

Mathematik für Bioinformatik und Systembiologie

- Kapitel Einführung in die Optimierung -

Roland Herzog und Dirk Lebiedz

WS 2009/10

Universität Freiburg

Dieses Vorlesungsskript ist auf der Basis von Vorlesungen zu Grundlagen der Optimierung und ihrer LaTeX-Ausarbeitung von Prof. Dr. Roland Herzog an der TU Chemnitz und PD Dr. Dirk Lebiedz an der Universität Freiburg zusammengestellt. Der Inhalt orientiert sich in weiten Teilen an den Darstellungen in den Literaturempfehlungen: Geiger und Kanzow: “Numerische Verfahren zur Lösung unrestringierten Optimierungsaufgaben”, Springer (1999), Geiger und Kanzow: “Theorie und Numerik restringierten Optimierungsaufgaben”, Springer (2002) und Nocedal und Wright: “Numerical Optimization”, 2nd editions, Springer (2006).

Fehler und Kommentare bitte an: dirk.lebiedz@biologie.uni-freiburg.de

Stand: 18. Dezember 2009

Inhaltsverzeichnis

Kapitel 0.	Einführung	5
1	Grundbegriffe	5
Kapitel 1.	Unbeschränkte Optimierung	9
3	Optimalitätsbedingungen	9
4	Ein allgemeines Abstiegsverfahren	9
5	Eine effiziente Schrittweitenstrategie: Armijo-Regel	11
6	Wahl der Suchrichtungen: Gradientenverfahren	13
6.1	Konvergenz bei quadratischer Zielfunktion	13
7	Wahl der Suchrichtungen: Newton-Verfahren	15
Kapitel 2.	Nichtlineare beschränkte Optimierung	17
8	Einführung	17
9	Notwendige Optimalitätsbedingungen	19
9.1	Optimalitätsbedingungen unter LICQ	20
10	Numerische Methoden für beschränkte nichtlineare Optimierungsprobleme	20

KAPITEL 0

Einführung

§ 1 Grundbegriffe

Allgemeines beschränktes Optimierungsproblem

$$\left. \begin{array}{ll} \text{Minimiere} & f(x) \quad \text{über } x \in \Omega \quad (\text{Zielfunktion}) \\ \text{sodass} & h_i(x) = 0, \quad i \in \mathcal{E} \quad (\text{Gleichungsnebenbedingungen}) \\ \text{und} & g_i(x) \leq 0, \quad i \in \mathcal{I} \quad (\text{Ungleichungsnebenbedingungen}) \end{array} \right\} \quad (1.1)$$

Ω heißt **Grundmenge** und x die **Variable(n)** der Aufgabe. Sei

- $\Omega \subset \mathbb{R}^n$
- $f, g_i, h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ hinreichend glatte Funktionen (2-mal stetig diff.bar)
- $\mathcal{E}$ und $\mathcal{I}$ endliche (ggf. leere) Indexmengen.

Definition 1.1

Die Menge

$$X = \{x \in \Omega : h_i(x) = 0 \text{ für alle } i \in \mathcal{E}, \quad g_i(x) \leq 0 \text{ für alle } i \in \mathcal{I}\}$$

heißt **zulässige Menge**, $x \in X$ **zulässiger Punkt**.

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

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

- (b) Ein globales Optimum heißt **strikt**, wenn gilt:

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

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

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

- (d) Ein lokales Optimum heißt **strikt**, wenn gilt:

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

Ein Optimum nennt man auch **Minimum**, **Minimalstelle** oder **Lösung** von (1.1).

Fragestellungen:

- Wann existieren Optimallösungen ?
- Wie überprüft man Optimalität ? ($\leadsto$ Optimalitätsbedingungen)
- Wie kann man Optimallösungen näherungsweise numerisch bestimmen ?
- Wie konstruiert man effiziente Algorithmen?

Beispiel 1.2 (Angebotsauswertung)

Ein Pharma-Unternehmen will eine bestimmte Menge M einer Basischemikalie einkaufen und holt dazu Angebote von n Lieferfirmen ein, von denen keine die gewünschte Gesamtmenge alleine liefern kann. Anbieter i liefert maximal m_i , wobei der Preis $f_i(x_i)$ von der Bestellmenge x_i abhängt. [f_i wird i.d.R. monoton wachsend sein und evtl. nichtlinear.] Die minimalen Beschaffungskosten ergeben sich durch Lösung der folgenden Optimierungsaufgabe:

$$\left. \begin{array}{ll} \min & f(x) := \sum_{i=1}^n f_i(x_i), \quad x \in \mathbb{R}^n \\ \text{sodass} & \sum_{i=1}^n x_i = M \\ \text{und} & 0 \leq x_i \leq m_i, \quad i = 1, \dots, n. \end{array} \right\} \quad (1.2)$$

Beispiel 1.3 (Parameteridentifizierungsproblem)

Eine an einer Feder befestigte Masse bewegt sich unter dem Einfluss von Dämpfung gemäß der Differentialgleichung

$$m\ddot{y} + r\dot{y} + ky = 0.$$

In einem Experiment mit Anfangsanregung

$$y(0) = y_0, \quad \dot{y}(0) = 0$$

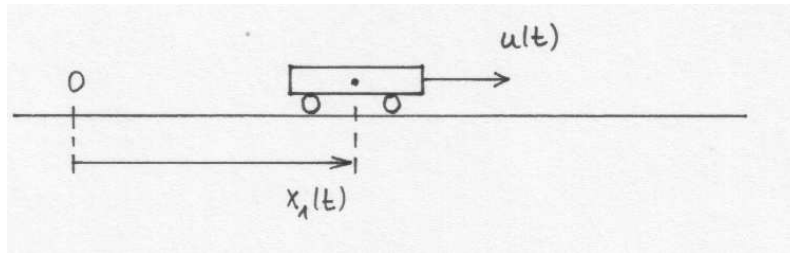
soll zu verschiedenen Zeitpunkten t_i , $i = 1, \dots, N$ die Auslenkung y_i gemessen werden. Daraus sollen bei bekannter Masse m die unbekannten Dämpfungs- und Federkonstanten (Parameter) r und k bestimmt werden.

