



UNIVERSITÄT ZU LÜBECK  
INSTITUTE OF MATHEMATICS AND  
IMAGE COMPUTING

# Semiglatte Optimierungsverfahren zweiter Ordnung in der Bildverarbeitung

*Semismooth Second Order Optimization in Image Processing*

## Masterarbeit

im Rahmen des Studiengangs  
Mathematik in Medizin und Lebenswissenschaften  
der Universität zu Lübeck

## Vorgelegt von

Caterina Rust, B. Sc.

## Ausgegeben und betreut von

Prof. Dr. Jan Lellmann  
Institute of Mathematics and Image Computing

## Mit Unterstützung von

Thomas Vogt, M. Sc.  
Institute of Mathematics and Image Computing

Lübeck, den 22.12.2017



### **Eidesstattliche Erklärung**

Ich versichere an Eides statt, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Quellen und Hilfsmittel angefertigt zu haben.

Lübeck, den 22. Dezember 2017

---

Caterina Rust



## Kurzfassung

In dieser Arbeit werden Lösungsverfahren zweiter Ordnung zur Lösung semiglatte, konvexer Optimierungsprobleme mit semiglatten Proximal-Abbildungen vorgestellt. Es werden Clarkes generalisierte Jacobi-Matrizen genutzt, um die primalen und dualen Optimalitätsbedingungen für das Problem aufzustellen. Diese werden mithilfe von Prox-Schritten umformuliert. Zur Lösung wird ein semiglattes Newton-Verfahren vorgestellt. Durch die Multiplikation mit einer Blockmatrix kann das Newton-System symmetrisiert werden. Zudem wird ein Levenberg-Marquardt-Algorithmus zur Lösung des Problems vorgestellt. Die Konvergenzanalyse dieses Algorithmus liefert eine Aussage darüber, in welchen Fällen in einer Iteration ein hinreichender Abstieg erzielt werden kann. Die Algorithmen werden auf das Rudin-Osher-Fatemi-Problem (ROF-Problem) angewendet. Dabei konvergieren alle drei Verfahren. Aufgrund der benötigten Cholesky-Zerlegungen ist die Rechenzeit des Levenberg-Marquardt-Verfahrens deutlich höher als die der beiden Newton-Algorithmen. Das semiglattes Newton-Verfahren hat für dieses Problem im Durchschnitt eine geringere Laufzeit als CVX mit dem Löser MOSEK.

## Abstract

In this work we present second order solution methods for solving semismooth, convex optimization problems with semismooth proximal operators. Clarke's generalized Jacobians are used to formulate the primal and dual optimality conditions for the problem. They are rewritten by proximal steps and solved using a semismooth Newton method. By multiplication by a block matrix the Newton system can be symmetrized. Moreover we present a Levenberg-Marquardt algorithm for solving the problem. The convergence analysis of this algorithm provides requirements for a sufficient descent in an iteration. The algorithms are applied to the Rudin-Osher-Fatemi problem (ROF problem). All methods converge. Because of the required Cholesky factorization, the computational costs of the Levenberg-Marquardt method are higher than the costs of the Newton methods. The Semismooth Newton method on average needs lower computation time than CVX with the MOSEK solver for this problem.



## Danksagung

An dieser Stelle möchte ich all jenen danken, die mich bei dieser Arbeit unterstützt haben. Ein herzlicher Dank geht an meinen Betreuer Prof. Dr. Jan Lellmann, der immer zu einem Gespräch bereit war, für die hilfreichen Denkanstöße und Anmerkungen. Ebenso möchte ich auch Thomas Vogt für seine Unterstützung und Mühe danken. Außerdem danke ich meiner Familie und meinen Freunden, dass sie immer ein offenes Ohr für mich hatten. Ein besonderes Dankeschön geht dabei an meine Mutter, die in allen Lebenslagen für mich da war. Auch an die Anderen aus dem Studenten-Pool des Instituts geht ein großes Dankeschön, für all den Spaß, den die Arbeit auch dank ihnen gemacht hat, und die Hilfe bei kleineren und größeren Problemen.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Notation und mathematische Konventionen . . . . .	5
2.2	Grundbegriffe und Aussagen . . . . .	5
<b>3</b>	<b>Problemstellung</b>	<b>13</b>
<b>4</b>	<b>Semiglattes Newton-Verfahren</b>	<b>15</b>
4.1	Newton-System . . . . .	16
4.2	Modifiziertes Newton-System . . . . .	17
4.2.1	Zusammenhang zum glattem Fall . . . . .	18
4.2.2	Symmetrie der Newton-Matrix . . . . .	20
4.2.3	Definitheit . . . . .	21
<b>5</b>	<b>Semiglatte Levenberg-Marquardt-Algorithmus</b>	<b>25</b>
5.1	Klassischer Levenberg-Marquardt-Algorithmus . . . . .	25
5.2	Modifizierter Levenberg-Marquardt-Algorithmus . . . . .	27
5.3	Analyse des modifizierten Levenberg-Marquardt-Algorithmus . . . . .	30
5.4	Ausblick für den Konvergenzbeweis . . . . .	35
<b>6</b>	<b>Ergebnisse</b>	<b>39</b>
6.1	ROF-Problem . . . . .	39
6.2	Experimente . . . . .	42
6.2.1	Semiglattes Newton-Verfahren . . . . .	43
6.2.2	Modifiziertes semiglattes Newton-Verfahren . . . . .	45
6.2.3	Modifizierter Levenberg-Marquardt-Algorithmus . . . . .	47
6.2.4	Vergleich mit weiteren Verfahren . . . . .	49
<b>7</b>	<b>Diskussion und Ausblick</b>	<b>53</b>
	<b>Literaturverzeichnis</b>	<b>55</b>



# Einleitung

Viele Aufgabenstellungen in der Bildverarbeitung, wie etwa die Rauschreduktion oder die Segmentierung, können als konvexe Minimierungsprobleme formuliert werden [6]. Allerdings sind die entsprechenden Zielfunktionen in vielen Anwendungen nicht differenzierbar. Daher können Optimierungsverfahren höherer Ordnung, welche im Allgemeinen eine schnellere Konvergenz liefern, aber die Existenz zweiter Ableitungen voraussetzen, nicht ohne weiteres auf derartige Probleme angewendet werden.

In dieser Arbeit sollen Probleme der Form

$$\min_{u \in \mathbb{R}^n} F(u) \tag{1.1}$$

für eine Funktion  $F: \mathbb{R}^n \rightarrow \mathbb{R}$  gelöst werden. Ist  $F$  zweimal stetig differenzierbar, so kann zur Lösung das Newton-Verfahren angewendet werden, welches auf der Taylorentwicklung zweiter Ordnung beruht und daher die Hesse-Matrix benötigt. Falls  $F$  nur einmal stetig differenzierbar ist, können gradientenbasierte Verfahren genutzt werden. Für nicht stetig differenzierbare Zielfunktionen  $F$ , welche konvex sind, können sogenannte prox-Methoden verwendet werden [9]. Die Frage, mit der sich diese Arbeit beschäftigt, ist, ob sich für konvexe, jedoch nicht differenzierbare Funktionen, auch Verfahren zweiter Ordnung nutzen lassen. Es werden Optimierungsalgorithmen zweiter Ordnung für eine bestimmte Klasse konvexer, sogenannter „semiglatte“ Funktionen entwickelt.

## Verwandte Arbeiten

Ein Beispiel für Optimierungsalgorithmen erster Ordnung für konvexe Probleme ist die „primal-dual hybrid gradient method“ (PDHG-Methode) [5]. Sie beruht auf einem Splitting der Zielfunktion, bei dem ein linearer Operator genutzt wird, und verwendet zum Update der primalen und dualen Variablen Prox-Schritte. Die Updates sind dabei immer sowohl von der primalen als auch von der dualen Variablen der vorherigen Iteration abhängig [5]. Die Schrittweitenwahl spielt für die Konvergenzrate dieser

## 1 Einleitung

Methode eine entscheidende Rolle [9, 13, 27]. Wird die Schrittweite ungeeignet gewählt, so kann keine Konvergenz erreicht werden; bei einer passenden, gegebenenfalls adaptiven, Wahl ist es jedoch möglich, lineare Konvergenz zu zeigen [16].

Zu den Verfahren zweiter Ordnung für konvexe Zielfunktionen gehören beispielsweise Interior-Point- und semiglatte Newton-Verfahren. Beide Verfahren nutzen Newton-Schritte und benötigen somit die zweite Ableitung der Zielfunktion beziehungsweise einer äquivalenten Formulierung.

Karmarkar stellte 1984 den ersten Interior-Point-Algorithmus für lineare Zielfunktionen vor [19]. Dieser Algorithmus konvergiert polynomial. Die Anwendung auf konvexe Funktionen erfolgt durch eine Umformulierung des Problems mithilfe von Kegeln. Diese kann allerdings sehr aufwändig sein und das Problem durch Einführung neuer Variablen und Nebenbedingungen vergrößern [14]. Bei Interior-Point-Verfahren wird auf die mit einem Strafparameter multiplizierte Zielfunktion des Optimierungsproblems eine sogenannte Barriere-Funktion addiert. Die dadurch gewonnene, noch immer konvexe, Funktion nimmt ihr Minimum im Inneren des zulässigen Bereichs an. Nun wird in jeder Iteration ein einzelner Newton-Schritt vollzogen und anschließend der Strafparameter erhöht [2]. Die Iterierten müssen bei diesem Verfahren jedoch nicht zwingend in der zulässigen Menge liegen [3].

Der Begriff Semiglattheit („semismoothness“) wurde 1977 von Mifflin eingeführt [20]. 1983 stellte Clarke ein neues Subdifferential vor [7], welches Qi und Sun 1993 nutzten, um ein semiglattes Newton-Verfahren zu entwickeln [29]. Semiglatte Newton-Verfahren konvergieren lokal mindestens  $Q$ -superlinear [17]. Hintermüller, Ito und Kunisch konnten 2002 zeigen, dass sich eine primal-duale Methode mit aktiven Mengen („primal-dual active-set“) zur Lösung von quadratischen Problemen mit linearen Nebenbedingungen auch als semiglatte Newton-Verfahren formulieren lässt [18]. Die Schrittweiten-Steuerung erfolgt bei diesen Verfahren meist mit Armijo-Schrittweiten. Es werden jedoch teilweise auch Filter-Methoden eingesetzt, um Schritte abzulehnen [21].

Für Probleme mit stetig differenzierbaren Zielfunktionen schlugen Nocedal und Yuan 1998 eine Kombination von Trust-Region-Verfahren und Liniensuche vor [26]. Yuan untersuchte 1983 die Eigenschaften von Trust-Region-Algorithmen für nicht-glatte Probleme [37]. Auch Trust-Region-Algorithmen nutzen Newton-Schritte, um das Subproblem zu lösen. Das Levenberg-Marquardt-Verfahren kombiniert das Gauß-Newton Verfahren mit dem Trust-Region-Ansatz, um kleinste Quadrate-Probleme zu lösen [25]. Michael Ulbrich nutzt in dem von ihm vorgeschlagenen Trust-Region-Verfahren für semiglatte Funktionen Projektionen im Newton-Algorithmus [33].

## Aufbau

Kapitel 2 enthält eine genaue Definition der Voraussetzungen. Darüber hinaus werden die Notation und weitere wichtige Begriffe für diese Arbeit eingeführt. Die Problemstellung wird in Kapitel 3 exakt formuliert. In Kapitel 4 wird ein semiglatte Newton Verfahren entwickelt. Der Algorithmus nutzt anstelle der Ableitungen der Zielfunktion Subdifferentialle von Prox-Schritten der gesplitteten Zielfunktion. Außer-

dem wird eine Modifikation vorgenommen, um eine mögliche Verbesserung der Eigenschaften des Systems zu untersuchen. Um eine bessere Konvergenz zu erzielen wird in Kapitel 5 eine modifizierte Form des Levenberg-Marquardt-Algorithmus entwickelt, welche ebenfalls die Subdifferentialen von Prox-Schritten nutzt. Des Weiteren wird das Konvergenzverhalten dieses Verfahrens untersucht. Die vorgestellten Algorithmen werden in Kapitel 6 auf das Rudin-Osher-Fatemi-Problem (ROF-Problem) [31], als typisches nichtglattes Bildverarbeitungsproblem, angewendet. Das Modell von Rudin, Osher und Fatemi dient der Entfernung von - üblicherweise normalverteiltem - Rauschen in Bildern unter Kantenerhalt. Die Ergebnisse der Experimente mit den neuen Algorithmen werden außerdem mit den Lösungen der PDHG-Methode und eines Interior-Point-Algorithmus verglichen. Im letzten Kapitel 7 werden die Ergebnisse dieser Arbeit diskutiert.



# Grundlagen

In diesem Kapitel werden neben der Notation die wichtigsten Begriffe und einige nützliche Aussagen zum Verständnis der Arbeit erläutert. Für Grundlagen aus der Analysis sei auf die Lehrbücher von Forster [10, 11, 12] verwiesen. Elementare Begriffe der Optimierung, wie sie beispielsweise im Anhang des Werkes von Nocedal und Wright [25] erläutert werden, werden ebenfalls als bekannt vorausgesetzt.

## 2.1 Notation und mathematische Konventionen

Die Menge der reellen Zahlen wird mit  $\mathbb{R}$  bezeichnet. Alle Vektoren  $x \in \mathbb{K}^d$  sind Spaltenvektoren der Dimension  $d$  über dem Körper  $\mathbb{K}$ . Der effektive Definitionsbereich  $\{x \in \mathbb{R}^n : f(x) < \infty\}$  einer Abbildung  $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  wird mit  $\text{dom}(f)$  bezeichnet. Mit  $\|\cdot\|$  werden Normen geschrieben. Spezielle Normen werden durch den entsprechenden Index gekennzeichnet. Wenn kein Index angegeben ist, wird für Vektoren die euklidische Norm und für Matrizen die Operatornorm verwendet. Die Einheitsmatrix wird mit  $I$  bezeichnet. Die Eigenwerte einer  $n$ -dimensionalen symmetrischen Matrix  $A$  werden mit  $\sigma_i(A)$ ,  $i = 1, \dots, n$ , bezeichnet. Dabei werden Eigenwerte mit einer algebraischen Vielfachheit über 1 auch mehrfach genannt und es gilt  $\sigma_1(A) \geq \sigma_2(A) \geq \dots \geq \sigma_n(A)$ . Mit  $f: X \rightrightarrows Y$  wird eine mengenwertige Abbildung  $f$  bezeichnet, welche Elemente aus  $X$  auf Teilmengen von  $Y$  abbildet [30].

## 2.2 Grundbegriffe und Aussagen

Ziel dieser Arbeit ist es, Optimierungsprobleme in der Bildverarbeitung zu lösen. Ein *Bild* ist dabei im Rahmen dieser Arbeit zunächst eine reelle, skalar- oder vektorwertige Funktion  $S: D \rightarrow \mathbb{R}^n$ , wobei  $D$  das Bildgebiet darstellt.

Um auch mit den erweiterten reellen Zahlen  $\mathbb{R} \cup \{-\infty, \infty\}$  arbeiten zu können, werden zuerst einige Rechenregeln eingeführt.

## 2 Grundlagen

### Definition 2.1 (Rechenregeln für erweiterte reelle Zahlen)

Es sei  $c \in \mathbb{R}^n$  beliebig. Dann gelten folgende Regeln:

1.  $\pm\infty + c = \pm\infty$ ,
2.  $\pm\infty \cdot 0 = 0$ ,
3.  $\inf \mathbb{R} = \sup \emptyset = -\infty$ ,
4.  $\inf \emptyset = \sup \mathbb{R} = \infty$  und
5.  $\pm\infty \mp \infty = \infty$ .

Da in dieser Arbeit mit nichtglatten Funktionen gearbeitet wird, werden verschiedene Arten von Subdifferentialen eingeführt. Subdifferentialen sind mengenwertige Operatoren. An Stellen, an denen eine Funktion differenzierbar ist, stimmen sie mit dem Differential überein. In den übrigen Punkten enthält ein Subdifferential mehrere Subgradienten, welche die Funktion jeweils nur in diesem einen Punkt berühren. Enthält ein Subdifferential die 0, so liegt an der Stelle entweder ein Optimum oder ein Sattelpunkt.

### Definition 2.2 (Subdifferential; [30, Def. 8.3])

Sei  $S: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  eine Funktion und  $\bar{x} \in \mathbb{R}^n$  ein Punkt, sodass  $S(x)$  endlich ist. Dann heißt die Menge

$$\hat{\partial}S(x) := \left\{ J \in \mathbb{R}^n : \liminf_{\substack{\bar{x} \rightarrow x \\ \bar{x} \neq x}} \frac{S(\bar{x}) - S(x) - \langle J, \bar{x} - x \rangle}{\|\bar{x} - x\|} \geq 0 \right\} \quad (2.1)$$

reguläres Subdifferential von  $S$  an der Stelle  $x$  und die Menge

$$\partial S(x) := \left\{ J \in \mathbb{R}^n : \exists \{x_k\} \subset \mathbb{R}^n \text{ mit } x_k \rightarrow x, S(x_k) \rightarrow S(x) \text{ und} \right. \\ \left. \exists \{J_k\} \in \hat{\partial}S(x_k) \text{ mit } J_k \rightarrow J \right\} \quad (2.2)$$

wird als *Subdifferential* von  $S$  an der Stelle  $x$  bezeichnet.

Im Vergleich zum regulären Subdifferential muss der Limes inferior im klassischen Differential mit dem Limes superior übereinstimmen und den Wert 0 annehmen.

Für eigentliche, konvexe Funktionen  $S$  stimmen das Subdifferential und das reguläre Subdifferential immer überein [30, Prop. 8.12]. Eine eigentliche Funktion nimmt niemals den Wert  $-\infty$  an und ist in mindestens einem Punkt endlich [30, S. 5].

Die weiteren Subdifferentialbegriffe setzen eine stärkere Form der Stetigkeit voraus, die Lipschitz-Stetigkeit, welche hier in Anlehnung an [25] eingeführt wird.

### Definition 2.3 (Lipschitz-Stetigkeit; [25, S. 624])

Sei  $\Omega \subseteq \mathbb{R}^n$  eine Menge und  $S: \Omega \rightarrow \mathbb{R}^m$  eine Funktion. Es bezeichnen  $\|\cdot\|_{\mathcal{V}}$  und

$\|\cdot\|_{\mathcal{W}}$  beliebige Normen auf  $\mathbb{R}^m$  bzw.  $\mathbb{R}^n$ . Falls für eine Teilmenge  $\mathcal{N} \subseteq \Omega$  eine *Lipschitz-Konstante*  $L > 0$  existiert, sodass für alle  $x, y \in \mathcal{N}$  gilt

$$\|S(x) - S(y)\|_{\mathcal{V}} \leq L\|x - y\|_{\mathcal{W}}, \quad (2.3)$$

so heißt  $S$  *Lipschitz-stetig* auf  $\mathcal{N}$ .

