6 Numerische Optimierung

In den ersten fünf Kapiteln dieses Skriptes haben wir Grundaufgaben der biomedizinischen Bildgebung eingeführt, im Sinne von Variationsmethoden modelliert und ihre Analyse in geeigneten Funktionenräumen diskutiert. Letztendlich ist aber das Ziel, die Methoden auf konkrete Daten der Biomedizin anzuwenden. Da Variationsmethoden auf Minimierungsaufgaben für kontinuierliche Funktionen beruhen, führt dies einerseits zu der Frage nach geeigneten Verfahren zur Diskretisierung und anderseits zu der Frage nach passenden numerischen Optimierungsverfahren. In diesen Themengebieten existiert eine sehr große Anzahl an Originalarbeiten, die sich entweder mit relativ spezifischen Variationsmethoden zur Bildverarbeitung oder andererseits mit sehr abstrakten Konzepten der Optimierung beschäftigen.

In diesem Kapitel geht es darum, die Frage der Diskretisierung von unterschiedlichen Seiten zu beleuchten und Werkzeuge der numerischen Optimierung zur Verfügung zu stellen, die die Lösung von vorgestellten Variationsproblemen ermöglichen. Wir starten mit dem sehr grundlegenden Sachverhalt der Diskretisierung unendlich dimensionaler Optimierungsprobleme.

6.1 Diskretisierung

Die Lösung der Variationsprobleme aus den letzten Kapiteln ist ein Problem der Variationsrechnung, bei der man sich mit Optimierungsproblemen der Form

$$J(u) = \int_\Omega G(x, (Ku)(x), u(x), \nabla u(x))\, dx \;\rightarrow\; \min_u \tag{6.1}$$

beschäftigt.

Es gibt zwei unterschiedliche Vorgehensweisen zur Lösung von unendlich dimensionalen Optimierungsproblemen. Zum einen das Prinzip, "First discretize, then optimize" und zum anderen das Prinzip "First optimize, then discretize". Beide Strategien werden in der Forschung eingesetzt und es besteht weiterhin fortlaufend eine Diskussion über ihre jeweiligen Vor- und Nachteile.

(a) **First discretize, then optimize**: Die Idee dieses Ansatzes ist die unmittelbare Diskretisierung eines gegebenen Optimierungsproblems, d.h. die Ersetzung aller auftretenden Funktionenräume durch endlich dimensionale Räume, sowie der Ersetzung aller auftretenden Operatoren durch geeignete diskrete Pendanten. D.h. anstelle von (6.1) für einen Funktionenraum $\mathcal{U}$ betrachtet man

$$\min_{u_h \in U_h} J_h(u_h)$$

mit $J_h : U_h \to \mathbb{R}$, $U_h \subset \mathcal{U}$ und einem Diskretisierungsparameter h. Dies führt im Allgemeinen zu einem Problem der (diskreten) *nichtlinearen* Optimierung (engl.: nonlinear programming) in $\mathbb{R}^n$. Der Hauptvorteil besteht darin, dass man eine Vielzahl an existierenden, effizienten Methoden der *nichtlinearen* Optimierung (z.B. basierend auf *Innere Punkte Verfahren* oder Sequentielle Quadratische Programmierung (SQP) Verfahren) einsetzen kann. Ein Nachteil ist der Mangel an quantitativen Approximationsresultaten für nichtilineare Probleme.

(b) **First optimize, then discretize**: Die Idee dieses Ansatzes ist die Formulierung der Optimierungsmethode in unendliche dimensionalen Räumen und einer anschließenden Diskretisierung lediglich für die Lösung von (linearen oder quadratischen) Teilproblemen und für die Auswertung des Zielfunktionals. Anders ausgedrückt, man geht erst über zum Optimalitätssystem (Karush-Kuhn-Tucker System für beschränkte Optimierungsprobleme) und transferiert dann alle auftretenden Funktionenräume und Operatoren für eine diskrete algorithmische Umsetzung. Der Hauptvorteil dieser Strategie besteht darin, dass quantitative Abschätzungen für die Konvergenz von Optimierungsmethoden mit Fehlerabschätzungen für die Diskretisierung von Teilproblemen kombiniert werden können. Damit kann man Abschätzungen für den gesamten Fehler einer numerischen Optimierungsmethode ableiten.

Bis heute gibt es kein allgemeines Rezept, welcher der beiden Diskretisierungsstrategien vorzuziehen ist. Vielmehr hängt es von der Anwendung und den Ressourcen zum wissenschaftlichen Rechnen ab. Wichtig ist allerdings, dass der gewählte numerische Ansatz die Struktur des unendlich dimensionalen Optimierungsproblems zu einem gewissen Grad widerspiegelt und erhält.

Darüber hinaus kann es für beschränkte Optimierungsprobleme auch Sinn machen, Diskretisierungskonzepte nicht direkt auf das Ausgangsproblem wie in (a) oder direkt auf das Optimalitätssystem wie in (b) anzuwenden, sondern zunächst einen SQP-Ansatz auf der kontinuierlichen Ebene anzusetzen, um dann wie oben beschrieben fortzufahren.

Im Folgenden starten wir mit numerischen Methoden der beschränkten Optimierung und konzentrieren uns auf Verfahren, die dem zweiten Ansatz genügen, d.h. wir formulieren Optimierungsmethoden im unendlich dimensionalen Fall. Man beachte aber, dass die Resultate auch im Fall $\mathcal{U} = \mathbb{R}^n$ angewandt werden können.

6.2 Gradientenverfahren

Im Folgenden nehmen wir an, dass $\mathcal{U}$ ein Hilbertraum sei, falls nicht anders festgelegt. Zur Herleitung eines sehr einfachen numerischen Optimierungsverfahrens für (unbeschränkte) Optimierungsprobleme der Form (6.1) betrachten wir das Beispiel eines Regularisierungsfunktionals bzgl. $\mathcal{U} := H^1(\Omega) = W^{1,2}(\Omega)$:

$$J(u) := \frac{1}{2} \int_\Omega |\nabla u|^2 \, dx \ .$$

Der sogenannte Gradientenfluss ist definiert als $\frac{\partial u}{\partial t} = -J'(u)$ und ist in diesem Fall durch die Wärmeleitungsgleichung

$$\frac{\partial u}{\partial t} = \Delta u$$

gegeben. Um ein allgemeines Optimierungsverfahren zu erhalten, können wir einen Gradientenfluss in einem Hilbertraum $\mathcal{U}$ als

$$\frac{\partial u}{\partial t} = -J'(u) \tag{6.2}$$

einführen, wobei $J'(u) \in \mathcal{U}$ eine Element des Hilbertraums darstellt, das mit dem Gradienten von J an der Stelle u identifiziert werden kann. Mit anderen Worten, wir definieren die Evolution dieser Gleichung durch

$$\left\langle \frac{\partial u}{\partial t}, v \right\rangle = -J'(u)v \qquad \forall v \in \mathcal{U} \ ,$$

wobei $\langle \cdot \, , \, \cdot \rangle$ das Skalarprodukt in $\mathcal{U}$ bezeichnet. Die Evolution des Gradientenflusses impliziert eine Evolution des Zielfunktionals, die gegeben ist durch:

$$\frac{\partial}{\partial t} \left(J(u) \right) = J'(u) \frac{\partial u}{\partial t} = - \left\| \frac{\partial u}{\partial t} \right\|^2 \leq 0 \ .$$

Dies bedeutet, dass das Zielfunktional monoton fallend ist, und $\frac{\partial}{\partial t}(J(u)) = 0$ gilt, genau dann wenn $\frac{\partial u}{\partial_t} = 0$. Darüber hinaus impliziert die Struktur des Gradientenflusses (6.2), dass $\frac{\partial u}{\partial_t} = 0$ gilt, genau dann wenn $J'(u) = 0$, d.h. u ist ein stationärer (man sagt auch: kritischer) Punkt, bzw. erfüllt die notwendige Optimalitätsbedingung erster Ordnung. Folglich können wir erwarten, dass das Zielfunktional mit dem Gradientenfluss abfällt bis schließlich ein stationärer Punkt erreicht wird.