Wir setzen $x := (r, k)^\top$. Die Lösung der Differentialgleichung zum Zeitpunkt t mit obigen Anfangsbedingungen wird mit $y(t; x)$ bezeichnet, um die Abhängigkeit von den zu bestimmenden Parametern x zu verdeutlichen. Wir formulieren folgende Aufgabe:

$$\min \quad f(x) := \frac{1}{2} \sum_{i=1}^N |y(t_i; x) - y_i|^2, \quad x \in \mathbb{R}^2, \quad (1.3)$$

damit die Messwerte y_i möglichst gut die Vorhersage der Position durch das Modell $y(t_i; x)$ wiedergeben.

Motivation für diese „Methode der kleinsten Quadrate“: Sind die Messfehler in den Daten y_i unabhängig und identisch verteilt und folgen sie einer Normalverteilung, so erhält man durch Minimierung der Fehlerquadratsumme gerade den wahrscheinlichsten Parameterwert x .

Beispiel 1.4 (Optimalsteuerungsproblem)

Problemstellung: Der Wagen soll möglichst schnell zum Nullpunkt gesteuert werden und dort stehenbleiben. Es seien

$x_1(t)$: Position des Wagens (Masse = 1) zur Zeit t

$x_2(t)$: Geschwindigkeit des Wagens

$u(t)$: Beschleunigung des Wagens, **Steuerfunktion**

$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$: **Zustand** des Systems.

Die Dynamik des Systems ist nach dem zweiten Newton'schen Axiom (Kraft = Masse $\times$ Beschleunigung) gegeben durch

$$\left. \begin{array}{ll} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t) \\ x(0) = x_0 \in \mathbb{R}^2 & \text{Anfangsbedingung} \\ x(T) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \text{Endbedingung} \\ u(t) \in [-1, 1] & \text{Steuerbeschränkung} \end{array} \right\} \quad (1.4)$$

Ziel: Bestimme eine stückweise stetige optimale Steuerung $u : [0, T] \rightarrow \mathbb{R}$ so, dass die Endzeit T minimal wird unter den Beschränkungen (1.4).

Optimalsteuerungsprobleme sind unendlich-dimensionale (infinite) Optimierungsprobleme, da die Optimierungsvariablen (Steuer-)Funktionen sind, also Elemente eines unendlich-dimensionalen Vektorraums. Man kann sie mit Hilfe sog. direkter numerischer Methoden behandeln, indem man die Steuerfunktionen parametrisiert (z.B. durch Polynome approximiert) und dadurch auf einen endlich-dimensionalen Raum mit endlich vielen Optimierungsvariablen projiziert (siehe Übung).

Optimalsteuerungsprobleme aus Biologie und Medizin werden in der Vorlesung **Einführung in die Systembiologie** vorgestellt. Dabei geht es z.B. darum, ein biologisches System durch externe Manipulation in einen gewünschten Zustand zu steuern oder medizinische Behandlungsstrategien bzgl. eines gewünschten Ziels zu optimieren. Optimierung ist oft auch im Zusammenhang mit biotechnologischen Prozessen von Bedeutung, in denen man z.B. die Synthesausbeute eines Bioprodukts maximieren möchte.

Definition 1.5 (Klassifikation von Optimierungsproblemen)

- (a) Die Optimierungsaufgabe (1.1) heißt **frei** oder **unrestringiert**, wenn $\mathcal{E} = \mathcal{I} = \emptyset$ ist, andernfalls **gleichungs-** und/oder **ungleichungs-restringiert** oder **-beschränkt**.

- (b) Ungleichungsbeschränkungen der Art

$$l_i \leq x_i \leq u_i, \quad i = 1, \dots, n$$

mit $l_i \in \mathbb{R} \cup \{-\infty\}$ und $u_i \in \mathbb{R} \cup \{\infty\}$ heißen **Box-Beschränkungen**.

- (c) Sind f , g und h (affin-)lineare Funktionen von x , so sprechen wir von **linearer Optimierung**. Ein lineares Optimierungsproblem heißt auch **lineares Programm (LP)**, also z.B.

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

- (d) Sind allgemeiner f und alle g_i konvexe Funktionen und sind alle h_i wieder (affin-)linear, so sprechen wir von **konvexer Optimierung**. Hierbei darf außerdem noch $\Omega \subset \mathbb{R}^n$ eine konvexe Teilmenge sein.
- (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 (QP)**.
- (f) Im allgemeinen Fall spricht man von **nichtlinearer Optimierung** und von einem **nichtlinearen Programm (NLP)**.

KAPITEL 1

Unbeschränkte Optimierung

Wir betrachten in diesem Kapitel das unrestringierte Optimierungsproblem (1.1) mit $\Omega = \mathbb{R}^n$ und $\mathcal{E} = \mathcal{I} = \emptyset$, also

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

Wir beschränken uns auf das Auffinden lokaler Optima. Im gesamten Kapitel 1 sei $f : \mathbb{R}^n \rightarrow \mathbb{R}$ mindestens einmal stetig differenzierbar.