Für einen beliebigen Punkt  $x \in \Omega$  wird  $S$  *lokal Lipschitz-stetig* in  $x$  genannt, falls eine offene Umgebung  $\mathcal{N} \subset \Omega$  von  $x$  und eine Konstante  $L > 0$  existieren, sodass die obige Bedingung (2.3) für alle  $y \in \mathcal{N}$  erfüllt ist.

Ein Subdifferentialbegriff, welcher nur für Lipschitz-stetige Funktionen definiert ist, ist das nun folgende B-Subdifferential. Ist eine Funktion in  $x$  lokal Lipschitz-stetig, so ist sie in einer offenen Umgebung von  $x$  fast überall differenzierbar [34].

**Definition 2.4** (B-Subdifferential; [34, Def. 2.1])

Sei  $\Omega \subseteq \mathbb{R}^n$  offen und  $S: \Omega \rightarrow \mathbb{R}^m$  eine in  $x \in \Omega$  lokal Lipschitz-stetige Funktion.  $D_S$  bezeichne die Menge aller Punkte, in denen  $S$  differenzierbar ist. Die Menge

$$\partial_B S(x) := \{J \in \mathbb{R}^{n \times m} : \exists \{x_k\} \subset D_S \text{ mit } x_k \rightarrow x, \nabla S(x_k) \rightarrow J\} \quad (2.4)$$

heißt *B-Subdifferential* von  $S$  in  $x$ , benannt nach Georges Bouligand.

Zur Veranschaulichung wird die Lipschitz-stetige Abbildung  $f: \mathbb{R} \rightarrow \mathbb{R}$  mit  $f(x) = |x|$  betrachtet. Das B-Subdifferential im Punkt  $x = 0$  ist die Menge  $\{-1, 1\}$ . An der Stelle 0 nimmt  $f$  das Minimum an. Im B-Subdifferential ist die 0 jedoch nicht enthalten. Daher kann es - zumindest intuitiv - nicht genutzt werden, um mittels des Newton-Verfahrens das Minimum zu bestimmen. Denn im Newton-Verfahren wird das Minimum durch das Finden einer Nullstelle des Differentials bestimmt (vgl. Kapitel 4). Frank Clarke führte in [7] einen allgemeineren Begriff ein:

**Definition 2.5** (Clarke's generalisierte Jacobi-Matrix; [7])

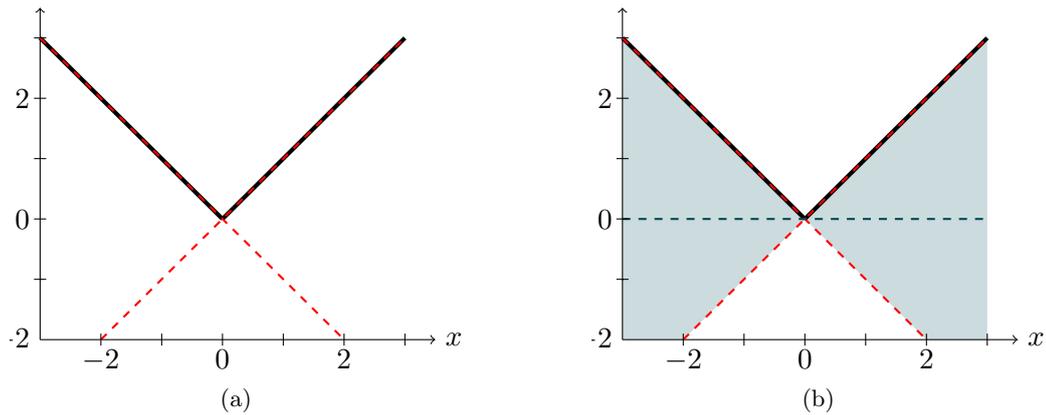
Sei  $\Omega \subseteq \mathbb{R}^n$  offen und  $S: \Omega \rightarrow \mathbb{R}^m$  eine in  $x \in \Omega$  lokal Lipschitz-stetige Funktion. Die konvexe Hülle des B-Subdifferentials  $\partial_C S(x) := \text{co}(\partial_B S(x))$  an der Stelle  $x \in \Omega$  heißt *Clarke's generalisierte Jacobi-Matrix*.

Betrachtet man nun erneut die Funktion  $f(x) = |x|$ , so erhält man  $\partial_C f(0) = [-1, 1]$ . Dieses Subdifferential enthält die 0 und ist somit dazu geeignet, die Optimalitätsbedingung  $0 \in \partial_C f(x)$  zu überprüfen. Der Vergleich der beiden Subdifferentialbegriffe ist in Abbildung 2.1 zu sehen.

Im Folgenden wird hauptsächlich mit Clarke's generalisierter Jacobi-Matrix gearbeitet.

Sie ist eine nichtleere, kompakte und konvexe Menge. Zudem ist sie lokal beschränkt und oberhalbstetig [34, Prop. 2.2]. Oberhalbstetigkeit bedeutet für eine mengenwertige Abbildung  $f: \Omega \rightrightarrows \mathbb{R}^m$ ,  $\Omega \subseteq \mathbb{R}^n$ , dass für jedes  $x \in \Omega$  und für alle  $\varepsilon > 0$  ein

## 2 Grundlagen



**Abbildung 2.1:** Vergleich (a) des B-Subdifferentials und (b) der generalisierten Jacobi-Matrix nach Clarke für die Betragsfunktion  $f$ . Die roten Geraden sind die Elemente des B-Subdifferentials. Die generalisierte Jacobi-Matrix enthält alle Geraden, die durch den hinterlegten Bereich verlaufen.

$\delta > 0$  existiert, sodass für alle  $y \in \Omega$  mit der Eigenschaft  $\|x - y\| < \delta$

$$f(y) \subseteq \{z + h : z \in f(x), \|h\| < \varepsilon\} \quad (2.5)$$

erfüllt ist [34, Def. A.7].

Um das Wachstum von Funktionen zu vergleichen, wird die O-Notation genutzt, welche von Landau eingeführt worden ist.

**Definition 2.6** (Landau-Symbole; [11, S. 133ff.])

Seien  $f, g: D \rightarrow \mathbb{R}$  zwei Funktionen und  $x_0$  ein Berührungspunkt von  $D$ . Dann wird

$$f(x) = o(g(x)), \text{ für } x \rightarrow x_0, x \in D \quad (2.6)$$

geschrieben, wenn für alle  $\varepsilon > 0$  ein  $\delta > 0$  existiert, sodass für jedes  $x \in D$ , für das  $|x - x_0| < \delta$  gilt,

$$|f(x)| \leq \varepsilon |g(x)| \quad (2.7)$$

erfüllt ist. Man schreibt

$$f(x) = \mathcal{O}(g(x)), \text{ für } x \rightarrow x_0, x \in D, \quad (2.8)$$

falls eine Konstante  $c > 0$  und ein  $a \in D$  existieren, sodass für alle  $x > a$  gilt, dass

$$|f(x)| \leq c |g(x)| \quad (2.9)$$

erfüllt ist.

In Anlehnung an [17] wird nun der Begriff Semiglattheit eingeführt:

**Definition 2.7** (Semiglattheit; [17, Theorem: 2.9])

Sei  $\Omega \subseteq \mathbb{R}^n$  offen und nichtleer. Eine Funktion  $S: \Omega \rightarrow \mathbb{R}^m$  heißt *semiglat*, wenn für alle  $\omega \in \Omega$  die folgenden drei Bedingungen erfüllt sind:

1.  $S$  ist lokal Lipschitz-stetig in  $\omega$ ,
2. die Richtungsableitungen

$$S'(\omega, d) = \lim_{t \downarrow 0} \frac{S(\omega + td) - S(\omega)}{t} \quad (2.10)$$

existieren für alle Richtungen  $d \in \Omega$  und

3. für jedes  $J \in \partial_C S(\omega + d)$  mit  $d \rightarrow 0$  gilt

$$\|S(\omega + d) - S(\omega) - Jd\| = o(\|d\|). \quad (2.11)$$

Der Begriff wurde von Mifflin [20] für Funktionale eingeführt und später von Qi und Sun auf vektorwertige Funktionen [29] erweitert.

Ein Beispiel für eine nicht semiglatte Funktion ist  $f: \mathbb{R} \rightarrow \mathbb{R}$  mit

$$f(x) = \begin{cases} x^2 \sin\left(\frac{1}{x}\right), & x \neq 0, \\ 0, & x = 0. \end{cases} \quad (2.12)$$

Diese Funktion  $f$  ist lokal Lipschitz-stetig und auch die Richtungsableitungen existieren, sodass Clarkes generalisierte Jacobi-Matrix

$$\partial_C f(x) = \begin{cases} 2x \sin\left(\frac{1}{x}\right) - \cos\left(\frac{1}{x}\right), & x \neq 0, \\ [-1, 1], & x = 0 \end{cases} \quad (2.13)$$

bestimmt werden kann. Allerdings ist die dritte Bedingung für Semiglattheit nicht erfüllt. Betrachtet man die Stelle  $x = 0$ , so erhält man für  $d \rightarrow 0$  nur

$$\left\| d^2 \sin\left(\frac{1}{d}\right) - 0 - 2d \sin\left(\frac{1}{d}\right) + \cos\left(\frac{1}{d}\right) \right\| = \mathcal{O}(\|d\|). \quad (2.14)$$

Diese Funktion ist also nicht semiglat. Dieses Beispiel veranschaulicht die Bedeutung der dritten Bedingung. Diese ist mit der totalen Differenzierbarkeit vergleichbar. Bei zu starken Änderungen der Richtungsableitungen in einem Punkt ist sie nicht mehr erfüllt.

Es werden zwei weitere Funktionen  $g, h: \mathbb{R} \rightarrow \mathbb{R}$  betrachtet. Neben der konvexen Betragsfunktion ist auch  $g(x) = \log(1 + |x|)$  semiglat, aber nicht konvex. Dagegen ist  $h(x) = \sqrt{|x|}$  nicht semiglat, da  $h$  nicht lokal Lipschitz-stetig ist.

Semiglattheit bleibt bei Addition, skalarer Multiplikation und Verkettung erhalten [20]. Stetige Differenzierbarkeit ist eine stärkere Eigenschaft als Semiglattheit.

## 2 Grundlagen

Konvexität sowie stückweise Differenzierbarkeit implizieren Semiglattheit [32, 36]. Normen sind ebenso wie stückweise affine Funktionen semiglatt [36].

Zur Beschreibung des Problems werden im Verlauf dieser Arbeit die Konjugierte einer Abbildung und Proximal-Abbildungen („proximal mappings“) benötigt. Die beiden Begriffe werden nun analog zu [30] eingeführt.

**Definition 2.8** (Konjugierte; [30, S. 473])

Für eine Funktion  $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  heißt die Funktion  $f^*: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  mit

$$f^*(v) := \sup_{u \in \mathbb{R}^n} \{\langle v, u \rangle - f(u)\} \quad (2.15)$$

die (*Legendre-Fenchel*-)Konjugierte von  $f$ .

Für eine konvexe und unterhalbstetige Funktion  $f$  gilt, dass  $f^{**} = f$  ist [3, S. 94]. Eine Funktion  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  heißt unterhalbstetig, wenn für alle  $\alpha \in \mathbb{R}$  gilt, dass die Menge

$$\{x \in \text{dom}(f) : f(x) \leq \alpha\} \quad (2.16)$$

abgeschlossen ist [30, Theorem 1.6].

**Definition 2.9** (Moreau-Hülle und Proximal-Abbildung; [30, Def. 1.22])

Sei  $S: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  eine eigentliche, unterhalbstetige Funktion und sei  $\lambda > 0$ . Die Funktion

$$e_{\lambda f}(z) = \min_{u \in \mathbb{R}^n} \left\{ f(u) + \frac{1}{2\lambda} \|u - z\|^2 \right\} \quad (2.17)$$

heißt *Moreau-Hülle*. Der zugehörige Minimierer

$$P_{\lambda f}(z) = \arg \min_{u \in \mathbb{R}^n} \left\{ f(u) + \frac{1}{2\lambda} \|u - z\|^2 \right\} \quad (2.18)$$

wird als *Proximal-Abbildung* oder *Prox-Schritt* bezeichnet.

In [22] werden zwei nützliche Eigenschaften für Proximal-Abbildungen gezeigt.

**Satz 2.10** ([22, Prop. 4a])

Sei  $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  eine konvexe und eigentliche Funktion und seien  $x, y, z \in \mathbb{R}^n$ . Dann gilt:

$$z = x + y \wedge f(x) + f^*(y) = \langle x, y \rangle \Leftrightarrow x = P_f(z) \wedge y = P_{f^*}(z). \quad (2.19)$$

Diese Aussage erleichtert die Berechnung des Prox-Schrittes, wenn die Proximal-Abbildung der Konjugierten bekannt ist, denn sie impliziert

$$P_{f^*}(z) = y = z - x = z - P_f(z). \quad (2.20)$$

Des Weiteren gilt:

**Satz 2.11** ([22, Prop. 6a])

Sei  $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  eine konvexe und eigentliche Funktion und sei  $z \in \mathbb{R}^n$ . Dann gilt:

$$x = P_{\lambda f}(z) \Leftrightarrow z \in x + \lambda \partial f(x). \quad (2.21)$$

Diese Aussage folgt direkt aus der Optimalitätsbedingung

$$0 \in \partial f(x) + \frac{1}{\lambda}(x - z) \quad (2.22)$$

für die Proximal-Abbildung.

Zuletzt wird noch die Monotonie einer Abbildung eingeführt.

**Definition 2.12** (Monotonie; [30, Def. 12.1])

Eine mengenwertige Abbildung  $T: \mathbb{R}^n \rightrightarrows \mathbb{R}^n$  heißt *monoton*, falls

$$\langle y_1 - y_0, x_1 - x_0 \rangle \geq 0 \quad (2.23)$$

für alle  $x_0, x_1 \in \mathbb{R}^n, y_0 \in T(x_0), y_1 \in T(x_1)$  gilt.

Wenn zusätzlich die Gleichheit nur im Fall  $x_0 = x_1$  gilt, so heißt  $T$  *strikt monoton*.

Monotonie spielt in der konvexen Analysis eine große Rolle. Wenn eine eigentliche Abbildung  $T: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  konvex ist, so sind die Elemente ihres Subdifferentials monoton. Für eigentliche, unterhalbstetige Abbildungen gilt sogar die Äquivalenz [30, Theorem 12.17].

Auf Grundlage der in diesem Kapitel eingeführten Begriffe und Aussagen werden im weiteren Verlauf der Arbeit Lösungsverfahren für die Problemstellung entwickelt und analysiert, welche im folgenden Kapitel vorgestellt wird.



## Problemstellung

In der Bildverarbeitung stößt man häufig auf konvexe Minimierungsprobleme. Sie treten beispielsweise bei der Rauschreduktion von Bildern oder auch bei Segmentierungsproblemen auf. Diese Probleme sind jedoch selten stetig differenzierbar.

In dieser Arbeit sollen konvexe, eigentliche, nicht differenzierbare Probleme der Form

$$\inf_{u \in \mathbb{R}^n} \langle c, u \rangle + g(u) + f(Au - b) \quad (3.1)$$

betrachtet werden. Dabei seien  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ . Die Funktion  $g: \mathbb{R}^n \rightarrow \mathbb{R}$  sei zweimal stetig differenzierbar und konvex,  $f: \mathbb{R}^m \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  sei unterhalbstetig, eigentlich und ebenfalls konvex.

Da die Zielfunktion in (3.1) konvex und eigentlich ist, existiert mindestens ein Minimierer. Aufgrund der Konvexität sind alle lokalen Minimierer sogar globale Minimierer und die Menge der Minimierer ist ebenfalls konvex. Bei strikter Konvexität ist der Minimierer sogar eindeutig [30, Theorem 2.6].

Mittels der Konjugierten lässt sich das duale Problem herleiten:

$$\inf_u \sup_v \langle c, u \rangle + g(u) + \langle v, Au - b \rangle - f^*(v) \quad (3.2)$$

$$\geq \sup_v \inf_u \langle c, u \rangle + g(u) + \langle A^\top v, u \rangle - \langle v, b \rangle - f^*(v) \quad (3.3)$$

$$= \sup_v -\langle v, b \rangle - f^*(v) - \sup_u \left\{ \langle -A^\top v - c, u \rangle - g(u) \right\} \quad (3.4)$$

$$= \sup_v -\langle v, b \rangle - g^*(-A^\top v - c) - f^*(v). \quad (3.5)$$

Aus den Sattelpunktproblemen (3.2) und (3.3) lassen sich nun die notwendigen primal-dualen Optimalitätsbedingungen

$$\begin{cases} 0 \in \partial g(u) - (-A^\top v - c), \\ 0 \in \partial f^*(v) - (Au - b) \end{cases} \quad (3.6)$$

### 3 Problemstellung

ableiten. Falls die Optimalitätsbedingungen für  $(u, v) = (u^*, v^*)$  erfüllt sind und der primale und der duale Zielfunktionswert gleich sind, also

$$\langle c, u^* \rangle + g(u^*) + f(Au^* - b) = -\langle v^*, b \rangle - g^*(-A^\top v^* - c) - f^*(v^*) \quad (3.7)$$

gilt, sind die Optimalitätsbedingungen auch hinreichend [25].

Mittels der Äquivalenz aus Satz 2.11 lassen sich die Optimalitätsbedingungen für ein beliebiges festes  $\tau > 0$  umschreiben zu:

$$0 \in \partial g(u) - (-A^\top v - c) \Leftrightarrow 0 \in \tau \partial g(u) - \tau(-A^\top v - c) \quad (3.8)$$

$$\Leftrightarrow 0 \in u - (u + \tau(-A^\top v - c)) + \tau \partial g(u) \quad (3.9)$$

$$\Leftrightarrow u = P_{\tau g}(u + \tau(-A^\top v - c)) \quad (3.10)$$

$$\Leftrightarrow 0 = u - P_{\tau g}(u + \tau(-A^\top v - c)), \quad (3.11)$$

analog gilt für ein beliebiges festes  $\sigma > 0$ :

$$0 \in \partial f^*(v) - (Au - b) \Leftrightarrow 0 = v - P_{\sigma f^*}(v + \sigma(Au - b)). \quad (3.12)$$

Diese Formulierung erlaubt es nun, das ursprüngliche Minimierungsproblem 3.1 durch Lösen des nichtlinearen Gleichungssystems

$$\begin{pmatrix} u - P_{\tau g}(u + \tau(-A^\top v - c)) \\ v - P_{\sigma f^*}(v + \sigma(Au - b)) \end{pmatrix} = 0 \quad (3.13)$$

zu lösen. Die Lösungsmenge dieses Gleichungssystems ist nichtleer, da das Minimierungsproblem 3.1 mindestens eine Lösung besitzt. Die Wahl der Parameter  $\sigma$  und  $\tau$  spielt für die Lösung des Gleichungssystems keine Rolle. Allerdings beeinflusst sie die Konvergenz iterativer Verfahren zur Lösung des Minimierungsproblems (3.1). Denn die Äquivalenzen in (3.8) - (3.12) beruhen auf der Annahme, dass  $(u, v)$  ein kritischer Punkt der Sattelpunktprobleme (3.2) und (3.3) ist. Diese ist jedoch für die Iterierten nicht erfüllt.