Um ein iteratives Optimierungsverfahren abzuleiten, diskretisieren wir das Optimalitätssystem in (6.2) mit Hilfe einer *expliziten Zeitdiskretisierung* des Flusses, d.h.

$$\begin{aligned} u_{k+1} &= u_k - \sigma_k \, J'(u_k) \\ &= u_k + \sigma_k \, d_k(u_k) \end{aligned}$$

mit $\sigma_k > 0$ als Schrittweite bzgl. einer (künstlichen) Iterations-Zeit und einer sogenannten Suchrichtung $d_k(u_k) := -J'(u_k)$. Dieses Verfahren bezeichnet man als *Gradientenverfahren*. Aus numerischer Sicht ist es offensichtlich, dass nur eine hinreichend kleine Wahl der Zeitschritte σ_k sinvoll ist, da explizite Zeitdiskretisierungen mit zu großen Schritten nicht stabil sind.

Das Gradientenverfahren, auch *Verfahren des steilsten Abstiegs* genannt, wurde bereits 1847 von Cauchy untersucht. Wie wir gesehen haben, bestimmt man bei diesem Verfahren im Punkt u_k diejenige Suchrichtung d_k, in der J am stärksten abnimmt. Man spricht von einer streng gradientenbezogenen Suchrichtung.

Die lokal optimale Sichtweise beim (projizierten) Gradientenverfahren muss global nicht die beste Vorgehensweise sein. Eine ungünstige Wahl der Schrittweite kann dazu führen, dass man keine globale Konvergenz erhält. Wir betrachten dazu das folgende Beispiel.

Beispiel 6.2.1. *Es sei $J(u) = u^2$ und $u_0 = 1$. Weiter sei*

$$d_k = -1 \quad \textit{und} \quad \sigma_k = \left(\frac{1}{2}\right)^{k+2} \qquad \forall k \geq 0 \; . \tag{6.3}$$

Dann ist

$$u_{k+1} = u_k - \sigma_k = u_0 - \sum_{i=0}^{k} \left(\frac{1}{2}\right)^{i+1} = \frac{1}{2} + \left(\frac{1}{2}\right)^{k+1} \; .$$

Also gilt $u_{k+1} < u_k$ und daher $J(u_{k+1}) < J(u_k)$ für alle $k \geq 0$, aber $u_k \to \frac{1}{2}$, d.h. (u_k) konvergiert nicht gegen das Minimum $u = 0$ von J.

Ist das Minimum eines Funktionals gesucht, so ist es bei gegebenem u_k naheliegend, bei der Berechnung von u_{k+1} das Ziel

$$J(u_{k+1}) < J(u_k) \tag{6.4}$$

anzustreben. Verfahren, die eine solche Strategie realisieren, nennt man *Abstiegsverfahren*. Auch wenn das Verfahren unter Umständen nicht gegen ein (lokales) Minimum konvergiert, so wird doch in jeder Iteration das Zielfunktional verkleinert und damit ein besserer Punkt berechnet, was in der Praxis oft schon zufriedenstellend ist.

Ein Abstiegsverfahren benutzt zur Berechnung von u_{k+1} eine Abstiegsrichtung, d.h. eine Suchrichtung mit der Eigenschaft

$$J(u_k + \sigma d_k) < J(u_k), \qquad \forall \sigma \in]0, s_k[$$

mit einem $s_k > 0$. Zur Konstruktion von Abstiegsverfahren gibt es zwei prinzipielle Vorgehensweisen:

(a) Verfahren mit Schrittweitensteuerung: Hier bestimmt man zunächst eine Abstiegsrichtung d_k aufgrund lokaler Informationen über die Zielfunktion im aktuellen Iterationspunkt u_k. Dann berechnet man eine Schrittweite $\sigma_k \in]0, s_k[$, mit der man einen möglichst großen Abstieg erzielt, und setzt $u_{k+1} = u_k + \sigma_k d_k$.

(b) Trust-Region-Verfahren: Hier wird basierend auf einem lokalen Modell des Zielfunktionals (beispielsweise einer quadratischen Approximation des Zielfunktionals) eine Trust-Region (Vertrauensbereich) berechnet, auf der das lokale Modell des Zielfunktionals hinreichend gut approximiert wird. Das lokale Modell erlaubt dann die Berechnung einer Abstiegsrichtung d_k, und man setzt $u_{k+1} = u_k + d_k$.

Im Folgenden konzentrieren wir uns auf Schrittweitenverfahren, um z.B. das Konvergenzverhalten des Gradientenverfahrens zu verbessern. Auf Trust-Region-Verfahren werden wir später im Zusammenhang von Levenberg-Marquardt nochmal eingehen.

6.3 Schrittweitenverfahren

Die fundamentale Idee der Verfahren in diesem Kapitel ist folgende:

(i) An einer Stelle u bestimmt man eine Suchrichtung d, bei der die Funktionalwerte reduziert werden (Abstiegsverfahren).

(ii) Beginnend bei u, bewegt man sich in Richtung d so weit, wie die Funktionalwerte von J *hinreichend reduziert* werden. (Schrittweiten Steuerung)

Um (global) konvergente Verfahren zu erhalten, müssen wir sogenannte effiziente Schrittweiten bestimmen.

Definition 6.3.1. *Für gegebenes u und gegebene Suchrichtung d mit $J'(u)d < 0$ erfüllt eine Schrittweite das* ***Prinzip des hinreichenden Abstiegs****, falls*

$$J(u + \sigma d) \leq J(u) + c_1 \sigma \, J'(u)d \tag{6.5}$$

und

$$\sigma \geq -c_2 \frac{J'(u)d}{\|d\|^2} \tag{6.6}$$

mit einer von u und d unabhängigen Konstanten $c_1, c_2 > 0$ gilt. Eine Schrittweite σ heißt ***effizient****, falls*

$$J(u + \sigma d) \leq J(u) - c \left(\frac{J'(u)d}{\|d\|} \right)^2$$

mit einer von u und d unabhängigen Konstanten $c = c_1 \, c_2 > 0$ gilt.

Erfüllt eine Schrittweite das Prinzip des hinreichenden Abstiegs, dann ist sie effizient.

Wir betrachten die Situation von Beispiel (6.3). Die Wahl der Schrittweitenfolge in diesem Beispiel erfüllt nicht das Prinzip des hinreichenden Abstiegs, da nach Ungleichung (6.6) die Bedingung

$$\sigma_k \geq 2c_2 u_k$$

mit einer von u unabhängigen Konstanten c_2 gelten müsste.

Eine naheliegende Schrittweitenstrategie besteht darin, die Schrittweite σ durch Lösung eines eindimensionalen Optimierungsproblems

$$\min_{\sigma \geq 0} \phi(\sigma) \; = \; J(u + \sigma d)$$

zu berechnen. Man bezeichnet die Lösung dieses Problems als *exakte Schrittweite*. Allerdings ist zu beachten, dass nur unter zusätzlichen Voraussetzungen(z.B. Konvexität von J) sichergestellt werden kann, dass σ globale Lösung des Problems ist. In der Praxis geht man deshalb sinnvollerweise zu nicht exakten Schrittweitenverfahren, die aber dennoch effiziente Schrittweiten liefern.

Im Folgenden werden wir Verfahren zur Berechnung solcher Schrittweiten betrachten, man spricht von *line-search* Verfahren. Da wir nur an der Minimierung von Zielfunktionalen interessiert sind, aber nicht an der exakten Approximation der Lösung eines Gradientenflusses, basiert eine Schrittweitensteuerung lediglich auf dem Ziel einen hinreichenden Abstieg des Zielfunktionals zu finden.

Ein klassischer Ansatz dafür sind die sogenannten *Armijo-Goldstein Regeln*. Die Armijo-Regel ist ein wichtiges Element für alternierende Schrittweitenverfahren (wie Armijo-Goldstein), die das Prinzip des hinreichenden Abstiegs erfüllen.