§ 3 Optimalitätsbedingungen

Satz 3.1 (Notwendige Bedingungen 1. Ordnung)

Sei x^* ein lokales Minimum und f stetig differenzierbar in einer Umgebung $U(x^*)$. Dann ist $\nabla f(x^*) = 0$.

Ist f stetig differenzierbar in einer Umgebung $U(x)$ und gilt $\nabla f(x) = 0$, so nennt man x einen **stationären Punkt** von f .

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

Satz 3.2 (Hinreichende Bedingungen 2. Ordnung)

Es sei f zweimal stetig differenzierbar in einer Umgebung $U(x^*)$, und es gelte

- (a) $\nabla f(x^*) = 0$
- (b) $\nabla^2 f(x^*)$ ist positiv definit (hinreichende Bedingung dafür ist, dass alle Eigenwerte der Hesse-Matrix $\nabla^2 f(x^*)$ positiv sind).

Dann ist x^* ein striktes lokales Minimum von f .

§ 4 Ein allgemeines Abstiegsverfahren

In den § 4-6 vernachlässigen wir zunächst die Optimalitätskriterien aus § 3 und orientieren uns nur an den Funktionswerten von f .

Definition 4.1 (Abstiegsrichtung)

Ein Vektor $d \in \mathbb{R}^n$ heißt **Abstiegsrichtung** von f im Punkt $x \in \mathbb{R}^n$, wenn es ein $\alpha > 0$ gibt mit

$$f(x + td) < f(x) \quad \text{für alle } t \in (0, \alpha). \quad \diamond$$

Idee: Suche entlang einer Abstiegsrichtung eine geeignete Schrittweite t , die f hinreichend verkleinert, und iteriere.

Lemma 4.2

Es seien $x \in \mathbb{R}^n$ und $d \in \mathbb{R}^n$ gegeben mit $\nabla f(x)^\top d < 0$. Dann ist d eine Abstiegsrichtung von f in x .

Beweis: Für die Richtungsableitung gilt

$$\delta f(x; d) = \lim_{t \searrow 0} \frac{f(x + td) - f(x)}{t} = \nabla f(x)^\top d < 0.$$

Also gibt es ein $\alpha > 0$ mit

$$\frac{f(x + td) - f(x)}{t} < 0 \quad \text{für alle } t \in (0, \alpha).$$

□

Anschaulich bedeutet $\nabla f(x)^\top d < 0$, dass der Winkel zwischen der Richtung d und dem negativen Gradienten $-\nabla f(x)$ kleiner als 90° ist.

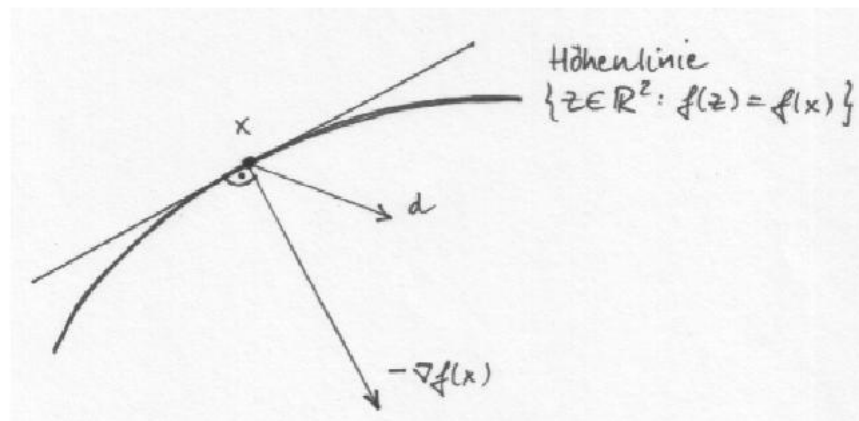


ABBILDUNG 4.1. Winkelbedingung beim Abstiegsverfahren

Beachte: Die Bedingung $\nabla f(x)^\top d < 0$ ist nicht notwendig für eine Abstiegsrichtung. Etwa in einem strikten lokalen Maximum x sind alle Richtungen $d \neq 0$ Abstiegsrichtungen, jedoch erfüllt wegen $\nabla f(x) = 0$ keine Richtung d die Bedingung $\nabla f(x)^\top d < 0$.

Bemerkung 4.3

Ist x kein stationärer Punkt von f , so ist $d = -\nabla f(x)$ immer eine Abstiegsrichtung von f in x .

Algorithmus 4.4 (Allgemeines Abstiegsverfahren)

- 1: Wähle $x^0 \in \mathbb{R}^n$ und setze $k := 0$
- 2: **while** Abbruchkriterium nicht erfüllt **do**
- 3: Bestimme eine Abstiegsrichtung d^k von f in x^k
- 4: Bestimme eine Schrittweite $t_k > 0$ mit $f(x^k + t_k d^k) < f(x^k)$
- 5: Setze $x^{k+1} := x^k + t_k d^k$ und $k := k + 1$
- 6: **end while**

Der Algorithmus lässt viele Freiheiten bei der Wahl der Abstiegsrichtungen d^k und der Schrittweiten t_k . Wir lassen zunächst das Abbruchkriterium außer Acht und wollen stets annehmen, dass der Algorithmus eine unendliche Folge $\{x^k\}$ liefert.

