Skript zur Vorlesung
Nichtlineare Optimierung

Prof. Dr. R. Herzog

gehalten im WS2016/17
Technische Universität Chemnitz

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Dieses Vorlesungsskript basiert auf einem früheren Skript von Roland Herzog und Gerd Wachsmuth und den Büchern Geiger and Kanzow (1999); Nocedal and Wright (2006); Ulbrich and Ulbrich (2012). Viele Beweise sind diesen Büchern entnommen bzw. adaptiert.

Material für: 20 Vorlesungen à 90 Minuten

Fehler und Kommentare bitte an: roland.herzog@mathematik.tu-chemnitz.de

Stand: 31. Januar 2017

Inhaltsverzeichnis

KAPITEL 0

Einführung

Inhalt

§ 1 Grundbegriffe

Literatur: (Geiger and Kanzow, 1999, Kapitel 1), (Ulbrich and Ulbrich, 2012, Kapitel 1–2)

Wir betrachten in dieser Vorlesung **numerische Lösungsverfahren** für mathematische Optimierungsaufgaben der Form

$$\left.\begin{array}{rll} \text{Minimiere} & f(x) & \text{über } x \in \Omega \\ \text{sodass} & g_i(x) \leq 0, & i = 1, \ldots, m \\ \text{und} & h_j(x) = 0, & j = 1, \ldots, p. \end{array}\right\} \qquad \textbf{(NLP)}$$

Ω heißt **Grundmenge** und x die **Variable(n)** der Aufgabe. Die Funktion f heißt **Zielfunktion** (*objective function*). Die Bedingungen $g_i(x) \leq 0$ und $h_j(x) = 0$ heißen **Ungleichungs-** bzw. **Gleichungsnebenbedingungen** (*inequality/equality constraints*).

In dieser Vorlesung sind $\Omega = \mathbb{R}^n$ und f, g_i, $h_j : \mathbb{R}^n \to \mathbb{R}$ hinreichend glatt (in der Regel C^2) und i. A. *nichtlinear*. Da die Grundmenge Ω ein Kontinuum ist, sprechen wir auch von **kontinuierlicher Optimierung** im Unterschied zur **diskreten (ganzzahligen) Optimierung** (etwa im Fall $\Omega \subset \mathbb{Z}^n$). Damit wird die Aufgabe (NLP) auch als **nichtlineares Programm** (*nonlinear program* oder *nonlinear programming problem*) (**NLP**) und deren Lösung als **nichtlineare Programmierung** (*nonlinear programming*) bezeichnet.[1]

[1] Die Grundsteine der linearen Optimierung wurden in den 1940er Jahren von einer Projektgruppe SCOOP (Scientific Computation of Optimum Programs) um George Dantzig (1914–2005) bei der U.S. Air Force gelegt. Im militärischen Sprachgebrauch wurde die Ressourcenplanung als die Erstellung eines Programms bezeichnet, und diese Bezeichnung hat sich erhalten. Die Bezeichnung *nonlinear programming* geht laut Kjeldsen (2000) auf die Arbeit Kuhn and Tucker (1951) zurück.

Definition 1.1 (Grundbegriffe)**.** Für eine Optimierungsaufgabe (**NLP**) heißt

$$X = \{x \in \mathbb{R}^n : g_i(x) \leq 0 \text{ für alle } i = 1, \ldots, m,\ h_j(x) = 0 \text{ für alle } j = 1, \ldots, p\}$$

die **zulässige Menge** (*feasible set*), und jedes $x \in X$ heißt ein **zulässiger Punkt**. Im Fall $X = \emptyset$ heißt das Problem (**NLP**) **unzulässig** (*infeasible*). Der Wert

$$f^* = \inf\{f(x) : x \in X\}$$

heißt **Optimalwert** (*optimal value*) und ist $+\infty$, falls $X = \emptyset$. Ist $f^* = -\infty$, so heißt das Problem **unbeschränkt** (*unbounded*).

(a) Ein $x^* \in X$ heißt **globales Optimum**, wenn gilt:

$$f(x^*) \leq f(x) \quad \text{für alle } x \in X.$$

Gilt sogar

$$f(x^*) < f(x) \quad \text{für alle } x \in X \setminus \{x^*\},$$

dann heißt x^* **striktes globales Optimum**.

(b) Ein $x^* \in X$ heißt **lokales Optimum**, wenn es eine Umgebung U von x^* gibt, sodass gilt:

$$f(x^*) \leq f(x) \quad \text{für alle } x \in X \cap U.$$

Gilt sogar

$$f(x^*) < f(x) \quad \text{für alle } x \in X \cap U \setminus \{x^*\},$$

dann heißt x^* **striktes lokales Optimum**.

Ein Optimum nennt man auch **Minimum**, **Minimalstelle** oder **Lösung** von (**NLP**).[2]

Beachte: Eine Maximierungsaufgabe $\max\limits_{x \in X} f(x)$ kann durch Übergang zu $\min\limits_{x \in X}(-f)(x)$ immer in eine Minimierungsaufgabe umgeschrieben werden.

Definition 1.2 (Klassifikation von Optimierungsaufgaben)**.**

(a) Die Optimierungsaufgabe (**NLP**) heißt **frei** oder **unrestringiert**[3] (*unconstrained*), wenn $m = p = 0$ ist (Kapitel 1), andernfalls **gleichungs-** und/oder **ungleichungsbeschränkt** oder **-restringiert** (*equality/inequality constrained*).

(b) Ungleichungsbeschränkungen der Art

$$\ell_i \leq x_i \leq u_i, \quad i = 1, \ldots, n$$

mit $\ell_i \in \mathbb{R} \cup \{-\infty\}$ und $u_i \in \mathbb{R} \cup \{\infty\}$ heißen **Box-Beschränkungen** oder **Schranken** (*bound/box constraints, simple bounds*).

(c) **(Affin-)lineare Beschränkungen** (*affine constraints*) haben die Form

$$A\,x = b \quad \text{bzw.} \quad B\,x \leq c$$

mit Matrizen $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{m \times n}$ und Vektoren $b \in \mathbb{R}^p$, $c \in \mathbb{R}^m$.

[2] Der Plural von Optimum bzw. Minimum ist *Optima* bzw. *Minima*.

(d) Sind sowohl die Zielfunktion f als auch g und h (affin-)lineare Funktionen von x, so sprechen wir von **linearer Optimierung**. Ein lineares Optimierungsproblem heißt auch **lineares Programm** (*linear program*) (**LP**), also z. B. in Normalform:

$$\min \quad c^\top x \quad \text{sodass} \quad A\,x = b \quad \text{und} \quad x \geq 0.$$

(Solche Aufgaben werden hier nicht behandelt; als Algorithmen sind das Simplex-Verfahren und ggf. Innere-Punkte-Verfahren aus der Vorlesung *Grundlagen der Optimierung* bekannt.)

(e) Ist f ein quadratisches Polynom und sind g und h (affin-)linear, so sprechen wir von **quadratischer Optimierung**. Ein quadratisches Optimierungsproblem heißt auch **quadratisches Programm** (*quadratic program*) (**QP**).

(f) Sind die Funktionen f sowie die g_i, $i = 1, \ldots, m$ konvex und sind die Funktionen h_j, $j = 1, \ldots, p$ (affin-)linear, dann heißt die Aufgabe (NLP) **konvex**. Wir werden sehen, dass dieser Fall besonders gutartig ist.

Unsere Fragestellungen in dieser Vorlesung sind:

- Welche Grundtypen von numerischen Verfahren zur Lösung unrestringierter und restringierter Optimierungsaufgaben gibt es?
- Wie setzt man sie in der Praxis um?
- Was haben diese Verfahren für Konvergenzeigenschaften?

Wir werden im Verlauf der Vorlesung sehen, dass sich die schnelleren Verfahren an Optimalitätsbedingungen erster Ordnung (statt nur an den Funktionswerten der Zielfunktion bzw. der Nebenbedingungsfunktionen) orientieren. Deshalb wiederholen wir diese Bedingungen jeweils zu Beginn der Kapitel 1 (freie Optimierung) und Kapitel 2 (beschränkte Optimierung).

§ 2 Notation und Wiederholung

§ 2.1 Vektor- und Matrixnormen

Für Vektoren $x, y \in \mathbb{R}^n$ ist $x^\top y = \sum_{i=1}^n x_i y_i$ das **Euklidische Skalarprodukt**. Dieses erzeugt die **Euklidische Norm**, also $\|x\| = \sqrt{x^\top x}$. Allgemeiner erzeugt jede symmetrisch positiv definite (spd) Matrix $M \in \mathbb{R}^{n\times n}$ ($M \succ 0$) ein Skalarprodukt und eine Norm auf $\mathbb{R}^n$:

$$(x, y)_M := x^\top M\, y, \quad \|x\|_M := \sqrt{x^\top M\, x}.$$

Für eine Matrix $A \in \mathbb{R}^{m\times n}$ ist $\|A\|$ die (durch die Euklidischen Normen in $\mathbb{R}^n$ und $\mathbb{R}^m$ induzierte) **Matrixnorm**, also

$$\|A\| = \max_{x\neq 0} \frac{\|A\,x\|}{\|x\|}.$$

Es gilt $\|A\,x\| \leq \|A\|\|x\|$ und $\|AB\| \leq \|A\|\|B\|$ für Matrizen B passender Größe.

[3] manchmal ungenau auch als **unbeschränkt** bezeichnet

$\|A\|$ wird auch als **Spektralnorm** von A bezeichnet, und es gilt der Zusammenhang

$$\|A\| = \sqrt{\lambda_{\max}(A^\top A)}$$

mit dem größten Eigenwert von $A^\top A$. Ist A symmetrisch, dann ist $\|A\|$ der Betrag des betragsgrößten EW, also $\|A\| = \max\{|\lambda_{\min}(A)|, |\lambda_{\max}(A)|\}$.

Beachte: Die Spektralnorm hängt stetig von den Einträgen in A ab.

§ 2.2 Eigenwerte und Eigenvektoren

Ende 1. V

Jede symmetrische Matrix $A \in \mathbb{R}^{n\times n}$ besitzt eine orthogonale Transformation auf Diagonalgestalt (**Eigenzerlegung** oder **Spektralzerlegung**, *eigendecomposition* oder *spectral decomposition*), d. h., es existiert eine orthogonale Matrix $V \in \mathbb{R}^{n\times n}$ und eine Diagonalmatrix $\Lambda \in \mathbb{R}^{n\times n}$, sodass gilt:

$$A = V\Lambda V^\top.$$

Die Diagonale von Λ enthält die Eigenwerte λ_i, und die Spalten v_i von V sind zugehörige Eigenvektoren. Diese Zerlegung löst vollständig das **Eigenwertproblem** (*eigenvalue problem*)

$$A\,v = \lambda\,v.$$

Wir benötigen außerdem das **verallgemeinerte Eigenwertproblem** (*generalized eigenvalue problem*)

$$A\,v = \lambda\,M\,v$$

für den Fall einer spd Matrix $M \in \mathbb{R}^{n\times n}$ ($M \succ 0$) und weiterhin $A \in \mathbb{R}^{n\times n}$ symmetrisch. Für dieses existiert analog eine Zerlegung

$$A = V\Lambda V^\top,$$

wobei nun V orthonormal bzgl. des M^{-1}-Skalarprodukts ist: $V^\top M^{-1} V = I$.

Nach dem Satz von Courant-Fischer für Eigenwerte symmetrischer Matrizen gilt für den **verallgemeinerten Rayleigh-Quotient** der Matrix A bzgl. M

$$\lambda_{\min}(A; M) \le \frac{x^\top A\,x}{x^\top M\,x} \le \lambda_{\max}(A; M) \quad \text{für alle } x \neq 0. \tag{2.1}$$

Die zugehörigen verallgemeinerten Eigenvektoren erfüllen die Ungleichungen jeweils mit Gleichheit.

§ 2.3 Funktionen und ihre Ableitungen

Für Funktionen $f : \mathbb{R}^n \to \mathbb{R}$ sind

$$f'(x) = \left(\frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_n}(x)\right) \quad \text{und} \quad \nabla^2 f(x) = \left(\frac{\partial^2 f}{\partial x_j\,\partial x_i}(x)\right)_{i,j=1}^{n}$$

die **Ableitung** bzw. die **Hessematrix** an der Stelle $x \in \mathbb{R}^n$. Den transponierten Vektor bezeichnen wir als **Gradient** (**bezüglich des Euklidischen Skalarproduktes**)

$$\nabla f(x) = f'(x)^\top = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{pmatrix}.$$

Für vektorwertige Funktionen $f : \mathbb{R}^n \to \mathbb{R}^m$ bezeichnen wir mit

$$f'(x) = \Big(\frac{\partial f_i}{\partial x_j}(x)\Big)_{i,j=1}^{n,m}$$

die **Jacobimatrix** an der Stelle $x \in \mathbb{R}^n$. Die Einträge der Jacobimatrix heißen die **partiellen Ableitungen** von f.

Eine Funktion $f : \mathbb{R}^n \to \mathbb{R}^m$ heißt **stetig partiell differenzierbar** oder eine **C^1-Funktion**, wenn alle partiellen Ableitungen existieren und als Funktionen der Stelle stetig sind. Analog spricht man von **k-mal stetig partiell differenzierbaren** oder **C^k-Funktionen**, wenn alle möglichen partiellen Ableitungen bis einschließlich zur Differentiationsordnung k existieren und stetig sind.

Ist die Funktion $f : \mathbb{R}^n \to \mathbb{R}^m$ C^1, dann ist sie auch **total differenzierbar**, es gilt also

$$\frac{f(x+h) - f(x) - f'(x)\,h}{\|h\|} \to 0 \quad \text{für } h \to 0.$$

Insbesondere ist sie in alle Richungen $d \in \mathbb{R}^n$ **richtungsdifferenzierbar**, und für die Richtungsableitungen gilt

$$\delta f(x; d) := \lim_{t \searrow 0} \frac{f(x + t\,d) - f(x)}{t} = f'(x)\,d.$$

Ist die Funktion $f : \mathbb{R}^n \to \mathbb{R}$ C^2, dann ist die Hessematrix $\nabla^2 f(x)$ nach dem Satz von Schwarz symmetrisch.

Satz 2.1 (Taylor mit Zwischenstelle[4])**.** Es sei $G \subset \mathbb{R}^n$ offen und $f : G \to \mathbb{R}$ $(k+1)$-mal eine C^{k+1}-Funktion auf G. Falls x_0 und $x_0 + d$ und die gesamte Verbindungsstrecke in G liegen, dann existiert $\xi_k \in (0, 1)$, sodass gilt:

$$k = 0: \qquad f(x_0 + d) = f(x_0) + f'(x_0 + \xi_0\,d)\,d \quad \textbf{(Mittelwertsatz)}$$

$$k = 1: \qquad f(x_0 + d) = f(x_0) + f'(x_0)\,d + \frac{1}{2} d^\top \nabla^2 f(x_0 + \xi_1\,d)\,d$$

etc.

Satz 2.2 (Taylor mit integralem Restglied[5])**.** Es sei $G \subset \mathbb{R}^n$ offen und $f : G \to \mathbb{R}^m$ eine C^1-Funktion auf G. Falls x_0 und $x_0 + d$ und die gesamte Verbindungsstrecke in G liegen, dann gilt:

$$f(x_0 + d) = f(x_0) + \int_0^1 f'(x_0 + t\,d)\,d\,\mathrm{d}t = f(x_0) + \int_0^1 f'(x_0 + t\,d)\,\mathrm{d}t\,d.$$

§ 2.4 Konvergenzraten

Zur Charakterisierung der Konvergenzgeschwindigkeit von Algorithmen führen wir ein:

Definition 2.3 (Q-Konvergenzraten[6])**.** Es sei $\{x_n\} \subset \mathbb{R}^n$ eine Folge und $x^* \in \mathbb{R}^n$.

[4] siehe z. B. (Heuser, 2002, Satz 168.1)

[5] Dies folgt aus dem Hauptsatz der Differential- und Integralrechnung, angewendet auf die Funktionen $\varphi_i(t) = f_i(x_0 + t\,d)$, $i = 1, \ldots, m$.

(a) $\{x_k\}$ konvergiert gegen x^* (mindestens) **q-linear**, falls ein $c \in (0,1)$ existiert mit

$$\|x_{k+1} - x^*\| \leq c \|x_k - x^*\| \quad \text{für alle } k \in \mathbb{N} \text{ hinreichend groß.}$$

(b) $\{x_k\}$ konvergiert gegen x^* (mindestens) **q-superlinear**, falls es eine Nullfolge $\{\varepsilon_k\}$ gibt mit

$$\|x_{k+1} - x^*\| \leq \varepsilon_k \|x_k - x^*\| \quad \text{für alle } k \in \mathbb{N}.$$

(c) Es gelte $x_k \to x^*$. $\{x_k\}$ konvergiert gegen x^* (mindestens) **q-quadratisch**, falls ein $C > 0$ existiert mit

$$\|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^2 \quad \text{für alle } k \in \mathbb{N}.$$

Unter Verwendung der o- bzw. $\mathcal{O}$-Notation wird die Eigenschaft (b) manchmal auch als $\|x_{k+1} - x^*\| = o\big(\|x_k - x^*\|\big)$ und die Eigenschaft (c) als $\|x_{k+1} - x^*\| = \mathcal{O}\big(\|x_k - x^*\|^2\big)$ notiert.

Beachte: Die Eigenschaften der q-superlinearen und q-quadratischen Konvergenz einer Folge sind unabhängig von der Norm, in der der Abstand der Folgenglieder zum Grenzwert x^* gemessen wird. Die Eigenschaft der q-linearen Konvergenz hingegen überträgt sich nicht ohne Weiteres von einer Norm in eine andere, da beim Übergang die Konstante c nicht mehr notwendig < 1 bleibt.[7]

§ 2.5 Sonstiges

Für $\delta > 0$ und $x^* \in \mathbb{R}^n$ bezeichnen wir die offene δ-**Kugel** mit Mittelpunkt x^* mit

$$U_\delta(x^*) := \{x \in \mathbb{R}^n : \|x - x^*\| < \delta\}.$$

und allgemeiner mit

$$U_\delta^M(x^*) := \{x \in \mathbb{R}^n : \|x - x^*\|_M < \delta\}$$

die offene δ-Kugel bzgl. der M-Norm.

Es ist $\mathbb{R}^{++}+ := \{r \in \mathbb{R} : r > 0\}$, $\mathbb{N} = \{1, 2, 3, \ldots\}$ und $\mathbb{N}_0 = \{0, 1, 2, 3, \ldots, \}$.

Bei Mengeninklusionen schreiben wir immer $A \subset B$ statt $A \subseteq B$.

[6] Das „Q“ steht hier für „Quotient“.

[7] Dies ist Gegenstand von Übung 1, Aufgabe 2. Beispiele für Folgen mit unterschiedlichen Konvergenzraten werden in Übung 1, Aufgabe 1 behandelt.

KAPITEL 1

Verfahren der unrestringierten Optimierung

§ 3 Existenzaussagen und Optimalitätsbedingungen

Literatur: (Geiger and Kanzow, 1999, Kapitel 2), (Ulbrich and Ulbrich, 2012, Kapitel 5–6)

Wir behandeln in diesem Kapitel numerische Lösungsverfahren für die freie (unrestringierte) Optimierungsaufgabe (**NLP**) mit $m = p = 0$, also

$$\text{Minimiere} \quad f(x) \quad \text{über } x \in \mathbb{R}^n. \tag{**FO**}$$

Dabei nehmen wir die Zielfunktion f grundsätzlich als glatt (C^2) an.

Wir beschränken uns auf das Auffinden *lokaler* Minima. Die Bestimmung eines *globalen* Minimums ist algorithmisch sehr viel schwieriger und gelingt nur unter zusätzlichen Annahmen an die Zielfunktion und in der Regel auch nur für kleine Dimensionen $n \in \mathbb{N}$.[1]

Ausnahme: Wenn die Zielfunktion f konvex ist, dann ist schon jedes lokale Minimum ein globales Minimum. Außerdem sind notwendige Bedingungen 1. Ordnung (Satz 3.2) bereits hinreichend. Bedingungen 2. Ordnung werden nicht benötigt.

Die Existenz mindestens eines globalen Minimums erhält man unter einer Kompaktheitsbedingung:

Lemma 3.1 (Existenz eines globalen Minimums)**.** Es sei f stetig, und für irgendein $x_0 \in \mathbb{R}^n$ sei die Sublevelmenge

$$\mathcal{M}_f(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$$

kompakt. Dann besitzt die Aufgabe (**FO**) wenigstens ein globales Minimum.

Beweis. Offenbar kann man sich bei der Suche nach globalen Minima auf die Menge $\mathcal{M}_f(x_0)$ beschränken. Nach dem Satz von Weierstraß nimmt aber die Funktion f auf dieser Menge ihr Minimum an.[2]

Diese Bedingung benötigen wir später auch, um die Konvergenz verschiedener Verfahren zu zeigen.

Wir werden sehen, dass viele Verfahren die Optimalitätsbedingungen nutzen. Daher werden diese wiederholt, vgl. Vorlesung *Grundlagen der Optimierung*. Die Beweise werden nicht angegeben, sind aber mit dem Satz von Taylor leicht zu führen.

[1] Die globale Optimierung ist ein eigenes Teilgebiet der Optimierung.

[2] Dafür reicht sogar schon die Unterhalbstetigkeit von f. In Übung 1, Aufgabe 3 kommen Beispiele für die Nicht-Existenz globaler Minima.

Satz 3.2 (Notwendige Bedingung 1. Ordnung)**.** Es sei x^* ein lokales Minimum von (**FO**) und f sei C^1 in einer Umgebung von x^*. Dann ist $f'(x^*) = 0$.

Ein Punkt x mit $f'(x) = 0$ heißt ein **stationärer Punkt** von f.

Beachte: Die Bedingung $f'(x) = 0$ ist keinesfalls hinreichend dafür, dass x ein lokales Minimum von f ist. Betrachte etwa $f(x) = -x^2$ bei $x^* = 0$.

Ist f konvex, dann ist die Bedingung aus Satz 3.2 bereits *hinreichend* für die (globale!) Optimalität.

Satz 3.3 (Notwendige Bedingung 2. Ordnung)**.** Es sei x^* ein lokales Minimum von (**FO**) und f sei C^2 in einer Umgebung von x^*. Dann ist die Hessematrix $\nabla^2 f(x^*)$ positiv semidefinit.

Beachte: Auch die Bedingungen $f'(x) = 0$ und $\nabla^2 f(x)$ positiv semidefinit zusammen sind nicht hinreichend dafür, dass x ein lokales Minimum von f ist. Betrachte etwa $f(x) = -x^4$ bei $x^* = 0$ oder $f(x) = x_1^2 - x_2^4$ bei $x^* = (0, 0)$.

Satz 3.4 (Hinreichende Bedingung 2. Ordnung)**.** Die Funktion f sei C^2 in einer Umgebung eines Punktes x^*, und es gelte

(a) $f'(x^*) = 0$ (d. h., x^* ist ein stationärer Punkt) und

(b) $\nabla^2 f(x^*)$ ist positiv definit mit kleinstem Eigenwert $\alpha > 0$.

Dann gilt: Zu jedem $\beta \in (0, \alpha)$ gibt es eine Umgebung $U(x^*)$ von x^* mit der Eigenschaft

$$f(x) \geq f(x^*) + \frac{\beta}{2}\|x - x^*\|^2 \quad \text{für alle } x \in U(x^*). \tag{3.1}$$

Insbesondere ist x^* ein striktes lokales Minimum von (**FO**).

Die Eigenschaft (3.1) besagt, dass f in der Nähe von x^* mindestens **quadratisches Wachstum** besitzt. Man sagt auch, f erfülle eine **quadratische Wachstumsbedingung** oder verhalte sich *lokal gleichmäßig konvex*[3].

Erfüllt f bei x^* die notwendige, aber nicht die hinreichende Bedingung 2. Ordnung, so ist damit keine Entscheidung über das Vorliegen eines lokales Minimums möglich. (Beispiel: $f(x) = x^3$ und $f(x) = -x^4$ bei $x^* = 0$.) Es gibt also eine „unentscheidbare Lücke“ zwischen diesen Bedingungen. Diese ist aber so klein wie möglich (wenn man nur die ersten beiden Abbildungen nutzt).

§ 4 Die Grundaufgabe der Minimierung quadratischer Funktionen

Literatur: (Geiger and Kanzow, 1999, Kapitel 8.2, 13), (Nocedal and Wright, 2006, Kapitel 5.1)

Wir betrachten in diesem Abschnitt die einfachste Klasse sinnvoller unrestringierter Optimierungsaufgaben, nämlich die Minimierung quadratischer Polynome:

$$\text{Minimiere} \quad \phi(x) := \frac{1}{2} x^\top A\, x - b^\top x + c, \quad x \in \mathbb{R}^n. \tag{4.1}$$

[3] In der Tat ist dann f in einer Umgebung von x^* gleichmäßig konvex, da die Hessematrix in einer Umgebung von x^* gleichmäßig positiv definit ist (der kleinste EW ist gleichmäßig von 0 weg beschränkt).

Dabei ist $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ und $c \in \mathbb{R}$. Es wird o. B. d. A. angenommen, dass A symmetrisch ist. (Warum?)

Lemma 4.1 (Lösbarkeit und Lösungsmenge von (4.1))**.**

Es sei $A \in \mathbb{R}^{n\times n}$ symmetrisch. Dann gilt:

(a) Falls A positiv semidefinit ist ($A \succeq 0$), dann ist die Zielfunktion in (4.1) konvex. In diesem Fall sind äquivalent:

(i) (4.1) besitzt mindestens eine (globale) Lösung.

(ii) Die Zielfunktion ist nach unten beschränkt.

(iii) $Ax = b$ ist lösbar.

Die Lösungsmenge von (4.1) stimmt überein mit der Lösungsmenge des linearen Gleichungssystems $Ax = b$.

(b) Falls A nicht positiv semidefinit ist (also mindestens einen negativen Eigenwert besitzt), so ist die Zielfunktion in (4.1) nach unten unbeschränkt, und die Aufgabe ist nicht lösbar.

Beweis. Übung 1, Aufgabe 4

Folgerung 4.2 (Eindeutige Lösbarkeit von (4.1))**.**

Die Aufgabe (4.1) besitzt genau dann eine eindeutige (globale) Lösung x^*, wenn A spd ist. In diesem Fall ist der Optimalwert

$$\phi(x^*) = c - \frac{1}{2}\|x^*\|_A^2.$$

Wir gehen im Rest von § 4 davon aus, dass A symmetrisch positiv definit (spd) ist. Die Lösung von (4.1) ist dann also äquivalent zur Bestimmung der eindeutigen Lösung x^* des LGS $Ax = b$. Natürlich kommen dafür zunächst einmal **direkte Löser** wie das Gauß-Verfahren oder besser: die Bestimmung der Cholesky-Zerlegung $A = LL^\top$ in Betracht. Falls jedoch die Aufgabe hochdimensional ist (Anhaltspunkt: $n \geq 10\,000$)[4], so können **iterative Löser** ihre Stärken ausspielen. Iterative Löser bestimmen die Lösung x^* nicht in einem Schritt, sondern erzeugen eine Folge $\{x_k\}$, die dagegen konvergiert. Neben der Fähigkeit, mit hochdimensionalen Aufgaben besser umgehen zu können, haben iterative Löser einen weiteren Vorteil: Die aktuelle Iterierte des Verfahrens kann stets als approximative Lösung von $Ax = b$ bzw. von der Aufgabe (4.1) verwendet werden, wenn etwa eine hinreichende Genauigkeit erreicht ist, das Zeitbudget aufgebraucht wurde oder eine unerwartete Situation (etwa: A ist nicht wie erwartet positiv definit) auftritt. Bei direkten Lösern gibt es bis zum Erreichen der (bis auf Fließkommafehler exakten) Lösung keine verwendbaren Zwischenlösungen.

Wir besprechen jetzt die zwei wichtigsten Vertreter iterativer Verfahren für (4.1). Verallgemeinerungen davon werden dann im weiteren Verlauf der Vorlesung auch auf allgemeine unrestringierte Aufgaben (**FO**) angewendet.

[4] **Beachte:** Die Laufzeit direkter Löser ohne Besonderheiten in der Matrix-Struktur ist $\sim n^3$.

§ 4.1 Das Gradientenverfahren für quadratische Zielfunktionen

Das **Gradientenverfahren** (*gradient descent method*) oder auch **Verfahren des steilsten Abstiegs** (*method of steepest descent*) basiert auf folgender einfachen

Idee: Gehe vom aktuellen Punkt x_k aus ein Stück in Richtung des steilsten Abstiegs.[5]

Dazu ist zunächst zu klären, was die **Richtung des steilsten Abstiegs** für eine Funktion $f : \mathbb{R}^n \to \mathbb{R}$ im Punkt x ist. Interessanterweise hängt die Antwort vom Skalarprodukt ab, das man im Raum $\mathbb{R}^n$ der Optimierungsvariablen verwendet. Sei dieses Skalarprodukt durch die spd Matrix M erzeugt. Für einen Punkt x mit $f'(x) \neq 0$ sind die Richtungen des steilsten Abstiegs per Definition gerade die Minimierer von

$$R(d) = \frac{f'(x)\, d}{\|d\|_M}, \quad d \in \mathbb{R}^n, d \neq 0,$$

also die Vektoren mit kleinster Richtungsableitung (bezogen auf die Länge des Vektors). Es gilt

$$R(d) = \frac{(f'(x)\, M^{-1})\, M\, d}{\|d\|_M} \overset{CSU}{\geq} -\frac{\|M^{-1}\, f'(x)^\top\|_M\, \|d\|_M}{\|d\|_M} = -\|M^{-1}\, f'(x)^\top\|_M,$$

wobei wir die Cauchy-Schwarzsche Ungleichung im Hilbertraum $(\mathbb{R}^n, \|\cdot\|_M)$ genutzt haben. Gleichheit gilt genau dann, wenn $d = -\lambda\, M^{-1} f'(x)^\top$ mit einem $\lambda \geq 0$ gilt.

Definition 4.3 (M-Gradient, Richtung des steilsten Abstiegs)**.**

Es sei $f : \mathbb{R}^n \to \mathbb{R}$ differenzierbar und $x \in \mathbb{R}^n$.

(a) Der Vektor

$$\nabla_M f(x) := M^{-1}\, f'(x)^\top = M^{-1}\, \nabla f(x) \tag{4.2}$$

heißt der **Gradient** von f im Punkt x **bezüglich des M-Skalarprodukts** oder kurz: der **M-Gradient** von f im Punkt x.

(b) Der Vektor $-\nabla_M f(x)$ und alle seine positiven Vielfachen heißen eine **Richtung des steilsten Abstiegs** von f im Punkt x **bezüglich des M-Skalarprodukts**.

Beachte: Im Allgemeinen ist also zur Berechnung des M-Gradienten ein lineares Gleichungssystem (LGS) zu lösen!

Ende 2. V

Mittels des M-Gradienten kann man beliebige Richtungsableitungen von f im Punkt x berechnen:

$$\big(\nabla_M f(x),\, d\big)_M = f'(x)\, d. \tag{4.3}$$

Hier ist $(\cdot,\, \cdot)_M$ das M-Skalarprodukt.[6]

Speziell für das Euklidische Skalarprodukt $M = I$ gilt $\nabla_I f(x) = \nabla f(x)$. In diesem Fall muss man also zur Berechnung des Gradienten kein LGS lösen.

[5] In der Übung 2, Aufgabe 5 interpretieren wir das Gradientenverfahren auch als ein explizites Euler-Verfahren für den Gradientenfluss zur Energie ϕ.

[6] Es ist also $\nabla_M f(x)$ nicht anderes als der Riesz-Repräsentant der Ableitung $f'(x)$. Diese Aussage gilt auch in allgemeinen Hilberträumen, vgl. Vorlesung *Funktionalanalysis*.

Für unsere Zielfunktion (4.1) gilt:

$$\begin{aligned} & \nabla\phi(x) = A\,x - b \\ \Rightarrow \quad & \nabla_M\phi(x) = M^{-1}(A\,x - b). \end{aligned}$$

Zur Abkürzung führen wir hier noch den Begriff des **Residuums** (*residual*) zur Stelle x bzgl. des LGS $A\,x = b$ ein:[7]

$$r := A\,x - b = \nabla\phi(x).$$

Eine Richtung des steilsten Abstiegs ist damit $d = -\nabla_M\phi(x) = -M^{-1}r$.

Nachdem das geklärt ist, bleibt die Frage der Schrittlänge, die zu gehen ist. Diese wird so gewählt, dass das (gleichmäßig konvexe) eindimensionale quadratische Polynom

$$\mathbb{R} \ni \alpha \mapsto \phi(x + \alpha\,d) \in \mathbb{R} \tag{4.4}$$

exakt minimiert wird, wobei die Richtung $d \in \mathbb{R}^n$ im Moment noch beliebig ist.

Lemma 4.4 (eindimensionale Minimierung)**.** Es sei $x \in \mathbb{R}^n$ und $d \neq 0$. Die eindeutige Lösung der Aufgabe (4.4) ist gegeben durch

$$\alpha^* := -\frac{d^\top(A\,x - b)}{d^\top A\,d} = -\frac{d^\top r}{d^\top A\,d}. \tag{4.5}$$

Als Differenz der Funktionswerte ergibt sich

$$\phi(x + \alpha^* d) - \phi(x) = -\frac{1}{2}\frac{(d^\top r)^2}{d^\top A\,d}. \tag{4.6}$$

Beweis. Zur Bestimmung der Minimalstelle bilden wir die Ableitung von (4.4) nach α und setzen diese gleich null:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\alpha}\phi(x + \alpha\,d) &= \frac{\mathrm{d}}{\mathrm{d}\alpha}\left[\frac{1}{2}(x + \alpha\,d)^\top A\,(x + \alpha\,d) - b^\top(x + \alpha\,d) + c\right] \\ &= d^\top A\,(x + \alpha\,d) - b^\top d \\ &= d^\top(A\,x - b) + \alpha\,d^\top A\,d \overset{!}{=} 0. \end{aligned}$$

Daraus folgt (4.5). Für die Differenz zweier Funktionswerte (zunächst zu beliebiger Schrittweite α) berechnen wir

$$\begin{aligned} \phi(x + \alpha\,d) - \phi(x) &= \frac{1}{2}(x + \alpha\,d)^\top A\,(x + \alpha\,d) - b^\top(x + \alpha\,d) + c \\ &\quad - \frac{1}{2}x^\top A\,x + b^\top x - c \\ &= \alpha\,d^\top A\,x + \frac{1}{2}\alpha^2 d^\top A\,d - \alpha\,b^\top d \\ &= \alpha\,d^\top r + \frac{1}{2}\alpha^2 d^\top A\,d. \end{aligned} \tag{4.7}$$

[7] **Achtung:** Manchmal wird r in der Literatur mit umgekehrtem Vorzeichen definiert.

Setzen wir nun den Wert für α^* ein, so ergibt sich weiter:

$$\begin{aligned} \cdots &= -\frac{(d^\top r)^2}{d^\top A\, d} + \frac{1}{2}\frac{(d^\top r)^2}{(d^\top A\, d)^2}\, d^\top A\, d \\ &= -\frac{1}{2}\frac{(d^\top r)^2}{d^\top A\, d}. \end{aligned} \tag{4.8}$$

Jetzt können wir das Verfahren des steilsten Abstiegs bzgl. des M-Skalarprodukts (also mit $d = -\nabla_M f(x) = -M^{-1}r$) für die quadratische Grundaufgabe (4.1) in implementierbarer Form angeben:

Algorithmus 4.5 (Gradientenverfahren bei quadratischer Zielfunktion)**.**
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: rechte Seite $b \in \mathbb{R}^n$
Eingabe: spd Matrix A (oder Matrix-Vektor-Produkte mit A)
Eingabe: spd Matrix M (oder Matrix-Vektor-Produkte mit M^{-1})
Ausgabe: näherungsweise Lösung der Aufgabe (4.1) bzw. von $A\,x = b$

1: Setze $k := 0$
2: Setze $r_0 := A\,x_0 - b$
3: Setze $d_0 := -M^{-1}r_0$
4: Setze $\delta_0 := -r_0^\top d_0$ $\{\delta_0 = \|\nabla_M \phi(x_0)\|_M^2\}$
5: **while** Abbruchkriterium nicht erfüllt **do**
6: Setze $q_k := A\,d_k$
7: Setze $\alpha_k := \delta_k/(d_k^\top q_k)$
8: Setze $x_{k+1} := x_k + \alpha_k\, d_k$
9: Setze $r_{k+1} := r_k + \alpha_k\, q_k$
10: Setze $d_{k+1} := -M^{-1}r_{k+1}$
11: Setze $\delta_{k+1} := -r_{k+1}^\top d_{k+1}$ $\{\delta_{k+1} = \|\nabla_M \phi(x_{k+1})\|_M^2\}$
12: Setze $k := k + 1$
13: **end while**
14: **return** x_k

Dieses Verfahren steht als `steepest_descent_quadratic.m` auf der Homepage zur Vorlesung zur Verfügung.

Bemerkung 4.6 (zum Gradientenverfahren bei quadratischer Zielfunktion)**.**

(a) Der Algorithmus 4.5 ist gleichzeitig ein iteratives Verfahren zur Lösung der Minimierungsaufgabe (4.1) wie auch des LGS $A\,x = b$ mit spd Matrix A.

(b) Im Unterschied zu direkten Gleichungslösern benötigt das Verfahren nur Matrix-Vektor-Produkte mit A und keinen Zugriff auf einzelne Elemente der Matrix. Dies ist ein weiterer Aspekt bei der Auswahl des Lösungsverfahrens für (4.1).

(c) Die Durchführung des Gradientenverfahrens im (vom Anwender zu wählenden) M-Skalarprodukt bezeichnet man auch als **Vorkonditionierung**. Die Matrix M selbst bezeichnet man in diesem Zusammenhang auch als **Vorkonditionierer** (*preconditioner*). In der Literatur wird daher der Algorithmus 4.5 oft als **vorkonditioniertes Gradientenverfahren** geführt.

Der Fall $M = I$ entspricht dem klassischen Gradientenverfahren (ohne Vorkonditionierung).

(d) Das Verfahren muss i. W. vier Vektoren speichern: Iterierte x_k, Residuen r_k, Suchrichtungen d_k und $q_k = A\,d_k$. Diese können zyklisch überschrieben werden.

(e) Pro Iteration wird ein Matrix-Vektor-Produkt mit A benötigt. Außerdem muss pro Iteration einmal der Vorkonditionierer „angewendet“ werden, d. h., M^{-1} mal Vektor berechnet bzw. das entsprechende LGS mit Koeffizientenmatrix M gelöst werden.

(f) Zur Vermeidung der Akkumulation von Rundungsfehlern kann regelmäßig (z. B. alle 50 Iterationen) einmal das Residuum gemäß $r_{k+1} := A\,x_{k+1} - b$ neu berechnet werden.

(g) Es gilt

$$0 < \lambda_{\min}(A; M) \leq \frac{1}{\alpha_k} = \frac{d_k^\top A\,d_k}{d_k^\top M\,d_k} \leq \lambda_{\max}(A; M), \tag{4.9}$$

solange $d_k \neq 0$ ist, also solange $x_k \neq x^*$ ist.

(h) Die Größe δ_k enthält jeweils die M-Norm des M-Gradienten, denn:

$$\begin{aligned}\delta_k &= -r_k^\top d_k = r_k^\top M^{-1} r_k = (\nabla\phi(x_k))^\top M^{-1}\nabla\phi(x_k)\\ &= (\nabla_M\phi(x_k))^\top M\,\nabla_M\phi(x_k) = \|\nabla_M\phi(x_k)\|_M^2.\end{aligned}$$

Wie wir später sehen werden, kann diese in einem Abbruchkriterium verwendet werden.

(i) Ohne wesentlichen zusätzlichen Aufwand kann der aktuelle Funktionswert $\phi_k := \phi(x_k)$ mitberechnet werden, falls der Benutzer neben A und b auch die Konstante c mit übergibt. Es gilt die Rekursion (Beweis?)

$$\begin{aligned}\phi(x_0) &= c + \frac{1}{2}x_0^\top(r_0 - b)\\ \phi(x_{k+1}) &= \phi(x_k) - \frac{1}{2}\,\alpha_k\,\delta_k.\end{aligned}$$

Es stellt sich nun die Frage nach dem Konvergenzverhalten sowie der Rolle des Vorkonditionierers/Skalarprodukts M. Dazu geben wir zunächst ein Hilfsresultat an, das die Funktionswerte, den Fehler $x - x^*$ und das Residuum in Beziehung setzt:

Lemma 4.7. Es gilt

$$\phi(x) - \phi(x^*) = \frac{1}{2}\|x - x^*\|_A^2 = \frac{1}{2}\|r\|_{A^{-1}}^2. \tag{4.10}$$

Beweis. Durch direkte Rechnung oder aber aus (4.6) mit $d = x^* - x$ und $\alpha^* = 1$ ergibt sich unter Verwendung von $r = A\,x - b = A\,(x - x^*)$:

$$\begin{aligned} -\big[\phi(x^*) - \phi(x)\big] &= \frac{1}{2}\frac{\big[(x^* - x)^\top r\big]^2}{d^\top A\,d} = \frac{1}{2}\frac{\big[(x - x^*)^\top A\,(x - x^*)\big]^2}{d^\top A\,d} \\ &= \frac{1}{2}\frac{\|x - x^*\|_A^4}{\|x - x^*\|_A^2} = \frac{1}{2}\|x - x^*\|_A^2 \\ &= \frac{1}{2}(A\,x - b)^\top A^{-1}(A\,x - b) = \frac{1}{2}\|r\|_{A^{-1}}^2. \end{aligned}$$

Für die Iterierten eines Verfahrens für (4.1) sogar mit zunächst beliebigen Suchrichtungen d_k, aber dazu passenden optimalen Schrittweiten α_k (wie Algorithmus 4.5) gilt nun:

$$\begin{aligned} \phi(x_{k+1}) - \phi(x^*) &= \frac{1}{2}\|r_{k+1}\|_{A^{-1}}^2 = \frac{1}{2}\|r_k + \alpha_k\,A\,d_k\|_{A^{-1}}^2 \\ &= \frac{1}{2}\|r_k\|_{A^{-1}}^2 + \alpha_k\,r_k^\top d_k + \frac{1}{2}\alpha_k^2\,d_k^\top A\,d_k \\ &= \frac{1}{2}\|r_k\|_{A^{-1}}^2 - \frac{(r_k^\top d_k)^2}{d_k^\top A\,d_k} + \frac{1}{2}\frac{(r_k^\top d_k)^2}{d_k^\top A\,d_k} \quad \text{wegen } \alpha_k = -\frac{d_k^\top r_k}{d_k^\top A\,d_k} \\ &= \left(1 - \frac{(r_k^\top d_k)^2}{(d_k^\top A\,d_k)\,(r_k^\top A^{-1} r_k)}\right)\big(\phi(x_k) - \phi(x^*)\big). \end{aligned}$$

Hier setzen wir nun den speziellen Zusammenhang $r_k = -M\,d_k$ für die Iterierten aus Algorithmus 4.5 ein:

$$= \left(1 - \frac{(d_k^\top M\,d_k)^2}{(d_k^\top A\,d_k)\,(d_k^\top M\,A^{-1} M\,d_k)}\right)\big(\phi(x_k) - \phi(x^*)\big). \tag{4.11}$$

Für die weitere Abschätzung benutzen wir die **Kantorovich-Ungleichung**, die wir hier zunächst für den Fall $M = I$ angeben:

Lemma 4.8 (Kantorovich-Ungleichung)**.** Es sei $Q \in \mathbb{R}^{n\times n}$ spd und $\alpha := \lambda_{\min}(Q)$ sowie $\beta := \lambda_{\max}(Q)$. Dann gilt

$$\frac{(x^\top Q\,x)\,(x^\top Q^{-1} x)}{\|x\|^4} \le \frac{(\alpha + \beta)^2}{4\,\alpha\,\beta} \le \frac{\beta}{\alpha} \tag{4.12}$$

für alle $x \in \mathbb{R}^n$, $x \neq 0$.

Beachte: Für den Rayleigh-Quotienten von Q gilt nach (2.1)

$$\frac{x^\top Q\,x}{\|x\|^2} \le \lambda_{\max}(Q) = \beta \quad \text{und analog} \quad \frac{x^\top Q^{-1}\,x}{\|x\|^2} \le \lambda_{\max}(Q^{-1}) = 1/\alpha.$$

Die zugehörigen Eigenvektoren erfüllen die Ungleichungen jeweils mit Gleichheit. Die offensichtliche Abschätzung

$$\frac{(x^\top Q\,x)\,(x^\top Q^{-1} x)}{\|x\|^4} \le \frac{\beta}{\alpha}$$

ist jedoch nicht scharf, da derselbe Vektor x i. A. nicht gleichzeitig Eigenvektor zum größten und zum kleinsten Eigenwert sein kann. (4.12) verbessert diese Abschätzung.

Ende 3. V

Mit Hilfe der **Konditionszahl** der Matrix Q

$$\kappa := \kappa(Q) := \frac{\beta}{\alpha} \geq 1 \tag{4.13}$$

können wir (4.12) auch in der äquivalenten Form

$$\frac{(x^\top Q\, x)\,(x^\top Q^{-1} x)}{\|x\|^4} \leq \frac{(\kappa+1)^2}{4\,\kappa} \leq \kappa \tag{4.14}$$

schreiben (nachrechnen!).

Beweis. [8] Es seien $\lambda_1, \ldots, \lambda_n > 0$ die Eigenwerte von Q und $v_1, \ldots, v_n$ ein Satz zugehöriger orthonormaler Eigenvektoren. Es sei $x \in \mathbb{R}^n$, $x \neq 0$ beliebig. Wir stellen x dar als $x = \sum_{i=1}^n \gamma_i\, v_i$. O. B. d. A. sei $\|x\|^2 = \sum_{i=1}^n \gamma_i^2 = 1$. Einsetzen in die linke Seite von (4.12) ergibt:

$$\frac{(x^\top Q\, x)\,(x^\top Q^{-1} x)}{\|x\|^4} = \underbrace{\Big[\sum_{i=1}^n \lambda_i\, \gamma_i^2\Big]}_{=\mathbb{E}(T)}\ \underbrace{\Big[\sum_{i=1}^n \frac{1}{\lambda_i}\, \gamma_i^2\Big]}_{=\mathbb{E}(1/T)}.$$

Es ist jetzt aus Gründen der Übersichtlichkeit hilfreich, diese Faktoren als Erwartungswerte einer „Zufallsvariablen“ T bzw. $1/T$ zu interpretieren, wobei T die Werte $\lambda_i \in [\alpha, \beta]$ mit „Wahrscheinlichkeit“ γ_i^2 annimmt. Für $0 < \alpha \leq T \leq \beta$ gilt

$$0 \leq (\beta - T)\,(T - \alpha) = (\beta + \alpha - T)\,T - \alpha\,\beta,$$

also auch

$$\frac{1}{T} \leq \frac{\alpha + \beta - T}{\alpha\,\beta}$$

und daher (Erwartungswert nehmen)

$$\begin{aligned}
\mathbb{E}(T)\,\mathbb{E}(1/T) &\leq \mathbb{E}(T)\,\frac{\alpha+\beta-\mathbb{E}(T)}{\alpha\,\beta} \\
&= \frac{(\alpha+\beta)^2}{4\,\alpha\,\beta} - \frac{1}{\alpha\,\beta}\big[\mathbb{E}(T) - \frac{1}{2}(\alpha+\beta)\big]^2 \\
&\leq \frac{(\alpha+\beta)^2}{4\,\alpha\,\beta}.
\end{aligned}$$

Damit ist die erste (wesentliche) Ungleichung in (4.12) bewiesen. Die zweite Ungleichung folgt elementar aus $0 < \alpha \leq \beta$.

Um die Kantorovich-Ungleichung zur Abschätzung von (4.11) verwenden zu können, benötigen wir noch eine Verallgemeinerung:

Folgerung 4.9 (verallgemeinerte Kantorovich-Ungleichung)**.** Es seien $A \in \mathbb{R}^{n\times n}$ und $M \in \mathbb{R}^{n\times n}$ beide spd und $\alpha := \lambda_{\min}(A; M)$ sowie $\beta := \lambda_{\max}(A; M)$. Dann gilt

$$\frac{(x^\top A\, x)\,(x^\top M\, A^{-1} M\, x)}{\|x\|_M^4} \leq \frac{(\alpha+\beta)^2}{4\,\alpha\,\beta} \leq \frac{\beta}{\alpha} \tag{4.15}$$

für alle $x \in \mathbb{R}^n$, $x \neq 0$.

[8] Wir folgen hier dem Beweis von Anderson (1971), wie er in der Masterarbeit (Alpargu, 1996, Abschnitt 1.2.2) wiedergegeben ist.

Beweis. Wir benutzen die Cholesky-Zerlegung $M = LL^\top$ und setzen $y := L^\top x$, also $x = L^{-\top} y$ ein:

$$\frac{(x^\top A\, x)\,(x^\top M\, A^{-1} M\, x)}{(x^\top M\, x)^2} = \frac{(y^\top L^{-1} A\, L^{-\top} y)\,(y^\top L^\top A^{-1} L\, y)}{(y^\top y)^2}.$$

Dies entspricht der Form in (4.12) mit der spd Matrix $Q := L^{-1} A\, L^{-\top}$. Deren Eigenpaare (λ, v) erfüllen

$$Q\, v = L^{-1} A\, L^{-\top} v = \lambda\, v, \quad v \neq 0,$$

also auch

$$A\, L^{-\top} v = \lambda\, L\, v.$$

Ersetzen wir noch v durch $L^\top w$, so erhalten wir

$$A\, w = \lambda\, M\, w. \tag{4.16}$$

Damit ist gezeigt, dass (λ, v) genau dann ein Eigenpaar von $Q = L^{-1} A\, L^{-\top}$ ist, wenn $(\lambda, w = L^{-\top} v)$ ein Eigenpaar des verallgemeinerten Eigenwertproblems (4.16) ist. Insbesondere sind die Eigenwerte gleich. Es seien nun wie angenommen $0 < \alpha \leq \beta$ die extremalen Eigenwerte von (4.16), dann sind dies auch die extremalen Eigenwerte von Q, und die Behauptung folgt aus (4.12).