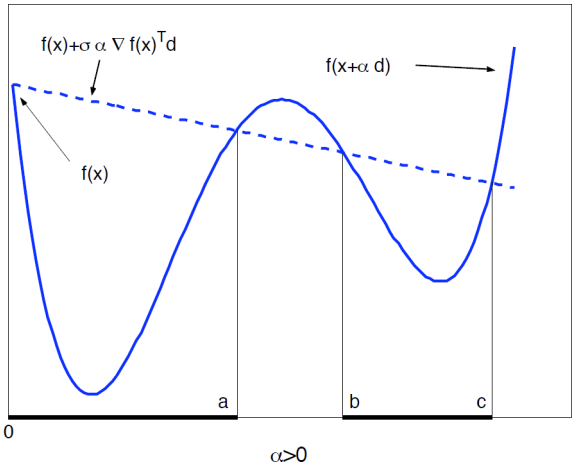


Figure 6.1: Illustration der Schrittweitensuche von Armijo

Definition 6.3.2 (Armijo Bedingung)**.** *Sei J ein Zielfunktional, u_k eine aktuelle Iterierte, d_k eine Suchrichtung und $c_1 \in (0,1)$ eine kleine Konstante. Dann führt eine geeignete Wahl der Schrittweite σ zu einem hinreichenden Abstieg des Zielfunktionals, d.h.*

$$J(u_k + \sigma d_k) \leq J(u_k) + c_1 \sigma \, J'(u_k) d_k \ . \tag{6.7}$$

Die praktische Umsetzung bei der Schrittweitensuche ist in der Regel folgende: Man startet mit einer Startschrittweite $\sigma_0 > 0$ und überprüft nach gewissen Bedingungen (hier Armijo), ob die Schrittweite zu einem hinreichenden Abstieg im Zielfunktional führt. Ist dies nicht der Fall, so wird die Schrittweite mit einem Parameter $\tau \in (0,1)$ solange verkleinert

$$\sigma_{k+1} = \tau \sigma_k \ ,$$

bis ein hinreichender Abstieg erreicht wird. Man spricht dabei von *backtracking line-search*. Allerdings allein die Armijo-Bedingung mittels *backtracking* zu verwenden, ist nicht ausreichend um einen hinreichenden Fortschritt der Minimierung zu garantieren, da die Bedingung unter Umständen schon für hinreichend kleine Werte der σ_k erfüllt sein kann.

Stattdessen testet man zusätzlich zur Armijo Bedingung noch auf eine darauf folgende Bedingung:

$$J(u_k + \sigma d_k) \geq J(u_k) + c_2 \sigma \, J'(u_k) d_k \ , \tag{6.8}$$

mit $0 < c_1 < c_2 < 1$. Die Schrittweitenstrategie (6.7) zusammen mit (6.8) bezeichnet man als *Armijo-Goldstein Regeln*. Die beiden Regeln sind gleichbedeutend mit dem Vergleich von *effektivem Abstieg*

$$D_{eff}(\sigma) := J(u_k + \sigma d_k) - J(u_k)$$

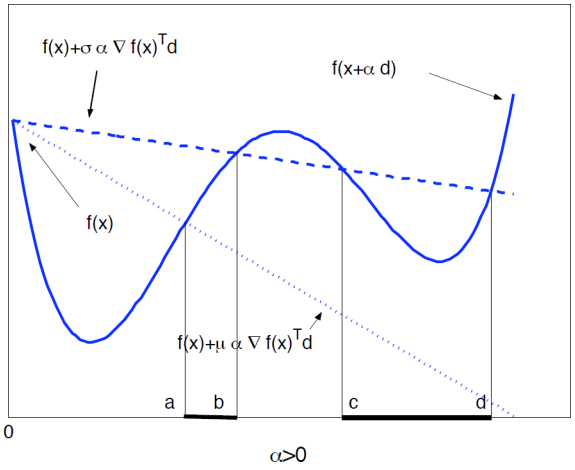


Figure 6.2: Illustration der Schrittweitensuche von Armijo-Goldstein

und *erwartetem Abstieg*

$$D_{exp}(\sigma) := \sigma J'(u_k)d_k \ .$$

Für hinreichend kleines σ stehen sie über die Taylorformel in Beziehung zueinander

$$D_{eff}(\sigma) = D_{exp}(\sigma) + o(\sigma) \ .$$

Man testet also, ob

$$c_2 D_{exp}(\sigma) \leq D_{eff}(\sigma) \leq c_1 D_{exp}(\sigma) \tag{6.9}$$

mit Konstanten $0 < c_1 < c_2 < 1$ erfüllt ist. Letztendlich akzeptieren wir eine Schrittweite falls (6.9) oder äquivalent (6.7) zusammen mit (6.8) erfüllt sind. Eine typische Wahl für die Konstanten sind

$$c_1 \approx 0.1, \quad c_2 \approx 0.9 \ .$$

Unter Verwendung der Armijo-Goldstein Regeln, kann man globale Konvergenz des Gradientenverfahrens beweisen, d.h. Konvergenz zu einem stationären Punkt für jeden beliebigen Startwert.

Theorem 6.3.3. *Sei J zweimal stetig Frechet-differenzierbar und schwach unterhalbstetig auf einem Hilbertraum $\mathcal{U}$. Weiterhin seien die Mengen*

$$\{u \in \mathcal{U} | J(u) \leq M\}$$

für jedes $M \in \mathbb{R}$ beschränkt in $\mathcal{U}$ und leer für M hinreichend klein. Dann hat die Folge (u_k) aus dem Gradientenverfahren mit Armijo-Goldstein Schrittweitensuche eine schwach konvergente Teilfolge, deren Grenzwert ein stationärer Punkt ist.

Proof. Da das Gradientenverfahren ein Abstiegsverfahren ist, gilt

$$J(u_k) \leq J(u_0)$$

für alle $k \geq 0$. Das bedeutet, dass die Folge (u_k) beschränkt ist und daher eine schwach konvergente Teilfolge (u_{k_l}) mit Grenzwert $\bar{u}$ existiert. Mit der ersten Bedingung aus der Schrittweitensuche nach Armijo-Goldstein erhalten wir damit folgende Abschätzung

$$\begin{aligned}
\sum_{k=0}^{N} \|u_{k+1} - u_k\|^2 &= -\sum_{k=0}^{N} \sigma_k \, J'(u_k)(u_{k+1} - u_k) \\
&\leq \frac{1}{c_1} \sum_{k=0}^{N} (J(u_k) - J(u_{k+1})) \\
&= \frac{1}{c_1} (J(u_0) - J(u_{N+1})) \\
&\leq \frac{1}{c_1} \Big(J(u_0) - \inf_u J(u) \Big) =: p \, .
\end{aligned}$$

Da p unabhängig von N ist, erhalten wir für $N \to \infty$

$$\sum_{l=0}^{\infty} \|u_{k_l+1} - u_{k_l}\|^2 \leq \sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2 \leq p \, .$$

Daher existiert eine Teilfolge von (u_{k_l}), ohne Beschränkung der Allgemeinheit sei diese (u_{k_l}) selbst, mit

$$\|\sigma_{k_l} \, J'(u_{k_l})\| = \|u_{k_l+1} - u_{k_l}\| \to 0 \, .$$

Da J zweimal unterhalbstetig Frechet-differenzierbar ist, existiert eine Konstante $C < 0$ mit

$$J''(u_{k_l})(v, v) \leq C \, \|v\|^2 \, , \quad \forall v \in \mathcal{U} \, .$$

Damit impliziert die zweite Bedingung in Armijo-Goldstein

$$\begin{aligned}
c_2 \, \sigma_{k_l} J'(u_{k_l}) &\leq J(u_{k_l}) - J(u_{k_l}) \\
&\quad J'(u_{k_l})(u_{k_l+1} - u_{k_l}) + \frac{C}{2} \|u_{k_l+1} - u_{k_l}\|^2 \, .
\end{aligned}$$

