzung ist die Darstellungsform des in Gleichung 2.6 beschriebenen Modells einer realen Diode. Während sich die Literatur bei der Erläuterung der Optimierungsalgorithmen im Allgemeinen auf eine explizite Zielfunktion bezieht, handelt es sich bei dem vorliegenden Problem um eine Funktion in impliziter Darstellung. Dies bedeutet, dass die Gleichung nicht in der Form $f(x) = y$,

sondern $f(x, y) = 0$, gegeben ist. Da sich die implizite Diodengleichung in Gleichung 2.6 nur schwierig in eine explizite Darstellung umformen lässt, bringt dies einige Probleme mit sich.

Im Allgemeinen arbeiten nichtlineare Optimierungsalgorithmen iterativ. Dies bedeutet, dass eine Anfangslösung schrittweise verbessert wird. Je nach Algorithmus wird hierbei das Ergebnis der Zielfunktion in unterschiedlichen Punkten berechnet. Ist diese Auswertung der Zielfunktion für eine explizite Funktion unproblematisch, muss im Falle der impliziten Diodengleichung die Gleichung numerisch gelöst werden. Da es sich um eine nichtlineare Funktion handelt, muss auch dieses Problem iterativ angegangen werden. Voraussetzung für einen reibungslosen Ablauf des Optimierungsalgorithmus ist eine hohe Genauigkeit der Berechnung des Wertes der Zielfunktion. Der Exaktheit der Lösung gegenüber steht die Geschwindigkeit des Algorithmus zum Lösen der Gleichung. Je nach Optimierungsverfahren und Anzahl der Messwerte muss die Zielfunktion in jeder Iteration mehrmals gelöst werden. Im Rahmen dieser Arbeit wurde diesbezüglich die Van Wijngaarden-Dekker-Brent-Methode nach Press et al. [6] implementiert.

3.5.2. Ableitung der Zielfunktion

Ein weiteres Problem stellt die Ableitung der impliziten Diodenfunktion dar. Zwar wird diese für den Nelder-Mead-Algorithmus nicht benötigt, jedoch sind die partiellen Ableitungen der Zielfunktion nach den unterschiedlichen Parametern für den Levenberg-Marquardt-Algorithmus eine Grundvoraussetzung um die benötigte Jacobi Matrix (vgl. Gl. 3.18) zu bestimmen. Im Allgemeinen lässt sich die Ableitung einer impliziten Funktion $f(x, y)$ über

$$\frac{dy}{dx} = -\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}} \quad (3.24)$$

bestimmen. Jedoch ist eine Berechnung per Hand fehleranfällig und unflexibel gegenüber Änderungen. So müssen bei jeder Variation der Diodengleichung auch die entsprechenden Ableitungen neu bestimmt werden. Ein automatisiertes Ableiten über numerische Verfahren ist dementsprechend wünschenswert. Dies zeigte sich allerdings in der Umsetzung als problematisch, da bereits kleine Abweichungen von der exakten Lösung den Algorithmus von Levenberg-Marquardt maßgeblich negativ beeinflussen. Testweise wurde hierfür Ridders Methode nach Press et al. [6] herangezogen. Für einfache Testfunktionen waren die Ableitungen hierbei ausreichend genau, so dass die Ergebnisse des Optimierungsalgorithmus nur geringfügig beeinflusst wurden. Für die deutlich komplexere und nichtlineare Diodenfunktion lagen die Abweichungen hingegen in Bereichen, welche sich deutlich auf das Endergebnis der Optimierung auswirkten. Aus diesem Grund wurde in diesem Punkt auf ein Stück Flexibilität verzichtet und stattdessen die Ableitungen direkt in den Programmcode eingepflegt.

3.5.3. Auffinden des globalen Minimums

Wie bereits in Kapitel 3.1 beschrieben, kann eine Funktion über eine Vielzahl an verschiedenen Minima (vgl. Abb. 3.5) verfügen. Das Auffinden des globalen Minimums ist hierbei ein grundsätzliches Problem der Optimierung, welchem natürlich auch die Algorithmen nach Nelder-Mead und Marquardt-Levenberg unterliegen. Zudem handelt es sich bei beiden um Algorithmen, welche rein für die Suche nach einem lokalen Minimum erarbeitet wurden. Ein Großteil der globalen Ansätze geht jedoch auf das Lösen lokaler Probleme zurück. Im Zusammenspiel mit globalen Strategien können somit beide Algorithmen also durchaus für globale Lösungsansätze herangezogen werden. Allerdings impliziert dies in diesem Fall auch eine mehrfache Durchführung des lokalen Optimierungsprozesses unter unterschiedlichen Startbedingungen und somit eine Vervielfachung der Berechnungsdauer.

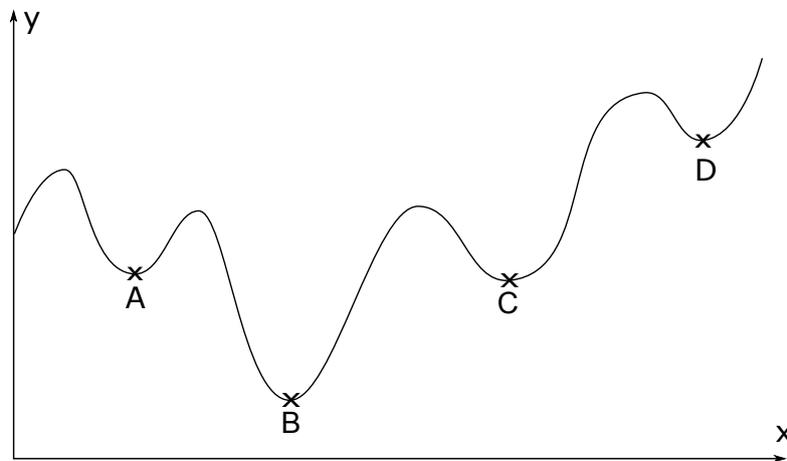


Abbildung 3.5.: Typischer Funktionsverlauf mit mehreren Minima. Die Punkte A,C und D stellen lokale Minima und der Punkt B das globale Minimum dar.

Im Zuge dieser Arbeit wurde hierbei eine Option für ein automatisiertes, wiederholtes Lösen des Optimierungsproblems implementiert. Je nach Einstellung in der Applikation wird der Algorithmus, über eine bestimmte Anzahl an Durchläufen, nahe der vorangegangenen Lösung neu gestartet. Die Startwerte des aktuellen Durchlaufes entsprechen demzufolge der Lösung des vorherigen Durchlaufes, welche um einen zufälligen Faktor variiert werden um potentiell bessere Lösungen zu finden. Falls mit den neuen Startparameter eine bessere Lösung gefunden werden kann, wird diese akzeptiert, andernfalls wieder verworfen. Für weitere, durchaus komplexere, Algorithmen mit erfolgversprechenden Aussichten, wie beispielsweise Simulated annealing oder den evolutionären Algorithmen, sei auf Horst et al. [7] verwiesen. Ein ebenfalls vielversprechender Ansatz ist die Verwendung eines Mehrgitterverfahrens, bei welchem das Problem zunächst in einer groben Auflösung betrachtet wird, die anschließend schrittweise erhöht wird [8].

3.5.4. Begrenzung der Parameter

Grundsätzlich arbeiten die implementierten Optimierungsalgorithmen nach Nelder-Mead und Levenberg-Marquardt auf einem unbeschränkten Parameterraum. Allerdings sind nicht alle mathematischen Lösungen auch physikalisch sinnvoll. So sind beispielsweise bereits im Vorfeld sämtliche negative Lösungen für die Parameter der realen Diode auszuschließen. Sowohl für den Nelder-Mead- als auch den Levenberg-Marquardt-Algorithmus ist in deren Grundversionen allerdings keine Beschränkung der Parameter vorgesehen.

Im Rahmen dieser Arbeit wurden beide Algorithmen für dieses Problem derart angepasst, dass vom Anwender einstellbare Maximalwerte nicht überschritten werden können. In jeder Iteration wird hierbei geprüft, ob sich die Parameter in ihrem gültigen Wertebereich befinden. Ist dies nicht der Fall werden die betroffenen Parameter bei Überschreitung durch die zugehörigen Maximalwerte, bei Unterschreitung durch deren Minimalwerte, ersetzt.