Mit Hilfe der **verallgeinerten Konditionszahl** von A bzgl. M,

$$\kappa := \kappa(A; M) := \frac{\beta}{\alpha} \geq 1 \tag{4.17}$$

können wir (4.15) auch wieder in der äquivalenten Form

$$\frac{(x^\top A\, x)\,(x^\top M\, A^{-1} M\, x)}{\|x\|_M^4} \leq \frac{(\kappa+1)^2}{4\,\kappa} \leq \kappa \tag{4.18}$$

schreiben.

Mit Hilfe von (4.15) folgt nun für die in (4.11) abzuschätzende Klammer:

$$1 - \frac{(d_k^\top M\, d_k)^2}{(d_k^\top A\, d_k)\,(d_k^\top M\, A^{-1} M\, d_k)} \leq 1 - \frac{4\,\alpha\,\beta}{(\alpha+\beta)^2} = \frac{(\beta-\alpha)^2}{(\beta+\alpha)^2} = \left(\frac{\kappa-1}{\kappa+1}\right)^2.$$

Beachte: Auf die absolute Skalierung von M kommt es dabei nicht an, denn $\kappa(A; M) = \kappa(A; \gamma\, M)$ für alle $\gamma > 0$.

Damit haben wir das klassische Konvergenzresultat des (vorkonditionierten) Gradientenverfahrens bewiesen:

Satz 4.10 (Konvergenzsatz des Gradientenverfahrens für (4.1))**.**

Es seien $A \in \mathbb{R}^{n\times n}$ und $M \in \mathbb{R}^{n\times n}$ beide spd. Weiter seien $0 < \alpha \leq \beta$ die extremalen Eigenwerte von (4.16). Dann gilt: Das Gradientenverfahren (Algorithmus 4.5) konvergiert für jeden beliebigen Startwert $x_0 \in \mathbb{R}^n$ gegen die eindeutige Lösung $x^* = A^{-1} b$ von (4.1), und es gilt mit κ aus (4.17):

$$\phi(x_{k+1}) - \phi(x^*) \leq \left(\frac{\kappa-1}{\kappa+1}\right)^2 \big(\phi(x_k) - \phi(x^*)\big) \tag{4.19a}$$

$$\|x_{k+1} - x^*\|_A \leq \left(\frac{\kappa-1}{\kappa+1}\right) \|x_k - x^*\|_A \tag{4.19b}$$

und daher auch

$$\|x_k - x^*\|_A \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|x_0 - x^*\|_A. \tag{4.19c}$$

Weiterhin ist die Folge $\{\phi(x_k)\}$ und damit die Fehlernorm $\|x_k - x^*\|_A$ monoton fallend.

In Übung 2, Aufgabe 6 schätzen wir für verschiedene Konditionszahlen κ die benötigten Iterationszahlen für das Gradientenverfahren ab, um eine vorgegebene Fehlerreduktion zu erzielen.

Bemerkung 4.11 (zu Satz 4.10)**.**

(a) (4.19b) zeigt die q-lineare Konvergenz der Folge $\{x_k\}$ gegen x^* in der A-Norm.

(b) Der Kontraktionsfaktor $0 \leq \frac{\kappa-1}{\kappa+1} < 1$ ist entscheidend für die Konvergenzgeschwindigkeit. Es kommt also auf das (Kontrast-)Verhältnis zwischen größtem und kleinstem Eigenwert von (4.16) an. Dieses gering zu halten ist die Aufgabe des Vorkonditionierers.

(c) Im günstigen Extremfall ist $\kappa = 1$, was Konvergenz zur Lösung in einem Schritt bedeutet. Das tritt genau dann ein, wenn M bis auf Vielfache mit A übereinstimmt. Da man aber M^{-1} im Verfahren „anwenden“ können muss, also Gleichungssysteme mit M lösen, kann man genauso gut $A^{-1}b$ direkt berechnen.

(d) Ein guter Vorkonditionierer/ein geeignetes Skalarprodukt ist ein Kompromiss zwischen einer nicht zu großen Konditionszahl κ und vertretbarem Aufwand bei der Anwendung von M^{-1}. Das Finden eines guten Vorkonditionieres setzt in aller Regel Problemwissen voraus.

(e) Geometrisch erkennt man einen guten Vorkonditionierer daran, dass die Niveaumengen der Zielfunktion ϕ im modifizierten Koordinatensystem $y = L\,x$ möglichst kreisförmig sind. I. A. sind diese Niveaumengen Ellipsoide, und κ ist das Verhältnis von der längsten zur kürzesten Halbachse.

(f) Da das Gradientenverfahren schrittweise die Zielfunktion (4.1) und damit — siehe (4.10) — die A-Norm des Fehlers $x_k - x^*$ minimiert, ist es natürlich, die Konvergenz in (4.19) auch in diesen Größen zu messen.

Es bleibt noch die Frage nach der Abbruchbedingung zu klären. Dabei können verschiedene Größen von Interesse sein:

(a) Ist man mit einem Punkt x_k zufrieden, der fast stationär ist, also mit $\|r_k\|_{M^{-1}}$ klein?

(b) Ist man mit einem Punkt x_k zufrieden, dessen Funktionswert in der Nähe des Optimalwertes liegt, also mit $\phi(x_k) - \phi(x^*)$ klein, d. h., $\|x_k - x_*\|_A$ klein?

(c) Ist man mit einem Punkt x_k zufrieden, dessen Abstand vom Optimum in der benutzerdefinierten Norm M des Vorkonditionierers klein ist, also $\|x_k - x_*\|_M$ klein?

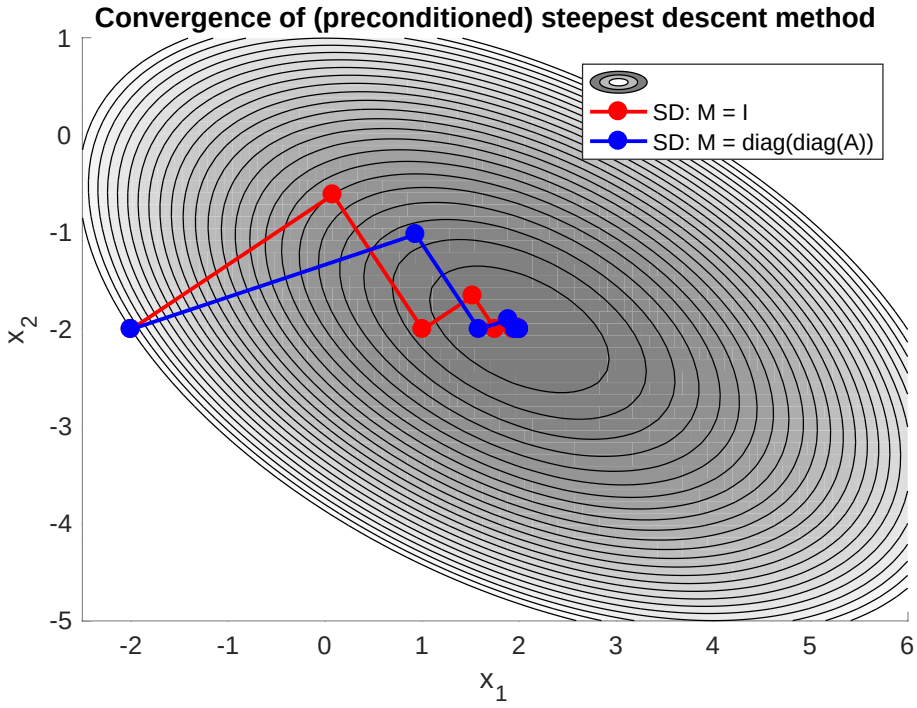

ABBILDUNG 4.1. Konvergenzverhalten des Gradientenverfahrens ohne Vorkonditionierung ($M = I$) und mit Vorkonditionierung ($M =$ Diagonale von A) am Beispiel $A = \left(\begin{smallmatrix} 3 & 2 \\ 2 & 6 \end{smallmatrix}\right)$, $b = \left(\begin{smallmatrix} 2 \\ -8 \end{smallmatrix}\right)$ und Startwert $x_0 = \left(\begin{smallmatrix} -2 \\ -2 \end{smallmatrix}\right)$.

Die einzige dieser drei Größen, die man ohne Kenntnis der Lösung x^* bzw. ohne Kenntnis von $\phi(x^*)$ messen kann, ist $\delta_k := \|r_k\|^2_{M^{-1}}$. Die anderen kann man aber mit Hilfe des folgenden Lemmas abschätzen:

Lemma 4.12. Es seien $A \in \mathbb{R}^{n\times n}$ und $M \in \mathbb{R}^{n\times n}$ beide spd. Weiter seien $0 < \alpha \le \beta$ die extremalen Eigenwerte von (4.16). Dann gilt

$$\alpha \le \frac{\|x\|^2_A}{\|x\|^2_M} \le \beta \quad \text{für alle } x \neq 0 \tag{4.20}$$

und

$$\alpha \le \frac{\|x\|^2_{A\,M^{-1}A}}{\|x\|^2_A} \le \beta \quad \text{für alle } x \neq 0. \tag{4.21}$$

Beweis. Behauptung (4.20) ist nichts anderes als (2.1), was wiederum aus dem Satz von Courant-Fischer folgte. Für (4.21) benutzen wir, dass

$$A\,x = \lambda\,M\,x \quad \Leftrightarrow \quad A\,M^{-1}y = \lambda\,y \quad \Leftrightarrow \quad A\,M^{-1}A\,z = \lambda\,A\,z$$

gilt. Das zeigt, dass die verallgemeinerten Eigenwerte des Paares (A, M) übereinstimmen mit denen des Paares $(A\,M^{-1}A, A)$. Daraus folgt die Behauptung.

In vielen Implementierungen wird eine der folgenden Kombinationen aus relativem und absolutem Kriterium als Abbruchbedingung verwendet:

$$\|r_k\|_{M^{-1}} \le \varepsilon_{\text{rel}}\,\|r_0\|_{M^{-1}}, \qquad \text{also } \delta_k \le \varepsilon^2_{\text{rel}}\,\delta_0 \tag{4.22a}$$

$$\|r_k\|_{M^{-1}} \le \varepsilon_{\text{abs}}, \qquad \text{also } \delta_k \le \varepsilon^2_{\text{abs}} \tag{4.22b}$$

$$\|r_k\|_{M^{-1}} \le \varepsilon_{\text{rel}}\,\|r_0\|_{M^{-1}} + \varepsilon_{\text{abs}}, \qquad \text{also } \delta_k^{1/2} \le \varepsilon_{\text{rel}}\,\delta_0^{1/2} + \varepsilon_{\text{abs}} \tag{4.22c}$$

$$\|r_k\|_{M^{-1}} \le \max\bigl\{\varepsilon_{\text{rel}}\,\|r_0\|_{M^{-1}},\ \varepsilon_{\text{abs}}\bigr\}, \qquad \text{also } \delta_k \le \max\bigl\{\varepsilon_{\text{rel}}^2\,\delta_0,\ \varepsilon_{\text{abs}}^2\bigr\}. \tag{4.22d}$$

Folgerung 4.13. Bei Verwendung der Kriterien aus (4.22) gilt jeweils:

$$\left.\begin{aligned} \|x_k - x^*\|_A &\le \sqrt{\kappa}\,\varepsilon_{\text{rel}}\,\|x_0 - x^*\|_A \\ \|x_k - x^*\|_M &\le \kappa\,\varepsilon_{\text{rel}}\,\|x_0 - x^*\|_M \end{aligned}\right\} \tag{4.23a}$$

$$\left.\begin{aligned} \|x_k - x^*\|_A &\le (1/\sqrt{\alpha})\,\varepsilon_{\text{abs}} \\ \|x_k - x^*\|_M &\le (1/\alpha)\,\varepsilon_{\text{abs}} \end{aligned}\right\} \tag{4.23b}$$

$$\left.\begin{aligned} \|x_k - x^*\|_A &\le \sqrt{\kappa}\,\varepsilon_{\text{rel}}\,\|x_0 - x^*\|_A + (1/\sqrt{\alpha})\,\varepsilon_{\text{abs}} \\ \|x_k - x^*\|_M &\le \kappa\,\varepsilon_{\text{rel}}\,\|x_0 - x^*\|_M + (1/\alpha)\,\varepsilon_{\text{abs}} \end{aligned}\right\} \tag{4.23c}$$

$$\left.\begin{aligned} \|x_k - x^*\|_A &\le \max\bigl\{\sqrt{\kappa}\,\varepsilon_{\text{rel}}\,\|x_0 - x^*\|_A,\ (1/\sqrt{\alpha})\,\varepsilon_{\text{abs}}\bigr\} \\ \|x_k - x^*\|_M &\le \max\bigl\{\kappa\,\varepsilon_{\text{rel}}\,\|x_0 - x^*\|_M,\ (1/\alpha)\,\varepsilon_{\text{abs}}\bigr\} \end{aligned}\right\} \tag{4.23d}$$

Beweis. Aus dem Zusammenhang

$$A\,(x_k - x^*) = A\,x_k - b = r_k$$

zwischen Fehler und Residuum folgt

$$\|r_k\|_{M^{-1}}^2 = r_k^\top M^{-1} r_k = (x_k - x^*)^\top A\,M^{-1} A\,(x_k - x^*) = \|x_k - x^*\|_{A\,M^{-1}A}^2.$$

Beispielsweise (4.22a) kann also auch als

$$\|x_k - x^*\|_{A\,M^{-1}A} \le \varepsilon_{\text{rel}}\,\|x_0 - x^*\|_{A\,M^{-1}A}$$

geschrieben werden. Mit Hilfe der Ungleichungen (4.20) und (4.21) folgt (4.23a). Für die Abschätzungen (4.23b)–(4.23d) gehen wir analog vor.

Beachte: Bei der Verwendung einer rein relativen Abbruchbedingung (4.22a) zeigt sich, dass eine gute Vorkonditionierung (κ nahe 1) zwei Funktionen hat: Sie sorgt einerseits dafür, dass die Konvergenz der Fehlernorm $\|x_k - x^*\|_A$ hinreichend schnell ist, und andererseits dafür, dass man $\|x_k - x^*\|_A$ und $\|x_k - x^*\|_M$ durch die berechenbare Größe $\|r_k\|_{M^{-1}}$ zuverlässig abschätzen kann.
Bei der Verwendung absoluter Abbruchtoleranzen ist Vorsicht geboten, da diese über den kleinsten Eigenwert α des verallgemeinerten Eigenwertproblems $A\,x = \lambda\,M\,x$ von der absoluten Skalierung des Vorkonditionieres M abhängen. Die Verwendung von $\gamma\,M$ ($\gamma > 0$) an Stelle von M führt also zwar zu identischen Iterierten $\{x_k\}$ in Algorithmus 4.5, jedoch zu nicht-äquivalenten Abbruchbedingungen.

§ 4.2 Das CG-Verfahren für quadratische Zielfunktionen

Der typische Zick-Zack-Verlauf der vom Gradientenverfahren (Algorithmus 4.5) gewählten Suchrichtungen d_k zeigt die typische „Gedächtnislosigkeit“ des Gradientenverfahrens.[9] Hier setzt das **Verfahren der konjugierten Gradienten**[10] (**CG-Verfahren**, *conjugate gradient method*) an, indem es mit A**-konjugierten** (A-orthogonalen) Suchrichtungen arbeitet:

Definition 4.14 (Konjugierte Richtungen)**.**

Es sei $A \in \mathbb{R}^{n\times n}$ spd. Eine Menge von Vektoren $\{d_0, \ldots, d_k\} \subset \mathbb{R}^n$ heißt A**-konjugiert**, wenn alle $d_i \neq 0$ sind und wenn gilt:

$$d_i^\top A\, d_j = 0 \quad \text{für } 0 \leq i, j \leq k, \quad i \neq j.$$

Mit anderen Worten, A-konjugierte Vektoren stehen paarweise senkrecht im A-Skalarprodukt. Sie sind insbesondere linear unabhängig (Nachweis?).

Das CG-Verfahren ist ein Vertreter der Verfahren konjugierter Richtungen (*conjugate direction method*). Dabei werden die Suchrichtungen $d_0, d_1, \ldots$ (auf eine bestimmte Weise, siehe später) A-konjugiert gewählt, und die Iterierten erfüllen

$$x_{k+1} = x_k + \alpha_k\, d_k. \tag{4.24}$$

Die Schrittlänge α_k wird wie beim Gradientenverfahren so gewählt, dass die konvexe eindimensionale quadratische Funktion

$$\alpha \mapsto \phi(x_k + \alpha\, d_k)$$

exakt minimiert wird, also — vgl. (4.5) —

$$\alpha_k := -\frac{d_k^\top (A\, x_k - b)}{d_k^\top A\, d_k} = -\frac{d_k^\top r_k}{d_k^\top A\, d_k}. \tag{4.25}$$

Zur Abkürzung haben wir hier wieder das **Residuum** (*residual*) der Iterierten x_k eingeführt:

$$r_k := A\, x_k - b = \nabla\phi(x_k).$$

Die Residuen erfüllen wie gehabt die Rekursion

$$r_{k+1} = r_k + \alpha_k\, A\, d_k. \tag{4.26}$$

Verfahren, die mit konjugierten Richtungen arbeiten, haben folgende bemerkenswerte Eigenschaft, dass die nacheinander ausgeführte Minimierung eindimensionaler Probleme gleichbedeutend ist zur Minimierung über komplette Unterräume:

Lemma 4.15 (Eigenschaften von Verfahren konjugierter Richtungen)**.**

Es sei $x_0 \in \mathbb{R}^n$ sowie eine Menge $\{d_0, d_1, \ldots, d_k\}$ A-konjugierter Suchrichtungen gegeben, $k \geq 1$. Die Iterierten $x_1, \ldots, x_k$ seien mittels (4.24), (4.25) erzeugt. Dann gilt:

(a)
$$r_k^\top d_i = 0 \quad \text{für alle } i = 0, 1, \ldots, k-1, \tag{4.27}$$

[9] siehe auch Nocedal et al. (2002)

[10] Das CG-Verfahren wurde eingeführt von Hestenes and Stiefel (1952) als Verfahren zur Lösung von $A\, x = b$ mit spd Matrix A. Die Bezeichnung ist eigentlich irreführend, da nicht die Gradienten paarweise konjugiert sind, sondern die daraus erzeugten Suchrichtungen d_k.

(b) x_k minimiert ϕ über dem affinen Unterraum $x_0 + \operatorname{span}\{d_0, d_1, \ldots, d_{k-1}\}$.

Beweis. (a) wird mittels Induktion über k gezeigt. Induktionsanfang ($k = 1$):

$$r_1^\top d_0 = (A\,x_1 - b)^\top d_0 = (A\,x_0 + \alpha_0 A\,d_0 - b)^\top d_0 = r_0^\top d_0 + \alpha_0\, d_0^\top A\,d_0 = 0$$

aufgrund der Definition von α_0, siehe (4.25). Induktionsschritt: Wir setzen $r_{k-1}^\top d_i = 0$ für alle $i = 0, 1, \ldots, k-2$ voraus. Es gilt

$$\begin{aligned} r_k^\top d_{k-1} &= (r_{k-1} + \alpha_{k-1} A\,d_{k-1})^\top d_{k-1} && \text{wegen (4.26)} \\ &= 0 && \text{wegen der Definition von } \alpha_{k-1}\text{, siehe (4.25).} \end{aligned}$$

Für die anderen Suchrichtungen d_i, $i = 0, 1, \ldots, k-2$ gilt:

$$\begin{aligned} r_k^\top d_i &= (r_{k-1} + \alpha_{k-1} A\,d_{k-1})^\top d_i && \text{wegen (4.26)} \\ &= \underbrace{r_{k-1}^\top d_i}_{=0 \text{ nach Voraussetzung}} + \alpha_{k-1} \underbrace{d_{k-1}^\top A\,d_i}_{=0 \text{ wegen der } A\text{-Konjugiertheit}} = 0. \end{aligned}$$

Für Aussage (b) betrachten wir die Funktion $h : \mathbb{R}^k \to \mathbb{R}$

$$h(\sigma) := \phi\Big(x_0 + \sum_{j=0}^{k-1} \sigma_j\, d_j\Big).$$

h ist gleichmäßig konvex (Beweis?), und der eindeutige Minimierer ist durch

$$\frac{\partial h(\sigma^*)}{\partial \sigma_i} = \nabla\phi\Big(x_0 + \sum_{j=0}^{k-1} \sigma_j^*\, d_j\Big)^\top d_i = 0$$

für alle $i = 0, \ldots, k-1$ gekennzeichnet. Aufgrund von Teil (a) erfüllt aber gerade die Iterierte

$$x_k = x_0 + \sum_{j=0}^{k-1} \alpha_j\, d_j \in x_0 + \operatorname{span}\{d_0, d_1, \ldots, d_{k-1}\}$$

diese Beziehung:

$$\nabla\phi\Big(x_0 + \sum_{j=0}^{k-1} \alpha_j\, d_j\Big)^\top d_i = \nabla\phi(x_k)^\top d_i = r_k^\top d_i = 0$$

für alle $i = 0, \ldots, k-1$.

Folgerung 4.16 (Eigenschaften von Verfahren konjugierter Richtungen)**.**

Jedes Verfahren A-konjugierter Richtungen (definiert durch (4.24)–(4.25)) konvergiert in höchstens n Schritten gegen die eindeutige Lösung von (4.1).

Beweis. Da die Suchrichtungen $\{d_k\}$ A-konjugiert und damit insbesondere linear unabhängig sind, entspricht $\operatorname{span}\{d_0, d_1, \ldots, d_{n-1}\}$ dem ganzen $\mathbb{R}^n$, sodass x_n die Funktion ϕ über ganz $\mathbb{R}^n$ minimiert und x_n damit auch (4.1) löst.

Diese Aussage wird in der Praxis natürlich durch Rundungsfehler abgeschwächt und ist außerdem für hochdimensionale Aufgaben irrelevant.

Bei der Erzeugung der paarweise A-konjugierten Richtungen gibt es viele Möglichkeiten. Das CG-Verfahren bestimmt die Suchrichtung d_k aus der vorherigen Suchrichtung d_{k-1} und dem aktuellen negativen (vorkonditionierten) Residuum r_k (der Richtung des steilsten Abstiegs für ϕ an der Stelle x_k bzgl. des M-Skalarprodukts):

$$d_k := -M^{-1}r_k + \beta_k\, d_{k-1}, \quad \text{bzw.} \quad d_0 := -M^{-1}r_0. \tag{4.28}$$

Der Koeffizient β_k wird so bestimmt, dass zunächst zumindest d_k und d_{k-1} A-konjugiert sind:

$$\beta_k := \frac{r_k^\top M^{-1} A\, d_{k-1}}{d_{k-1}^\top A\, d_{k-1}}. \tag{4.29}$$

Dass das so beschriebene Verfahren tatsächlich A-konjugierte Suchrichtungen erzeugt, zeigt folgendes Lemma:

Lemma 4.17 (Eigenschaften der Iterierten im CG-Verfahren)**.**
Es sei $x_0 \in \mathbb{R}^n$ sowie die Suchrichtungen $\{d_0, d_1, \ldots, d_k\}$ und weiteren Iterierten durch (4.24)–(4.25), (4.28)–(4.29) erzeugt. Dann gilt für $k \geq 0$:

(a)

$$\begin{aligned} \operatorname{span}\{r_0, r_1, \ldots, r_k\} &= \operatorname{span}\{r_0, (A\,M^{-1})\, r_0, \ldots, (A\,M^{-1})^k\, r_0\} \\ \operatorname{span}\{d_0, d_1, \ldots, d_k\} &= M^{-1} \operatorname{span}\{r_0, (A\,M^{-1})\, r_0, \ldots, (A\,M^{-1})^k\, r_0\} \end{aligned} \tag{4.30}$$

(b)

$$r_k^\top M^{-1} r_i = 0 \quad \text{für alle } i = 0, 1, \ldots, k-1. \tag{4.31}$$

(c)

$$d_k^\top A\, d_i = 0 \quad \text{für alle } i = 0, 1, \ldots, k-1. \tag{4.32}$$

Der Raum

$$\mathcal{K}_{k+1}(A\,M^{-1}; r_0) := \operatorname{span}\{r_0, (A\,M^{-1})\, r_0, \ldots, (A\,M^{-1})^k\, r_0\} \tag{4.33}$$

wird als **Krylov-Unterraum** (der Dimension $k+1$) zur Matrix $A\,M^{-1}$ mit Startvektor r_0 bezeichnet. Das CG-Verfahren ist damit ein Vertreter der Krylov-Unterraum-Verfahren (*Krylov subspace methods*). Die Eigenschaften (4.31) und (4.32) besagen gerade, dass das Verfahren gleichzeitig eine expandierende Folge von M^{-1}-orthogonalen Basisvektoren der Räume $\mathcal{K}_k(A\,M^{-1}; r_0)$ wie auch eine expandierende Folge von A-orthogonalen Basisvektoren der Räume $M^{-1}\mathcal{K}_k(A\,M^{-1}; r_0)$ erzeugt.

Beweis. Der Beweis erfolgt durch Induktion und wird hier nicht angegeben, siehe z. B. (Nocedal and Wright, 2006, p.109) für den Fall ohne Vorkonditionierung.

Mit Hilfe der gezeigten Eigenschaften der Iterierten lassen sich die Gleichungen (4.25) für α_k und (4.29) für β_k beim CG-Verfahren noch umschreiben als

$$\alpha_k \overset{(4.25)}{=} -\frac{d_k{}^\top r_k}{d_k^\top A\, d_k} \overset{(4.28)}{=} \frac{r_k^\top M^{-1} r_k}{d_k^\top A\, d_k} - \beta_k \frac{d_{k-1}^\top r_k}{d_k^\top A\, d_k} \overset{(4.27)}{=} \frac{r_k^\top M^{-1} r_k}{d_k^\top A\, d_k} \tag{4.25'}$$

und

$$\beta_{k+1} \overset{(4.29)}{=} \frac{r_{k+1}^\top M^{-1} A\, d_k}{d_k^\top A\, d_k} \overset{(4.26)}{=} \frac{r_{k+1}^\top M^{-1}(r_{k+1}-r_k)}{d_k{}^\top (r_{k+1}-r_k)}$$
$$\overset{(4.28)}{=} \frac{r_{k+1}^\top M^{-1}(r_{k+1}-r_k)}{(-M^{-1}r_k+\beta_k\, d_{k-1})^\top (r_{k+1}-r_k)} \overset{(4.27),(4.31)}{=} \frac{r_{k+1}^\top M^{-1} r_{k+1}}{r_k^\top M^{-1} r_k}. \qquad (4.29')$$

Diese Beziehungen gelten auch für den Sonderfall $k = 0$ in (4.28).

Wir erhalten somit die übliche Form des (vorkonditionierten) CG-Verfahrens:[11]:

Algorithmus 4.18 (CG-Verfahren bei quadratischer Zielfunktion)**.**
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: rechte Seite $b \in \mathbb{R}^n$
Eingabe: spd Matrix A (oder Matrix-Vektor-Produkte mit A)
Eingabe: spd Matrix M (oder Matrix-Vektor-Produkte mit M^{-1})
Ausgabe: näherungsweise Lösung der Aufgabe (4.1) bzw. von $A\,x = b$

1: Setze $k := 0$
2: Setze $r_0 := A\,x_0 - b$
3: Setze $d_0 := -M^{-1} r_0$
4: Setze $\delta_0 := -r_0^\top d_0$ $\{\delta_0 = \|\nabla_M \phi(x_0)\|_M^2\}$
5: **while** Abbruchkriterium nicht erfüllt **do**
6: Setze $q_k := A\,d_k$
7: Setze $\alpha_k := \delta_k / (d_k^\top q_k)$
8: Setze $x_{k+1} := x_k + \alpha_k\, d_k$
9: Setze $r_{k+1} := r_k + \alpha_k\, q_k$
10: Setze $d_{k+1} := -M^{-1} r_{k+1}$
11: Setze $\delta_{k+1} := -r_{k+1}^\top d_{k+1}$ $\{\delta_{k+1} = \|\nabla_M \phi(x_{k+1})\|_M^2\}$
12: Setze $\beta_{k+1} := \delta_{k+1}/\delta_k$
13: Setze $d_{k+1} := d_{k+1} + \beta_{k+1}\, d_k$
14: Setze $k := k + 1$
15: **end while**
16: **return** x_k

Bemerkung 4.19 (zum CG-Verfahren bei quadratischer Zielfunktion)**.**

(a) Das CG-Verfahren ist von der Implementierung sehr ähnlich wie das Gradientenverfahren (Algorithmus 4.5). Der einzige (aber wesentliche!) Unterschied besteht in der Modifikation der Gradientenrichtung in den zusätzlichen Zeilen 12 und 13.

(b) Die Bemerkung 4.6 gilt wörtlich auch für das CG-Verfahren. Es muss jedoch ein Vektor mehr gespeichert werden, da man d_k und d_{k+1} gleichzeitig benötigt. Außerdem ist die Abschätzung (4.9) nicht mehr gültig.

(c) Die Überlegungen zu Abbruchbedingungen, siehe (4.22)–(4.23), gelten identisch auch für das CG-Verfahren, da sie auf derselben berechenbaren Größe $\|r_k\|_{M^{-1}}$ beruhen.

[11] Implementierung und Tests in Übung 2, Aufgabe 7

(d) **Achtung:** Das CG-Verfahren `pcg` von MATLAB ist hier inkonsequent, da das Residuum auch bei Verwendung eines Vorkonditionierers immer in der Euklidischen Norm gemessen wird. Es wird eine Abbruchbedingung vom Typ (4.22a) verwendet, jedoch mit $M = I$. Damit verliert man die Möglichkeit, die gewünschte Lösungsgenauigkeit in einer problemangepassten Norm anzugeben.

Unser nächstes Ziel ist wieder ein Konvergenzresultat. Anders als in Satz 4.10 bekommen wir aber kein aussagekräftiges Resultat für die Reduktion des Fehlers von Iteration zu Iteration, sondern nur eine Aussage für die Reduktion gegenüber dem Anfangsfehler. Ein Beweis findet sich beispielsweise in (Elman et al., 2014, Theorem 2.4) und weitere Aussagen in (Nocedal and Wright, 2006, Kapitel 5.1).

Satz 4.20 (Konvergenzsatz des CG-Verfahrens für (4.1))**.**

Es seien $A \in \mathbb{R}^{n\times n}$ und $M \in \mathbb{R}^{n\times n}$ beide spd. Weiter seien $0 < \alpha \leq \beta$ die extremalen Eigenwerte von (4.16). Dann gilt: Das CG-Verfahren Algorithmus 4.18 konvergiert für jeden beliebigen Startwert $x_0 \in \mathbb{R}^n$ gegen die eindeutige Lösung $x^* = A^{-1}b$ von (4.1), und es gilt mit κ aus (4.17):

$$\phi(x_k) - \phi(x^*) \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2k} \big(\phi(x_0) - \phi(x^*)\big) \tag{4.34a}$$

$$\|x_k - x^*\|_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{k} \|x_0 - x^*\|_A, \tag{4.34b}$$

Weiterhin ist die Folge $\{\phi(x_k)\}$ und damit die Fehlernorm $\|x_k - x^*\|_A$ monoton fallend.

Beachte: Ein Konvergenzresultat vom Typ (4.34b) wird auch als r-lineare Konvergenz[12] bezeichnet, weil für den „durchschnittlichen“ Reduktionsfaktor für den Fehler in der A-Norm gilt:

$$\limsup_{k\to\infty} \left(\frac{\|x_k - x^*\|_A}{\|x_0 - x^*\|_A}\right)^{1/k} \leq \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} < 1.$$

Zum Vergleich: Beim Gradientenverfahren ist dieser Faktor $\dfrac{\kappa-1}{\kappa+1}$ und damit wesentlich ungünstiger (näher bei 1).[13]

Bemerkung 4.21 (zum Vergleich des Gradienten- und des CG-Verfahrens)**.** Das Gradientenverfahren in der Form von Algorithmus 4.5 ist dem CG-Verfahren im Konvergenzverhalten unterlegen. Aus Übung 2, Aufgabe 6 wissen wir beispielsweise, dass die erwartete Iterationszahl des Gradientenverfahrens, um den Fehler um einen Faktor ε_{rel} zu reduzieren, proportional zur Konditionszahl κ ist. Dagegen ist sie beim CG-Verfahren nur proportional zu $\sqrt{\kappa}$.

Diese Aussage gilt jedoch nur bei Verwendung der exakten Schrittweite (4.5) — manchmal auch **Cauchy-Schrittweite** genannt —, wie wir dies in Algorithmus 4.5

[12] Das „R“ steht hier für *root*, also Wurzel.

[13] In Übung 2, Aufgabe 6 schätzen wir für verschiedene Konditionszahlen κ die benötigtigten Iterationszahlen für das CG- und das Gradientenverfahren ab, um eine vorgegebene Fehlerreduktion zu erzielen.

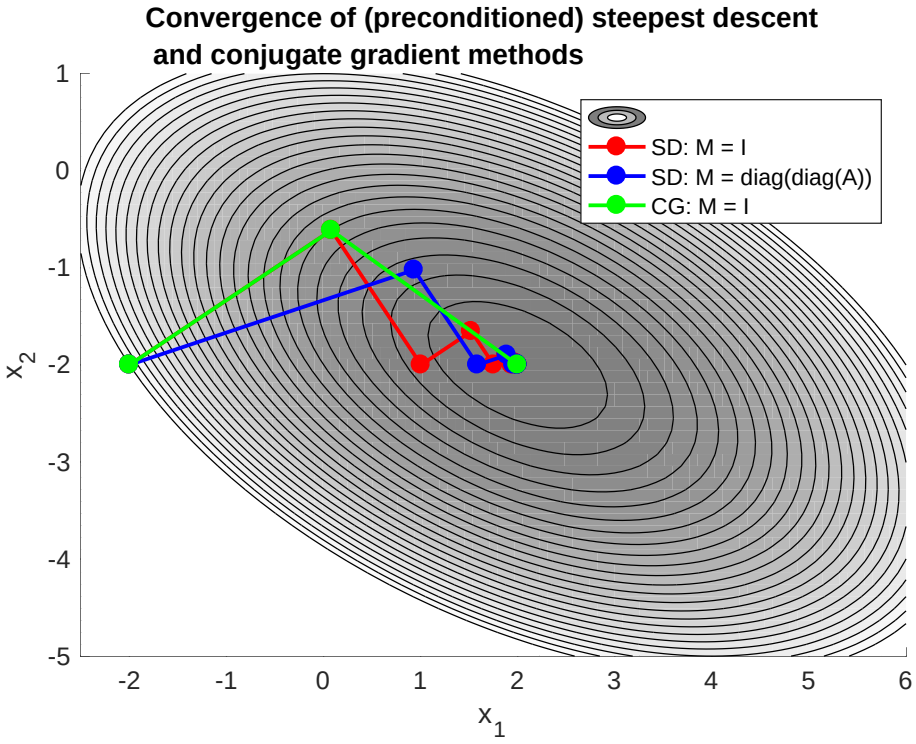

Abbildung 4.2. Vergleich des Konvergenzverhaltens des Gradientenverfahrens (Daten wie in Abbildung 4.1) und zusätzlich des CG-Verfahrens (ohne Vorkonditionierung, $M = I$)

getan haben. Es lassen sich jedoch wesentlich bessere Schrittweiten α_k für das Gradientenverfahren angeben, siehe etwa Barzilai and Borwein (1988). Bei Kenntnis (oder einer brauchbaren Abschätzung) von $\lambda_{\min}(A; M)$ und $\lambda_{\max}(A; M)$ kann man mit den richtigen Schrittweiten sogar dieselbe(!) Komplexität (Iterationszahl $\sim \sqrt{\kappa}$) wie beim CG-Verfahren erhalten, siehe Gonzaga (2016).

Die Iterierten des CG-Verfahrens haben noch eine weitere bemerkenswerte Eigenschaft, die wir später ausnutzen werden:

Lemma 4.22 (Wachstum des Abstands vom Startpunkt[14])**.**
Die Folge $\{\|x_k - x_0\|_M\}$ ist streng monoton wachsend, solange $x_k \neq x^*$ gilt.

Für das Gradientenverfahren gilt diese Eigenschaft nicht.

Beweis. Aus Lemma 4.15 (a) folgt zunächst

$$r_k^\top (x_k - x_0) = \sum_{i=0}^{k-1} \alpha_i \underbrace{r_k^\top d_i}_{=0} = 0 \quad \text{für alle } k \geq 0. \tag{$*$}$$

[14] In der Literatur findet man dieses Resultat häufig nur für den Fall $x_0 = 0$ angegeben, siehe etwa (Nocedal and Wright, 2006, Theorem 7.3).

Wir zeigen nun per Induktion, dass $(x_k - x_0)^\top M\, d_k > 0$ für $k \geq 1$ ist. Zunächst gilt für $k = 1$ wiederum wegen Lemma 4.15 (a):

$$\begin{aligned}(x_1 - x_0)^\top M\, d_1 &= \alpha_0\, d_0^\top M\, (-M^{-1} r_1 + \beta_1\, d_0) \\ &= \underbrace{\alpha_0}_{>0}\, \underbrace{\beta_1}_{>0}\, \underbrace{d_0^\top M\, d_0}_{>0} > 0.\end{aligned}$$

Wir machen nun den Schritt von k auf $k+1$:

$$\begin{aligned}(x_{k+1} - x_0)^\top M\, d_{k+1} &= (x_{k+1} - x_0)^\top M\, (-M^{-1} r_{k+1} + \beta_{k+1}\, d_k) \\ &\overset{(*)}{=} \beta_{k+1}\, (x_{k+1} - x_0)^\top M\, d_k \\ &= \beta_{k+1}\, (x_k + \alpha_k\, d_k - x_0)^\top M\, d_k \\ &= \beta_{k+1}\, (x_k - x_0)^\top M\, d_k + \alpha_k\, \beta_{k+1}\, d_k^\top M\, d_k > 0. \qquad (**)\end{aligned}$$

Wegen der Induktionsvoraussetzung sowie $\alpha_k > 0$, $\beta_{k+1} > 0$ und $d_k^\top M\, d_k > 0$ ist der gesamte Ausdruck positiv.

Die eigentliche Aussage folgt nun leicht aus

$$\begin{aligned}\|x_{k+1} - x_0\|_M^2 &= \|x_k + \alpha_k\, d_k - x_0\|_M^2 \\ &= \|x_k - x_0\|_M^2 + 2\, \underbrace{\alpha_k}_{>0}\, \underbrace{(x_k - x_0)^\top M\, d_k}_{>0} + \underbrace{\alpha_k^2\, \|d_k\|_M^2}_{>0}. \qquad (***)\end{aligned}$$

Die Beziehungen $(**)$ und $(***)$ erlauben es, die informativen Größen

$$\eta_k := \|x_k - x_0\|_M^2 \tag{4.35a}$$

$$\zeta_k := (x_k - x_0)^\top M\, d_k \tag{4.35b}$$

$$\gamma_k := \|d_k\|_M^2 \tag{4.35c}$$

im CG-Verfahren rekursiv ohne nennenswerten Zusatzaufwand mitzuführen, indem an geeigneter Stelle in Algorithmus 4.18 die Beziehungen

$$\eta_0 := 0, \quad \eta_{k+1} := \eta_k + 2\,\alpha_k\,\zeta_k + \alpha_k^2\,\gamma_k \quad \text{siehe } (***) \tag{4.36a}$$

$$\zeta_0 := 0, \quad \zeta_{k+1} := \beta_{k+1}\,(\zeta_k + \alpha_k\,\gamma_k) \quad \text{siehe } (**) \tag{4.36b}$$

$$\gamma_0 := \delta_0, \quad \gamma_{k+1} := \delta_{k+1} + \beta_{k+1}^2\,\gamma_k \quad \text{(nachrechnen)} \tag{4.36c}$$

eingefügt werden. Das Bemerkenswerte daran ist, dass dies gelingt, ohne dass die Matrix M (oder Matrix-Vektor-Produkte mit M) vorliegen muss, da man im Verfahren i. A. nur Matrix-Vektor-Produkte mit M^{-1} zur Verfügung hat.

Wir werden das CG-Verfahren und Variationen davon im weiteren Verlauf von Kapitel 1 noch mehrfach antreffen, nämlich

- als Löser innerhalb eines Line-Search-Newton-Verfahrens (truncated Newton-CG), siehe § 5.5.1,
- als Löser innerhalb eines Trust-Region-Newton-Verfahrens (Steihaug-CG), siehe § 6.3
- sowie als eigenständiges Verfahren in *einer* nichtlinearen Variante auch für nicht-quadratische Zielfunktionen, siehe § 5.6.

Ende 5. V

§ 5 Liniensuchverfahren

Literatur: (Ulbrich and Ulbrich, 2012, Kapitel 7–13)

In diesem Abschnitt betrachten wir eine große Klasse von Verfahren zur Lösung der nichtlinearen unrestringierten Aufgabe

$$\text{Minimiere} \quad f(x) \quad \text{über } x \in \mathbb{R}^n, \tag{FO}$$

sogenannte **Liniensuch-Verfahren** (*line search methods*). In jeder Iteration wird zunächst eine **Suchrichtung** (*search direction*) d_k bestimmt und anschließend eine **Schrittlänge** (*step size*) α_k, die zur nächsten Iterierten x_{k+1} führt:

$$x_{k+1} := x_k + \alpha_k\, d_k.$$

Voraussetzung 5.1. Im gesamten § 5 sei $f : \mathbb{R}^n \to \mathbb{R}$ überall stetig differenzierbar (eine C^1-Funktion).

Die meisten (und auch die hier betrachteten) Line-Search-Verfahren verlangen, dass d_k eine **Abstiegsrichtung** für f an der Stelle x_k ist, also dass gilt:

$$\underbrace{f'(x_k)\, d_k = \nabla f(x_k)^\top d_k}_{\text{Richtungsableitung von } f \text{ im Punkt } x_k \text{ in Richtung } d_k} < 0. \tag{5.1}$$

Dies bedeutet, dass die Zielfunktion entlang der Suchrichtung $d_k \neq 0$ zunächst streng monoton fallend ist, sodass zumindest für hinreichend kleine positive Schrittlängen α_k gilt: $f(x_{k+1}) < f(x_k)$. Deshalb spricht man auch von **Abstiegsverfahren**.

Alle hier besprochenen Verfahren bestimmen die Suchrichtung d_k, indem sie ein lokales **quadratisches Modell** der Zielfunktion betrachten:[15]

$$q_k(d) = f(x_k) + f'(x_k)\, d + \frac{1}{2}\, d^\top H_k\, d. \tag{5.2}$$

In dieses Modell gehen u. a. die Daten $f(x_k)$ und $f'(x_k)$ an der Stelle x_k ein.[16] Im Fall $H_k = \nabla^2 f(x_k)$ ist (5.2) gerade das Taylorpolynom 2. Ordnung von f an der Stelle x_k, aber i. A. kann H_k auch eine andere (o. B. d. A. symmetrische) Matrix sein. Die Wahl von H_k, also der Hessematrix des Modells (5.2), ist sogar ein wesentliches Unterscheidungsmerkmal verschiedener Liniensuchverfahren, wie wir sehen werden.

Die Suchrichtung d_k bestimmt sich aus der (möglicherweise inexakten) Lösung der quadratischen Aufgabe

$$\text{Minimiere} \quad q_k(d), \quad d \in \mathbb{R}^n. \tag{5.3}$$

Wie wir aus Lemma 4.1 wissen, gibt es folgende Fälle:

(a) H_k ist positiv definit, dann ist die eindeutige Lösung von (5.3) gegeben durch die eindeutige Lösung des LGS

$$H_k\, d_k = -\nabla f(x_k). \tag{5.4}$$

(b) H_k ist positiv semidefinit, dann stimmen die Lösungen von (5.3) überein mit der Lösungsmenge des LGS (5.4).[17]

[15] **Ausnahme:** Nichtlineare CG-Verfahren motivieren die Bestimmung der Suchrichtung anders.

[16] Man sagt, das Modell (5.2) sei **funktionswerttreu** und **ableitungstreu**, da $q_k(0) = f(x_k)$ und $\nabla q_k(0) = \nabla f(x_k)$ gilt.

[17] Dies ist entweder die leere Menge oder ein affiner Unterraum des $\mathbb{R}^n$ der Dimension $\ker H_k$.

(c) H_k ist nicht positiv semidefinit (besitzt mindestens einen negativen Eigenwert), dann ist (5.3) unbeschränkt. Das LGS (5.4) kann dennoch eindeutig lösbar sein oder aber mehrdeutig lösbar oder nicht lösbar. (Die Lösungen sind dann entweder alles Sattelpunkte von q_k oder alles globale Maxima. Warum?)

Zur Lösung von (5.3) bzw. (5.4) kann das CG-Verfahren aus § 4 bzw. Varianten davon verwendet werden, die auf das Auftreten ggf. nicht-positiver Eigenwerte angemessen reagieren. Dazu später mehr.

§ 5.1 Ein allgemeines Abstiegsverfahren

Wir betrachten zunächst folgenden Modellalgorithmus eines Abstiegsverfahrens:

Algorithmus 5.2 (allgemeines Abstiegsverfahren mit Liniensuche)**.**
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: Funktionsauswertungen f und f' bzw. ∇f
Ausgabe: näherungsweise stationärer Punkt der Aufgabe (**FO**)

1: Setze $k := 0$
2: **while** Abbruchkriterium nicht erfüllt **do**
3: Bestimme eine Suchrichtung d_k mit $f'(x_k)\, d_k < 0$
4: Wähle eine Schrittweite $\alpha_k > 0$ mit $f(x_k + \alpha_k\, d_k) < f(x_k)$
5: Setze $x_{k+1} := x_k + \alpha_k\, d_k$
6: Setze $k := k + 1$
7: **end while**
8: **return** x_k

Zur Untersuchung der theoretischen Konvergenzeigenschaften des Verfahrens gehen wir von dem Fall aus, dass Algorithmus 5.2 unendliche Folgen $\{x_k\}$, $\{d_k\}$, $\{\alpha_k\}$ erzeugt, also nicht vorzeitig abbricht.[18]

Mit der Bedingung aus Lemma 3.1 können wir zumindest die *Existenz* eines Häufungspunktes des Folge $\{x_k\}$ sichern:

Lemma 5.3. Für die Startiterierte x_0 sei die Sublevelmenge

$$\mathcal{M}_f(x_0) = \{x \in \mathbb{R}^n : f(x) \le f(x_0)\}$$

kompakt. Dann hat die von Algorithmus 5.2 erzeugte Folge $\{x_k\}$ einen Häufungspunkt.

Beweis. Wegen der Forderung $f(x_{k+1}) < f(x_k)$ in Algorithmus 5.2 liegen alle Iterierten in der kompakten Menge $\mathcal{M}_f(x_0)$. Damit hat die Folge einen Häufungspunkt.

Die Häufungspunkte der Folge der Iterierten $\{x_k\}$ sollten ausgezeichnete Punkte sein, daher wäre die Aussage

$$\text{Ist } x^* \text{ ein Häufungspunkt der Folge } \{x_k\}, \text{ dann gilt } \nabla f(x^*) = 0. \tag{5.5}$$

wünschenswert. Mehr kann man i. A. nicht erwarten.

[18] Insbesondere nehmen wir an, dass kein x_k ein stationärer Punkt ist, sonst gibt es ja keine Abstiegsrichtung d_k mehr. In der Praxis verwendet man natürlich andere Abbruchbedingungen als $\nabla f(x_k) = 0$, ist also etwa mit einem „fast stationären Punkt“ zufrieden.

Die (relativ schwache) Eigenschaft (5.5) bezeichnet man oft als die **globale Konvergenz** des Verfahrens.[19] Wir stellen jetzt die Frage, unter welchen Minimalanforderungen an die Wahl der Suchrichtungen d_k und Schrittweiten α_k der Algorithmus 5.2 global konvergent ist. Dazu sind zwei Eigenschaften entscheidend:

(1) Die Suchrichtungen d_k sind „gute Abstiegsrichtungen“.

(2) Die Wahl der Schrittweiten α_k erfolgt so, dass von dem entlang der gegebenen Suchrichtung d_k möglichen Abstieg „nicht zuviel verschenkt“ wird.

Wir betrachen zuerst die Anforderungen an die Suchrichtungen. Der Raum $\mathbb{R}^n$ der Optimierungsvariablen und damit auch der Suchrichtungen sei dazu wieder mit dem (benutzerdefinierten) M-Skalarprodukt ausgestattet. Zwar sind alle Normen im $\mathbb{R}^n$ zueinander äquivalent und damit alle Begriffe und Eigenschaften von Verfahren im Rest von § 5 *qualitativ unabhängig* von der konkreten Wahl von M, aber dennoch spielt M eine wichtige Rolle, weil es das Konvergenzverhalten und die Abbruchbedingungen beeinflusst.

Definition 5.4 (Zulässige Suchrichtungen)**.** Die Folge $\{d_k\}$ von Suchrichtungen heißt **zulässig**, falls

$$\frac{f'(x_k)\, d_k}{\|d_k\|_M} \to 0 \quad \Rightarrow \quad f'(x_k) \to 0. \tag{5.6}$$

Beachte: Zulässigkeit ist eine Eigenschaft, die eine von einem *bestimmten* Algorithmus erzeugte Folge von Suchrichtungen, angewendet auf ein *bestimmtes* Problem (Zielfunktion f) und gestartet mit einem *bestimmten* Startwert x_0 besitzt oder eben nicht besitzt. Natürlich ist man daran interessiert, Algorithmen anzugeben, deren Suchrichtungen (unter bestimmten Voraussetzungen) für *beliebiges* f und *beliebigen* Startwert x_0 Folgen zulässiger Schrittweiten konstruieren. Aufgrund der Äquivalenz aller Normen ist (5.6) unabhängig von der verwendeten Norm $\|\cdot\|_M$.

Der Ausdruck $f'(x_k)\, d_k/\|d_k\|_M$ ist der (normalisierte) Anstieg von f in Richtung d_k. Die Forderung (5.6) heißt also: Gehen die Anstiege in Richtung der d_k gegen 0, so kann das nur daran liegen, dass die Ableitung selbst gegen null geht. Grob gesagt: Eine Folge zulässiger Richtungen muss also genug Abstieg bieten, falls welcher vorhanden ist.