Zur Lösung von (3.13) gibt es verschiedene Möglichkeiten. Hier werden zwei Lösungsansätze vorgestellt. Als erstes wird in Kapitel 4 das semiglatte Newton-Verfahren eingeführt, welches auf der Idee des klassischen Newton-Verfahrens beruht. Anstelle von Ableitungen werden jedoch Clarkes generalisierte Jacobi-Matrizen genutzt. Anschließend wird in Kapitel 5 eine Adaption des Levenberg-Marquardt-Algorithmus eingeführt. Dazu wird die Nullstellensuche erneut als Minimierungsproblem formuliert, um dann wie im vorherigen Ansatz Clarkes generalisierte Jacobi-Matrizen verwenden zu können.

## Semiglattes Newton-Verfahren

In diesem Kapitel wird ein semiglattes Newton-Verfahren vorgestellt, um das Problem 3.13 aus Kapitel 3 zu lösen. Das Newton-Verfahren wird genutzt, um Nullstellen einer Funktion zu finden [25]. Es ist ein iteratives Verfahren, das die Taylor-Approximation erster Ordnung nutzt. Für eine differenzierbare Funktion  $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$  ergibt sich aus dieser in der  $k$ -ten Iteration die sogenannte *Newton-Richtung*

$$d_k = -(\nabla f(x_k))^{-1} f(x_k). \quad (4.1)$$

Dazu muss die Jacobi-Matrix invertierbar sein. Damit  $d_k$  eine Abstiegsrichtung ist, muss  $d_k^\top f(x_k) < 0$  sein. Mit (4.1) folgt

$$\nabla f(x_k) d_k = -f(x_k) \quad (4.2)$$

$$\Leftrightarrow -d_k^\top \nabla f(x_k) d_k = d_k^\top f(x_k). \quad (4.3)$$

Aufgrund der Negativität der rechten Seite muss die Jacobi-Matrix  $\nabla f(x_k)$  positiv definit sein [25]. Die grundlegende Iteration hat dann die Form

$$x_{k+1} = x_k + \nu_k d_k. \quad (4.4)$$

Dabei ist  $\nu_k > 0$  die Schrittweite. Zur Schrittweiten-Steuerung werden unterschiedliche Methoden angewendet. Eine Möglichkeit, die auch in dieser Arbeit genutzt wird, sind Armijo-Schrittweiten. Die Armijo-Bedingung für ein  $c \in (0,1)$

$$f(x_k + \nu_k d_k) \leq f(x_k) + c \nu_k \nabla f_k^\top d_k \quad (4.5)$$

sorgt dafür, dass die Verringerung des Funktionswertes proportional zu dem Gradienten und der Schrittweite  $\nu_k$  der aktuellen Iteration ist [25].

Im Unterschied zum Newton-Verfahren für differenzierbare Funktionen wird im semiglatten Newton-Algorithmus ein beliebiges Element aus Clarkes generalisierter Jacobi-Matrix (siehe Definition 2.5) anstelle des Gradienten zur Bestimmung der

## 4 Semiglattes Newton-Verfahren

Suchrichtung genutzt. Trotz dieser Beliebigkeit, kann man zeigen, dass das Verfahren lokal mindestens Q-superlinear konvergiert, wenn alle Elemente aus Clarkes generalisierter Jacobi-Matrix an der Nullstelle  $x^*$  von  $f$  invertierbar sind. Das bedeutet, dass

$$\|x_{k+1} - x^*\| = o(\|x_k - x^*\|) \quad (4.6)$$

für  $k \rightarrow \infty$  gilt [17].

**Definition 4.1** (Semiglattes Newton-Verfahren [17, Algorithmus 2])

Für eine semiglatte Funktion  $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ , ist das *semiglatte Newton-Verfahren* wie folgt definiert:

---

**Algorithmus 1** Semiglattes Newton-Verfahren

---

**Eingabe:**  $x_0 \in \mathbb{R}^n$

- 1:  $k := 0$
  - 2: **while** Abbruchkriterium nicht erfüllt **do**
  - 3:   Wähle  $G(x_k) \in \partial_C F(x_k)$  beliebig
  - 4:   Löse  $G(x_k)d_k = -F(x_k)$  nach  $d_k$
  - 5:    $x_{k+1} = x_k + d_k$
  - 6:    $k = k + 1$
  - 7: **end while**
- 

Der Schritt in Zeile 4 ist ohne weitere Voraussetzungen nicht immer möglich. Daher wird hier in der Implementierung ein Vielfaches der Einheitsmatrix auf die Newton-Matrix  $G(x_k)$  addiert (vgl. Kapitel 6).

### 4.1 Newton-System

Die Optimalitätsbedingungen (3.6) für das betrachtete Minimierungsproblem (3.1) sind mengenwertig. Daher lässt sich das semiglatte Newton-Verfahren nicht direkt auf diese anwenden. Für semiglatte Prox-Schritte, lässt sich das Problem (3.1) jedoch über das Gleichungssystem (3.13) lösen. Dieses bietet den Vorteil, dass es keine Inklusionen mehr enthält, sondern nur noch Gleichungen. Man nutzt also die Funktion

$$F(u, v) = \begin{pmatrix} u - P_{\tau g}(u + \tau(-A^\top v - c)) \\ v - P_{\sigma f^*}(v + \sigma(Au - b)) \end{pmatrix} \quad (4.7)$$

für das semiglatte Newton-Verfahren. Unter der Voraussetzung, dass die Prox-Schritte semiglatt sind, existieren auch Clarkes generalisierte Jacobi-Matrizen, denn Semiglattheit setzt die Existenz aller Richtungsableitungen (vgl. Definition 2.7 Bedingung 2) voraus. Dies sichert die Existenz des B-Subdifferentials und damit auch der generalisierten Jacobi-Matrix nach Clarke.

Das klassische Newton-Verfahren kann nicht zur Lösung des Gleichungssystems (3.13) genutzt werden, da die Proximal-Abbildungen nicht unbedingt stetig differenzierbar sind. Ein Beispiel dafür ist die konvexe, semiglatte Betragsfunktion  $h: \mathbb{R} \rightarrow \mathbb{R}$  mit  $h(x) = |x|$ . Die zugehörige Proximal-Abbildung

$$P_h(x) = \begin{cases} x + 1, & x < -1, \\ 0, & |x| \leq 1, \\ x - 1, & x > 1 \end{cases} \quad (4.8)$$

ist nicht differenzierbar [8].

Aus Algorithmus 1, Zeile 4, erhält man somit das Newton-System

$$\begin{pmatrix} I - K & \tau K A^\top \\ -\sigma H A & I - H \end{pmatrix} \begin{pmatrix} u^k \\ v^k \end{pmatrix} = \begin{pmatrix} u - P_{\tau g}(u + \tau(-A^\top v - c)) \\ v - P_{\sigma f^*}(v + \sigma(Au - b)) \end{pmatrix}. \quad (4.9)$$

Dabei sind  $K$  und  $H$  so gewählt, dass

$$K \in \partial_C P_{\tau g}(u + \tau(-A^\top v - c)) \quad \text{und} \quad (4.10)$$

$$H \in \partial_C P_{\sigma f^*}(v + \sigma(Au - b)). \quad (4.11)$$

Dieses System ist jedoch nicht in jedem Fall eindeutig lösbar. Ergibt sich beispielsweise  $K = 1/2I$  und  $H = I$ , so erhält man auf der linken Seite des Newton-Systems (4.9) die Newton-Matrix

$$\begin{pmatrix} 1/2I & \tau/2A^\top \\ -\sigma A & 0 \end{pmatrix}. \quad (4.12)$$

Im Kern dieser Matrix ist für  $v \in \ker(A^\top)$  auch der Vektor  $(0, v)^\top$ . In dem Fall, dass  $A$  aus der Diskretisierung eines Differentialoperators stammt, liegen alle Konstanten  $v$  in  $\ker(A^\top)$ . Dieser Fall kann beispielsweise bei entsprechender Wahl von  $\tau$  und  $\sigma$  bei der Lösung des Rudin-Osher-Fatemi-Problems auftreten (vgl. Kapitel 6.1); dort entspricht  $A^\top$  dem Divergenzoperator.

## 4.2 Modifiziertes Newton-System

Es wird nun versucht, das System (4.9) durch eine Multiplikation mit einer Blockmatrix einfacher lösbar und effizienter implementierbar zu machen. Das neue System soll symmetrisch sein, sodass der Speicherbedarf reduziert werden kann. Wenn die neue Newton-Matrix zudem auch positiv definit ist, besitzt das System eine eindeutige Lösung. Außerdem kann das Newton-System dann über effizientere Verfahren für symmetrisch positiv definite Systeme gelöst werden. Zur Modifikation wird die Blockmatrix

$$B = \begin{pmatrix} (\tau K)^{-1} & 0 \\ 0 & (-\sigma H)^{-1} \end{pmatrix} \quad (4.13)$$

#### 4 Semiglattes Newton-Verfahren

genutzt. Dabei wird angenommen, dass die Inversen  $K^{-1}$  und  $H^{-1}$  zu den Matrizen  $K$  und  $H$  aus (4.10) und (4.11) existieren. Aus der Multiplikation von links mit der Matrix  $B$  aus (4.13) ergibt sich das modifizierte System

$$\begin{aligned} \begin{pmatrix} \frac{1}{\tau}(K^{-1} - I) & A^\top \\ A & -\frac{1}{\sigma}(H^{-1} - I) \end{pmatrix} \begin{pmatrix} u^k \\ v^k \end{pmatrix} &= B \begin{pmatrix} u - P_{\tau g}(u + \tau(-A^\top v - c)) \\ v - P_{\sigma f^*}(v + \sigma(Au - b)) \end{pmatrix} \\ &= \begin{pmatrix} (\tau K)^{-1}(u - P_{\tau g}(u + \tau(-A^\top v - c))) \\ (-\sigma H)^{-1}(v - P_{\sigma f^*}(v + \sigma(Au - b))) \end{pmatrix}. \end{aligned} \quad (4.14)$$

Das Problem dieses Ansatzes ist, dass zur Aufstellung des Systems die Elemente aus Clarkes generalisierter Jacobi-Matrix zu den Prox-Schritten bestimmt werden müssen.

Das modifizierte System besitzt, wie im Folgenden gezeigt wird, einige interessante Eigenschaften. Zuerst wird der Zusammenhang zum Newton-System im glatten Fall gezeigt. Anschließend wird die Newton-Matrix auf Symmetrie und Definitheit untersucht.

##### 4.2.1 Zusammenhang zum glatten Fall

Ziel dieses Abschnittes ist es, zu zeigen, dass die modifizierte Newton-Matrix (4.14) im Fall stetiger Differenzierbarkeit der klassischen Newton-Matrix entspricht.

Dazu wird das restringierte Optimierungsproblem

$$\inf_{z \in \mathbb{R}^n} g(z), \quad \text{so dass } Az - b = 0 \quad (4.15)$$

$$= \inf_{z \in \mathbb{R}^n} g(z) + f(Az - b), \quad f(x) = \begin{cases} 0, & x = 0, \\ \infty, & \text{sonst} \end{cases} =: \delta_{\{0\}}(x) \quad (4.16)$$

$$= \inf_{z \in \mathbb{R}^n} \sup_{v \in \mathbb{R}^m} \left\{ g(z) + v^\top (Az - b) \right\} \quad (4.17)$$

betrachtet. Dabei sei  $g: \mathbb{R}^n \rightarrow \mathbb{R}$  eine quadratische Funktion mit  $z \mapsto \frac{1}{2}z^\top Mz + c^\top z$ . Die Matrix  $M \in \mathbb{R}^{n \times n}$  sei positiv definit und  $c \in \mathbb{R}^n$ . Um das Newton-System aufzustellen, werden die Prox-Schritte  $P_{\tau g}(u)$  über die Optimalitätsbedingung für den Prox-Schritt (2.22)

$$0 = z - u + \tau \nabla g(z) \quad (4.18)$$

$$\Leftrightarrow 0 = z - u + \tau(Mz + c) \quad (4.19)$$

$$\Leftrightarrow 0 = (I + \tau M)z + \tau c - u \quad (4.20)$$

$$\Leftrightarrow z = (I + \tau M)^{-1}(u - \tau c), \quad (4.21)$$

das heißt  $P_{\tau g}(u) = (I + \tau M)^{-1}(u - \tau c)$ , und für  $f^*(u) = 0$

$$P_{\sigma f^*}(u) = \arg \min_z \frac{1}{2} \|z - u\|^2 + \sigma \cdot 0 = u \quad (4.22)$$

bestimmt. Daraus ergeben sich

$$K = \partial_C P_{\tau g}(u) = (I + \tau M)^{-1} \quad (4.23)$$

$$\Rightarrow M = \frac{1}{\tau}(K^{-1} - I) \quad (4.24)$$

und

$$H = \partial_C P_{\sigma f^*}(u) = I. \quad (4.25)$$

Denn für stetig differenzierbare Funktionen stimmen Clarkes generalisierte Jacobi-Matrix und die Jacobi-Matrix überein. Der Grund dafür ist, dass in jedem Punkt einer solchen Funktion alle Richtungsableitungen gleich sind. Daher enthält das B-Subdifferential in jedem Punkt nur ein Element, nämlich die Jacobi-Matrix. Da Clarkes generalisierte Jacobi-Matrix der konvexen Hülle des B-Subdifferentials entspricht, enthält auch diese jeweils nur ein Element. Man erhält somit im modifizierten System (4.14) auf der linken Seite die Newton-Matrix

$$\begin{pmatrix} M & A^\top \\ A & 0 \end{pmatrix} = \begin{pmatrix} \nabla^2 g & A^\top \\ A & 0 \end{pmatrix}. \quad (4.26)$$

Dies entspricht der Matrix im klassischen Newton-Verfahren für gleichungsbeschränkte Probleme [25]. Es lässt sich auch zeigen, dass die rechten Seiten der Systeme übereinstimmen. Im klassischen Verfahren würde man die Optimalitätsbedingungen für die Lagrange-Funktion

$$g(z) + \lambda^\top (Az - b) = \frac{1}{2} z^\top M z + c^\top z + \lambda^\top (Az - b) \quad (4.27)$$

mit  $\lambda_i > 0$  für alle  $i = 1, \dots, m$  betrachten und erhalte somit die rechte Seite

$$\begin{pmatrix} Mz + c + A^\top \lambda \\ Az - b \end{pmatrix}. \quad (4.28)$$

Setzt man im modifizierten System (4.14) die berechneten Proximal-Abbildungen und ihre Differentiale ein, so erhält man

$$\begin{aligned} & \begin{pmatrix} \frac{1}{\tau}(I + \tau M) \left( z - \left( (I + \tau M)^{-1} (z + \tau(-A^\top \lambda - c)) - \tau c \right) \right) \\ -\frac{1}{\sigma}(\lambda - \lambda - \sigma(Az - b)) \end{pmatrix} \\ &= \begin{pmatrix} Mz + c + A^\top \lambda \\ Az - b \end{pmatrix} \end{aligned} \quad (4.29)$$

Die Systeme sind also für quadratische, gleichungsbeschränkte, differenzierbare Probleme gleich.

### 4.2.2 Symmetrie der Newton-Matrix

Zur Untersuchung der Symmetrie der Newton-Matrix im modifizierten semiglatten Newton-Verfahren, welches auf dem System (4.14) beruht, werden einige Begrifflichkeiten benötigt, die hier in Anlehnung an [30] definiert werden. Zunächst wird der Begriff der Prox-Begrenztheit eingeführt.

**Definition 4.2** (Prox-Begrenztheit; [30, Def. 1.23])

Eine Funktion  $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  heißt *prox-begrenzt*, falls es ein  $\lambda > 0$  gibt, sodass für beliebige  $x \in \mathbb{R}^n$  gilt, dass die Moreau-Hülle  $e_\lambda f(x) > -\infty$  ist.

Besitzt eine Funktion also eine eigentliche Moreau-Hülle, so ist sie prox-begrenzt. Des Weiteren wird die Prox-Regularität vorgestellt.

**Definition 4.3** (Prox-Regularität; [30, Def. 13.27])

Eine Funktion  $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  heißt *prox-regulär* in  $\tilde{x}$  für  $\tilde{v} \in \partial f(\tilde{x})$ , wenn  $f$  endlich und lokal unterhalbstetig in  $\tilde{x}$  ist und ein  $\epsilon > 0$  sowie ein  $\rho > 0$  existieren, sodass

$$f(x') \geq f(x) + \langle v, x' - x \rangle - \frac{\rho}{2} |x' - x|^2 \text{ für alle } x' \in \mathcal{B}(\tilde{x}, \epsilon), \quad (4.30)$$

$$\text{wenn } v \in \partial f(x), |v - \tilde{v}| < \epsilon, |x - \tilde{x}| < \epsilon \text{ und } f(x) < f(\tilde{x}) + \epsilon. \quad (4.31)$$

Dabei bezeichne  $\mathcal{B}(\tilde{x}, \epsilon)$  die abgeschlossene Kugel um  $\tilde{x}$  mit Radius  $\epsilon$ .

Nach Rockafellar und Wets [30] gilt:

**Lemma 4.4** ([30, Bsp. 13.30])

Eine eigentliche, unterhalbstetige, konvexe Funktion  $S$  ist in jedem Punkt  $\tilde{x} \in \text{dom}(S)$  prox-regulär für alle  $\tilde{v} \in \partial S(\tilde{x})$ .

Die Symmetrie im Minimierer folgt mit:

**Satz 4.5** ([30, Korollar 13.53])

Sei  $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$  eine prox-begrenzte Funktion und  $\tilde{x}$  ein Punkt mit der Eigenschaft  $0 \in \partial f(\tilde{x})$ . Sei  $f$  außerdem prox-regulär für  $\tilde{v} = 0$  in  $\tilde{x}$ . Dann existiert eine offene Menge  $O$ , die  $\tilde{x}$  enthält, sodass alle Elemente der nicht-leeren und kompakten Menge  $\partial_B(P_\lambda f)(\bar{x})$  für  $\bar{x} \in O$  symmetrisch sind.