Durch die Beschränkung der Parameter wird zudem vermieden, dass sich der Algorithmus in Bereichen mit zu großen Funktionswerten bewegt. Ohne Beschränkung kann es durch den exponentiellen Charakter der realen Diodenfunktion dazu kommen, dass Parametersätze als Lösung getestet werden, deren Funktionswert den Wertebereich der Maschinenrepräsentation einer Fließkommazahl mit doppelter Genauigkeit überschreitet.

3.5.5. Logarithmierte Kurvenanpassung

Wie sich bei der Auswertung von verschiedenen Diodenkennlinien zeigte, liefert eine Kurvenanpassung im linearen Maßstab häufig keine zufriedenstellenden Ergebnisse, wenn die Resultate im logarithmiertem Maßstab betrachtet werden (vgl. Abschnitt 4.1). Abhilfe kann diesbezüglich die Abschwächung des exponentielle Charakter der Diodenfunktion (vgl. Gl. 2.6) schaffen. Im Verlaufe dieser Arbeit wurde dementsprechend ein weiterer Ansatz entwickelt, bei welchem eine logarithmierte Variante der Diodenfunktion

$$I_{log} = \log \left\{ I_{Diff,0} \left[\exp \left(\frac{q \left(U - I_{log}^{10} R_s \right)}{n_1 k T} - 1 \right) \right] + I_{GR,0} \left[\exp \left(\frac{q \left(U - I_{log}^{10} R_s \right)}{n_2 k T} - 1 \right) \right] + \frac{U - I_{log}^{10} R_s}{R_p} - I_{ph} \right\} \quad (3.25)$$

in Verbindung mit einem logarithmiertem Datensatz, zum Einsatz kommt. Durch die, mit der Logarithmierung verbundene, Veränderung der Struktur der Zielfunktion kommt es zu einer unterschiedlichen Kurvenanpassung. Während bei einer nicht-logarithmierten Betrachtung die normale Diodenfunktion häufig eine tendenziell bessere Kurvenanpas-

sung liefert, wendet sich das Bild bei logarithmischer Auftragung. Einen direkten Vergleich der beiden Verfahren liefert Abschnitt 4.1.

Da der Logarithmus lediglich für einen positiven Bereich definiert ist, darf dessen Argument in Gleichung 3.25 keine negativen Werte annehmen. Dies ist jedoch lediglich für Kennlinien ohne Photostrom der Fall. Aus diesem Grund werden Kennlinien, welche über einen Photostrom verfügen, schlicht über einen Offset in den positiven Bereich verschoben um eine reibungslose Kurvenanpassung zu ermöglichen.

4. Praktische Auswertung von Diodenkennlinien

Nach Betrachtung der theoretischen Grundlagen im vorangegangenen Kapitel widmet sich dieser Abschnitt der praktischen Auswertung von Diodenkennlinien. Herangezogen wird hierfür die für diese Arbeit geschriebene Applikation *Parameter Extractor*. Die Anwendung bietet grundsätzlich zwei verschiedene Optimierungsalgorithmen, die zur Kurvenanpassung eingesetzt werden können. Zum einen ist dies der Levenberg-Marquardt- zum anderen der Nelder-Mead-Algorithmus. Die beiden Algorithmen stellen die Basis für die nachfolgenden Betrachtungen dar.

Für die Evaluierung der Ergebnisse wird im Folgenden die normalisierte Variante der Wurzel aus mittlerem quadratischen Fehler (engl. Normalized Root Mean Square Error, NRMSE) herangezogen. Dieser wird als

$$\text{NRMSE} = \frac{\text{RMSE}}{y_{\max} - y_{\min}} \quad (4.1)$$

mit

$$\text{RMSE} = \sqrt{\frac{\text{LSQ}}{\text{Anz. Messwerte}}} \quad (4.2)$$

definiert. LSQ stellt hierbei die Summe der kleinsten Quadrate entsprechend Gleichung 3.2 dar, während y_{\max} und y_{\min} den minimalen bzw. maximalen Messwerten entsprechen.

Der angegebene Fehler beinhaltet hierbei eine Reihe von Fehlerquellen:

Modellfehler Diskrepanzen zwischen dem zur Optimierung herangezogenen Modell und der gemessenen Kurve (vgl. 4.3)

Messfehler Abweichung zwischen realem und gemessenem Wert bei der Messung der von Strom und Spannung. Zum Beispiel durch endliche Messgenauigkeit oder Anwendungsfehler.

Diskretisierungsfehler Fehlerbehaftete Umwandlung von kontinuierlichen Strom- bzw. Spannungswerte in diskrete Werte.

Numerische Fehler (truncation error) Durch die Verwendung der Taylor-Approximation ergeben sich für den Levenberg-Marquardt-Algorithmus Abweichungen zwischen

der verwendeten und der tatsächlichen Zielfunktion, welche zu einem weiteren Fehler führen. Des Weiteren sind Abweichungen durch die endliche Genauigkeit reeller Zahlen in einem Computer-System (Maschinenrepräsentation) in diesem Fehler enthalten.

Anpassungsfehler Fehler der bedingt durch den Optimierungsalgorithmus selbst hervorgerufen wird. Abweichung zwischen optimaler und tatsächlich gefundener Lösung. Zum Beispiel durch das Auffinden eines lokalen statt globalen Minimum.

Der im folgenden angegebene NRMSE-Wert stellt somit kein ausschließliches Bewertungskriterium für die Qualität des Fittingsprozesses dar, sondern beinhaltet weitere Fehlerquellen, welche es zu berücksichtigen gilt.

4.1. Vergleich von linearer und logarithmischer Kurvenanpassung

Neben der Kurvenanpassung auf Basis der unveränderten Messdaten wurde im Zuge dieser Arbeit auch eine Anpassungsvariante auf den logarithmischen Datensatz implementiert, wie in Kapitel 3.5.5 beschrieben. Die Kurvenanpassung auf dem unveränderten Datensatz wird im Folgenden als lineare, jene auf dem logarithmierten Datensatz als logarithmische Anpassung bezeichnet. Während bei der Betrachtung im linearen Maßstab in der Regel beide Varianten gute Ergebnisse liefern, sind die Ergebnisse der linearen Anpassung bei Auftragung im logarithmischen Maßstab häufig unbrauchbar.

Die Abbildungen 4.1 und 4.2 zeigen exemplarisch die Auswertung einer Silicium-Diode mittels des Levenberg-Marquardt-Algorithmus in linearer und die Abbildungen 4.3 und 4.4 in logarithmischer Darstellung. Bei Betrachtung im linearen Maßstab liefern sowohl das lineare als auch das logarithmische Anpassungsverfahren in diesem Beispiel mit einem NRMSE von 0,51% bzw. 0,98% gute Ergebnisse. Bereits an den Parametern lässt sich jedoch erkennen, dass beide Verfahren unterschiedliche Lösungen des Problems gefunden haben. Während die extrahierten Parameter für I_{Diff} , R_S und n_1 bei beiden Varianten noch in ähnlichen Regionen liegen, unterscheiden sich die anderen Parameterwerte teils in mehreren Zehner-Potenzen. Dieser Umstand ist darin begründet, dass die Änderung jener Parameter ab bestimmten Grenzwerten, keinen nennenswerten Einfluss mehr auf die simulierte Kennlinie in linearer Darstellung hat. Sowohl der Nelder-Mead-, als auch der Levenberg-Marquardt-Algorithmus tendieren in solchen Fällen dazu gegen die hinterlegten Grenzwerte zu laufen.

Im logarithmischen Maßstab werden die beschriebenen Mängel der linearen Anpassung besonders deutlich (vgl. Abb. 4.3 und 4.4). Während bei linearer Anpassung die modellierte Kurve korrekt in die höheren Funktionswerte eingepasst wurde, weicht die Modellkurve bei kleineren Funktionswerten deutlich von den Messwerten ab. Abhilfe

kann in solchen Fällen eine logarithmische Anpassung schaffen, da kleinere Messwerte deutlich stärker gewichtet werden, als dies bei der linearen Anpassung der Fall ist. Bereiche, die bei der linearen Anpassung kaum Einfluss auf das Ergebnis hatten, fließen nun deutlich stärker in die Parameter-Extraktion ein. Dies führt dazu, dass die zuvor problematischen Parameter auch trotz kleiner Werte noch einen relevanten Einfluss auf die simulierte Kurve haben.