Oft wird die Konvergenz von Liniensuchverfahren unter einer Winkelbedingung

$$\cos \measuredangle(-\nabla_M f(x_k), d_k) = \frac{-f'(x_k)\, d_k}{\|\nabla_M f(x_k)\|_M \, \|d_k\|_M} = \frac{(-\nabla_M f(x_k),\, d_k)_M}{\|\nabla_M f(x_k)\|_M \, \|d_k\|_M} \geq \eta \tag{5.7}$$

mit $\eta \in (0, 1)$ gezeigt. Dies bedeutet gerade, dass die Suchrichtungen d_k einen gleichmäßig spitzen Winkel mit der jeweils aktuellen Richtung des steilsten Abstiegs $-\nabla_M f(x_k)$ bilden müssen.

Lemma 5.5 (Winkelbedingung impliziert Zulässigkeit der Suchrichtungen)**.**

Für die Folge $\{d_k\}$ gelte die Winkelbedingung (5.7) mit einem $\eta \in (0, 1)$ Dann ist diese Folge von Suchrichtungen zulässig.

[19] Globale Konvergenz bedeutet also insbesondere nicht, dass man ein globales Minimum findet, sondern dass man Konvergenzaussagen für alle Startwerte bekommt. Unter zusätzlichen Annahmen kann man mehr zeigen, etwa die Eindeutigkeit des Häufungspunktes, Konvergenz der Folge gegen ein lokales Minimum usw.

Beweis. Wegen

$$f'(x_k)\,d_k = (\nabla f(x_k),\, d_k) = (\nabla_M f(x_k),\, d_k)_M$$

folgt aus (5.7):

$$-\frac{f'(x_k)\,d_k}{\|d_k\|_M} \geq \eta\,\|\nabla_M f(x_k)\|_M = \eta\,\|f'(x_k)^\top\|_{M^{-1}} \geq 0.$$

Daraus folgt (5.6).

Wie bereits bemerkt, bestimmen die später im Detail besprochenen Verfahren ihre Suchrichtung aus

$$H_k\,d_k = -\nabla f(x_k) \tag{5.4}$$

mit einer spd Matrix H_k (der Hessematrix des Modells (5.2)). Wegen

$$f'(x_k)\,d_k = (\nabla f(x_k),\, -H_k^{-1}\,\nabla f(x_k)) = -\nabla f(x_k)^\top\,H_k^{-1}\,\nabla f(x_k) < 0 \tag{5.8}$$

ist dies eine Abstiegsrichtung, solange $f'(x_k) \neq 0$ ist.

Falls die Hessematrizen „vernünftig“ bleiben, so erfüllt die Folge so erzeugter Suchrichtungen die Winkelbedingung:

Lemma 5.6 (Beschränkte Konditionszahlen implizieren Winkelbedingung[20])**.**
Es sei $\{H_k\} \in \mathbb{R}^{n\times n}$ eine Folge von spd Matrizen sowie $M \in \mathbb{R}^{n\times n}$ spd. Für die verallgemeinerten Konditionszahlen der H_k gelte

$$\kappa(H_k; M) := \frac{\lambda_{\max}(H_k; M)}{\lambda_{\min}(H_k; M)} \leq \kappa.$$

Dann erfüllt die gemäß (5.4) erzeugte Folge $\{d_k\}$ die Winkelbedingung (5.7) mit

$$\eta = \frac{2\sqrt{\kappa}}{\kappa+1} \geq \frac{1}{\sqrt{\kappa}}.$$

Beweis. Zu zeigen ist:

$$\begin{aligned}
& -\nabla f(x_k)^\top d_k \geq \frac{2\sqrt{\kappa}}{\kappa+1}\|\nabla_M f(x_k)\|_M\,\|d_k\|_M \\
\Leftrightarrow\quad & d_k^\top H_k\,d_k \geq \frac{2\sqrt{\kappa}}{\kappa+1}\|M^{-1}H_k\,d_k\|_M\,\|d_k\|_M \qquad \text{denn } H_k\,d_k = -\nabla f(x_k) \\
\Leftrightarrow\quad & (d_k^\top H_k\,d_k)^2 \geq \frac{4\,\kappa}{(\kappa+1)^2}\|M^{-1}H_k\,d_k\|_M^2\,\|d_k\|_M^2 \\
\Leftrightarrow\quad & \frac{(d_k^\top H_k\,M^{-1}H_k\,d_k)\,(d_k^\top M\,d_k)}{(d_k^\top H_k\,d_k)^2} \leq \frac{(\kappa+1)^2}{4\,\kappa}.
\end{aligned}$$

Dies ist aber gerade die Aussage der verallgemeinerten Kantorovich-Ungleichung (4.18). Diese ist anwendbar, da d_k nach Voraussetzung eine Abstiegsrichtung und damit insbesondere $\neq 0$ ist.

[20] In der Literatur findet man dieses Resultat oft nur für den Fall $M = I$ und mit der nicht so scharfen Schranke $\eta = \frac{1}{\kappa}$; siehe z. B. (Ulbrich and Ulbrich, 2012, S.32) oder (Nocedal and Wright, 2006, eq.(3.19)).

Als Nächstes geht es um geeignete Schrittweiten α_k. Dass es nicht ausreicht, bei der Wahl der Schrittweiten einfach nur

$$f(x_k + \alpha_k\, d_k) < f(x_k)$$

zu fordern, sodass $\{f(x_k)\}$ streng monoton fällt, zeigt folgendes Beispiel:[21]

Beispiel 5.7 (zu kleine Schrittweiten)**.** Es sei $f(x) = x^2$, $x_0 = 1$ und $d_k = -1$ sowie die Schrittweiten $\alpha_k = \left(\frac{1}{2}\right)^{k+2}$. Dann ist

$$x_{k+1} = x_k + \alpha_k\,(-1) = x_0 - \sum_{i=0}^{k} \left(\tfrac{1}{2}\right)^{i+2} = \tfrac{1}{2} + \left(\tfrac{1}{2}\right)^{k+2}.$$

Daraus folgt $x_{k+1} < x_k$ und $f(x_{k+1}) < f(x_k)$. Jedoch konvergiert $x_k \to x^* = 1/2$, also nicht gegen das Minimum von f bei $x = 0$.

Die folgende allgemeine Bedingung sichert — zusammen mit der Zulässigkeit der Suchrichtungen — bereits die globale Konvergenz des Modellverfahrens Algorithmus 5.2:

Definition 5.8 (Zulässige Schrittweiten)**.** Die Folge $\{\alpha_k\}$ von Schrittweiten heißt **zulässig**, falls gilt:

$$f(x_k + \alpha_k\, d_k) \le f(x_k) \quad \text{für alle } k \in \mathbb{N}_0 \tag{5.9a}$$

$$f(x_k + \alpha_k\, d_k) - f(x_k) \to 0 \quad \Rightarrow \quad \frac{f'(x_k)\, d_k}{\|d_k\|_M} \to 0. \tag{5.9b}$$

Anschaulich heißt dies, dass der Fortschritt $f(x_{k+1}) - f(x_k)$ nur dann klein werden darf, wenn der normalisierte Abstieg von f in die Suchrichtungen d_k klein wird. Zulässige Schrittweiten machen also hinreichenden Gebrauch von dem möglichen Abstieg, den die Suchrichtungen d_k bieten.

Eine Möglichkeit, diese Zulässigkeit zu sichern, sind *effiziente* Schrittweiten.

Definition 5.9 (Effiziente Schrittweiten)**.** Die Schrittweiten $\{\alpha_k\}$ heißen **effizient**, wenn ein $\theta > 0$ existiert, sodass für alle $k \in \mathbb{N}_0$ gilt:

$$f(x_k + \alpha_k\, d_k) \le f(x_k) - \theta \left(\frac{f'(x_k)\, d_k}{\|d_k\|_M} \right)^2. \tag{5.10}$$

Ist die Ableitung f' Lipschitz-stetig auf $\mathcal{M}_f(x_0)$, so kann man garantieren, dass eine effiziente Schrittweite existiert.

Lemma 5.10 (Effizienz impliziert Zulässigkeit der Schrittweiten)**.**
Die Schrittweiten $\{\alpha_k\}$ seien effizient. Dann sind sie auch zulässig.

Beweis. Es sei $\{\alpha_k\}$ eine Folge von effizienten Schrittweiten. Dann ist (5.9a) klar. Es gelte nun

$$f(x_k + \alpha_k\, d_k) - f(x_k) \to 0.$$

[21] aus (Alt, 2002, Beispiel 4.4.1)

Aus der Effizienz folgt

$$0 \leq \theta \left(\frac{f'(x_k)\, d_k}{\|d_k\|} \right)^2 \leq f(x_k) - f(x_k + \alpha_k\, d_k)$$

für alle $k \in \mathbb{N}_0$. Da die rechte Seite eine Nullfolge ist und $\theta > 0$ gilt, folgt

$$\frac{f'(x_k)\, d_k}{\|d_k\|} \to 0.$$

Damit ist die Folge $\{\alpha_k\}$ zulässig.

Mit diesen Bedingungen an die Schrittweiten und die Suchrichtungen erhalten wir gleich einen Satz über die globale Konvergenz von Algorithmus 5.2.

Beachte: Nach der zu erwartenden Konvergenzaussage (5.5) müssen wir mit Häufungspunkten (Grenzwerten von *Teilfolgen*) arbeiten. Daher müssen Definition 5.4 (zulässige Suchrichtungen), Definition 5.8 (zulässige Schrittweiten) und Definition 5.9 (effiziente Schrittweiten) auf Teilfolgen $\{\cdot\}_{k \in K}$ (kurz: $\{\cdot\}_K$) erweitert werden:

$$\left\{ \frac{f'(x_k)\, d_k}{\|d_k\|_M} \right\}_K \to 0 \quad \Rightarrow \quad \{f'(x_k)\}_K \to 0. \tag{5.6}$$

$$f(x_k + \alpha_k\, d_k) \leq f(x_k) \quad \text{für alle } k \in \mathbb{N}_0 \tag{5.9a}$$

$$f(x_k + \alpha_k\, d_k) - f(x_k) \to 0 \quad \Rightarrow \quad \left\{ \frac{f'(x_k)\, d_k}{\|d_k\|_M} \right\}_K \to 0 \tag{5.9b}$$

$$f(x_k + \alpha_k\, d_k) \leq f(x_k) - \theta \left(\frac{f'(x_k)\, d_k}{\|d_k\|} \right)^2 \quad \text{für alle } k \in K. \tag{5.10}$$

Die Aussagen von Lemma 5.5, Lemma 5.6 und Lemma 5.10 gelten weiter, wenn die gesamte Folge durch eine Teilfolge ersetzt wird.

Satz 5.11 (Globaler Konvergenzsatz)**.** Es sei x^* ein Häufungspunkt der Iterierten $\{x_k\}$ von Algorithmus 5.2 und $\{x_k\}_K$ eine gegen x^* konvergente Teilfolge. Die Teilfolgen von Suchrichtungen $\{d_k\}_K$ und Schrittweiten $\{\alpha_k\}_K$ seien zulässig. Dann gilt $f'(x^*) = 0$.

Beweis. Wegen der Stetigkeit von f gilt $\{f(x_k)\}_K \to f(x^*)$. Da zusätzlich die komplette Folge $\{f(x_k)\}$ monoton fallend ist, muss sie damit insgesamt konvergieren. Sie ist also eine Cauchy-Folge und es gilt

$$f(x_{k+1}) - f(x_k) = f(x_k + \alpha_k\, d_k) - f(x_k) \to 0.$$

Aus der Zulässigkeit der Schrittweiten (5.9) folgt damit insbesondere

$$\left\{ \frac{f'(x_k)\, d_k}{\|d_k\|} \right\}_K \to 0,$$

und aus der Zulässigkeit der Suchrichtung (5.6) ergibt sich

$$\left\{ f'(x_k) \right\}_K \to 0.$$

Schließlich folgt zusammen mit der stetigen Differenzierbarkeit von f und der Konvergenz von $\{x_k\}_K$ die Stationarität $f'(x^*) = 0$.

Ende 6. V

§ 5.2 Schrittweitenstrategien

Zuerst wollen wir uns mit der Wahl der Schrittweite auseinandersetzen. Das Ziel ist dabei, die Effizienz (Definition 5.9) oder zumindest die Zulässigkeit (Definition 5.8) der Schrittweiten zu erhalten.

§ 5.2.1 Armijo-Schrittweitensuche

Die Armijo-Regel stellt die einfachste Schrittweitensteuerung dar und ist für viele Verfahren ausreichend. Um einen hinreichenden Abstieg zu garantieren, wird gefordert, dass die Schrittweite α_k die **Armijo-Bedingung**

$$f(x_k + \alpha\, d_k) \leq f(x_k) + \sigma\, \alpha\, f'(x_k)\, d_k \tag{5.12}$$

für einen festen Parameter $\sigma \in (0,1)$ erfüllt, siehe Abbildung 5.1. Nutzen wir die Hilfsfunktion

$$\varphi(\alpha) = f(x_k + \alpha\, d_k),$$

so können wir die Bedingung (5.12) äquivalent in der Form

$$\varphi(\alpha) \leq \varphi(0) + \sigma\, \alpha\, \varphi'(0) \tag{5.12}$$

schreiben. Aufgrund der Kettenregel überträgt sich die C^1-Eigenschaft von f auf φ. Es gilt

$$\varphi'(\alpha) = f'(x_k + \alpha\, d_k)\, d_k$$
$$\text{und insbesondere} \quad \varphi'(0) = f'(x_k)\, d_k.$$

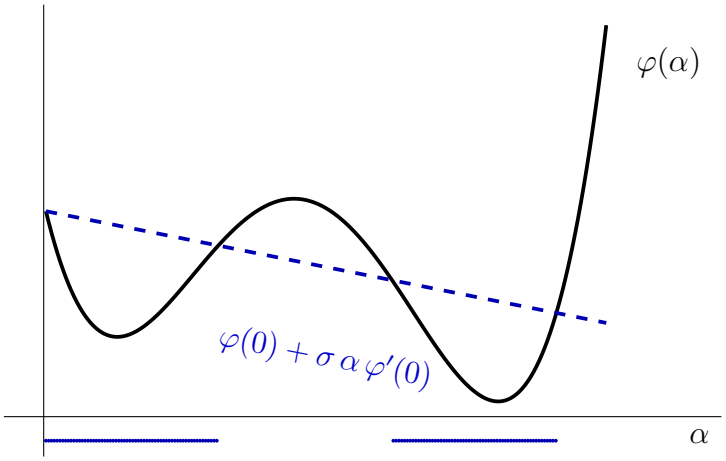

Abbildung 5.1. Illustration derjenigen Schrittweiten $\alpha \geq 0$, die die Armijo-Bedingung (5.12) erfüllen (blau). Als Parameter ist hier $\sigma = 0.05$ gewählt.

Lemma 5.12 (Existenz von Armijo-Schrittweiten)**.** Es sei d_k eine Abstiegsrichtung für f im Punkt x_k. Dann gibt es ein $\bar{\alpha} > 0$, sodass (5.12) für alle $\alpha \in [0, \bar{\alpha}]$ erfüllt ist.

Beweis. Da φ' in 0 stetig ist, gibt es ein $\bar{\alpha} > 0$ mit

$$\varphi'(\alpha) < \sigma\,\varphi'(0) \quad \text{für alle } \alpha \in [0, \bar{\alpha}].$$

Aus der Taylorformel mit Zwischenstelle folgt mit einem $\xi \in [0, \alpha]$

$$\begin{aligned}\varphi(\alpha) &= \varphi(0) + \alpha\,\varphi'(\xi)\\ &\leq \varphi(0) + \sigma\,\alpha\,\varphi'(0).\end{aligned}$$

Also ist (5.12) für alle $\alpha \in [0, \bar{\alpha}]$ erfüllt.

Damit ist die Armijo-Bedingung erfüllbar, allerdings müssen wir noch verhindern, dass die Schrittweiten zu klein werden, vgl. Beispiel 5.7. Dies wird durch eine **Backtracking** genannte Strategie erreicht: Man sucht von größeren hin zu kleineren Schrittweiten, bis zum ersten Mal die Armijo-Bedingung (5.12) erfüllt ist.

Algorithmus 5.13 (Armijo-Liniensuche mit Backtracking)**.**
Eingabe: Startschrittweite $\alpha_0 > 0$
Eingabe: Funktionsauswertung φ
Eingabe: ggf. vorberechnete Funktionswerte $\varphi(0)$ und $\varphi'(0)$
Eingabe: Parameter $\beta \in (0,1)$, $\sigma \in (0,1)$
Ausgabe: Schrittweite α, die die Armijo-Bedingung (5.12) erfüllt

```
1: Setze ℓ := 0
2: while Armijo-Bedingung (5.12) für α_ℓ verletzt do
3:     Setze α_{ℓ+1} := β α_ℓ
4:     Setze ℓ := ℓ + 1
5: end while
6: return α_ℓ
```

Bemerkung 5.14 (zum Armijo-Backtracking)**.**

(a) Typisch sind die Werte $\beta = 1/2$ und $\sigma = 10^{-2}$.

(b) Jede Iteration erfordert eine weitere Auswertung von φ, also eine Auswertung der Zielfunktion f.

(c) Mit Lemma 5.12 ist klar, dass Algorithmus 5.13 nach endlich vielen Schritten terminiert, und es gilt $\alpha_\ell \geq \bar{\alpha}\,\beta$. (Dabei ist $\bar{\alpha}$ das maximale $\bar{\alpha}$, sodass (5.12) für alle $\alpha \in [0, \bar{\alpha}]$ erfüllt ist.)

(d) In der praktischen Realisierung wird man Algorithmus 5.13 noch mit weiteren Abbruchbedingungen als (5.12) ausstatten, falls etwa wider Erwarten $\varphi'(0) \geq 0$ gilt oder die Schrittweiten α zu klein werden bzw. zuviele Fehlversuche (Iterationen) auftreten.

Geeignete Werte für die Startschrittweite α_0 hängen vom „äußeren“ Optimierungsverfahren ab, das verwendet wird, also von der Vorschrift, mit der die Suchrichtung d_k generiert wird. Dazu später mehr bei den konkreten Verfahren. Zu beachten ist jedoch, dass die Backtracking-Strategie die Startschrittweite immer nur verkürzt. Daher ist darauf zu achten, dass α_0 hinreichend groß ist, damit man zulässige Schrittweiten im Sinne der Definition 5.8 erhält und den entlang d_k möglichen Abstieg hinreichend gut ausnutzen kann. Davon handelt das folgende Resultat:

Lemma 5.15 (Zulässigkeit der Armijo-Schrittweiten)**.**

Die Schrittweiten α_k in Algorithmus 5.2 seien mittels des Armijo-Backtracking-Verfahrens (Algorithmus 5.13) erzeugt. In Schritt k werde die Startschrittweite $\alpha_{k,0} > 0$ verwendet. Weiter sei $\{x_k\}_K$ eine beschränkte Teilfolge, und es gelte

$$\alpha_{k,0} \|d_k\|_M \geq \psi\Big(\frac{-f'(x_k)\, d_k}{\|d_k\|_M}\Big) \quad \text{für alle } k \in K \tag{5.13}$$

für eine streng monoton wachsende Funktion $\psi : [0,\infty) \to [0,\infty)$. Dann sind die Schrittweiten $\{\alpha_k\}_K$ zulässig.

Beachte: Mit der Armijo-Liniensuche und geeigneter Startschrittweite (und in Verbindung mit zulässigen Suchrichtungen) lassen sich also die Voraussetzungen des globalen Konvergenzsatzes 5.11 erfüllen. Da dort die Zulässigkeit der Schrittweiten nur für eine Teilfolge genutzt werden (sodass $\{x_k\}_K$ konvergiert), reicht es hier aus, sich ebenfalls nur auf Teilfolgen zu beschränken.

Beweis. Da die Teilfolge $\{x_k\}_K$ beschränkt ist, hat sie eine konvergente Teilfolge mit Indixmenge K'. Damit konvergiert auch $\{f(x_k)\}_{K'}$. Da die gesamte Folge $\{f(x_k)\}$ monoton fallend ist, ist sie auch insgesamt konvergent.

Wegen der Armijo-Bedingung (5.12) gilt also

$$0 \leftarrow f(x_{k+1}) - f(x_k) \leq \sigma\, \alpha_k\, f'(x_k)\, d_k < 0$$

und somit

$$\alpha_k\, f'(x_k)\, d_k \to 0. \tag{$*$}$$

Die Bedingung (5.9a) ist also erfüllt, und wir müssen (5.9b) zeigen, also:

$$\left\{\frac{f'(x_k)\, d_k}{\|d_k\|_M}\right\}_K \to 0.$$

Die Idee dazu ist die folgende Fallunterscheidung bei den Indizes:

$$\alpha_k \|d_k\|_M \text{ ist „groß“} \quad \Rightarrow \quad \frac{\alpha_k\, f'(x_k)\, d_k}{\alpha_k \|d_k\|_M} \to 0$$

$$\alpha_k \|d_k\|_M \text{ ist „klein“} \quad \Rightarrow \quad \begin{cases} \text{nutze die Annahme (5.13),} & \text{falls } \alpha_k = \alpha_{k,0}, \\ \text{nutze: (5.12) ist für } \alpha_k/\beta \text{ verletzt,} & \text{falls } \alpha_k < \alpha_{k,0}. \end{cases}$$

Sei nun $\varepsilon > 0$ gegeben.

Da die Folge $\{x_k\}_K$ beschränkt ist, ist die stetige Funktion f' „in der Nähe der $\{x_k\}_K$“ auch gleichmäßig stetig, genauer: f' ist gleichmäßig stetig in der kompakten Menge

$$\operatorname{cl} \bigcup_{k\in K} U_\delta^M(x_k).$$

Es gibt also ein $\bar\delta > 0$ mit

$$\big\|f'(x_k + d) - f'(x_k)\big\|_{M^{-1}} \leq (1-\sigma)\,\varepsilon \quad \text{für alle } k \in K,\ \|d\|_M \leq \bar\delta.$$

Wir setzen nun

$$\delta := \min\big\{\bar\delta\,\beta, \psi(\varepsilon)\big\} > 0.$$

Wegen $(*)$ gibt es ein $k_0 \in \mathbb{N}$ mit

$$\alpha_k\, |f'(x_k)\, d_k| \leq \varepsilon\, \delta \quad \text{für alle } k \geq k_0.$$

Sei nun $k \in K$, $k \geq k_0$ beliebig. Wir unterscheiden folgende Fälle:

Fall 1: $\alpha_k \|d_k\|_M \geq \delta$
Es folgt sofort

$$0 \leq \frac{-f'(x_k)\, d_k}{\|d_k\|_M} = \frac{-\alpha_k\, f'(x_k)\, d_k}{\alpha_k \|d_k\|_M} \leq \frac{\varepsilon\, \delta}{\delta} = \varepsilon.$$

Fall 2: $\alpha_k \|d_k\|_M < \delta$ und $\alpha_k = \alpha_{k,0}$
Aus (5.13) folgt

$$\psi(\varepsilon) \geq \delta > \alpha_{k,0} \|d_k\|_M \geq \psi\Big(\frac{-f'(x_k)\, d_k}{\|d_k\|_M}\Big)$$

und somit

$$0 \leq \frac{-f'(x_k)\, d_k}{\|d_k\|_M} \leq \varepsilon,$$

da ψ streng monoton wachsend ist.

Fall 3: $\alpha_k \|d_k\|_M < \delta$ und $\alpha_k < \alpha_{k,0}$
Da $\alpha_k < \alpha_{k,0}$ ist, ist die Armijo-Bedingung (5.12) insbesondere für die zuvor getestete Schrittweite α_k/β noch verletzt:

$$\sigma \frac{\alpha_k}{\beta} f'(x_k)\, d_k < f\big(x_k + \frac{\alpha_k}{\beta} d_k\big) - f(x_k).$$

Nach dem Satz von Taylor 2.1 existiert $\xi_k \in (0,1)$, und wir haben weiter:

$$= \frac{\alpha_k}{\beta} f'\big(x_k + \frac{\alpha_k}{\beta} \xi_k\, d_k\big)\, d_k.$$

Daraus folgt

$$\begin{aligned} \sigma\, f'(x_k)\, d_k &\leq f'\big(x_k + \frac{\alpha_k}{\beta} \xi_k\, d_k\big)\, d_k \\ &\leq f'(x_k)\, d_k + \big\| f'\big(x_k + \underbrace{\frac{\alpha_k}{\beta} \xi_k\, d_k}_{=:d}\big) - f'(x_k)\big\|_{M^{-1}} \|d_k\|_M. \end{aligned}$$

Weil $\|d\|_M = \frac{\alpha_k}{\beta} \xi_k \|d_k\|_M \leq \delta/\beta \leq \bar{\delta}$ ist, erhalten wir

$$0 \leq \frac{-f'(x_k)\, d_k}{\|d_k\|_M} \leq \frac{1}{1-\sigma} \big\| f'\big(x_k + \frac{\alpha_k}{\beta} \xi_k\, d_k\big) - f'(x_k)\big\|_{M^{-1}} \leq \varepsilon.$$

Dies zeigt insgesamt wie gewünscht

$$\left\{ \frac{f'(x_k)\, d_k}{\|d_k\|_M} \right\}_K \to 0.$$

Gilt sogar

$$\alpha_{k,0} \|d_k\|_M \geq c \frac{-f'(x_k)\, d_k}{\|d_k\|_M} \tag{5.14}$$

mit einer Konstanten $c > 0$ (also $\psi(z) = c\, z$) und ist f' Lipschitz in $\mathcal{M}_f(x_0)$, dann kann man zeigen, dass Algorithmus 5.13 nicht nur zulässige, sondern sogar effiziente Schrittweiten liefert.

Abschließend betrachten wir noch eine Modifikation der Armijo-Backtracking-Strategie, die in der Praxis oft effizientere Schrittweitenvorschläge liefert als das simple Backtracking ($\alpha_{\ell+1} := \beta\, \alpha_\ell$). Dazu machen wir Gebrauch von den während der Suche

ohnehin verfügbaren Informationen über die zu minimierende Liniensuchfunktion φ. Ist α_ℓ die aktuelle Testschrittweite, dann haben wir die Daten

$$\varphi(0), \quad \varphi'(0) < 0 \quad \text{und} \quad \varphi(\alpha_\ell).$$

zur Verfügung. Damit kann ein quadratisches Modell (Hermite-Interpolationspolynom) von φ erstellt werden:

$$p(\alpha) = a + b\,\alpha + c\,\alpha^2.$$

Mit den Bedingungen $p(0) = \varphi(0)$, $p'(0) = \varphi'(0)$ und $p(\alpha_\ell) = \varphi(\alpha_\ell)$ erhält man

$$a = \varphi(0), \qquad b = \varphi'(0), \qquad c = \frac{1}{\alpha_\ell^2}\big(\varphi(\alpha_\ell) - \varphi(0) - \varphi'(0)\,\alpha_\ell\big). \tag{5.15}$$

Das quadratische Modell wird natürlich nur aufgestellt, falls die Armijo-Bedingung (5.12) mit der Testschrittweite α_ℓ noch verletzt ist, also falls

$$\varphi(\alpha_\ell) - \varphi(0) - \varphi'(0)\,\alpha_\ell > \varphi(\alpha_\ell) - \varphi(0) - \sigma\,\varphi'(0)\,\alpha_\ell > 0$$

und damit $c > 0$ gilt. Damit existiert die globale Minimalstelle $\alpha^* = -\frac{b}{2c}$ von p als

$$\alpha^* = \frac{-\varphi'(0)\,\alpha_\ell^2}{2\,\big(\varphi(\alpha_\ell) - \varphi(0) - \varphi'(0)\,\alpha_\ell\big)} > 0.$$

Die nächste Testschrittweite $\alpha_{\ell+1}$ wird dann z. B. gemäß

$$\alpha_{\ell+1} := \min\big\{\max\{\alpha^*, \underline{\beta}\,\alpha_\ell\}, \overline{\beta}\,\alpha_\ell\big\} = \begin{cases} \underline{\beta}\,\alpha_\ell, & \text{falls } \alpha^* < \underline{\beta}\,\alpha_\ell \\ \alpha^*, & \text{falls } \underline{\beta}\,\alpha_\ell \le \alpha^* \le \overline{\beta}\,\alpha_\ell \\ \overline{\beta}\,\alpha_\ell, & \text{falls } \alpha^* > \overline{\beta}\,\alpha_\ell. \end{cases}$$

mit Parametern $0 < \underline{\beta} < \overline{\beta} < 1$ gewählt.[22] Die wesentlichen Eigenschaften der einfachen Armijo-Backtracking-Strategie, insbesondere die Zulässigkeit (Lemma 5.15) und ggf. Effizienz der Schrittweiten, bleiben dadurch erhalten.

Die Veränderungen gegenüber der einfachen Backtracking-Strategie von Algorithmus 5.13 sind nur gering. Beide Verfahren stehen Ihnen in einer gemeinsamen Implementierung `Armijo.m` auf der Homepage zur Vorlesung zur Verfügung.[23]

Algorithmus 5.16 (Armijo-Liniensuche mit Backtracking und Interpolation)**.**
Eingabe: Startschrittweite $\alpha_0 > 0$
Eingabe: Funktionsauswertung φ
Eingabe: ggf. vorberechnete Funktionswerte $\varphi(0)$ und $\varphi'(0)$
Eingabe: Parameter $0 < \underline{\beta} < \overline{\beta} < 1$, $\sigma \in (0, 1)$
Ausgabe: Schrittweite α, die die Armijo-Bedingung (5.12) erfüllt

1: Setze $\ell := 0$
2: **while** Armijo-Bedingung (5.12) für α_ℓ verletzt **do**
3: Setze $\alpha_\ell^* := \dfrac{-\varphi'(0)\,\alpha_\ell^2}{2\,\big(\varphi(\alpha_\ell) - \varphi(0) - \varphi'(0)\,\alpha_\ell\big)}$
4: Setze $\alpha_{\ell+1} := \min\big\{\max\{\alpha_\ell^*, \underline{\beta}\,\alpha_\ell\}, \overline{\beta}\,\alpha_\ell\big\}$
5: Setze $\ell := \ell + 1$
6: **end while**
7: **return** α_ℓ

[22] Mit $\underline{\beta} = \overline{\beta} = \beta$ erhalten wir wieder die einfache Backtracking-Strategie.
[23] Ein Test erfolgt in Übung 3, Aufgabe 11.

Ende 7. V

§ 5.2.2 Wolfe-Powell-Liniensuche

Literatur: (Geiger and Kanzow, 1999, Kapitel 5, 6)

Bei der Armijo-Bedingung

$$f(x_k + \alpha\, d_k) \le f(x_k) + \sigma\, \alpha\, f'(x_k)\, d_k \quad \text{bzw.} \quad \varphi(\alpha) \le \varphi(0) + \sigma\, \alpha\, \varphi'(0) \tag{5.12}$$

mussten wir das Backtracking nutzen, also uns „von oben“ einer hinreichend guten Schrittweite nähern, da (5.12) allein keine zu kleinen Schritte verhindert. Eine andere Möglichkeit ist, zusätzlich zu (5.12) die **Krümmungsbedingung**

$$f'(x_k + \alpha\, d_k)\, d_k \ge \tau\, f'(x_k)\, d_k \quad \text{bzw.} \quad \varphi'(\alpha) \ge \tau\, \varphi'(0) \tag{5.16}$$

oder die **strenge Krümmungsbedingung**

$$|f'(x_k + \alpha\, d_k)\, d_k| \le -\tau\, f'(x_k)\, d_k \quad \text{bzw.} \quad |\varphi'(\alpha)| \le -\tau\, \varphi'(0) \tag{5.17}$$

mit einem Parameter $\tau \in (\sigma, 1)$ zu fordern. Die Krümmungsbedingung (5.16) fordert, dass an der Stelle α weniger Abstieg als im Startpunkt $\alpha = 0$ vorhanden ist, insbesondere ist Aufstieg erlaubt, siehe Abbildung 5.2. Damit werden zu kurze Schritte (Schrittweiten α nahe 0) ausgeschlossen. Die strenge Krümmungsbedingung (5.17) verlangt, dass auch ein eventueller Anstieg an der Stelle α klein ist.

Beachte: An einem lokalen Minimum von φ ist (5.17) mit $\tau = 0$ erfüllt.

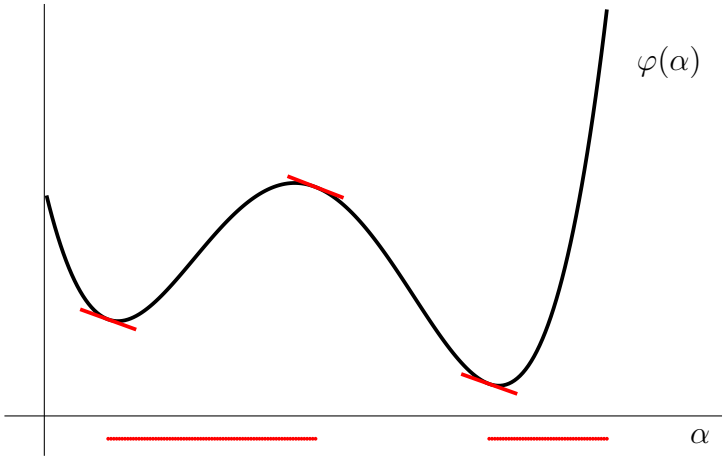

ABBILDUNG 5.2. Illustration derjenigen Schrittweiten $\alpha \ge 0$, die die Krümmungsbedingung (5.16) erfüllen (rot). Als Parameter ist hier $\tau = 0.1$ gewählt.

Die Bedingungen (5.12) und (5.16) (bzw. (5.17)) heißen zusammen die **(strenge) Wolfe-(Powell)-Bedingung**, siehe Abbildung 5.3.

Lemma 5.17 (Existenz von (strengen) Wolfe-Schrittweiten)**.**

Es sei d_k eine Abstiegsrichtung für f im Punkt x_k. Weiter sei die Funktion f auf dem Strahl $\{x_k + \alpha\, d_k,\ \alpha \ge 0\}$ nach unten beschränkt und $0 < \sigma < \tau < 1$. Dann gibt es eine Schrittweite $\alpha_2 > 0$, sodass die die Bedingungen (5.12) und (5.17) (und damit auch (5.16)) ein einer Umgebung von α_2 erfüllt sind.

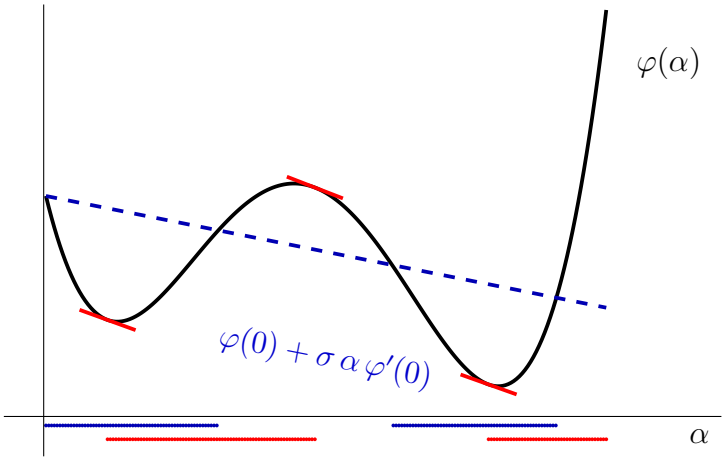

Abbildung 5.3. Illustration derjenigen Schrittweiten $\alpha \geq 0$, die die Armijo-Bedingung (5.12) (blau) und die Krümmungsbedingung (5.16) erfüllen (rot). Als Parameter sind hier $\sigma = 0.05$ und $\tau = 0.1$ gewählt.

Beweis. Wir setzen wieder $\varphi(\alpha) := f(x_k + \alpha\, d_k)$. Da φ nach Voraussetzung nach unten beschränkt ist, schneidet diese Funktion die Armijo-Gerade

$$\alpha \mapsto \varphi(0) + \sigma \underbrace{\varphi'(0)}_{<0} \alpha$$

in mindestens einem positiven Punkt. Es sei α_1 der kleinste (Warum existiert der?) positive Schnittpunkt, es gilt also

$$\varphi(\alpha_1) = \varphi(0) + \sigma\,\varphi'(0)\,\alpha_1.$$

Da $\varphi(0) < 0$ ist, gilt (5.12) für alle $\alpha \in [0, \alpha_1]$. Mit dem Mittelwertsatz 2.1 gibt es ein $\alpha_2 \in (0, \alpha_1)$ mit

$$\varphi'(\alpha_2) = \frac{\varphi(\alpha_1) - \varphi(0)}{\alpha_1} = \sigma\,\varphi'(0).$$

Somit gilt

$$|\varphi'(\alpha_2)| = -\sigma\,\varphi'(0) < -\tau\,\varphi'(0).$$

Wegen der Stetigkeit von φ' sind nun auch (5.16) und (5.17) für alle α in der Nähe von α_2 erfüllt.

Die Idee für einen Algorithmus zur Bestimmung einer Wolfe-Schrittweite folgt diesem Beweis und wird im folgenden Lemma verschärft. Wir nutzen zur Vereinfachung der Notation die Hilfsfunktion (vorzeichenbehafteter Abstand zwischen φ und der Armijo-Geraden)

$$\psi(\alpha) := \varphi(\alpha) - \varphi(0) - \sigma\,\varphi'(0)\,\alpha$$

und erhalten

$$\begin{aligned}
(5.12) \quad &\Longleftrightarrow \quad \psi(\alpha) \leq 0, \\
(5.16) \quad &\Longleftrightarrow \quad -(\tau - \sigma)\,|\varphi'(0)| \leq \psi'(\alpha) \\
(5.17) \quad &\Longleftrightarrow \quad -(\tau - \sigma)\,|\varphi'(0)| \leq \psi'(\alpha) \leq (\tau + \sigma)\,|\varphi'(0)|.
\end{aligned}$$

Wir beschränken uns im Folgenden auf die einfache Wolfe-Bedingung (5.12), (5.16). Für die strenge Wolfe-Bedingung siehe z. B. (Geiger and Kanzow, 1999, Kapitel 6.3).

Lemma 5.18 (Einschachtelung von Wolfe-Schrittweiten)**.**

Es seien $0 \leq a < b$ so gewählt, dass die Bedingungen (siehe Abbildung 5.4)

$$\psi(a) \leq 0 \quad \text{und} \quad \psi'(a) < 0 \tag{5.18a}$$

$$\text{sowie} \quad \psi(b) \geq 0. \tag{5.18b}$$

Dann existiert ein Punkt $\alpha^* \in (a, b)$ mit

$$\psi(\alpha^*) < 0, \quad \psi'(\alpha^*) = 0.$$

Insbesondere sind in einer Umgebung des Punktes α^* (5.12) und (5.16) beide erfüllt.

Beweis. Es sei α^* ein globaler Minimierer von

$$\text{Minimiere} \quad \psi(\alpha) \quad \text{auf dem Intervall } [a, b].$$

Aus den Bedingungen an a und b folgt, dass $\alpha^* \in (a, b)$ liegt und somit $\psi'(\alpha^*) = 0$ gilt. Wegen $\psi(a) \leq 0$ gilt auch $\psi(\alpha^*) < 0$. Aus der Stetigkeit von ψ und ψ' folgt die letzte Aussage.

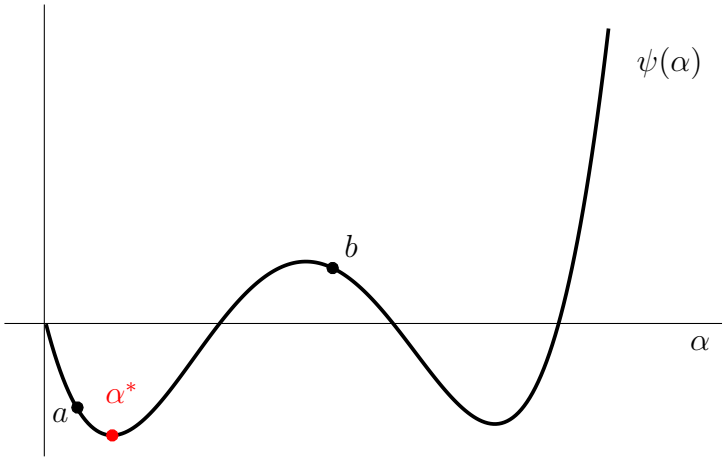

ABBILDUNG 5.4. Illustration der Bedingung (5.18) und der Aussage von Lemma 5.18.

Beachte: Die Bedingung (5.18a) wird bereits durch den Punkt $a = 0$ erfüllt. Dadurch ergibt sich die Strategie, zunächst eine rechte Grenze b zu finden, sodass auch (5.18b) erfüllt ist, und dann α^* einzuschachteln. Dies führt auf folgendes Verfahren, das Ihnen unter `Wolfe.m` auf der Homepage zur Vorlesung zur Verfügung steht.[24]

Algorithmus 5.19 ((einfache) Wolfe-Liniensuche)**.**

Eingabe: Startschrittweite $\alpha_0 > 0$
Eingabe: Funktionsauswertungen φ und φ'
Eingabe: ggf. vorberechnete Funktionswerte $\varphi(0)$ und $\varphi'(0)$
Eingabe: Parameter $\gamma > 1$, $\underline{\gamma}, \overline{\gamma} \in (0, 1/2]$, $0 < \sigma < \tau < 1$
Ausgabe: Schrittweite α, die die Wolfe-Bedingungen (5.12) und (5.16) erfüllt

1: Setze $a := 0$ und $b := \alpha_0$
2: Setze $\ell := 0$
3: **while** $\varphi(b) < \varphi(0) + \sigma\,\varphi'(0)\,b$ und $\varphi'(b) < \tau\,\varphi'(0)$ **do**

[24] Ein Test erfolgt in Übung 3, Aufgabe 12.

4: Setze $b := \gamma\, b$
5: Setze $\ell := \ell + 1$
6: **end while** {Nun gilt (5.18) oder sogar bereits (5.12) und (5.16)}
7: Setze $\alpha_\ell := b$
8: **while** Armijo-Bedingung (5.12) oder Krümmungsbedingung (5.16) für α_ℓ verletzt **do**
9: Wähle $\alpha_{\ell+1} \in [a + \underline{\gamma}\,(b-a), b - \overline{\gamma}\,(b-a)]$
10: **if** $\varphi(\alpha_{\ell+1}) \geq \varphi(0) + \sigma\,\varphi'(0)\,\alpha_{\ell+1}$ **then**
11: Setze $b := \alpha_{\ell+1}$
12: **else**
13: Setze $a := \alpha_{\ell+1}$
14: **end if**
15: Setze $\ell := \ell + 1$
16: **end while** {Nun gelten (5.12) und (5.16)}
17: **return** α_ℓ

Bemerkung 5.20 (zur Wolfe-Liniensuche, vgl. Bemerkung 5.14)**.**

(a) Der Parameter σ wird wie bei der Armijo-Liniensuche klein gewählt, z. B. $\sigma = 10^{-2}$. Je nach äußerem Verfahren (zur Bestimmung der Suchrichtungen) muss τ „klein“ gewählt werden, z. B. $\tau = 0.1$, sonst wählt man τ „groß“, z. B. $\tau = 0.9$.

(b) Jede Iteration erfordert eine weitere Auswertung von φ und ggf. auch eine Auswertung von φ', also eine Auswertung der Zielfunktion f und ggf. die Auswertung einer Richtungsableitung von f.

(c) Mit Lemma 5.18 ist leicht zu sehen, dass Algorithmus 5.19 unter den Voraussetzungen von Lemma 5.17 terminiert:

- Die bei Schritt 3 beginnende Schleife bricht ab, da für hinreichend große b die Armijo-Bedingung (5.12) verletzt ist, somit gilt $\psi(b) > 0$, und (5.18b) ist erfüllt.
- Beim ersten Durchlauf der bei Schritt 8 beginnenden Schleife sind die Bedingungen (5.18) von Lemma 5.18 erfüllt und damit auch in allen weiteren Schritten.
- Da die Länge des Intervalls $[a, b]$ gegen null strebt und es eine offene Menge von Punkten gibt, die (5.12) und (5.16) erfüllen, muss der Algorithmus terminieren.

(d) Die akzeptierte Schrittweite α_ℓ kann kleiner oder größer sein als die Startschrittweite α_0.

(e) In der praktischen Realisierung wird man Algorithmus 5.19 noch mit weiteren Abbruchbedingungen ausstatten, falls etwa wider Erwarten $\varphi'(0) \geq 0$ gilt oder die Schrittweiten α zu klein oder zu groß werden bzw. zuviele Fehlversuche (Iterationen) auftreten.

(f) Für einen Algorithmus zur *strengen* Wolfe-Liniensuche verweisen wir auf (Geiger and Kanzow, 1999, Kapitel 6.3).

Wie bei der Armijo-Liniensuche kann die Zulässigkeit der Schrittweiten gezeigt werden.

Lemma 5.21 (Zulässigkeit der Wolfe-Schrittweiten)**.**

Die Schrittweiten α_k in Algorithmus 5.2 seien so gewählt, dass dass (5.12) und (5.16) erfüllt sind (etwa mittels Algorithmus 5.19).[25] Weiter sei $\{x_k\}_K$ eine beschränkte Teilfolge. Dann sind die Schrittweiten $\{\alpha_k\}_K$ zulässig.

Daraus folgt natürlich auch die Zulässigkeit von Schrittweiten, die sogar die *strengen* Wolfe-Bedingungen (5.12) und (5.17) erfüllen.

Beweis. Wie im Beweis von Lemma 5.15 erhalten wir die Aussage

$$-\alpha_k f'(x_k)\, d_k \to 0. \tag{$*$}$$

Wir müssen nun

$$\left\{ \frac{f'(x_k)\, d_k}{\|d_k\|} \right\}_K \to 0.$$

zeigen. Sei dazu $\varepsilon > 0$ gegeben.

Ebenso wie im Beweis von Lemma 5.15 gilt: Da die Folge $\{x_k\}_K$ beschränkt ist, ist die stetige Funktion f' „in der Nähe der $\{x_k\}_K$“ auch gleichmäßig stetig. Es gibt also ein $\delta > 0$ mit

$$\left\| f'(x_k + d) - f'(x_k) \right\|_{M^{-1}} \leq (1 - \tau)\, \varepsilon \quad \text{für alle } k \in K,\ \|d\|_M \leq \delta.$$

Wegen $(*)$ gibt es ein $k_0 \in \mathbb{N}$ mit

$$\alpha_k \left| f'(x_k)\, d_k \right| \leq \varepsilon\, \delta \quad \text{für alle } k \geq k_0.$$

Sei nun $k \in K$, $k \geq k_0$ beliebig. Wir unterscheiden folgende Fälle:

Fall 1: $\alpha_k \|d_k\|_M \geq \delta$
Es folgt sofort

$$0 \leq \frac{-f'(x_k)\, d_k}{\|d_k\|_M} = \frac{-\alpha_k f'(x_k)\, d_k}{\alpha_k \|d_k\|_M} \leq \frac{\varepsilon\, \delta}{\delta} = \varepsilon.$$

Fall 2: $\alpha_k \|d_k\|_M < \delta$
Die Erfülltheit der Wolfe-Bedingung (5.16) für α_k ergibt:

$$\tau\, f'(x_k)\, d_k \leq f'(x_k + \alpha_k\, d_k)\, d_k.$$

Addiert man $|f'(x_k)\, d_k| = -f'(x_k)\, d_k$, so erhält man

$$\begin{aligned}(1 - \tau) \left| f'(x_k)\, d_k \right| &\leq f'(x_k + \alpha_k\, d_k)\, d_k - f'(x_k)\, d_k \\ &\leq \left| f'\big(x_k + \alpha_k\, d_k\big)\, d_k - f'(x_k)\, d_k \right| \\ &\leq \left\| f'\big(x_k + \alpha_k\, d_k\big) - f'(x_k) \right\|_{M^{-1}} \|d_k\|_M.\end{aligned}$$

Zusammen mit der gleichmäßigen Stetigkeit erhalten wir

$$\cdots \leq (1 - \tau)\, \varepsilon\, \|d_k\|_M,$$

also

$$0 \leq \frac{-f'(x_k)\, d_k}{\|d_k\|_M} \leq \varepsilon.$$

[25] Eine Bedingung an die Startschrittweite $\alpha_{k,0}$ ist nicht erforderlich.

Analog zum Armijo-Verfahren kann man auch die Effizienz der Schrittweiten zeigen, wenn f' Lipschitz-stetig in $\mathcal{M}_f(x_0)$ ist.

Abschließend bemerken wir noch, dass wir in Schritt 9 von Algorithmus 5.19 einige Freiheit bei der Wahl von $\alpha_{\ell+1}$ haben. Aufgrund der vorliegenden Daten $\varphi(a)$, $\varphi'(a)$, $\varphi(b)$ und $\varphi'(b)$ bietet sich dafür eine kubische (Hermite)-Interpolation an:

$$p(\alpha) = a + b\,\alpha + c\,\alpha^2 + d\,\alpha^3.$$

Sofern das (eindeutige) lokale Minimum α^* von p existiert, kann es explizit berechnet und anschließend ins Intervall $[a, b]$ projiziert werden:

$$\alpha_{\ell+1} := \max\{a, \min\{b, \alpha^*\}\}.$$

Zu beachten ist noch, dass nicht in jedem Fall $\varphi'(a)$ und $\varphi'(b)$ in der aktuellen Iteration von Algorithmus 5.19 bekannt sein müssen. In dem Fall kann man auf ein quadratisches Interpolationspolynom analog zum Armijo-Verfahren zurückgreifen.

Bemerkung 5.22 (zur Skalierungs-Invarianz der Armijo- und Krümmungsbedingungen[26])**.**

Die Bedingungen (5.12), (5.16) und (5.17) sind invariant gegenüber affinen Skalierungen im Bild- und Definitionsbereich:

$$f(x) \quad \rightsquigarrow \quad g(x) := \gamma\, f(A\,x + b) + \delta$$

mit $A \in \mathbb{R}^{n\times n}$ und $b \in \mathbb{R}^n$, $\gamma > 0$ und $\delta \in \mathbb{R}$.