Setzt man $u_{k_l+1} - u_{k_l} = -\sigma_{k_l} J'(u_{k_l})$ ein, so erhalten wir

$$(1 - c_2)\sigma_{k_l} \|J'(u_{k_l})\|^2 \leq \frac{C}{2} \sigma_{k_l}^2 \|J'(u_{k_l})\|^2 \, .$$

Damit gilt entweder $J'(u_{k_l}) = 0$ oder

$$\sigma_{k_l} \geq \frac{2(1 - \alpha)}{c} \, .$$

Falls $J'(u_k) = 0$ gilt, so hat der Algorithmus einen stationären Punkt erreicht und stoppt, d.h. $u_j = u_{k_l}$ für alle $j \geq k_l$, und die Konvergenz ist trivial. Im zweiten Fall ist σ_{k_l} gleichmäßig nach unten von Null weg beschränkt und deshalb gilt $\|J'(u_{k_l})\| \to 0$. Dies impliziert, dass $J'(\bar{u}) = 0$, d.h. der Grenzwert $\bar{u}$ ist ein stationärer Punkt. □

Betrachtet man für die Schrittweitensuche zusätzlich neben der Armijo-Schrittweite noch eine Krümmungseigenschaft, die garantiert, dass die Steigung von J hinreichend stark reduziert wird, so spricht man von den Wolfe-Bedingungen. Es wird auch Verfahren von Powell genannt, da er es bei der Untersuchung der globalen Konvergenz von Quasi-Newton-Verfahren benutzt hat, das wir in einem späteren Abschnitt noch kennenlernen werden.

Die vorgestellten Schrittweitenverfahren sind selbstverständlich nicht auf das Gradientenverfahren beschränkt, sondern sie können grundsätzlich bei Abstiegsverfahren eingesetzt werden.

6.4 Landweber und Iterative Regularisierung

In diesem Abschnitt betrachten wir Gradientenverfahren für eine spezielle Klasse von Variationsproblemen und ihre Interpretation im Sinne von Regularisierung durch Iteration. Die sogenannte *Landweber Iteration* kann man betrachten als ein einfaches Gradientenverfahren für L^2 Daten-Fitting Funktionale der Form

$$J(u) := \frac{1}{2} \|Ku - f\| \tag{6.10}$$

mit konstanter Schrittweite sowie linearem und kompaktem Operator K. Die Frechet Ableitung des Funktionals J ist gegeben durch

$$J'(u) = K^*(Ku - f) \ .$$

Die Landweber Iteration (L. Landweber, 1951) ist dann einfach gegeben durch eine Zeitdiskretisierung des Gradientenflusses mit fixer Schrittweite, d.h. mit Startwert $u_0 = 0$ erhalten wir die Iteration

$$u_{k+1} \ = \ u_k - \sigma K^*(Ku_k - f) \ = \ (I - \sigma K^* K)u_k + \sigma K^* f$$

wobei $0 < \sigma < \frac{2}{\|K\|^2}$. Da das Gradientenverfahren ein Abstiegsverfahren ist, erhalten wir für hinreichend kleines σ einen Abstieg des *least-squares* Funktionals.

Bei exakten, rauschfreien Daten kann man zeigen, dass bei Startwert $u_0 = 0$ und unter der Annahme, dass K invertierbar ist, die Iterierten u_k gegen $u^* = K^{-1}f$ konvergieren. Genauer setzt man eine Singulärwertzerlegung an, nutzt aus, dass σ über

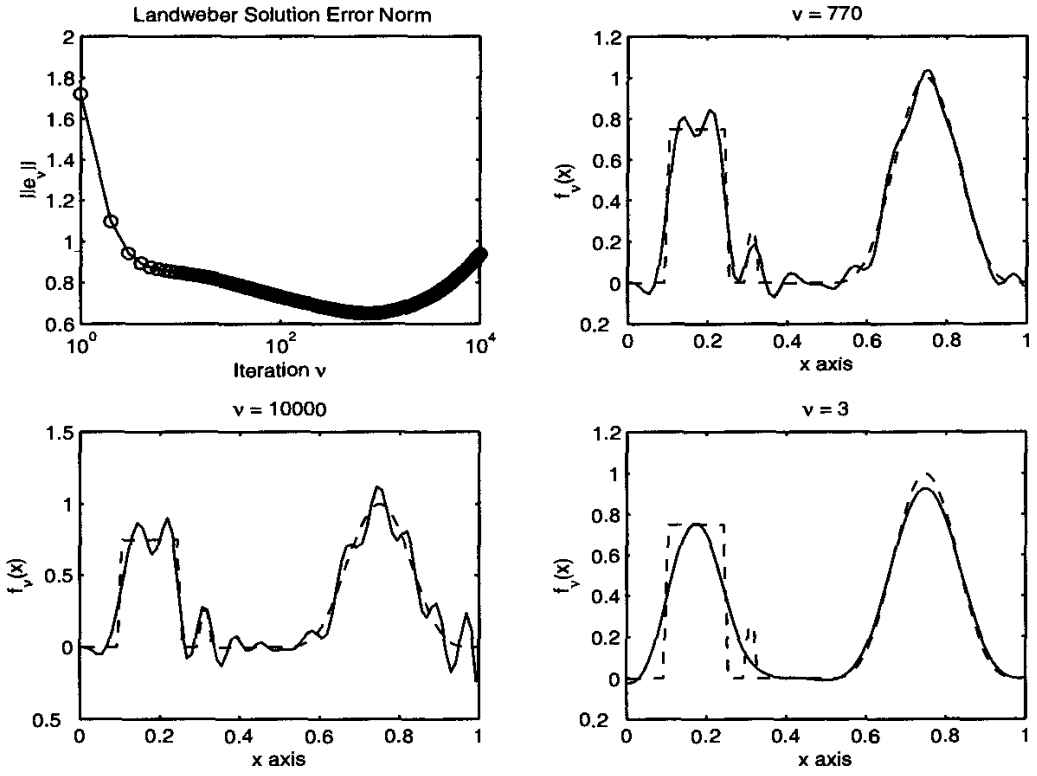


Figure 6.3: aus C. R. Vogel, Computational Methods for Inverse Problems

die Norm von K beschränkt ist und interpretiert $\frac{1}{k}$ als einen Regularisierungsparameter. Neben einem Abstieg des Zielfunktionals kann man folgende Konvergenzgesschwindigkeit bzw. Fehlerabschätzung zeigen

$$\left\|u_k - u^\dagger\right\| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right) \ .$$

Diese Betrachtungsweise ist selbstverständlich nicht sinnvoll für gegebene Daten f^δ mit Rauschen in der Größenordnung δ. In Abbildung 6.4 kann durch einen Plot der Norm der Residuen gegen die Anzahl der Iterationen erkennen, dass diese die Rolle eines Regularisierungsparameters $\frac{1}{\alpha}$ spielt. Eine zu kleine Iterationenanzahl liefert eine überall überglättete approximative Lösung, während bei zu vielen Iterationen die Rekonstruktionen starke Oszillationen aufweisen. Die Frage, die sich ergibt, ist der geeignete Abbruch dieses Verfahrens.

Für iterative Methoden wie die Landweber Iteration, die auf L^2 Daten-Fitting Funktionalen beruhen, ist es einfach a-posteriori Abbruchregeln wie das *Diskrepanzprinzip* (Morozov),

$$k_*(\delta, f^\delta) := \inf\left\{k \in \mathbb{N} \mid \left\|Ku_k^\delta - f^\delta\right\| < \eta\,\delta\right\} \ ,$$

mit $\eta \geq \frac{2}{2-\sigma\|K\|}$ zu verwenden. Grob gesprochen bedeutet das, man stoppt die Iteration, wenn der Fehler das erste mal eine Größenordnung unterhalb des Rausch-Levels

erreicht hat. Für die praktische Umsetzung dieses Abbruchkriteriums, beobachtet man die Residuen $Ku_k^\delta - f^\delta$, die man sowieso bei der Landweber-Iteration berechnet, und vergleicht ihre Norm mit dem Rauschlevel.