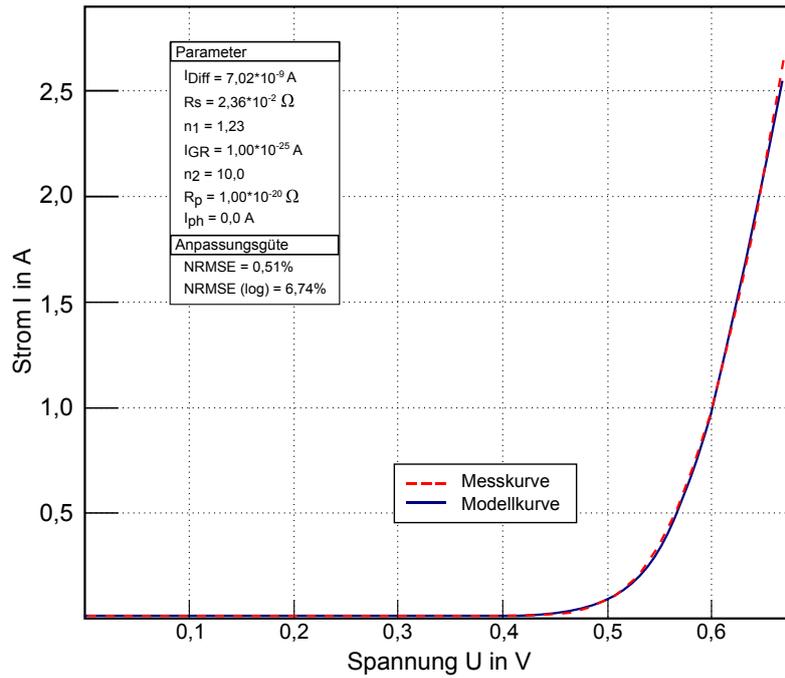


Abbildung 4.1.: Lineare Kurvenanpassung einer Silicium-Diode in linearer Auftragung

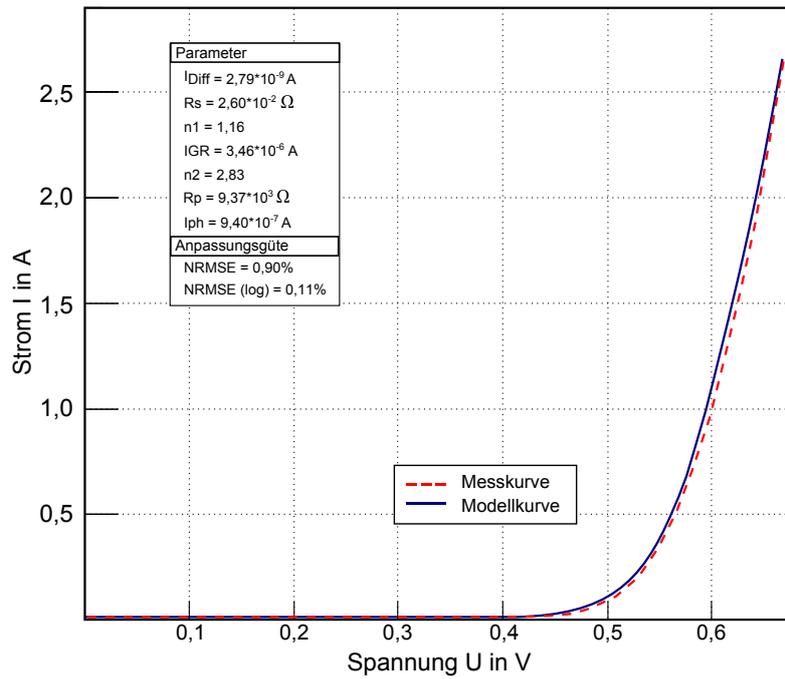


Abbildung 4.2.: Logarithmische Kurvenanpassung einer Silicium-Diode in linearer Auftragung

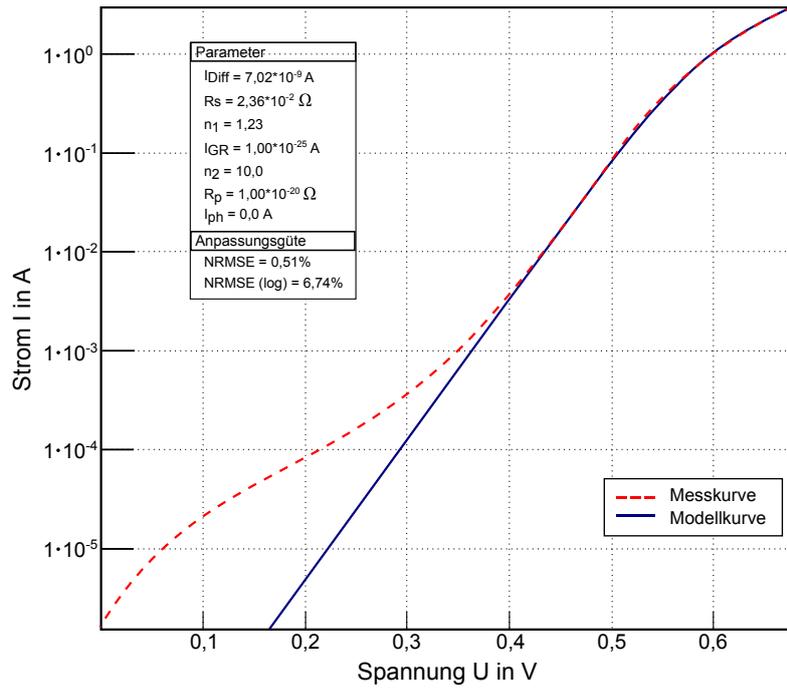


Abbildung 4.3.: Lineare Kurvenanpassung einer Silicium-Diode in logarithmischer Auftragung

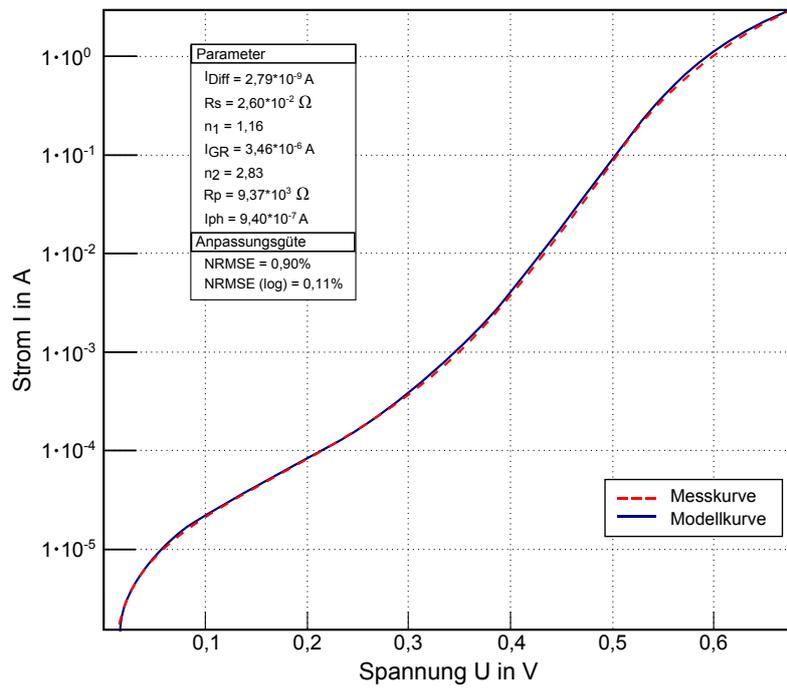


Abbildung 4.4.: Logarithmische Kurvenanpassung einer Silicium-Diode in logarithmischer Auftragung

4.2. Vergleich von Levenberg-Marquardt- und Nelder-Mead-Algorithmus

Im Rahmen dieser Arbeit wurden mit dem Levenberg-Marquardt- und Nelder-Mead-Algorithmus zwei verschiedene Optimierungsalgorithmen implementiert, wie sie in den Abschnitten 3.3 und 3.4.4 vorgestellt wurden. Im Folgenden wird die Anpassungsgüte der beiden Algorithmen für die Auswertung verschiedener Kennlinien untersucht. Alle Auswertungen wurden initial mit den gleichen Startparametern $I_{Diff,0} = 1 \cdot 10^{-6} A$, $R_s = 0,1 \Omega$, $n_1 = 1,0$, $I_{GR,0} = 1 \cdot 10^{-6} A$, $n_2 = 2,0$, $R_p = 1000 \Omega$ und $I_{ph} = 1 \cdot 10^{-5} A$ angestoßen. Jede Auswertung beinhaltet hierbei drei Optimierungsdurchläufe, wie sie in Abschnitt 3.5.3 beschrieben werden. Die Ergebnisse des aktuellen Durchlaufes werden hierbei zufällig in einem Bereich von $\pm 30 \%$ variiert und als Startwert für den nächsten Durchlauf herangezogen. Während in der Applikation grundsätzlich Zufallszahlen für die Variation der Parameter herangezogen werden, wurde für die nachfolgenden Betrachtung fix hinterlegte genutzt um gleiche Bedingungen für beide Algorithmen zu schaffen. Die maximale Anzahl an Iterationen pro Durchlauf wurde für beide Algorithmen auf 1000 festgesetzt.