Genauer: Schrittweiten α, die eine der Bedingungen (5.12), (5.16), (5.17) für die Funktion g an der Stelle x und mit der Suchrichtung d erfüllen, erfüllen diese auch für die Funktion f an der Stelle $A\,x + b$ und mit der Suchrichtung $A\,d$. Da die Skalierung des Optimierungsproblems bei der Modellierung oft willkürlich ist, ist dies eine wünschenswerte Eigenschaft.

Ende 8. V

§ 5.3 Das Gradientenverfahren

Im weiteren Verlauf von § 5 besprechen wir verschiedene konkrete Realisierungen des allgemeinen Abstiegsverfahrens aus Algorithmus 5.2. Die Verfahren unterscheiden sich in der Erzeugung der Suchrichtungen d_k und in der Auswahl des Liniensuchverfahrens zur Bestimmung der Schrittweite α_k. Wie bereits erwähnt, ist allen besprochenen Verfahren gemeinsam, dass sie die Suchrichtung an der Stelle x_k durch die Minimierung eines quadratischen Ersatzmodells der Zielfunktion

$$q_k(d) = f(x_k) + f'(x_k)\,d + \frac{1}{2}\,d^\top H_k\,d \tag{5.2}$$

erzeugen, also — falls H_k spd ist —, durch Lösung des LGS

$$H_k\,d_k = -\nabla f(x_k). \tag{5.4}$$

Das **Gradientenverfahren** (**Verfahren des steilsten Abstiegs**) für unsere allgemeine nichtlineare Aufgabe

$$\text{Minimiere} \quad f(x) \quad \text{über } x \in \mathbb{R}^n \tag{FO}$$

[26] Beweis in Übung 3, Aufgabe 10

erzeugt seine Suchrichtungen auf dieselbe Art wie bereits in § 4.1 für den Fall, dass f ein quadratisches Polynom war. Es gilt also

$$M\,d_k = -\nabla f(x_k) \quad \text{bzw.} \quad d_k = -M^{-1}\nabla f(x_k) = -\nabla_M f(x_k). \tag{5.19}$$

Dies entspricht der Verwendung einer konstanten Hessematrix $H_k \equiv M$ im Ersatzmodell (5.2):

$$q_k(d) = f(x_k) + f'(x_k)\,d + \frac{1}{2}\,d^\top M\,d.$$

Das Skalarprodukt M ist dabei vom Anwender zu wählen. Wie bereits in Bemerkung 4.6 erwähnt, bezeichnet man genauer den Fall $M = I$ als **klassisches Gradientenverfahren** (ohne Vorkonditionierung) und spricht ansonsten vom **vorkonditionierten Gradientenverfahren** mit dem **Vorkonditionierer** M.

Diese Wahl von d_k impliziert natürlich die Winkelbedingung (5.7) mit dem Maximalwert $\eta = 1$. Insbesondere ist die Suchrichtung d_k also eine Abstiegsrichtung für f an der Stelle x_k, solange $f'(x_k) \neq 0$ ist.

Für die Bestimmung der Schrittweiten reicht bereits irgendeine einfache Strategie aus, die zulässige Schrittweiten (Definition 5.8) erzeugt, typischerweise die Armijo-Liniensuche mit Backtracking (Algorithmus 5.13) und geeigneter Startschrittweite $\alpha_{k,0}$. Die Bedingung (5.14) verlangt für letztere wegen $d_k = -\nabla_M f(x_k)$ lediglich

$$\alpha_{k,0} \geq c\,\frac{-f'(x_k)\,d_k}{\|d_k\|_M^2} = c\,\frac{-\big(\nabla_M f(x_k),\,d_k\big)_M}{\|d_k\|_M^2} = c\,\frac{\|d_k\|_M^2}{\|d_k\|_M^2} = c$$

mit irgendeiner Konstanten $c > 0$, die wir wegen ihrer Bedeutung weiter unten im Algorithmus 5.23 auch mit $\alpha_{0,\min}$ bezeichnen.

Zusätzlich ist es sinnvoll, aus der Vergangenheit zu lernen. Geht man davon aus, dass der Abstieg im aktuellen Schritt in erster Näherung gleich groß sein wird wie der im letzten Schritt (zur akzeptierten Schrittweite α_{k-1}), so ergibt sich der initiale Schrittweitenvorschlag (für Iteration $k \geq 1$)

$$\begin{aligned} &\alpha_{k,0}\,f'(x_k)\,d_k = \alpha_{k-1} f'(x_{k-1})\,d_{k-1} \\ \Rightarrow\quad &\alpha_{k,0} = \alpha_{k-1}\,\frac{f'(x_{k-1})\,d_{k-1}}{f'(x_k)\,d_k}, \end{aligned}$$

speziell beim Gradientenverfahren also

$$\alpha_{k,0} = \alpha_{k-1}\,\frac{\|\nabla_M f(x_{k-1})\|_M^2}{\|\nabla_M f(x_k)\|_M^2} = \alpha_{k-1}\,\frac{\|d_{k-1}\|_M^2}{\|d_k\|_M^2}.$$

Verwendet man statt der Linearisierung den tatsächlich erreichten Abstieg im letzten Schritt, so erhält man stattdessen

$$\alpha_{k,0} = \frac{f(x_{k-1}) - f(x_k)}{\|\nabla_M f(x_k)\|_M^2} = \frac{f(x_{k-1}) - f(x_k)}{\|d_k\|_M^2}.$$

Der Vollständigkeit halber geben wir das (vorkonditionierte) Gradientenverfahren als Spezialisierung des allgemeinen Verfahrens Algorithmus 5.2 mit den obigen Überlegungen zur Schrittweite hier nochmals im Detail an. Die Konvergenz des Verfahrens (in dem Sinne: jeder Häufungspunkt ist stationär) folgt sofort aus dem globalen Konvergenzsatz 5.11.

Algorithmus 5.23 (Gradientenverfahren).
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: Funktionsauswertungen f und f' bzw. ∇f
Eingabe: spd Matrix M (oder Matrix-Vektor-Produkte mit M^{-1})
Eingabe: Parameter $\beta \in (0,1)$, $\sigma \in (0,1)$ für die Armijo-Liniensuche
Eingabe: Parameter $\alpha_{0,\min}$ als minimale Startschrittweite für die Armijo-Liniensuche
Ausgabe: näherungsweise stationärer Punkt der Aufgabe (**FO**)

Setze $k := 0$
Setze $f_0 := f(x_0)$
Setze $r_0 := f'(x_0)^\top = \nabla f(x_0)$
Setze $d_0 := -M^{-1} r_0$
Setze $\delta_0 := -r_0^\top d_0$ $\{\delta_0 = \|\nabla_M f(x_0)\|_M^2 = \|d_0\|_M^2\}$
while Abbruchkriterium nicht erfüllt **do**
  Setze $\alpha_{k,0} := \alpha_{0,\min}$ $(k=0)$ oder $\alpha_{k,0} := \max\{\alpha_{0,\min}, \frac{f_k - f_{k-1}}{\delta_k}\}$ $(k \geq 1)$
  Bestimme eine Schrittweite $\alpha_k > 0$ mittels Armijo-Liniensuche mit Backtracking (Algorithmus 5.13 oder Algorithmus 5.16), angewendet auf $\varphi(\alpha) := f(x_k + \alpha\, d_k)$ und mit Startschrittweite $\alpha_{k,0}$ $\{\varphi(0) = f_k$ und $\varphi'(0) = -\delta_k$ sind bekannt$\}$
  Setze $x_{k+1} := x_k + \alpha_k\, d_k$
  Setze $f_{k+1} := f(x_{k+1})$ {ist aus der Liniensuche bereits bekannt}
  Setze $r_{k+1} := f'(x_{k+1})^\top = \nabla f(x_{k+1})$
  Setze $d_{k+1} := -M^{-1} r_{k+1}$
  Setze $\delta_{k+1} := -r_{k+1}^\top d_{k+1}$ $\{\delta_{k+1} = \|\nabla_M f(x_{k+1})\|_M^2 = \|d_{k+1}\|_M^2\}$
  Setze $k := k+1$
end while
return x_k

In der Praxis kann man als Abbruchbedingung wieder eine der Bedingungen aus (4.22) verwenden, also nach der relativen oder absoluten Größe der Ableitung bzw. des Gradienten

$$\|r_k\|_{M^{-1}} = \|f'(x_k)\|_{M^{-1}} = \|\nabla_M f(x_k)\|_M = \|d_k\|_M = \delta_k^{1/2}$$

stoppen, die man ohnehin zur Verfügung hat. Auch eine eingeschränkte Interpretation im Sinne von Folgerung 4.13 ist noch möglich, denn: Falls die Folge $\{x_k\}$ gegen ein lokales Minimum x^* konvergiert, das die hinreichenden Bedingungen zweiter Ordnung (Satz 3.4) erfüllt, dann gilt beispielsweise: Für alle $\varepsilon > 0$ existiert ein $\delta > 0$, sodass

$$\|x_k - x^*\|_M \leq \delta \quad \text{und} \quad \|f'(x_k)\|_{M^{-1}} \leq \varepsilon_{\text{abs}} \quad \Rightarrow \quad \|x_k - x^*\|_M \leq \left(\frac{1}{\alpha} + \varepsilon\right) \varepsilon_{\text{abs}},$$

wobei $\alpha = \lambda_{\min}(\nabla^2 f(x^*); M)$ ist.[27]

[27] Kurz gesagt: Wenn man bereits hinreichend nah an einer Lösung ist, die die hinreichenden Bedingungen 2. Ordnung erfüllt, dann ist die Norm der Ableitung bzw. des Gradienten bis auf die Konstante $1/\alpha$ ein geeignetes Maß für den Abstand zur Lösung.

Weitere oft verwendete Abbruchbedingungen sind etwa

$$\|x_k - x_{k-1}\|_M \leq \varepsilon^x_{\text{abs}} + \varepsilon^x_{\text{rel}} \|x_k - x_0\|_M \quad \text{und}$$
$$|f(x_k) - f(x_{k-1})| \leq \varepsilon^f_{\text{abs}} + \varepsilon^f_{\text{rel}} |f(x_k) - f(x_0)|,$$

also mangelnder Fortschritt in den Iterierten oder den Funktionswerten, wobei typischerweise $\tau^f_\cdot = (\tau^x_\cdot)^2$ gesetzt wird.

Beachte: Bemerkenswert ist hier wieder, dass es möglich ist, die Größen $\|x_k - x_{k-1}\|_M$ und $\|x_k - x_0\|_M$ rekursiv zu berechnen, ohne dass die Matrix M (oder Matrix-Vektor-Produkte mit M) vorliegen muss. Es reichen allein Matrix-Vektor-Produkte mit M^{-1} aus.

§ 5.4 Das Newton-Verfahren

Literatur: (Geiger and Kanzow, 1999, Kapitel 7, 9) und (Ulbrich and Ulbrich, 2012, Kapitel 10)

Das Newton-Verfahren ist allgemein als ein Verfahren zur Lösung der (nichtlinearen) Gleichung $F(x) = 0$ bekannt, wobei $F : \mathbb{R}^n \to \mathbb{R}^n$ als stetig differenzierbar (eine C^1-Funktion) angenommen wird. Später wenden wir das Verfahren an auf die notwendige Bedingung 1. Ordnung, also $F(x) = \nabla f(x) = 0$ und setzen dann f als C^2-Funktion voraus.

Idee: Es sei x_0 die Schätzung einer Nullstelle von F. Wir legen im Punkt x_0 das lineare Taylor-Modell an die Funktion an und bestimmen ersatzweise *dessen* Nullstelle:

$$F(x_0) + F'(x_0)(x - x_0) = 0 \quad \Leftrightarrow \quad x = x_0 - F'(x_0)^{-1} F(x_0).$$

Diese Nullstelle dient als nächste Iterierte x_1 usw. Diese Iterationsvorschrift heißt **lokales Newton-Verfahren**.

§ 5.4.1 Einige Hilfsresultate

Wir wiederholen zunächst einige Hilfsresultate, die aus der Vorlesung *Grundlagen der Optimierung* bekannt sein sollten.

Lemma 5.24 (Banach-Lemma[28])**.**

(a) Es sei $K \in \mathbb{R}^{n\times n}$ mit $\|K\| < 1$. Dann ist $I - K$ regulär (invertierbar), und es gilt

$$\|(I - K)^{-1}\| \leq \frac{1}{1 - \|K\|}.$$

(b) Es seien $A, B \in \mathbb{R}^{n\times n}$ mit $\|I - BA\| < 1$. Dann sind A und B regulär, und es gilt

$$\|B^{-1}\| \leq \frac{\|A\|}{1 - \|I - BA\|} \quad \text{und} \quad \|A^{-1}\| \leq \frac{\|B\|}{1 - \|I - BA\|}.$$

Beachte: Aussage (a) besagt, dass „kleine“ Störungen die Einheitsmatrix invertierbar belassen. Aussage (b) besagt, dass wenn $I - BA$ „klein“ ist, also $B \approx A^{-1}$, notwendig A und B invertierbar sind.

[28] Hier darf sogar $\|\cdot\|$ eine beliebige, mit der beliebigen Vektornorm $\|\cdot\|$ kompatible Matrix-Norm sein, also $\|Ax\| \leq \|A\|\|x\|$.

Beweis. (a): Für $x \in \mathbb{R}^n$ gilt

$$\|(I-K)\,x\| = \|x - Kx\| \geq \|x\| - \|Kx\| \geq \big(\underbrace{1 - \|K\|}_{>0}\big)\,\|x\|.$$

Es folgt $(I-K)\,x \neq 0$ für $x \neq 0$, d. h., $I-K$ ist injektiv und damit regulär.

Es sei nun $y \in \mathbb{R}^n$ beliebig und $x := (I-K)^{-1}y$. Dann folgt aus der Abschätzung oben

$$\begin{aligned} & \|y\| \geq (1 - \|K\|)\|(I-K)^{-1}y\| \\ \Rightarrow \quad & \|(I-K)^{-1}\| = \max_{y\neq 0} \frac{\|(I-K)^{-1}y\|}{\|y\|} \leq \frac{1}{1-\|K\|}. \end{aligned}$$

(b): Es sei $K = I - BA$, also $\|K\| < 1$. Wegen (a) ist $I - K = I - (I - BA) = BA$ regulär, d. h., A und B sind beide regulär. Weiter gilt

$$\begin{aligned} & (I-K)^{-1} = (BA)^{-1} = A^{-1}B^{-1} \\ \Rightarrow \quad & B^{-1} = A(I-K)^{-1} \\ \Rightarrow \quad & \|B^{-1}\| \leq \|A\|\|(I-K)^{-1}\| \overset{(a)}{\leq} \frac{\|A\|}{1-\|K\|} = \frac{\|A\|}{1-\|I-BA\|}. \end{aligned}$$

Die andere Ungleichung folgt analog.

Lemma 5.25 (Konsequenzen der Invertierbarkeit der Jacobimatrix)**.**

Es sei $F : \mathbb{R}^n \to \mathbb{R}^n$ eine C^1-Funktion, $x^* \in \mathbb{R}^n$ beliebig und die Jacobimatrix $F'(x^*)$ regulär.

(a) Dann existieren eine Umgebung $U_\delta(x^*)$ und eine Konstante $c > 0$, sodass $F'(x)$ für alle $x \in U_\delta(x^*)$ regulär ist, und es gilt:

$$\|F'(x)^{-1}\| \leq c \quad \text{für alle } x \in U_\delta(x^*). \tag{5.20}$$

(b) Es gelte nun zusätzlich $F(x^*) = 0$. Dann existieren eine Umgebung $U_\delta(x^*)$ und eine Konstante $\beta > 0$, sodass gilt:

$$\beta\,\|x - x^*\| \leq \|F(x)\| \quad \text{für alle } x \in U_\delta(x^*). \tag{5.21}$$

Beweis.

(a) Da F' im Punkt x^* stetig ist, existiert ein $\delta > 0$ mit

$$\|F'(x^*) - F'(x)\| \leq \varepsilon := \frac{1}{2\,\|F'(x^*)^{-1}\|}$$

für alle $x \in U_\delta(x^*)$, also auch

$$\begin{aligned} \|I - F'(x^*)^{-1}\,F'(x)\| &= \|F'(x^*)^{-1}\big(F'(x^*) - F'(x)\big)\| \\ &\leq \|F'(x^*)^{-1}\|\|F'(x^*) - F'(x)\| \\ &\leq \frac{1}{2} < 1. \end{aligned}$$

Nach dem Banach-Lemma 5.24(b) [mit $A = F'(x)$ und $B = F'(x^*)^{-1}$] folgt, dass $F'(x)$ für $x \in U_\delta(x^*)$ regulär ist, und es gilt

$$\|F'(x)^{-1}\| \le \frac{\|F'(x^*)^{-1}\|}{1 - \|I - F'(x^*)^{-1}F'(x)\|} \le 2\,\|F'(x^*)^{-1}\| =: c.$$

(b) Da F in x^* diffbar ist, existiert zum selben Wert ε wie oben ein $\delta > 0$ mit

$$\|F(x) - F(x^*) - F'(x^*)(x - x^*)\| \le \varepsilon\,\|x - x^*\| \quad \text{für alle } x \in U_\delta(x^*).$$

Deshalb gilt für alle $x \in U_\delta(x^*)$ nach Dreiecksungleichung

$$\begin{aligned}\|F(x)\| &\ge \|F'(x^*)(x - x^*)\| - \|F(x) - \underbrace{F(x^*)}_{=0} - F'(x^*)(x - x^*)\| \\ &\ge \sigma_{\min}\big(F'(x^*)\big)\,\|x - x^*\| - \varepsilon\,\|x - x^*\| \\ &= \frac{1}{\|F'(x^*)^{-1}\|}\|x - x^*\| - \varepsilon\,\|x - x^*\| = \varepsilon\,\|x - x^*\|,\end{aligned}$$

und die Behauptung folgt mit $\beta = \varepsilon$.

Lemma 5.26.

Es sei $F : \mathbb{R}^n \to \mathbb{R}^n$ eine C^1-Funktion und $x^* \in \mathbb{R}^n$. Für alle $\varepsilon > 0$ existiert $\delta > 0$ mit

$$\|F(x) - F(x^*) - F'(x)(x - x^*)\| < \varepsilon\,\|x - x^*\|$$

für alle $\|x - x^*\| < \delta$. Kurz schreibt man auch:

$$\|F(x) - F(x^*) - F'(x)(x - x^*)\| = o\big(\|x - x^*\|\big).$$

Beweis. Es sei $\varepsilon > 0$ gegeben. Aus der Dreiecksungleichung ergibt sich

$$\begin{aligned}&\|F(x) - F(x^*) - F'(x)(x - x^*)\| \\ &\qquad \le \|F(x) - F(x^*) - F'(x^*)(x - x^*)\| + \|F'(x^*) - F'(x)\|\|x - x^*\|.\end{aligned}$$

Da F nach Voraussetzung in x^* diffbar ist, existiert $\delta_1 > 0$ mit

$$\|F(x) - F(x^*) - F'(x^*)(x - x^*)\| < \frac{\varepsilon}{2}\,\|x - x^*\|$$

für alle $\|x - x^*\| < \delta_1$. Andererseits ist F' stetig in x^*, sodass $\delta_2 > 0$ existiert mit

$$\|F'(x^*) - F'(x)\| < \frac{\varepsilon}{2}$$

für alle $\|x - x^*\| < \delta_2$. Mit $\delta := \min\{\delta_1, \delta_2\}$ folgt die Behauptung.

Ende 9. V

§ 5.4.2 Das lokale Newton-Verfahren für $F(x) = 0$

Wir können nun einen Konvergenzsatz für das lokale Newton-Verfahren beweisen:

Satz 5.27 (Konvergenzsatz für das lokale Newton-Verfahren)**.**

Es sei $F : \mathbb{R}^n \to \mathbb{R}^n$ eine C^1-Funktion und $x^* \in \mathbb{R}^n$ ein Punkt mit $F(x^*) = 0$ und $F'(x^*)$ regulär. Dann existiert eine Umgebung $U_\delta(x^*)$ von x^*, sodass gilt:

(a) x^* ist die einzige Nullstelle von F in $U_\delta(x^*)$.

(b) Für jeden Startwert $x_0 \in U_\delta(x^*)$ ist das lokale Newton-Verfahren wohldefiniert und erzeugt eine Folge $\{x_k\}$, die gegen x^* konvergiert.

(c) Die Konvergenzrate ist q-superlinear.

(d) Ist F' Lipschitz-stetig in $U_\delta(x^*)$, so ist die Konvergenzrate q-quadratisch.

Beachte: Die Konvergenzanalysis wird hier in der Euklidischen Norm aufgeschrieben. Wie in Übung 1, Aufgabe 2 gesehen, überträgt sich die q-superlineare bzw. q-quadratische Konvergenz jedoch auch auf die (benutzerdefinierte) M-Norm.

Beweis.

(a) Nach Lemma 5.25 (b) gibt es ein $\delta_0 > 0$, soadss x^* die einzige Nullstelle von F auf $U_{\delta_0}(x^*)$ ist.

(b) Nach Lemma 5.25 (a) existieren $\delta_1 > 0$ und $c > 0$, sodass $F'(x)$ für alle $x \in U_{\delta_1}(x^*)$ regulär ist mit

$$\|F'(x)^{-1}\| \le c = 2\,\|F(x^*)^{-1}\|. \tag{$*$}$$

Nach Lemma 5.26 existiert zu $\varepsilon = 1/(2c)$ ein $\delta_2 > 0$ mit

$$\|F(x) - F(x^*) - F'(x)(x - x^*)\| \le \frac{1}{2c}\|x - x^*\|$$

für alle $x \in U_{\delta_2}(x^*)$. Setze $\delta := \min\{\delta_0, \delta_1, \delta_2\}$ und wähle $x_0 \in U_\delta(x^*)$ beliebig. Dann ist der Schritt $x_1 := x_0 - F'(x_0)^{-1}F(x_0)$ wohldefiniert, und es gilt

$$\begin{aligned}
\|x_1 - x^*\| &= \|x_0 - x^* - F'(x_0)^{-1}F(x_0)\| \\
&= \|F'(x_0)^{-1}\big[F'(x_0)(x_0 - x^*) - F(x_0) + \overbrace{F(x^*)}^{=0}\big]\| \\
&\le \|F'(x_0)^{-1}\|\|F(x_0) - F(x^*) - F'(x_0)(x_0 - x^*)\| \\
&\le c\,\frac{1}{2c}\|x_0 - x^*\| = \frac{1}{2}\|x_0 - x^*\|,
\end{aligned}$$

also liegt auch x_1 wieder in $U_\delta(x^*)$. Per Induktion ist x_k wohldefiniert, gehört zu $U_\delta(x^*)$, und $x_k \to x^*$ q-linear.

(c) Wir stellen zunächst eine Gleichung für den Fehler auf:[29]

$$\begin{aligned}
x_{k+1} - x^* &= x_k - x^* - F'(x_k)^{-1}\big(F(x_k) - F(x^*)\big) \\
&= F'(x_k)^{-1}\big[F'(x_k)(x_k - x^*) - \big(F(x_k) - F(x^*)\big)\big] \\
&= F'(x_k)^{-1}\Big[F'(x_k)(x_k - x^*) - \int_0^1 F'(x_k + t\,(x^* - x_k))(x_k - x^*)\,\mathrm{d}t\Big] \\
&= F'(x_k)^{-1}\Big[\int_0^1 F'(x_k) - F'(x_k + t\,(x^* - x_k))\,\mathrm{d}t\Big](x_k - x^*).
\end{aligned}$$

Daraus erhalten wir folgende wichtige Abschätzung:

$$\|x_{k+1} - x^*\| \leq \big\|F'(x_k)^{-1}\big\| \underbrace{\int_0^1 \overbrace{\big\|F'(x_k) - F'(x_k + t\,(x^* - x_k))\big\|}^{=:D_k(t)}\,\mathrm{d}t}_{=:I_k} \|x_k - x^*\|. \quad (**)$$

Wegen $x_k \to x^*$ gilt $x_k + t\,(x^* - x_k) \to x^*$ gleichmäßig auf $t \in [0,1]$. Außerdem ist F' stetig. Zu jedem $\varepsilon > 0$ existiert also ein Index $k_0 \in \mathbb{N}$ mit

$$\big\|D_k(t)\big\| \leq \varepsilon \quad \text{für alle } k \geq k_0 \text{ und alle } t \in [0,1].$$

$$\Rightarrow \quad 0 \leq I_k = \int_0^1 \big\|D_k(t)\big\|\,\mathrm{d}t \leq \varepsilon \quad \text{für alle } k \geq k_0.$$

Das bedeutet aber: $I_k \to 0$. Jetzt liefern $(*)$ und $(**)$:

$$\|x_{k+1} - x^*\| \leq c\, I_k\, \|x_k - x^*\|,$$

also die q-superlineare Konvergenz.

(d) Da x_k und $x_k + t\,(x^* - x_k)$ für alle $t \in [0,1]$ in $U_\delta(x^*)$ liegen, können wir das Integral unter den stärkeren Voraussetzungen besser abschätzen:

$$I_k = \int_0^1 \big\|F'(x_k) - F'(x_k + t\,(x^* - x_k))\big\|\,\mathrm{d}t \leq \int_0^1 L\,t\,\|x^* - x_k\|\,\mathrm{d}t = \frac{L}{2}\|x_k - x^*\|.$$

Aus $(**)$ erhalten wir nun:

$$\|x_{k+1} - x^*\| \leq c\,\frac{L}{2}\|x_k - x^*\|^2.$$

Bemerkung 5.28 (zum lokalen Newton-Verfahren)**.**

(a) Das lokale Newton-Verfahren kann abbrechen, denn $F'(x^{(k)})$ muss nicht regulär sein, falls man außerhalb der (unbekannten) garantierten Konvergenzumgebung $U_\delta(x^*)$ startet.

(b) Das sogenannte **vereinfachte Newton-Verfahren**, bei dem an Stelle von $F'(x_k)$ die feste (invertierbare) Matrix $F'(x_0)$ verwendet wird, konvergiert noch lokal q-linear.

§ 5.4.3 Das lokale Newton-Verfahren in der Optimierung

Das Newton-Verfahren in der Optimierung lässt sich auf zwei verschiedene Weisen motivieren:

(a) Die notwendige Optimalitätsbedingung 1. Ordnung für **(FO)** lautet

$$\nabla f(x) = 0,$$

siehe Satz 3.2. Wenden wir zur Lösung dieser i. A. nichtlinearen Gleichung (Nullstellensuche) das Newton-Verfahren mit $F(x) = \nabla f(x)$ und $F'(x) = \nabla^2 f(x)$ an, so erhalten wir die Iterationsvorschrift

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k). \quad (5.22)$$

[29] **Beachte:** Unter dem Integral stehen Matrizen.

(b) Im aktuellen Iterationspunkt x_k ersetzen wir **(FO)** durch die Minimierung des **quadratischen Ersatzmodells** (Taylorpolynoms)

$$q_k(d) = f(x_k) + f'(x_k)\,d + \frac{1}{2}\,d^\top H_k\,d. \tag{5.2}$$

mit der Wahl $H_k = \nabla^2 f(x_k)$. Falls H_k positiv definit ist, so ist die eindeutige Lösung nach (5.4) charakterisiert durch das LGS

$$\nabla^2 f(x_k)\,d_k = -\nabla f(x_k).$$

Verwendet man die feste Schrittweite $\alpha_k = 1$, setzt also

$$x_{k+1} := x_k + d_k,$$

so ergibt sich wiederum die Iterationsvorschrift (5.22).

Bemerkung 5.29 (zum lokalen Newton-Verfahren).

(a) Satz 5.27 liefert die lokal q-superlineare bzw. q-quadratische Konvergenz des lokalen Newton-Verfahrens mit $F(x) = \nabla f(x)$ gegen eine Nullstelle x^* von F, also gegen einen stationären Punkt x^* von f. Dieser kann auch ein lokales Maximum oder ein Sattelpunkt von f sein, falls wir $\nabla^2 f(x^*)$ nur als regulär voraussetzen und nichts über die Definitheit annehmen.

(b) Ist $\nabla^2 f(x_k)$ spd, so ist die aus dem LGS

$$\nabla^2 f(x_k)\,d_k = -\nabla f(x_k)$$

erhaltene Newton-Richtung d_k eine Abstiegsrichtung für f, solange $f'(x_k) \neq 0$ ist, vergleiche (5.8):

$$f'(x_k)\,d_k = -\nabla f(x_k)^\top\,\nabla^2 f(x_k)^{-1}\,\nabla f(x_k) < 0.$$

Wegen der festen Schrittweite $\alpha_k = 1$ ist jedoch i. A. kein Abstieg in f garantiert, wenn x_k noch „weit“ von einem lokalen Minimum x^* entfernt ist.

(c) Das Newton-Verfahren ist im Gegensatz zum Gradientenverfahren invariant gegenüber affiner Skalierung.[30]

§ 5.4.4 Ein globalisiertes Newton-Verfahren in der Optimierung

In diesem Abschnitt wollen wir uns damit beschäftigen, wie man das (lokale) Newton-Verfahren globalisieren kann. Um den globalen Konvergenzsatz 5.11 anwenden zu können, benötigen wir also zulässige Suchrichtungen und Schrittweiten. Dies werden wir über eine (modifizierte) Winkelbedingung und eine Liniensuche erhalten. Diese Modifikationen sollen aber nicht die schnelle lokale Konvergenz verhindern!

Algorithmus 5.30 (globalisiertes Newton-Verfahren).
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: Funktionsauswertungen f und f' bzw. ∇f
Eingabe: Funktionsauswertungen f'' bzw. $\nabla^2 f$ (oder bei iterativer Lösung der LGS: Matrix-Vektor-Produkte mit $\nabla^2 f$)
Eingabe: spd Matrix M (oder Matrix-Vektor-Produkte mit M^{-1})
Eingabe: Parameter $\beta \in (0,1)$, $\sigma \in (0, 1/2)$ für die Armijo-Liniensuche
Eingabe: Parameter $\rho_{1,2} > 0$, $p > 0$ für die Globalisierungsstrategie

[30] siehe Übung 4, Aufgabe 15

Ausgabe: näherungsweise stationärer Punkt der Aufgabe (**FO**)

Setze $k := 0$
Setze $f_0 := f(x_0)$
Setze $r_0 := f'(x_0)^\top = \nabla f(x_0)$
Setze $d_0^G := -M^{-1} r_0$
Setze $\delta_0 := -r_0^\top d_0^G$ $\{\delta_0 = \|\nabla_M f(x_0)\|_M^2 = \|d_0^G\|_M^2\}$
while Abbruchkriterium nicht erfüllt **do**
  Versuche, das LGS $\nabla^2 f(x_k)\, d_k^N = -r_k$ zu lösen
  if LGS eindeutig lösbar **then**
    Setze $d_k := d_k^N$
  end if
  if LGS nicht oder nicht eindeutig lösbar oder

$$-f'(x_k)\, d_k^N \le \min\{\rho_1, \rho_2 \|d_k^N\|_M^p\} \|d_k^N\|_M^2 \tag{5.23}$$

  then
    Setze $d_k := d_k^G$
  end if
  Bestimme eine Schrittweite $\alpha_k > 0$ mittels Armijo-Liniensuche mit Backtracking (Algorithmus 5.13 oder Algorithmus 5.16), angewendet auf $\varphi(\alpha) := f(x_k + \alpha\, d_k)$ und mit Startschrittweite $\alpha_{k,0} = 1$ $\{\varphi(0) = f_k$ und $\varphi'(0) = -\delta_k$ (falls $d_k = d_k^G$) bzw. $\varphi'(0) = r_k^\top d_k^N$ (falls $d_k = d_k^N$) sind bekannt$\}$
  Setze $x_{k+1} := x_k + \alpha_k\, d_k$
  Setze $f_{k+1} := f(x_{k+1})$ {ist aus der Liniensuche bereits bekannt}
  Setze $r_{k+1} := f'(x_{k+1})^\top = \nabla f(x_{k+1})$
  Setze $d_{k+1}^G := -M^{-1} r_{k+1}$
  Setze $\delta_{k+1} := -r_{k+1}^\top d_{k+1}^G$ $\{\delta_{k+1} = \|\nabla_M f(x_{k+1})\|_M^2 = \|d_{k+1}^G\|_M^2\}$
  Setze $k := k + 1$
end while
return x_k

Die Grundidee von Algorithmus 5.30 besteht also darin, in jedem Schritt ggf. die Richtung des negativen M-Gradienten d_k^G als Ersatz zu verwenden, falls die Newton-Richtung d_k^N entweder nicht eindeutig definiert ist oder diese keine hinreichend gute (oder überhaupt keine) Abstiegsrichtung ist. Als eine solche ungeeignete Richtung bezeichnen wir d_k^N dann, wenn

$$-f'(x_k)\, d_k^N \le \min\{\rho_1, \rho_2 \|d_k^N\|_M^p\} \|d_k^N\|_M^2 \tag{5.23}$$

erfüllt ist. Dies interpretieren wir als Verletzung einer verallgemeinerten Winkelbedingung; vgl. die uns bekannte Winkelbedingung (5.7) für d_k^N:

$$-f'(x_k)\, d_k^N \ge \eta \|\nabla_M f(x_k)\|_M \|d_k^N\|_M.$$

Bemerkung 5.31 (zum globalisierten Newton-Verfahren)**.**

(a) Die Parameter $\rho_{1,2}$ und p werden oft recht klein gewählt, etwa

$$\rho_1 = \rho_2 = 10^{-6} \quad \text{und} \quad p = 10^{-1}.$$

(b) Wie in unseren bisherigen Verfahren auch liegt auch liegt der Vorkonditionierer i. A. nur in Form der Matrix-Vektor-Multiplikation mit M^{-1} vor. Zur Auswertung der Bedingung in (5.23) sieht es jedoch so aus, dass wir zusätzlich entweder M^{-1} oder M als Matrix benötigen oder aber Matrix-Vektor-Produkte mit M.

Es gibt jedoch einen eleganten Ausweg: Falls wir das Newton-System in Schritt 7 mittels CG-Verfahren (Algorithmus 4.18) lösen und dabei M als Vorkonditionierer und 0 als Startpunkt verwenden, dann können wir mit Hilfe der Rekursion (4.36) die (streng monoton wachsende) Norm $\|\cdot\|_M$ der Folge von Näherungslösungen für d_k^N ohne nennenswerten Aufwand bestimmen.

Weiterhin lässt sich das CG-Verfahren leicht so implementieren, dass es mit der Meldung „Gleichungssystem nicht oder nicht eindeutig lösbar“ abbricht, sobald es festgestellt, dass die Systemmatrix mindestens einen nichtpositiven Eigenwert aufweist. Unsere Implementierung `cg_quadratic.m` enthält diesen Test bereits.

(c) Es gibt natürlich noch andere Ansätze zur Globalisierung, bei denen die Newton-Richtung bei Bedarf durch andere Modifikationen zu einer hinreichend guten Abstiegsrichtung verändert wird.

Beispielsweise wird in (Geiger and Kanzow, 1999, S.93) und (Nocedal and Wright, 2006, S.51) bei unzureichender positiver Definitheit von $\nabla^2 f(x_k)$ ein Vielfaches der Einheitsmatrix (bzw. im vorkonditionierten Verfahren ein Vielfaches von M) hinzuaddiert, sodass

$$\left[\nabla^2 f(x_k) + \tau\, M\right] d_k = -\nabla f(x_k)$$

gelöst wird.

Ende 10. V

Wir wollen nun zunächst die globale Konvergenz von Algorithmus 5.30 zeigen.

Satz 5.32 (Konvergenzsatz für das globalisierte Newton-Verfahren)**.**

Es sei f eine C^2-Funktion. Weiter sei x^* ein Häufungspunkt der Iterierten $\{x_k\}$ von Algorithmus 5.30 und $\{x_k\}_K$ eine gegen x^* konvergente Teilfolge. Dann sind die Suchrichtungen $\{d_k\}_K$ und Schrittweiten $\{\alpha_k\}_K$ zulässig, und daher gilt $f'(x^*) = 0$.

Beweis. Wir weisen die Voraussetzungen des globalen Konvergenzsatz 5.11 nach, woraus dann $f'(x^*) = 0$ folgt. Dazu setzen wir

$$K_N := \{k \in K : d_k = d_k^N\} \quad \text{(Indexmenge der Newtonschritte)}$$
$$K_G := K \setminus K_N \quad \text{(Indexmenge der Gradientenschritte)}.$$

Schritt (i): Wir zeigen zunächst die Zulässigkeit der Suchrichtungen, müssen also nachweisen:

$$\left\{\frac{f'(x_k)\, d_k}{\|d_k\|}\right\}_K \to 0 \quad \Rightarrow \quad \{f'(x_k)\}_K \to 0 \tag{5.6}$$

Für $k \in K_G$ gilt wegen $d_k = -M^{-1}\nabla f(x_k)$

$$-\frac{f'(x_k)\,d_k}{\|d_k\|_M} = \frac{\|\nabla_M f(x_k)\|_M^2}{\|\nabla_M f(x_k)\|_M} = \|\nabla_M f(x_k)\|_M.$$

Aus der linken Seite von (5.6) folgt also $\{f'(x_k)\}_{K_G} \to 0$.

Für $k \in K_N$ gilt (siehe (5.23))

$$-\frac{f'(x_k)\,d_k}{\|d_k\|_M} > \min\{\rho_1, \rho_2\,\|d_k\|_M^p\}\,\|d_k\|_M \geq 0.$$

Somit folgt aus der linken Seite von (5.6) auch $\{\|d_k\|_M\}_{K_N} \to 0$. Weiter gilt

$$\|\nabla f(x_k)\|_{M^{-1}} = \|\nabla^2 f(x_k)\,d_k\|_{M^{-1}} \leq \underbrace{\|\nabla^2 f(x_k)\|_{M\to M^{-1}}}_{=\max_{d\neq 0}\frac{\|\nabla^2 f(x_k)\,d\|_{M^{-1}}}{\|d\|_M} \leq C} \|d_k\|_M \leq C\,\|d_k\|_M.$$

Dabei gilt die Beschränktheit mit einer Konstanten, weil $\{x_k\}_K$ konvergiert und somit $\{\nabla^2 f(x_k)\}_{K_N}$ beschränkt ist. Somit folgt nun auch $\{f'(x_k)\}_{K_N} \to 0$, und (5.6) ist gezeigt.

Schritt (*ii*): Die Zulässigkeit der (Armijo-)Schrittweiten folgt wegen

$$1\,\|d_k\|_M \geq \min\Big\{\frac{1}{C}, 1\Big\}\,\|\nabla_M f(x_k)\|_M \geq \min\Big\{\frac{1}{C}, 1\Big\}\,\frac{-f'(x_k)\,d_k}{\|d_k\|_M}$$

aus Lemma 5.15.

Der globale Konvergenzsatz 5.11 ist also anwendbar.

Wir zeigen nun, dass Algorithmus 5.30 unter gewissen Voraussetzungen in das lokale Newton-Verfahren übergeht, also dass

$$d_k = d_k^N \quad \text{und} \quad \alpha_k = 1 \tag{5.24}$$

für hinreichend große k gilt. Damit ist dann auch der lokale Konvergenzsatz 5.27 anwendbar, der die schnelle (mindestens q-superlineare) Konvergenz der *gesamten Folge* zeigt, sobald man sich hinreichend nahe an einem lokalen Minimum befindet, dass die hinreichenden Bedingungen 2. Ordnung erfüllt.

Satz 5.33 (Übergang zu schneller lokaler Konvergenz)**.**

Zusätzlich zu den Voraussetzungen von Satz 5.32 sei die Sublevelmenge $\mathcal{M}_f(x_0)$ kompakt, und es gelte $\nabla^2 f(x^*) \succ 0$. Dann konvergiert bereits die ganze Folge $\{x_k\}$ gegen x^*, und es gilt (5.24) für hinreichend große k. Folglich konvergiert $x_k \to x^*$ mindestens q-superlinear.

Beweis. Nach Satz 5.32 ist x^* als Grenzwert einer konvergenten Teilfolge von $\{x_k\}$ ein stationärer Punkt. Durch die positive Definitheit von $\nabla^2 f(x^*)$ ist x^* sogar ein isolierter stationärer Punkt, d. h., es existiert $\varepsilon_1 > 0$, sodass gilt:[31]

$$f'(x) \neq 0 \quad \text{für alle } x \in U_{\varepsilon_1}^M(x^*) \setminus \{x^*\}.$$

Schritt (*i*): Abschätzung von $\|d_k\|_M$
Wegen der C^2-Eigenschaft von f existieren weiterhin $\varepsilon_2 > 0$ und $\delta > 0$, sodass gilt:

$$\lambda_{\min}(\nabla^2 f(x); M) \geq \delta > 0 \quad \text{für alle } x \in U_{\varepsilon_2}^M(x^*).$$

Wir setzen jetzt $\varepsilon := \min\{\varepsilon_1, \varepsilon_2\}$. Für $x_k \in U_\varepsilon^M(x^*)$ ist also die Newton-Richtung

$$d_k^N = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

wohldefiniert, und es gilt (siehe gleich)

$$\|d_k^N\|_M \leq \frac{1}{\lambda_{\min}(\nabla^2 f(x_k); M)} \|\nabla f(x_k)\|_{M^{-1}} \leq \delta^{-1} \|\nabla f(x_k)\|_{M^{-1}}. \qquad (*)$$

Da das Verfahren evtl. auch die Gradientenrichtung $d_k^G = -M^{-1}\nabla f(x_k)$ als Suchrichtung d_k wählt, gilt insgesamt

$$\|d_k\|_M \leq \underbrace{\max\{1, \delta^{-1}\}}_{=:c} \|\nabla f(x_k)\|_{M^{-1}}.$$

Nachtrag zur Abschätzung $(*)$: Mit der Abkürzung $H := \nabla^2 f(x_k)$ gilt

$$\|d_k^N\|_M \leq \max_{r \neq 0} \frac{\|H^{-1} r\|_M}{\|r\|_{M^{-1}}} \|\nabla f(x_k)\|_{M^{-1}}.$$

Um das Maximum zu bestimmen, können wir z. B. die Cholesky-Faktorisierung $M = L\,L^\top$ benutzen:

$$\max_{r \neq 0} \frac{\|H^{-1} r\|_M}{\|r\|_{M^{-1}}} \overset{r = L\,s}{=} \max_{s \neq 0} \frac{\|H^{-1} L\, s\|_M}{\|L\, s\|_{M^{-1}}} = \max_{s \neq 0} \frac{\|L^\top H^{-1} L\, s\|_2}{\|s\|_2} = \lambda_{\max}(L^\top H^{-1} L).$$

Beachte: Die Eigenwerte von $L^\top H^{-1} L$ sind positiv wegen der positiven Definitheit von H und des Trägheitssatzes von Sylvester.
Die Eigenwerte von $L^\top H^{-1} L$ sind identisch mit den Eigenwerten des verallgemeinerten Eigenwertproblems zum Paar (H^{-1}, M^{-1}) und identisch mit den *Kehrwerten* der Eigenwerte des verallgemeinerten Eigenwertproblems zum Paar (H, M):

$$L^\top H^{-1} L\, x = \lambda\, x \quad \overset{x = L^{-1} y}{\Leftrightarrow} \quad H^{-1} y = \lambda\, M^{-1} y \quad \overset{y = M z}{\Leftrightarrow} \quad M\, z = \lambda\, H\, z$$
$$\Leftrightarrow \quad H\, z = \frac{1}{\lambda}\, M\, z.$$

Daher gilt abschließend wie behauptet

$$\|d_k^N\|_M \leq \max_{r \neq 0} \frac{\|H^{-1} r\|_M}{\|r\|_{M^{-1}}} \|\nabla f(x_k)\|_{M^{-1}} = \frac{1}{\lambda_{\min}(H; M)} \|\nabla f(x_k)\|_{M^{-1}}.$$

Schritt (ii): Konvergenz der gesamten Folge $\{x_k\}$
Wir haben gezeigt:

$$\alpha_k \|d_k\|_M \leq 1 \|d_k\|_M \overset{(i)}{\leq} c \|\nabla f(x_k)\|_{M^{-1}} \qquad (**)$$

für alle $x_k \in U_\varepsilon^M(x^*)$. Weiterhin gilt für alle hinreichend großen $k \in K$:

$$\|\nabla f(x_k)\|_{M^{-1}} \leq \frac{\varepsilon}{4\,c}. \qquad (***)$$

Es sei nun X die Menge aller Häufungspunkte der Folge $\{x_k\}$. Dann existiert $k_0 \in \mathbb{N}$, sodass

$$\operatorname{dist}^M(x_k, X) := \inf\{\|x_k - x\|_M : x \in X\} \leq \frac{\varepsilon}{4} \quad \text{für alle } k \geq k_0 \qquad (****)$$

gilt. (Andernfalls gäbe es eine Teilfolge $\{x_k\}_L$ mit $\operatorname{dist}^M(x_k, X) > \varepsilon/4$ für alle $k \in L$. Da diese Teilfolge in der kompakten Menge $\mathcal{M}_f(x_0)$ liegt, besitzt sie einen Häufungspunkt, der notwendigerweise auch in X enthalten ist; Widerspruch.) Es sei nun $k \in K$, $k \geq k_0$ so gewählt, dass

$$\|x_k - x^*\|_M \leq \frac{\varepsilon}{4}$$

gilt. Daraus folgt

$$\begin{aligned}
\|x_{k+1} - x^*\|_M &\leq \|x_k - x^*\|_M + \alpha_k \|d_k\|_M \\
&\leq \frac{\varepsilon}{4} + c\,\|\nabla f(x_k)\|_{M^{-1}} && \text{wegen } (**), \text{ beachte } x_k \in U_\varepsilon^M(x^*) \\
&\leq \frac{\varepsilon}{4} + \frac{\varepsilon}{4} = \frac{\varepsilon}{2}. && \text{wegen } (***).
\end{aligned}$$

Nach Voraussetzung ist x^* der einzige stationäre Punkt in $U_\varepsilon^M(x^*)$, also gilt $X \cap U_\varepsilon^M(x^*) = \{x^*\}$. Zusammen mit $(****)$ muss nun aber $\|x_{k+1} - x^*\| \leq \varepsilon/4$ gelten. Per Induktion gilt dies für die gesamte Restfolge. Damit konvergiert die gesamte Folge gegen x^*.

Schritt (iii): $d_k = d_k^N$ gilt für große k
Zunächst ist für hinreichend große Indizes k wegen $\lambda_{\min}(\nabla^2 f(x_k); M) \geq \delta > 0$ das Newton-System $\nabla^2 f(x_k)\, d_k^N = -\nabla f(x_k)$ eindeutig lösbar. Wir zeigen nun, dass für große Indizes k auch die verallgemeinerte Winkelbedingung erfüllt ist, dass also die Bedingung (5.23) in Algorithmus 5.30 *nicht* gilt. Es gilt nämlich

$$-f'(x_k)\, d_k^N = (d_k^N)^\top \nabla^2 f(x_k)\, d_k^N \geq \lambda_{\min}(\nabla^2 f(x_k); M)\, \|d_k^N\|_M^2 \geq \delta\, \|d_k^N\|_M^2.$$

Da $f'(x_k) \to f'(x^*) = 0$ gilt, konvergiert wegen $(*)$ auch $d_k^N \to 0$. Daher folgt weiter:

$$-f'(x_k)\, d_k^N \geq \delta\, \|d_k^N\|_M^2 > \rho_2\, \|d_k^N\|_M^{2+p} \geq \min\{\rho_1, \rho_2\, \|d_k^N\|_M^p\}\, \|d_k^N\|_M^2$$

für alle hinreichend großen Indizes k.[32] Daraus folgt, dass die Bedingung (5.23), die ungeeignete Newton-Richtungen identifiziert, nicht erfüllt ist. Also gilt $d_k^N = d_k$ für alle hinreichend großen Indizes k.

Schritt (iv): $\alpha = 1$ erfüllt die Armijo-Bedingung für große k
Für jedes hinreichend große k existiert $\xi_k \in (0, 1)$, sodass wir abschätzen können:

$$\begin{aligned}
&f(x_k + 1\, d_k) - f(x_k) - \sigma\, f'(x_k)\, d_k \\
&\quad = (1 - \sigma) \underbrace{f'(x_k)\, d_k}_{-d_k^\top \nabla^2 f(x_k)\, d_k} + \frac{1}{2}\, d_k^\top \nabla^2 f(x_k + \xi_k\, d_k)\, d_k \\
&\quad = -(1 - \sigma)\, d_k^\top \nabla^2 f(x_k)\, d_k + \frac{1}{2}\, d_k^\top \nabla^2 f(x_k + \xi_k\, d_k)\, d_k \\
&\quad = -\Big(1 - \sigma - \frac{1}{2}\Big)\, d_k^\top \nabla^2 f(x_k)\, d_k + \frac{1}{2}\, d_k^\top \big[\nabla^2 f(x_k + \xi_k\, d_k) - \nabla^2 f(x_k)\big]\, d_k \\
&\quad \leq -\Big(\frac{1}{2} - \sigma\Big)\, \delta\, \|d_k\|_M^2 + \frac{1}{2}\, \lambda_{\max}\big(\nabla^2 f(x_k + \xi_k\, d_k) - \nabla^2 f(x_k); M\big)\, \|d_k\|_M^2 \\
&\quad \leq 0,
\end{aligned}$$

da $\nabla^2 f$ stetig ist und $\sigma \in (0, 1/2)$ gilt.

Analog kann man auch zeigen, dass $\alpha = 1$ die strengen Krümmungsbedingungen (5.17) für hinreichend große Indizes k erfüllt.

Ende 11. V

§ 5.5 Newtonartige Verfahren

Literatur: (Geiger and Kanzow, 1999, Kapitel 10–12) und (Ulbrich and Ulbrich, 2012, Kapitel 11–13)

Nach Konstruktion ist das globalisierte Newton-Verfahren (Algorithmus 5.30) dem Gradientenverfahren (Algorithmus 5.23) überlegen. Jedoch steht in vielen Aufgabenstellungen die Hessematrix $\nabla^2 f$ nicht zur Verfügung oder ist aufwändig zu berechnen/aufzustellen. Falls man das Newton-System $\nabla^2 f(x_k)\, d_k^N = -\nabla f(x_k)$ iterativ löst (dazu später mehr), werden immerhin Matrix-Vektor-Produkte mit $\nabla^2 f(x_k)$ benötigt, die ebenfalls nicht immer zur Verfügung stehen. Und selbst wenn zweite Ableitungen verfügbar sind, kann es noch immer zu aufwändig sein, das Newton-System exakt zu lösen.

