Kapitel III

Nichtlineare Optimierung ohne Restriktionen

Einleitung

Die im 17. Jahrhundert durch Newton und Leibniz begründete Infinitesimalrechnung liefert das theoretische Fundament der Nichtlinearen Optimierung. Mit entsprechenden analytischen Hilfsmitteln der Analysis lassen sich z.B. Minima von reellen Funktionen $f : \boldsymbol{R}^n \to \boldsymbol{R}$ charakterisieren. Die konkrete Bestimmung der Minima, also die Berechnung von reellen Vektoren, die die charakterisierenden Bedingungen erfüllen, ist in den seltensten Fällen explizit mit theoretischen Mitteln möglich.

In der Einführung zur Mathematischen Optimierung werden nur nichtlineare Optimierungsprobleme ohne explizite Restriktionen behandelt. Bei diesen Aufgaben muß das Minimum einer gegebenen Funktion $f : \boldsymbol{R}^n \to \boldsymbol{R}$ bestimmt werden, d.h. es soll eine Lösung der Aufgabe

$$\min_{x \in \boldsymbol{R}^n} f(x) =: f(\hat{x})$$

gefunden werden. So benutzt man zum Beispiel in der Regressionsrechnung Optimierungsmodelle zur Fixierung unbekannter Parameter.

Regressionsrechnung

Zur mathematischen Formulierung von Gesetzmäßigkeiten, die ein technisches oder physikalisches Phänomen beschreiben, wird häufig eine Hypothese über den möglichen funktionalen Zusammenhang bekannter und beobachtbarer Variablen formuliert, die noch etliche freie Parameter enthält, etwa

$$z = h(x, \lambda).$$

Hierbei ist $x \in \boldsymbol{R}^n$ ein Vektor, der die unabhängigen Variablen eines Experiments enthält, $\lambda \in \boldsymbol{R}^p$ ein Vektor unbekannter Parameter, und z eine Variable, deren Abhängigkeit modelliert werden soll.

Nach Auswertung von m Experimenten kennt man m Paare von Variablenwerten z_i, x_i, die unter Berücksichtigung möglicher Beobachtungsfehler ϵ_i in etwa die funktionale

Abhängigkeit erfüllen sollten, d.h. bei passenden Parametern muß

$$z_i = h(x_i, \lambda) + \epsilon_i$$

für alle $i \in \{1, 2, \ldots, n\}$ gelten. Um möglichst kleine Abweichungen ϵ_i zu erhalten, die man als Beobachtungsfehler deuten kann, versucht man die unbekannten Parameter λ optimal anzupassen, indem man entsprechende Optimierungsaufgaben löst, z.B.

$$\min \left\{ \sum_{i=1}^{m} (z_i - h(x_i, \lambda))^2 \mid \lambda \in \boldsymbol{R}^p \right\}$$

oder

$$\min \left\{ \sum_{i=1}^{m} w_i (z_i - h(x_i, \lambda))^r \mid \lambda \in \boldsymbol{R}^p \right\},$$

wobei $r \geq 1$ gerade, $w > 0$. $w > 0$ ist komponentenweise zu verstehen, wobei 0 den 0-Vektor entsprechender Dimension bezeichnet. Analog benutzen wir diese Konvention auch für andere Vektoren mit identischen Komponenten, insbesondere für den häufig auftretenden 1-Vektor.

1 Minimierung in einer Variablen

Die Lösung nichtlinearer Optimierungsaufgaben im eindimensionalen Fall tritt als Teilproblem in den meisten Verfahren zur Lösung nichtlinearer Optimierungsaufgaben in höheren Dimensionen auf. Zu gegebenem abgeschlossenen Intervall $I \subseteq \boldsymbol{R}$ und gegebener Funktion $f : I \to \boldsymbol{R}$ wird ein $\hat{x} \in I$ mit

$$f(\hat{x}) = \min_{x \in I} f(x) \tag{1.1}$$

gesucht. Um ein solches Minimum mit Hilfe eines Verfahrens effektiv bestimmen zu können, müssen einschränkende Annahmen über die Funktion gemacht werden. Je nach Eigenschaften der Funktion ist dann ein passender Algorithmus zu wählen.