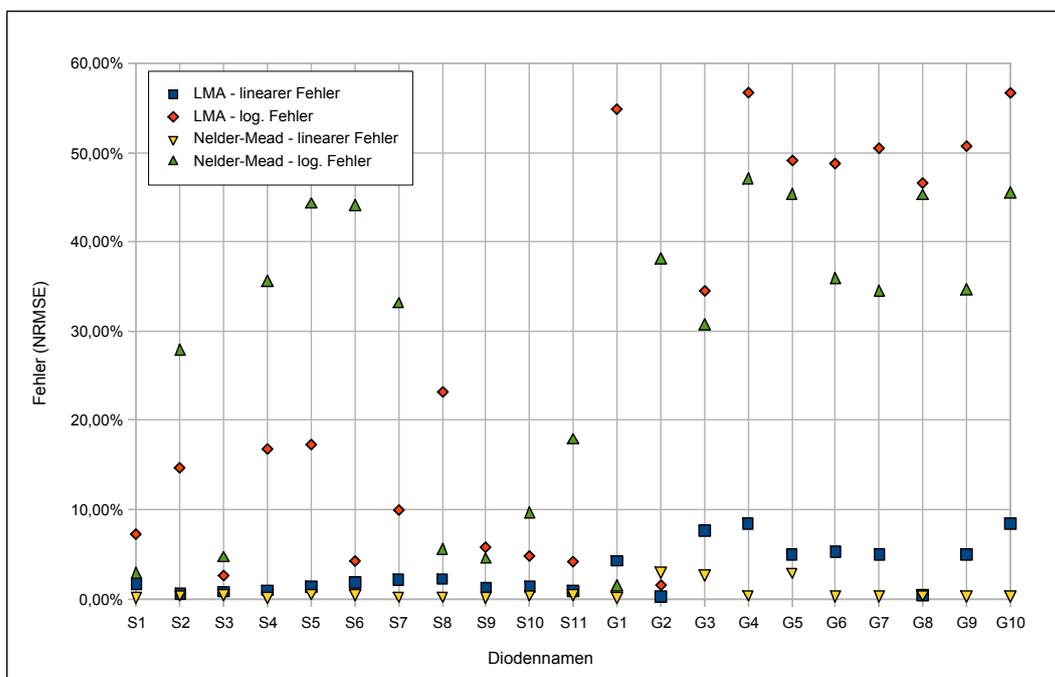


Abbildung 4.5.: Fehler bei der Auswertung verschiedener Silicium- (Sx) und Germaniumdioden (Gx) mittels linearer Kurvenanpassung unter Verwendung des Levenberg-Marquardt-Algorithmus (LMA) und des Nelder-Mead-Algorithmus.

Das Diagramm 4.5 zeigt eine Übersicht der Auswertungen der verschiedenen Kenn-

linien von Silicium- und Germanium-Dioden, welche mittels linearer Kurvenanpassung ermittelt wurden. Bei Betrachtung im linearen Maßstab liefern beide Algorithmen für Silicium-Dioden durchwegs brauchbare Ergebnisse mit einem NRMSE unterhalb von drei Prozent. Während dem Nelder-Mead-Algorithmus auch die Kennlinien der Germanium-Dioden keine Probleme bereiten, schneidet der Levenberg-Marquardt-Algorithmus diesbezüglich schlechter ab. Für einige dieser Kennlinien konnte hier mit Fehlerwerten oberhalb von fünf Prozent bei den vorgegeben Startwerten keine zufriedenstellenden Lösungen gefunden werden. Wie bereits in Kapitel 3.5.5 angedeutet, wird zudem unabhängig vom Algorithmus deutlich, dass sich die lineare Kurvenanpassung in keiner Weise für eine Parameter-Auswertung im logarithmischen Maßstab eignet. Die NRMSE-Werte im logarithmischen Maßstab liegen nahezu durchgehend im zweistelligen Prozent-Bereich.

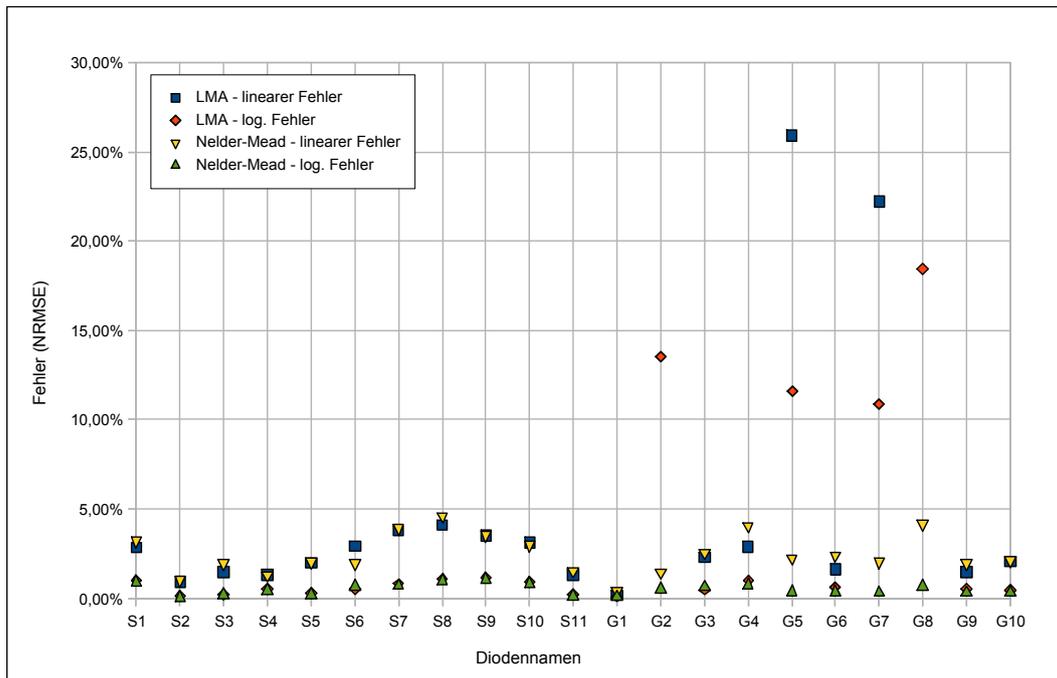


Abbildung 4.6.: Fehler bei der Auswertung verschiedener Silicium- (Sx) und Germaniumdioden (Gx) mittels logarithmischer Kurvenanpassung unter Verwendung des Levenberg-Marquardt-Algorithmus (LMA) und des Nelder-Mead-Algorithmus.

Eine etwas andere Ausgangsposition ergibt sich bei der Betrachtung der Auswertungen mittels logarithmischer Kurvenanpassung (siehe Tabelle 4.6). Die Auswertungen des Levenberg-Marquardt-Algorithmus weisen insbesondere für Silicium-Dioden eine tendenziell leicht bessere Anpassungsgüte im logarithmischen Maßstab auf, als jene mit dem Nelder-Mead-Algorithmus. Im Falle der Germanium-Dioden sind jedoch auch in dieser Kombination einzelne Auswertungen zu erkennen, in denen keine sinnvolle Lösung mit den gewählten Startparametern gefunden wurde.