Um Satz 4.5 auf das modifizierte System (4.14) anwenden zu können, wird die Abbildung  $\tilde{g}(u) = g(u) - \langle -A^\top v, u \rangle + \langle c, u \rangle$  betrachtet, da sich aus der ersten Optimalitätsbedingung in (3.6)

$$0 \in \partial g(u) - (A^\top v - c) \quad (4.32)$$

$$\Leftrightarrow 0 \in \partial \tilde{g}(u). \quad (4.33)$$

ergibt. Betrachtet man nun den Prox-Schritt auf  $\tilde{g}$ , erhält man für ein primal-dual optimales Paar  $(u^*, v^*)$  des Problems 3.1:

$$u = P_{\tilde{g}}(u^*) \quad (4.34)$$

$$\Leftrightarrow 0 \in \partial g(u) - (-A^\top v^* - c) + u - u^* \quad (4.35)$$

$$\Leftrightarrow 0 \in \partial g(u) + u - (u^* - A^\top v^* - c) \quad (4.36)$$

$$\Leftrightarrow u = P_g(u^* - A^\top v^* - c). \quad (4.37)$$

Die letzte Zeile entspricht der ersten genutzten Optimalitätsbedingung in (3.6). Analog gilt dies für  $f^*$ . Dazu wird  $\tilde{f}^*(v) = f^*(v) - \langle Au, v \rangle + \langle b, v \rangle$  untersucht:

$$v = P_{\tilde{f}^*}(v^*) \quad (4.38)$$

$$\Leftrightarrow 0 \in \partial f^*(v) - (Au^* - b) + v - v^* \quad (4.39)$$

$$\Leftrightarrow 0 \in \partial f^*(v) + v - (v^* + Au^* - b) \quad (4.40)$$

$$\Leftrightarrow u = P_{f^*}(v^* + Au^* - b). \quad (4.41)$$

Es folgt also mit Satz 4.5 die Symmetrie der Elemente aus dem B-Subdifferential im Minimum des ursprünglichen Problems (3.1). Daraus folgt direkt die Symmetrie der Elemente aus Clarkes generalisierter Jacobi-Matrix im Optimum. Somit sind auch deren Inverse symmetrisch. Da die Subtraktion der Identität die Symmetrie nicht beeinflusst, sind die beiden Blöcke  $1/\tau(K^{-1} - I)$  und  $-1/\sigma(H^{-1} - I)$  ebenfalls symmetrisch. Die beiden anderen Blöcke der modifizierten Newton-Matrix sind jeweils die Transponierten voneinander, sodass die Matrix insgesamt symmetrisch ist.

### 4.2.3 Definitheit

Für symmetrisch positiv definite Matrizen  $A$  können effizientere Lösungsverfahren, wie das Cholesky-Verfahren, für Gleichungssysteme der Form  $Ax = y$  angewendet werden. Zudem würde die positive Definitheit die eindeutige Lösbarkeit garantieren. Daher soll nun die Definitheit der modifizierten Newton-Matrix untersucht werden. Dazu wird  $T: \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $u \mapsto u - P_{\tilde{g}}(u)$  betrachtet. Es gilt

$$\langle T(u) - T(u'), u - u' \rangle \quad (4.42)$$

$$= \langle u - P_{\tilde{g}}(u) - u' + P_{\tilde{g}}(u'), u - u' \rangle \quad (4.43)$$

$$= \langle u - u', u - u' \rangle - \langle P_{\tilde{g}}(u) - P_{\tilde{g}}(u'), u - u' \rangle \quad (4.44)$$

$$= \|u - u'\|^2 - \langle P_{\tilde{g}}(u) - P_{\tilde{g}}(u'), u - u' \rangle. \quad (4.45)$$

Da der Prox-Operator kontrahiert [22, Prop. 5b], kann man den hinteren Term mithilfe der Cauchy-Schwarz'schen-Ungleichung durch

$$\langle P_{\tilde{g}}(u) - P_{\tilde{g}}(u'), u - u' \rangle \leq \|P_{\tilde{g}}(u) - P_{\tilde{g}}(u')\| \|u - u'\| \quad (4.46)$$

$$\leq \|u - u'\| \|u - u'\| = \|u - u'\|^2 \quad (4.47)$$

#### 4 Semiglattes Newton-Verfahren

abschätzen. Somit ergibt sich insgesamt die Ungleichung

$$\langle T(u) - T(u'), u - u' \rangle \quad (4.48)$$

$$\geq \|u - u'\|^2 - \|u - u'\|^2 \quad (4.49)$$

$$= 0. \quad (4.50)$$

Es folgt also die Monotonie von  $T$ .

In Anlehnung an [36] kann die positive Semidefinitheit aller Elemente aus dem B-Subdifferential von  $T$  gezeigt werden, falls der Prox-Operator  $P_{\tilde{g}}$  semiglatt und damit Lipschitz-stetig ist.

**Satz 4.6** ([36, Lemma 3.5])

Sei  $S: \mathbb{R}^n \rightarrow \mathbb{R}^n$  eine monotone, Lipschitz-stetige Abbildung und  $x \in \mathbb{R}^n$ . Dann ist jedes Element aus  $\partial_B S(x)$  positiv semidefinit.

*Beweis.* Sei  $\bar{x}$  ein differenzierbarer Punkt.

Es wird zunächst gezeigt, dass  $\nabla S(\bar{x})$  positiv semidefinit ist.

Angenommen  $\nabla S(\bar{x})$  ist nicht positiv semidefinit, das heißt es existieren eine Konstante  $\alpha > 0$  und ein  $d \in \mathbb{R}^n$  mit  $\|d\| = 1$  und  $d^\top \nabla S(\bar{x})d = -\alpha$ . Für ein beliebiges  $t > 0$  sei  $\phi(t) = S(\bar{x} + td) - S(\bar{x}) - t\nabla S(\bar{x})d$ .

Da  $S$  in  $\bar{x}$  differenzierbar ist, gilt  $\|\phi(t)\| = o(t)$ , für  $t \rightarrow 0$ . Aus der Monotonie von  $S$  folgt außerdem

$$0 \leq \langle td, S(\bar{x} + td) - S(\bar{x}) \rangle \quad (4.51)$$

$$= \langle td, \phi(t) + t\nabla S(\bar{x})d \rangle \quad (4.52)$$

$$= \langle td, t\nabla S(\bar{x})d \rangle + \langle td, \phi(t) \rangle \quad (4.53)$$

$$= -t^2\alpha + \langle td, \phi(t) \rangle \quad (4.54)$$

und mit der Cauchy-Schwarz'schen Ungleichung

$$\leq -t^2\alpha + t\|d\|\|\phi(t)\| \quad (4.55)$$

$$= -t^2\alpha + o(t^2). \quad (4.56)$$

Dies ist jedoch ein Widerspruch zur Positivität von  $\alpha$ . Somit ist  $\nabla S(\bar{x})$  positiv semidefinit.

Für jedes  $x \in \mathbb{R}^n$  und für alle  $J \in \partial_B S(x)$  existiert eine Folge von differenzierbaren Punkten  $x^k \rightarrow x$ , sodass  $\nabla S(x^k) \rightarrow J$  gilt. Da alle  $\nabla S(x^k)$  positiv semidefinit sind, ist auch  $J$  positiv semidefinit.  $\square$

Wenn  $P_{\tilde{g}}$  semiglatt ist, so ist Satz 4.6 auf die betrachtete Abbildung  $T$  anwendbar. Die positive Semidefinitheit der Elemente des B-Subdifferentials liefert auch die positive Semidefinitheit der Elemente aus Clarkes generalisierter Jacobi-Matrix, da diese nur Konvexkombinationen aus dem B-Subdifferential enthält.

Somit ist der Block  $I - K$  im ursprünglichen System (4.9) positiv semidefinit. Auch der Block  $1/\tau(K^{-1} - I)$  im modifizierten System (4.14) ist positiv semidefinit. Dies lässt sich über die Diagonalisierung von  $K$  zeigen, da

$$K = U^\top D U \quad (4.57)$$

$$\Leftrightarrow I - K = U^\top U - U^\top D U \quad (4.58)$$

$$\Leftrightarrow I - K = U^\top (I - D) U \quad (4.59)$$

gilt, wobei  $U$  orthogonal und  $D$  eine Diagonalmatrix ist. Zudem gilt für eine positiv semidefinite Matrix  $I - K$ , dass  $I - D$  nur nichtnegative Einträge enthält. Damit besitzt auch  $D^{-1} - I$ , da  $K$  kontrahierend ist, ausschließlich nichtnegative Einträge. Daraus folgt

$$U^\top (D^{-1} - I) U = U^\top D^{-1} U - I = K^{-1} - I \quad (4.60)$$

und somit insgesamt auch die positive Semidefinitheit der Matrix  $K^{-1} - I$ . Analog lässt sich auch die positive Semidefinitheit der Matrix  $H^{-1} - I$  zeigen.

Da jedoch keine Aussage über die Definitheit von  $A$  getroffen werden kann, ist es nicht möglich, die Definitheit der modifizierten Newton-Matrix anzugeben.

Somit ist es mit der verwendeten Blockmatrix  $B$  nicht gelungen, ein System aufzustellen, welches symmetrisch positiv definit ist. Es kann also auch in dem Fall, dass die Inversen der Ableitungen der Prox-Schritte bestimmt werden können, kein Lösungsverfahren für symmetrisch positiv semidefinite Systeme angewendet werden. Daher wird im nächsten Kapitel an Stelle eines Newton-Verfahrens ein Levenberg-Marquardt-Ansatz gewählt, um das Problem (3.1) zu lösen.



# Semiglatte Levenberg-Marquardt-Algorithmus

In diesem Kapitel wird zunächst der klassische Levenberg-Marquardt-Algorithmus erläutert. Darauf aufbauend wird eine modifizierte Version des Algorithmus für semiglatte Funktionen entwickelt. Abschließend erfolgt die Konvergenzanalyse des vorgestellten modifizierten Algorithmus.

## 5.1 Klassischer Levenberg-Marquardt-Algorithmus

Der Levenberg-Marquardt-Algorithmus dient der Lösung des Minimierungsproblems

$$\arg \min_{x \in \mathbb{R}^n} \frac{1}{2} \|F(x)\|^2 \quad (5.1)$$

für eine stetig differenzierbare Funktion  $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ . Zur Minimierung wird ein Gauß-Newton-Ansatz genutzt, wobei anstelle einer Liniensuche das Trust-Region-Verfahren angewendet wird [23].

Die Idee eines Trust-Region-Verfahrens ist es, über eine lokale Approximation der Funktion innerhalb der aktuellen Trust-Region die nächste Iterierte zu bestimmen. Dazu muss in jeder Iteration eine Modell-Funktion über die Trust-Region minimiert werden. Die Größe der Region wird dabei in jeder Iteration angepasst, um eine ausreichende Verbesserung des Zielfunktionswertes zu erreichen [25].

Die Lösung des Problems (5.1) entspricht der Lösung von

$$x + \arg \min_{d \in \mathbb{R}^n} \frac{1}{2} \|F(x + d)\|^2 \quad (5.2)$$

für ein beliebiges  $x \in \mathbb{R}^n$ . Linearisiert man den Term  $\psi(d) = 1/2 \|F(x + d)\|^2$ , so erhält man  $\tilde{\psi}(d) = 1/2 \|F(x) + \nabla F(x)d\|^2$ . Es soll nun  $\tilde{\psi}$  unter der Bedingung, dass

## 5 Semiglatte Levenberg-Marquardt-Algorithmus

$\|d\|^2 \leq \Delta^2$  mit  $\Delta > 0$  gilt, minimiert werden. Im Folgenden gelten die Bezeichnungen:  $f_k := F(x_k)$  und  $\nabla F(x_k) := G_k$ . Für den Minimierer gilt

$$0 = \nabla_d \frac{1}{2} \|f + Gd\|^2 + \frac{\lambda}{2} (\|d\|^2 - \Delta^2) \quad (5.3)$$

$$\Leftrightarrow 0 = G^\top (f + Gd) + \lambda d \quad (5.4)$$

$$\Leftrightarrow d = - \left( G^\top G + \lambda I \right)^{-1} G^\top f \quad (5.5)$$

mit  $\lambda \geq 0$  [23]. Es muss nun also nach Nocedal und Wright [25] das Trust-Region Subproblem

$$d = - \left( G^\top G + \lambda I \right)^{-1} G^\top f, \quad (5.6)$$

so dass  $\|d\| \leq \Delta$

gelöst werden.

Aufgrund der Symmetrie von  $G^\top G$  ist eine Eigenwertzerlegung  $G^\top G = Q\Lambda Q^\top$  möglich. Daraus ergibt sich

$$d = - \left( Q\Lambda Q^\top + \lambda I \right)^{-1} G^\top f \quad (5.7)$$

$$= -Q (\Lambda + \lambda I)^{-1} Q^\top G^\top f, \quad (5.8)$$

da  $Q$  eine orthogonale Matrix ist [25].

Es bezeichne  $q_j$  die  $j$ -te Spalte von  $Q$  und  $\sigma_j(G^\top G)$  den  $j$ -ten Eigenwert von  $G^\top G$ . Dann lässt sich für ein beliebiges  $\lambda$ , welches nicht das Negative eines Eigenwertes von  $G^\top G$  ist, das Subproblem aufgrund der Orthogonalität von  $Q$  schreiben als

$$d = - \sum_{j=1}^n \frac{q_j^\top G^\top f}{\sigma_j(G^\top G) + \lambda} q_j, \quad (5.9)$$

$$\text{sodass } \|d\| = \sqrt{\sum_{j=1}^n \frac{(q_j^\top G^\top f)^2}{(\sigma_j(G^\top G) + \lambda)^2}} \leq \Delta. \quad (5.10)$$

Für den Fall, dass  $\lambda$  größer ist als  $-\sigma_n(G^\top G)$ , ist  $\|d\|$  auf dem offenen Intervall  $(-\sigma_n(G^\top G), \infty)$  stetig und monoton fallend gegen 0 für  $\lambda \rightarrow \infty$ . Zudem gilt, falls  $q_1^\top G^\top f \neq 0$  ist, dass  $\lim_{\lambda \rightarrow -\sigma_1(G^\top G)} \|d\| = \infty$ . Somit gibt es in diesem Fall eine eindeutige Lösung  $\lambda^*$  der Nebenbedingung auf diesem Intervall. Andere Lösungen außerhalb des Intervalls sind auch möglich. Diese werden hier nicht berücksichtigt, da sie nicht die positive Definitheit von  $G^\top G + \lambda I$  liefern [25]. Positive Semidefinitheit ist jedoch nach folgendem Lemma von Moré und Sorensen [24] eine notwendige Bedingung für eine Lösung des Trust-Region-Subproblems.

**Lemma 5.1** ([24, Lemma 2.1])

Für eine Lösung  $d^*$  des Trust-Region-Subproblems (5.6) existiert ein  $\lambda \geq 0$ , sodass

- $G^\top G + \lambda I$  positiv semidefinit ist und
- $\lambda(\Delta - \|d^*\|) = 0$  gilt.

Der Fall  $q_1^\top G^\top f = 0$  wird nach [24] als „hard case“ bezeichnet, da er nicht zwingend eine Lösung des Subproblems (5.6) im Intervall  $(-\sigma_n(G^\top G), \infty)$  liefert, weil der Grenzwert  $\lim_{\lambda \rightarrow -\sigma_n(G^\top G)} \|d\|$  nicht gegen  $\infty$  geht.

Um eine Lösung für das Subproblem (5.6) zu finden, werden zwei Funktionen eingeführt:

$$\phi_1(\lambda) = \|d(\lambda)\| - \Delta \quad (5.11)$$

und

$$\phi_2(\lambda) = \frac{1}{\|d(\lambda)\|} - \frac{1}{\Delta}. \quad (5.12)$$

Zur Erfüllung der Nebenbedingung in (5.6) muss  $\phi_1(\lambda) = 0$  gelten. Für einen Trust-Region-Radius  $\Delta > 0$  ist dies äquivalent zu der Bedingung, dass  $\phi_2(\lambda) = 0$  gilt. Somit kann das Subproblem (5.6) durch Anwendung des Newton-Verfahrens auf  $\phi_2$  gelöst werden. Die Updates haben dann nach [25] die Form

$$\lambda_{k+1} = \lambda_k - \frac{\phi_2(\lambda_k)}{\phi_2'(\lambda_k)}. \quad (5.13)$$

Nachdem das Subproblem (5.6) näherungsweise gelöst worden ist, werden der Trust-Region-Radius angepasst und die nächste Iterierte bestimmt. Dies geschieht jeweils in Abhängigkeit von dem Verhältnis

$$\rho = \frac{\frac{1}{2}\|F(x_k)\|^2 - \frac{1}{2}\|F(x_k + d_k)\|^2}{\frac{1}{2}\|F(x_k)\|^2 - \frac{1}{2}\|F(x_k) + \nabla F(x_k)d_k\|^2} \quad (5.14)$$

der tatsächlichen Verbesserung des Zielfunktionswertes durch die Anwendung des bestimmten Schrittes und der mittels der Linearisierung vorhergesagten Verbesserung. Je kleiner dieses Verhältnis ist, desto deutlicher weicht die tatsächliche Reduktion von der Vorhersage ab. Ist das Verhältnis klein, so wird der Trust-Region-Radius verringert und für sehr kleine  $\rho$  wird ohne Update der Iterierten erneut das Subproblem (5.6) gelöst. Ist  $\rho$  jedoch groß, so wird der Radius vergrößert, falls die Lösung des Subproblems (5.6) auf dem Rand des zulässigen Bereichs liegt [25].

## 5.2 Modifizierter Levenberg-Marquardt-Algorithmus

Es wird nun ein Levenberg-Marquardt-Algorithmus für das Problem (3.1) entwickelt. Um ein Minimierungsproblem der Form (5.1) zu erhalten, werden erneut (3.11) und

## 5 Semiglatte Levenberg-Marquardt-Algorithmus

(3.12) betrachtet. Da diese eine Nullstellensuche erfordern, kann auch die Norm der beiden rechten Seiten, also des Vektors

$$F(u, v) := \begin{pmatrix} u - P_{\tau g}(u + \tau(-A^\top v - c)) \\ v - P_{\sigma f^*}(v + \sigma(Au - b)) \end{pmatrix}, \quad (5.15)$$

betrachtet werden. Diese nimmt ihr Minimum an der Nullstelle an.

Im Folgenden sei  $x = (u^\top, v^\top)^\top$ .

Für die Linearisierung wird aufgrund der Nicht-Glattheit von  $F(u, v)$ , wie im semiglaten Newton-Ansatz zuvor, auch Clarkes generalisierte Jacobi-Matrix genutzt. Man erhält für ein beliebiges  $G^* \in \partial_C F(x)$

$$\tilde{\psi}(d) = \frac{1}{2} \|F(x) + G^* d\|^2 \quad (5.16)$$

als Linearisierung. Daraus ergibt sich das Trust-Region Problem

$$\min_{d \in \mathbb{R}^n} \frac{1}{2} \|F(x) + G^* d\|^2, \quad (5.17)$$

$$\text{so dass } \|d\| \leq \Delta, \quad \Delta > 0. \quad (5.18)$$

Durch Einführung des Lagrange-Multiplikators  $\lambda \geq 0$  erhält man durch Differentiation im Optimum

$$\begin{aligned} 0 &= \nabla_d \left( \frac{1}{2} \|F(x) + G^* d\|^2 + \frac{\lambda}{2} (\|d\|^2 - \Delta^2) \right) \\ \Leftrightarrow 0 &= G^{*\top} (F(x) + G^* d) + \lambda d \\ \Leftrightarrow d &= - \left( G^{*\top} G^* + \lambda I \right)^{-1} G^{*\top} F(x). \end{aligned} \quad (5.19)$$

Zur Untersuchung der Verbesserung wird die Linearisierung in das Ausgangsproblem (5.1) eingesetzt. Da hauptsächlich der Quotient aus realer Verbesserung und vorhergesagter Verbesserung betrachtet wird, wird

$$\phi_k(d_k) := \|F(x_k) + G_k^* d_k\|^2 \quad (5.20)$$

für das zu  $G_k^* \in G$  gehörende Element  $d_k$  definiert.