Definition 4.5 (Schrittweitenstrategie)

- (a) Eine Abbildung T auf $\mathbb{R}^n \times \mathbb{R}^n$, die jedem Paar (x, d) eine Teilmenge von $\mathbb{R}^+$ zuordnet, heißt **Schrittweitenstrategie**.
- (b) Sie heißt **wohldefiniert**, wenn $T(x, d) \neq \emptyset$ ist für alle (x, d) mit $\nabla f(x)^\top d < 0$.
- (c) Sie heißt **effizient**, wenn es für alle (x, d) mit d Abstiegsrichtung von f in x eine von x und d unabhängige Konstante $\theta > 0$ gibt mit

$$f(x + td) \leq f(x) - \theta \left(\frac{\nabla f(x)^\top d}{\|d\|} \right)^2 \quad (4.1)$$

für alle $t \in T(x, d)$.

Wir nennen eine Schrittweite $t \in T(x, d)$ selbst **effizient**, wenn T effizient ist. Ein Beispiel einer effizienten Schrittweitenstrategie werden wir in § 5 behandeln. Da jeweils f entlang der Richtung d^k „durchsucht“ wird, spricht man auch von Algorithmen zur **Linienuche**.

Satz 4.6 (Ein globaler Konvergenzsatz)

Es seien $\{x^k\}$, $\{d^k\}$ und $\{t_k\}$ durch Algorithmus 4.4 erzeugte Folgen mit folgenden Eigenschaften:

- (a) Die Schrittweiten $t_k > 0$ sind effizient.
- (b) Die Suchrichtungen d^k erfüllen die sogenannte **Winkelbedingung**: Es existiert eine Konstante $c > 0$ mit

$$-\nabla f(x^k)^\top d^k \geq c \|\nabla f(x^k)\| \|d^k\|. \quad (4.2)$$

Dann ist jeder Häufungspunkt x^* von $\{x^k\}$ ein stationärer Punkt von f , erfüllt also $\nabla f(x^*) = 0$.

Bemerkung 4.7

Für den Winkel φ_k zwischen der Suchrichtung d^k und dem negativen Gradienten $-\nabla f(x^k)$ gilt

$$\cos \varphi_k = - \frac{\nabla f(x^k)^\top d^k}{\|\nabla f(x^k)\| \|d^k\|}.$$

Die Winkelbedingung (4.2) besagt also, dass der Winkel zwischen d^k und $-\nabla f(x^k)$ gleichmäßig von 90° weg beschränkt bleibt. $\diamond$

§ 5 Eine effiziente Schrittweitenstrategie: Armijo-Regel

Es stellt sich die Frage, wie man eine wohldefinierte und effiziente Schrittweitenstrategie im Sinne von Definition 4.5 algorithmisch umsetzen kann. Die naheliegende **Minimierungsregel**

$$\text{„Bestimme } t^k := t_{\min}^k \text{ so, dass } f(x^k + t_{\min}^k d^k) = \min_{t>0} f(x^k + td^k) \text{ gilt“} \quad (5.1)$$

ist zwar unter bestimmten Annahmen wohldefiniert und effizient, aber wegen ihres Aufwands außer in Sonderfällen für f nicht praktikabel.

Wir behandeln hier deshalb eine andere realisierbare Schrittweitenstrategie: Die **Armijo-Regel** ist eine Bedingung, die einen hinreichenden Abstieg von f sichert. Sie ist bestimmt durch die Parameter $\sigma \in (0, 1)$ und $\beta \in (0, 1)$ und die Startschrittweite $s > 0$. Zu $(x, d) \in \mathbb{R}^n \times \mathbb{R}^n$ mit $\nabla f(x)^\top d < 0$ bestimme man

$$t_A := \max\{s\beta^l : l = 0, 1, 2, \dots\},$$

sodass

$$f(x + t_A d) \leq f(x) + \sigma t_A \nabla f(x)^\top d \quad (5.2)$$

gilt.

Man testet also die Schrittweiten $s, s\beta, s\beta^2$ etc., bis zum ersten Mal (5.2) erfüllt ist, und liefert dann $T(x, d) = \{t_A\}$ zurück („**Backtracking**“).