Insgesamt lässt sich für keinen der beiden Algorithmen eine eindeutige Empfehlung aussprechen. Je nach Anwendungsgebiet, können beide Algorithmen Stärken und Schwächen aufweisen. Für die ausgewerteten Kennlinien zeigte sich der Nelder-Mead-Algorithmus in Zusammenspiel mit den vorgegebenen Startwerten etwas robuster. Gerade bei den Auswertungen der Germanium-Dioden konnte der Levenberg-Marquardt-Algorithmus in einigen Fällen keine Lösung finden. Dafür zeigten die Auswertungen im logarithmischen Maßstab eine hohe Genauigkeit. In der Praxis stellt sich in vielen Fällen eine Kombination von beiden Algorithmen als sinnvoll heraus. Je nach Problemstellung kann hierfür einer der beiden Algorithmen für eine grobe Parameterbestimmung herangezogen werden, während der andere anschließend diese Lösung weiter verbessert.

4.3. Einfluss der Startparameter

Sämtliche Optimierungsroutinen für nichtlineare Problemstellungen haben eines gemein: Es handelt sich um iterative Verfahren, bei denen eine Anfangslösung schrittweise verbessert wird. Schritt für Schritt wird die aktuelle Lösung verbessert, bis nach Möglichkeit ein Minimum aufgefunden wird. Im Falle der Kurvenanpassung kann die Zielfunktion eine Vielzahl von Minima durchlaufen. Die Wahl der Startparameter kann hierbei einen entscheidenden Einfluss darauf haben, welches Minimum, oder ob überhaupt ein Minimum, aufgefunden wird.

Allerdings ist die Wahl der Anfangslösung ein mitunter schwieriges Unterfangen für das kein universeller Lösungsansatz existiert. Durch die Einschränkung der Parameter in gewisse Wertebereiche kann das Problem jedoch deutlich entschärft werden, da so eine große Anzahl von Minima bereits im Vorfeld ausgeschlossen werden kann. Die Abbildungen 4.7 und 4.8 zeigen die Problematik exemplarisch anhand einer Germanium-Diode. Diese wurde unter grundsätzlich gleichen Bedingungen – jedoch mit unterschiedlichen Startparametern – ausgewertet, woraus zwei verschiedene Lösungen resultierten.

Um den Rahmen dieser Arbeit nicht zu sprengen, beschränkt sich die Applikation auf die Vorgabe vordefinierter Startparameter. Diese wurden experimentell ermittelt und zeigten sich bei der Auswertung verschiedener Silicium- und Germanium-Dioden als zuverlässige Ausgangslösung. Weiterhin wird dem Anwender die Möglichkeit überlassen, die Startparameter unter Zuhilfenahme einer Vorschau-Funktion beliebig anzupassen.

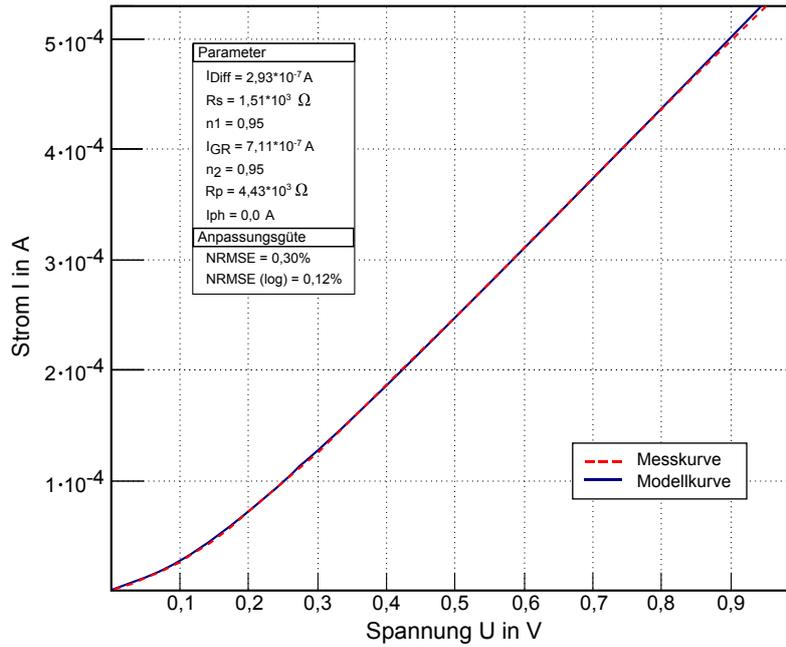


Abbildung 4.7.: Auswertung einer Germanium Diode; Startparameter: $I_{Diff,0} = 1,0 \cdot 10^{-6} A$, $R_s = 0,1 \Omega$, $n_1 = 1,0$, $I_{GR,0} = 1,0 \cdot 10^{-6} A$, $n_2 = 2,0$, $R_p = 1000 \Omega$ und $I_{ph} = 1,0 \cdot 10^{-5} A$

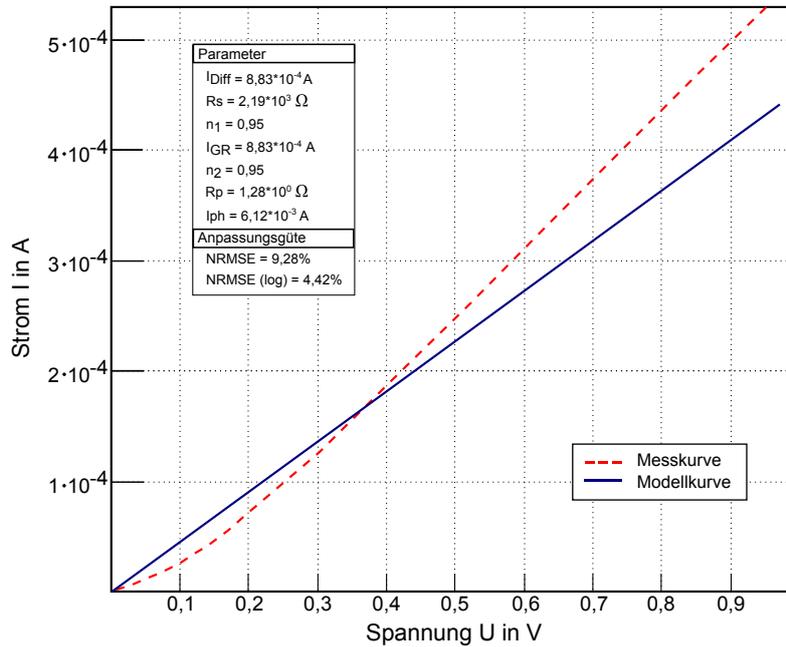


Abbildung 4.8.: Auswertung einer Germanium Diode; Startparameter: $I_{Diff,0} = 5,0 \cdot 10^{-4} A$, $R_s = 1 \cdot 10^3 \Omega$, $n_1 = 1,0$, $I_{GR,0} = 1,0 \cdot 10^{-4} A$, $n_2 = 1,0$, $R_p = 1 \cdot 10^{-2} \Omega$ und $I_{ph} = 3 A$

4.4. Auswirkungen von Modellabweichungen

Von essentieller Bedeutung bei der Kurvenanpassung ist das zugrundeliegende mathematische Modell. Auch ein perfekter Optimierungsalgorithmus kann keine zufriedenstellende Lösung des Kurvenanpassungsproblems finden, wenn sich das mathematische Modell nicht für die gegebenen Messwerten eignet. Ebenfalls sollte man sich diesbezüglich vor Augen führen, dass ein Optimierungsalgorithmus ein fehlerhaftes Modell – soweit wie möglich – über eine Variation der Parameter kompensiert.

Zur Veranschaulichung wird im Folgenden die Kennlinie einer Silicium-Diode betrachtet, welche einen niedrigen Photostrom-Anteil enthält. Abbildung 4.9 zeigt die Auswertung der Kennlinie unter Nutzung des Modells der realen Diode nach Gl. 2.6. Die Auswertung in Abbildung 4.10 wurde prinzipiell unter gleichen Bedingungen erstellt, allerdings kam ein vereinfachtes Diodenmodell ohne Berücksichtigung des Photostroms I_{Ph} zum Einsatz. Neben der schlechteren Anpassungsgüte (höherer NRMSE-Wert), ist auch die Abweichung der anderen Parameter im Vergleich zur Anpassung mit vollwertigen Diodenmodell deutlich erkennbar. Ein falsches Diodenmodell kann in diesem Zusammenhang also nicht nur für einen allgemein schlechteren NRMSE-Wert sorgen, sondern auch für eine Verfälschung der extrahierten Parameter.