Man kann zeigen, dass das Diskrepanzprinzip eine konvergente Regularisierungmethode darstellt. Insbesondere kann man für den Fall, dass die Lösung $u_\dagger$ eine Quellbedingung *source condition*, $u_\dagger = K^*p$ mit dualer Variablen p erfüllt, zeigen, dass

$$\left\| u_{k_*}^\delta - u_\dagger \right\| = \mathcal{O}\left(\sqrt{\delta}\right) \ .$$

Wenn man statt des inversen Ausgangsproblem $Ku = f$ zu einer Darstellung mit Pseudoinversen übergeht, so erhält man das sogenannte *ART (Algebraic reconstruction technique)* Vrefahren, das man auch als eine sogenannte *Kaczmarz* Methode interpretieren kann. Im Spezialfall ohne Dämpfung bzw. Vorkonditionierung fällt ART mit Landweber zusammen.

6.5 Projizierte Gradientenverfahren

Bislang habe wir uns mit Gradienten-artigen Verfahren für Variationsmethoden ohne Nebenbedingung beschäftigt. In diesem Abschnitt wollen wir unsere Betrachtungsweise nun erweitern auf numerische Optimierungsverfahren für Variationsmethoden mit einfachen Nebenbedingung wie z.B. Positivitäts-Nebenbedingungen. Wir betrachten Probleme der Form

$$\begin{aligned} &J(u) \to \min_u \\ &\text{unter} \quad u \in C \ . \end{aligned}$$

Es sei C eine abgeschlossene konvexe Menge und mit Π_C bezeichnen wir eine Projektion in die konvexe Menge.

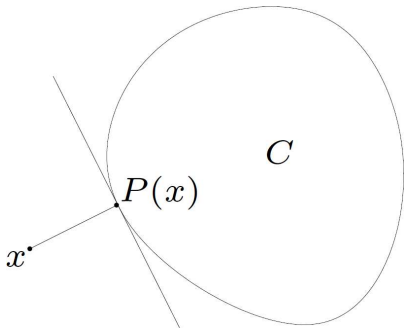
Beispiele:
Projektion auf l^2-Normkörper (l^2-Ball):

Für einen konvexen l^2-Ball für

$$C := \{u | \, \|u\|_2 \leq 1\}$$

kann man eine zugehörige Projektion leicht realisieren

$$\Pi_C(u) = \begin{cases} \frac{1}{\|u\|_2} \, u & \text{falls } \|u\|_2 \geq 1 \\ u & \text{sonst} \end{cases}$$



Eine Projektion auf einen l^1-Normkörper (l^1-Ball) lässt sich analog realisieren.

Orthogonale Projektion:

$$\Pi_C(u) := \arg\min_{z \in C} \|u - z\|_2^2$$

$P_K(u)$ existiert und existiert für alle u. (–> Verweis auf Splitting Verfahren)

Projiziertes Gradientenverfahren (J Frechet differenzierbar):
Mit Startwert $x_0 \in K$

$$u_{k+1} = \Pi_C \left(u_k - \sigma_k \, J'(u_k) \right), \quad k = 0, 1, \cdots \ .$$

Man erhält ähnliche Konvergenzaussagen wir bei dem normalen Gradientverfahren ohne Projektion. Selbstverständlich kann man auch das projizierte Gradientenverfahren mit einer effizienten Schrittweite kombinieren.

Beim *projiziertes Subgradientenverfahren* ersetzt man lediglich die Frechet Ableitung durch allgemeinere Subgradienten und erhält als Update

$$u_{k+1} = \Pi_C \left(u_k - \sigma_k \, p_k \right), \quad k = 0, 1, \cdots$$

mit $p_k \in \partial J(u_k)$.

Mit einem zusätzlichen stabilisierenden Teilschritt erhält man das projizierte Gradientenverfahren von *Nesterov*
Mit Startwert $u_0 \in K$ und $v_0 = u_0$ ist für $k = 0, 1, \cdots$ die Zwei-Schritt Iteration gegeben durch

$$\begin{aligned} u_{k+1} &= \Pi_C \left(v_k - \sigma_k \, p_k \right), \\ v_{k+1} &= u_{k+1} + \frac{k-1}{k+2} \left(u_{k+1} - u_k \right) \ . \end{aligned}$$

Schnellere Konvergenz als einfaches Gradientenverfahren
Man beachte aber, dass die Hilfsvariablen v in den Teilschritten nicht notwendigerweise zulässig sein müssen.

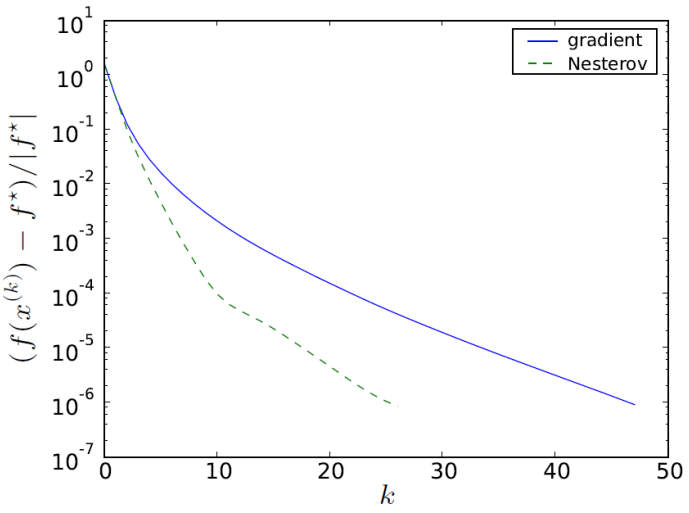


Figure 6.4: Gradientenverfahren vs. Nesterov: n = 3000, feste Schrittweite $\sigma = \frac{1}{\lambda_{max}}$, wobei λ_{max} maximaler Eigenwert des Systems ist.

6.6 Konjugiertes Gradientenverfahren und Konvergenzraten

Zur Motivation des konjugierten Gradientenverfahrens wollen wir zunächst nochmal das einfache Gradientenverfahren und quadratische Funktionale $J : \mathbb{R}^n \to \mathbb{R}$ der Form

$$J(u) = \frac{1}{2} \langle Au, u \rangle + \langle b, u \rangle + c \tag{6.11}$$

mit $c \in \mathbb{R}$, $b \in \mathbb{R}^n$ und symmetrischer Matrix A, betrachten. Man beachte, dass $A = Hess(J(u))$ gilt. Eine quadratisches Funktional nennt man positiv, falls die Matrix A symmetrisch positiv definit ist. In diesem speziellen Fall ist J strikt konvex und besitzt ein eindeutiges Minimum. Die exakte Schrittweite ist in diesem Fall gegeben durch

$$\sigma_k = \frac{\|g_k\|^2}{\langle Ag_k, g_k \rangle} \ ,$$

wobei $g_k := J'(u_k)$ (siehe Übungsblatt 10, Aufgabe 3). Man erhält damit für das Gradientenabstiegsverfahren folgende Konvergenzaussage.