Zur Veranschaulichung führen wir die Liniensuchfunktion

$$\varphi(t) = f(x + td) \quad (5.3)$$

ein. Es gilt: φ ist stetig differenzierbar in $[0, \infty)$ und

$$\varphi'(t) = \nabla f(x + td)^\top d.$$

Also lautet die Armijo-Bedingung (5.2) alternativ

$$\varphi(t_A) \leq \varphi(0) + \sigma t_A \varphi'(0). \quad (5.4)$$

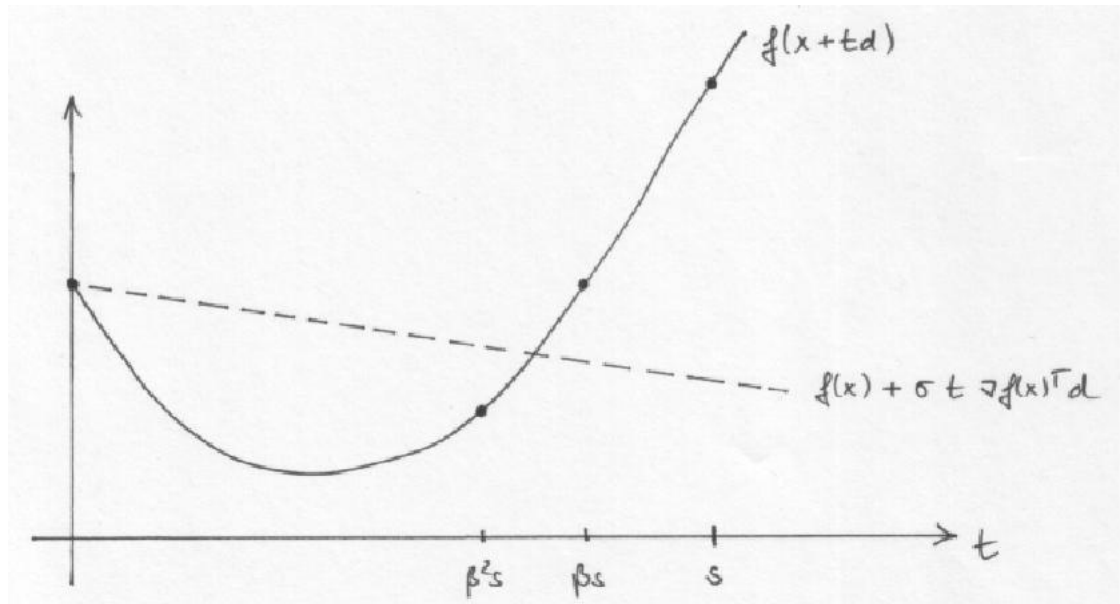


ABBILDUNG 5.1. Darstellung der Armijo-Bedingung (5.2)

Satz 5.1 (Wohldefiniertheit und Effizienz der Armijo-Regel)

Es seien $\sigma \in (0, 1)$, $\beta \in (0, 1)$ sowie $s > 0$ gegeben. Man kann unter sehr allgemeinen Bedingungen beweisen, dass die Armijo-Regel wohldefiniert und effizient ist.

§ 6 Wahl der Suchrichtungen: Gradientenverfahren

Bisher haben wir uns im Zusammenhang mit dem allgemeinen Abstiegsverfahren (Algorithmus 4.4) nur mit der Schrittweitenstrategie beschäftigt. In § 6 und § 7 folgen nun zwei Strategien, um die Suchrichtungen $\{d^k\}$ zu bestimmen.

Hier wählen wir zunächst

$$d = -\nabla f(x),$$

die Richtung des steilsten Abstiegs von f .

Algorithmus 6.1 (Gradientenverfahren, Verfahren des steilsten Abstiegs)

- 1: Wähle $x^0 \in \mathbb{R}^n$, $\sigma \in (0, 1)$, $\beta \in (0, 1)$, $\varepsilon > 0$ und setze $k := 0$
 - 2: **while** $\|\nabla f(x^k)\| > \varepsilon$ **do**
 - 3: Setze $d^k := -\nabla f(x^k)$
 - 4: Bestimme eine Schrittweite t_k mit der Armijo-Regel mit der Startschrittweite $s = 1$, d.h., $t_k := \max\{\beta^l : l = 0, 1, 2, \dots\}$, sodass gilt:

$$f(x^k + t_k d^k) \leq f(x^k) + \sigma t_k \nabla f(x^k)^\top d^k$$
 - 5: Setze $x^{k+1} := x^k + t_k d^k$ und $k := k + 1$
 - 6: **end while**
-

§ 6.1 Konvergenz bei quadratischer Zielfunktion

Um die Konvergenzgeschwindigkeit des Gradientenverfahrens zu untersuchen, wenden wir es auf eine quadratische Funktion f an:

$$f(x) = \frac{1}{2} x^\top Q x + c^\top x + \gamma$$

mit $Q \in \mathbb{R}^n \times \mathbb{R}^n$ symmetrisch positiv definit (s.p.d.), $c \in \mathbb{R}^n$ und $\gamma \in \mathbb{R}$. Die eindeutige (globale) Lösung der Aufgabe $\min_{x \in \mathbb{R}^n} f(x)$ ist

$$x^* = -Q^{-1}c, \tag{6.1}$$

denn dies ist die einzige Lösung der notwendigen Bedingungen (Satz 3.1), und die hinreichenden Bedingungen (Satz 3.2) sind dort erfüllt. Natürlich wird man das Gradientenverfahren nur anwenden, wenn die direkte Berechnung von x^* aus (6.1) zu aufwendig ist [also Q zu groß ist oder nicht explizit vorliegt].

Im Fall der quadratischen Zielfunktion lässt sich die exakte Schrittweite aus der Minierungsregel (5.1)

$$t_{min}^k = \arg \min_{t \geq 0} f(x^k + t d^k)$$

im k -ten Schritt berechnen. Setzen wir

$$g^k = \nabla f(x^k) = Q x^k + c,$$

so ergibt sich

$$t_k := t_{min}^k = \frac{(g^k)^\top g^k}{(g^k)^\top Q g^k}.$$

Wir wählen also hier statt der Armijo-Regel in Algorithmus 6.1 stets die exakte Schrittweite t_{min}^k .

Es seien $\lambda_{min}(Q)$ und $\lambda_{max}(Q) > 0$ der kleinste und größte Eigenwert der s.p.d. Matrix Q und

$$\kappa = \text{cond}_2(Q) = \frac{\lambda_{max}(Q)}{\lambda_{min}(Q)}$$

die (spektrale) Konditionszahl von Q .

Satz 6.2 (Globaler Konvergenzsatz für quadratische Zielfunktionen)

Es sei Q s.p.d. Das Gradientenverfahren mit exakter Schrittweite t_{min} konvergiert für jeden Startvektor $x^0 \in \mathbb{R}^n$ gegen das eindeutige globale Minimum x^* , und es gilt

$$f(x^{k+1}) - f(x^*) \leq \left(\frac{\kappa - 1}{\kappa + 1} \right)^2 (f(x^k) - f(x^*))$$

$$\|x^k - x^*\| \leq \sqrt{\kappa} \left(\frac{\kappa - 1}{\kappa + 1} \right)^k \|x^0 - x^*\|$$