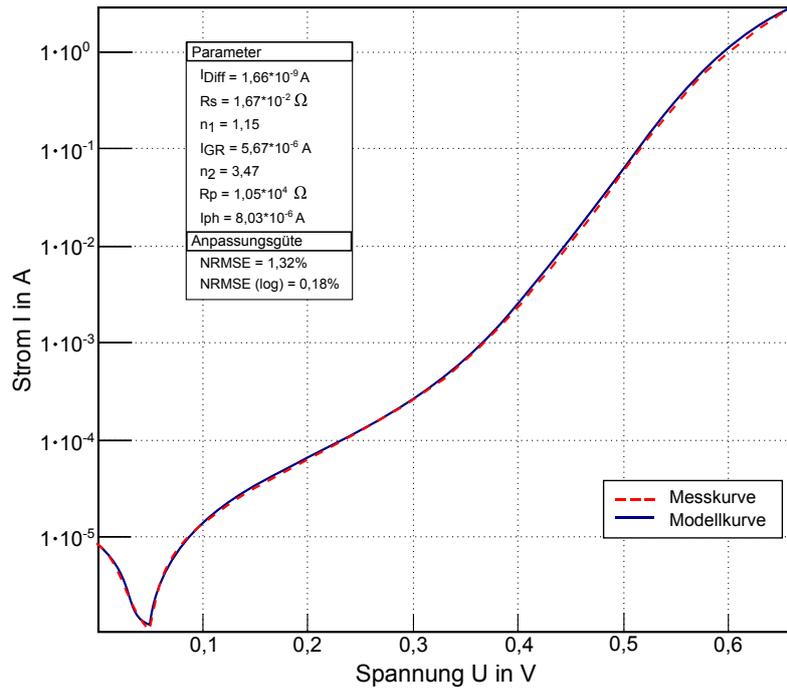


Abbildung 4.9.: Auswertung einer Silizium-Diode - Modell mit Photostrom

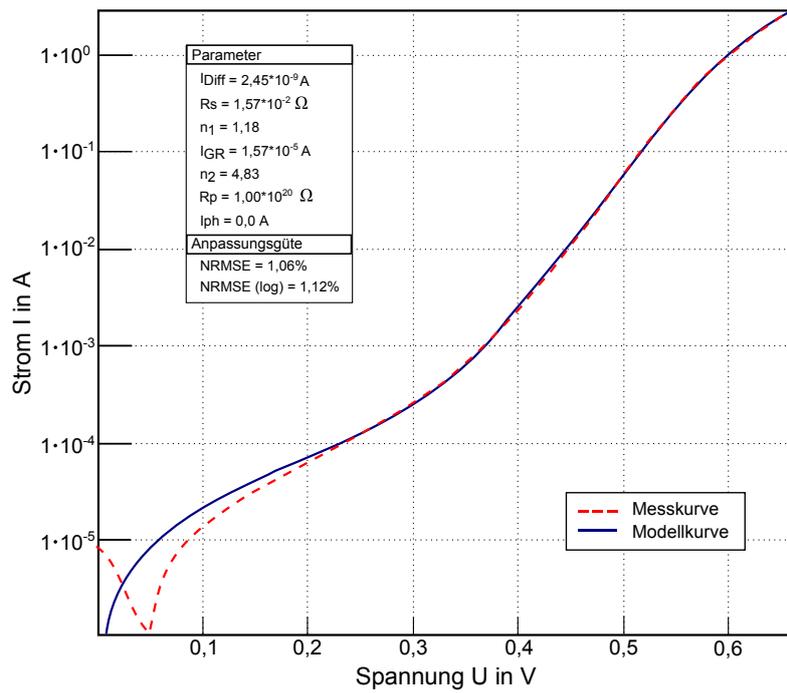


Abbildung 4.10.: Auswertung einer Silizium-Diode - Modell ohne Photostrom

5. Fazit und Ausblick

Ziel der vorliegenden Arbeit war die Entwicklung einer Computer-Applikation, welche den Anwender bei der Auswertung von Diodenkennlinien unterstützt. Um die Systemvoraussetzungen so gering wie möglich zu halten, wurde auf die Nutzung proprietärer, kostenpflichtiger Anwendungen - wie beispielsweise der Mathematik-Software MATLAB - verzichtet. Stattdessen kam mit Java eine plattformunabhängige Sprache zum Einsatz, welche die Nutzung der Anwendung auf sämtlichen Betriebssystemen ermöglicht, für die eine Java-Laufzeitumgebung existiert. Die intuitive, grafische Oberfläche der entwickelten Anwendung visualisiert neben den Messpunkten auch die modellierte Kurve und unterstützt den Anwender auf diese Weise interaktiv bei der Parameter-Extraktion aus der gemessenen Kennlinie.

Als mathematisches Modell für die Auswertung kommt ein Zwei-Dioden-Modell zum Einsatz, welches den Vorwärtsbereich einer Halbleiterdiode beschreibt. Das Modell berücksichtigt neben dem Diffusionsstrom auch den Generations-/Rekombinationsstrom, sowie den Parallelwiderstand, Serienwiderstand und Photostrom einer Halbleiterdiode. Die Berücksichtigung der genannten Merkmale erfolgt im mathematischen Modell hierbei über insgesamt sieben verschiedene Parameter. Über die Parameter wird die Modellkurve unter Nutzung des Prinzips der kleinsten Quadrate im Zusammenspiel mit einem Optimierungsalgorithmus in die Messwerte eingepasst. Für den Optimierungsprozess stellt die Anwendung hierbei mit dem Nelder-Mead- und dem Levenberg-Marquardt-Algorithmus zwei verschiedene Methoden zur Verfügung. Zusätzlich zur Optimierung im linearen Maßstab wurde eine Optimierungsmethode entwickelt, welche die Kurvenanpassung im logarithmierten Maßstab vornimmt.

Bei der Prüfung der Praxistauglichkeit der Implementierungen der beiden Algorithmen nach Nelder-Mead und Levenberg-Marquardt konnten beide Varianten mit guten Ergebnissen aufwarten. Mit beiden Algorithmen konnten Auswertungen mit einer Anpassungsgüte von bis zu 0,1 Prozent (NRMSE) vorgenommen werden, wenn die Fehlerwerte im jeweils zugehörigen Maßstab betrachtet werden (logarithmischer Maßstab bei logarithmischer Kurvenanpassung und linearer Maßstab bei linearer Kurvenanpassung). Insbesondere bei der Auswertung von Germanium-Dioden zeigte sich jedoch der Nelder-Mead-Algorithmus als robuster als der Levenberg-Marquardt-Algorithmus. Während letzterer in einigen Fällen mit Fehlerwerten im Bereich von 10 bis 20 Prozent keine sinnvolle Lösung finden konnte, blieb der Fehler bei den Auswertungen mittels des Nelder-Mead-Algorithmus grundsätzlich unterhalb von fünf Prozent.

Im Falle der linearen Kurvenanpassung zeigt sich, dass die Fehlerwerte bei der Betrachtung im linearen Maßstab zwar in sehr niedrigen Bereichen liegen, im logarithmischen Maßstab jedoch die Modellkurve meist sehr stark von der gemessenen Kennlinie abweicht. Fehlerwerte von über 30 Prozent sind diesbezüglich, unabhängig vom Algorithmus, keine Seltenheit. Abhilfe kann hier die, ebenfalls implementierte, logarithmierte Kurvenanpassung schaffen. Mittels dieser Anpassungsmethode lassen sich mit beiden Algorithmen Parameter aus der Kennlinie extrahieren, welche für beide Darstellungsarten plausible Ergebnisse liefern.