Theorem 6.6.1. *Angenommen J sei ein positives, quadratisches Funktional der Form (6.11). Dann konvergiert das Gradientenabstiegsverfahren mit exakter Schrittweite für einen beliebigen Startwert u_0 mit **linearer Rate** gegen das Optimum u^*, genauer*

$$\|u_k - u^*\|_A \leq \left(\frac{cond(A) - 1}{cond(A) + 1} \right)^k \|u_0 - u^*\|_A \ , \tag{6.12}$$

wobei $\|u\|_A := \sqrt{\langle u,u\rangle_A} = \sqrt{\langle Au,u\rangle}$ *die durch A induzierte Norm (energy norm) bezeichnet und die Kondition durch das Verhältnis zwischen größtem und kleinsten Eigenwert von A gegeben ist.*

Einen ähnlichen Satz kann man auch für den allgemeineren (nichtquadratischen) Fall beweisen, wobei man die Matrix A durch die Hessematrix von J an der Stelle u^* ersetzt. Aus diesem Satz können wir ableiten, dass wenn die Hessematrix eines Funktionals schlecht-konditioniert ist, so kann die Konvergenz des einfachen Gradientenabstiegsverfahrens sehr langsam sein. Ein schneller konvergierendes Verfahren ist das *konjugierte Gradientenverfahren* (CG). Wir starten mit einer Version von CG für positive quadratische Funktionale der Form (6.11) mit symmetrisch positiv definitem A, oder äquivalent zur Lösung von $Au = -b$:

Algorithm 6.6.2 (CG Verfahren, quadratisches Optimierungsproblem)**.**
$u_0 = Startwert$
$d_0 = -J'(u_0)$ % *Initiale Suchrichtung*
$\delta_0 = \|g_0\|^2$
Für alle CG Iterationen:

$$\begin{cases} \sigma_k & = & -\dfrac{\langle J'(u_k), d_k\rangle}{\langle d_k, A\, d_k\rangle} & \% \; exakteSchrittweite \\ u_{k+1} & = & u_k + \sigma_k\, d_k & \% \; Update\; Unbekannte \\ J'(u_{k+1}) & = & J'(u_k) + \sigma_k\, A\, d_k & \% \; Update\; Gradient \\ d_{k+1} & = & -J'(u_{k+1}) + \dfrac{\|J'(u_{k+1})\|^2}{\|J'(u_k)\|^2}\, d_k & \% \; Update\; Suchrichtung \end{cases}$$

Das CG Verfahren benutzt hier die exakten Schrittweiten und orthogonalisiert die Richtungen $-J'(u_k)$.

Das Verfahren lässt sich besonders gut zur Lösung von sehr großen, stark strukturierten linearen Systemen $Au = -b$ einsetzen, bei denen direkte Lösungsverfahren wie die Cholesky Zerlegung nicht mehr praktikabel sind. Im Vergleich zum Gradientenverfahren, das bei schlecht-konditionierten Systemen eine schlechte Konvergenzrate aufweist, liefert das iterative konjugierte Gradientenverfahren (zusammen mit einer Konvergenzbeschleunigung genannt Vorkonditionierung) einen sehr effizienten Weg zur Lösung solcher System.

Das CG Verfahren kann mit einer kleinen Anpassung auch für nichtlineare Optimierungsprobleme eingesetzt werden (Fletcher-Reeves). Anstelle der exakten Schrittweiten wird man praktisch zu Schrittweitenverfahren, wie z.B. Armijo-Goldstein, übergehen.

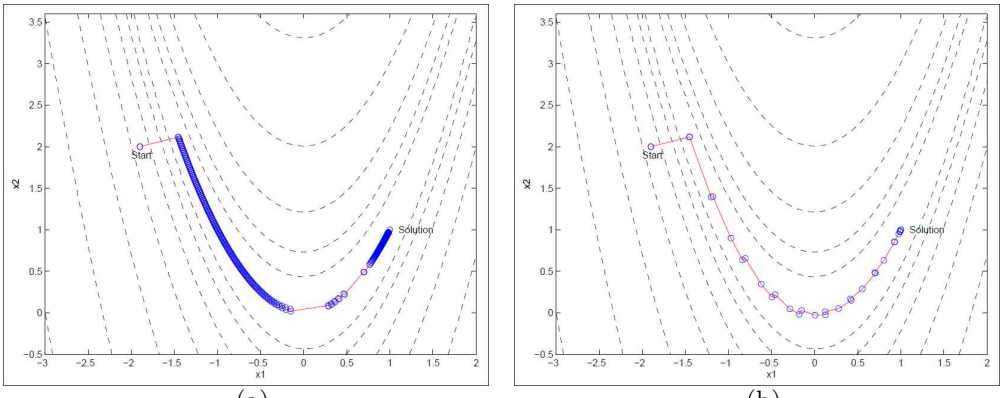
Später im Abschnitt zum Newton Verfahren werden wir sehen, dass man das konjugierte Gradientenverfahren effizient für enstehende Teilprobleme einsetzen kann. Gerade in diesem Zusammenhang werden Techniken zur Vorkonditionierung eine wichtige Rolle spielen.

Im Fall des quadratischen Optimierungsproblems von oben mit einer symmetrisch positiv definiten Matrix A kann man zeigen, dass das CG Verfahren eine *exakte Lösung* des Systems $Au = -b$ mit *maximal* n *Iterationen* berechnet, wobei n die Dimension der Matrix A charakterisiert.

Zudem kann man mit einer geeigneten Wahl von Chebyshev Polynomen auch hier eine Konvergenzrate für das Verfahren beweisen,

$$\|u_k - u^*\|_A \leq 2 \left(\frac{\sqrt{cond(A)} - 1}{\sqrt{cond(A)} + 1} \right)^k \|u_0 - u^*\|_A \ .$$

Im Vergleich zur Konvergenzrate (6.12) beim einfachen Gradientenverfahren erhält man mit CG eine deutlich verbesserte lineare Konvergenz. Insbesondere ist die Schranke wesentlich kleiner als beim Gradientenverfahren, falls die Kondition von A schlecht ist. Man erhält schnelle Konvergenz, falls die Eigenwerte von A weg von der Null gruppiert liegen.



(a) (b)

Figure 6.5: (a) Gradientenverfahren (626 Iterationen) vs. (b) CG (47 Iterationen) anhand der Rosenbrock Funktion, Tol = 10^{-4}

6.7 EM Algorithmus

6.8 Newton Verfahren und Varianten

In den vorherigen Kapiteln haben wir grundlegende Gradienten-basierte Verfahren kennen gelernt, die man zur numerischen Optimierung von (unbeschränkten) Variationsmethoden einsetzen kann. Typischerweise zeigen diese Verfahren ein Konvergenzverhalten erster Ordnung. Möchte man die Konvergenzordnung deutlich verbessern, so macht es Sinn zu Newton-artigen Verfahren überzugehen. Das ist insbesondere bei Anwendungen sinnvoll, in denen eine relativ hohe Genauigkeit einer Lösung essentiell ist. In diesem Kapitel werden wir deshalb grundlegende Konzepte Newton-artiger Verfahren für unbeschränkte nichtlineare Variationsmethoden der Form

$$\min_u J(u) \tag{6.13}$$

mit J zweimal stetig Frechet differenzierbar behandeln, Varianten des Newton-Verfahrens für unterschiedliche Anwendungen kennen lernen und auf ihr Konvergenzverhalten eingehen.

6.8.1 (Exaktes) Newton Verfahren

In der numerischen Analysis ist das Newton-Verfahren (oder auch Newton-Raphson Verfahren) ein Verfahren zur Bestimmung von Nullstellen einer Gleichung in einer oder mehr Dimensionen. Die grundlegende Idee des Newton Verfahrens für eine nichtlineare Gleichung wie z.B.

$$J'(u) = 0 \ , \tag{6.14}$$

der notwendigen Optimalitätsbedingung unseres Ausgangsproblems (6.13), ist eine *lokale Linearisierung* an der Stelle u_k um

$$u_{k+1} = u_k + d_k \tag{6.15}$$

zu berechnen, d.h.

$$J'(u_k) + J''(u_k)\, d_k \ = \ 0 \ \Leftrightarrow \ J''(u_k)\, d_k \ = \ -J'(u_k) \tag{6.16}$$

wobei J' die Jacobi Matrix und J'' die Hesse Matrix von J darstellt, und die Suchrichtung

$$d_k = -J''(u_k)^{-1} J'(u_k)$$

als sogenannte Newton-Richtung bezeichnet wird. Für eine Visualisierung des (exakten) Newton Verfahrens für die Gleichung (6.14) betrachten wir Abbildung 6.6. Eine zweite Interpretation des Newton Verfahrens aus Sicht der numerischen Optimierung