Beide Anliegen können wir gemeinsam behandeln. Wir betrachten dazu Methoden, die im Ersatzmodell (5.2) die Matrix H_k geeignet wählen und schließlich das LGS

$$H_k\, d_k = -\nabla f(x_k) \tag{5.4}$$

ggf. nur inexakt lösen; man löst also effektiv

$$H_k\, d_k = -\nabla f(x_k) + \zeta_k \tag{5.25}$$

mit einem (implizit definierten) Residuum ζ_k. Dafür wird man in einem konkreten Verfahren üblicherweise eine geforderte Toleranz der Form $\|\zeta_k\|_{M^{-1}} \le \varepsilon_k$ angeben. Wir betrachten als Ausgangspunkt zunächst ein lokales Verfahren (ohne Liniensuche).

Algorithmus 5.34 (allgemeines lokales newtonartiges Verfahren)**.**
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: Modell-Hessematrix $H_0 \in \mathbb{R}^{n\times n}$ spd
Eingabe: Funktionsauswertungen f und f' bzw. ∇f
Eingabe: Methode zur Bestimmung der (spd) Modell-Hessematrizen H_k
Ausgabe: näherungsweise stationärer Punkt der Aufgabe (**FO**)

```
setze k := 0
while Abbruchkriterium nicht erfüllt do
    Bestimme eine Modell-Hessematrix H_k
    Bestimme d_k als eine inexakte Lösung von H_k d_k = −∇f(x_k)
    (also gilt H_k d_k = −∇f(x_k) + ζ_k mit einem Residuum ζ_k)
    Setze x_{k+1} := x_k + d_k
    Setze k := k + 1
end while
return x_k
```

Es stellen sich folgende Fragen:

[31] Benutze Satz 5.27 (a), angewendet auf $F(x) := \nabla f(x)$.

[32] Die Verwendung der Potenz $2 + p$ heilt hier gewissermaßen die Unkenntnis der Konstanten δ, die man daher im Test auf die Qualität der Newton-Richtung d_k^N nicht verwenden kann.

- Welche Anforderungen müssen H_k und ζ_k erfüllen, um schnelle („newtonartige“, also q-superlineare) lokale Konvergenz zu erhalten?
- Welche praktischen Möglichkeiten gibt es, die Matrix H_k und eine Schranke für das Residuum ζ_k zu wählen, vor allem im Hinblick auf den erforderlichen Aufwand?

Wie bereits beim lokalen Newton-Verfahren (Satz 5.27) betrachten wir Algorithmus 5.34 zunächst etwas allgemeiner im Kontext der Nullstellensuche einer C^1-Funktion $F : \mathbb{R}^n \to \mathbb{R}^n$. In diesem Sinne ist H_k ein Ersatz für die Jacobimatrix $F'(x_k)$. Die H_k sollen zunächst nur regulär sein (nicht notwendig spd). Die Iterationsvorschrift lautet

$$\begin{aligned} &\text{Löse} \quad H_k\, d_k = -F(x_k) \quad \text{inexakt} \\ &\text{Setze} \quad x_{k+1} := x_k + d_k. \end{aligned} \tag{5.26}$$

Lemma 5.35 (Charakterisierung von schneller lokaler Konvergenz)**.**

Es sei $F : \mathbb{R}^n \to \mathbb{R}^n$ eine C^1-Funktion. Die Folge $\{x_k\}$ sei durch die Iterationsvorschrift (5.26) erzeugt. Sie konvergiere gegen einen Punkt x^*, für den $F'(x^*)$ regulär ist. Dann sind die folgenden Aussagen äquivalent.

(a) $\{x_k\}$ konvergiert q-superlinear, und es gilt $F(x^*) = 0$.

(b) $\|F(x_k) + F'(x_k)\,(x_{k+1} - x_k)\| = o(\|x_{k+1} - x_k\|)$.

(b') $\|F(x_k) + F'(x_k)\,(x_{k+1} - x_k)\| = o(\|x_k - x^*\|)$.

(c) $\|F(x_k) + F'(x^*)\,(x_{k+1} - x_k)\| = o(\|x_{k+1} - x_k\|)$.

Beachte: Dieses Lemma bringt also die schnelle lokale Konvergenz in Verbindung damit, dass die mittels (5.26) erzeugte Folgenglieder gleichzeitig noch hinreichend genaue Lösungen der „echten“ Newton-Systeme darstellen.[33] Auf die Wahl der Norm kommt es hier nicht an, wir können also auch mit der Euklidischen Norm $\|\cdot\|$ an Stelle von $\|\cdot\|_{M^{-1}}$ (für die Terme auf der linken Seite) bzw. $\|\cdot\|_M$ (rechte Seite) arbeiten.

Die o-Notation (**Landau-Symbol**) erklären wir am Beispiel der Aussage (b). Sie bedeutet: Für alle $\varepsilon > 0$ existiert $k_0 \in \mathbb{N}$, sodass gilt:

$$\|F(x_k) + F'(x_k)\,(x_{k+1} - x_k)\| \le \varepsilon\, \|x_{k+1} - x_k\| \quad \text{für alle } k \ge k_0.$$

Oft sagt man kurz: „$F(x_k) + F'(x_k)\,(x_{k+1} - x_k)$ konvergiert schneller gegen null als $x_{k+1} - x_k$.“

Beachte: Die q-superlineare Konvergenz von $\{x_k\}$ gegen x^* kann auch als $\|x_{k+1} - x^*\| = o(\|x_k - x^*\|)$ ausgedrückt werden.

Beweis. Vorüberlegung: Aufgrund der C^1-Eigenschaft von F und der Voraussetzung, dass $F'(x^*)$ regulär ist, sind in einer Umgebung von x^* sowohl $\|F'(x)\| \le C$ als auch $\|F'(x)^{-1}\| \le 1/c$ beschränkt mit geeigneten Konstanten $c, C > 0$, vgl. Lemma 5.25. Aus dem Satz von Taylor folgt weiter

$$F(x_{k+1}) = F(x^*) + F'\big(x^* + \xi_k\,(x_{k+1} - x^*)\big)\,(x_{k+1} - x^*)$$

mit einem $\xi_k \in (0, 1)$. Aus beidem zusammen erhalten wir für hinreichend große k

$$c\,\|x_{k+1} - x^*\| \le \|F(x_{k+1}) - F(x^*)\| \le C\,\|x_{k+1} - x^*\|. \tag{$*$}$$

[33] Das exakte Newton-Verfahren erfüllt ja $F(x_k) + F'(x_k)\,(x_{k+1} - x_k) = 0$.

Nochmals mit dem Satz von Taylor erhalten wir

$$\begin{aligned} F(x_{k+1}) &= F(x_k) + F'\big(x_k + \widehat{\xi}_k\,(x_{k+1}-x_k)\big)(x_{k+1}-x_k) \quad \text{mit } \widehat{\xi}_k \in (0,1) \\ &= F(x_k) + F'(x_k)(x_{k+1}-x_k) + \big[F'\big(x_k + \widehat{\xi}_k\,(x_{k+1}-x_k)\big) - F'(x_k)\big](x_{k+1}-x_k), \end{aligned}$$

also auch

$$\begin{aligned} &\|F(x_{k+1}) - F(x_k) - F'(x_k)(x_{k+1}-x_k)\| \\ &\qquad\qquad \le \|F'\big(x_k + \widehat{\xi}_k\,(x_{k+1}-x_k)\big) - F'(x_k)\|\,\|x_{k+1}-x_k\|. \end{aligned}$$

Wie im Beweis von Lemma 5.21 können wir jetzt die gleichmäßige Stetigkeit von F' „in der Nähe der $\{x_k\}$“ nutzen. Daraus ergibt sich, dass für jedes $\varepsilon > 0$ ein Index $k_0 \in \mathbb{N}$ existiert, sodass gilt:

$$\|F(x_{k+1}) - F(x_k) - F'(x_k)(x_{k+1}-x_k)\| \le \varepsilon\,\|x_{k+1}-x_k\|$$

für alle $k \ge k_0$, kurz:

$$\|F(x_{k+1}) - F(x_k) - F'(x_k)(x_{k+1}-x_k)\| = o(\|x_{k+1}-x_k\|). \tag{$**$}$$

(a) $\Rightarrow$ (b) und (b'): Wegen der q-superlinearen Konvergenz folgt für große k

$$\|x_k - x^*\| \le \|x_k - x_{k+1}\| + \|x_{k+1} - x^*\| \le \|x_k - x_{k+1}\| + \frac{1}{2}\,\|x_k - x^*\|,$$

also $\|x_k - x^*\| \le 2\,\|x_{k+1} - x_k\|$. Analog folgt für große k auch

$$\|x_{k+1} - x_k\| \le \|x_{k+1} - x^*\| + \|x^* - x_k\| \le 1\,\|x_k - x^*\| + \|x^* - x_k\|,$$

also $\|x_{k+1} - x_k\| \le 2\,\|x_k - x^*\|$.
Mit anderen Worten: Die Größen $\|x_{k+1} - x_k\|$ und $\|x_k - x^*\|$ „kontrollieren sich gegenseitig“ für hinreichend große k.

Wir können nun abschätzen:

$$\begin{aligned} &\|F(x_k) + F'(x_k)\,(x_{k+1}-x_k)\| \\ &\quad \le \|F(x_{k+1}) - F(x_k) - F'(x_k)(x_{k+1}-x_k)\| + \|F(x_{k+1}) - \underbrace{F(x^*)}_{=0}\| \\ &\quad \overset{(**)}{=} o(\|x_{k+1}-x_k\|) + \|F(x_{k+1}) - F(x^*)\|. \end{aligned}$$

Wegen $(*)$ und der q-superlinearen Konvergenz von $\{x_k\}$ gilt weiter:

$$\|F(x_{k+1}) - F(x^*)\| \le C\,\|x_{k+1} - x^*\| = o(\|x_k - x^*\|).$$

Da wir eben gesehen haben, dass $\|x_k - x^*\|$ und $\|x_{k+1} - x_k\|$ „gleich schnell“ konvergieren, schließen wir

$$\begin{aligned} &\|F(x_k) + F'(x_k)\,(x_{k+1}-x_k)\| \\ &\qquad = o(\|x_k - x^*\|) + o(\|x_{k+1} - x_k\|) = o(\|x_{k+1} - x_k\|) = o(\|x_k - x^*\|). \end{aligned}$$

(b) oder (b') $\Rightarrow$ (a): Wir schätzen ab:

$$\begin{aligned} &\|F(x_{k+1})\| \\ &\quad \le \|F(x_{k+1}) - F(x_k) - F'(x_k)\,(x_{k+1}-x_k)\| + \|F(x_k) + F'(x_k)(x_{k+1}-x_k)\| \\ &\quad \overset{(**)}{=} o(\|x_{k+1}-x_k\|) + \|F(x_k) + F'(x_k)(x_{k+1}-x_k)\|. \end{aligned}$$

Nach Voraussetzung (b) oder (b') ist der zweite Term entweder ebenfalls $o(\|x_{k+1} - x_k\|)$ oder $o(\|x_k - x^*\|)$. In jedem Fall aber gilt

$$\|F(x_{k+1})\| \leq o(\|x_{k+1} - x_k\|) + o(\|x_k - x^*\|).$$

Damit folgt $F(x_{k+1}) \to 0$ und somit $F(x^*) = 0$.

Für jedes $\varepsilon > 0$ und insbesondere für jedes $\varepsilon \in (0, c)$ gilt für hinreichend große Indizes k:

$$\begin{aligned} c\,\|x_{k+1} - x^*\| \overset{(*)}{\leq} \|F(x_{k+1})\| &\leq \varepsilon\,\|x_{k+1} - x_k\| + \varepsilon\,\|x_k - x^*\| \\ &\leq \varepsilon\,\|x_{k+1} - x^*\| + 2\,\varepsilon\,\|x_k - x^*\|, \end{aligned}$$

also

$$\|x_{k+1} - x^*\| \leq \frac{2\,\varepsilon}{c - \varepsilon}\,\|x_k - x^*\|.$$

Da $2\,\varepsilon/(c - \varepsilon)$ für $\varepsilon \searrow 0$ gegen null konvergiert, haben wir auch die q-superlineare Konvergenz von $\{x_k\}$ gezeigt, d. h. $\|x_{k+1} - x^*\| = o(\|x_k - x^*\|)$.

(b) $\Leftrightarrow$ (c): Für die Differenz der linken Seiten gilt

$$\|[F'(x_k) - F'(x^*)]\,(x_{k+1} - x_k)\| \leq \|F'(x_k) - F'(x^*)\|\,\|x_{k+1} - x_k\| = o(\|x_{k+1} - x_k\|).$$

Daraus folgt mit der Dreiecksungleichung die Äquivalenz von (b) und (c).

Wir wenden dieses Lemma nun auf Algorithmus 5.34 an. Dabei gilt $F(x) = \nabla f(x)$ und $F'(x) = \nabla^2 f(x)$ und

$$\begin{aligned} &F(x_k) + F'(x_k)\,(x_{k+1} - x_k) \\ &\quad = \nabla f(x_k) + \nabla^2 f(x_k)\,d_k \\ &\quad = \nabla f(x_k) + \nabla^2 f(x_k)\,d_k - \nabla f(x_k) - H_k\,d_k + \zeta_k \\ &\quad = \left[\nabla^2 f(x_k) - H_k\right] d_k + \zeta_k. \end{aligned}$$

Damit erhalten wir den folgenden Satz.

Satz 5.36 (Schnelle lokale Konvergenz newtonartiger Verfahren)**.**

Es sei $f : \mathbb{R}^n \to \mathbb{R}$ eine C^2-Funktion. Die Folge $\{x_k\}$ sei durch Algorithmus 5.34 erzeugt. Sie konvergiere gegen einen Punkt x^*, für den $\nabla^2 f(x^*)$ spd ist. Dann sind die folgenden Aussagen äquivalent.

(a) $\{x_k\}$ konvergiert q-superlinear, und es gilt $\nabla f(x^*) = 0$.

(b) $\|[\nabla^2 f(x_k) - H_k]\,d_k + \zeta_k\| = o(\|x_{k+1} - x_k\|)$.

(b') $\|[\nabla^2 f(x_k) - H_k]\,d_k + \zeta_k\| = o(\|x_k - x^*\|)$.

(c) $\|[\nabla^2 f(x^*) - H_k]\,d_k + \zeta_k\| = o(\|x_{k+1} - x_k\|)$.

Diese Bedingungen werden auch **Dennis-Moré-Bedingungen** genannt.[34] Sie zeigen, dass es für die q-superlineare Konvergenz nur darauf ankommt, dass $\nabla^2 f(x_k)\,d_k$ und $H_k\,d_k$ hinreichend gut übereinstimmen. Es ist also nicht erforderlich, dass H_k die *gesamte* Hessematrix $\nabla^2 f(x_k)$ gut approximiert!

[34] erstmals erwähnt in Dennis and Moré (1974)

Wir wollen nun zwei Typen von Verfahren kennenlernen, die Spezialisierungen von Algorithmus 5.34 sind. In einem Fall wird $H_k = \nabla^2 f(x_k)$ verwendet, im anderen Fall ist $\zeta_k = 0$.

§ 5.5.1 Inexaktes Newton-Verfahren

Bei den **inexakten Newton-Verfahren** verwendet man im Newtonsystem (5.4) die richtige Hessematrix $H_k = \nabla^2 f(x_k)$, löst das System aber nur inexakt, es bleibt also ein Residuum ζ_k:

$$\nabla^2 f(x_k)\, d_k = -\nabla f(x_k) + \zeta_k$$

Um die Genauigkeit der Lösung zu beschreiben, fordert man

$$\|\zeta_k\|_{M^{-1}} = \|\nabla^2 f(x_k)\, d_k + \nabla f(x_k)\|_{M^{-1}} \leq \eta_k \|\nabla f(x_k)\|_{M^{-1}} \tag{5.27}$$

mit $\eta_k \in (0, 1)$. Die Folge $\{\eta_k\}$ wird auch **forcing sequence** genannt.

Da ζ_k das Residuum zur inexakten Lösung d_k darstellt und andererseits das Residuum zum Nullvektor gerade $\nabla f(x_k)$ ist, gilt

$$\frac{\|\text{Residuum zu } d_k\|_{M^{-1}}}{\|\text{Residuum zu } 0\|_{M^{-1}}} = \frac{\|\zeta_k\|_{M^{-1}}}{\|\nabla f(x_k)\|_{M^{-1}}} \leq \eta_k. \tag{5.28}$$

Die *forcing sequence* bestimmt also die relative Genauigkeit der Lösung im Vergleich zum Nullvektor. Es ist klar, dass $\eta_k < 1$ sein sollte, da ansonsten $d_k = 0$ bereits eine hinreichend genaue Lösung darstellt.

Wir bezeichnen den Algorithmus 5.34 bei Verwendung von $H_k = \nabla^2 f(x_k)$ als **lokales inexaktes Newton-Verfahren**. Der Vollständigkeit halber geben wir das Verfahren hier nochmals an:

Algorithmus 5.37 (lokales inexaktes Newton-Verfahren)**.**
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: Funktionsauswertungen f, f' bzw. ∇f und f'' bzw. $\nabla^2 f$
Eingabe: spd Matrix M (oder Matrix-Vektor-Produkte mit M^{-1})
Eingabe: Methode zur Bestimmung der *forcing sequence* η_k
Ausgabe: näherungsweise stationärer Punkt der Aufgabe (**FO**)

1: setze $k := 0$
2: **while** Abbruchkriterium nicht erfüllt **do**
3: Bestimme d_k als eine inexakte Lösung von $\nabla^2 f(x_k)\, d_k = -\nabla f(x_k)$, sodass das Residuum $\zeta_k := \nabla^2 f(x_k)\, d_k + \nabla f(x_k)$ die Abschätzung

$$\|\zeta_k\|_{M^{-1}} \leq \eta_k \|\nabla f(x_k)\|_{M^{-1}} \tag{5.27}$$

erfüllt
4: Setze $x_{k+1} := x_k + d_k$
5: Setze $k := k + 1$
6: **end while**
7: **return** x_k

Beachte: Falls man $\eta_k \equiv 0$ setzt, so erhält man wieder das exakte (lokale) Newton-Verfahren.

Wir geben nun einen (lokalen) Konvergenzsatz für dieses Verfahren an.

Satz 5.38 (Konvergenzsatz für das lokale inexakte Newton-Verfahren)**.**

Es sei $f : \mathbb{R}^n \to \mathbb{R}$ eine C^2-Funktion. Der Punkt x^* sei stationär, und es sei $\nabla^2 f(x^*)$ spd. Die Folge $\{x_k\}$ sei mit dem lokalen inexakten Newton-Verfahren erzeugt, wobei die Residuen (5.27) erfüllen. Dabei gelte $\eta_k \leq \overline{\eta} < 1$ für alle $k \in \mathbb{N}_0$. Dann existiert eine Umgebung $U_\delta^M(x^*)$ von x^*, sodass gilt:

(a) x^* ist der einzige stationäre Punkt von f in $U_\delta^M(x^*)$.

(b) Für jeden Startwert $x_0 \in U_\delta^M(x^*)$ ist das lokale inexakte Newton-Verfahren wohldefiniert und erzeugt eine Folge $\{x_k\}$, die gegen x^* konvergiert. Die Konvergenzrate ist q-linear.

(c) Gilt zusätzlich $\eta_k \searrow 0$, dann ist die Konvergenzrate sogar q-superlinear.

(d) Ist $\nabla^2 f$ Lipschitz-stetig in $\mathcal{M}_f(x_0)$ und gilt neben $\eta_k \searrow 0$ sogar $\eta_k \leq c \, \|\nabla f(x_k)\|_{M^{-1}}$ mit einer Konstanten $c > 0$, dann konvergiert $\{x_k\}$ sogar q-quadratisch gegen x^*.

Beweisskizze. Aussage (a) folgt wie in Satz 5.27. Eine Anleitung zum Beweis der Aussage (b) findet man in (Geiger and Kanzow, 1999, Satz 10.3).

Für Aussage (c) nutzen wir Satz 5.36. (Wie bereits erwähnt, können wir dabei o. B. d. A. zu den $\|\cdot\|_{M^{-1}}$- bzw. $\|\cdot\|_M$-Normen übergehen.) Es gilt

$$\begin{aligned} \|(\underbrace{\nabla^2 f(x_k) - H_k}_{=0}) \, d_k + \zeta_k\|_{M^{-1}} &= \|\zeta_k\|_{M^{-1}} \\ &\leq \eta_k \, \|\nabla f(x_k)\|_{M^{-1}} \quad \text{nach Voraussetzung (5.27).} \end{aligned}$$

Wie im Beweis von Lemma 5.35, siehe $(*)$, gilt $\|\nabla f(x_k)\|_{M^{-1}} \leq C \, \|x_k - x^*\|_M$ für hinreichend große k:

$$\begin{aligned} &\leq \eta_k \, C \, \|x_k - x^*\|_M \\ &= o(\|x_k - x^*\|_M) \quad \text{wegen } \eta_k \searrow 0. \end{aligned}$$

(d): Die quadratische Konvergenz kann ähnlich wie in Satz 5.27 bewiesen werden.

Ende 12. V

Eine mögliche Regel für die Wahl von η_k, die wegen $\eta_k \searrow 0$ die lokale q-superlineare Konvergenz des inexakten Newton-Verfahrens garantiert, ist

$$\eta_k := \min\{\overline{\eta}, \|\nabla f(x_k)\|_{M^{-1}}^{\theta}\}$$

mit $\overline{\eta} < 1$ und $\theta \in (0, 1]$, z. B. $\overline{\eta} = 1/2$ und $\theta = 0.5$.[35]

Für den Rest des Abschnitts betrachten wir eine Möglichkeit, wie die inexakte Lösung des Newton-System praktisch realisiert und Algorithmus 5.37 gleichzeitig globalisiert werden kann. Auf folgende Dinge muss dabei geachtet werden:

(1) Das Newton-System $\nabla^2 f(x_k) \, d_k = -\nabla f(x_k)$ soll *iterativ* gelöst werden, wobei beim Erreichen der Genauigkeitsbedingung (5.27) gestoppt wird.

[35] Genauer erhält man dann sogar die q-superlineare Konvergenz mit der Rate $1 + \theta$.

(2) Die so erhaltene inexakte Newton-Richtung d_k^N sollte auf jedem Fall eine *Abstiegsrichtung* sein, damit Liniensuch-Algorithmen (§ 5.2) anwendbar sind.

(3) Die inexakte Newton-Richtung d_k^N wird anschließend wie im globalisierten exakten Newton-Verfahren (Algorithmus 5.30) auf hinreichend gute Abstiegseigenschaft überprüft. Ggf. wird auf die Gradientenrichtung zurückgefallen.

Die ersten beiden Ziele können durch die clevere Nutzung des (vorkonditionierten) CG-Verfahrens (Algorithmus 4.18) erreicht werden. Dieses wird auf das Gleichungssystem $A\, d_k^N = b$ mit

$$A = \nabla^2 f(x_k) \quad \text{und} \quad b = -\nabla f(x_k)$$

angewendet. Als Abbruchbedingung wird das relative Kriterium (4.22a) mit $\varepsilon_{\text{rel}} = \eta_k$ und als Startwert der Nullvektor verwendet. Falls das CG-Verfahren „ohne besondere Vorkommnisse“ durchläuft, dann ist — wegen (5.28) — die zurückgegebene Lösung eine hinreichend genaue Lösung des Newton-Systems im Sinne von (5.27).

Welche „besonderen Vorkommnisse“ können im CG-Verfahren auftreten? Einerseits könnte die maximale Iterationszahl überschritten werden, bevor die relative Toleranz erreicht wird. Andererseits aber ist die (symmetrische) Matrix A möglicherweise nicht positiv definit, d. h., die Funktion $\phi(z) := \frac{1}{2} z^\top A\, z - b^\top z$ besitzt mindestens eine Richtung $p \in \mathbb{R}^n$, $p \neq 0$ nicht-positiver Krümmung: $p^\top A\, p \leq 0$.[36] Nicht notwendigerweise wird eine solche Richtung im Verfahren auch angetroffen, denn möglicherweise wird die erforderliche Genauigkeit vorher erreicht.[37] Falls jedoch auf dem Weg zur inexakten Lösung eine Richtung p_ℓ auftritt, für die $\kappa_\ell := p_\ell^\top A\, p_\ell \leq 0$ gilt, so sollte reagiert werden: Im Fall $\kappa_\ell = 0$ würde eine Division durch null auftreten. Im Fall $\kappa_\ell < 0$ könnte das CG-Verfahren zwar prinzipiell weiterlaufen, jedoch verliert man ggf. die Eigenschaft, dass die Näherungslösungen Abstiegsrichtungen für die Zielfunktion f sind, wie man durch Beispiele bestätigen kann. Solange nur Richtungen positiver Krümmung $\kappa_\ell > 0$ auftreten, bleibt die Abstiegseigenschaft erhalten, siehe Lemma 5.40.

Aus diesem Grund verwendet man in Kombination mit dem inexakten Newton-Verfahren gerne eine Variante des CG-Verfahrens, die als **truncated CG-Verfahren** bezeichnet wird: Man startet mit Startschätzung null und iteriert, bis entweder die relative Abbruchbedingung (4.22a) erfüllt ist oder eine Suchrichtung nicht-positiver Krümmung auftritt. In jedem Fall gibt man die letzte Iterierte d_ℓ als Näherungslösung zurück. Der Test auf nicht-positive Krümmung ist in unserem CG-Verfahren `cg_quadratic.m` bereits implementiert.[38]

Der Vollständigkeit halber geben wir das truncated CG-Verfahren hier nochmals an. Um Verwechselungen mit den Iterierten und Suchrichtungen des äußeren inexakten Newton-Verfahren zu vermeiden, bezeichnen wir im Unterschied zu Algorithmus 4.18 die Iterierten mit d_ℓ und die Suchrichtungen mit p_ℓ.

[36] Wir vermeiden hier die Variablen x und d, die für die Iterierten und Suchrichtungen des äußeren inexakten Newton-Verfahrens reserviert sind.

[37] Selbst bei exakter Lösung ($\varepsilon_{\text{rel}} = 0$) werden bei bestimmten rechten Seiten b die Richtungen nicht-negativer Krümmung nicht verwendet.

[38] Dieser Fall äußert sich dort über den Rückgabewert `flag == 2`.

Algorithmus 5.39 (truncated CG-Verfahren)**.**
Eingabe: rechte Seite $b \in \mathbb{R}^n$
Eingabe: Matrix A (oder Matrix-Vektor-Produkte mit A); symmetrisch, aber nicht notwendig positiv definit
Eingabe: spd Matrix M (oder Matrix-Vektor-Produkte mit M^{-1})
Ausgabe: näherungsweise Lösung von $A\,d = b$

Setze $\ell := 0$
Setze $d_0 := 0$
Setze $r_0 := -b$
Setze $p_0 := -M^{-1} r_0$
Setze $\delta_0 := -r_0^\top p_0$ $\{\delta_0 = \|r_0\|^2_{M^{-1}}\}$
while Abbruchkriterium nicht erfüllt **do**
  Setze $q_\ell := A\,p_\ell$
  Setze $\kappa_\ell := p_\ell^\top q_\ell$
  if $\kappa_\ell > 0$ **then**
    Setze $\alpha_\ell := \delta_\ell / \kappa_\ell$
    Setze $d_{\ell+1} := d_\ell + \alpha_\ell\, p_\ell$
    Setze $r_{\ell+1} := r_\ell + \alpha_\ell\, q_\ell$
    Setze $p_{\ell+1} := -M^{-1} r_{\ell+1}$
    Setze $\delta_{\ell+1} := -r_{\ell+1}^\top p_{\ell+1}$ $\{\delta_{\ell+1} = \|r_{\ell+1}\|^2_{M^{-1}}\}$
    Setze $\beta_{\ell+1} := \delta_{\ell+1} / \delta_\ell$
    Setze $p_{\ell+1} := p_{\ell+1} + \beta_{\ell+1}\, p_\ell$
    Setze $\ell := \ell + 1$
  else
    Abbruch der **while**-Schleife
  end if
end while
return d_ℓ

Wir wollen noch beweisen, dass die Näherungslösung d_ℓ, die das truncated CG-Verfahren zurückgibt und die als inexakte Newton-Richtung d_k^N verwendet werden soll, tatsächlich wie gewünscht eine Abstiegsrichtung für die Zielfunktion f and der Stelle x_k ist. Das bedeutet, dass $\nabla f(x_k)^\top d_\ell < 0$ gilt oder äquivalent: $b^\top d_\ell > 0$.

Lemma 5.40 (Das truncated CG-Verfahren generiert Abstiegsrichtungen)**.**
Es sei $b \neq 0$, und die Iterierten $d_0, d_1, \ldots, d_\ell$, $\ell \geq 1$, seien von Algorithmus 5.39 erzeugt. Dann gilt:

(a) $b^\top M^{-1} r_j = 0$ für $j = 1, \ldots, \ell$.

(b) $b^\top p_j = \|r_j\|^2_{M^{-1}}$ für $j = 0, \ldots, \ell$.

(c) $b^\top d_\ell = \sum_{j=0}^{\ell-1} \alpha_j \|r_j\|^2_{M^{-1}}$ ist positiv und streng monoton wachsend in ℓ.

Beweis.

(a) Da der Nullvektor als Startwert verwendet wird, gilt $r_0 = A\,0 - b = -b$. Also gilt

$$b^\top M^{-1} r_j = -r_0^\top M^{-1} r_j = 0 \quad \text{für } j \geq 1$$

gemäß (4.31).

(b) Die erste Suchrichtung ist $p_0 = -M^{-1} r_0$, daher gilt

$$b^\top p_0 = r_0^\top M^{-1} r_0 = \|r_0\|^2_{M^{-1}}.$$

Weiter folgt per Induktion für $j \geq 0$:

$$\begin{aligned} b^\top p_{j+1} &= b^\top \left(-M^{-1} r_{j+1} + \beta_{j+1}\, p_j\right) \\ &= 0 + \beta_{j+1}\, b^\top p_j && \text{nach Teil (a)} \\ &= \frac{\|r_{j+1}\|^2_{M^{-1}}}{\|r_j\|^2_{M^{-1}}}\, b^\top p_j && \text{nach (4.29')} \\ &= \|r_{j+1}\|^2_{M^{-1}} && \text{nach Induktionsvoraussetzung.} \end{aligned}$$

(c) Da das Verfahren die Iterierten $d_0, d_1, \ldots, d_\ell$ für $\ell \geq 1$ erzeugt hat, sind die Zahlen $\kappa_0, \ldots, \kappa_{\ell-1}$ alle positiv. Damit ist auch $\alpha_j = \delta_j/\kappa_j > 0$ für $j = 0, \ldots, \ell - 1$. Folglich gilt

$$b^\top d_\ell = b^\top \sum_{j=0}^{\ell-1} \alpha_j\, p_j \overset{(b)}{=} \sum_{j=0}^{\ell-1} \alpha_j\, \|r_j\|^2_{M^{-1}}.$$

Die Residuen $r_0, \ldots, r_{\ell-1}$ sind alle $\neq 0$, sonst würde die Abbruchbedingung in Algorithmus 5.39 greifen. Daher ist der obige Ausdruck in ℓ streng monoton wachsend.

Bemerkung 5.41 (zum truncated CG-Verfahren)**.**

(a) Die Monotonie von $\nabla f(x_k)^\top d_\ell = -b^\top d_\ell$ in der Iterationsanzahl ℓ des truncated CG-Verfahrens bedeutet, dass die im Laufe der CG-Iterationen generierten Richtungen d_ℓ zunehmend bessere Abstiegseigenschaften aufweisen. Dies gilt, solange nur Suchrichtungen p_ℓ positiver Krümmung auftreten. Daher ist es sinnvoll, wie in Algorithmus 5.39 vorgesehen, das CG-Verfahren solange laufen zu lassen, bis entweder die vorgesehene Toleranz erreicht ist oder eine Suchrichtung nicht-positiver Krümmung auftritt.

(b) Es kann passieren, dass bereits im nullten Iterationsschritt $\kappa_0 \leq 0$ gilt. In diesem Fall wird $d_k^N = 0$ zurückgegeben, was als Suchrichtung im äußeren inexakten Newton-Verfahren natürlich ungeeignet ist. Die inexakte Newton-Richtung d_k^N wird aber ohnehin zu Zwecken der Globalisierung einem Qualitätstest unterzogen und ggf. auf die Gradientenrichtung $d_k^G = -M^{-1}\nabla f(x_k) = M^{-1} b$ zurückgefallen.

Wie bereits angedeutet, können zur Globalisierung des inexakten Newton-Verfahrens dieselben Techniken wie in Algorithmus 5.30 eingesetzt werden. Dies führt zu folgendem Verfahren.

Algorithmus 5.42 (globalisiertes inexaktes Newton-Verfahren)**.**
Eingabe: Startwert $x_0 \in \mathbb{R}^n$

Eingabe: Funktionsauswertungen f und f' bzw. ∇f
Eingabe: Funktionsauswertungen f'' bzw. $\nabla^2 f$ (oder Matrix-Vektor-Produkte mit $\nabla^2 f$)
Eingabe: spd Matrix M (oder Matrix-Vektor-Produkte mit M^{-1})
Eingabe: Methode zur Bestimmung der *forcing sequence* η_k
Eingabe: Parameter $\beta \in (0,1)$, $\sigma \in (0, 1/2)$ für die Armijo-Liniensuche
Eingabe: Parameter $\rho_{1,2} > 0$, $p > 0$ für die Globalisierungsstrategie
Ausgabe: näherungsweise stationärer Punkt der Aufgabe (**FO**)

Setze $k := 0$
Setze $f_0 := f(x_0)$
Setze $r_0 := f'(x_0)^\top = \nabla f(x_0)$
Setze $d_0^G := -M^{-1} r_0$
Setze $\delta_0 := -r_0^\top d_0^G$ $\{\delta_0 = \|\nabla_M f(x_0)\|_M^2 = \|d_0^G\|_M^2\}$
while Abbruchkriterium nicht erfüllt **do**
  Bestimme die inexakte Newton-Richtung d_k^N aus Algorithmus 5.39 mit relativer Toleranz η_k
  if

$$-f'(x_k)\, d_k^N \le \min\{\rho_1, \rho_2 \|d_k^N\|_M^p\} \|d_k^N\|_M^2 \tag{5.23}$$

  then
    Setze $d_k := d_k^G$
  end if
  Bestimme eine Schrittweite $\alpha_k > 0$ mittels Armijo-Liniensuche mit Backtracking (Algorithmus 5.13 oder Algorithmus 5.16), angewendet auf $\varphi(\alpha) := f(x_k + \alpha\, d_k)$ und mit Startschrittweite $\alpha_{k,0} = 1$ $\{\varphi(0) = f_k$ und $\varphi'(0) = -\delta_k$ (falls $d_k = d_k^G$) bzw. $\varphi'(0) = r_k^\top d_k^N$ (falls $d_k = d_k^N$) sind bekannt$\}$
  Setze $x_{k+1} := x_k + \alpha_k\, d_k$
  Setze $f_{k+1} := f(x_{k+1})$ {ist aus der Liniensuche bereits bekannt}
  Setze $r_{k+1} := f'(x_{k+1})^\top = \nabla f(x_{k+1})$
  Setze $d_{k+1}^G := -M^{-1} r_{k+1}$
  Setze $\delta_{k+1} := -r_{k+1}^\top d_{k+1}^G$ $\{\delta_{k+1} = \|\nabla_M f(x_{k+1})\|_M^2 = \|d_{k+1}^G\|_M^2\}$
  Setze $k := k + 1$
end while
return x_k

Bemerkung 5.43 (zum globalisierten inexakten Newton-Verfahren)**.**

(a) Zur Wahl der Globalisierungsparameter ρ_1, ρ_2 und p siehe Bemerkung 5.31.

(b) Die für die Auswertung der Qualitätsbedingung (5.23) erforderliche Größe $\|d_k^N\|_M^2$ kann — wie ebenfalls bereits in Bemerkung 5.31 gesagt — im (truncated) CG-Verfahren (Algorithmus 5.39) ohne zusätzlichen Aufwand und ohne Zugriff auf die Matrix M (oder Matrix-Vektor-Produkte mit M) mitberechnet werden.

Die globale Konvergenz des Verfahrens kann ähnlich wie in Satz 5.32 gezeigt werden (vgl. dazu auch (Geiger and Kanzow, 1999, Satz 10.5)), indem man die Zulässigkeit der Suchrichtungen und Schrittweiten auf einer konvergenten Teilfolge verifiziert. Den Übergang zu schneller lokaler Konvergenz kann man dann ähnlich wie in Satz 5.33 beweisen. Genauer zeigt man wieder, dass unter den Voraussetzungen von

Satz 5.33 für hinreichend große k

$$d_k = d_k^N \quad \text{und} \quad \alpha_k = 1 \tag{5.24}$$

gilt, und die Konvergenzrate (q-linear, q-superlinear oder sogar q-quadratisch) folgt dann je nach Vorgabe der *forcing sequence* mit Satz 5.38; siehe (Geiger and Kanzow, 1999, Satz 10.8).

Die beschriebene Kombination des (inexakten) Newton-Verfahrens mit dem (truncated) CG-Verfahren als inneren Löser wird oft auch als **(truncated) Newton-CG-Verfahren** bezeichnet. Da die Hesse-Matrix $\nabla^2 f(x_k)$ an den Iterierten nicht vollständig aufgestellt werden muss, sondern Matrix-Vektor-Produkte mit $\nabla^2 f(x_k)$ ausreichen, spricht man (etwas missverständlich) auch von einem **Hessematrix-freien Optimierungsverfahren** (*Hessian-free*).

Ende 13. V

§ 5.5.2 Quasi-Newton-Verfahren

Literatur: (Geiger and Kanzow, 1999, Kapitel 11) und (Ulbrich and Ulbrich, 2012, Kapitel 13)

Im Gegensatz zu dem inexakten Newton-Verfahren machen Quasi-Newton-Verfahren auf eine andere Art Gebrauch von der Freiheit newtonartiger Verfahren, vom reinen Newton-Schritt abzuweichen: Ein solches Verfahren wählt (auf eine bestimmte Art und Weise) eine symmetrische Approximation H_k der Hessematrix $\nabla^2 f(x_k)$, löst anschließend aber das LGS

$$H_k\, d_k = -\nabla f(x_k) \tag{5.4}$$

exakt. Dann wird ein Schritt in Richtung d_k (zunächst mit Schrittweite eins) gemacht. Für die Bestimmung der nächsten Matrix H_{k+1} nutzt man Informationen, die man im aktuellen Schritt gewonnen hat.

Betrachtet man eine Taylorentwicklung für ∇f, so erhält man mit einem $\xi_k \in (0,1)$

$$\nabla f(x_{k+1}) - \nabla f(x_k) = \nabla^2 f\big(x_k + \xi_k\,(x_{k+1} - x_k)\big)\,(x_{k+1} - x_k) \approx \nabla^2 f(x_{k+1})\,(x_{k+1} - x_k).$$

Daraus erhält man die sogenannte **Sekantenbedingung**

$$H_{k+1}\,(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k), \tag{5.29}$$

die auch **Quasi-Newton-Bedingung** genannt wird. Ein Verfahren, das diese Bedingung verwendet, um seine symmetrischen Ersatz-Hessematrizen $\{H_k\}$ zu generieren, heißt **Quasi-Newton-Verfahren**.

Die Bedingung (5.29) lässt sich auch wie folgt motivieren: Wie in (5.2) betrachten wir die beiden quadratischen Modelle an der Stelle x_k und x_{k+1}:

$$m_k(x) = f(x_k) + \nabla f(x_k)^\top (x - x_k) + \frac{1}{2}(x - x_k)^\top H_k\,(x - x_k) \tag{5.30a}$$

$$m_{k+1}(x) = f(x_{k+1}) + \nabla f(x_{k+1})^\top (x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^\top H_{k+1}\,(x - x_{k+1}). \tag{5.30b}$$

Nach Konstruktion stimmt der Gradient des Modells m_k mit dem Gradienten von f an der Stelle x_k überein. Man fordert, dass auch der Gradient des *neuen* Modells

mit dem Gradienten von f an der *alten* Stelle x_k übereinstimmt[39], also:

$$\begin{aligned} & \nabla m_{k+1}(x_k) = \nabla f(x_k) \\ \Leftrightarrow \quad & \nabla f(x_{k+1}) + H_{k+1}\,(x_k - x_{k+1}) = \nabla f(x_k), \quad \text{denn } H_{k+1} \text{ ist symmetrisch} \\ \Leftrightarrow \quad & (5.29). \end{aligned}$$

Diese Forderung ergibt also auch gerade die Sekantenbedingung.

Mit Hilfe von Satz 5.36 über die schnelle lokale Konvergenz newtonartiger Verfahren können wir nun charakterisieren, wann ein lokales[40] Quasi-Newton-Verfahren q-superlinear konvergiert.

Satz 5.44 (Schnelle lokale Konvergenz von Quasi-Newton-Verfahren)**.**

Die Folge $\{x_k\}$ sei mit dem lokalen newtonartigen Verfahren Algorithmus 5.34 erzeugt, wobei die Matrizen $\{H_k\}$ symmetrisch sind und (5.29) erfüllen und die Residuen $\zeta_k = 0$ sind. Die Folge $\{x_k\}$ konvergiere gegen einen Punkt x^*, für den $\nabla f^2(x^*)$ spd ist. Schließlich gelte $\|(H_{k+1} - H_k)\,d_k\| = o(\|d_k\|)$. Dann sind die Bedingungen aus Satz 5.36 erfüllt, und $\{x_k\}$ konvergiert q-superlinear gegen x^*.

Beachte: Auf die Wahl der Norm kommt es hier wieder nicht an, wir könnten den Satz also äquivalent auch mit $\|(H_{k+1} - H_k)\,d_k\|_{M^{-1}} = o(\|d_k\|_M)$ formulieren.

Beweis. Es gilt mit einer Taylorentwicklung und (5.29)

$$\begin{aligned} \|(\nabla^2 f(x_k) - H_k)\,d_k\| &\le \|(\nabla^2 f(x_k) - H_{k+1})\,d_k\| + \|(H_{k+1} - H_k)\,d_k\| \\ &= \|\nabla^2 f(x_k)\,d_k - \nabla f(x_{k+1}) + \nabla f(x_k)\| + o(\|d_k\|) \\ &= o(\|x_{k+1} - x_k\|) \end{aligned}$$

wegen der gleichmäßigen Stetigkeit von $\nabla^2 f$ „in der Nähe der $\{x_k\}$“, vgl. Beweis von Lemma 5.21.

Beachte: Die Konvergenz $\|H_{k+1} - H_k\| \to 0$ ist hinreichend, um die Voraussetzung $\|(H_{k+1} - H_k)\,d_k\| = o(\|d_k\|)$ zu erfüllen, denn:

$$\|(H_{k+1} - H_k)\,d_k\| \le \|H_{k+1} - H_k\|\,\|d_k\|.$$

In Quasi-Newton-Verfahren suchen wir nun ein Formel für H_{k+1}, sodass

- die Sekantenbedingung (5.29) erfüllt ist,
- die Matrix H_{k+1} „in der Nähe“ von H_k liegt[41] und
- die Matrix H_{k+1} spd ist.

Der letzte Punkt liefert, dass d_k garantiert eine Abstiegsrichtung ist und dass Algorithmus 5.34 auch ohne Globalisierung gute globale Konvergenzeigenschaften hat.

Zur Vereinfachung der Notation setzen wir nun

$$s_k := x_{k+1} - x_k \quad \text{und} \quad y_k := \nabla f(x_{k+1}) - \nabla f(x_k).$$

Ab sofort sehen wir die Verwendung einer Liniensuche vor, sodass

$$x_{k+1} := x_k + \alpha_k\,d_k$$

[39] an der Stelle x_{k+1} tut er dies wieder nach Konstruktion

[40] ohne Liniensuche, also mit Schrittweite 1; es gilt $x_{k+1} = x_k + d_k$

[41] um die Konvergenz $\|H_{k+1} - H_k\| \to 0$ bzw. $\|(H_{k+1} - H_k)\,d_k\| = o(\|d_k\|)$ erfüllen zu können

verwendet wird, also gilt $s_k = \alpha_k d_k$. Die Sekantenbedingung (5.29) lautet nun $H_{k+1} s_k = y_k$. Multipliziert man diese Gleichung mit s_k, so erhalten wir

$$0 < s_k^\top H_{k+1} s_k = y_k^\top s_k = \big(\nabla f(x_{k+1}) - \nabla f(x_k)\big)^\top (x_{k+1} - x_k) \tag{5.31}$$

als notwendige Bedingung für $H_{k+1} \succ 0$. Für gleichmäßg konvexe Funktionen f ist (5.31) immer erfüllt[42], ansonsten muss dies über die Liniensuche sichergestellt werden. Dazu bietet sich die Wolfe-Liniensuche an.[43]

Lemma 5.45. Es sei d_k eine Abstiegsrichtung für f an der Stelle x_k. Erfüllt $\alpha_k > 0$ die Krümmungsbedingung (5.16) mit $\tau < 1$, so ist auch (5.31) erfüllt.

Beweis. Die Krümmungsbedingung (5.16), ausgewertet an der Schrittweite α_k, ergibt

$$f'(x_{k+1})\, d_k \geq \tau\, f'(x_k)\, d_k > f'(x_k)\, d_k.$$

Die letztere Ungleichung folgt, da d_k eine Abstiegsrichtung ist. Es gilt also

$$y_k^\top d_k = \big(\nabla f(x_{k+1}) - \nabla f(x_k)\big)^\top d_k > 0,$$

und wegen $s_k = \alpha_k d_k$ mit positiver Schrittweite α_k folgt (5.31).

Es gibt (zumindest im Fall $n > 1$) unendlich viele Möglichkeiten, die Sekantenbedingung (5.29) zu erfüllen.[44] Verschiedene Quasi-Newton-Verfahren unterscheiden sich in der Konstruktionsidee, wie H_{k+1} aus den Daten H_k, y_k und s_k gebildet wird. Die geläufigsten Varianten geben wir nun an.

- **SR1** (Symmetric rank 1):

  Wir betrachten ein symmetrisches Rang-1-Update der Form[45]

  $$H_{k+1}^{\mathrm{SR1}} = H_k + \lambda\, u\, u^\top$$

  mit $\lambda \in \mathbb{R}$ und $u \in \mathbb{R}^n$. Durch die Sekantenbedingung (5.29) wird das Update (jedenfalls im Fall $H_k s_k \neq y_k$) eindeutig, und wir erhalten die Vorschrift[46]

  $$H_{k+1}^{\mathrm{SR1}} = H_k + \frac{(y_k - H_k s_k)\,(y_k - H_k s_k)^\top}{(y_k - H_k s_k)^\top s_k}. \tag{5.32}$$

  Der Nenner ist hier problematisch: Haben wir etwa einen vollen Schritt ($\alpha_k = 1$) gemacht, dann gilt $s_k = d_k$, und aus dem Quasi-Newton-System $H_k d_k = -\nabla f(x_k)$, das die Suchrichtung d_k bestimmt, erhalten wir

  $$(y_k - H_k s_k)^\top s_k = \nabla f(x_{k+1})^\top d_k \approx 0,$$

  falls $\alpha_k = 1$ ein gute Approximation des Minimierers in der Liniensuche darstellt. Weiterhin kann man auch nicht die positive Definitheit von H_{k+1}^{SR1} garantieren, selbst wenn H_k positiv definit war, da der Nenner in (5.32) auch negativ werden kann. (Dennoch hat das SR1-Verfahren seine Berechtigung, und zwar vor allem dann, wenn zur Globalisierung keine Liniensuche, sondern ein Trust-Region-Ansatz verwendet wird, siehe § 6.)

[42] Dies ist gerade die starke Monotonie des Gradienten.

[43] Für die bisherigen Verfahren war stets die einfachere Armijo-Liniensuche ausreichend.

[44] Denn (5.29) liefert nur n Bedingungen für die $n\,(n+1)/2$ Freiheitsgrade einer symmetrischen $n \times n$-Matrix.

[45] Der besseren Lesbarkeit wegen schreiben wir H_k statt H_k^{SR1}.

[46] siehe (Ulbrich and Ulbrich, 2012, S.67)

- **PSB** (Powell-symmetric-Broyden):

  Um H_{k+1} in der Nähe von H_k zu wählen, betrachten wir die Hilfsaufgabe

$$\begin{aligned} \text{Minimiere} \quad & \|H_{k+1} - H_k\|_F, \quad H_{k+1} \in \mathbb{R}^{n\times n}_{\text{sym}} \\ \text{sodass} \quad & (5.29) \text{ gilt.} \end{aligned} \tag{5.33}$$

  Hier ist $\|\cdot\|_F$ die Frobeniusnorm.[47] Man erhält die eindeutige Lösung[48]

$$\begin{aligned} H_{k+1}^{\text{PSB}} = H_k + & \frac{(y_k - H_k\, s_k)\, s_k^\top + s_k\, (y_k - H_k\, s_k)^\top}{s_k^\top s_k} \\ & - (y_k - H_k\, s_k)^\top s_k\, \frac{s_k\, s_k^\top}{(s_k^\top s_k)^2}. \end{aligned} \tag{5.34}$$

  Dies ist ein Rang-2-Update.[49] Auch hier hat man wieder Probleme mit der positiven Definitheit.