Aus (5.19) erhält man

$$\begin{aligned} \left( G_k^{*\top} G_k^* + \lambda I \right) d &= -G_k^{*\top} F(x) \\ \Leftrightarrow -\|G_k^* d\|^2 - \lambda \|d\|^2 &= d^\top G_k^{*\top} F(x), \end{aligned} \quad (5.21)$$

wodurch sich die vorhergesagte Verbesserung

$$\begin{aligned} \phi_k(0) - \phi_k(d_k) &= \|F(x_k)\|^2 - \|F(x_k) + G_k^* d_k\|^2 \\ &= -2F(x_k)^\top G_k^* d_k - \|G_k^* d_k\|^2 \\ &= 2\|G_k^* d_k\|^2 + 2\lambda \|d_k\|^2 - \|G_k^* d_k\|^2 \\ &= \|G_k^* d_k\|^2 + 2\lambda \|d_k\|^2 \end{aligned} \quad (5.22)$$

ergibt.

Da die Überprüfung der positiven Definitheit im klassischen Trust-Region-Ansatz sehr rechenintensiv ist, haben Nocedal und Yuan eine Kombination aus Trust-Region-Verfahren und Liniensuche [26] vorgeschlagen. Diese wird durchgeführt, falls in der ersten Iteration zur Lösung des Trust-Region Subproblems (5.19) keine Verbesserung des Zielfunktionswertes erreicht werden konnte. Die Liniensuche erfolgt dabei in der Richtung, welche in der ersten Iteration bestimmt worden ist.

Für eine Funktion  $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ , die semiglatt ist, wird der folgende Levenberg-Marquardt-Algorithmus betrachtet:

---

**Algorithmus 2** Levenberg-Marquardt
 

---

**Eingabe:**  $x_0 \in \mathbb{R}^n$ ,  $\Delta_k > 0$ ,  $0 < \rho < 1$ ,  $0 < \sigma_1 < 1 < \sigma_2$ ,  $\lambda_{mc} > \varepsilon > 0$

```

1:  $k = 0$ 
2: while Abbruchkriterium nicht erfüllt do
3:   Wähle  $G_k \in \partial_C F(x_k)$  beliebig
4:    $b_r = \|G_k^\top G_k\| + \frac{1+\lambda_{mc}}{\Delta_k} \|G_k^\top F(x_k)\|$ 
5:    $linsearch = \text{FALSE}$ 
6:    $\lambda = 0$ 
7:   while  $\|d_k\| > \Delta_k$  do
8:     while TRUE do
9:       Führe Cholesky-Zerlegung  $G_k^\top G_k + \lambda I = R_k^\top R_k$  durch
10:      if Cholesky-Zerlegung erfolgreich then
11:        break
12:      else
13:         $\lambda = \min\{b_r, \lambda + \varepsilon\}$ 
14:      end if
15:    end while
16:    Löse  $R_k^\top R_k d_k = -G_k^\top F(x_k)$  nach  $d_k$ 
17:    if  $\|d_k\| > \Delta_k$  then
18:      Löse  $R^\top q_k = d_k$ , nach  $q_k$ 
19:      Setze  $\lambda = \lambda + \frac{\|d_k\|^2}{\|q_k\|^2} \cdot \frac{\|d_k\| - \Delta_k}{\Delta_k}$ 
20:    end if
21:  end while
22:  while  $\|F(x_k + d_k)\| > \|F(x_k)\|$  do
23:     $linsearch = \text{TRUE}$ 
24:     $\alpha = \max\left\{0, 9; 1/\left(2 + \frac{2(\|F(x_k)\| - \|F(x_k + d_k)\|)}{d_k^\top G_k^\top F(x_k)}\right)\right\}$ 
25:     $d_k = \alpha d_k$ 
26:  end while
27:   $x_{k+1} = x_k + d_k$ 
28:  if  $linsearch = \text{TRUE}$  then
29:     $\Delta_k = 0,9\Delta_k$ 
30:  else
31:     $\rho_k = \frac{\|F(x_k)\|^2 - \|F(x_k + d_k)\|^2}{2\lambda\|d_k\|^2 + \|(G_k d_k)\|^2}$ 

```

```

32:   if  $\rho_k \geq \rho$  then
33:     if  $\|d_k\| = \Delta_k$  then
34:        $\Delta_k = \sigma_2 \Delta_k$ 
35:     end if
36:   else
37:      $\Delta_k = \sigma_1 \Delta_k$ 
38:   end if
39: end if
40:  $k = k + 1$ 
41: end while

```

---

### 5.3 Analyse des modifizierten Levenberg-Marquardt-Algorithmus

Für die Analyse des modifizierten Levenberg-Marquardt-Algorithmus werden im Folgenden die Ideen aus dem Konvergenzbeweis von [26, Kapitel 3] genutzt.

Es gelte  $f_k = F(x_k)$ . Es wird zunächst eine untere Schranke für die vorhergesagte maximale Verbesserung gesucht. Die maximale Verbesserung in der  $k$ -ten Iteration für eine feste Wahl  $G_k^* \in \partial_C F(x_k)$  ergibt sich aus der Linearisierung und hat die Form

$$\phi_k(0) - \phi_k(d_k^*) = -2f_k^\top G_k^* d_k^* - \|G_k^* d_k^*\|^2, \quad (5.23)$$

wobei  $d_k^*$  die Lösung des Trust-Region Subproblems (5.19) in der  $k$ -ten Iteration ist. Diese kann durch eine beliebige Wahl von  $d_k$  nach unten abgeschätzt werden, da der Minimierer des Problems die Verbesserung maximiert.

Der folgende Satz liefert eine untere Schranke für eine Wahl von  $d_k = -\gamma G_k^{*\top} f_k$  mit  $\gamma \geq 0$  und damit auch eine Abschätzung für die maximale Verbesserung der Vorhersage.

**Satz 5.2** ([28, Theorem 4])

Sei  $d_k^* = \arg \min_{d_k \in \mathbb{R}^n} \phi_k(d_k)$  unter den Nebenbedingungen, dass  $\|d_k\| \leq \Delta_k$  und außerdem  $d_k = -\gamma G_k^{*\top} f_k$  für  $\gamma \geq 0$  gilt. Dann folgt, dass

$$\phi_k(0) - \phi_k(d_k^*) \geq \|G_k^{*\top} f_k\| \min \left\{ 2\Delta_k, \frac{\|G_k^{*\top} f_k\|}{\|G_k^{*\top} G_k^*\|} \right\}. \quad (5.24)$$

*Beweis.* Betrachtet man  $\phi_k(d_k^*)$ , so ergibt sich

$$\phi_k(0) - \phi_k(d_k^*) = -2f_k^\top G_k^* d_k^* - \|G_k^* d_k^*\|^2 \quad (5.25)$$

$$= 2\gamma f_k^\top G_k^* G_k^{*\top} f_k - \gamma^2 \|G_k^* G_k^{*\top} f_k\|^2 \quad (5.26)$$

$$= 2\gamma \|G_k^{*\top} f_k\|^2 - \underbrace{\gamma^2 \|G_k^* G_k^{*\top} f_k\|^2}_{\geq 0}. \quad (5.27)$$

Es werden nun zwei Fälle betrachtet.

*Fall 1:*  $\|G_k^* G_k^{*\top} f_k\|^2 = 0$ .

Man erhält in diesem Fall  $\phi_k(0) - \phi_k(d_k^*) = 2\gamma \|G_k^{*\top} f_k\|^2$ . Da diese Verbesserung maximal sein soll, wird  $\gamma$  maximal gewählt. Aus den Nebenbedingungen erhält man

$$\gamma = \frac{\|d_k^*\|}{\|G_k^{*\top} f_k\|} \quad (5.28)$$

$$\leq \frac{\Delta_k}{\|G_k^{*\top} f_k\|}. \quad (5.29)$$

Durch Einsetzen des maximalen Wertes für  $\gamma$  folgt also

$$\phi_k(0) - \phi_k(d_k^*) = 2\Delta_k \|G_k^{*\top} f_k\|. \quad (5.30)$$

*Fall 2:*  $\|G_k^* G_k^{*\top} f_k\|^2 > 0$ .

In diesem Fall wird das Maximum durch Ableiten nach  $\gamma$  bestimmt. Es ergibt sich die Bedingung

$$0 = 2\|G_k^{*\top} f_k\|^2 - 2\gamma \|G_k^* G_k^{*\top} f_k\|^2 \quad (5.31)$$

$$\geq 2\|G_k^{*\top} f_k\|^2 - 2\gamma \|G_k^{*\top} f_k\|^2 \|G_k^{*\top} G_k^*\|. \quad (5.32)$$

Daraus folgt

$$\gamma \geq \frac{1}{\|G_k^{*\top} G_k^*\|} \quad (5.33)$$

und

$$\gamma \|G_k^{*\top} f_k\|^2 = \gamma^2 \|G_k^* G_k^{*\top} f_k\|^2. \quad (5.34)$$

Damit erhält man die untere Schranke

$$\phi_k(0) - \phi_k(d_k^*) = \gamma \|G_k^{*\top} f_k\|^2 \quad (5.35)$$

$$\geq \frac{\|G_k^{*\top} f_k\|^2}{\|G_k^{*\top} G_k^*\|}. \quad (5.36)$$

Insgesamt folgt also die Behauptung

$$\phi_k(0) - \phi_k(d_k^*) \geq \|G_k^{*\top} f_k\| \min \left\{ 2\Delta_k, \frac{\|G_k^{*\top} f_k\|}{\|G_k^{*\top} G_k^*\|} \right\}. \quad (5.37) \quad \square$$

Zusätzlich zu der Schranke für die maximale Verbesserung in der Vorhersage, kann auch eine obere Schranke für  $d_k^{*\top} G_k^{*\top} f_k$  bestimmt werden. Über diese können Bedingungen für einen angemessenen Abstieg angegeben werden.

**Satz 5.3** ([26, Lemma 2.4])

Sei  $d_k^* = \arg \min_{d_k \in \mathbb{R}^n} \phi_k(d_k)$  unter der Nebenbedingung, dass  $\|d_k\| \leq \Delta_k$ . Dann gilt

$$d_k^{*\top} G_k^{*\top} f_k \leq -\frac{1}{2} \|G_k^{*\top} f_k\| \min \left\{ \Delta_k, \frac{\|G_k^{*\top} f_k\|}{\|G_k^{*\top} G_k^*\|} \right\}. \quad (5.38)$$

*Beweis.* Es wird eine Fallunterscheidung durchgeführt.

*Fall 1:*  $\|d_k^*\| < \Delta$ .

Mit der dritten Bedingung aus Lemma 5.1 folgt in diesem Fall, dass  $\lambda = 0$ . Einsetzen in (5.19) liefert

$$d_k^{*\top} G_k^{*\top} f_k = -f_k^\top G_k^* \left( G_k^{*\top} G_k^* \right)^{-1} G_k^{*\top} f_k \quad (5.39)$$

$$\leq -\frac{\|G_k^{*\top} f_k\|^2}{\|G_k^{*\top} G_k^*\|} \quad (5.40)$$

$$\leq -\frac{\|G_k^{*\top} f_k\|^2}{2\|G_k^{*\top} G_k^*\|}. \quad (5.41)$$

*Fall 2:*  $\|d_k^*\| = \Delta_k$ .

Zur Abschätzung werden die Eigenwerte der positiv semidefiniten Matrix  $G_k^{*\top} G_k^* + \lambda I$  genutzt. Die positive Semidefinitheit garantiert, dass alle Eigenwerte nichtnegativ sind.

Für eine beliebige reelle Matrix  $A$  gibt es eine Diagonalisierung  $A^\top A = U^\top D U$ , wobei  $D$  eine Diagonalmatrix und  $U$  eine orthogonale Matrix ist. Daher gilt die Gleichheit

$$\|Ad\|^2 = \langle d, A^\top A d \rangle = \langle d, U^\top D U d \rangle = \sum_i \sigma_i(A)^2 (Ud)_i^2. \quad (5.42)$$

Die  $d_i$  bezeichnen dabei die Elemente des Vektors  $d$  und die  $\sigma_i(A)$  sind die Singulärwerte der Matrix  $A$  und ihre Quadrate sind die Singulärwerte von  $D$ .

Man kann nun die Singulärwerte durch den betragsmäßig kleinsten Singulärwert  $\sigma_n(A)$  abschätzen und erhält

$$\|Ad\|^2 \geq \sum_i \sigma_n(A)^2 (Ud)_i^2. \quad (5.43)$$

Durch Ziehen der Wurzel ergibt sich weiter

$$\Rightarrow \|Ad\| \geq |\sigma_n(A)| \|Ud\| = |\sigma_n(A)| \|d\|, \quad (5.44)$$

### 5.3 Analyse des modifizierten Levenberg-Marquardt-Algorithmus

da  $U$  orthogonal ist. Zusammen mit (5.19) erhält man die Abschätzung

$$\left(G_k^{*\top} G_k^* + \lambda I\right) d_k = -G_k^{*\top} f_k \quad (5.45)$$

$$\Rightarrow \sigma_n \left(G_k^{*\top} G_k^* + \lambda I\right) \|d\| \leq \|G_k^{*\top} f_k\| \quad (5.46)$$

$$\Rightarrow \sigma_n \left(G_k^{*\top} G_k^*\right) + \lambda \leq \frac{\|G_k^{*\top} f_k\|}{\|d\|} \quad (5.47)$$

$$\begin{aligned} \Leftrightarrow \lambda &\leq \frac{\|G_k^{*\top} f_k\|}{\|d\|} - \sigma_n \left(G_k^{*\top} G_k^*\right) \\ &= \frac{\|G_k^{*\top} f_k\|}{\Delta_k} - \sigma_n \left(G_k^{*\top} G_k^*\right) \end{aligned} \quad (5.48)$$

für  $\lambda$ . Es wird nun erneut (5.19) betrachtet

$$d_k^{*\top} G_k^{*\top} f_k = -f_k^\top G_k^* \left(G_k^{*\top} G_k^* + \lambda I\right)^{-1} G_k^{*\top} f_k \quad (5.49)$$

$$\leq -\frac{\|G_k^{*\top} f_k\|^2}{\sigma_1 \left(G_k^{*\top} G_k^* + \lambda I\right)} \quad (5.50)$$

$$= -\frac{\|G_k^{*\top} f_k\|^2}{\sigma_1 \left(G_k^{*\top} G_k^*\right) + \lambda} \quad (5.51)$$

und die Abschätzung für  $\lambda$  genutzt, sodass

$$d_k^{*\top} G_k^{*\top} f_k \leq -\frac{\|G_k^{*\top} f_k\|^2}{\sigma_1 \left(G_k^{*\top} G_k^*\right) - \sigma_n \left(G_k^{*\top} G_k^*\right) + \frac{\|G_k^{*\top} f_k\|}{\Delta_k}} \quad (5.52)$$

gilt. Da  $G_k^{*\top} G_k^* + \lambda I$  nur nicht-negative Eigenwerte besitzt, gilt

$$\|G_k^{*\top} G_k^*\| \geq \sigma_1 \left(G_k^{*\top} G_k^*\right) \quad (5.53)$$

$$\geq \sigma_1 \left(G_k^{*\top} G_k^*\right) - \sigma_n \left(G_k^{*\top} G_k^*\right). \quad (5.54)$$

Damit ergibt sich

$$d_k^{*\top} G_k^{*\top} f_k \leq -\frac{\|G_k^{*\top} f_k\|^2}{\|G_k^{*\top} G_k^*\| + \frac{\|G_k^{*\top} f_k\|}{\Delta_k}} \quad (5.55)$$

$$\leq -\frac{\|G_k^{*\top} f_k\|^2}{2 \max \left\{ \|G_k^{*\top} G_k^*\|, \frac{\|G_k^{*\top} f_k\|}{\Delta_k} \right\}} \quad (5.56)$$

$$= -\frac{1}{2} \|G_k^{*\top} f_k\| \min \left\{ \Delta_k, \frac{\|G_k^{*\top} f_k\|}{\|G_k^{*\top} G_k^*\|} \right\}. \quad (5.57)$$

Zusammen folgt die Behauptung.  $\square$

## 5 Semiglatte Levenberg-Marquardt-Algorithmus

In Satz 5.2 und Satz 5.3 wurde jeweils von einer exakten Lösung des Trust-Region Subproblems (5.19) ausgegangen. Im Allgemeinen wird der Algorithmus jedoch nur eine näherungsweise Lösung liefern. Daher werden die Grenzen noch einmal unter der zusätzlichen Voraussetzung untersucht, dass

$$\lambda \leq \|G_k^{*\top} G_k^*\| + (1 + \lambda_{mc}) \frac{\|G_k^{*\top} f_k\|}{\Delta_k}, \quad (5.58)$$

wobei  $\lambda_{mc} > 0$ , sodass  $G_k^\top G_k + \lambda I$  positiv definit ist. Diese Bedingung ist auch in Zeile 13 von Algorithmus 2 zu finden. Eine derartige Beschränkung von  $\lambda$  stellt außerdem keine Einschränkung dar. Denn aus (5.48) im Beweis zu Satz 5.3 ist bekannt, dass

$$\lambda \leq \frac{\|G_k^{*\top} f_k\|}{\Delta_k} - \sigma_n \left( G_k^{*\top} G_k^* \right) \quad (5.59)$$

$$\leq \|G_k^{*\top} G_k^*\| + (1 + \lambda_{mc}) \frac{\|G_k^{*\top} f_k\|}{\Delta_k} \quad (5.60)$$

gilt.

### Lemma 5.4 ([26, S. 9f.]

Sei  $d_k^*(\lambda) = \arg \min_{d_k(\lambda) \in \mathbb{R}^n} \phi_k(d_k(\lambda))$  unter den Nebenbedingungen, dass  $\|d_k(\lambda)\| \leq \Delta_k$

und außerdem  $d_k(\lambda) = -\gamma G_k^{*\top} f_k$  für  $\gamma \geq 0$  gilt. Zudem sei die Matrix  $G_k^{*\top} G_k^* + \lambda I$  positiv definit, wobei  $\lambda$  die Ungleichung

$$\lambda \leq \|G_k^{*\top} G_k^*\| + (1 + \lambda_{mc}) \frac{\|G_k^{*\top} f_k\|}{\Delta_k} \quad (5.61)$$

für  $\varepsilon > 0$  erfülle. Dann sind die Ungleichungen

$$d_k^*(\lambda)^\top G_k^{*\top} f_k \leq -\frac{\|G_k^{*\top} f_k\|}{2} \min \left\{ \frac{\|G_k^{*\top} f_k\|}{2\|G_k^{*\top} G_k^*\|}, \frac{\Delta_k}{1 + \lambda_{mc}} \right\} \quad (5.62)$$

und

$$\phi_k(0) - \phi_k(d_k^*(\lambda)) \geq \frac{\|G_k^{*\top} f_k\|}{2} \min \left\{ \frac{\|G_k^{*\top} f_k\|}{2\|G_k^{*\top} G_k^*\|}, \frac{\Delta_k}{1 + \lambda_{mc}} \right\} \quad (5.63)$$

erfüllt.