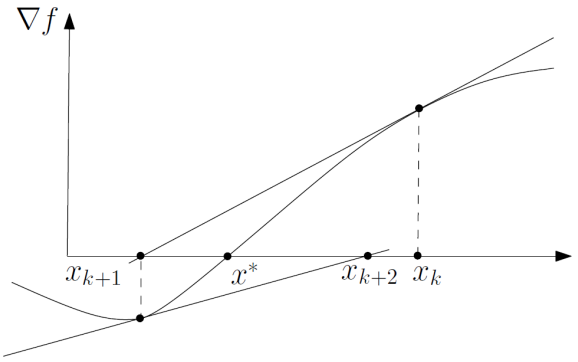


Figure 6.6: Erste Interpretation: Newton-Verfahren als lokale Linearisierung

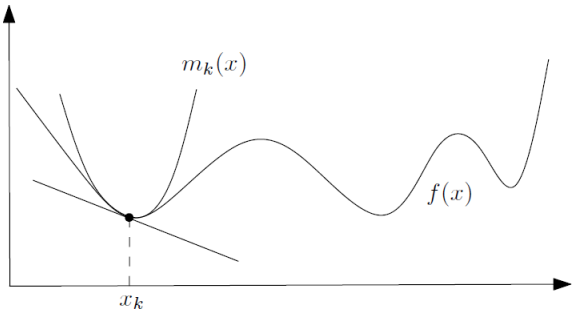


Figure 6.7: Zweite Interpretation: Newton-Verfahrens durch quadratisches Modell

erhält man durch ein quadratisches Modell des Zielfunktionals bzgl. der Suchrichtung d (ein quadratisches Modell kann verhältnismäßig leicht gelöst werden), d.h. einer Taylor Approximation zweiter Ordnung von J um u_k,

$$\begin{aligned} M_k(u_k + d) &:= J(u_k) + J'(u_k)d + \frac{1}{2}dJ''(u_k)d \\ &\approx J(u_k + d) \ . \end{aligned} \tag{6.17}$$

Eine Visualisierung dieser zweiten Interpretation ist in Abbildung 6.7 zu sehen. Falls die Hessematrix $J''(u_k)$ positiv definit ist, dann ist der Minimierer d_k von M_k eine eindeutige Lösung von $M_k'(u_k + d) = 0$, d.h.

$$0 = J'(u_k) + J''(u_k)d$$

und damit erhält man wieder das (exakte) Newton-Verfahren

$$\begin{aligned} d_k &= -J''(u_k)^{-1}J'(u_k) \\ \Leftrightarrow u_{k+1} &= u_k - J''(u_k)^{-1}J'(u_k) \ . \end{aligned}$$

Selbstverständlich wird dabei nicht explizit die Inverse der Hessematrix berechnet, sondern stattdessen zur Berechnung der Suchrichtung d_k in (6.16) ein lineares Gleichungssystem gelöst und anschließend ein Update gemäß (6.15) durchgeführt. Verwendet man zusätzlich eine effiziente Schrittweite σ_k, so spricht man vom gedämpften

Newton-Verfahren. Startet man nicht sehr nahe an einem Minimum, dann ist eine effiziente Schrittweite in der Regel < 1, d.h., die Schrittweite des Newton-Verfahrens wird gedämpft.

Algorithm 6.8.1 (Gedämpftes Newton Verfahren)**.**
Wähle Startwert u_0 und setze $k := 0$.
Ist $J'(u_k) = 0$, % Stopp.
Berechne d_k aus

$$J''(u_k)d_k = -J'(u_k) \ ,$$

und mit effizienter Schrittweite σ_k setze

$$u_{k+1} = u_k + \sigma_k d_k.$$

Setze $k := k + 1$.

6.8.2 Varianten des Newton-Verfahrens

In diesem Abschnitt diskutieren wir unterschiedliche Varianten des Newton bzw. Variable-Metrik Verfahrens. Wir wollen dabei zwei Fragestellungen mit berücksichtigen: *Wie schnell ist die lokale Konvergenz der Verfahren? Kann man eine Konvergenz für jeden Startwert garantieren?* (globale Konvergenz)

Um das Konvergenzverhalten unterschiedlicher Newton-artiger Verfahren vergleichen zu können, wollen wir unterschiedliche Konvergenzraten einführen.

Definition 6.8.2 (Konvergenzraten)**.** *Angenommen $u_k \to u^*$. Dann konvergiert die Folge u_k:*

(i) Q-linear ("Quotient"-linear) $\Leftrightarrow$

$$\|u_{k+1} - u^*\| \leq C \|u_k - u^*\| \ \text{, mit } C < 1$$

für alle $k \geq k_0$.

$$\Leftrightarrow \ \limsup_{k\to\infty} \frac{\|u_{k+1} - u^*\|}{\|u_k - u^*\|} < 1$$

(ii) Q-superlinear $\Leftrightarrow$

$$\|u_{k+1} - u^*\| \leq C \|u_k - u^*\| \ \text{, mit } C \to 0$$

$$\Leftrightarrow \ \limsup_{k\to\infty} \frac{\|u_{k+1} - u^*\|}{\|u_k - u^*\|} = 0 \ .$$

(iii) Q-quadratisch $\Leftrightarrow$

$$\|u_{k+1} - u^*\| \leq C \|u_k - u^*\|^2 \ , \text{mit } C < \infty$$

$$\Leftrightarrow \ \limsup_{k\to\infty} \frac{\|u_{k+1} - u^*\|}{\|u_k - u^*\|^2} < \infty \ .$$

Beispiele: ...

Jede Iteration der Form

$$u_{k+1} = u_k - A_k^{-1} J'(u_k) \tag{6.18}$$

mit A_k invertierbar, bezeichnet man als *Newton-artige Iteration* in der Optimierung. Für $A_k = J''(u_k)$ erhält man wieder das (exakte) Newton Verfahren, üblicherweise wählt man in der Praxis aber $A_k \approx J''(u_k)$. Man spricht bei Verfahren der Form (6.18) auch von Variable-Metrik Verfahren. Ist $A = J''(u)$ positiv definit, dann wird durch die Norm bzgl. A

$$\|u\|_A = \sqrt{\langle u, u\rangle_A} = \sqrt{\langle u, Au\rangle}$$

eine Metrik definiert. Zum Beispiel erhält man bei einer sehr einfachen Metrik bzgl. $A_k = I$, als Spezialfall wieder das bekannte Gradientenabstiegsverfahren. Der Vorteil des gedämpften Newton-Verfahrens gegenüber dem Gradientenverfahren ist, dass mit $J''(u_k)$ auch Informationen über die Krümmung von J in $u = u_k$ benutzt werden. Beim gedämpften Newton-Verfahren wird die Metrik zur Bestimmung des steilsten Abstiegs "variabel" an die Krümmung von J angepasst.

Allgemein ist in jeder Iteration ein quadratisches Teilproblem zu lösen, um eine neue Suchrichtung d_k zu bestimmen:

$$d_k = \arg\min_d M_k(u_k + d) \ .$$

mit

$$M_k(u_k + d) = J(u_k) + J'(u_k)d + \frac{1}{2} dA_k(u_k)d \ ,$$

analog zu (6.17). Die Optimalitätsbedingung dieses Modells führt zum Update einer Suchrichtung in einer Newton-artigen Iteration:

$$\begin{aligned} 0 = M_k'(u_k + d_k) &= A_k d_k + J'(u_k) \\ \Longleftrightarrow \ d_k &= -A_k^{-1} J'(u_k) \ . \end{aligned}$$

Man beachte, dass d_k nur dann ein Minimierer von $M_k(u_k + d_k)$ ist, falls $A_k \succ 0$ (positiv definit). Für das (exakte) Newton Verfahren muss das nicht notwendigerweise der Fall sein, falls u_k weit weg ist von einer Lösung u^*.