Hinsichtlich des Software-Aufbaus lag bei der Entwicklung ein wesentlicher Schwerpunkt auf der Erweiterbarkeit der Anwendung. Im Zuge zukünftiger Entwicklungen können hierbei sowohl neue Algorithmen als auch neue Modelle für Dioden oder anderer Bauteile problemlos eingepflegt werden. Die bereitgestellte Software stellt ein Grundgerüst dar. Dieses kann ohne tiefere Kenntnisse des Programmcodes um neue Modelle und Algorithmen erweitert werden.

Erweiterungen sind in diesem Hinblick unter anderem im Bereich des verwendeten mathematischen Diodenmodells vorstellbar. Modellverbesserungen im Vorwärtsbereich sind durch die Implementierung der Hochinjektion denkbar, welche derzeit vernachlässigt wird. Zudem könnte über ein weiteres Modell auch der Rückwärtsbereich einer Halbleiterdioden umgesetzt werden, welches auch die Durchbruchmechanismen abdeckt.

Verbesserungen beim Optimierungsprozess sind durch die Wahl besser geeigneter Startparameter zu erwarten. Derzeit ist lediglich ein Satz empirisch ermittelter Startparameter, welcher für eine Vielzahl von verschiedenen Kennlinien gute Ergebnisse lieferte, in der Applikation hinterlegt. Diesbezüglich wäre als Erweiterung im Zuge zukünftiger Entwicklungen eine automatisierte, kennlinienspezifische Voreinschätzung der Parameter ein vielversprechender Ansatz. Die Vorauswahl könnte hierbei beispielsweise über eine grafische Analyse der Kennlinie im logarithmischen Maßstab erfolgen.

Ein anderer Ansatz, welcher auch in Kombination mit der bereits angesprochenen Parameter-Voreinschätzung möglich wäre, ist die Verfolgung globaler Optimierungsstrategien. Derzeit wurde diesbezüglich bereits ein einfacher Variante implementiert, welcher die Optimierungsalgorithmen nach einem Durchlauf mit leicht variierten Lösungsparametern neu startet. Das Multigrid-Verfahren, evolutionäre Algorithmen oder Simulated Annealing sind beispielhaft genannte Verfahren, welche vielversprechende Ansätze in diesem Bereich darstellen.

A. Software-Aufbau und Funktionsweise der Applikation

Dieses Kapitel dient der grundsätzlichen Beschreibung, der für diese Arbeit programmierten Applikation *Parameter Extractor*, welche einen zentralen Punkt dieser Arbeit darstellt. Neben der grundlegenden Funktion die Parameter aus einer Kennlinie zu extrahieren, war ein weiteres Ziel das Schaffen einer Basis, in welche zukünftige Entwicklungen leicht integriert werden können. Um diese Vorgabe zu verwirklichen, kam mit Java eine objektorientierte Programmiersprache zum Einsatz, welche sich neben der Objektorientierung auch durch seine Plattformunabhängigkeit auszeichnet.

A.1. UML-Modellierung

Die Unified Modeling Language (UML) stellt eine Sprache zur Modellierung von Software dar. Das Klassendiagramm der UML im Speziellen dient zur grafischen Darstellung von Klassen und Schnittstellen, sowie deren Beziehung untereinander. Das Klassendiagramm wird im Folgenden dazu verwendet, um einen Überblick über den grundsätzlichen Aufbau des *Parameter Extractors* zu schaffen. Abbildung A.1 zeigt ein UML-Klassendiagramm, welches die wesentlichen Merkmale der Applikation enthält. Nicht berücksichtigt sind die Elemente der grafischen Oberfläche und Funktionen, die für die Schilderung der grundlegenden Funktionsweise nicht relevant sind. Für eine vollständige Code-Dokumentation wird auf den beigelegten Datenträger [9] verwiesen.

Infolge des objektorientierten Ansatzes ist die Anwendung in verschiedene Klassen strukturiert. Die Klassen-Instanzen interagieren untereinander um die gewünschte Funktionalität bereitzustellen. Wesentliche Bestandteil der Applikation sind die beiden Klassen *FittingModel* und *DiodeModel*.

Über die abstrakte Klasse *DiodeModel* wird das verwendete Diodenmodell spezifiziert. Die konkrete Implementierung erfolgt über die beiden Subklassen *DiodeModelReal* und *DiodeModelRealLog10*. Während Erstere das Modell der realen Diode nach Gleichung 2.6 umsetzt, stellt Letztere die Implementierung der logarithmierten Diodenfunktion nach Gleichung 3.25 dar. Neben der Berechnung des Funktionswertes (Funktion *calculateValue*) stellt jede von *DiodeModel* abgeleitete Klasse auch die Berechnung der partiellen Ab-

leitung nach einem bestimmten Parameter (Funktion *calculateGrad*) zur Verfügung. Wie in Kapitel 3.5.1 beschrieben, ist eine direkte Berechnung der Funktionswerte aufgrund der impliziten Darstellungsform der realen Diode nicht möglich. Aufgrund dessen nutzen beide implementierten Diodenmodelle den Van Wijngaarden-Dekker-Brent-Algorithmus (Klasse *RootFindingDekkerBrent*) zum Lösen der Gleichung. Da dies prinzipiell ein Verfahren zur Nullstellensuche ist, wird die Diodenfunktion als $f(x, y) = 0$ über die Klasse *ZeroDifferenceFunc* bereitgestellt.

Die Bereitstellung der unterschiedlichen Optimierungsalgorithmen erfolgt über die abstrakte *FittingModel*-Klasse. Der Levenberg-Marquardt-Algorithmus wird hierbei von der Klasse *FittingModelLMA* umgesetzt und die Nelder-Mead-Methode über die Klasse *FittingModelSimplex* realisiert. Jede Klasse auf Basis der *FittingModel*-Klasse muss eine Funktion zur Ausführung des Optimierungsalgorithmus (Funktion *fit*), sowie Methoden zur Festlegung der Startparameter (Funktion *setStartGuess*), Diodenmodells (Funktion *setDiodeModel*) und zur Fixierung bestimmter Parameter (Funktion *fixParameter*) bereitstellen.

Die Klasse *ParameterExtractor* stellt eine weitere Abstraktionsschicht dar und kann für die Parameter Extraktion herangezogen werden. Das gewünschte Diodenmodell, sowie der Algorithmus zur Optimierung können über die entsprechenden set-Methoden (Methoden *setDiodeModel* und *setFittingModel*) festgelegt werden. Zusätzlich kann die Anzahl der Durchläufe (Methode *setRestarts*) festgelegt werden. Der verwendete Datensatz wird über die Klasse *DataContainer* definiert (Methode *setDataContainer*).

Die Klasse *DataContainer* verfügt zusätzlich über verschiedene Operationen zur Datenmanipulation. Über das Listener-Prinzip [10] können andere Klassen über Änderungen am Datensatz informiert werden. Dieses Prinzip wird beispielsweise von der Zeichenfunktion des Plotters genutzt um Änderungen am Datensatz direkt auch grafisch anzuzeigen.

A.2. Erweiterbarkeit

Die Applikation wurde im Hinblick auf Erweiterung für zukünftige Entwicklung geschrieben. Aus diesem Grund handelt es sich bei den beiden zentralen Klassen *FittingModel* und *DiodeModel* um so genannte abstrakte Klassen. Abstrakte Klassen können nicht instantiiert werden, es können also keine Objekte von diesen Klassen erzeugt werden. Sie verfügen über mindestens eine abstrakte Methode. Abstrakte Methoden beinhalten lediglich eine Methodensignatur aber keine konkrete Implementierung. Subklassen, die von der abstrakten Klasse abgeleitet werden, müssen sämtliche abstrakten Methoden überschreiben und implementieren. Auf diese Weise können über abstrakte Klassen grundlegende Eigenschaften ihrer Unterklassen festgelegt werden, ohne die konkrete Implementierung vorzugeben.