*Beweis.* Unter den Voraussetzungen gilt mit (5.19)

$$d_k^*(\lambda)^\top G_k^{*\top} f_k = -f_k^\top G_k^* \left( G_k^\top G_k + \lambda I \right)^{-1} G_k^{*\top} f_k \quad (5.64)$$

$$\leq -\frac{\|G_k^{*\top} f_k\|^2}{\|G_k^{*\top} G_k^*\| + \lambda} \quad (5.65)$$

$$\leq -\frac{\|G_k^{*\top} f_k\|^2}{2\|G_k^{*\top} G_k^*\| + (1 + \lambda_{mc}) \frac{\|G_k^{*\top} f_k\|}{\Delta_k}} \quad (5.66)$$

$$\leq -\frac{\|G_k^{*\top} f_k\|^2}{2 \max \left\{ 2\|G_k^{*\top} G_k^*\|, (1 + \lambda_{mc}) \frac{\|G_k^{*\top} f_k\|}{\Delta_k} \right\}} \quad (5.67)$$

$$= -\frac{\|G_k^{*\top} f_k\|}{2} \min \left\{ \frac{\|G_k^{*\top} f_k\|}{2\|G_k^{*\top} G_k^*\|}, \frac{\Delta_k}{1 + \lambda_{mc}} \right\}. \quad (5.68)$$

Damit lässt sich auch eine Schranke für die vorhergesagte Verbesserung angeben. Dazu werden die beiden Gleichungen (5.21) und (5.22) genutzt:

$$\phi_k(0) - \phi_k(d_k^*(\lambda)) = \|G_k^* d_k(\lambda)\|^2 + 2\lambda \|d_k(\lambda)\|^2 \quad (5.69)$$

$$= -d_k(\lambda)^\top G_k^{*\top} f_k + \lambda \|d_k(\lambda)\|^2 \quad (5.70)$$

$$\geq \frac{\|G_k^{*\top} f_k\|}{2} \min \left\{ \frac{\|G_k^{*\top} f_k\|}{2\|G_k^{*\top} G_k^*\|}, \frac{\Delta_k}{1 + \lambda_{mc}} \right\}. \quad (5.71) \quad \square$$

Es könnte also eine Schranke gefunden werden, die angibt, wie groß die Verbesserung in der  $k$ -ten Iteration mindestens ist, wenn man nur das linearisierte Problem betrachtet. Außerdem konnte gezeigt werden, dass über die berechnete Richtung  $d_k$  eine Abstiegsrichtung erreicht wird, die, falls  $\|G_k^{*\top} f_k\|$  nicht gegen 0 geht und  $\|G_k^{*\top} G_k^*\|$  nach oben beschränkt ist, mit  $G_k^{*\top} f_k$  einen Winkel über  $90^\circ$  einschließt.

## 5.4 Ausblick für den Konvergenzbeweis

Zum Beweis globaler Konvergenz des semiglatten Levenberg-Marquardt-Algorithmus sind weitere Schritte nötig. Diese gehen jedoch über den Rahmen dieser Arbeit hinaus. Daher soll in diesem Abschnitt ein Ansatz geliefert werden, um einen Konvergenzbeweis unter Zusatzvoraussetzungen zu führen.

Um ein Konvergenzkriterium zu finden, wird die Optimalitätsbedingung für das Problem (5.1) für ein semiglattes  $F$  betrachtet. Diese hat die Form

$$0 \in (\partial_C F(x_k))^\top F(x_k). \quad (5.72)$$

Wenn die Zielfunktion in (5.1) konvex ist, reicht diese Bedingung aus. Zudem liefert die Konvexität die Monotonie der Elemente aus Clarkes generalisierter Jacobi-Matrix [30]. Somit kann in diesem Fall wie beim Verfahren für differenzierbare Funktionen über die Eigenschaft

$$0 \in \lim_{k \rightarrow \infty} \|G_k^{*\top} F(x_k)\| \quad (5.73)$$

für eine beliebige Folge  $\{G_k^*\}$  von Subgradienten die globale Konvergenz des Verfahrens gezeigt werden.

Es werden zwei Annahmen getroffen, die für einen Beweis der globalen Konvergenz des Algorithmus 2 in analoger Form in der Arbeit von Nocedal und Wright [26] benötigt werden.

**Annahme 5.5**

1. Die Folge der Iterierten  $\{x_k\}$  liegt in einer abgeschlossenen, konvexen Teilmenge des  $\mathbb{R}^n$ .
2. Es existiert eine Konstante  $M$ , sodass für alle  $x \in \mathbb{R}^n$  und alle  $G^* \in \partial_C F(x)$  gilt, dass  $\|G^{*\top} G^*\| \leq M$ .

In dem Algorithmus für zweimal stetig differenzierbare Funktionen von Nocedal und Yuan [26] wird die globale Konvergenz über einen Widerspruchsbeweis gezeigt. Daher wird auch hier die Negation des Konvergenzkriteriums (5.73)

$$\forall G_k^* \in G_k : \|G_k^* f_k\| \geq \delta > 0 \tag{5.74}$$

betrachtet. Daraus folgt mit Satz 5.2 und Satz 5.3, dass eine Konstante  $\mu > 0$  existiert, sodass

$$\phi_k(0) - \phi_k(d_k) \geq \mu \min \left\{ \Delta_k, \frac{1}{\|G_k^\top G_k\|} \right\} \tag{5.75}$$

und

$$d_k^\top G_k^{*\top} f_k \leq -\mu \min \left\{ \Delta_k, \frac{1}{\|G_k^\top G_k\|} \right\}. \tag{5.76}$$

Mittels des folgenden Satzes lässt sich die Suchrichtung kontrollieren.

**Satz 5.6** ([26, Lemma 3.3])

Es gelten die Annahmen 5.5. Zudem gebe es für einen Startindex  $N$  unendlich viele Indizes  $k_j > N$ ,  $j = 1, 2, \dots$ , für die keine Liniensuche durchgeführt werden muss,  $\rho_{k_j} < \rho$  gilt und außerdem die Ungleichung

$$\|d_{k_j}\| < \frac{1}{M_{k_j}} \min \{\mu(1 - \rho), 1\} \tag{5.77}$$

für

$$M_k = 1 + \max_{1 \leq t \leq k} \|G_t^{*\top} G_t^*\| \tag{5.78}$$

erfüllt ist. Dann gilt

$$\|d_{k_j}\| \leq \frac{\mu(1 - \rho)}{\|G_{k_j}^{*\top} G_{k_j}^*\|}. \tag{5.79}$$

*Beweis.* Die Voraussetzungen implizieren für alle  $j = 1, 2, \dots$ , da aufgrund des Algorithmus  $\|d_{k_j}\| \leq \Delta_{k_j}$  ist, dass

$$\frac{\|d_{k_j}\|}{\min\{\Delta_{k_j}, 1/M_{k_j}\}} \leq 1 \quad (5.80)$$

gilt.

Es wird zunächst gezeigt, dass

$$\lim_{k \rightarrow \infty} \min\left\{\Delta_k, \frac{1}{M_k}\right\} = 0 \quad (5.81)$$

erfüllt ist. Dafür wird zuerst nur eine endliche Indexmenge  $\mathcal{I}$ , welche alle  $i$  mit  $\rho_i \geq \rho$  enthält, betrachtet. Der Algorithmus garantiert, dass die Folge  $\{\|f_i\|^2\}$  monoton fällt. Zudem ist sie durch 0 nach unten beschränkt. Daher kann die tatsächliche Gesamtverbesserung im Algorithmus mittels  $\frac{\|f_i\|^2 - \|f_{i+1}\|^2}{\phi_k(0) - \phi_k(d_i)} = \rho_i \geq \rho$  durch

$$\infty > \sum_{i=1}^{\infty} \|f_i\|^2 - \|f_{i+1}\|^2 \geq \sum_{i \in \mathcal{I}} \|f_i\|^2 - \|f_{i+1}\|^2 \quad (5.82)$$

$$\geq \rho \sum_{i \in \mathcal{I}} \phi_k(0) - \phi_k(d_i) \quad (5.83)$$

$$\geq \rho \mu \sum_{i \in \mathcal{I}} \min\left\{\Delta_i, \frac{1}{\|G_i^{*\top} G_i^*\|}\right\} \quad (5.84)$$

abgeschätzt werden. Somit muss auch

$$\sum_{i \in \mathcal{I}} \min\left\{\Delta_i, \frac{1}{\|M_i\|}\right\} < \infty \quad (5.85)$$

gelten. Dies impliziert, dass die Aussage auf der Menge  $\mathcal{I}$

$$\lim_{\substack{i \rightarrow \infty \\ i \in \mathcal{I}}} \min\left\{\Delta_i, \frac{1}{M_i}\right\} = 0 \quad (5.86)$$

wahr ist. Für Indizes  $i \notin \mathcal{I}$  werden zwei Fälle unterschieden, in denen jeweils  $\rho_i < \rho$  ist.

*Fall 1:*  $\mathcal{I}$  ist endlich.

Aus Zeile 37 von Algorithmus 2 folgt direkt, dass  $\Delta_{i+1} < \Delta_i$ . Dies muss für alle großen  $i$  gelten, da  $\mathcal{I}$  endlich ist. Somit konvergiert  $\{\Delta_i\}$  gegen 0 und damit gilt die Aussage (5.81) für ein endliches  $\mathcal{I}$ .

*Fall 2:*  $\mathcal{I}$  ist nicht endlich.

Sei  $l \notin \mathcal{I}$  und  $\hat{l}$  der nächstkleinere Index, der in  $\mathcal{I}$  enthalten ist. Da  $\{M_k\}$  monoton

## 5 Semiglatte Levenberg-Marquardt-Algorithmus

fällt und aus dem Algorithmus folgt, dass  $\Delta_l \leq \sigma_1 \Delta_{\hat{l}}$ , wobei  $\sigma_1 < 1$  gewählt werden muss, gilt

$$\min \left\{ \Delta_l, \frac{1}{M_l} \right\} \leq \min \left\{ \sigma_1 \Delta_{\hat{l}}, \frac{1}{M_{\hat{l}}} \right\}. \quad (5.87)$$

Es kann wieder der Grenzwert in (5.86) betrachtet werden, da  $\hat{l}$  in  $\mathcal{I}$  liegt. Somit folgt die Aussage (5.81) auch für eine nicht endliche Menge  $\mathcal{I}$ .

Da  $\|d_{k_j}\| \leq \Delta_{k_j}$  sein muss, gilt mit (5.86)

$$\lim_{j \rightarrow \infty} \|d_{k_j}\| = 0. \quad (5.88)$$

Mit (5.77) kann gefolgert werden, dass

$$M_{k_j} \|d_{k_j}\| \leq \mu(1 - \rho) \quad (5.89)$$

gilt und über die Definition der  $M_{k_j}$  ergibt sich dann die Aussage

$$\|d_{k_j}\| \leq \frac{\mu(1 - \rho)}{\|G_{k_j}^{*\top} G_{k_j}^*\|}. \quad (5.90) \quad \square$$

Mithilfe von Satz 5.6 und weiteren Voraussetzungen könnte eine untere Schranke für  $\|d_k\|$  gefunden werden. Mittels dieser Schranke könnte es dann auch möglich sein, in Anlehnung an [26] und mithilfe der hier gezeigten Aussagen, Bedingungen für die globale Konvergenz des Verfahrens zu formulieren und eine Konvergenzordnung anzugeben. Diese Überlegungen werden jedoch im Rahmen dieser Arbeit nicht fortgeführt.

# Ergebnisse

In diesem Abschnitt werden Experimente für das konvexe Rudin-Osher-Fatemi-Problem [31] mit den verschiedenen in dieser Arbeit vorgestellten Algorithmen präsentiert.

## 6.1 ROF-Problem

Das ROF-Modell ist eine variationelle Methode zur Rekonstruktion eines durch gaußsches Rauschen gestörten diskretisierten Bildes  $h \in \mathbb{R}^n$ . Dazu wird ein  $\mathcal{L}^2$ -Datenterm genutzt, der mit totaler Variation regularisiert wird. Diese ist wie folgt definiert:

**Definition 6.1** (Totale Variation; [4, Def. 1.1])

Für eine Abbildung  $u \in \mathcal{L}^1(\Omega, \mathbb{R})$  ist

$$\text{TV}(u) := \sup_{\substack{\phi \in \mathcal{C}_c^1(\Omega, \mathbb{R}^n), \\ \|\phi\| \leq 1}} \left\{ - \int_{\Omega} u \operatorname{div} \phi(x) dx \right\} \quad (6.1)$$

die *totale Variation*.

Für ein stetig differenzierbares  $u$  gilt nach [4]

$$\text{TV}(u) = \int_{\Omega} \|\nabla u(x)\|_2 dx. \quad (6.2)$$

Da nur diskretisierte Bilder betrachtet werden sollen, wird auch die totale Variation diskretisiert. Für ein diskretisiertes Bild auf einem Gebiet  $\Omega$  hat sie dann die Form

$$\text{TV}(u) = \sum_{x \in \Omega} \|\nabla u(x)\|_2 =: \|\nabla u\|_{2,1}. \quad (6.3)$$

## 6 Ergebnisse

Damit lässt sich das ROF-Problem für ein diskretes Bild der Größe  $n$  als

$$\min_{u \in \mathbb{R}^n} \frac{1}{2} \|u - h\|_2^2 + \beta \|\nabla u\|_{2,1} \quad (6.4)$$

schreiben [4].

Die Regularisierung mit dem Gradienten von  $u$  im betrachteten Modell sorgt für eine lokale Glättung des Eingabebildes  $h$ . Ein Nachteil des Modells ist jedoch, dass es zu einem Kontrastverlust führt.

Dieses Minimierungsproblem ist konvex und hat die Form 3.1. Dabei beschreibt, in der Notation aus Kapitel 3, die Matrix  $A$  den diskretisierten Gradienten und  $b$  und  $c$  sind die entsprechenden Nullvektoren. Die Funktion  $g(u) = 1/2 \|u - h\|_2^2$  ist konvex und zweimal stetig differenzierbar. Die Funktion  $f(z) = \beta \|z\|_{2,1}$  ist unterhalbstetig, eigentlich und konvex.

Um die Algorithmen anwenden zu können, wird die Konjugierte  $f^*$  benötigt. Da für eigentliche Funktionen  $f_i(x_i)$  gilt, dass  $(\sum_i f_i(x_i))^* = \sum_i f_i(x_i)^*$  ist [30, Prop. 11.22], wird wie in [3, Bsp 3.26] die zu  $f_a(z) = \|z\|$  konjugierte Norm  $\|v\|_* = \sup_{\|z\| \leq 1} z^\top v$  betrachtet. Nach der Definition der Konjugierten 2.8 ist  $f_a^*(v) = \sup_z \{v^\top z - \beta \|z\|\}$ . Es wird nun eine Fallunterscheidung durchgeführt.

*Fall 1:*  $\|v\|_* \leq \beta$ .

Es gilt also für alle  $z$ , dass  $v^\top z \leq \beta \|z\|$  ist. Gleichheit gilt dabei genau dann, wenn  $z = 0$  ist. Somit muss in diesem Fall  $f_a^*(v) = 0$  gelten.

*Fall 2:*  $\|v\|_* > \beta$ .

Es wird ein  $z$  betrachtet, für welches  $\|z\| \leq 1/\beta$  gilt. Dann muss  $z^\top v > 1$  sein. Da  $f_a^*(v)$  ein Supremum über alle  $z$  ist, gilt also für ein beliebiges  $t > 0$ , dass

$$f_a^*(v) \geq v^\top(tz) - \beta \|tz\| = t(v^\top z - \beta \|z\|). \quad (6.5)$$

Für  $t \rightarrow \infty$  geht der letzte Term gegen unendlich.

Daher hat die Konjugierte von  $f_a$  die Form

$$f_a^*(v) = \begin{cases} 0, & \|v\|_* \leq \beta, \\ \infty, & \text{sonst} \end{cases} =: \delta_{\beta B(0,1)}(v). \quad (6.6)$$

Dies ist die Indikatorfunktion für eine Kugel mit dem Radius  $\beta$  mit dem Zentrum im Ursprung. Die hier betrachtete euklidische Norm ist zu sich selbst dual. Somit erhält man

$$(f^*(v))_i = \begin{cases} 0, & \|v_i\| \leq \beta, \\ \infty, & \text{sonst} \end{cases} \quad (6.7)$$

für alle  $i = 1, \dots, n$ .

Im nächsten Schritt müssen die Prox-Schritte auf  $f^*$  und  $g$  berechnet werden. Da  $g$  differenzierbar ist, kann der Prox-Schritt exakt über die Optimalitätsbedingung (2.22)

bestimmt werden:

$$P_{\tau g}(u) = \arg \min_z \left\{ g(z) + \frac{1}{2\tau} \|z - u\|^2 \right\} \quad (6.8)$$

$$\Rightarrow 0 = \nabla g(z) + \frac{1}{\tau}(z - u) \quad (6.9)$$

$$= z - h + \frac{1}{\tau}(z - u) \quad (6.10)$$

$$\Leftrightarrow z = \frac{1}{1 + \tau}(u + \tau h). \quad (6.11)$$

Somit ist die Proximal-Abbildung auf  $g$  stetig differenzierbar und daher auch semiglatt. Die Differentiation liefert dann

$$\partial_C P_{\tau g}(u) = \frac{1}{1 + \tau} I. \quad (6.12)$$

$\partial_C P_{\tau g}$  ist also konstant.