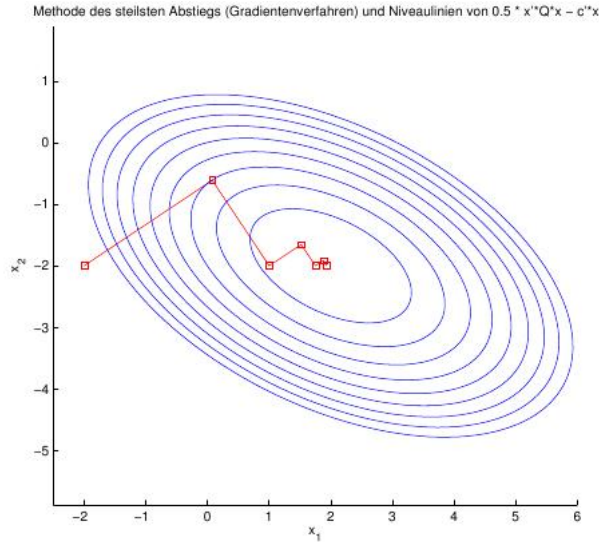


ABBILDUNG 6.1. Konvergenz des Gradientenverfahrens

Bemerkung 6.3

Für große Konditionszahlen κ ist die Konvergenz sehr langsam (Zick-Zack-Effekt).

Zur Charakterisierung der Konvergenzgeschwindigkeit von Algorithmen führen wir ein:

Definition 6.4 (Q-Konvergenzraten)

Sei $\{x^n\} \subset \mathbb{R}^n$ eine Folge und $x^* \in \mathbb{R}^n$.

- (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) Gilt $x^k \rightarrow x^*$, so konvergiert $\{x^k\}$ gegen x^* (mindestens) **q-quadratisch**, falls ein $C > 0$ existiert mit

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

§ 7 Wahl der Suchrichtungen: Newton-Verfahren

[Hier orientieren wir uns jetzt wieder an den notwendigen Bedingungen!]

In diesem Abschnitt wird f als zweimal stetig differenzierbar angenommen. Wir betrachten wieder

$$\min_{x \in \mathbb{R}^n} f(x). \quad (7.1)$$

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

- (a) Die notwendige Optimalitätsbedingung 1. Ordnung lautet

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

siehe Satz 3.1. Wende zur Lösung dieser i.A. nichtlinearen Gleichung das Newton-Verfahren an. Wir erhalten die Iterationsvorschrift

$$x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k).$$

- (b) Im aktuellen Iterationspunkt x^k ersetze man (7.1) durch die Minimierung des quadratischen Modells

$$q_k(x) = f(x^k) + \nabla f(x^k)^\top (x - x^k) + \frac{1}{2} (x - x^k)^\top \nabla^2 f(x^k) (x - x^k). \quad (7.2)$$

Ist die Hessematrix $\nabla^2 f(x^k)$ positiv definit, so ist das eindeutige Minimum durch

$$0 = \nabla q_k(x) = \nabla f(x^k) + \nabla^2 f(x^k)(x - x^k)$$

charakterisiert. Wir wählen die Lösung dieser Aufgabe als nächste Iterierte x^{k+1} und erhalten wieder

$$x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k).$$

Das Newton-Verfahren ist von der Bauart des allgemeinen Abstiegsverfahrens (Algorithmus 4.4), wobei die Suchrichtung durch die Lösung des linearen Gleichungssystems

$$\nabla^2 f(x^k) d^k = -\nabla f(x^k)$$

und die feste Schrittweite durch $t_k = 1$ gegeben sind:

Algorithmus 7.1 (Lokales Newton-Verfahren)

- 1: Wähle $x^0 \in \mathbb{R}^n$ und $\varepsilon > 0$ und setze $k := 0$
 - 2: **while** $\|\nabla f(x^k)\| > \varepsilon$ **do**
 - 3: Löse $\nabla^2 f(x^k) d^k := -\nabla f(x^k)$
 - 4: Setze $x^{k+1} := x^k + d^k$ und $k := k + 1$
 - 5: **end while**
-

Wir können nun einen lokalen Konvergenzsatz formulieren:

Satz 7.2 (Lokaler Konvergenzsatz für Newton-Verfahren)

Es sei $x^* \in \mathbb{R}^n$ ein stationärer Punkt und $\nabla^2 f(x^*)$ regulär. Dann existiert eine Umgebung $U(x^*)$ von x^* , sodass für jedes $x^0 \in U(x^*)$ gilt:

- (a) Das lokale Newton-Verfahren ist wohldefiniert und erzeugt eine Folge $\{x^k\}$, die gegen x^* konvergiert.
- (b) Die Konvergenzrate ist q-superlinear.
- (c) Ist $\nabla^2 f$ stetig diff.bar in $U(x^*)$, so ist die Konvergenzrate q-quadratisch.

Beachte: Das vereinfachte Newton-Verfahren, bei dem statt $\nabla f(x^k)$ die feste Matrix $\nabla^2 f(x^0)$ bei der Schrittberechnung verwendet wird, konvergiert linear.

Bemerkung 7.3

Koppelt man das Newton-Verfahren an eine effiziente Schrittweitenstrategie, kann man globale Konvergenz erreichen.

KAPITEL 2

Nichtlineare beschränkte Optimierung

§ 8 Einführung

Wir betrachten in diesem Abschnitt das allgemeine nichtlineare Optimierungsproblem (vgl. (1.1)), auch NLP (Nonlinear Program) genannt:

$$\left. \begin{array}{ll} \text{Minimiere} & f(x) \quad \text{über } x \in \mathbb{R}^n \\ \text{sodass} & g_i(x) \leq 0, \quad i = 1, \dots, m \\ \text{und} & h_j(x) = 0, \quad j = 1, \dots, p. \end{array} \right\} \quad (8.1)$$

mit $m \geq 0$ und $p \geq 0$.