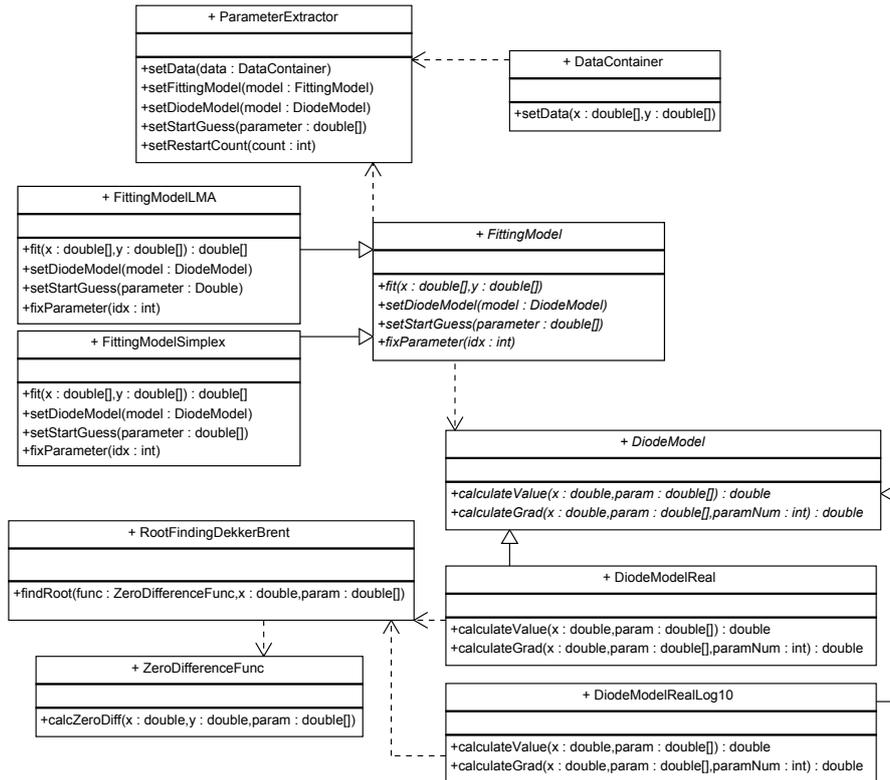


Abbildung A.1.: UML-Diagramm des *Parameter Extractors*

Für zukünftige Diodenmodelle muss dementsprechend lediglich eine Subklasse der abstrakten Klasse *DiodeModel* erstellt werden (vgl A.2). In der Subklasse wird die konkrete Implementierung des Dioden Modells bestimmt. Da die beiden Methoden *calculateValue* und *calculateGrad* als abstrakte Methoden definiert sind, müssen diese in den Subklassen überschrieben und implementiert werden. Wie die konkrete Implementierung aussieht, ist hierbei dem Entwickler überlassen. Gleiches gilt für die Klasse *FittingModel*, welche ebenfalls in abstrakter Form vorliegt. Neue Optimierungsalgorithmen müssen über die abstrakte Methode *fit* in einer abgeleiteten Klasse umgesetzt werden.

Durch die Definition der abstrakten Klassen können sowohl neue Optimierungsalgorithmen als auch Diodenmodelle implementiert werden, ohne Änderungen an der grundsätzlichen Software-Struktur vorzunehmen. Für ein alternatives Diodenmodell muss dementsprechend lediglich das Diodenmodell in der *FittingModel*-Klasse gesetzt werden. Gleiches gilt für neue Varianten des *FittingModels*, welche dementsprechend in der *ParameterExtractor*-Klasse eingetragen werden müssen.

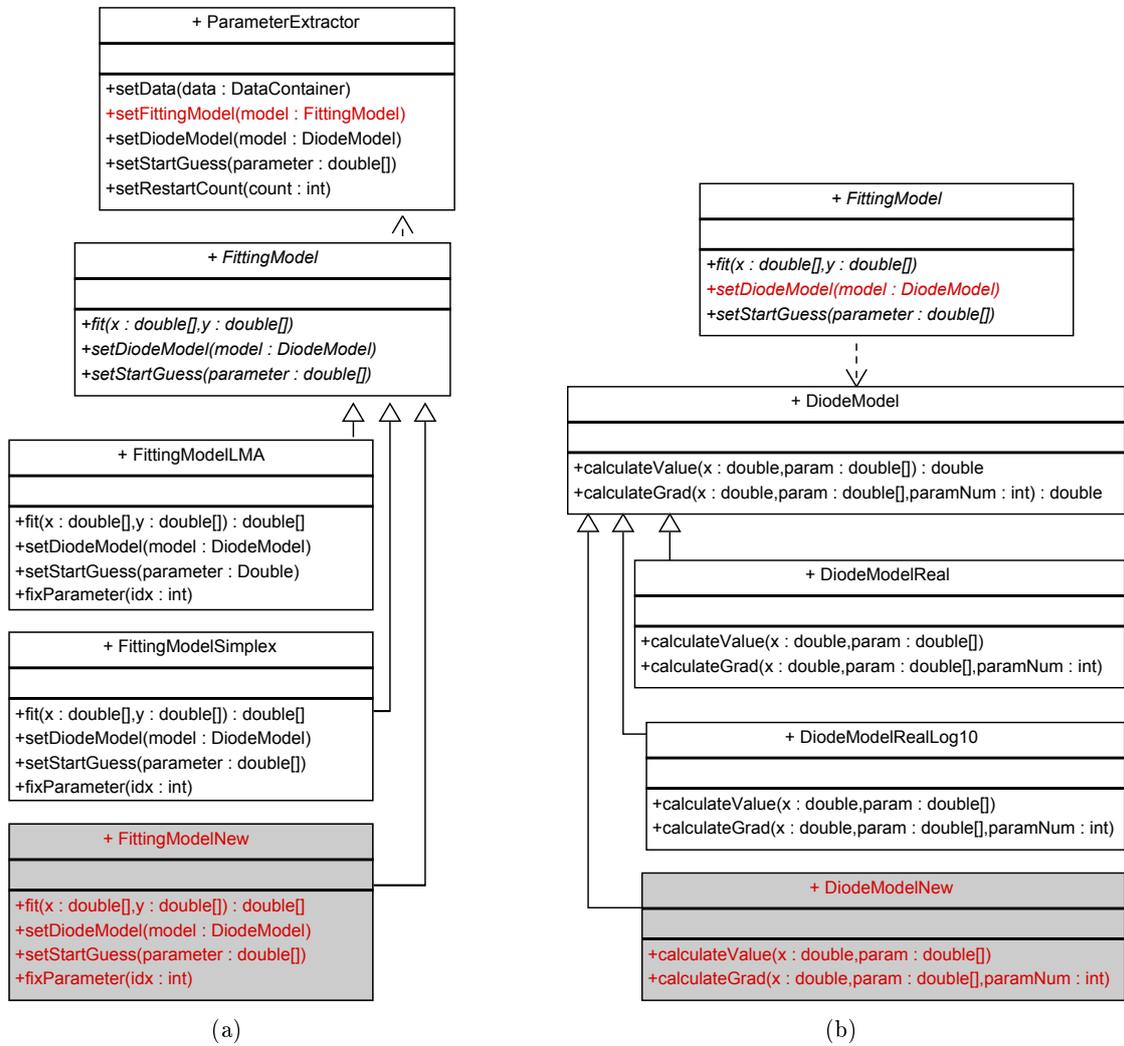


Abbildung A.2.: Exemplarisches UML-Diagramm mit zukünftigen Modellen und Algorithmen (rot markiert)

Literaturverzeichnis

- [1] Raymond A. Serway and John W. Jewett. *Principles of Physics*. Saunders College Publishing, 2. edition, 1997.
- [2] Simon M. Sze and K. Ng Kwok. *Physics of Semiconductor Devices*. John Wiley and Sons, 3. edition, 2006.
- [3] W. Shockley. The theory of p-n junctions in semiconductors and p-n junction transistors. *Bell System Technical Journal*, 28:435–489, 1949.
- [4] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 2nd edition, 1999.
- [5] Th. Jongen Hubertus, Klaus Meer, and Eberhard Triesch. *Optimization Theory*. Kluwer Academic Publishers, 2004.
- [6] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 2007.
- [7] Reiner Horst. *Introduction to Global Optimization (Nonconvex Optimization and Its Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.
- [8] Harald Köstler. *A Multigrid Framework for Variational Approaches in Medical Image Processing and Computer Vision*. PhD thesis, Friedrich Alexander Universität Erlangen-Nürnberg, 2008.
- [9] Leander Sturm. Beigelegte CD mit Parameter Extractor inkl. Quellcode und Dokumentation. Technical report, Friedrich Alexander Universität Erlangen-Nürnberg, 2010.
- [10] Christian Ullenboom. *Java ist auch eine Insel*. Galileo Computing, 8. edition, 2009.