Auch zur Berechnung des Prox-Schrittes auf  $f^*$  genügt es wieder, nur  $f_a$  zu betrachten, da die euklidischen Normen  $\|(\nabla u)_i\|$  nicht gekoppelt sind. Für den Prox-Schritt auf  $f_a^*$  vereinfacht sich das Minimierungsproblem, denn

$$P_{\sigma f_a^*}(v) = \arg \min_z \left\{ f^*(z) + \frac{1}{\sigma} \|z - v\|^2 \right\} \quad (6.13)$$

$$= \arg \min_{\|z\| \leq \beta} \left\{ \frac{1}{\sigma} \|z - v\|^2 \right\} \quad (6.14)$$

$$= \arg \min_{\|z\| \leq \beta} \{ \|z - v\|^2 \} \quad (6.15)$$

entspricht der Projektion auf die Kugel mit dem Radius  $\beta$ . Somit ist

$$P_{\sigma f_a^*}(v) = \begin{cases} v, & \|v\| \leq \beta, \\ \beta \frac{v}{\|v\|}, & \text{sonst} \end{cases} \quad (6.16)$$

und

$$\partial_C P_{\sigma f_a^*}(v) \ni \begin{cases} I, & \|v\| \leq \beta, \\ \frac{\beta}{\|v\|} \left( I - \frac{vv^\top}{\|v\|^2} \right), & \text{sonst.} \end{cases} \quad (6.17)$$

Also gilt für  $f$

$$(\partial_C P_{\sigma f^*}(v))_i \ni \begin{cases} I, & \|v_i\| \leq \beta, \\ \frac{\beta}{\|v_i\|} \left( I - \frac{v_i v_i^\top}{\|v_i\|^2} \right), & \text{sonst} \end{cases} \quad (6.18)$$

für alle  $i = 1, \dots, n$ . Der Prox-Schritt (6.16) ist nach Wu et al. [35] auch semiglatt, da die metrische Projektion auf eine Kugel stückweise affin ist.

Nun können die verschiedenen vorgestellten Algorithmen auf das ROF-Problem angewendet werden.

## 6.2 Experimente

Im Folgenden werden der semiglatte Newton-Algorithmus ebenso wie das modifizierte semiglatte Newton-Verfahren und der vorgestellte modifizierte Levenberg-Marquardt-Algorithmus genutzt, um ein Phantom (siehe Abbildung 6.1) sowie einen Ausschnitt aus dem linken Auge in dem Bild „Lena“ (siehe Abbildung 6.2), welche mit gaußischem Rauschen gestört wurden, zu rekonstruieren. Dazu wurden die drei Algorithmen in MATLAB [1] implementiert. Die beiden Testbilder haben eine Größe



**Abbildung 6.1:** (a) manuell erstelltes Originalphantom, (b) mit gaußischem Rauschen gestörtes Phantom



**Abbildung 6.2:** (a) linkes Auge und (b) mit gaußischem Rauschen gestörtes Auge aus dem Bild „Lena“

von  $20 \times 20$  Pixeln. Die Grauwerte sind auf das Intervall  $[0,1]$  skaliert. Da ein primal-dualer Ansatz gewählt wurde, werden als Abbruchkriterien einerseits die Größe der normierten Duality-Gap und andererseits die Höhe der Infeasibility

verwendet.

Die normierte Duality-Gap ist für die primalen und dualen Zielfunktionswerte  $o_p$  und  $o_d$  durch

$$\frac{o_p - o_d}{o_d} \quad (6.19)$$

definiert. Sie soll kleiner als  $2^{-26}$  sein. Dieser Wert entspricht der Wurzel aus der Präzision `eps` in MATLAB. Die Infeasibility gibt an, wie groß die maximale Verletzung einer Nebenbedingung im primalen und dualen Problem ist. Für die  $n_p$  Nebenbedingungen des primalen und  $n_d$  Nebenbedingungen des dualen Problems der Form

$$\eta_{p_i}(u) \leq 0, \quad i = 1, \dots, n_p \quad (6.20)$$

$$\eta_{d_j}(v) \leq 0, \quad j = 1, \dots, n_d \quad (6.21)$$

ergibt sich also die Infeasibility

$$\max \left\{ 0, \eta_{p_1}, \dots, \eta_{p_{n_p}}, \eta_{d_1}, \dots, \eta_{d_{n_d}} \right\}. \quad (6.22)$$

In den durchgeführten Untersuchungen sollte sie jeweils kleiner als  $10^{-10}$  sein. Diese Werte orientieren sich an der Genauigkeit des MOSEK-Lösers in CVX [14, 15].

Der Regularisierungsparameter  $\beta$  wurde für das Entrauschen des Phantoms auf 0,2 gesetzt, für das Auge von „Lena“ auf 0,1. Die Parameter  $\sigma$  und  $\tau$  wurden jeweils auf 1 gesetzt. Diese Werte ergaben sich aus dem optischen Eindruck bei Parametertests. Die Berechnungen wurden auf einem System mit

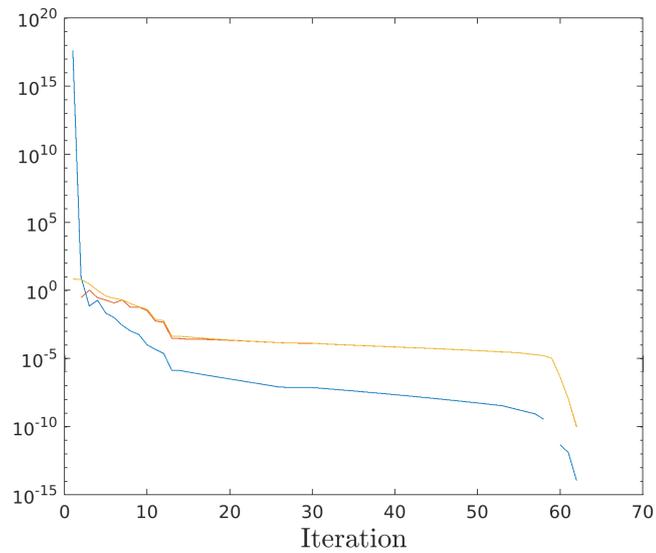
- Intel Core i7-2600 Prozessor,
- Nvidia GeForce GTX 480 Grafikkarte und
- 16 GB Arbeitsspeicher

durchgeführt.

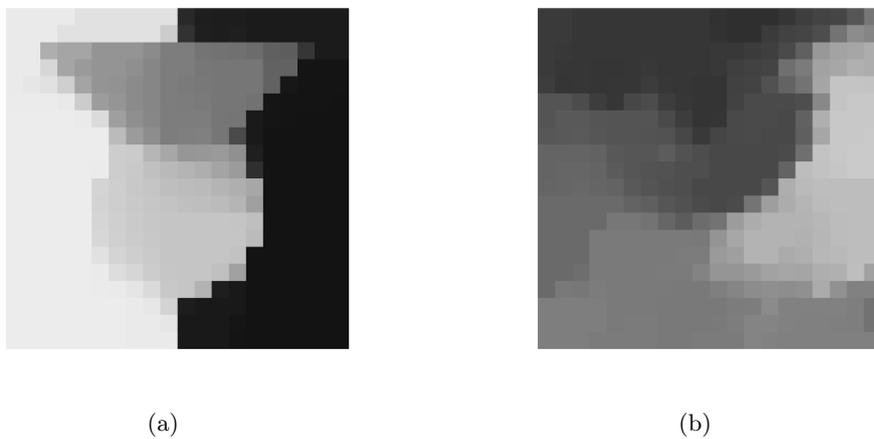
### 6.2.1 Semiglattes Newton-Verfahren

Mittels der obigen Berechnungen lässt sich das Newton-System aufstellen. Zu der Newton-Matrix muss, um die Invertierbarkeit zu sichern, ein Vielfaches der Einheitsmatrix addiert werden. Die Einheitsmatrix wurde in den Berechnungen mit einem Faktor von  $10^{-7}$  multipliziert.

Wie in Abbildung 6.3 zu erkennen ist, konvergiert das Verfahren für das Phantom nach 61 Iterationen. Die Lücke in der blauen Kurve ist dadurch zu erklären, dass die normierte Duality-Gap in dieser Iteration negativ ist. Dies ist jedoch theoretisch mathematisch nicht möglich. Die Ursache dafür liegt vermutlich in numerischen Ungenauigkeiten. Die Laufzeit beträgt für dieses Problem im Schnitt ungefähr 0,55 Sekunden.



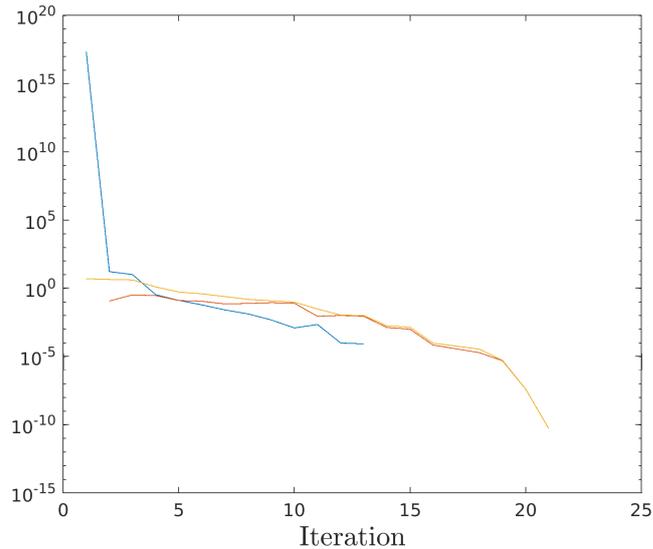
**Abbildung 6.3:** Normierte Duality-Gap (blau), Infeasibility (rot) und Residuum (gelb) in Abhängigkeit von der Iterationszahl bei Lösung des ROF-Problems für das Phantom in Abbildung 6.1 mit dem unmodifizierten semiglatten Newton-Verfahren.



**Abbildung 6.4:** Ergebnisse für das Entrauschen des (a) Phantoms und des (b) Auges mittels des ROF-Modells mit dem unmodifizierten semiglatten Newton-Verfahren.

Das geglättete Ergebnisbild ist in Abbildung 6.4 (a) zu sehen. Man erkennt im Vergleich mit dem Original, dass das Entrauschen mit dem ROF-Modell tatsächlich zu einem Kontrastverlust geführt hat.

Für das Auge konvergiert das Verfahren, wie in Abbildung 6.5 zu sehen ist, bereits nach weniger Iterationen. Die Laufzeit für dieses Bild beträgt im Schnitt etwa 0,14 Sekunden. Man erkennt in Abbildung 6.4 (b), dass durch das Entrauschen neben dem Kontrastverlust auch kleine Strukturen, wie beispielsweise die Wimpern unterhalb des Auges, nicht mehr erkennbar sind.

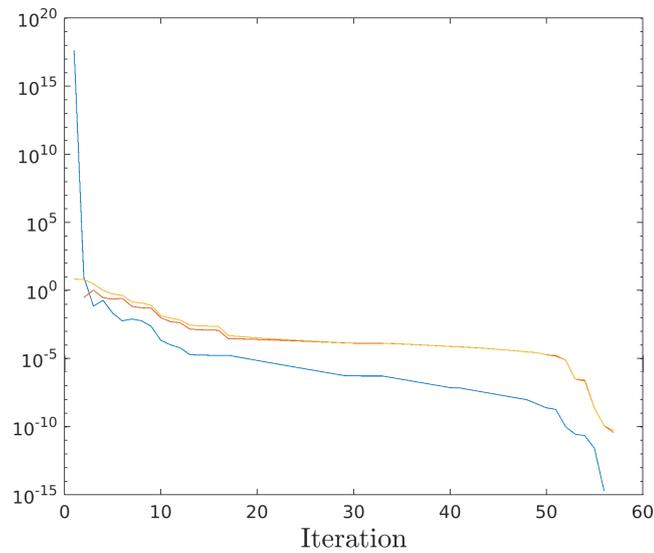


**Abbildung 6.5:** Normierte Duality-Gap (blau), Infeasibility (rot) und Residuum (gelb) in Abhängigkeit von der Iterationszahl bei Lösung des ROF-Problems für das Auge in Abbildung 6.2 mit dem unmodifizierten semiglatten Newton-Verfahren.

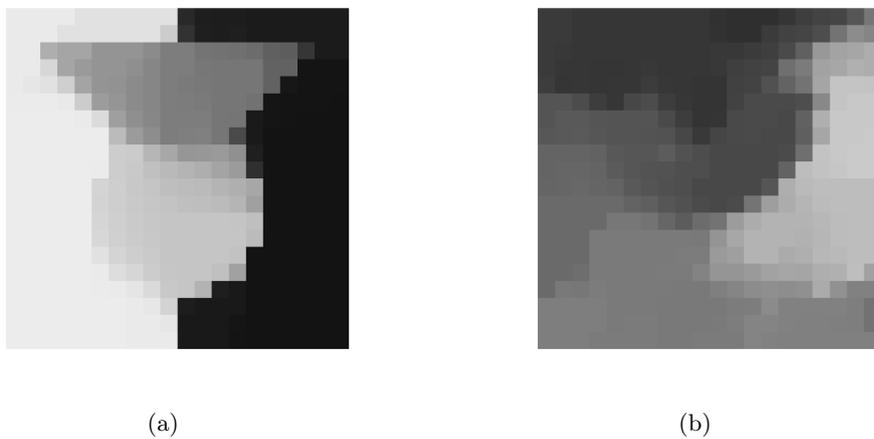
### 6.2.2 Modifiziertes semiglattes Newton-Verfahren

Im modifizierten semiglatten Newton-Verfahren werden zusätzlich zur Newton-Matrix auch auf die Matrizen  $K$  und  $H$  Vielfache der Einheitsmatrix addiert, um deren Invertierbarkeit zu sichern. Hier wurde ein Faktor von  $10^{-7}$  verwendet. Für das Newton-System wurde erneut der Faktor  $10^{-7}$  verwendet.

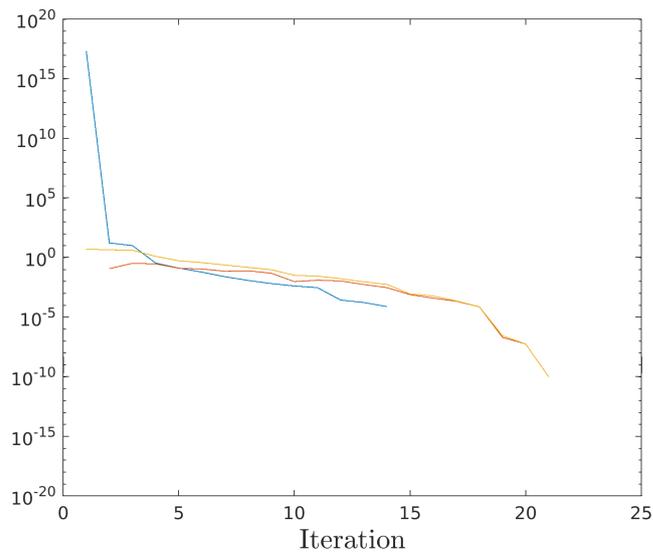
Auch mit dem modifizierten System konvergiert der Algorithmus für beide Testbilder. Es werden, wie in Abbildung 6.6 zu sehen ist, für das Phantom weniger Iterationen als im unmodifizierten semiglatten Newton-Verfahren benötigt. Allerdings wird mit 0,92 Sekunden ungefähr die 1,6-fache Laufzeit benötigt, da für die Modifikation Inverse berechnet werden müssen. Für das Auge unterscheidet sich die Anzahl der Iterationen nicht, siehe Abbildung 6.8. Die Laufzeit ist mit etwa 0,35



**Abbildung 6.6:** Normierte Duality-Gap (blau), Infeasibility (rot) und Residuum (gelb) in Abhängigkeit von der Iterationszahl bei Lösung des ROF-Problems für das Phantom in Abbildung 6.1 mit dem modifizierten semiglaten Newton-Verfahren.



**Abbildung 6.7:** Ergebnisse für das Entrauschen des (a) Phantoms und des (b) Auges mittels des ROF-Modells mit dem modifizierten semiglaten Newton-Verfahren.



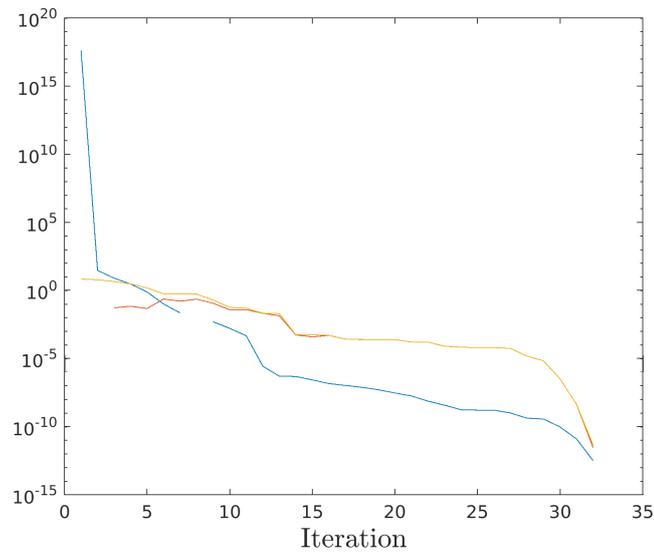
**Abbildung 6.8:** Normierte Duality-Gap (blau), Infeasibility (rot) und Residuum (gelb) in Abhängigkeit von der Iterationszahl bei Lösung des ROF-Problems für das Auge in Abbildung 6.2 mit dem modifizierten semiglatten Newton-Verfahren.

Sekunden jedoch mehr als doppelt so hoch wie im unmodifizierten Verfahren. Das Konvergenzverhalten der beiden Algorithmen ist jedoch für die zwei Bilder ähnlich. Auch die Ergebnisbilder (siehe Abbildung 6.7) unterscheiden sich optisch nicht. Da im modifizierten Verfahren eine zusätzliche Regularisierung bei der Berechnung der Inversen notwendig ist, bestehen zwischen dem modifizierten und dem unmodifizierten semiglatten Newton-Verfahren geringe Unterschiede, obwohl beide Algorithmen auf der Lösung äquivalenter Gleichungssysteme beruhen.

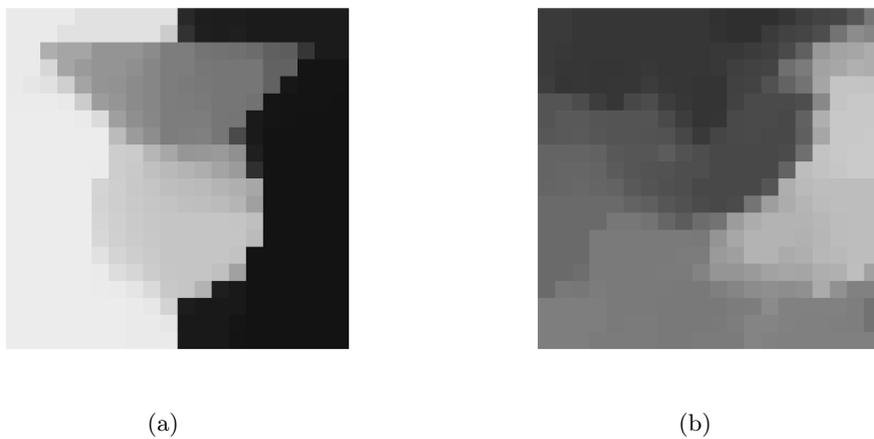
### 6.2.3 Modifizierter Levenberg-Marquardt-Algorithmus

Im modifizierten Levenberg-Marquardt-Algorithmus wurden die zusätzlichen Parameter für beide Testbilder folgendermaßen gewählt: Die Parameter zur Anpassung des Trust-Region-Radius betragen  $\sigma_1 = 0,5$  und  $\sigma_2 = 2$ . Der Lagrange-Multiplikator  $\lambda$  wird über die Parameter  $\lambda_{mc} = 2$ ,  $\varepsilon = 10^{-10}$  erhöht und das Verhältnis der Verbesserungen wird mit  $\rho = 0,75$  verglichen. Der Trust-Region-Radius wurde für die erste Iteration auf 1,5 gesetzt.