Ziel: Optimalitätsbedingungen für (8.1), die numerisch überprüfbar sind und als Grundlage für Algorithmen dienen können.

Wir nehmen im gesamten Kapitel 2 an, dass f , g und h stetig differenzierbar auf $\mathbb{R}^n$ sind. Zur Erinnerung:

$$X = \{x \in \mathbb{R}^n : g_i(x) \leq 0 \quad \text{für alle } i = 1, \dots, m, \quad h_j(x) = 0 \quad \text{für alle } j = 1, \dots, p\} \quad (8.2)$$

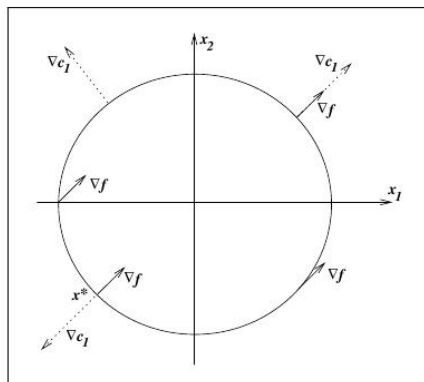
heißt **zulässige Menge** von (8.1).

Beispiel 8.1 (gleichungsbeschränktes NLP)

$$\begin{array}{ll} \min & f(x) = x_1 + x_2 \\ \text{unter} & c_1(x) = x_1^2 + x_2^2 - 2 = 0, \end{array} \quad (8.3)$$

d.h. $X = \{(x_1, x_2) \in \mathbb{R}^2 : x_1^2 + x_2^2 - 2 = 0\}$.

Man berechnet $\nabla f(x) = (1, 1)^T$, $\nabla c_1(x) = (2x_1, 2x_2)^T$, Visualisierung:



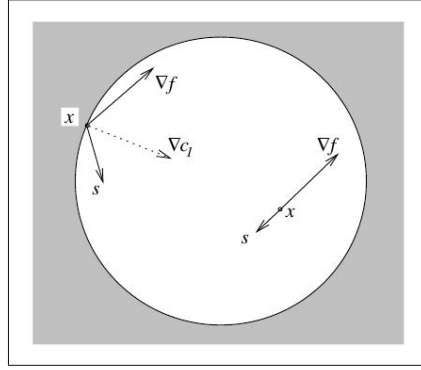
Man sieht hier, dass in einem Punkt x^* immer Abstiegsrichtungen möglich sind, außer wenn ∇f und ∇c_1 parallel sind, d.h. wenn es ein λ^* gibt mit $\nabla f(x^*) = \lambda^* \nabla c_1(x^*)$.

Dies ist also eine notwendige Bedingung für ein lokales Minimum von (8.3).

Beispiel 8.2 (ungleichungsbeschränktes NLP)

$$\begin{aligned} \min f(x) &= x_1 + x_2 \\ \text{unter } c_1(x) &= 2 - (x_1^2 + x_2^2) \geq 0, \end{aligned} \quad (8.4)$$

Man berechnet wieder $\nabla f(x) = (1, 1)^T$, $\nabla c_1(x) = (-2x_1, -2x_2)^T$, Visualisierung:



Eine Abstiegsrichtung s im Punkt x^* muss die Bedingung $\langle \nabla f(x^*), s \rangle = \nabla f(x^*)^T s < 0$ erfüllen. Bzgl. der Zulässigkeit von s gilt in linearer Näherung (Taylor-Entwicklung 1. Ordnung):

$$0 \leq c_1(x^*) + \nabla c_1(x^*)^T s \quad (8.5)$$

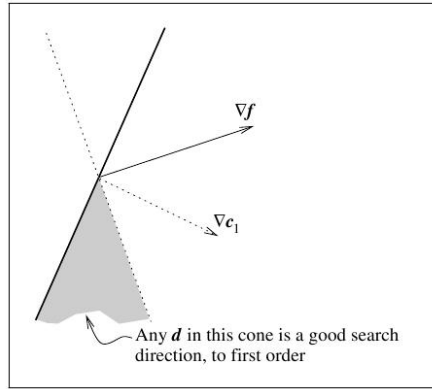
Sei nun x^* eine optimale Lösung von (8.2) und s eine Abstiegsrichtung.
Fallunterscheidung:

- a) x^* liegt im Inneren der zulässigen Menge X , dann lässt sich immer eine Abstiegsrichtung s mit $\nabla f(x^*)^T s < 0$ finden, falls $\nabla f(x^*) \neq 0 \Rightarrow \nabla f(x^*) = 0$ ist notwendige Bedingung für ein lokales Minimum.
- b) x^* liegt am Rand der zulässigen Menge X , dann gilt $c_1(x^*) = 0$ und mit (8.5) folgt

$$\nabla c_1(x^*)^T s \geq 0 \quad (8.6)$$

Zusammen mit $\nabla f(x^*)^T s < 0$ folgt also, dass $\nabla c_1(x^*)$ und $\nabla f(x^*)$ parallel sein und in die gleiche Richtung zeigen müssen, damit keine zulässige Abstiegsrichtung existiert (siehe folgende Abbildung), d.h.

$$\nabla c_1(x^*) = \lambda_1 \nabla f(x^*), \quad \lambda_1 \geq 0 \quad (8.7)$$



Wir führen nun die sog. Lagrange-Funktion ein, die uns erlauben wird, notwendige Bedingungen für beschränkte Optimierungsprobleme in bequemer Form zu schreiben und die obige Fallunterscheidung zusammenzuführen.