- **DFP** (Davidon-Fletcher-Powell):

  Hier betrachtet man einen gewichteten Abstand

$$\begin{aligned} \text{Minimiere} \quad & \|W^{-1}(H_{k+1} - H_k)\, W^{-1}\|_F, \quad H_{k+1} \in \mathbb{R}^{n\times n}_{\text{sym}} \\ \text{sodass} \quad & (5.29) \text{ gilt,} \end{aligned} \tag{5.35}$$

  wobei W eine reguläre Matrix mit der Eigenschaft $W^2\, s_k = y_k$ ist. Man erhält (unabhängig von W)

$$H_{k+1}^{\text{DFP}} = (I - \gamma_k\, y_k\, s_k^\top)\, H_k\, (I - \gamma_k\, s_k\, y_k^\top) + \gamma_k\, y_k\, y_k^\top, \tag{5.36}$$

  wobei $\gamma_k = (y_k^\top s_k)^{-1}$ ist. Dies ist ein Rang-2-Update (ausmultiplizieren!), und man kann die positive Definitheit garantieren, siehe Lemma 5.46.

- **BFGS** (Broyden-Fletcher-Goldfarb-Shanno):

  Analog zur DFP-Formel betrachtet man

$$\begin{aligned} \text{Minimiere} \quad & \|W\, (H_{k+1}^{-1} - H_k^{-1})\, W\|_F, \quad H_{k+1} \in \mathbb{R}^{n\times n}_{\text{sym}} \\ \text{sodass} \quad & (5.29) \text{ gilt.} \end{aligned} \tag{5.37}$$

  wobei wieder $W^2\, s_k = y_k$ gilt. Dies resultiert in der Rang-2-Update-Formel

$$H_{k+1}^{\text{BFGS}} = H_k - \frac{H_k\, s_k\, s_k^\top H_k}{s_k^\top H_k\, s_k} + \frac{y_k\, y_k^\top}{y_k^\top s_k}. \tag{5.38}$$

  Auch hier kann man wieder die positive Definitheit von H_{k+1}^{BFGS} garantieren, siehe Lemma 5.46.

---

[47] Die Frobeniusnorm einer Matrix $A \in \mathbb{R}^{n\times m}$ ist definiert als

$$\|A\|_F := \Big(\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}^2\Big)^{1/2} = \big(\text{trace}(A^\top A)\big)^{1/2}.$$

[48] siehe etwa die Übungsaufgabe (Ulbrich and Ulbrich, 2012, S.76)

[49] Das heißt, H_{k+1}^{PSB} und H_k unterscheiden sich um eine Matrix, die maximal Rang 2 hat. **Beachte:** $\text{rank}(A+B) \le \text{rank}(A) + \text{rank}(B)$.

- **Broyden-Klasse**: Die Broyden-Klasse ist gerade eine Linearkombination aus DFP und BFGS. Für $\lambda \in \mathbb{R}$ setzen wir also

$$H_{k+1}^{\lambda} = (1-\lambda)\, H_{k+1}^{\mathrm{BFGS}} + \lambda\, H_{k+1}^{\mathrm{DFP}}. \tag{5.39}$$

  Mit der Wahl $\lambda \in [0,1]$ erhält man die **konvexe Broyden-Klasse**.

Wir hatten oben die Bedingung (5.31), also $y_k^\top s_k > 0$, als notwendige Bedingung dafür identifiziert, dass H_{k+1} positiv definit ist. In der Tat ist diese Bedingung beim DFP- und beim BFGS-Update bereits hinreichend. Die positive Definitheit kann also einfach durch eine Wolfe-Liniensuche garantiert werden.

Lemma 5.46 (Positive Definitheit für DFP- und BFGS-Updates)**.** Falls H_k positiv definit ist und $y_k^\top s_k > 0$ gilt, dann sind auch H_{k+1}^{DFP} und H_{k+1}^{BFGS} positiv definit.

Beachte: Man kann dieses Resultat sogar für H_{k+1}^{λ} mit $\lambda \geq 0$ zeigen, siehe (Ulbrich and Ulbrich, 2012, Satz 13.4).

Beweis. Es gilt für $v \in \mathbb{R}^n$, $v \neq 0$:

$$v^\top H_{k+1}^{\mathrm{DFP}}\, v = (v^\top - \gamma_k\, (v^\top y_k)\, s_k^\top)\, H_k\, (v - \gamma_k\, s_k\, (v^\top y_k)) + \gamma_k\, (v^\top\, y_k)^2 \geq 0.$$

Gleichheit in dieser Ungleichung kann nur gelten, falls $v = t\, s_k$ (vorderer Teil) gilt, aber dann ist wegen $\gamma_k^{-1} = y_k^\top s_k > 0$ der hintere Teil positiv.

Weiter gilt

$$\begin{aligned} v^\top H_{k+1}^{\mathrm{BFGS}}\, v &= v^\top H_k\, v - \frac{(s_k^\top H_k\, v)^2}{s_k^\top H_k\, s_k} + \frac{(y_k^\top v)^2}{y_k^\top s_k} \\ &\geq v^\top H_k\, v - \frac{(s_k^\top H_k\, s_k)\, (v\, H_k\, v)}{s_k^\top H_k\, s_k} + \frac{(y_k^\top v)^2}{y_k^\top s_k} = \frac{(y_k^\top v)^2}{y_k^\top s_k} \geq 0. \end{aligned}$$

Bei der ersten Ungleichung haben wir Cauchy-Schwarz bzgl. des H_k-Skalarprodukts benutzt. Dort gilt also nur Gleichheit, falls $v = t\, s_k$. Dann ist aber die zweite Ungleichung strikt, da $y_k^\top s_k$ nach Voraussetzung > 0 ist.

Wir haben gesehen, dass typische Updates für die Hessematrix Rang 2 haben. Da wir mit jeder Matrix H_k ein LGS (5.4) zur Bestimmung der Suchrichtung d_k lösen müssen, kann man sich fragen, ob wir nicht auch direkt mit der inversen Matrix

$$B_k := H_k^{-1}$$

arbeiten können. Und in der Tat kann man entpsrechende Aufdatierungsformeln auch für B_k angeben, wobei man ausnutzt, dass $H_{k+1} - H_k$ einen kleinen Rang hat. Dies gelingt durch Anwendung des folgenden Resultats.

Lemma 5.47 (Sherman-Morrison-Woodbury-Formel)**.**

Die Matrizen $A \in \mathbb{R}^{n\times n}$ und $C \in \mathbb{R}^{k\times k}$ seien invertierbar und $U \in \mathbb{R}^{n\times k}$, $V \in \mathbb{R}^{k\times n}$. Dann ist $A + U\, C\, V$ genau dann invertierbar, wenn $C^{-1} + V\, A^{-1} U$ invertierbar ist. In diesem Fall gilt

$$(A + U\, C\, V)^{-1} = A^{-1} - A^{-1}\, U\, (C^{-1} + V\, A^{-1} U)^{-1}\, V\, A^{-1}. \tag{5.40}$$

Beweis. „⇐“: nachrechnen.

„⇒“: erhält man aus der ersten Richtung und Vertauschen der Rollen

$$A \rightsquigarrow C^{-1}, \quad C \rightsquigarrow A^{-1}, \quad V \rightsquigarrow U, \quad U \rightsquigarrow V.$$

Wir betrachten nun den für uns interessanten Fall $k = 2$. Dann besagt (5.40): Die Inverse eines Rang-2-Updates ist ein Rang-2-Update der Inversen. Kennen wir also $B_k := H_k^{-1}$, dann können wir direkt und effizient $B_{k+1} := H_{k+1}^{-1}$ berechnen.

Dies führt für DFP und BFGS auf die folgenden Formeln.

- **DFP invers**: Nutzt man (5.40) für (5.36), so erhält man

$$B_{k+1}^{\mathrm{DFP}} = B_k - \frac{B_k\, y_k\, y_k^\top B_k}{y_k^\top B_k\, y_k} + \frac{s_k\, s_k^\top}{s_k^\top y_k}. \tag{5.41}$$

- **BFGS invers**: Aus (5.38) erhält man

$$B_{k+1}^{\mathrm{BFGS}} = (I - \gamma_k\, s_k\, y_k^\top)\, B_k\, (I - \gamma_k\, y_k\, s_k^\top) + \gamma_k\, s_k\, s_k^\top, \tag{5.42}$$

  mit $\gamma_k = (y_k^\top s_k)^{-1}$.

Beachte: Die DFP- und BFGS-Updates sind dual zueinander, d. h., man erhält (5.36) $\rightsquigarrow$ (5.42) und (5.38) $\rightsquigarrow$ (5.41), wenn man $H_k \rightsquigarrow B_k$ und $y_k \rightsquigarrow s_k$ tauscht. Dies wird auch aus den obigen Herleitungen von (5.36) und (5.38) als Lösungen bestimmter Optimierungsaufgaben klar.

Zum Nachweis genügt es, die Beziehungen $B_{k+1}^{\mathrm{DFP}}\, H_{k+1}^{\mathrm{DFP}} = B_{k+1}^{\mathrm{BFGS}}\, H_{k+1}^{\mathrm{BFGS}} = I$ induktiv zu beweisen (eigene Übung).

Wenn man direkt mit den Inversen rechnet, so wird aus dem Quasi-Newton-System

$$H_k\, d_k = -\nabla f(x_k) \tag{5.4}$$

einfach ein Matrix-Vektor-Produkt

$$d_k = -B_k\, \nabla f(x_k).$$

Dadurch wird der Aufwand eines Schrittes des Optimierungsverfahrens erheblich reduziert.

In der Literatur findet sich vielfach die Aussage, dass sich in praktischen Vergleichen das BFGS-Update als leistungsfähiger als andere Quasi-Newton-Formeln herausgestellt hat; siehe etwa (Ulbrich and Ulbrich, 2012, p.69). Daher betrachten wir nun einen vollständigen Algorithmus für ein globalisiertes Quasi-Newton-Verfahren mit inversem BFGS-Update.

Algorithmus 5.48 (globalisiertes Quasi-Newton-Verfahren mit inversem BFGS-Update)**.**
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: anfängliche inverse BFGS-Matrix $B_0^{\mathrm{BFGS}} \in \mathbb{R}^{n\times n}$, spd
Eingabe: Funktionsauswertungen f und f' bzw. ∇f
Eingabe: Parameter $\gamma > 1$, $\underline{\gamma}, \overline{\gamma} \in (0, 1/2]$, $0 < \sigma < \tau < 1$ mit $\sigma \in (0, 1/2)$ für die Wolfe-Liniensuche
Ausgabe: näherungsweise stationärer Punkt der Aufgabe **(FO)**
1: Setze $k := 0$

2: Setze $f_0 := f(x_0)$
3: Setze $r_0 := f'(x_0)^\top = \nabla f(x_0)$
4: Setze $d_0^G := -M^{-1} r_0$
5: Setze $\delta_0 := -r_0^\top d_0^G$ $\{\delta_0 = \|\nabla_M f(x_0)\|_M^2 = \|d_0^G\|_M^2\}$
6: **while** Abbruchkriterium nicht erfüllt **do**
7: Bestimme die Quasi-Newton-Richtung d_k aus

$$d_k = -B_k^{\mathrm{BFGS}} \nabla f(x_k).$$

8: Bestimme eine Schrittweite $\alpha_k > 0$ mittels Wolfe-Liniensuche (Algorithmus 5.19), angewendet auf $\varphi(\alpha) := f(x_k + \alpha\, d_k)$ und mit Startschrittweite $\alpha_{k,0} = 1$ $\{\varphi(0) = f_k$ und $\varphi'(0) = -r_k^\top d_k^N$ sind bekannt$\}$
9: Setze $s_k := \alpha_k\, d_k$
10: Setze $x_{k+1} := x_k + s_k$
11: Setze $f_{k+1} := f(x_{k+1})$ {ist aus der Liniensuche bereits bekannt}
12: Setze $r_{k+1} := f'(x_{k+1})^\top = \nabla f(x_{k+1})$
13: Setze $d_{k+1}^G := -M^{-1} r_{k+1}$
14: Setze $\delta_{k+1} := -r_{k+1}^\top d_{k+1}^G$ $\{\delta_{k+1} = \|\nabla_M f(x_{k+1})\|_M^2 = \|d_{k+1}^G\|_M^2\}$
15: Setze $y_k := r_{k+1} - r_k$
16: Bestimme B_{k+1}^{BFGS} gemäß (5.42)
17: Setze $k := k + 1$
18: **end while**
19: **return** x_k

Bemerkung 5.49 (zum Quasi-Newton-Verfahren mit inversem BFGS-Update)**.**

(a) Es bietet sich an, als anfängliche inverse BFGS-Matrix $B_0^{\mathrm{BFGS}} = M^{-1}$ zu wählen.

(b) Das benutzerdefinierte Skalarprodukt kommt ansonsten nur bei der Berechnung der Größe δ_k vor, die wie in den anderen Verfahren zuvor zur Auswertung einer Abbruchbedingung genutzt werden kann. Man könnte erwarten, dass die Matrix M auch bei der Bestimmung des Quasi-Newton-Updates eine Rolle spielen sollte, da es etwas unnatürlich erscheint, dass in den Hilfsaufgaben (5.33), (5.35) und (5.37) stets die Frobeniusnorm zur Abstandsmessung zwischen Matrizen verwendet wird. Dies zu untersuchen könnte zum Gegenstand einer Masterarbeit werden.

Leider sind die Konvergenzaussagen zu Algorithmus 5.48 nicht so reichhaltig wie für andere Verfahren (siehe dazu etwa (Geiger and Kanzow, 1999, Kapitel 11)).

- Man erhält lokal q-superlineare Konvergenz (mit der Schrittweitenwahl $\alpha_k = 1$), wenn man hinreichend nah bei einem Minimierer x^* mit $\nabla^2 f(x^*) \succ 0$ startet und B_0 hinreichend nah an $\nabla^2 f(x^*)^{-1}$ liegt.
- Man erhält globale Konvergenz, falls die Konditionszahl der Matrizen B_k beschränkt bleibt, dies folgt aus Lemma 5.6 und dem globalen Konvergenzsatz 5.11. Leider kann diese Bedingung nicht a-priori garantiert werden. Ein möglicher Ausweg ist es, einen (verallgemeinerten) Winkeltest wie (5.23)

durchzuführen und B_k neu zu initialisieren (z. B. mit B_0), falls dieser fehlschlägt.

In der Praxis zeigt Algorithmus 5.48 oft eine q-superlineare Konvergenz. Dies ist bemerkenswert, da ja nur erste Ableitungen von f verwendet werden.

Ende 14. V

§ 5.5.3 Limited-Memory-BFGS-Verfahren

Ein Nachteil von Quasi-Newton-Verfahren wie Algorithmus 5.48 ist der hohe Speicherbedarf für die Matrix B_k^{BFGS}, falls die Problemdimension n groß ist. Typischerweise (aber nicht immer) ist bei großer Problemdimension die tatsächliche Hessematrix der Zielfunktion dünn besetzt (*sparse*), wohingegen B_k^{BFGS} und H_k^{BFGS} immer voll besetzt sind (sich jedoch nur um eine Rang-$2k$-Matrix von B_0^{BFGS} bzw. H_0^{BFGS} unterscheiden). Bei einer noch moderaten Problemgröße von $n = 10\,000$ benötigt man daher bereits

$$\frac{n\,(n+1)}{2}\,8\mathrm{B} = 381\,\mathrm{MiB}$$

an Speicher, bei $n = 100\,000$ sind es schon 37 GiB[50]!

Die Kombination zweier Ideen schafft hier Abhilfe:

(a) An Stelle der vollständigen Matrizen (wie B_k^{BFGS}) werden nur die Vektorpaare $(y_0, s_0), (y_1, s_1), \ldots$ gespeichert und damit das benötigte Matrix-Vektor-Produkt $B_k^{\mathrm{BFGS}}\,\nabla f(x_k)$ rekursiv berechnet.

(b) Da auch dieses Vorgehen mit der Zeit mehr und mehr Speicher benötigt (eben zwei zusätzliche vollbesetzte Vektoren der Länge n pro Iteration), verwendet man immer nur die letzten m Vektoren und verwirft die vorhergehenden. Dieses Vorgehen führt auf ein sogenanntes **Limited-Memory-Quasi-Newton-Verfahren** (**LM-Quasi-Newton-Verfahren**).

Diese Ideen sind bei allen gängigen Quasi-Newton-Verfahren sinnvoll anwendbar, da B_{k+1} aus B_k bzw. H_{k+1} aus H_k jeweils durch eine Niedrang-Korrektur hervorgeht. Im Folgenden konzentrieren wir uns aber auf das inverse BFGS-Update. Dazu betrachten nochmals die zugehörige Updateformel:

$$B_{k+1}^{\mathrm{BFGS}} = V_k^\top\, B_k^{\mathrm{BFGS}}\, V_k + \gamma_k\, s_k\, s_k^\top, \tag{5.42}$$

wobei $\gamma_k := (y_k^\top s_k)^{-1}$ und $V_k := I - \gamma_k\, y_k\, s_k^\top$ gesetzt wurden. Damit erhält man rekursiv

$$\begin{aligned} B_1^{\mathrm{BFGS}} &= V_0^\top\, B_0^{\mathrm{BFGS}}\, V_0 + \gamma_0\, s_0\, s_0^\top \\ B_2^{\mathrm{BFGS}} &= V_1^\top\, B_1^{\mathrm{BFGS}}\, V_1 + \gamma_1\, s_1\, s_1^\top \\ &= V_1^\top\, V_0^\top\, B_0^{\mathrm{BFGS}}\, V_0\, V_1 + \gamma_0\, V_1^\top s_0\, s_0^\top\, V_1 + \gamma_1\, s_1\, s_1^\top \end{aligned}$$

usw. Da wir nur Matrix-Vektor-Produkte wie

$$B_2^{\mathrm{BFGS}}\, g = V_1^\top\, V_0^\top\, B_0^{\mathrm{BFGS}}\, V_0\, V_1\, g + \gamma_0\, V_1^\top s_0\, s_0^\top\, V_1\, g + \gamma_1\, s_1\, s_1^\top g$$

berechnen müssen, kann dies wie folgt effizient rekursiv realisiert werden.

[50] Ein Mebibyte (MiB) sind 2^{20} Byte, ein Gibibyte (GiB) entsprechend 2^{30} Byte. Die Präfixe „Mebi“ und Gibi ersetzen die früher in diesem Kontext ebenfalls verwendeten „Mega“ und „Giga“, die jedoch eigentlich für 10^6 und 10^9 reserviert sind.

Algorithmus 5.50 (rekursive Auswertung von $B_k^{\text{BFGS}}\, g$).
Eingabe: Startmatrix B_0^{BFGS} (oder Matrix-Vektor-Produkte mit B_0^{BFGS})
Eingabe: Vektor $g \in \mathbb{R}^n$
Eingabe: Vektorpaare $\{(y_i, s_i)\}_{i=0,\dots,k-1}$ und Skalare $\gamma_i = (y_i^\top s_i)^{-1}$, $i = 0, \dots, k-1$
Ausgabe: $B_k^{\text{BFGS}}\, g$

Setze $r := g$
for $i := k-1, k-2, \dots, 0$ **do**
  Setze $\alpha_i := \gamma_i\, s_i^\top r$
  Setze $r := r - \alpha_i\, y_i$
end for $\{r = V_0\, V_1 \cdots V_{k-1}\, g\}$
Setze $d := B_0^{\text{BFGS}}\, r$
for $i := 0, 1, \dots, k-1$ **do**
  Setze $\beta := \gamma_i\, y_i^\top d$
  Setze $d := d + (\alpha_i - \beta)\, s_i$
end for
return d $\{d = B_k^{\text{BFGS}}\, g\}$

Bemerkung 5.51 (zur rekursiven Auswertung $B_k^{\text{BFGS}}\, g$).

(a) Die Matrix B_0^{BFGS} muss nicht explizit vorliegen, da lediglich Matrix-Vektor-Produkte mit ihr benötigt werden.

(b) Es besteht sogar die algorithmische Möglichkeit, die Matrix B_0^{BFGS} im laufenden Quasi-Newton-Verfahren zu verändern. Dies ist bei direkter Verwendung der Update-Formel (5.42) natürlich nicht möglich.

Wir kommen zurück zur zweiten Idee, nicht alle Vektorpaare (y_i, s_i) zu speichern, sondern lediglich die neuesten m Paare $\{(y_i, s_i)\}_{i=k-m,\dots,k-1}$. Diese Idee lässt sich in Algorithmus 5.50 sehr einfach einbauen:

Algorithmus 5.52 (Auswertung von $B_k^{\text{LM-BFGS}}\, g$).
Eingabe: Startmatrix B_0^{BFGS} (oder Matrix-Vektor-Produkte mit B_0^{BFGS})
Eingabe: Vektor $g \in \mathbb{R}^n$
Eingabe: $\{(y_i, s_i)\}_{i=k-m,\dots,k-1}$
Ausgabe: $B_k^{\text{LM-BFGS}}\, g$

Setze $r := g$
for $i := k-1, k-2, \dots, k-m$ **do**
  Setze $\alpha_i := \gamma_i\, s_i^\top r$
  Setze $r := r - \alpha_i\, y_i$
end for $\{r = V_{k-m}\, V_{k-m+1} \cdots V_{k-1}\, g\}$
Setze $d := B_0^{\text{BFGS}}\, r$
for $i := k-m, \dots, k-2, k-1$ **do**
  Setze $\beta := \gamma_i\, y_i^\top d$
  Setze $d := d + (\alpha_i - \beta)\, s_i$
end for
return d $\{d = B_k^{\text{LM-BFGS}}\, g\}$

Bemerkung 5.53 (zum inversen LM-BFGS-Verfahren)**.**

(a) Zum Starten wird die Anzahl der verwendeten Vektorpaare gemäß $m = \min\{k, m_{\max}\}$ bis auf $m_{\max}$ erhöht. Typisch ist etwa $3 \leq m_{\max} \leq 10$.

(b) Ein vollständiges Verfahren entsteht dadurch, dass im Quasi-Newton-Verfahren mit *unlimitiertem* inversen BFGS-Update (Algorithmus 5.48) einfach die Bestimmung der Quasi-Newton-Richtung durch

$$d_k = -B_k^{\text{LM-BFGS}} \nabla f(x_k)$$

ausgetauscht wird. Die Bestimmung von $B_{k+1}^{\text{LM-BFGS}}$ entfällt und wird ersetzt durch das Speichern des Vektorpaares (y_k, s_k).

(c) I. A. kann man bei einem LM-Quasi-Newton-Verfahren keine q-superlineare Konvergenz mehr erwarten.

§ 5.6 Nichtlineare CG-Verfahren

Literatur: (Geiger and Kanzow, 1999, Kapitel 13.2–13.5)

Im bisherigen Verlauf von § 5 haben wir mit dem Gradientenverfahren (§ 5.3) als erstes Beispiel eines Liniensuchverfahrens begonnen. Dieses kommt mit ersten Ableitungen der Zielfunktion aus, erreicht jedoch i. A. keine q-superlineare Konvergenz. Dies führte uns zum Newton-Verfahren (§ 5.4), das q-superlineare oder sogar q-quadratische Konvergenz erreicht, durch die Verwendung zweiter Ableitungen und die Notwendigkeit, lineare Gleichungssysteme mit $\nabla^2 f(x_k)$ zu lösen, jedoch wesentlich aufwändiger ist. Als Kompromiss betrachteten wir insbesondere Quasi-Newton-Verfahren (§ 5.5.2 und § 5.5.3), die mit ersten Ableitungen auskommen und dennoch q-superlineare Konvergenz erreichen können.

Eine alternative Idee zu Quasi-Newton-Verfahren, die ebenfalls mit ersten Ableitungen auskommt, ergibt sich daraus, das CG-Verfahren für beliebige (glatte) nichtlineare Zielfunktionen zu erweitern. Wir hatten ja bereits im Falle quadratischer Zielfunktionen in § 4 gesehen, dass das CG-Verfahren dem Gradientenverfahren deutlich überlegen ist.

Die wesentlichen Eckpunkte des CG-Verfahrens bei quadratischen Zielfunktionen waren:

(a) Jede neue Suchrichtung d_{k+1} wurde aus der aktuellen Suchrichtung d_k und der Richtung des steilsten Abstiegs so linearkombiniert, dass d_k und d_{k+1} im A-Skalarprodukt senkrecht standen (A-Konjugiertheit). Die A-Konjugiertheit zu allen früheren Suchrichtungen ergab sich automatisch.

(b) Entlang einer jeden Suchrichtung wurde eine exakte Liniensuche ausgeführt, was aufgrund der quadratischen Zielfunktion einfach möglich war.

Im Falle allgemeiner nichtlinearer Zielfunktionen ist Folgendes zu beachten:

- Eine exakte Liniensuche ist nicht mehr möglich. Stattdessen verwendet man meist eine (strenge) Wolfe-Liniensuche.
- Wir bezeichnen auch hier $\nabla f(x_k)$ als das Residuum r_k. Dieses hat jedoch nicht mehr die Form $r_k = A\,x_k - b$.

- Da es keine Matrix A mehr gibt, lässt man die Forderung der A-Orthogonalität der Suchrichtungen fallen. Man behält aber die Konstruktionsvorschrift

$$d_k := -M^{-1} r_k + \beta_k \, d_{k-1}, \quad \text{bzw.} \quad d_0 := -M^{-1} r_0 \tag{4.28}$$

bei und verwendet eine der verschiedenen Formeln aus (4.29') zur Bestimmung von β_k, die nun *nicht* mehr äquivalent sind.

Algorithmus 5.54 (nichtlineares CG-Verfahren)**.**
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: spd Matrix M (oder Matrix-Vektor-Produkte mit M^{-1})
Ausgabe: näherungsweise stationärer Punkt der Aufgabe **(FO)**

Setze $k := 0$
Setze $r_0 := \nabla f(x_0)$
Setze $d_0 := -M^{-1} \, r_0$
Setze $\delta_0 := -r_0^\top d_0$ $\quad \{\delta_0 = \|\nabla_M f(x_0)\|_M^2\}$
while Abbruchkriterium nicht erfüllt **do**
  Bestimme α_k über eine Liniensuche
  Setze $x_{k+1} := x_k + \alpha_k \, d_k$
  Setze $r_{k+1} := \nabla f(x_{k+1})$
  Setze $d_{k+1} := -M^{-1} r_{k+1}$
  Setze $\delta_{k+1} := -r_{k+1}^\top d_{k+1}$ $\quad \{\delta_{k+1} = \|\nabla_M f(x_{k+1})\|_M^2\}$
  Setze $y_k := r_{k+1} - r_k$
  Wähle β_{k+1}
  Setze $d_{k+1} := d_{k+1} + \beta_{k+1} \, d_k$
  Setze $k := k + 1$
end while
return x_k

Verschiedene Varianten des nichtlinearen CG-Verfahrens unterscheiden sich in der Wahl von β_{k+1}. Insgesamt hatten wir in den Ausdrücken in (4.29') für β_{k+1} u. a. die beiden Zähler

$$(r_{k+1} - r_k)^\top \, M^{-1} \, r_{k+1}, \quad r_{k+1}^\top \, M^{-1} \, r_{k+1}$$

und die drei Nenner

$$(r_{k+1} - r_k)^\top \, d_k, \quad -r_k^\top \, d_k, \quad r_k^\top \, M^{-1} \, r_k$$

gesehen.[51] Alle sechs Kombinationen (sowie Varianten) tauchen in der Literatur auf und liefern sinnvolle Verfahren. Sie sind in der Tabelle 5.1 zusammengefasst, wobei zur Abkürzung $y_k = \nabla f(x_{k+1}) - \nabla f(x_k) = r_{k+1} - r_k$ verwendet wurde.

Man muss beachten, dass die Liniensuche auf die Wahl von β_k abgestimmt sein muss, damit man eine Abstiegsrichtung erhält. Typischerweise wird geraten, dass die Liniensuche genau genug gemacht wird, etwa mit den strengen Wolfe-Bedingungen und etwa $\tau = 10^{-1}$ oder $\tau = 10^{-2}$.

Benutzt man (theoretisch) eine exakte Liniensuche, sodass α_k ein stationärer Punkt der Liniensuchfunktion $\varphi(\alpha) = f(x_k + \alpha \, d_k)$ ist, dann gilt $r_{k+1}^\top d_k = f'(x_{k+1}) \, d_k = \varphi'(\alpha_k) = 0$ und (aus dem vorherigen Schritt) $r_k^\top d_{k-1} = 0$ und somit

$$(r_{k+1} - r_k)^\top \, d_k = -r_k^\top \, d_k = r_k^\top \, (M^{-1} \, r_k - \beta_k \, d_{k-1}) = r_k^\top \, M^{-1} \, r_k.$$

[51] Der Ausdruck $-r_k^\top \, d_k$ stand explizit nicht dort, jedoch folgt aus (4.27) sofort $(r_{k+1} - r_k)^\top \, d_k = -r_k^\top \, d_k$.

In diesem Fall fallen also die drei verschiedenen Ausdrücke für den Nenner von β_{k+1} zusammen.

Die Konvergenzbeweise für nichtlineare CG-Verfahren sind deutlich technischer als bei den anderen Verfahren, und wir gehen hier nicht darauf ein. Die Verfahren mit Zähler $r_k^\top M^{-1} r_k$ haben eine bessere Konvergenztheorie, aber die Verfahren mit Zähler $(r_k - r_{k-1})^\top M^{-1} r_k$ schlagen sich in der Praxis besser.

Als Abbruchbedingung bietet sich wieder ein relatives und/oder absolutes Kriterium für $r_k = \nabla f(x_k)$ an, vgl. (4.22).

Ende 15. V

	Wahl von β_{k+1}	Bemerkung
Hestenes–Stiefel (1952)	$\beta_{k+1}^{\mathrm{HS}} = \dfrac{y_k^\top M^{-1} r_{k+1}}{y_k^\top d_k}$	
Fletcher–Reeves (1964)	$\beta_{k+1}^{\mathrm{FR}} = \dfrac{\|r_{k+1}\|_{M^{-1}}^2}{\|r_k\|_{M^{-1}}^2}$	starke Wolfe-Bed. (5.12), (5.17) mit $0 < \sigma < \tau < 1/2$
Polak–Ribière (1969)	$\beta_{k+1}^{\mathrm{PR}} = \dfrac{y_k^\top M^{-1} r_{k+1}}{\|r_k\|_{M^{-1}}^2}$	Abstieg nicht garantiert, daher oft $\beta_{k+1}^{\mathrm{PR}+} := \max\{0, \beta_{k+1}^{\mathrm{PR}}\}$ ($\beta_{k+1} = 0 \mathrel{\widehat{=}}$ Gradientenschritt)
Powell (1985)	$\beta_{k+1}^{\mathrm{PR}+} = \max\{0, \beta_{k+1}^{\mathrm{PR}}\}$	Verfeinerung der starken Wolfe-Bed., siehe (Gilbert and Nocedal, 1992, (4.1) und Abschnitt 6)
Fletcher (1987)	$\beta_{k+1}^{\mathrm{F}} = \dfrac{\|r_{k+1}\|_{M^{-1}}^2}{-r_k^\top d_k}$	
Liu–Storey (1991)	$\beta_{k+1}^{\mathrm{LS}} = \dfrac{y_k^\top M^{-1} r_{k+1}}{-r_k^\top d_k}$	
Gilbert–Nocedal (1992)	$\beta_{k+1}^{\mathrm{GN}} = \begin{cases} -\beta_{k+1}^{\mathrm{FR}}, & \text{falls } \beta_{k+1}^{\mathrm{PR}} < -\beta_{k+1}^{\mathrm{FR}} \\ \beta_{k+1}^{\mathrm{PR}}, & \text{falls } \lvert\beta_{k+1}^{\mathrm{PR}}\rvert \le \beta_{k+1}^{\mathrm{FR}} \\ \beta_{k+1}^{\mathrm{FR}}, & \text{falls } \beta_{k+1}^{\mathrm{PR}} > \beta_{k+1}^{\mathrm{FR}} \end{cases}$	starke Wolfe-Bed. (5.12), (5.17) mit $0 < \sigma < \tau < 1/2$
Dai–Yuan (1999)	$\beta_{k+1}^{\mathrm{DY}} = \dfrac{\|r_{k+1}\|_{M^{-1}}^2}{y_k^\top d_k}$	
Hager–Zhang (2005)	$\beta_{k+1}^{\mathrm{HZ}} = \left(M^{-1} y_k - 2\, d_k \dfrac{\|y_k\|_{M^{-1}}^2}{y_k^\top d_k} \right)^\top \dfrac{r_{k+1}}{y_k^\top d_k}$	Wolfe-Bed. (5.12), (5.16) mit $0 < \sigma < \tau < 1$

Tabelle 5.1. Einige gängige nichtlineare CG-Verfahren

§ 6 Trust-Region-Verfahren

Die Liniensuchverfahren aus § 5 bestimmen zunächst eine Suchrichtung d_k durch (ggf. inexakte) Minimierung eines quadratischen Modells

$$q_k(d) = f(x_k) + f'(x_k)\,d + \frac{1}{2}\,d^\top H_k\,d \tag{5.2}$$

bzw. (ggf. inexakte) Lösung des LGS

$$H_k\,d_k = -\nabla f(x_k). \tag{5.4}$$

Anschließend wird eine geeignete Schrittlänge $\alpha_k > 0$ gewählt und

$$x_{k+1} := x_k + \underbrace{\alpha_k\,d_k}_{s_k}$$

gesetzt.

Trust-Region-Verfahren bestimmen dagegen Schrittrichtung und -länge *gleichzeitig*. Der Schritt s_k wird gewonnen als eine (i. d. R. inexakte) Lösung des **Trust-Region-Teilproblems**:

$$\begin{aligned} \text{Minimiere} \quad & q_k(s) = f(x_k) + f'(x_k)\,s + \frac{1}{2}\,s^\top H_k\,s, \quad s \in \mathbb{R}^n \\ \text{unter} \quad & \|s\|_M \le \Delta_k. \end{aligned} \tag{6.1}$$

Die Größe $\Delta_k > 0$, der sogenannte **Trust-Region-Radius**, beschreibt dabei den **Vertrauensbereich** (*trust region*)

$$\{s \in \mathbb{R}^n : \|s\|_M \le \Delta_k\},$$

den man in das quadratische Modell hat.

Beachte: Da im Punkt $s = 0$ die nullte und erste Ableitung[52] von q_k mit der von $s \mapsto f(x_k + s)$ übereinstimmt, wird für kleine Werte von $\|s\|_M$ das Modell $q_k(s)$ gut mit $f(x_k + s)$ übereinstimmen.

Im Modell ist H_k wieder eine symmetrische Matrix, die aber (im Unterschied zu den meisten Verfahren in § 5) nicht notwendig positiv definit sein muss, da (6.1) als Minimierungsaufgabe einer stetigen Zielfunktion über einer kompakten Menge in jedem Fall eine globale Lösung besitzt.

Analog wie bei Liniensuchverfahren muss zur globalen Konvergenz eines Trust-Region-Verfahrens die Qualität des Schrittes (Stichwort: hinreichender Abstieg) überwacht und gesteuert werden. Ist s_k eine (inexakte) Lösung von (6.1), dann bewertet man den Schritt s_k, indem man die tatsächliche Reduktion der Zielfunktion (*actual reduction*)

$$\mathrm{ared}_k(s_k) := f(x_k) - f(x_k + s_k)$$

mit der durch das verwendete Modell gemachten Vorhersage (*predicted reduction*)

$$\mathrm{pred}_k(s_k) := q_k(0) - q_k(s_k) = f(x_k) - q_k(s_k) = -f'(x_k)\,s_k - \frac{1}{2}\,s_k^\top H_k\,s_k$$

vergleicht. Dazu berechnen wir das Verhältnis

$$\rho_k(s_k) := \frac{\mathrm{ared}_k(s_k)}{\mathrm{pred}_k(s_k)}, \tag{6.2}$$

[52] und im Fall $H_k = \nabla^2 f(x_k)$ auch die zweite

das auch **Fortschrittsquotient** genannt wird. Mit Hilfe von zwei algorithmischen Parametern $0 < \eta_1 < \eta_2 < 1$ können wir nun den Schrittvorschlag s_k bewerten:

(a) Gilt $\rho_k(s_k) \leq \eta_1$, dann ist der Schritt unbefriedigend, was darauf zurückzuführen ist, dass das Modell q_k auf der aktuellen *trust region* nicht gut mit der ursprünglichen Funktion f übereinstimmt. Wir verwerfen den Schritt, indem wir $x_{k+1} := x_k$ setzen, und wählen einen neuen Trust-Region-Radius $\Delta_{k+1} < \Delta_k$. (Das Modell q_k wird typischerweise nicht verändert, also als q_{k+1} wiederverwendet.[53])

(b) Gilt hingegen $\rho_k(s_k) > \eta_1$, dann ist der Schritt hinreichend gut, und wir setzen $x_{k+1} := x_k + s_k$. Der Vertrauensradius Δ_{k+1} für den folgenden Schritt wird auch in Abhängigkeit von $\rho_k(s_k)$ gewählt:

(i) War die Übereinstimmung zwischen vorhergesagter und tatsächlicher Reduktion des Funktionswertes sogar ausgesprochen gut, d. h., gilt sogar $\rho_k(s_k) > \eta_2$, dann erlauben wir für den nächsten Schritt eine Vergrößerung der *trust region*. Dies ist aber nur dann sinnvoll, wenn der Schritt s_k die *trust region* auch ausgeschöpft hat, wenn also $\|s_k\|_M = \Delta_k$ gilt.

(ii) Andernfalls setzen wir $\Delta_{k+1} := \Delta_k$.

Mit diesen Ideen erhalten wir folgendes allgemeine Trust-Region-Verfahren (vgl. das allgemeine Abstiegsverfahren mit Liniensuche aus Algorithmus 5.2).

Algorithmus 6.1 (allgemeines Trust-Region-Verfahren)**.**
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: Modell-Hessematrix $H_0 \in \mathbb{R}^{n\times n}$ spd
Eingabe: anfänglicher Trust-Region-Radius $\Delta_0 > 0$
Eingabe: Funktionsauswertungen f und f' bzw. ∇f
Eingabe: Methode zur Bestimmung der Modell-Hessematrizen H_k
Eingabe: Parameter $0 < \eta_1 < \eta_2 < 1$ und $0 < \gamma_1 < 1 < \gamma_2$
Ausgabe: näherungsweise stationärer Punkt der Aufgabe (**FO**)

```
Setze k := 0
while Abbruchkriterium nicht erfüllt do
  Bestimme eine inexakte Lösung s_k des TR-Teilproblems (6.1)
  Berechne ρ_k gemäß (6.2)
  if ρ_k(s_k) > η_1 then {                                  erfolgreicher Schritt}
    Setze x_{k+1} := x_k + s_k
    if ρ_k(s_k) > η_2 und ‖s_k‖_M = Δ_k then {sehr guter Schritt und TR zu klein}
      Setze Δ_{k+1} := γ_2 Δ_k
    else {                                  mäßig guter Schritt oder TR groß genug}
      Setze Δ_{k+1} := Δ_k
    end if
  else {                                                    schlechter Schritt}
    Setze x_{k+1} := x_k
    Setze Δ_{k+1} := γ_1 Δ_k oder sogar Δ_{k+1} := γ_1 ‖s_k‖_M
  end if
  Setze k := k + 1
```

[53] Es bestünde ja die Möglichkeit, die Hessematrix des Modells zu „verbessern“.

17: **end while**
18: **return** x_k

Bemerkung 6.2 (zum allgemeinen Trust-Region-Verfahren)**.** Die Auswertung von $\rho_k(s_k)$ erfordert pro Iteration

- die Auswertung eines Funktionswertes $f(x_k + s_k)$ und
- die Auswertung von $q_k(s_k) = f(x_k) + f'(x_k)\, s_k + \frac{1}{2}\, s_k^\top H_k\, s_k$.

Falls der Schritt erfolgreich war, so dient $f(x_k + s_k)$ auch gleich als $f(x_{k+1})$, sodass wirklich nur eine f-Auswertung pro Iteration benötigt wird. Die Auswertung von $q_k(s_k)$ erfolgt i. d. R. gleichzeitig mit der Bestimmung der inexakten Lösung s_k des Trust-Region-Teilproblems (6.1).

Im weiteren Verlauf dieses Abschnittes betrachten wir die folgenden Punkte:

- Welche Voraussetzungen müssen wir an Algorithmus 6.1 stellen, insbesondere für die Wahl der Matrizen H_k und die Genauigkeit der Lösung des Trust-Region-Teilprobleme, um globale Konvergenz zu erhalten?
- Wie können wir schnelle lokale Konvergenz erhalten?
- Wie kann das Teilproblem (6.1) algorithmisch sinnvoll gelöst werden?

§ 6.1 Globale Konvergenz

Voraussetzung 6.3. Im § 6.1 sei $f : \mathbb{R}^n \to \mathbb{R}$ überall stetig differenzierbar (eine C^1-Funktion).

Wie bei den Liniensuchverfahren müssen wir eine Forderung an Algorithmus 6.1 stellen, um die globale Konvergenz zu erhalten. Bei Liniensuchverfahren bildete die Richtung des steilsten Abstiegs $d_k^G = -\nabla_M f(x_k)$, also die Richtung des (negativen) M-Gradienten, eine gute Referenz.

Da wir bei Trust-Region-Verfahren Suchrichtung und Schrittweite gemeinsam betrachten müssen, führt dieselbe Idee hier darauf, als Referenz den sogenannten **Cauchy-Punkt** oder **Cauchy-Schritt** s^C zu verwenden. Dieser ist definiert als die eindeutige Lösung des Trust-Region-Teilproblems (6.1), eingeschränkt auf die Gradientenrichtung:[54]

$$\begin{aligned} \text{Minimiere} \quad & q(s), \quad s \in \mathbb{R}^n,\ \tau \in \mathbb{R} \\ \text{sodass} \quad & \|s\|_M \le \Delta, \\ \text{und} \quad & s = -\tau\, \nabla_M f(x). \end{aligned} \tag{6.3}$$

Beachte: Die Aufgabe (6.3) kann natürlich auf die skalare Optimierungsvariable $\tau \in \mathbb{R}$ reduziert werden. Setzen wir zur Abkürzung

$$g := \nabla_M f(x),$$

[54] Der Einfachheit halber lassen wir vorübergehend den Iterationsindex weg.

so lautet die Aufgabe für den Cauchy-Koeffizienten τ^C:

$$\begin{aligned} &\text{Minimiere} \quad q(-\tau\, g) = f(x) - \tau\, \|g\|_M^2 + \frac{\tau^2}{2}\, g^\top H\, g, \quad \tau \in \mathbb{R} \\ &\text{sodass} \quad |\tau| \le \frac{\Delta}{\|g\|_M}. \end{aligned} \tag{6.4}$$

Es verbleibt also die Minimierung eines univariaten quadratischen Polynoms über einem zulässigen Intervall (symmetrisch zur Null). Die Lösung kann sehr einfach bestimmt werden:

Lemma 6.4 (Bestimmung des Cauchy-Punktes)**.** Es sei $g \neq 0$ und $\Delta \ge 0$. Dann ist die eindeutige Lösung von (6.3) gegeben durch $s^C = -\tau^C g$ mit

$$\tau^C = \begin{cases} \min\left\{ \dfrac{\|g\|_M^2}{g^\top H\, g}, \dfrac{\Delta}{\|g\|_M} \right\}, & \text{falls } g^\top H\, g > 0, \\ \dfrac{\Delta}{\|g\|_M} & \text{sonst.} \end{cases} \tag{6.5}$$

Somit gilt für den durch das Modell vorhergesagten Abstieg im Cauchy-Punkt s^C:

$$\operatorname{pred}(s^C) = f(x) - q(s^C) \ge \frac{1}{2}\, \|g\|_M \, \min\left\{ \Delta, \frac{\|g\|_M^3}{\max\{0, g^\top H\, g\}} \right\}. \tag{6.6}$$

Hierbei verstehen wir $\|g\|_M^3/0$ als $+\infty$.

Beweis. Wir bezeichnen die Zielfunktion (quadratisches Polynom) von (6.4) mit

$$\varphi(\tau) := f(x) - \tau\, \|g\|_M^2 + \frac{\tau^2}{2}\, g^\top H\, g.$$

Wir unterscheiden zwei Fälle:

Fall 1: $g^\top H\, g > 0$ (φ ist gleichmäßig konvex)
Ableiten und Nullsetzen von φ liefert

$$\tau^* = \frac{\|g\|_M^2}{g^\top H\, g} > 0.$$

Wenn dieser Wert zulässig ist, dann ist er die eindeutige Lösung. Andernfalls Wegen ist wegen $\varphi'(0) = -\tau\, \|g\|_M^2 < 0$ und der Monotonie von φ' gerade der Maximalwert $\tau = \frac{\Delta}{\|g\|_M}$ die eindeutige Lösung der Aufgabe. Zusammen also:

$$\tau^C = \min\left\{ \frac{\|g\|_M^2}{g^\top H\, g}, \frac{\Delta}{\|g\|_M} \right\}.$$

Um den Wert für $\operatorname{pred}(s^C)$ zu ermitteln, unterscheiden wir, von welchem Term das Minimum angenommen wird. Im Allgemeinen gilt für $s = -\tau\, g$:

$$\operatorname{pred}(s) = \operatorname{pred}(-\tau\, g) = f(x) - q(-\tau\, g) = \tau\, \|g\|_M^2 - \frac{\tau^2}{2}\, g^\top H\, g.$$

Fall 1.1: $\tau^C = \dfrac{\|g\|_M^2}{g^\top H\, g} \le \dfrac{\Delta}{\|g\|_M}$

$$\operatorname{pred}(s^C) = \frac{\|g\|_M^4}{g^\top H\, g} - \frac{1}{2}\, \frac{\|g\|_M^4}{g^\top H\, g} = \frac{1}{2}\, \frac{\|g\|_M^4}{g^\top H\, g} = \frac{1}{2}\, \|g\|_M\, \frac{\|g\|_M^3}{\max\{0, g^\top H\, g\}},$$

also gilt in diesem Fall (6.6).

Fall 1.2: $\tau^C = \dfrac{\Delta}{\|g\|_M} \le \dfrac{\|g\|_M^2}{g^\top H\, g}$

$$\begin{aligned}\operatorname{pred}(s^C) &= \tau^C \left[\|g\|_M^2 - \frac{1}{2}\left[\frac{\Delta}{\|g\|_M}\right] g^\top H\, g\right] \\ &\ge \tau^C \left[\|g\|_M^2 - \frac{1}{2}\left[\frac{\|g\|_M^2}{g^\top H\, g}\right] g^\top H\, g\right] = \frac{1}{2}\,\tau^C \|g\|_M^2 = \frac{1}{2}\,\Delta\,\|g\|_M,\end{aligned}$$

woraus wiederum (6.6) folgt.

Fall 2: $g^\top H\, g \le 0$
Da die zu minimierende Funktion φ konkav ist, liegt die Lösung τ^C auf dem Rand des zulässigen Intervalles, und wegen $\varphi'(0) = -\tau\,\|g\|_M^2 < 0$ gilt

$$\tau^C = + \frac{\Delta}{\|g\|_M}.$$

Für den Wert von $\operatorname{pred}(s^C)$ gilt in diesem Fall

$$\operatorname{pred}(s^C) = \Delta\,\|g\|_M + \frac{1}{2}\left[\frac{\Delta}{\|g\|_M}\right]^2 |g^\top H\, g| \ge \Delta\,\|g\|_M,$$

was auch in diesem Fall (6.6) zeigt.

Folgerung 6.5. Aufgrund der Abschätzung $\frac{g^\top H\, g}{\|g\|_M^2} \le \lambda_{\max}(H; M)$ folgt unter den Voraussetzungen von Lemma 6.4 aus (6.6) weiter auch

$$\operatorname{pred}(s^C) \ge \frac{1}{2}\,\|g\|_M \,\min\left\{\Delta, \frac{\|g\|_M}{\max\{0, \lambda_{\max}(H; M)\}}\right\}. \tag{6.7}$$

Um eine Bedingung für die globale Konvergenz des allgemeinen Trust-Region-Verfahrens (Algorithmus 6.1) zu formulieren, vergleichen wir den Fortschritt $\operatorname{pred}_k(s_k)$ (also in den Funktionswerten des Modells), den der approximative Minimierer s_k des Teilproblems (6.1) erreicht, mit dem Fortschritt $\operatorname{pred}_k(s_k^C)$, den der Cauchy-Punkt s_k^C aufgrund der Abschätzung (6.7) mindestens aufweist. Man fordert, dass s_k mindestens einen festen Bruchteil des Fortschrittes liefert, den der Cauchy-Punkt garantiert.

Voraussetzung 6.6.

Die Folge $\{x_k\}$ sei mit Algorithmus 6.1 erzeugt[55], wobei die Schrittvorschläge $\{s_k\}$ gemacht werden und $\{H_k\}$ die Modell-Hessematrizen der quadratischen Modelle sind sowie die Abkürzung $g_k := \nabla_M f(x_k)$ gilt.

(a) Die Schrittvorschläge s_k erfüllen

$$\|s_k\|_M \le \Delta_k \tag{6.8a}$$

und $$\operatorname{pred}_k(s_k) \ge \alpha\,\|g_k\|_M \,\min\left\{\Delta_k, \frac{\|g_k\|_M}{\max\{0, \lambda_{\max}(H_k; M)\}}\right\} \tag{6.8b}$$

mit einer Konstanten $\alpha \in (0, 1/2]$.

(b) Die Approximationen H_k der Hessematrix sind beschränkt, es gilt also

$$-C \le \lambda_{\min}(H_k; M) \quad \text{und} \quad \lambda_{\max}(H_k; M) \le C \tag{6.9}$$

für eine Konstante $C > 0$.

Die (6.8b) heißt auch eine **Fraction-of-Cauchy-decrease**-Bedingung.

Beachte: Unter (6.8b) ist insbesondere $\operatorname{pred}_k(s_k) > 0$, d. h., dass ein akzeptierter Schritt wegen

$$f(x_k) - f(x_k + s_k) = \operatorname{ared}_k(s_k) = \rho_k(s_k) \operatorname{pred}_k(s_k) > \eta_1 \operatorname{pred}_k(s_k) > 0$$

monoton fallende Funktionswerte generiert. Das Trust-Region-Verfahren wird also zu einem **Abstiegsverfahren**.

Wir zeigen nun als erstes Teilergebnis, dass die Schrittvorschläge unter Voraussetzung 6.6 „beliebig gut“ werden (Fortschrittsquotient geht gegen 1), wenn nur der Trust-Region-Radius Δ klein genug ist.