Wie in Abbildung 6.9 zu sehen, liefert der Algorithmus für das Entrauschen des Phantoms bereits nach 31 Iterationen ein ausreichend genaues Ergebnis. Der Grund für die Lücke in der blauen Kurve für die normierte Duality-Gap ist erneut ein negativer Wert. Auch hier ist die Ursache vermutlich numerischen Ursprungs. Da in jeder Iteration des Verfahrens mindestens eine Cholesky-Zerlegung durchgeführt



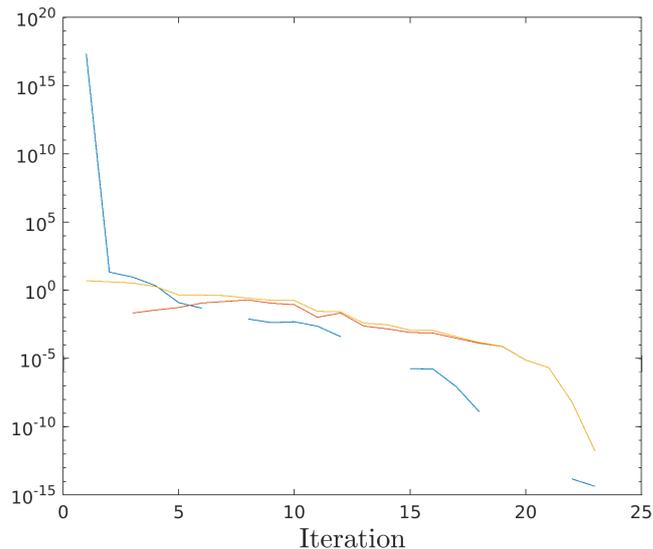
**Abbildung 6.9:** Normierte Duality-Gap (blau), Infeasibility (rot) und Residuum (gelb) in Abhängigkeit von der Iterationszahl bei Lösung des ROF-Problems für das Phantom in Abbildung 6.1 mit dem semiglattem Levenberg-Marquardt-Verfahren.



**Abbildung 6.10:** Ergebnisse für das Entrauschen des (a) Phantoms und des (b) Auges mittels des ROF-Modells mit dem semiglattem Levenberg-Marquardt-Verfahren.

werden muss, ist die Laufzeit mit etwa 9,48 Sekunden für das Phantom im Vergleich mit den beiden semiglatten Newton-Verfahren deutlich höher. Sie beträgt somit mehr als das zehnfache der Laufzeit des modifizierten semiglatten Newton-Algorithmus. Die Rechenzeit für das Auge ist mit ungefähr 7,43 Sekunden sogar mehr als zwanzig Mal höher als im modifizierten semiglatten Newton-Verfahren.

Die Ergebnisse in Abbildung 6.10 zeigen, dass auch mit dem Levenberg-Marquardt-Algorithmus visuell keine Unterschiede zu den Lösungen der semiglatten Newton-Algorithmen zu erkennen sind.



**Abbildung 6.11:** Normierte Duality-Gap (blau), Infeasibility (rot), Residuum (gelb) in Abhängigkeit von der Iterationszahl bei Lösung des ROF-Problems für das Auge in Abbildung 6.2 mit dem semiglatten Levenberg-Marquardt-Verfahren.

Ebenso wie bei der Entrauschung des Phantoms, sieht man auch in Abbildung 6.11 für das Auge von „Lena“, dass die Konvergenz in den letzten Iterationen annähernd linear ist.

#### 6.2.4 Vergleich mit weiteren Verfahren

Die drei vorgestellten Verfahren werden in diesem Abschnitt mit CVX [14, 15] mit dem Löser MOSEK, einem Interior-Point-Algorithmus, und der PDHG-Methode von Chambolle und Pock [5] mit festen Schrittweiten verglichen.

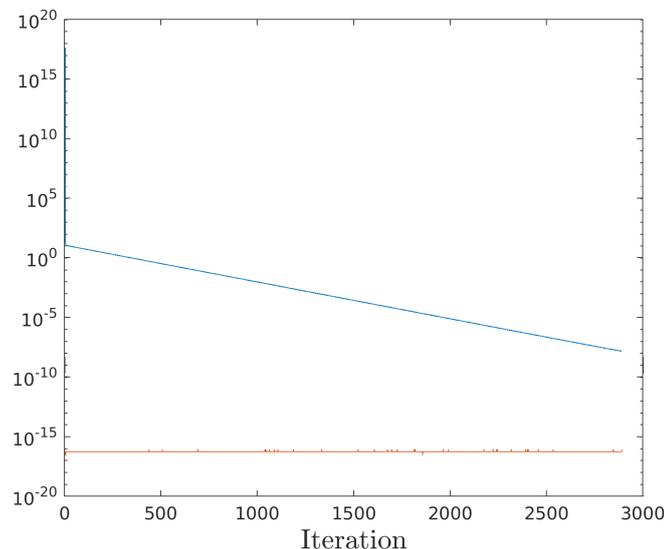
Die Abbruchkriterien von CVX sind mit den hier für die Beispiele verwendeten vergleichbar. CVX benötigt für das Entrauschen des Phantoms etwa 0,76 Sekunden. Damit ist das semiglatten Newton-Verfahren für das Phantom circa 28% schneller als CVX. Die maximale Abweichung des Grauwertes in einem Pixel zwischen den drei

Algorithmus	Laufzeit in Sekunden	
	Phantom	Auge
semiglattes Newton-Verfahren	0,55	0,14
modifiziertes semiglattes Newton-Verfahren	0,92	0,35
semiglattes Levenberg-Marquardt-Verfahren	9,48	7,43
CVX	0,76	0,64
PDHG-Methode	0,44	0,43

**Tabelle 6.1:** Laufzeitvergleich der drei vorgestellten Algorithmen mit CVX und der PDHG-Methode

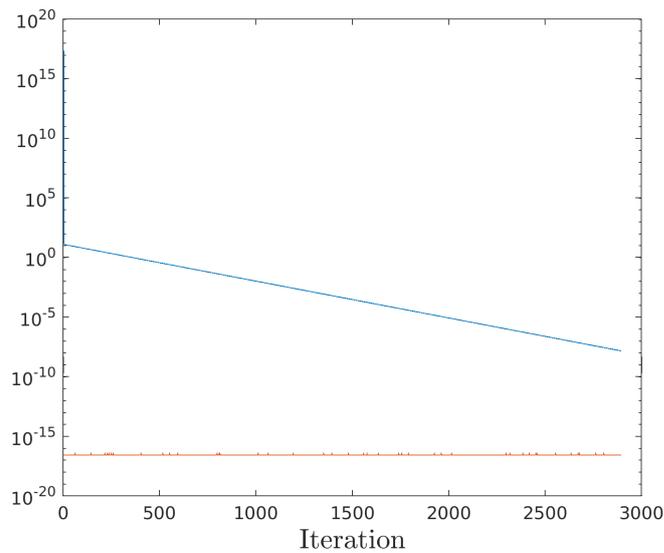
Algorithmen und CVX beträgt jeweils knapp  $3,1 \cdot 10^{-7}$ .

Für die Entfernung des gaußschen Rauschens aus dem Auge von „Lena“ benötigt CVX circa 0,64 Sekunden. Für dieses Bild sind also beide semiglatten Newton-Verfahren schneller als CVX. Die maximale Grauwertdifferenz zu dem MOSEK-Löser beträgt für alle drei Algorithmen etwa  $5,9 \cdot 10^{-7}$ .



**Abbildung 6.12:** Normierte Duality-Gap (blau) und Infeasibility (rot) in Abhängigkeit von der Iterationszahl bei Lösung des ROF-Problems für das Phantom in Abbildung 6.1 mit der PDHG-Methode.

Die PDHG-Methode benötigt im Vergleich zu den vorgestellten Algorithmen deutlich mehr Iterationen, siehe Abbildung 6.12 und Abbildung 6.13. Eine Iteration ist jedoch schneller, da keine Gleichungssysteme gelöst werden müssen. Sowohl die Laufzeit als auch die Entwicklung der Duality-Gap und der Infeasibility unterscheiden sich für die beiden Testbilder kaum. Die Infeasibility ist niedriger als die Maschinengenauigkeit und somit vernachlässigbar. Die PDHG-Methode liefert also für die hier betrachteten



**Abbildung 6.13:** Normierte Duality-Gap (blau) und Infeasibility (rot) in Abhängigkeit von der Iterationszahl bei Lösung des ROF-Problems für das Auge in Abbildung 6.2 mit der PDHG-Methode.

Probleme nur zulässige Lösungen.

Die maximale Grauwertdifferenz zu CVX liegt für beide Bilder im Bereich von  $10^{-5}$  und ist damit höher als für die drei neuen Algorithmen. Über die Qualität der Rekonstruktion lässt sich damit jedoch keine Aussage treffen. Betrachtet man die maximale Grauwertdifferenz zu den nicht verrauschten Bildern, so zeigt sich, dass die drei vorgestellten Algorithmen und CVX für das Phantom eine niedrigere Differenz, für das Auge jedoch eine höhere Differenz als die PDHG-Methode aufweisen.

Für Probleme, bei denen die Berechnung der Prox-Schritte aufwändiger oder nur approximativ möglich ist, bieten die entwickelten Verfahren zweiter Ordnung den Vorteil, weniger Iterationen zu benötigen.



## Diskussion und Ausblick

Es gibt viele Aufgabenstellungen in der Bildverarbeitung, die auf semiglatten, konvexe Optimierungsprobleme führen. Häufig, jedoch nicht immer, sind auch die zugehörigen Proximal-Abbildungen semiglatt.

Für derartige Probleme wurden drei Algorithmen vorgestellt, welche für das ROF-Problem an zwei Beispielen alle qualitativ gleich gute Ergebnisse liefern. Allerdings konnten Unterschiede in der Konvergenz und Laufzeit festgestellt werden. Die semiglatten Newton-Verfahren haben ein ähnliches Konvergenzverhalten. Grund dafür ist, dass beide Verfahren äquivalente Gleichungssysteme lösen. Die Unterschiede beruhen auf der zusätzlichen Regularisierung im modifizierten Verfahren und numerischen Ungenauigkeiten. Auch der semiglatten Levenberg-Marquardt-Algorithmus zeigt für das betrachtete Beispiel einen ähnlichen Konvergenzverlauf. Die Anzahl der benötigten Iterationen ist jedoch deutlich geringer. In den ersten Iterationen verbessert sich die Lösung in allen Verfahren deutlich, bevor sich sowohl das Residuum, als auch die Duality-Gap und die Unsicherheit nur noch langsam verbessern. Der Vergleich der Rechenzeiten zeigt, dass das Levenberg-Marquardt-Verfahren deutlich langsamer als die beiden Newton-Algorithmen ist, da in jeder Iteration mindestens eine Cholesky-Zerlegung durchgeführt werden muss. Je größer das Problem wird, desto deutlicher sind auch die Laufzeitdifferenzen.

Für den semiglatten Levenberg-Marquardt-Algorithmus konnten Voraussetzungen angegeben werden, unter denen in jeder Iteration ein ausreichender Abstieg vollzogen wird. Es ist bekannt, dass semiglatten Newton-Verfahren lokal mindestens  $Q$ -superlinear konvergieren. Dies gilt auch für die vorgeschlagenen Algorithmen, die auf semiglatten Prox-Schritten beruhen.

Ob ein ähnliches Ergebnis auch für den modifizierten Levenberg-Marquardt-Algorithmus gilt, bleibt in dieser Arbeit offen.

Als Verfahren zweiter Ordnung konvergieren die Algorithmen bereits nach weniger Iterationen als die zum Vergleich betrachtete PDHG-Methode. Dies kann bezüglich der Laufzeiten der Algorithmen ein Vorteil sein, wenn die Bestimmung der Proximal-Abbildungen oder einer Approximation an diese rechenintensiv ist. Durch eine effi-

## *7 Diskussion und Ausblick*

zientorientierte Implementierung können die Algorithmen beschleunigt werden. Zudem kann in zukünftigen Arbeiten eine optimierte Parameterwahl die Ergebnisse noch verbessern.

# Literaturverzeichnis

- [1] (2017). MATLAB Version 9.2 (R2017a). The Mathworks, Inc. Natick, Massachusetts.
- [2] Ben-Tal, A. und Nemirovski, A. (2001). Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications. MPS-SIAM Series on Optimization, SIAM, Philadelphia.
- [3] Boyd, S. und Vandenberghe, L. (2004). Convex Optimization. Cambridge University Press, Cambridge.
- [4] Chambolle, A., Caselles, V., Cremers, D., Novaga, M. und Pock, T. (2010). An introduction to Total Variation for Image Analysis. Theoretical Foundations and Numerical Methods for Sparse Recovery 9, 227.
- [5] Chambolle, A. und Pock, T. (2011). A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. Journal of Mathematical Imaging and Vision 40, 120–145.
- [6] Chan, T. F., Esedoglu, S. und Nikolova, M. (2006). Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models. SIAM Journal on Applied Mathematics 66, 1632–1648.
- [7] Clarke, F. H. (1983). Optimization and Nonsmooth Analysis. John Wiley & Sons, New York, Chichester.
- [8] Combettes, P. L. und Pesquet, J.-C. (2011). Proximal Splitting Methods in Signal Processing. In Fixed-Point Algorithms for Inverse Problems in Science and Engineering S. 185–212. Springer.
- [9] Esser, E., Zhang, X. und Chan, T. F. (2010). A General Framework for a Class of First Order Primal-Dual Algorithms for Convex Optimization in Imaging Science. SIAM Journal on Imaging Sciences 3, 1015–1046.
- [10] Forster, O. (2013). Analysis 2, Differentialrechnung im  $\mathbb{R}^n$ , gewöhnliche Differentialgleichungen. Zehnte Ausgabe, Springer Spektrum, Wiesbaden.

- [11] Forster, O. (2016). Analysis 1, Differential- und Integralrechnung einer Veränderlichen. Zwölfte Ausgabe, Springer Spektrum, Wiesbaden.
- [12] Forster, O. (2017). Analysis 3: Maß- und Integrationstheorie, Integralsätze im  $\mathbb{R}^n$  und Anwendungen. Achte Ausgabe, Springer Spektrum.
- [13] Goldstein, T., Li, M., Yuan, X., Esser, E. und Baraniuk, R. (2015). Adaptive Primal-Dual Hybrid Gradient Methods for Saddle-Point Problems. Preprint arXiv: 1305.0546.
- [14] Grant, M. und Boyd, S. (2008). Graph Implementations for Nonsmooth Convex Programs. In Recent Advances in Learning and Control, (Blondel, V., Boyd, S. und Kimura, H., eds), Lecture Notes in Control and Information Sciences S. 95–110. Springer-Verlag Limited.
- [15] Grant, M. und Boyd, S. (2014). CVX: Matlab Software for Disciplined Convex Programming, Version 2.1. <http://cvxr.com/cvx>.
- [16] He, B., You, Y. und Yuan, X. (2014). On the Convergence of Primal-Dual Hybrid Gradient Algorithm. SIAM Journal on Imaging Sciences 7, 2526–2537.
- [17] Hintermüller, M. (2010). Semismooth Newton Methods and Applications. In Oberwolfach Seminar on „Mathematics of PDE-Constrained Optimization“.
- [18] Hintermüller, M., Ito, K. und Kunisch, K. (2002). The Primal-Dual Active Set Strategy As a Semismooth Newton Method. SIAM J. on Optimization 13, 865–888.
- [19] Karmarkar, N. (1984). A New Polynomial-Time Algorithm for Linear Programming. In Proceedings of the sixteenth annual ACM symposium on Theory of computing S. 302–311, ACM.
- [20] Mifflin, R. (1977). Semismooth and Semiconvex Functions in Constrained Optimization. SIAM Journal on Control and Optimization 15, 959–972.
- [21] Milzarek, A. und Ulbrich, M. (2014). A Semismooth Newton Method with Multidimensional Filter Globalization for  $l_1$ -Optimization. SIAM Journal on Optimization 24, 298–333.
- [22] Moreau, J. (1965). Proximité et dualité dans un espace hilbertien. Bulletin de la Société Mathématique de France 93, 273–299.
- [23] Moré, J. J. (1978). The Levenberg-Marquardt Algorithm: Implementation and Theory. In Numerical Analysis, (Watson, G., ed.), vol. 630, of Lecture Notes in Mathematics S. 105–116. Springer Berlin Heidelberg.
- [24] Moré, J. J. und Sorensen, D. (1983). Computing a Trust Region Step. SIAM Journal on Scientific and Statistical Computing 3, 553–572.

- [25] Nocedal, J. und Wright, S. J. (2006). Numerical Optimization. Zweite Ausgabe, Springer, New York, NY, USA.
- [26] Nocedal, J. und Yuan, Y.-x. (1998). Combining Trust Region and Line Search Techniques. In Advances in nonlinear programming S. 153–175. Springer.
- [27] Pock, T. und Chambolle, A. (2011). Diagonal Preconditioning for First Order Primal-Dual Algorithms in Convex Optimization. In Computer Vision (ICCV), 2011 IEEE International Conference on S. 1762–1769, IEEE.
- [28] Powell, M. J. D. (1975). Convergence Properties of a Class of Minimization Algorithms. Nonlinear Programming 2, 1–27.
- [29] Qi, L. und Sun, J. (1993). A Nonsmooth Version of Newton’s Method. Mathematical programming 58, 353–367.
- [30] Rockafellar, R. T. und Wets, R. J.-B. (2009). Variational Analysis. Dritte Ausgabe, Springer Verlag, Heidelberg, Berlin, New York.
- [31] Rudin, L. I., Osher, S. und Fatemi, E. (1992). Nonlinear Total Variation Based Noise Removal Algorithms. Physica D 60, 259–268.
- [32] Sun, D. (2001). A Further Result on an Implicit Function Theorem for Locally Lipschitz Functions. Operations Research Letters 28, 193–198.
- [33] Ulbrich, M. (2001). Nonmonotone Trust-Region Methods for Bound-Constrained Semismooth Equations with Applications to Nonlinear Mixed Complementarity Problems. SIAM Journal on Optimization 11, 889–917.
- [34] Ulbrich, M. (2011). Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- [35] Wu, B., Ding, C., Sun, D. und Toh, K.-C. (2014). On the Moreau–Yosida Regularization of the Vector  $k$ -Norm Related Functions. SIAM Journal on Optimization 24, 766–794.
- [36] Xiao, X., Li, Y., Wen, Z. und Zhang, L. (2016). A Regularized Semi-Smooth Newton Method with Projection Steps for Composite Convex Programs. Preprint arXiv: 1603.07870.
- [37] Yuan, Y. (1983). Some Properties of Trust Region Algorithms for Nonsmooth Optimization. Technical Report DAMTP/NA4 University of Cambridge.