Definition 8.3 (Lagrange-Funktion)

Die Funktion

$$\begin{aligned}\mathcal{L}(x, \lambda, \mu) &:= f(x) + \sum_{i=1}^m \mu_i g_i(x) + \sum_{j=1}^p \lambda_j h_j(x) \\ &= f(x) + \mu^\top g(x) + \lambda^\top h(x)\end{aligned}$$

heißt die zu (8.1) gehörige **Lagrange-Funktion**. $\diamond$

Beachte: Man sagt, die Nebenbedingungen $g_i(x) \leq 0$ und $h_j(x) = 0$ werden an die Zielfunktion $f(x)$ „angekoppelt“.

§ 9 Notwendige Optimalitätsbedingungen der nichtlinearen Optimierung

Definition 9.1 (KKT-Bedingungen und -Punkt)

(a) Die Bedingungen

$$\mathcal{L}_x(x, \lambda, \mu) = 0 \quad (9.1a)$$

$$h(x) = 0 \quad (9.1b)$$

$$\mu \geq 0, \quad g(x) \leq 0, \quad \mu^\top g(x) = 0 \quad (9.1c)$$

heißen die **Karush-Kuhn-Tucker-Bedingungen** (kurz: **KKT-Bedingungen**) des Problems (8.1). Dabei ist

$$\begin{aligned}\mathcal{L}_x(x, \lambda, \mu) &= \nabla f(x) + \sum_{i=1}^m \mu_i \nabla g_i(x) + \sum_{j=1}^p \lambda_j \nabla h_j(x) \\ &= \nabla f(x) + \underbrace{g'(x)^\top}_{\text{Jacobi-Matrix}} \mu + h'(x)^\top \lambda\end{aligned}$$

der Gradient von $\mathcal{L}(x, \lambda, \mu)$ bzgl. x .

(b) Ein Punkt $(x^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m$, der (9.1) erfüllt, heißt ein **KKT-Punkt** von (8.1). Die Vektoren λ^* und μ^* heißen dann **Lagrange-Multiplikatoren** zu den Beschränkungen $h(x^*) = 0$ bzw. $g(x^*) \leq 0$. $\diamond$

Bemerkung 9.2

Sei (x^*, λ^*, μ^*) ein KKT-Punkt.

- (a) Die **Komplementaritätsbedingung** (9.1c) ist äquivalent zu

$$\mu_i^* \geq 0, \quad g_i(x^*) \leq 0, \quad \mu_i^* g_i(x^*) = 0 \quad \text{für alle } i = 1, \dots, m,$$

d.h., mindestens eine der Zahlen μ_i^* und $g_i(x^*)$ ist gleich null. Insbesondere gehört also zu einer inaktiven Beschränkung $g_i(x^*) < 0$ ein Lagrange-Multiplikator $\mu_i^* = 0$.

- (b) Die Bedingung (9.1a) besagt gerade, dass

$$-\nabla f(x^*) = \sum_{i=1}^m \mu_i^* \nabla g_i(x^*) + \sum_{j=1}^p \lambda_j^* \nabla h_j(x^*)$$

gilt mit Skalaren $\mu_i^* \geq 0$ und $\mu_i^* = 0$ für $i \in \mathcal{I}(x^*)$ wegen (9.1c). Außerdem sichern (9.1b) und (9.1c) die Zulässigkeit $g(x^*) \leq 0$ und $h(x^*) = 0$.

- (c) Durch die KKT-Bedingungen wird die Optimierungsaufgabe (8.1) in ein nichtlineares Gleichungs- und Ungleichungssystem (9.1) transformiert, das als Basis für numerische Verfahren dienen kann.
- (d) Man nennt die KKT-Bedingungen auch **notwendige Bedingungen 1. Ordnung**, da sie (nur) erste Ableitungen von f , g und h verwenden.

§ 9.1 Optimalitätsbedingungen unter LICQ

Definition 9.3 (LICQ)

Sei x_0 zulässiger Punkt des Optimierungsproblems (8.1) und $\mathcal{A}(x_0)$ die Menge der aktiven Indizes, d.h. die Indizes derjenigen Ungleichungen $g_i(x) \leq 0$, für die in x^* die Gleichheit $g_i(x) = 0$ erfüllt ist. Man sagt: x_0 erfüllt die **Regularitätsbedingungen der linearen Unabhängigkeit** (kurz: **LICQ**, Linear Inequality Constraint Qualification), wenn gilt: Die Gradienten $\{\nabla h_j(x_0)\}_{j=1}^p \cup \{\nabla g_i(x_0)\}_{i \in \mathcal{A}(x_0)}$ sind linear unabhängig. $\diamond$

Satz 9.4 (KKT-Bedingungen unter LICQ)

Es sei x^* ein lokales Optimum von (8.1), das die LICQ erfüllt. Dann existieren *eindeutig bestimmte* Lagrange-Multiplikatoren $\lambda^* \in \mathbb{R}^p$ und $\mu^* \in \mathbb{R}^m$, so dass (x^*, λ^*, μ^*) ein KKT-Punkt von (8.1) ist.

§ 10 Numerische Methoden für beschränkte nichtlineare Optimierungsprobleme

Numerische Methoden setzen die in Kapitel 1 vorgestellten Methoden zur Berechnung lokaler Minima ein, indem sie das nichtlineare Gleichungssystem der KKT-Bedingungen (9.1a - 9.1c) lösen. Das Einführen der Lagrange-Funktion hat also den Sinn, ein beschränktes Optimierungsproblem für eine Zielfunktion in ein unbeschränktes für die Lagrange-Funktion zu überführen, indem die Beschränkungen über Lagrange-Multiplikatoren “angekoppelt” werden.