Lemma 6.7. Es sei $x \in \mathbb{R}^n$ ein Punkt mit $f'(x) \neq 0$ und $\{x_k\}$, $\{s_k\}$ und $\{H_k\}$ durch Algorithmus 6.1 erzeugte Folgen, die die Voraussetzung 6.6 erfüllen. Weiter sei $\eta \in (0, 1)$ beliebig. Dann existieren Konstanten[56] $\Delta > 0$ und $\delta > 0$, sodass

$$\rho_k(s_k) > \eta$$

gilt für alle Indizes k, für die $\|x - x_k\|_M \leq \delta$ und $\Delta_k \leq \Delta$ bleiben.

Beweis. Es gilt

$$\rho_k(s_k) = \frac{\operatorname{ared}_k(s_k)}{\operatorname{pred}_k(s_k)} = 1 - \frac{\operatorname{pred}_k(s_k) - \operatorname{ared}_k(s_k)}{\operatorname{pred}_k(s_k)} \overset{!}{>} \eta. \qquad (*)$$

Wir müssen also den Zähler des Bruches nach oben und den Nenner nach unten abschätzen.

Zum Nenner: Wegen der Stetigkeit von f' existiert ein $\delta > 0$, sodass

$$\|g_k\|_M = \|\nabla_M f(x_k)\|_M \geq \frac{\|\nabla_M f(x)\|_M}{2}$$

für $\|x - x_k\|_M \leq \delta$ gilt. Wir setzen zunächst $\Delta := \|\nabla_M f(x)\|_M / (2\,C)$ und betrachten nur solche Folgenglieder, für die neben $\|x - x_k\|_M \leq \delta$ auch $\Delta_k \leq \Delta$ gilt. Für den Nenner gilt dann wegen

$$\Delta_k \leq \Delta \leq \frac{\|\nabla_M f(x)\|_M}{2\,C} \leq \frac{\|g_k\|_M}{C} \leq \begin{cases} \frac{\|g_k\|_M}{\lambda_{\max}(H_k; M)}, & \text{falls } \lambda_{\max}(H_k; M) > 0 \\ \infty & \text{falls } \lambda_{\max}(H_k; M) \leq 0 \end{cases}$$

auch

$$\operatorname{pred}_k(s_k) \geq \alpha \|g_k\|_M \min\left\{\Delta_k, \frac{\|g_k\|_M}{\max\{0, \lambda_{\max}(H_k; M)\}}\right\} = \alpha \|g_k\|_M \Delta_k.$$

[55] ohne Berücksichtigung einer Abbruchbedingung

[56] abhängig von x und η

Zum Zähler: Mit einer Taylorentwicklung gilt für den Zähler mit einem $\xi_k \in (0,1)$

$$\begin{aligned}\mathrm{pred}_k(s_k) - \mathrm{ared}_k(s_k) &= f(x_k + s_k) - q_k(s_k)\\ &= f'(x_k + \xi_k\, s_k)\, s_k - f'(x_k)\, s_k - \frac{1}{2}\, s_k^\top H_k\, s_k\\ &\le \|f'(x_k + \xi_k\, s_k) - f'(x_k)\|_{M^{-1}}\, \|s_k\|_M - \frac{1}{2}\, \lambda_{\min}(H_k; M)\, \|s_k\|_M^2\\ &\le \|f'(x_k + \xi_k\, s_k) - f'(x_k)\|_{M^{-1}}\, \Delta_k + \frac{1}{2}\, C\, \Delta_k^2.\end{aligned}$$

Durch die Stetigkeit von f' sowie $\|x_k + \xi_k\, s_k - x\|_M \le \delta + \Delta$ und $\|x_k - x\|_M \le \delta$ können wir Δ und δ falls erforderlich so verkleinern, dass

$$\|f'(x_k + \xi_k\, s_k) - f'(x_k)\|_{M^{-1}} + \frac{1}{2}\, C\, \Delta_k < (1-\eta)\, \alpha\, \frac{\|\nabla_M f(x)\|_M}{2}$$

wird. Nun gilt

$$\begin{aligned}\mathrm{pred}_k(s_k) - \mathrm{ared}_k(s_k) &\le \|f'(x_k + \xi_k\, s_k) - f'(x_k)\|_{M^{-1}}\, \Delta_k + \frac{1}{2}\, C\, \Delta_k^2\\ &< (1-\eta)\, \alpha\, \frac{\|\nabla_M f(x)\|_M}{2}\, \Delta_k \le (1-\eta)\, \alpha\, \|g_k\|_M\, \Delta_k.\end{aligned}$$

Somit liefert $(*)$

$$\rho_k > 1 - \frac{(1-\eta)\, \alpha\, \|g_k\|_M\, \Delta_k}{\alpha\, \|g_k\|_M\, \Delta_k} = 1 - (1-\eta) = \eta.$$

Aus dieser Aussage ergibt sich sofort, dass unter Voraussetzung 6.6 der Algorithmus 6.1 (sofern er nicht mit $f'(x_k) = 0$ abbricht) unendlich oft einen Schrittvorschlag akzeptiert:

Folgerung 6.8. Die Iterierten x_k in Algorithmus 6.1 sollen $f'(x_k) \neq 0$ erfüllen, und es gelte Voraussetzung 6.6. Dann gibt es unendlich viele k mit $\rho_k(s_k) > \eta_1$.

Beweis. Dies folgt sofort aus Lemma 6.7, denn: Im Falle $\rho_k(s_k) \le \eta_1$ für alle $k \ge k_0$ wird die Folge $\{x_k\}$ ab dem Index k_0 konstant (alle Schrittvorschläge werden verworfen). Da aber ab diesem Index $\Delta_{k+1} = \gamma_1\, \Delta_k$ mit $\gamma_1 < 1$ gewählt wird, konvergiert $\Delta_k \to 0$. Aus Lemma 6.7 mit $x = x_{k_0}$ folgt, dass für hinreichend kleine Trust-Region-Radien (also hinreichend große k) aber $\rho_k(s_k) \le \eta_1$ sein wird; Widerspruch.

Die Folgerung 6.8 bedeutet, dass auf einen erfolgreichen Schritt immer nur endlich viele nicht erfolgreiche Schritte folgen können.

Ende 16. V

Ein weiterer Bestandteil für die Konvergenzbeweise ist das folgende Lemma.

Lemma 6.9. Die Iterierten $\{x_k\}$ in Algorithmus 6.1 sollen $f'(x_k) \neq 0$ erfüllen, es gelte Voraussetzung 6.6, und die Funktion f sei nach unten beschränkt[57]. Ist $K \subset \mathbb{N}$ eine unendliche Menge erfolgreicher Schritte ($\rho_k(s_k) > \eta_1$ für alle $k \in K$) mit der Eigenschaft $\|g_k\|_M \ge \varepsilon$, dann gilt

$$\sum_{k \in K} \Delta_k < \infty.$$

Beweis. Für $k \in K$ gilt nach Voraussetzung $\rho_k(s_k) > \eta_1$ und somit

$$\begin{aligned} f(x_k) - f(x_{k+1}) &= \operatorname{ared}_k(s_k) \\ &> \eta_1 \operatorname{pred}_k(s_k) \\ &\geq \eta_1 \alpha \|g_k\|_M \min\{\Delta_k, \|g_k\|_M / C\} \qquad \text{wegen (6.8b), (6.9)} \\ &\geq \eta_1 \alpha \varepsilon \min\{\Delta_k, \varepsilon/C\}. \end{aligned}$$

Da die Funktionswerte $\{f(x_k)\}$ monoton fallen und f nach unten beschränkt ist, muss

$$\sum_{k \in K} \eta_1 \alpha \varepsilon \min\{\Delta_k, \varepsilon/C\} < \infty$$

gelten. Genauer: Es sei $k_{\min}$ der kleinste Index in K. Dann gilt

$$\begin{aligned} \sum_{k \in K} \eta_1 \alpha \varepsilon \min\{\Delta_k, \varepsilon/C\} &\leq \sum_{k \in K} f(x_k) - f(x_{k+1}) \\ &\leq \sum_{k \geq k_{\min}} f(x_k) - f(x_{k+1}) \\ &\leq f(x_{k_{\min}}) - \inf_{x \in \mathbb{R}^n} f(x) \\ &< \infty. \end{aligned}$$

Daraus folgt die Behauptung.

Mit diesen Vorarbeiten können wir eine globale Konvergenzaussage von Algorithmus 6.1 beweisen.

Satz 6.10 (Globale Konvergenz des allgemeinen Trust-Region-Verfahrens)**.**

Es gelte Voraussetzung 6.6, und die Funktion f sei nach unten beschränkt. Dann terminiert Algorithmus 6.1 mit einem stationären Punkt, oder es gilt

$$\liminf_{k \to \infty} \|g_k\|_M = 0. \tag{6.10}$$

Ist f' sogar gleichmäßig stetig auf $\{x_k\}$, dann gilt sogar

$$\lim_{k \to \infty} \|g_k\|_M = 0. \tag{6.11}$$

Beachte: Der Satz sagt nichts über die Konvergenz der Folge $\{x_k\}$ oder einer Teilfolge aus. Aus (6.11) folgt aber, dass alle Häufungspunkte von $\{x_k\}$ stationäre Punkte sind.

Beweis. Es gelte $f'(x_k) \neq 0$ für alle Iterierten.

Angenommen, (6.10) gilt nicht. Dann gibt es ein $\varepsilon > 0$, sodass $\|g_k\|_M \geq \varepsilon$ für alle $k \in \mathbb{N}$ gilt. Wir bezeichnen mit K die Menge der erfolgreichen Schritte. Diese ist nach Folgerung 6.8 unendlich. Aus Lemma 6.9 erhalten wir

$$\sum_{k \in K} \Delta_k < \infty. \tag{$*$}$$

[57] Es reicht auch, dass die Folge $\{f(x_k)\}$ nach unten beschränkt ist.

Für beliebige Indizes $k > \ell$ gilt

$$\|x_k - x_\ell\|_M \leq \sum_{\substack{j \in K \\ \ell \leq j < k}} \|s_j\|_M \leq \sum_{\substack{j \in K \\ \ell \leq j < k}} \Delta_j \leq \sum_{\substack{j \in K \\ \ell \leq j}} \Delta_j \to 0 \text{ für } \ell \to \infty \text{ (Reihenreste).}$$

Damit ist $\{x_k\}$ eine Cauchyfolge und konvergiert gegen ein $\bar{x}$. Aus der Stetigkeit von f' folgt $\|\nabla_M f(\bar{x})\|_M \geq \varepsilon$. Nun wenden wir Lemma 6.7 mit $x = \bar{x}$ und $\eta = \eta_2$ an und erhalten $\Delta > 0$ und einen Index $L \in \mathbb{N}$, sodass $\rho_k(s_k) > \eta_2$ gilt, falls $\Delta_k \leq \Delta$ und $k \geq L$ gilt.

Induktiv zeigen wir nun

$$\Delta_k \geq \min\{\Delta_L, \gamma_1 \Delta\} \quad \text{für alle } k \geq L, \tag{6.12}$$

die Trust-Region-Radien bleiben also nach unten beschränkt. Für $k = L$ gilt es trivialerweise. Sei nun ein $k \geq L$ gegeben, das (6.12) erfüllt. Ist $\Delta_k < \Delta$, dann ist $\rho_k(s_k) > \eta_2$. Somit gilt

$$\Delta_{k+1} = \gamma_2 \Delta_k \geq \Delta_k \geq \min\{\Delta_L, \gamma_1 \Delta\}.$$

Andernfalls gilt

$$\Delta_{k+1} \geq \gamma_1 \Delta_k \geq \gamma_1 \Delta.$$

In jedem Fall gilt (6.12) für $k+1$. Dies zeigt (6.12) für alle $k \geq L$, steht aber im Widerspruch zu $\{\Delta_k\}_{k \in K} \to 0$, was aus $(*)$ folgt.

Nun sei f' gleichmäßig stetig auf $\{x_k\}$, und wir zeigen (6.11). Angenommen, (6.11) gilt nicht. Dann gibt es $\varepsilon > 0$, sodass $\|g_k\|_M = \|\nabla_M f(x_k)\|_M \geq 2\varepsilon$ für $k \in K_{2\varepsilon}$ aus einer unendlichen Menge $K_{2\varepsilon}$ gilt. Wegen der gleichmäßigen Stetigkeit von f' gibt es ein $\delta > 0$, sodass

$$\|\nabla_M f(x_\ell)\|_M \geq \varepsilon \text{ für alle } x_\ell \text{ mit der Eigenschaft } \|x_\ell - x_k\|_M \leq \delta \text{ für ein } k \in K_{2\varepsilon} \tag{$**$}$$

gilt. Es sei nun K_ε die Menge erfolgreicher Schritte k mit $\|g_k\|_M \geq \varepsilon$. Aus Lemma 6.9 folgt

$$\sum_{k \in K_\varepsilon} \Delta_k < \infty.$$

Damit gibt es $k \in K_{2\varepsilon}$ mit

$$\sum_{\substack{j \in K_\varepsilon \\ j \geq k}} \Delta_j < \delta \quad \text{(Reihenreste).}$$

Wir zeigen nun induktiv, dass $\|x_\ell - x_k\|_M < \delta$ für $\ell \geq k$ gilt. Für $\ell = k$ ist dies trivial. Angenommen, es gibt $\ell \geq k$, sodass $\|x_j - x_k\|_M < \delta$ für $j = k, \ldots, \ell$ gilt. Nach $(**)$ gilt insbesondere $j \in K_\varepsilon$ für alle erfolgreichen Schritte j mit $k \leq j \leq \ell$. Somit folgt

$$\|x_{\ell+1} - x_k\|_M \leq \sum_{\substack{j \in K_\varepsilon \\ k \leq j < \ell+1}} \Delta_j \leq \sum_{\substack{j \in K_\varepsilon \\ j \geq k}} \Delta_j < \delta.$$

Vollständige Induktion liefert also $\|x_\ell - x_k\|_M < \delta$ für alle $\ell \geq k$, und nach $(**)$ folgt $\|g_\ell\|_M \geq \varepsilon$ für alle $\ell \geq k$. Das steht im Widerspruch zu (6.10).

Bemerkung 6.11 (zu den Konvergenzaussagen von Trust-Region-Verfahren)**.**

Für die globalen Konvergenzaussagen ist es nicht streng erforderlich, dass die Schrittvorschläge $\|s_k\|_M \leq \Delta_k$ erfüllen. Eine relaxierte Form $\|s_k\|_M \leq \beta\,\Delta_k$ ist ebenso möglich, um mehr Flexibilität im Verfahren zu erhalten. Die Beweise müssen dann geringfügig angepasst werden.

§ 6.2 Schnelle lokale Konvergenz

In diesem Abschnitt zeigen wir, dass unter gewissen Voraussetzungen das allgemeine Trust-Region-Verfahren Algorithmus 6.1 in ein lokales (inexaktes) Newton-Verfahren übergeht. Das bedeutet, dass die Trust-Region-Beschränkung $\|s_k\|_M \leq \Delta_k$ ab einem gewissen Index inaktiv wird. Sofern man dann die Teilprobleme hinreichend genau löst (Stichwort *forcing sequence* wie beim inexakten Newton-Verfahren in § 5.5.1), kann man q-superlineare oder sogar q-quadratische Konvergenz erhalten.

Wir beschränken uns hier auf die Wahl $H_k = \nabla^2 f(x_k)$, also sogenannte **Trust-Region-Newton-Verfahren**. Es sind aber auch Quasi-Newton-Aufdatierungen für H_k möglich[58].

Satz 6.12 (Übergang zu schneller lokaler Konvergenz)**.**

Die Funktion f sei zweimal stetig differenzierbar (eine C^2-Funktion). Die Folge $\{x_k\}$ sei von Algorithmus 6.1 mit $H_k = \nabla^2 f(x_k)$ unter Beachtung von Voraussetzung 6.6 (a), d. h. (6.8) erzeugt. Die Sublevelmenge $\mathcal{M}_f(x_0)$ sei kompakt.

(a) Der Punkt x^* sei ein Häufungspunkt der Folge $\{x_k\}$ mit $\nabla^2 f(x^*) \succ 0$. Dann konvergiert die ganze Folge $\{x_k\}$ gegen x^*.

(b) Es gibt einen Index $k_0 \in \mathbb{N}$, sodass für alle $k \geq k_0$ gilt:

(*i*) $\rho_k(s_k) > \eta_1$ (Schritt wird akzeptiert)

(*ii*) $\nabla^2 f(x_k) \succ 0$ und $\|\nabla^2 f(x_k)^{-1}\,\nabla f(x_k)\|_M \leq \dfrac{\Delta_k}{2}$.

(c) Gilt mit einer Nullfolge $\{\eta_k\}$ die Abschätzung

$$\|\nabla^2 f(x_k)\,s_k + \nabla f(x_k)\|_{M^{-1}} \leq \eta_k\,\|\nabla f(x_k)\|_{M^{-1}}, \tag{6.13}$$

dann konvergiert die Folge $\{x_k\}$ q-superlinear.

Die Aussage (*ii*) bedeutet, dass ab einem gewissen Index k_0 die vollen exakten Newton-Schritte brauchbare Schrittvorschläge wären, die nicht nur kompatibel mit der Beschränkung $\|s_k\|_M \leq \Delta_k$ sind, sondern „satt“ innerhalb der *trust region* liegen.

Die Bedingung (6.13) verlangt, dass der Schrittvorschlag s_k „irgendwann“ dem exakten Newton-Schritt im Sinne der durch eine *forcing sequence* vorgegebenen Genauigkeit nahekommt; vgl. die Bedingung

$$\|\zeta_k\|_{M^{-1}} = \|\nabla^2 f(x_k)\,d_k + \nabla f(x_k)\|_{M^{-1}} \leq \eta_k\,\|\nabla f(x_k)\|_{M^{-1}} \tag{5.27}$$

beim inexakten Newton-Verfahren mit Liniensuche.

[58] Im Unterschied zu den Quasi-Newton-Liniensuch-Verfahren aus § 5.5.2 ist man nicht mehr auf die positive Definitheit von H_k angewiesen! Ein weiterer Unterschied ist, dass man mit Update-Formeln für die Modell-Hessematrix H_k arbeitet und nicht für deren Inverse B_k, da man für die inexakte (iterative) Lösung der Trust-Region-Teilprobleme (6.1) Matrix-Vektor-Produkte mit H_k benötigt.

Beweis. Schritt (i): Wir zeigen zuerst mit Satz 6.10, dass x^* ein stationärer Punkt ist.

Da die Sublevelmenge $\mathcal{M}_f(x_0)$ kompakt ist, ist f auf dieser Menge nach unten beschränkt. Alle Iterierten befinden sich aufgrund der Abstiegseigenschaft des Verfahrens in $\mathcal{M}_f(x_0)$. Weiter folgt aus der Kompaktheit, dass $H_k = \nabla^2 f(x_k)$ beschränkt ist, also gilt (6.9), d. h., Voraussetzung 6.6 (b) ist automatisch erfüllt. Die stetige Funktion f' ist auf der kompakten Menge $\{x_k\}$ gleichmäßig stetig. Satz 6.10 liefert also $\nabla_M f(x_k) = g_k \to 0$ und somit $f'(x^*) = 0$.

Schritt (ii): Wir zeigen: Die gesamte Folge $\{x_k\}$ konvergiert gegen x^*.

In einer Umgebung von x^* ist die Hessematrix (uniform) positiv definit, also $\lambda_{\min}(\nabla^2 f(x); M) > \delta$ für $\|x - x^*\|_M \le \varepsilon$. Analog zum Beweis von Lemma 5.25 und von Lemma 5.35 kann man daher zeigen:

$$\delta \, \|x - x^*\|_M \le \|\nabla_M f(x)\|_M \le C \, \|x - x^*\|_M \tag{$*$}$$

für alle x mit $\|x - x^*\|_M \le \varepsilon$.

Wegen $\|\nabla_M f(x_k)\|_M \to 0$ gibt es ein $k_0 \in \mathbb{N}$, sodass

$$\|\nabla_M f(x_k)\|_M \le \frac{\varepsilon \, \delta}{2\, C \, \delta^{-1} + 1} \quad \text{für alle } k \ge k_0$$

bleibt. Da x^* nach Voraussetzung ein Häufungspunkt von $\{x_k\}$ ist, gibt es ein $k_1 \ge k_0$ mit

$$\|x_{k_1} - x^*\|_M \le \frac{\varepsilon}{2\, C \, \delta^{-1} + 1} \le \varepsilon. \tag{$**$}$$

Aus der Bedingung (6.8b) folgt

$$\begin{aligned} 0 &> -\mathrm{pred}_k(s_k) \\ &= f'(x_k)\, s_k + \frac{1}{2}\, s_k^\top H_k \, s_k \\ &= (\nabla_M f(x_k),\, s_k)_M + \frac{1}{2}\, s_k^\top \nabla^2 f(x_k)\, s_k \\ &\ge -\|\nabla_M f(x_k)\|_M \, \|s_k\|_M + \frac{\delta}{2}\, \|s_k\|_M^2 \end{aligned}$$

und somit

$$0 < \frac{\delta}{2}\, \|s_k\|_M \le \|\nabla_M f(x_k)\|_M \tag{$***$}$$

für alle x_k mit der Eigenschaft $\|x_k - x^*\|_M \le \varepsilon$ (insbesondere für k_1).

Wir zeigen jetzt induktiv $\|x_k - x^*\|_M \le \frac{\varepsilon}{2\, C\, \delta^{-1}+1}$ für *alle* $k \ge k_1$. Der Induktionsanfang ist klar nach $(**)$. Wegen $x_{k+1} \in x_k + \{0, s_k\}$ gilt

$$\begin{aligned} \|x_{k+1} - x^*\|_M &\le \|s_k\|_M + \|x_k - x^*\|_M \\ &\le \frac{2}{\delta}\, \|\nabla_M f(x_k)\|_M + \|x_k - x^*\|_M \\ &\overset{(*)}{\le} \left(\frac{2}{\delta}\, C + 1\right) \|x_k - x^*\|_M \\ &\le \varepsilon \quad \text{nach Induktionsvoraussetzung.} \end{aligned}$$

Somit können wir $(*)$ für x_{k+1} anwenden und erhalten

$$\|x_{k+1} - x^*\|_M \le \frac{1}{\delta}\, \|\nabla_M f(x_{k+1})\|_M \le \frac{1}{\delta} \cdot \frac{\varepsilon \, \delta}{2\, C \, \delta^{-1} + 1} = \frac{\varepsilon}{2\, C \, \delta^{-1} + 1},$$

was die Induktion abschließt.

Wegen (∗) ist x^* aber der einzige stationäre Punkt von f in $\{x \in \mathbb{R}^n : \|x - x^*\|_M \leq \varepsilon\}$, und nach Satz 6.10 sind alle Häufungspunkte von $\{x_k\}$ stationäre Punkte. Also konvergiert die ganze Folge $\{x_k\}$ gegen x^*, was Behauptung (a) beweist.

Schritt (*iii*): Für den Nachweis der Aussage (b) zeigen wir zuerst, dass allein aufgrund der Konvergenz $x_k \to x^*$ bereits $\rho_k(s_k) \to 1$ folgt.

Für x_k mit $\|x_k - x^*\|_M \leq \varepsilon$ gilt

$$\begin{aligned}
\operatorname{pred}_k(s_k) &\geq \alpha \, \|g_k\|_M \, \min\Big\{\Delta_k, \frac{\|g_k\|_M}{\max\{0, \lambda_{\max}(H_k; M)\}}\Big\} \quad \text{wegen (6.8b)} \\
&\geq \frac{\alpha\,\delta}{2} \, \|s_k\|_M \, \min\Big\{\|s_k\|_M, \frac{\delta \, \|s_k\|_M}{2\,C}\Big\} \quad \text{wegen (6.8a) und } (***) \\
&= c \, \|s_k\|_M^2
\end{aligned}$$

mit einer Konstanten c. Eine Taylorentwicklung liefert mit einem $\xi_k \in (0, 1)$

$$\begin{aligned}
|\rho_k(s_k) - 1| &= \frac{|\operatorname{ared}_k(s_k) - \operatorname{pred}_k(s_k)|}{\operatorname{pred}_k(s_k)} \\
&= \frac{\big|f(x_k) - f(x_k + s_k) + f'(x_k)\, s_k + \frac{1}{2}\, s_k^\top \nabla f(x_k)\, s_k\big|}{\operatorname{pred}_k(s_k)} \\
&\leq \frac{\big|s_k^\top \big[\nabla^2 f(x_k + \xi_k\, s_k) - \nabla^2 f(x_k)\big]\, s_k\big|}{2\,c\,\|s_k\|_M^2} \\
&\leq \frac{\|\nabla^2 f(x_k + \xi_k\, s_k) - \nabla^2 f(x_k)\|_{M,M}}{2\,c} \, \frac{\|s_k\|_M^2}{\|s_k\|_M^2} \to 0
\end{aligned}$$

wegen der gleichmäßigen Stetigkeit von $\nabla^2 f$ „in der Nähe der $\{x_k\}$“ unter Beachtung von $x_k \to x^*$ und $s_k \to 0$, vgl. Beweis von Lemma 5.21.[59] Es gibt also einen Index $k_0 \in \mathbb{N}$ mit $\rho_k(s_k) > \eta_1$ für alle $k \geq k_0$. Ab Schritt k_0 sind demnach alle Schritte erfolgreich, und es gilt $\Delta_k \geq \Delta_{k_0}$.

Wegen der gleichmäßig beschränkten Invertierbarkeit von $\nabla^2 f(x_k)$ für hinreichend große k folgt für ein $c > 0$

$$\|\nabla^2 f(x_k)^{-1} \nabla f(x_k)\|_M \leq c \, \|\nabla f(x_k)\|_{M^{-1}} = c \, \|\nabla_M f(x_k)\|_M .$$

Für große k ist dies kleiner als $\Delta_k/2$, da — wie gerade gesehen — $\Delta_k \geq \Delta_{k_0}$ bleibt. Damit gilt ab einem gewissen Iterationsindex stets $k \in K$, und wegen $\rho_k(s_k) \geq \eta_1$ werden alle Schritte akzeptiert. Das Verfahren geht also in das lokale inexakte Newton-Verfahren über, und die q-superlineare Konvergenz folgt aus Satz 5.38.

Ende 17. V

[59] Die verwendete Abbildungsnorm einer symmetrischen Matrix A ist definiert als $\|A\|_{M,M} := \max\limits_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \frac{x^\top A\, x}{\|x\|_M^2}$

§ 6.3 Lösung des Trust-Region-Teilproblems

In diesem Abschnitt wollen wir uns genauer mit dem Trust-Region-Teilproblem (6.1), also mit einer Aufgabe der Form

$$\begin{aligned} \text{Minimiere} \quad & q(s) = f + b^\top s + \frac{1}{2}\, s^\top H\, s, \quad s \in \mathbb{R}^n \\ \text{unter} \quad & \|s\|_M \le \Delta \end{aligned} \tag{6.14}$$

beschäftigen, wobei $f \in \mathbb{R}$, $b \in \mathbb{R}^n$ und $H \in \mathbb{R}^{n\times n}$ symmetrisch ist und $\Delta > 0$ gilt. (Der Übersichtlichkeit halber lassen wir den Iterationsindex k hier wieder weg.) Obwohl für die globale Konvergenz eines Trust-Region-Verfahrens bereits der Cauchy-Punkt als hinreichend genaue Lösung ausreicht (Satz 6.10) und die schnelle lokale Konvergenz erst dann einsetzen kann, wenn die Trust-Region-Beschränkung $\|s\|_M \le \Delta$ keine Rolle spielt, also inaktiv ist (Satz 6.12), wollen wir dennoch zunächst der Vollständigkeit halber die exakte(n) globale(n) Lösung(en) der Aufgabe (6.14) charakterisieren.

Beachte: Aufgrund des möglichen Vorkommens negativer Eigenwerte in H ist die Aufgabe (6.14) i. A. nicht konvex!

Satz 6.13 (Charakterisierung der globalen Lösungen des Trust-Region-Teilproblems)**.**

(a) Es sei $s \in \mathbb{R}^n$ ein globales Minimum von (6.14). Dann existiert ein $\mu \in \mathbb{R}$, sodass gilt:

$$\mu \ge 0, \quad \|s\|_M - \Delta \le 0, \quad \mu\,(\|s\|_M - \Delta) = 0 \tag{6.15a}$$

$$(H + \mu\, M)\, s = -b \tag{6.15b}$$

$$H + \mu\, M \text{ ist positiv semidefinit.} \tag{6.15c}$$

Die Zahl μ ist eindeutig bestimmt.

(b) Umgekehrt seien $(s, \mu) \in \mathbb{R}^n \times \mathbb{R}$ gegeben, sodass (6.15) erfüllt ist. Dann ist s ein globales Minimum von (6.14).

(c) Gilt zusätzlich zur Voraussetzung (b), dass $H + \mu\, M$ positiv definit ist, dann ist s das *eindeutige* globale Minimum von (6.14).

Beweis.

(a) Es sei s ein globales Minimum von (6.14). Dann ist $\|s\|_M \le \Delta$ klar.

Fall 1: Es gilt $\|s\|_M < \Delta$.
Dann ist natürlich insbesondere s auch ein lokales Minimum der unrestringierten Aufgabe

$$\text{Minimiere} \quad q(s), \quad s \in \mathbb{R}^n.$$

Folglich gilt aufgrund der notwendigen Optimalitätsbedingungen 1. und 2. Ordnung (Satz 3.2 und Satz 3.3)

$$\nabla q(s) = H\, s + b = 0$$

und $H \succeq 0$, was (6.15) mit $\mu = 0$ zeigt.[60] Aufgrund der Komplementarität ist $\mu = 0$ auch eindeutig.

Fall 2: Es gilt $\|s\|_M = \Delta$ (insbesondere $s \neq 0$).
Zu zeigen ist zunächst: Es existiert $\mu \geq 0$, sodass $(H + \mu M)\, s = -b$ gilt. Wir führen einen Widerspruchsbeweis und nehmen also an, dass für alle $\mu \geq 0$ die Beziehung $(H + \mu M)\, s \neq -b$ gilt. Für den Fall $\mu = 0$ folgt daraus $y := \nabla q(s) = H\, s + b \neq 0$. Für die Fälle $\mu > 0$ folgt daraus, dass die Vektoren y und $M\, s$ nicht anti-parallel sind. Also sind auch $M^{-1}y$ und s nicht anti-parallel. Für den Winkel α (im M-Skalarprodukt) zwischen diesen gilt also

$$\cos\alpha = \frac{(M^{-1}y)^\top M\, s}{\|M^{-1}y\|_M\, \|s\|_M} = \frac{y^\top s}{\|y\|_{M^{-1}}\, \|s\|_M} > -1.$$

Wir setzen v als Richtung der Winkelhalbierenden zwischen $-M^{-1}y$ und $-s$:

$$v := -\frac{M^{-1}y}{\|y\|_{M^{-1}}} - \frac{s}{\|s\|_M} \neq 0.$$

Dann gilt

$$\begin{aligned} y^\top v &= y^\top \left(-\frac{M^{-1}y}{\|y\|_{M^{-1}}} - \frac{s}{\|s\|_M} \right) \\ &= -\|y\|_{M^{-1}} - \frac{y^\top s}{\|s\|_M} \frac{\|y\|_{M^{-1}}}{\|y\|_{M^{-1}}} \\ &= -\|y\|_{M^{-1}} \left(1 + \cos\alpha\right) < 0. \end{aligned}$$

Damit ist $q'(s)\, v = y^\top v < 0$, also v eine Abstiegsrichtung für q an der Stelle s. Wegen

$$\begin{aligned} \left[\frac{\mathrm{d}}{\mathrm{d}t} \frac{1}{2} \|s + t\, v\|_M^2\right]_{t=0} &= v^\top M\, s = \left(-\frac{M^{-1}y}{\|y\|_{M^{-1}}} - \frac{s}{\|s\|_M} \right)^\top M\, s \\ &= -\frac{y^\top s}{\|y\|_{M^{-1}}} \frac{\|s\|_M}{\|s\|_M} - \|s\|_M \\ &= -\|s\|_M \big(\cos\alpha + 1\big) < 0 \end{aligned}$$

ergibt sich für kleine $t > 0$: $\|s + t\, v\|_M < \|s\|_M = \Delta$, sodass man einen zulässigen Punkt mit echt kleinerem Funktionswert als s erhalten kann. Dies steht im Widerspruch zur Optimalität von s. Demzufolge muss es auch im Fall 2 möglich sein, für ein $\mu \geq 0$ die Bedingungen (6.15a) und (6.15b) zu erfüllen.

Wie man leicht sieht, ist auch in diesem Fall μ eindeutig, denn die Annahme $H\, s + \mu_1\, M\, s = -b = H\, s + \mu_2\, M\, s$ führt auf $(\mu_1 - \mu_2)\, M\, s = 0$, was wegen $s \neq 0$ und der positiven Definitheit von M nur für $\mu_1 = \mu_2$ möglich ist.

Es bleibt zu zeigen, dass $H + \mu\, M$ positiv semidefinit ist, also (6.15c). Wir betrachten dazu zunächst eine Richtung $d \in \mathbb{R}^n$ mit der Eigenschaft $d^\top M\, s < 0$. Wir müssen zeigen:

$$d^\top (H + \mu\, M)\, d \geq 0.$$

Setze dazu $t := \dfrac{-2\, d^\top M\, s}{\|d\|_M^2} > 0$. Dann gilt

$$\|s + t\, d\|_M^2 = \|s\|_M^2 + 2\, t\, d^\top M\, s + t^2\, \|d\|_M^2 = \|s\|_M^2 \leq \Delta^2.$$

Aus der globalen Optimalität von s erhalten wir

$$\begin{aligned}
0 \le q(s+t\,d) - q(s) &= q'(s)\,(t\,d) + \frac{1}{2}(t\,d)^\top H\,(t\,d) \\
&= t\,y^\top d + \frac{t^2}{2} d^\top H\,d \\
&= -t\,\mu\,s^\top M\,d + \frac{t^2}{2} d^\top H\,d, \quad \text{denn } y = H\,s + b = -\mu\,M\,s \\
&= \frac{t^2}{2}\mu\,\|d\|_M^2 + \frac{t^2}{2} d^\top H\,d, \quad \text{denn } \frac{t}{2}\,\|d\|_M^2 = -d^\top M\,s \\
&= \frac{t^2}{2} d^\top (H + \mu\,M)\,d.
\end{aligned}$$

Damit ist $d^\top (H+\mu\,M)\,d \ge 0$ zunächst für solche Richtungen $d \in \mathbb{R}^n$ gezeigt, die $d^\top M\,s < 0$ erfüllen. Da es auf das Vorzeichen von d nicht ankommt, gilt die Behauptung natürlich ebenfalls für Richtungen mit $d^\top M\,s > 0$. Im verbleibenden Fall $d^\top M\,s = 0$ folgt die Behauptung aus Stetigkeitsgründen.

(b) Es seien $(s,\mu) \in \mathbb{R}^n \times \mathbb{R}$ gegeben, sodass (6.15) erfüllt ist. Es sei $\overline{s} \in \mathbb{R}^n$ mit $\|\overline{s}\|_M \le \Delta$ ein beliebiger zulässiger Vergleichspunkt, $d := \overline{s} - s$ und wie oben $y := \nabla q(s) = H\,s + b$. Wir schätzen ab:

$$\begin{aligned}
q(\overline{s}) - q(s) &= y^\top d + \frac{1}{2} d^\top H\,d \\
&= -\mu\,s^\top M\,d + \frac{1}{2} d^\top H\,d, \quad \text{denn } y = H\,s + b = -\mu\,M\,s \text{ nach (6.15b)} \\
&\ge -\mu\,s^\top M\,d - \frac{1}{2}\mu\,\|d\|_M^2 \quad \text{nach (6.15c)} \\
&= -\frac{\mu}{2}\big(2\,s^\top M\,d + \|d\|_M^2\big) \\
&= -\frac{\mu}{2}\big(\|d+s\|_M^2 - \|s\|_M^2\big) \\
&= -\frac{\mu}{2}\big(\|\overline{s}\|_M^2 - \|s\|_M^2\big) \\
&\ge 0.
\end{aligned}$$

Die letzte Ungleichung gilt im Fall $\mu = 0$ trivialerweise. Andernfalls impliziert (6.15a) gerade $\|s\|_M^2 = \Delta^2$, und nach Annahme gilt $\|\overline{s}\|_M^2 \le \Delta^2$. Damit ist die globale Optimalität von s für (6.14) gezeigt.

(c) Falls $H + \mu\,M$ sogar positiv definit ist, dann kann für $\overline{s} \ne s$, also $d \ne 0$, die obige Abschätzung verschärft werden:

$$\begin{aligned}
q(\overline{s}) - q(s) &= -\mu\,s^\top M\,d + \frac{1}{2} d^\top H\,d \\
&> -\mu\,s^\top M\,d - \frac{1}{2}\mu\,\|d\|_M^2 = \cdots = -\frac{\mu}{2}\big(\|\overline{s}\|_M^2 - \|s\|_M^2\big) \ge 0.
\end{aligned}$$

Das zeigt, dass s sogar das eindeutige globale Minimum von (6.14) ist.

[60] In diesem Fall ist q wegen $H \succeq 0$ konvex, und s ist gleichzeitig auch ein globales Minimum der unrestringierten Aufgabe.

Bemerkung 6.14 (zur Charakterisierung der globalen Lösungen)**.**

(a) Bemerkenswert an der Aussage von Satz 6.13 ist, dass die Optimalitätsbedingungen (6.15) gleichzeitig notwendig und hinreichend sind, selbst wenn die Aufgabe (6.14) nicht konvex ist.

(b) Alternativ könnte man auch die Optimalitätstheorie nach Karush-Kuhn-Tucker („Lagrangesche Multiplikatorenregel“) benutzen, um zu beweisen, dass (6.15a) und (6.15b) *notwendige* Bedingungen (sogar bereits für lokale Minima von (6.14)) darstellen. Dabei zeigt sich, dass die Zahl μ als Lagrange-Multiplikator zur Nebenbedingung

$$\frac{1}{2}\left(\|s\|_M^2 - \Delta^2\right) \leq 0$$

verstanden werden kann, die natürlich äquivalent zu $\|s\|_M \leq \Delta$ ist. Wie man leicht sieht, ist die *linear independence constraint qualification* (LICQ) hier immer erfüllt.[61] Bei diesem Vorgehen bekommt man aber nicht die Bedingung (6.15c) heraus, die — wie wir in den Beweisteilen (a) und (b) gesehen haben — charakteristisch für *globale* Minima ist.

(c) Wie in Martínez (1994) bewiesen wurde, besitzt (6.14) neben seinen globalen Minimierern höchstens ein weiteres lokales Minimum, das nicht auch globales Minimum ist.

Aufbauend auf der Charakterisierung (6.15) kann man Verfahren zur *exakten* Lösung des Trust-Region-Teilproblems (6.1) bzw. (6.14) angeben. Das gängigste Vorgehen basiert auf dem (eindimensionalen) Newton-Verfahren für die Gleichung[62]

$$\left\|(H + \mu M)^{-1} b\right\|_M - \Delta = 0$$

zur Bestimmung des Multiplikators μ, falls nicht $\mu = 0$ und dazu passendes s bereits zu einer Lösung von (6.15) führt. Näheres siehe (Nocedal and Wright, 2006, Chapter 4.3).

Im Rest des Abschnitts betrachten wir noch eine konkrete Möglichkeit, das Trust-Region-Teilproblem *inexakt* zu lösen. Dabei wollen wir

(a) die Voraussetzungen für die globale Konvergenz (Satz 6.10) erfüllen, d. h. die Einhaltung der Trust-Region-Beschränkung und die Fraction-of-Cauchy-decrease-Bedingung beachten, siehe (6.8),

(b) die Bedingungen für die schnelle lokale Konvergenz (Satz 6.12) ebenfalls erfüllen.

[61] Im Fall $\|s\|_M < \Delta$ ist die einzige Beschränkung inaktiv und nichts zu zeigen. Andernfalls ist $\|s\|_M = \Delta > 0$, also $s \neq 0$. Damit ist der Gradient der einzigen aktiven Nebenbedingung, also $M\,s$, ungleich 0, d. h. linear unabhängig.

[62] Eigentlich wird sogar die äquivalente Gleichung $\frac{1}{\|(H+\mu M)^{-1} b\|_M} - \frac{1}{\Delta} = 0$ verwendet, die numerisch vorteilhafter ist.

Es wird sich herausstellen, dass wiederum eine clevere Modifikation des CG-Verfahrens — das sogenannte **Steihaug(-Toint)-CG-Verfahren**[63] — diese Forderungen umsetzen kann.[64]

Dieses CG-Verfahrens wird angewandt auf das LGS $H\,s = -b$. Im Falle, dass H positiv definit ist, sind dies ja gerade die notwendigen und hinreichenden Optimalitätsbedingungen zur eindeutigen Lösung des *unrestringierten* Trust-Region-Teilproblems (6.14), also mit $\Delta = \infty$. In der Steihaug(-Toint)-Variante des CG-Verfahrens gibt es zwei wesentliche Unterschiede zur Ausgangsversion:[65]

(1) Ähnlich wie beim truncated CG-Verfahren (Algorithmus 5.39) starten wir das Verfahren mit $s_0 := 0$ und reagieren auf das Auftreten einer Suchrichtung p_ℓ mit nicht-positiver Krümmung: $p_\ell^\top H\,p_\ell \leq 0$. Im Unterschied zum sofortigen Abbruch beim truncated CG-Verfahren gehen wir jedoch noch einen Schritt zu $s_{\ell+1} := s_\ell + \alpha_\ell^*\,p_\ell$. Dabei kommt jedoch nicht die übliche CG-Schrittlänge zum Einsatz, sondern wir wählen α_ℓ^* möglichst groß, sodass $s_{\ell+1}$ auf dem Rand der *trust region* liegt. Dann stoppt das Verfahren.

(2) Im Fall, dass der aktuelle Schritt die *trust region* verlassen würde, d. h. falls $\|s_\ell + \alpha_\ell\,p_\ell\|_M > \Delta$ gilt, gehen wir nicht die volle Schrittlänge α_ℓ, sondern wiederum nur bis zum Rand der *trust region* und stoppen dann das Verfahren.

Treten diese Fälle nicht auf, dann wird das CG-Verfahren abgebrochen, sobald die Lösung hinreichend genau ist. Im Einklang mit Satz 6.12 geschieht dies wieder mit der relativen Norm des Residuums, wobei die Toleranz η durch eine geeignete *forcing sequence* vorgegeben wird (vgl. (5.28)):

$$\frac{\|\text{Residuum zu } s_\ell\|_{M^{-1}}}{\|\text{Residuum zu } 0\|_{M^{-1}}} = \frac{\|H\,s_\ell + b\|_{M^{-1}}}{\|b\|_{M^{-1}}} \leq \eta$$

Im Fall, dass der Iterationsindex k des Trust-Region-Verfahrens in der Menge K liegt (siehe Satz 6.12), kann man garantieren, dass das Steihaug(-Toint)–CG-Verfahren nicht aus den beiden obigen Gründen abbricht. Somit ist die Bedingung (6.13) an die Genauigkeit der Lösung des Trust-Region-Teilproblems erfüllt. Wir erhalten somit aus Satz 6.12 bei Vorliegen der weiteren Voraussetzungen die lokale q-superlineare Konvergenz.

Das Steihaug(-Toint)-CG-Verfahren zur inexakten Lösung von (6.14) hat folgende Gestalt:

Algorithmus 6.15 (Steihaug(-Toint)-CG-Verfahren)**.**
Eingabe: (negative) rechte Seite $b \in \mathbb{R}^n$
Eingabe: Matrix H (oder Matrix-Vektor-Produkte mit H); symmetrisch, aber nicht notwendig positiv definit
Eingabe: spd Matrix M (oder Matrix-Vektor-Produkte mit M^{-1})

[63] nach Steihaug (1983) und Toint (1981)

[64] Für andere inexakte Lösungsverfahren des Trust-Region-Teilproblems verweisen wir auf die Literatur, insbesondere das **Dogleg-Verfahren** und das **Verfahren der Minimierung über zweidimensionale Unterräume** (*two-dimensional subspace minimization*), (Nocedal and Wright, 2006, Chapter 4.1).

[65] Wir beschreiben das Verfahren (wie schon beim truncated CG-Verfahren) in einer problemangepassten Notation, verwenden also s_ℓ für die Iterierten und p_ℓ für die Suchrichtungen.

Eingabe: Trust-Region-Radius $\Delta > 0$
Ausgabe: näherungsweise Lösung des Trust-Region-Teilproblems (6.14)

Setze $\ell := 0$
Setze $s_0 := 0$
Setze $r_0 := +b$
Setze $p_0 := -M^{-1} r_0$
Setze $\delta_0 := -r_0^\top p_0$ $\{\delta_0 = \|r_0\|_{M^{-1}}^2\}$
while Abbruchkriterium nicht erfüllt **do**
  Setze $q_\ell := H\, p_\ell$
  Setze $\kappa_\ell := p_\ell^\top q_\ell$
  if $\kappa_\ell > 0$ **then**
    Setze $\alpha_\ell := \delta_\ell / \kappa_\ell$
    Setze $s_{\ell+1} := s_\ell + \alpha_\ell\, p_\ell$
    if $\|s_{\ell+1}\|_M > \Delta$ **then** { Schritt zu lang}
      Bestimme α_ℓ^* als die positive Lösung von $\|s_\ell + \alpha\, p_\ell\|_M = \Delta$
      Setze $s_{\ell+1} := s_\ell + \alpha_\ell^*\, p_\ell$
      Setze $\ell := \ell + 1$
      Abbruch der **while**-Schleife
    end if
    Setze $r_{\ell+1} := r_\ell + \alpha_\ell\, q_\ell$
    Setze $p_{\ell+1} := -M^{-1} r_{\ell+1}$
    Setze $\delta_{\ell+1} := -r_{\ell+1}^\top p_{\ell+1}$ $\{\delta_{\ell+1} = \|r_{\ell+1}\|_{M^{-1}}^2\}$
    Setze $\beta_{\ell+1} := \delta_{\ell+1} / \delta_\ell$
    Setze $p_{\ell+1} := p_{\ell+1} + \beta_{\ell+1}\, p_\ell$
    Setze $\ell := \ell + 1$
  else { Suchrichtung nicht-positiver Krümmung}
    Bestimme α_ℓ^* als die positive Lösung von $\|s_\ell + \alpha\, p_\ell\|_M = \Delta$
    Setze $s_{\ell+1} := s_\ell + \alpha_\ell^*\, p_\ell$
    Setze $\ell := \ell + 1$
    Abbruch der **while**-Schleife
  end if
end while
return s_ℓ

Bemerkung 6.16 (zum Steihaug(-Toint)-CG-Verfahren)**.**

(a) Die erste Iterierte s_1 ist gerade der Cauchy-Punkt der Aufgabe, vgl. (6.3).

(b) Wir betrachten die drei Fälle, die in einer Iteration auftreten können:

  (*i*) Falls man in Iteration ℓ einen gewöhnlichen CG-Schritt macht, dann gilt

$$q(s_{\ell+1}) = q(s_\ell) - \frac{1}{2} \underbrace{\alpha_\ell}_{>0} \underbrace{\delta_\ell}_{>0},$$

  siehe Bemerkung 4.6.

  (*ii*) Gilt dagegen $\kappa_\ell > 0$, aber der volle Schritt $s_\ell + \alpha_\ell\, p_\ell$ landet außerhalb der *trust region*, dann ist es naheliegend, die konvexe Funktion

$$\alpha \mapsto \varphi(\alpha) := f + b^\top (s_\ell + \alpha\, p_\ell) + \frac{1}{2} (s_\ell + \alpha\, p_\ell)^\top H\, (s_\ell + \alpha\, p_\ell)$$

für einen möglichst großen Abstieg bis zum Rand der *trust region* zu verfolgen, und zwar in Richtung $\alpha > 0$, da

$$\varphi'(0) = b^\top p_\ell + s_\ell^\top H\, p_\ell = r_\ell^\top p_\ell = -\delta_\ell = -\|r_\ell\|_{M^{-1}}^2 < 0$$

ist.

(*iii*) Hat man dagegen nicht-positive Krümmung $\kappa_\ell \le 0$, dann ist die Funktion φ konkav, und wiederum liegt es nahe, in Richtung $\alpha > 0$ bis zum Rand zu gehen.

(c) Die obige Überlegung zeigt: Bricht das Verfahren nicht bereits mit der Iterierten s_1 ab, sondern läuft weiter, dann reduziert sich in den weiteren Iterationen der Zielfunktionswert von q monoton. Somit gilt für die zurückgegebene inexakte Lösung s_ℓ stets die Fraction-of-Cauchy-decrease-Bedingung (6.8b) mit $\alpha = 1/2$.

(d) Die Begründung für die Beendigung des Verfahrens, sobald die Iterierten im Begriff sind, die *trust region* zu verlassen, ist folgende: Wie wir nach Lemma 4.22 wissen, ist die Folge der Normen $\|s_\ell - 0\|_M$ streng monoton wachsend. Ließen wir das CG-Verfahren weiterlaufen, würden wir also nie wieder in die *trust region* zurückkehren.

(e) Zur rekursiven Mitberechnung der Normen $\|s_\ell\|_M$ ohne Verwendung der Matrix M verweisen wir auf (4.36). Es werden also (in der aktuellen Notation) die Größen

$$\eta_\ell := \|s_\ell - 0\|_M^2, \quad \zeta_\ell := (s_\ell - 0)^\top M\, p_\ell, \quad \gamma_\ell := \|p_\ell\|_M^2$$

mitgeführt.