Lemma 6.8.1 (Abstiegsrichtung)**.** *Falls $A_k \succ 0$, dann ist $d_k = -A_k^{-1} J'(u_k)$ eine Abstiegsrichtung.*

Proof.

$$J'(u_k)d_k = \underbrace{-J'(u_k)\underbrace{A_k^{-1}}_{\succ 0} J'(u_k)}_{>0} < 0 \ .$$

□

Definition 6.8.3 (Varianten des Newton Verfahrens)**.** *Varianten des Newton Verfahrens, die häufig in der Literatur verwendet werden, sind u.a. folgende:*

(a) *Quasi-Newton Verfahren*
Das Newton-Verfahren konvergiert zwar lokal quadratisch, jedoch liegt ein Problem bei der Anwendung des (exakten) Newton-Verfahrens darin, dass man die Hessematrix des Funktionals benötigt. Gerade bei hochdimensionalen Problemen in der Bildverarbeitung kann es in der Praxis schwierig sein die vollständige Hessematrix eines zu minimierenden Funktionals J zu berechnen (z.B. Speicherprobleme, Probleme mit der Rechenzeit).

Für solche Situationen wurden sogenannte *Quasi-Newton Verfahren* entwickelt, die eine Approximation der Hessematrix verwenden. Man approximiert dabei die Hessematrix A_{k+1} rekursiv mittels einer alten Approximation der Hessematrix A_k und Auswertungen der ersten Ableitung, $J'(u_{k+1})$ und $J'(u_k)$. Betrachtet man eine Taylor-Entwicklung von J'

$$J'(u_k) = J'(u_{k+1}) + J''(u_{k+1})(u_k - u_{k+1}) + o(\|u_k - u_{k+1}\|) \ ,$$

so erhält man mit $d_k = u_{k+1} - u_k$ die folgende wichtige Gleichung

$$A_{k+1}(u_{k+1} - u_k) = J'(u_{k+1}) - J'(u_k) \quad \text{(Sekanten-Bedingung)} \ . \tag{6.19}$$

Man bezeichnet diese Gleichung auch als Quasi-Newton-Gleichung. Der Preis für die Approximation der Hessematrix ist ein Verlust an Konvergenzgeschwindigkeit. Man kann beweisen, dass Quasi-Newton-Verfahren lokal superlinear konvergieren.

Das wichtigste Verfahren der Quasi-Newton-Klasse ist das sogenannte BFGS-Verfahren, das auf der folgenden Updateformel basiert,

$$A_{k+1} = A_k - \frac{A_k d d^T A_k}{d^T B_k d} + \frac{y y^T}{d^T y}$$

mit d und y definiert als

$$\begin{aligned} d &= u_{k+1} - u_k \ , \\ y &= J'(u_{k+1}) - J'(u_k) \ . \end{aligned}$$

Die Updateformel wurde von **B**royden, **F**letcher, **G**oldfarb und **S**hanno unabhängig voneinander gefunden. Man kann leicht nachrechnen, dass $A_{k+1}s = y$ gilt, also damit die Quasi-Newton-Gleichung in (6.19) erfüllt ist. Das BFGS-Verfahren ist eine sehr erfolgreiche Methode und man kann zeigen, dass die Folge der A_k gegen die Hessematrix J'' an der Stelle u^* konvergiert.

(b) *Gauss-Newton und Levenberg-Marquardt*
Einen schönen Zusammenhang zwischen Newton-Verfahren und inversen Problemen (z.B. in der Bildgebung) liefert die Klasse der Gauss-Newton und Levenberg-Marquardt Verfahren. Insbesondere für *nichtlineare* inverse Probleme der Form

$$F(u) - y = 0$$

mit einem *nichtlinearen* Operator F und gegebenen Daten y ist diese Art der Betrachtung interessant. Die Grundidee des Newton-Verfahrens für diese Gleichung ist eine lokale Linearisierung (1. Interpretation, Abbildung 6.6). Ein Schritt des Newton-Verfahrens würde die Lösung des linearen Systems

$$F'(u_k)\,(u_{k+1} - u_k) = -\,(F(u_k) - y) \tag{6.20}$$

beinhalten. Da im Fall eines inversen Problems $F'(u_k)$ keinen *regulären* linearen Operator darstellt, ist die Gleichung in (6.20) selbst ein lineares schlechtgestelltes Problem und folglich ist u_{k+1} nicht wohldefiniert. Eine übliche Strategie zur Konstruktion von Newton-Verfahren für nichtlineare schlechtgestellte Probleme ist (6.20) mit Hilfe einer Regularisierungstechnik für lineare schlecht-gestellte Probleme zu erweitern. Zum Beispiel erhält man durch Anwendung einer linearen Tikhonov-Regularisierung (betrachte $u_{k+1} - u_k$ als Unbekannte) das sogenannte *Levenberg-Marquardt Verfahren*

$$(F'(u_k)^* F'(u_k) + \alpha_k I)\,(u_{k+1} - u_k) = -F'(u_k)^*\,(F(u_k) - y)\ .$$

Im Sinne von Newton-Verfahren für Funktionale in Variationsmethoden kann man das folgendermaßen erklären. Für das zu minimierende L^2 Fitting-Funktional

$$J(u) := \frac{1}{2}\,\|F(u) - y\|_2^2$$

betrachten wir das folgende quadratische Modell

$$\begin{aligned} M_k(u_k + d) &= \frac{1}{2}\,\|F(u_k) - y + F'(u_k)d\|_2^2 \ + \ \frac{\alpha_k}{2}\,\|d\|_2^2 \\ &= \frac{1}{2}\,\|F(u_k)\|_2^2 + \langle F'(u_k)d, F(u_k) - y\rangle + \langle d, (F'(u_k)^* F'(u_k) + \alpha_k I)\,d\rangle \end{aligned}$$

Man beachte, dass α_k hier einen variablen Regularisierungsparameter darstellt und nicht mit einer Schrittweite σ_k verwechselt werden sollte. Im Vergleich zu Quasi-Newton-Verfahren haben wir hier

$$A_k = F'(u_k)^* F'(u_k) + \alpha_k I$$

und als Suchrichtung $d_k = -A_k^{-1} J'(u_k)$.

Nun stellt sich die Frage: Wann ist A_k nahe bei $J''(u_k)$? Berechnet man die Hessematrix $J''(u_k)$, so erhält man für den Unterschied

$$J''(u_k) - A_k = \langle (F(u) - y)'', F(u) - y \rangle \ ,$$

d.h. der Fehler wird klein, falls

a) die Komponenten von $(F(u) - y)''$ klein sind (F nahezu linear)

b) die Residuen $F(u) - y$ klein sind (gute Anpassung, wenig Restrauschen).

Hier ist wieder die Notwendigkeit einer Regularisierung sichtbar und zeigt, dass man (nur) im Fall einer Lösung mit perfektem Fitting eine *lokal quadratische* Konvergenz bei den letzten Iterierten erwarten kann. Im Allgemeinen kann man nur Q-lineare Konvergenz erwarten.

(c) *Inexakte Newton bzw. Newton-Krylow-Verfahren*
Bei inexakten Newton-Verfahren löst man das lineare System

$$J''(u_k)d = -J'(u_k)$$

in jedem Schritt des Newton-Verfahrens inexakt, z.B. durch iterative lineare Algebra. Dieser Ansatz ist gut geeignet für large-scale Probleme. Analog zur Minimierung von Variationsproblemen bietet sich auch die numerische Lösung nichtlinearer partieller Differentialgleichungen prinzipiell das Newton-Verfahren als Grundlöser an. Die entsprechende Jacobi-Matrix ist immer dünnbesetzt und so bieten sich Krylow-Unterraum-Verfahren zur Lösung der linearen Gleichungssysteme an. Man spricht dann von Newton-Krylow-Verfahren. Ein wichtiger Repräsentant dieser Klasse ist das Newton-CG Verfahren. Im Krylow-Verfahren selbst tritt die Jacobi-Matrix nur in Matrix-Vektorprodukten auf, welche als Richtungsableitungen interpretiert werden können. Approximiert man diese durch Finite Differenzen, so erhält man komplett matrixfreie Verfahren.

Übersicht Konvergenzraten:

a) *Exaktes Newton Verfahren* ist Q-*quadratisch* konvergent.

b) *Quasi-Newton-Verfahren* ist Q-*superlinear* konvergent.

c) *Gauss-Newton, Levenberg-Marquardt* und *Gradientenabstiegsverfahren* sind Q-*linear* konvergent.