(f) Mit deren Hilfe lässt sich dann auch die bis zum Rand benötigte Schrittweite α_ℓ^* durch Lösen der quadratischen Gleichung

$$\|s_\ell + \alpha\, p_\ell\|_M^2 = \eta_\ell + \alpha\, \zeta_\ell + \alpha^2 \gamma_\ell \stackrel{!}{=} \Delta^2$$

berechnen. Wegen $\eta_\ell = \|s_\ell\|_M^2 < \Delta$ gibt es genau eine positive Lösung, und zwar

$$\alpha_\ell^* := -\frac{\zeta_\ell}{2\,\gamma_\ell} + \frac{\zeta_\ell}{2\,\gamma_\ell}\Big(1 - 4\,\gamma_\ell\,(\eta_\ell - \Delta^2)\Big)^{1/2}.$$

Ende 18. V

KAPITEL 2

Verfahren der restringierten Optimierung

In diesem Kapitel betrachten wir noch überblicksartig die wichtigsten Verfahrensklassen zur Lösung restringierter Aufgaben

$$\left.\begin{array}{rll} \text{Minimiere} & f(x) & \text{über } x \in \Omega = \mathbb{R}^n \\ \text{sodass} & g_i(x) \le 0, & i = 1, \dots, m \\ \text{und} & h_j(x) = 0, & j = 1, \dots, p. \end{array}\right\} \qquad \textbf{(NLP)}$$

Hier sind $m, p \in \mathbb{N}_0$ die Anzahl der Ungleichungs- bzw. Gleichungsnebenbedingungen. Wir bezeichnen mit

$$X = \{x \in \mathbb{R}^n : g_i(x) \le 0 \text{ für alle } i = 1, \dots, m,\ h_j(x) = 0 \text{ für alle } j = 1, \dots, p\}$$

die zulässige Menge. Eine Ungleichungsbeschränkung $g_i(x) \le 0$ heißt **aktiv** im Punkt x, wenn $g_i(x) = 0$ gilt, **inaktiv** im Fall $g_i(x) < 0$ und **verletzt** im Fall $g_i(x) > 0$: Dafür definieren wir die Indexmengen

$$\begin{aligned} \mathcal{A}(x) &:= \big\{i \in \{1, \dots, m\} : g_i(x) = 0\big\} && \textbf{aktive Menge}, \\ \mathcal{I}(x) &:= \big\{i \in \{1, \dots, m\} : g_i(x) < 0\big\} && \textbf{inaktive Menge}. \end{aligned}$$

Zur Formulierung notwendiger Optimalitätsbedingungen erster Ordnung führen wir die **Lagrange-Funktion** $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ der Aufgabe (NLP) ein als

$$\begin{aligned} \mathcal{L}(x, \mu, \lambda) &= f(x) + \sum_{i=1}^{m} \mu_i\, g_i(x) + \sum_{j=1}^{p} \lambda_j\, h_j(x) \\ &= f(x) + \mu^\top g(x) + \lambda^\top h(x). \end{aligned}$$

Satz 6.17 (Notwendige Optimalitätsbedingungen erster Ordnung)**.**

Es sei x^* ein lokales Minimum von (NLP). An der Stelle x^* gelte eine *constraint qualification* (CQ).[1] Dann existieren **(Lagrange-)Multiplikatoren** $\mu^* \in \mathbb{R}^m$ und $\lambda^* \in \mathbb{R}^p$, sodass gilt:

$$\nabla_x \mathcal{L}(x^*, \mu^*, \lambda^*) = \left\{\begin{array}{ll} \nabla f(x^*) + \sum_{i=1}^{m} \mu_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \lambda_j^* \nabla h_j(x^*) \\ \nabla f(x^*) + g'(x^*)^\top \mu^* \qquad + h'(x^*)^\top \lambda^* \end{array}\right\} = 0, \tag{6.16a}$$

$$\mu^* \ge 0, \quad g(x^*) \le 0, \quad (\mu^*)^\top g(x^*) = 0, \tag{6.16b}$$

$$h(x^*) = 0. \tag{6.16c}$$

[1] CQs stellen sicher, dass $\mathcal{T}_X(x^*)^\circ = \mathcal{T}_X^{\text{lin}}(x^*)^\circ$, also dass die Polarkegel des Tangentialkegels und des Linearisierungskegels übereinstimmen; siehe Vorlesung *Grundlagen der Optimierung*. Die wichtigsten CQs sind LICQ $\Rightarrow$ MFCQ $\Rightarrow$ ACQ $\Rightarrow$ GCQ.

Beachte: Im Falle $m = p = 0$ reduziert sich (6.16) zur notwendigen Bedingung 1. Ordnung $\nabla f(x^*) = 0$ der unrestringierten Optimierung (Satz 3.2).

Die Bedingungen (6.16) heißen **Karush-Kuhn-Tucker-Bedingungen** (**KKT-Bedingungen**). Das Tripel (x^*, μ^*, λ^*) heißt **KKT-Punkt**. Manchmal wird auch x^* alleine als KKT-Punkt bezeichnet, falls dazu passende Multiplikatoren existieren, sodass (6.16) erfüllt ist. Man spricht weiterhin bei x^* von den primalen und bei (μ^*, λ^*) von den dualen Variablen der Aufgabe. Die Bedingung (6.16b) heißt eine **Komplementaritätsbedingung**.

Die KKT-Bedingungen werden in Algorithmen (analog zu $\|\nabla_M f(x_k)\|_M \leq \varepsilon$) dazu verwendet, eine Abbruchbedingung für das jeweilige Verfahren zu formulieren.

Ähnlich wie wir für Algorithmen für unrestringierte Probleme (Kapitel 1) i. A. nur sicherstellen konnten, dass sie gegen stationäre Punkte konvergieren, werden wir im restringierten Fall i. A. Konvergenz gegen KKT-Punkte erhalten. Zur weiteren Überprüfung, ob tatsächlich ein Minimum vorliegt, dienen dann notwendige bzw. hinreichende Bedingungen 2. Ordnung (hier nicht aufgeführt).

Die algorithmische Behandlung von Ungleichungsnebenbedingungen ist um einiges schwieriger als die Behandlung von Gleichungsnebenbedingungen. Ohne Ungleichungen reduzieren sich die KKT-Bedingungen (6.16) zu

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = \nabla f(x^*) + h'(x^*)^\top \lambda^* = 0, \tag{6.17a}$$

$$h(x^*) = 0, \tag{6.17b}$$

bestehen also selbst nur aus Gleichungen. Die Schwierigkeit in der Behandlung von Ungleichungen besteht i. W. darin, dass man von vornherein nicht weiß, ob sie in einer Lösung x^* aktiv oder inaktiv sein werden. Da es i. A. 2^m Möglichkeiten für die Struktur der aktiven/inaktiven Ungleichungen gibt, ist es in aller Regel ineffizient, diese durch systematisches Ausprobieren zu finden.

Verschiedene Klassen von Algorithmen in der restringierten Optimierung unterscheiden sich i. W. in der Art, wie sie mit den Nebenbedingungen umgehen. Wir geben hier nur grobe Basis-Algorithmen an. In jedem Fall müssen diese, um in der Praxis robust zu funktionieren, mit Globalisierungstechniken kombiniert werden.[2] Dazu stehen wieder Liniensuch-Techniken, Trust-Region-Techniken sowie Filter-Techniken (eingeführt in Fletcher and Leyffer (2002)) zur Verfügung.

Um nun Bedingungen zweiter Ordnung einzuführen, müssen wir $\mathcal{T}_X^{\text{lin}}(x^*)$ verfeinern.

Definition 6.18. Für ein zulässiges $x \in X$ bezeichnen wir mit

$$\mathcal{T}_X^{\text{krit}}(x) := \{d \in \mathcal{T}_X^{\text{lin}}(x) : \nabla f(x)^\top d = 0\}$$

den **kritischen Kegel**.

Ist (x^*, μ^*, λ^*) ein KKT-Punkt, dann gilt gerade

$$\mathcal{T}_X^{\text{krit}}(x^*) = \{d \in \mathcal{T}_X^{\text{lin}}(x^*) : \nabla g_i(x^*)^\top d = 0 \text{ für alle } i \in \mathcal{A}_>(x^*, \mu^*)\},$$

[2] Eine beliebte Aufgabensammlung zum Testen von Algorithmen für Aufgaben vom Typ (**NLP**) ist CUTE (Constrained and Unconstrained Testing Environment), CUTEr (Constrained and Unconstrained Testing Environment, revisited) und nun CUTEst (Constrained and Unconstrained Testing Environment on steroids).

wobei

$$\mathcal{A}_>(x^*, \mu^*) = \{i \in \mathcal{A}(x^*) : \mu_i^* > 0\},$$
$$\mathcal{A}_0(x^*, \mu^*) = \{i \in \mathcal{A}(x^*) : \mu_i^* = 0\}$$

die **stark** und **schwach aktiven Indizes** umfassen.

Satz 6.19 (Hinreichende Bedingung zweiter Ordnung)**.** Es sei (x^*, μ^*, λ^*) ein KKT-Punkt von (NLP) und für ein $\alpha > 0$ gelte

$$d^\top \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*, \mu^*)\, d \geq \alpha \|d\|^2 \quad \text{für alle } d \in \mathcal{T}_X^{\text{krit}}(x^*). \tag{SSC}$$

Dann gibt es für jedes $\beta \in (0, \alpha)$ ein $\varepsilon > 0$ mit

$$f(x) \geq f(x^*) + \frac{\beta}{2} \|x - x^*\|^2 \quad \text{für alle } x \in B_\varepsilon(x^*) \cap X.$$

Insbesondere ist x^* ein striktes lokales Minimum von (NLP).

Beachte: Wir benötigen hier die Hessematrix der Lagrangefunktion, da $\nabla^2 f(x^*)$ keine Information über die Krümmung der Nebenbedingungen enthält.

Unter LICQ erhält man auch eine notwendige Bedingung zweiter Ordnung mit $d^\top \nabla^2_{xx} \mathcal{L}(\cdot)\, d \geq 0$ für alle $d \in \mathcal{T}_X^{\text{krit}}(x^*)$. Wie im unrestringierten Fall ist die Lücke zwischen hinreichender und notwendiger Bedingung zweiter Ordnung so klein wie möglich.

Definition 6.20. Ein KKT-Punkt (x^*, μ^*, λ^*) heißt **strikt komplementär**, falls $\mathcal{A}_0(x^*, \mu^*) = \emptyset$, d. h.

$$g_i(x^*) = 0 \quad \Leftrightarrow \quad \mu_i^* > 0,$$

gilt.

Im Falle strikter Komplementarität ist der kritische Kegel $\mathcal{T}_X^{\text{krit}}(x^*)$ gerade der *Unterraum*

$$\{d \in \mathbb{R}^n : \nabla g_i(x^*)^\top d = 0 \text{ für alle } i \in \mathcal{A}(x^*),\ \nabla h_j(x^*)^\top d = 0 \text{ für alle } j = 1, \ldots, l\}.$$

§ 7 Behandlung von Box-Beschränkungen

Einfache Ungleichungen der Form

$$\ell \leq x \leq u, \tag{7.1}$$

mit $\ell \in (\mathbb{R} \cup \{-\infty\})^n$ und $u \in (\mathbb{R} \cup \{+\infty\})^n$ heißen **Box-Beschränkungen**, *box constraints* oder *simple bounds*. Sie werden hier gesondert aufgeführt, da sie algorithmisch keine sehr große Herausforderung darstellen.

Wir gehen zunächst davon aus, dass (7.1) die einzigen Beschränkungen in der Aufgabe (NLP) sind. Die Vorgehensweisen, die wir vorstellen, lassen sich später auch mit Algorithmen für kompliziertere Aufgabenstellungen kombinieren.

Die Lagrange-Funktion der Aufgabe lautet hier

$$\mathcal{L}(x, \mu^+, \mu^-) = f(x) + (\mu^+)^\top (x - u) + (\mu^-)^\top (\ell - x).$$

Wir halten außerdem fest, dass die notwendigen Bedingungen 1. Ordnung (KKT-Bedingungen) hier folgende Gestalt haben:[3]

$$\nabla_x \mathcal{L}(x, \mu^+, \mu^-) = \nabla f(x) + \mu^+ - \mu^- = 0, \tag{7.2a}$$

$$\mu^+ \geq 0, \quad x - u \leq 0, \quad (\mu^+)^\top (x - u) = 0, \tag{7.2b}$$

$$\mu^- \geq 0, \quad \ell - x \leq 0, \quad (\mu^-)^\top (\ell - x) = 0. \tag{7.2c}$$

Daraus folgt leicht, dass die KKT-Bedingungen genau dann erfüllt sind, wenn für alle Komponenten $i = 1, \ldots, n$ gilt:

$$\begin{aligned} x_i = u_i \quad &\Rightarrow \quad [\nabla f(x)]_i \leq 0 \\ x_i = \ell_i \quad &\Rightarrow \quad [\nabla f(x)]_i \geq 0 \\ u_i < x_i < \ell_i \quad &\Rightarrow \quad [\nabla f(x)]_i = 0. \end{aligned}$$

Da sich diese Bedingungen aber für einen Test auf die „ungefähre“ Erfüllung der KKT-Bedingungen nicht eignen, suchen wir nach einer anderen Möglichkeit, (7.2) ein Residuum zuzuordnen, dessen Größe man messen kann.

Oft fasst man die Multiplikatoren $\mu^\pm$ zu einem Multiplikator $\mu := \mu^+ - \mu^-$ zusammen, der dann vorzeichenbehaftet ist. Die Notation μ^+ und μ^- können wir beibehalten, diese Größen entsprechen dann gerade dem positiven bzw. dem negativen Teil von μ. Wie man leicht durch Fallunterscheidung sieht, können wir die Komplementaritätsbedingungen (7.2b) und (7.2c) gemeinsam in Form einer (nicht-glatten) Gleichung schreiben. Insgesamt ist (7.2) dann äquivalent zu

$$\nabla f(x) + \mu = 0 \tag{7.3a}$$

$$\mu - \max\{0, \mu + c\,(x - u)\} - \min\{0, \mu + c\,(\ell - x)\} = 0, \tag{7.3b}$$

sodass sich ein zugehöriges Residuum definieren lässt. Die Konstante $c > 0$ ist hier beliebig. Man kann nachrechnen, dass dieses Vorgehen implizit bedeutet, dass wir im Raum der Optimierungsvariablen das durch $M = c\,I$ definierte Skalarprodukt verwenden.

Für Verfahren, die nur mit der primalen Variable x umgehen, ist weiterhin nur das Residuum der reduzierten Gleichung

$$\nabla f(x) + \max\{0, -\nabla f(x) + c\,(x - u)\} + \min\{0, -\nabla f(x) + c\,(\ell - x)\} = 0 \tag{7.4}$$

relevant.

§ 7.1 Verfahren mit Projektion und Proximalpunktverfahren

Das Gradientenverfahren (Algorithmus 5.23) lässt sich auf die Aufgabenstellung

$$\begin{aligned} \text{Minimiere} \quad & f(x), \quad x \in \mathbb{R}^n \\ \text{unter} \quad & \ell \leq x \leq u \end{aligned}$$

verallgemeinern, indem man die Liniensuche auf die Funktion

$$\varphi(\alpha) := f\big(\mathrm{proj}_{[\ell,u]}(x_k + \alpha\, d_k)\big)$$

anwendet und dann

$$x_{k+1} := \mathrm{proj}_{[\ell,u]}(x_k + \alpha\, d_k)$$

[3] Eine *constraint qualification* ist wegen der Linearität der Nebenbedingungen für diese Aufgabe automatisch erfüllt.

setzt. Dabei ist $d_k = -M^{-1}\nabla f(x_k)$ die aktuelle Gradientenrichtung und $\text{proj}_{[\ell,u]}$ die koordinatenweise Projektion auf das zulässige Intervall $[\ell_i, u_i]$, also

$$\text{proj}_{[\ell,u]}(v) := \max\{\min\{v, u\}, \ell\}.$$

Man spricht dann von einem **projizierten Gradientenverfahren**.

Bemerkung 7.1 (zum projizierten Gradientenverfahren)**.**

(a) Die Liniensuchfunktion φ hat jetzt nur noch stückweise die Differenzierbarkeitseigenschaften von f und ist ansonsten nur stetig. Die Armijo-Bedingung (5.12) wird ggf. angepasst.[4]

(b) Weitere Anpassungen sind gegenüber Algorithmus 5.23 bei der Auswertung der Abbruchbedingung erforderlich, da nicht mehr $\|\nabla f(x)\|_{M^{-1}}$ relevant ist, sondern die $\|\cdot\|_{M^{-1}}$-Norm des durch die linke Seite von (7.4) definierten Residuums.

(c) Leider kann die Verwendung beliebiger Vorkonditionierer M dazu führen, dass das Verfahren stagniert, siehe (Kelley, 1999, Chapter 5.5.1). Diagonale (spd) Vorkonditionierer sind aber zulässig, siehe Bertsekas (1982).

Projizierte Newton-Verfahren sind in der Literatur ebenfalls beschrieben, siehe Bertsekas (1982). Auch bei diesen kann aber nicht einfach die Update-Vorschrift

$$x_{k+1} := \text{proj}_{[\ell,u]}(x_k + \alpha\, d_k)$$

mit der Newton-Richtung $d_k = -\nabla^2 f(x_k)^{-1}\nabla f(x_k)$ verwendet werden, sondern die Hessematrix muss geeignet modifiziert werden.

In den letzten Jahren sind **proximale Gradienten-/Newton-Verfahren** ins Interesse gerückt. Bei diesen wird der Schritt von x_k zu x_{k+1} wie folgt ausgeführt:

$$\begin{aligned} \text{Minimiere} \quad & \big(x_k - M^{-1}\nabla f(x_k)\big)^\top M\, z + \frac{1}{2} z^\top M\, z, \quad z \in \mathbb{R}^n \\ \text{unter} \quad & \ell \le z \le u. \end{aligned}$$

Es sei z_k die (eindeutige) Lösung dieser Aufgabe.[5] Anschließend wird

$$x_{k+1} := x_k + \alpha\,(z_k - x_k)$$

gesetzt, wobei α aus einer Armijo-Liniensuche kommt, die erfolgreich ist, sobald

$$f\big(x_k + \alpha\,(z_k - x_k)\big) \le f(x_k) + \sigma\,\alpha\, f'(x_k)\,(z_k - x_k)$$

gilt und $x_k + \alpha\,(z_k - x_k)$ die Beschränkungen erfüllt. Der Vorkonditionierer M kann nun beliebig (spd) sein. Ersetzt man M durch die Hessematrix $\nabla^2 f(x_k)$ oder durch eine Quasi-Newton-Approximation, so erhält man ein proximales Newton- bzw. Quasi-Newton-Verfahren.

Bemerkung 7.2 (zu Trust-Region-Verfahren mit Box-Beschränkungen)**.** Auch Trust-Region-Verfahren lassen sich mit Box-Beschränkungen kombinieren, siehe etwa (Conn et al., 2000, Kapitel 7.8).

[4] So in (Kelley, 1999, eq.(5.13)), nicht jedoch in (Bertsekas, 1982, eq.(7)).

[5] Kurz schreibt man: $z_k = \text{prox}_M(x_k - M^{-1}\nabla f(x_k))$.

§ 7.2 Primal-duale Aktive-Mengen-Strategie

Eine in den letzten Jahren stark popularisierte Methoden, die sogenannte **primal-duale Aktive-Mengen-Strategie**, ergibt sich aus der Formulierung (7.3) der KKT-Bedingungen. Es sei dazu

$$\begin{aligned}\mathcal{A}_k^+ &:= \{1 \le i \le n : \mu_k + c\,(x_k - u) \ge 0\}\\ \mathcal{A}_k^- &:= \{1 \le i \le n : \mu_k + c\,(\ell - x_k) \le 0\}\\ \mathcal{I}_k &:= \{1 \le i \le n\} \setminus \left(\mathcal{A}_k^+ \cup \mathcal{A}_k^-\right)\end{aligned}$$

eine Schätzung der aktiven und inaktiven Mengen[6] in Iteration k. Dann löst man die i. W. unrestringierte Aufgabe

$$\begin{aligned}\text{Minimiere} \quad & f(x), \quad x \in \mathbb{R}^n\\ \text{unter} \quad & [x]_i = u_i \text{ für } i \in \mathcal{A}_k^+\\ \text{sowie} \quad & [x]_i = \ell_i \text{ für } i \in \mathcal{A}_k^-,\end{aligned}$$

deren Lösung dann die nächste Iterierte x_{k+1} wird, und setzt $\mu_{k+1} := -\nabla f(x_{k+1})$.

§ 8 Straftermverfahren

Literatur: (Ulbrich and Ulbrich, 2012, Kapitel 18), (Geiger and Kanzow, 2002, Kapitel 5.2)

Wir kommen nun zu verschiedenen algorithmischen Ansätzen zur Behandlung allgemeinerer Nebenbedingungen in Form von Gleichungen und/oder Ungleichungen.

Die Idee eines Straftermverfahrens (*penalty method*) ist es, einige oder alle Nebenbedingungen in (**NLP**) durch einen Strafterm zu ersetzen. Wir betrachten dazu eine Straffunktion $\pi : \mathbb{R}^n \to \mathbb{R}$ mit den Eigenschaften $\pi(x) = 0$ für $x \in X$ (d. h. x ist zulässig) und $\pi(x) > 0$ für $x \notin X$. Die zu lösenden Teilprobleme sind dann (falls man *alle* Nebenbedingungen penalisiert[7])

$$\text{Minimiere} \quad f(x) + \gamma\,\pi(x) \tag{8.1}$$

mit Strafparameter $\gamma > 0$.

Je größer γ ist, desto stärker wird die Nichtzulässigkeit $x \notin X$ bestraft. Typischerweise sind die Teilprobleme (8.1) dann aber auch schwerer zu lösen.

Algorithmus 8.1 (allgemeines Straftermverfahren)**.**
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: anfänglicher Strafparameter $\gamma_0 > 0$
Eingabe: Lösungsverfahren für (8.1)
Ausgabe: näherungsweiser KKT-Punkt der Aufgabe (**NLP**)
1: Setze $k := 0$
2: Setze $\bar{x}_0 := x_0$
3: **while** Abbruchkriterium nicht erfüllt **do**
4: Finde ein Minimum x_{k+1} von (8.1) mit $\gamma = \gamma_k$

[6] Die Beteiligung primaler und dualer Variablen an diesem Schritt verleiht der Methode ihren Namen.

[7] Behält man einige Nebenbedingungen explizit bei, z. B. $\ell \le x \le u$, so werden diese der Aufgabe (8.1) einfach hinzugefügt.

5: Wähle einen neuen Strafparameter $\gamma_{k+1} > \gamma_k$
6: Setze $k := k + 1$
7: **end while**
8: **return** x_k

Bemerkung 8.2 (zu Straftermverfahren)**.**

(a) Die Iterierten $\{x_k\}$ sind im Allgemeinen unzulässig.

(b) Für einen Konvergenzbeweis benötigt man, dass x_k jeweils ein globales Minimum der penalisierten Aufgabe (8.1) mit $\gamma = \gamma_k$ ist. Lokale Minimierer können auch gegen unzulässige Punkte konvergieren.

Ein typischer Strafterm ist die **quadratische Penalty-Funktion**

$$\pi_2(x) := \frac{1}{2}\sum_{i=1}^{m}\big([g_i(x)]^+\big)^2 + \frac{1}{2}\sum_{j=1}^{p} h_j(x)^2 = \|[g(x)]^+\|_2^2 + \|h(x)\|_2^2,$$

wobei $[g_i(x)]^+ = \max\{0, g_i(x)\}$ ist. Die Funktion π_2 ist stetig differenzierbar, falls g und h stetig differenzierbar sind. Man erhält

$$\nabla\pi_2(x) = \sum_{i=1}^{m}[g_i(x)]^+\,\nabla g_i(x) + \sum_{j=1}^{p} h_j(x)\,\nabla h_j(x).$$

Im Allgemeinen ist er durch das $[\cdot]^+$ aber nicht zweimal stetig differenzierbar.

Alternativ wird oft auch die ℓ_1**-Penalty-Funktion**

$$\pi_1(x) = \sum_{i=1}^{m}[g_i(x)]^+ + \sum_{j=1}^{p}|h_j(x)| = \|[g(x)]^+\|_1 + \|h(x)\|_1$$

verwendet. Diese ist zwar nicht differenzierbar[8], dafür aber exakt, d. h., dass ein lokales Minimum der Ersatzaufgabe (8.1) auch ein lokales Minimum der Originalaufgabe (**NLP**) ist, falls nur der Strafparameter γ hinreichend groß ist. Insbesondere sind also die Nebenbedingungen dann exakt erfüllt.

Barriere-Methoden

Relativ ähnlich zu den Straftermverfahren sind die **Barriere-Methoden**. Dabei wird zu der Zielfunktion eine Barriere-Funktion addiert, die verhindert, dass der zulässige Bereich verlassen wird. Eine wichtige Barriere-Funktion ist der Logarithmus.

[8] Man kann zeigen, dass exakte Penalty-Funktionen generisch nicht-differenzierbar sind. Die Aufgabe (8.1) mit der ℓ_1-Penalty-Funktion lässt sich aufgrund der speziellen Struktur der 1-Norm zwar glatt reformulieren:

$$\begin{aligned} \text{Minimiere} \quad & f(x) + \gamma\Big\{\sum_{i=1}^{m} t_i + \sum_{j=1}^{p}(r_j + s_j)\Big\}, \quad x \in \mathbb{R}^n, t \in \mathbb{R}^m, r, s \in \mathbb{R}^p \\ \text{unter} \quad & g(x) \le t, \quad h(x) = r - s \\ \text{und} \quad & r, s, t \ge 0. \end{aligned}$$

Jedoch hat man dann natürlich i. W. wieder dieselben Beschränkungen wie vor der Penalisierung im Problem.

Man minimiert dann die Hilfsfunktion

$$f(x) - \alpha \sum_{i=1}^{m} \ln(-g_i(x))$$

und lässt α gegen 0 laufen.

Gleichungsnebenbedingungen können so aber nicht behandelt werden.

Ende 19. V

§ 9 Augmentierte-Lagrange-Verfahren

Analog zu den Straftermverfahren wird auch bei den Augmentierten-Lagrange-Verfahren[9] eine Folge von unrestringierten Hilfsproblemen gelöst.

§ 9.1 Der gleichungsrestringierte Fall

Wir betrachten zuerst den gleichungsrestringierten Fall ohne Ungleichungen, also $m = 0$.

Addiert („augmentiert“) man nun zu der Lagrange-Funktion $\mathcal{L}$ noch den quadratischen Penaltyterm,

$$\mathcal{L}^a_\gamma(x, \lambda) := \mathcal{L}(x, \lambda) + \gamma\, \pi_2(x) = f(x) + \lambda^\top h(x) + \frac{\gamma}{2} \|h(x)\|^2,$$

dann erhalt man das folgende, bemerkenswerte Resultat:

Satz 9.1. Es sei (x^*, λ^*) ein KKT-Punkt, in dem **(SSC)** gilt (siehe Satz 6.19). Dann gibt es ein $\gamma_0 > 0$, sodass für alle $\gamma \geq \gamma_0$ der Punkt x^* ein strikter lokaler Minimierer von $\mathcal{L}^a_\gamma(\cdot, \lambda^*)$ ist und sogar die hinreichenden Bedingungen zweiter Ordnung erfüllt.

In einem Algorithmus ersetzt man nun den unbekannten Multiplikator λ^* durch eine Approximation λ und minimiert die Hilfsfunktion

$$\mathcal{L}^a_\gamma(x, \lambda) = f(x) + \lambda^\top h(x) + \frac{\gamma}{2} \|h(x)\|^2$$

bezüglich x. Die Approximationen der Multiplikatoren spielen hier also eine aktive Rolle und werden nicht nur (wie beim Straftermverfahren) als Ausgabe des Algorithmus betrachtet. Man spricht daher von einer „primal-dualen“ Methode, und wir werden viel bessere Eigenschaften als für das quadratische Straftermverfahren erhalten. Dieses minimiert ja gerade

$$f(x) + \frac{\gamma}{2} \|h(x)\|^2 = \mathcal{L}^a_\gamma(x, 0),$$

verwendet also implizit $\lambda = 0$.

[9] auch: *Augmented Lagrangian Methods* (ALM), *methods of multipliers*, *multiplier penalty method*

§ 9.2 Der Fall mit Ungleichungsnebenbedingungen

Nachdem wir die Konstruktion der augmentierten Lagrange-Funktion nun für den gleichungsrestringierten Fall kennengelernt haben, wollen wir nun auch Ungleichungen berücksichtigen. Durch das Einführen von Schlupfvariablen ist (**NLP**) zu dem Problem

$$\begin{aligned} \text{Minimiere} \quad & f(x), \quad x \in \mathbb{R}^n, s \in \mathbb{R}^m \\ \text{sodass} \quad & g_i(x) + s_i = 0, \quad i = 1, \dots, m \\ \text{sowie} \quad & h_j(x) = 0 \quad , \quad j = 1, \dots, p \\ \text{und} \quad & s_i \geq 0 \qquad\quad i = 1, \dots, m \end{aligned}$$

äquivalent. Stellen wir für dieses Problem (ohne Berücksichtigung der Nebenbedingungen an s) die augmentierte Lagrange-Funktion auf, so erhalten wir

$$f(x) + \mu^\top (g(x) + s) + \frac{\gamma}{2} \|g(x) + s\|^2 + \lambda^\top h(x) + \frac{\gamma}{2} \|h(x)\|^2. \tag{9.1}$$

Diese Funktion müssen wir nun bezüglich $x \in \mathbb{R}^n$ und $s \in \mathbb{R}^m$, $s \geq 0$, minimieren. Durch die spezielle Gestalt der Funktion können wir die Minimierung bezüglich der s_i separat durchführen. Für jedes i müssen wir also

$$\text{Minimiere} \quad \mu_i (g_i(x) + s_i) + \frac{\gamma}{2} (g_i(x) + s_i)^2 \quad \text{bezüglich } s_i \geq 0$$

lösen. Man erhält die Lösung

$$s_i^* = \max\Big\{-\frac{\mu_i}{\gamma} - g_i(x), 0\Big\}$$

und den Optimalwert

$$\begin{aligned} & \mu_i (g_i(x) + s_i^*) + \frac{\gamma}{2} (g_i(x) + s_i^*)^2 \\ & \quad = \mu_i \max\Big\{-\frac{\mu_i}{\gamma}, g_i(x)\Big\} + \frac{\gamma}{2} \max\Big\{-\frac{\mu_i}{\gamma}, g_i(x)\Big\}^2 \\ & \quad = \mu_i \left(\max\Big\{0, \frac{\mu_i}{\gamma} + g_i(x)\Big\} - \frac{\mu_i}{\gamma}\right) + \frac{\gamma}{2} \left(\max\Big\{0, \frac{\mu_i}{\gamma} + g_i(x)\Big\} - \frac{\mu_i}{\gamma}\right)^2 \\ & \quad = \frac{\gamma}{2} \max\Big\{0, \frac{\mu_i}{\gamma} + g_i(x)\Big\}^2 - \frac{\mu_i^2}{2\gamma} \\ & \quad = \frac{1}{2\gamma} \Big(\max\big\{0, \mu_i + \gamma\, g_i(x)\big\}^2 - \mu_i^2\Big). \end{aligned}$$

Setzt man dies nun in (9.1) ein, so erhält man die augmentierte Lagrange-Funktion für den allgemeinen Fall der Aufgabe (**NLP**)

$$\begin{aligned} & \mathcal{L}_\gamma^a(x, \mu, \lambda) \\ & \quad = f(x) + \frac{1}{2\gamma} \sum_{i=1}^m \Big(\max\big\{0, \mu_i + \gamma\, g_i(x)\big\}^2 - \mu_i^2\Big) + \lambda^\top h(x) + \frac{\gamma}{2} \|h(x)\|^2 \\ & \quad = f(x) + \frac{1}{2\gamma} \Big(\|[\mu + \gamma\, g(x)]^+\|^2 - \|\mu\|^2\Big) + \frac{1}{2\gamma} \Big(\|\lambda + \gamma\, h(x)\|^2 - \|\lambda\|^2\Big). \end{aligned} \tag{9.2}$$

Wie im gleichungsrestringierten Fall erhalten wir eine exakte Minimierungseigenschaft:

Satz 9.2. Es sei (x^*, μ^*, λ^*) ein strikt komplementärer KKT-Punkt, in dem **(SSC)** gilt. Dann gibt es ein $\gamma_0 > 0$, sodass für alle $\gamma \geq \gamma_0$ der Punkt x^* ein strikter lokaler Minimierer von $\mathcal{L}^a_\gamma(\cdot, \mu^*, \lambda^*)$ ist und sogar die hinreichenden Bedingungen zweiter Ordnung erfüllt.

Ein Beweis findet sich z. B. in (Fernández and Solodov, 2012, Proposition 3.1).

§ 9.3 Das Augmentierte-Lagrange-Verfahren

Wir wollen nun die augmentierte Lagrange-Funktion verwenden, um einen Algorithmus zur Lösung von **(NLP)** zu erhalten.

Dazu sei zunächst eine Schätzung der Multiplikatoren (μ_k, λ_k) gegeben. In jedem Schritt wird nun die Aufgabe

$$\text{Minimiere} \quad \mathcal{L}^a_{\gamma_k}(x, \mu_k, \lambda_k), \quad x \in \mathbb{R}^n \tag{9.3}$$

gelöst. Man erhält dann ein x_{k+1} mit

$$\begin{aligned} 0 &= \nabla_x \mathcal{L}^a_{\gamma_k}(x_{k+1}, \mu_k, \lambda_k) \\ &= \nabla f(x_{k+1}) + \sum_{i=1}^{m} \max\{0, \mu_{k,i} + \gamma_k\, g_i(x_{k+1})\}\, \nabla g_i(x_{k+1}) \\ &\qquad + \sum_{j=1}^{p} \big(\lambda_{k,j} + \gamma_k\, h_j(x_{k+1})\big)\, \nabla h_j(x_{k+1}). \end{aligned}$$

Diese Darstellung legt es nahe,

$$\mu_{k+1} := \max\{0, \mu_k + \gamma_k\, g(x_{k+1})\} \quad \text{und} \quad \lambda_{k+1} := \lambda_k + \gamma_k\, h(x_{k+1}) \tag{9.4}$$

als nächste Approximation der Multiplikatoren zu verwenden. Dies liefert direkt (mit der Original-Lagrange-Funktion)

$$0 = \nabla_x \mathcal{L}(x_{k+1}, \mu_{k+1}, \lambda_{k+1}),$$

jedoch ist natürlich i. A. die Zulässigkeit von x_{k+1} noch nicht gegeben.

Bemerkung 9.3 (zu Augmentierten Lagrange-Verfahren)**.**

(a) Im Gegensatz zum quadratischen Straftermverfahren muss man nicht $\gamma \to \infty$ laufen lassen.

(b) Der Mehraufwand gegenüber dem Penalty-Verfahren ist äußerst gering.

Algorithmus 9.4 (allgemeines Augmentiertes Lagrange-Verfahren[10])**.**
Eingabe: Startwert $x_0 \in \mathbb{R}^n$
Eingabe: Startwert $\mu_0 \in \mathbb{R}^m$ und $\lambda_0 \in \mathbb{R}^p$
Eingabe: anfänglicher Strafparameter $\gamma_0 > 0$
Eingabe: Lösungsverfahren für (8.1)
Ausgabe: näherungsweiser KKT-Punkt der Aufgabe **(NLP)**
1: Setze $k := 0$
2: **while** Abbruchkriterium nicht erfüllt **do**
3: Finde, ausgehend vom Startwert x_k, ein Minimum x_{k+1} von (8.1)
4: Setze $\mu_{k+1} := \max\{0, \mu_k + \gamma_k\, g(x_{k+1})\}$

Setze $\lambda_{k+1} := \lambda_k + \gamma_k\, h(x_{k+1})$
Wähle einen neuen Strafparameter γ_{k+1}
Setze $k := k + 1$
end while
return x_k, μ_k und λ_k

§ 10 SQP-Verfahren

Der Name dieser Verfahrensklasse, *sequential quadratic programming* (SQP) lautet sich aus der Tatsache ab, dass eine Folge von *quadratic programs* (QPs) gelöst wird, das sind Aufgaben mit quadratischer Zielfunktion und linearen Gleichungs-/Ungleichungsnebenbedingungen.

§ 10.1 Das lokale SQP-Verfahren für gleichungsrestringierte Probleme

Wir betrachten zunächst das nichtlineare, gleichungsrestringierte Problem

$$\begin{aligned} \text{Minimiere} \quad & f(x) \qquad \text{über } x \in \mathbb{R}^n \\ \text{sodass} \quad & h(x) = 0. \end{aligned} \tag{10.1}$$

Gilt in einer Lösung x^* nun eine CQ, dann gibt es einen Multiplikator $\lambda^* \in \mathbb{R}^p$, sodass die KKT-Bedingungen

$$\begin{aligned} \nabla_x \mathcal{L}(x^*, \lambda^*) &= 0, \\ h(x^*) &= 0 \end{aligned}$$

erfüllt sind. Diese Bedingungen schreiben wir als

$$\Phi(x, \lambda) := \begin{pmatrix} \nabla_x \mathcal{L}(x, \lambda) \\ h(x) \end{pmatrix} = 0$$

mit $\Phi : \mathbb{R}^{n+p} \to \mathbb{R}^{n+p}$. Dieses System wollen wir nun mit dem Newton-Verfahren lösen. Sind f und die h_j zweimal stetig differenzierbar, dann ist Φ stetig differenzierbar, und es gilt

$$\Phi'(x, \lambda) = \begin{pmatrix} \nabla^2_{xx} \mathcal{L}(x, \lambda) & \nabla^2_{x\lambda} \mathcal{L}(x, \lambda) \\ h'(x) & 0 \end{pmatrix} = \begin{pmatrix} \nabla^2_{xx} \mathcal{L}(x, \lambda) & h'(x)^\top \\ h'(x) & 0 \end{pmatrix}.$$

Es gilt folgendes Kriterium für die Invertierbarkeit von $\Phi'(x^*, \lambda^*)$:

Lemma 10.1. Hat die Jacobimatrix $h'(x^*)$ vollen Zeilenrang und ist $\nabla^2_{xx} \mathcal{L}(x^*, \lambda^*)$ positiv definit auf $\ker h'(x^*)$, dann ist $\Phi'(x^*, \lambda^*)$ invertierbar.

Beachte: Die Bedingungen sind gerade äquivalent dazu, dass LICQ und (**SSC**) in x^* erfüllt sind.

[10] ähnlich (Geiger and Kanzow, 2002, Algorithmus 5.21), vgl. auch (Nocedal and Wright, 2006, Algorithm 17.4)

In diesem Fall erhalten wir die lokale q-superlineare Konvergenz[11] des lokalen Newton-Verfahrens zum Lösen der Gleichung

$$\Phi(x^*, \lambda^*) = 0.$$

Das entstehende Verfahren heißt **Lagrange-Newton-Verfahren**.

Wir wollen nun die Newtongleichung $\Phi'(x_k, \lambda_k)\,(\delta x_k, \delta\lambda_k) = -\Phi(x_k, \lambda_k)$, also

$$\begin{pmatrix} \nabla^2_{xx}\mathcal{L}(x_k, \lambda_k) & h'(x_k)^\top \\ h'(x_k) & 0 \end{pmatrix} \begin{pmatrix} \delta x_k \\ \delta\lambda_k \end{pmatrix} = \begin{pmatrix} -\nabla_x \mathcal{L}(x_k, \lambda_k) \\ -h(x_k) \end{pmatrix} = \begin{pmatrix} -\nabla f(x_k) - h'(x_k)^\top \lambda_k \\ -h(x_k) \end{pmatrix} \tag{10.2}$$

als KKT-Bedingungen eines QPs deuten.

Zur Abkürzung setzen wir $H_k = \nabla^2_{xx}\mathcal{L}(x_k, \lambda_k)$ und betrachten das QP

$$\begin{aligned} \text{Minimiere} \quad & \nabla f(x_k)^\top d + \frac{1}{2}\, d^\top H_k\, d, \\ \text{sodass} \quad & h(x_k) + h'(x_k)\, d = 0. \end{aligned} \tag{10.3}$$

Da die Nebenbedingungen affin sind, existieren in jeder Lösung d_k^{qp} Multiplikatoren λ_k^{qp}, sodass

$$\nabla f(x_k) + H_k\, d_k^{\text{qp}} + h'(x_k)^\top \lambda_k^{\text{qp}} = 0, \quad h(x_k) + h'(x_k)\, d_k^{\text{qp}} = 0. \tag{10.4}$$

Der Vergleich dieser Bedingungen mit (10.2) liefert sofort:

Lemma 10.2. Es sei $(x_k, \lambda_k) \in \mathbb{R}^{n+p}$ beliebig. Für ein $(\delta x_k, \delta\lambda_k)$ sind äquivalent:

- $(\delta x_k, \delta\lambda_k)$ erfüllt (10.2)
- $(d_k^{\text{qp}}, \lambda_k^{\text{qp}}) = (\delta x_k, \lambda_k + \delta\lambda_k)$ ist ein KKT-Punkt des QPs (10.3).

Schließlich können wir an der Lösung von (10.3) erkennen, ob x_k bereits eine Lösung von (10.1) ist.

Lemma 10.3. Es sei $(x_k, \lambda_k) \in \mathbb{R}^{n+p}$ beliebig. Dann sind äquivalent:

(a) (x_k, λ_k) ist ein KKT-Punkt von (10.1) in dem (**SSC**) erfüllt ist,

(b) $d_k^{\text{qp}} = 0$ ist der eindeutige globale Minimierer von (10.3), und $\lambda_k^{\text{qp}} = \lambda_k$ ist ein dazugehöriger Multiplikator,

(c) $d_k^{\text{qp}} = 0$ ist ein striktes lokales Minimum von (10.3), und $\lambda_k^{\text{qp}} = \lambda_k$ ist ein dazugehöriger Multiplikator.

§ 10.2 Das lokale SQP-Verfahren für Aufgaben mit Gleichungs- und Ungleichungsnebenbedingungen

Wir betrachten nun den allgemeinen Fall der Aufgabe (**NLP**), wobei alle Funktionen zweimal stetig differenzierbar sind. Haben wir nun eine Iterierte (x_k, μ_k, λ_k), dann

[11] Sind die zweiten Ableitungen von f und h_j sogar Lipschitz-stetig, dann folgt die Lipschitz-stetigkeit von Φ', und somit konvergiert das Verfahren lokal q-quadratisch.

betrachten wir analog zu (10.3) das QP

$$\begin{aligned} \text{Minimiere} \quad & \nabla f(x_k)^\top d + \frac{1}{2} d^\top H_k d, \\ \text{sodass} \quad & g(x_k) + g'(x_k) d \leq 0, \\ \text{und} \quad & h(x_k) + h'(x_k) d = 0. \end{aligned} \tag{10.5}$$

Hier ist wieder $H_k = \nabla^2_{xx} \mathcal{L}(x_k, \mu_k, \lambda_k)$. Zuerst betrachten wir ein Analogon zu Lemma 10.3.

Lemma 10.4. Es sei $(x_k, \mu_k, \lambda_k) \in \mathbb{R}^{n+m+p}$ beliebig. Dann sind äquivalent:

(a) (x_k, μ_k, λ_k) ist ein KKT-Punkt von **(NLP)**, in dem **(SSC)** erfüllt ist,

(b) $d_k^{\text{qp}} = 0$ ist ein striktes lokales Minimum von (10.5), und $(\mu_k^{\text{qp}}, \lambda_k^{\text{qp}}) = (\mu_k, \lambda_k)$ sind dazugehörige Multiplikatoren.

Anders als in Lemma 10.3 erhält man keine Äquivalenz zur *globalen* Optimalität.

§ 10.3 Globalisiertes SQP-Verfahren

Um die Konvergenz auch für Startpunkte zu sichern, die weit von einer Lösung entfernt liegen, benötigt man eine Globalisierung. Dafür gibt es (wie im unbeschränkten Fall) mehrere Möglichkeiten, wir betrachten nur die Globalisierung mit einer Liniensuche.

Wir betrachten nun noch einmal das zu lösende QP, aber ohne den Iterationsindex:

$$\begin{aligned} \text{Minimiere} \quad & \nabla f(x)^\top d + \frac{1}{2} d^\top H d, \\ \text{sodass} \quad & g(x) + g'(x) d \leq 0, \\ \text{und} \quad & h(x) + h'(x) d = 0. \end{aligned} \tag{10.6}$$

Da der Schritt d sowohl die Zielfunktion verkleinern, als auch die Zulässigkeit verbessern/erhalten soll, kann die Liniensuche nicht nur mit der Zielfunktion arbeiten. Typischerweise verwendet man die ℓ^1-**Merit-Funktion**

$$\phi(x) = f(x) + \gamma\, \pi_1(x) = f(x) + \gamma \sum_{i=1}^{m} [g_i(x)]^+ + \gamma \sum_{j=1}^{p} |h_j(x)| \tag{10.7}$$

und führt eine Armijo-Liniensuche durch. Die Funktionen π_1 und (somit auch) ϕ sind allerdings nicht überall differenzierbar, es existieren aber für alle $x, d \in \mathbb{R}^n$ die Richtungsableitungen

$$\psi'(x; d) = \lim_{t \searrow 0} \frac{\psi(x + t\, d) - \psi(x)}{t}, \quad \text{für } \psi \in \{\pi_1, \phi\}.$$

Man erhält[12]

$$\begin{aligned} \pi_1'(x; d) = & \sum_{i: g_i(x) < 0} 0 + \sum_{i: g_i(x) = 0} \left[\nabla g_i(x)^\top d\right]^+ + \sum_{i: g_i(x) > 0} \nabla g_i(x)^\top d \\ & + \sum_{j: h_j(x) < 0} (-\nabla h_j(x)^\top d) + \sum_{j: h_j(x) = 0} \left|\nabla h_j(x)^\top d\right| + \sum_{j: h_j(x) > 0} \nabla h_j(x)^\top d. \end{aligned}$$

[12] Durch Ausrechnen, siehe (Ulbrich and Ulbrich, 2012, Satz 19.10), oder mit einer Kettenregel, siehe (Geiger and Kanzow, 2002, Korollar 5.34).

Unter gewissen Voraussetzungen liefert ein KKT-Punkt von (10.6) eine Abstiegsrichtung für ϕ. In diesem Fall kann eine Armijo-Liniensuche (Algorithmus 5.13) durchgeführt werden, die Armijo-Bedingung (5.12) wird dabei durch

$$\phi(x + \alpha\, d^{\mathrm{qp}}) \leq \phi(x) + \sigma\, \alpha\, \phi'(x; d^{\mathrm{qp}}) \tag{10.8}$$

ersetzt.

Ende 20. V

Literaturverzeichnis

Alpargu, G. (1996). The Kantorovich Inequality, with some Extensions and with some Statistical Applications. Master thesis, Department of Mathematics and Statistics, McGill University, Montreal, Canada.

Alt, W. (2002). *Nichtlineare Optimierung*. Vieweg Studium: Aufbaukurs Mathematik. [Vieweg Studies: Mathematics Course]. Friedr. Vieweg & Sohn, Braunschweig, doi: 10.1007/978-3-322-84904-5, eine Einführung in Theorie, Verfahren und Anwendungen. [An introduction to theory, procedures and applications].

Anderson, T. W. (1971). *The statistical analysis of time series*. John Wiley & Sons, Inc., New York-London-Sydney.

Barzilai, J. and Borwein, J. M. (1988). Two-point step size gradient methods. *IMA Journal of Numerical Analysis* 8: 141–148, doi: 10.1093/imanum/8.1.141.

Bertsekas, D. (1982). Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization* 20: 221–246, doi: 10.1137/0320018.

Conn, A., Gould, N. and Toint, P. (2000). *Trust-Region Methods*. Philadelphia: SIAM.

Dennis, J. E., Jr. and Moré, J. J. (1974). A characterization of superlinear convergence and its application to quasi-Newton methods. *Mathematics of Computation* 28: 549–560.

Elman, H., Silvester, D. and Wathen, A. (2014). *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, 2nd ed.

Fernández, D. and Solodov, M. V. (2012). Local convergence of exact and inexact augmented Lagrangian methods under the second-order sufficient optimality condition. *SIAM Journal on Optimization* 22: 384–407, doi: 10.1137/10081085X.

Fletcher, R. and Leyffer, S. (2002). Nonlinear programming without a penalty function. *Mathematical Programming* 91: 239–269.

Geiger, C. and Kanzow, C. (1999). *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben*. New York: Springer, doi: 10.1007/978-3-642-58582-1.

Geiger, C. and Kanzow, C. (2002). *Theorie und Numerik restringierter Optimierungsaufgaben*. New York: Springer, doi: 10.1007/978-3-642-56004-0.

Gilbert, J. C. and Nocedal, J. (1992). Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on Optimization* 2: 21–42, doi: 10.1137/0802003.

Gonzaga, C. (2016). On the worst case performance of the steepest descent algorithm for quadratic functions. *Mathematical Programming Series A* 160: 307–320, doi: 10.1007/s10107-016-0984-8.

Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards* 49: 409–436 (1953).

Heuser, H. (2002). *Lehrbuch der Analysis - Teil 2*. Stuttgart: B.G.Teubner.

Kelley, C. T. (1999). *Iterative Methods for Optimization*, Frontiers in Applied Mathematics *18*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).

Kjeldsen, T. H. (2000). A contextualized historical analysis of the Kuhn-Tucker theorem in nonlinear programming: the impact of World War II. *Historia Mathematica* 27: 331–361, doi: 10.1006/hmat.2000.2289.

Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*. Berkeley and Los Angeles: University of California Press, 481–492.

Martínez, J. M. (1994). Local minimizers of quadratic functions on Euclidean balls and spheres. *SIAM Journal on Optimization* 4: 159–176, doi: 10.1137/0804009.

Nocedal, J., Sartenaer, A. and Zhu, C. (2002). On the behavior of the gradient norm in the steepest descent method. *Computational Optimization and Applications. An International Journal* 22: 5–35, doi: 10.1023/A:1014897230089.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. New York: Springer, 2nd ed., doi: 10.1007/978-0-387-40065-5.

Steihaug, T. (1983). The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis* 20: 626–637.

Toint, P. (1981). Towards an Efficient Sparsity Exploiting Newton Method for Minimization. In Duff, I. (ed.), *Sparse Matrices and Their Uses*. London: Academic Press, 57–88, based on the Proceedings of the IMA Numerical Analysis Group Conference, organised by the Institute of Mathematics and Its Applications and held at the University of Reading, 9th–11th July, 1980.

Ulbrich, M. and Ulbrich, S. (2012). *Nichtlineare Optimierung*. New York: Springer, doi: 10.1007/978-3-0346-0654-7.