Unimodale Funktionen

Definition (Unimodalität): f heißt *unimodal*, falls für ein $\hat{x} \in I := [a, b]$ gilt:
f streng monoton fallend auf $[a, \hat{x}]$, f streng monoton wachsend auf $[\hat{x}, b]$.

Intervallreduktion

Unimodale Funktionen erlauben nach Auswertung der Funktion an zwei Stellen $x < y$ im Intervall die Reduktion des Intervalls. Unter gewissen Annahmen liefert die rekursive Anwendung dieser Beobachtung einen festgelegten Algorithmus:

1. Im ersten Schritt zwei Auswertungen, in den weiteren Schritten eine weitere Auswertung von f im Restintervall zur Reduktion verwenden,

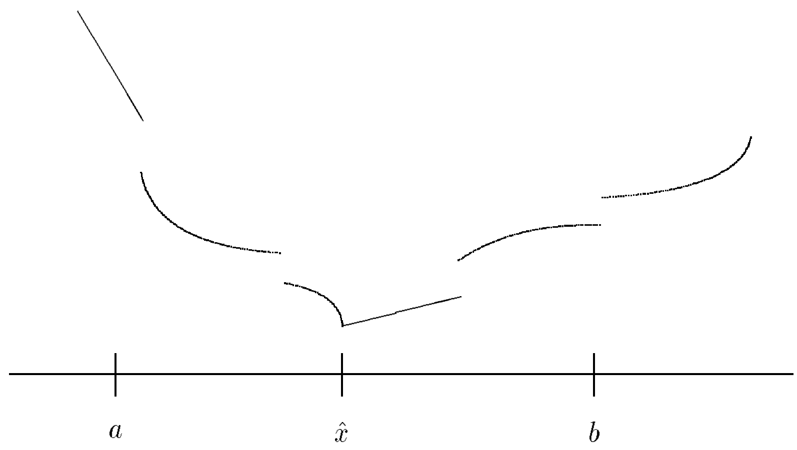

Abbildung III.1: Unimodale Funktionen

2. x und y stets symmetrisch im Restintervall wählen,

3. Reduktionsfaktor σ sei konstant, $\frac{1}{2} < \sigma < 1$.

Durch die symmetrische Wahl von x, y im Restintervall wird der Reduktionsfaktor unabhängig von der Funktion f. Verlangt man einen konstanten Reduktionsfaktor in allen Iterationen, so ist dieser eindeutig bestimmt. Um diesen von f unabhängigen Faktor zu berechnen, genügt es oBdA eine streng monoton wachsende Funktion $f : [0,1] \to \boldsymbol{R}$ zu betrachten. Im ersten Intervall sei oBdA $0 < x < y < 1$, und daher $y = \sigma$, $x = 1 - \sigma$.

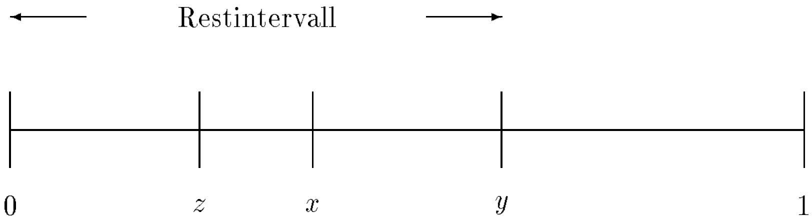

Abbildung III.2: Intervallreduktion bei unimodalen Funktionen

Im ersten Schritt wird das Intervall auf das Restintervall $[0, y]$ reduziert. In diesem Restintervall wird dann z symmetrisch zu x gewählt. Je nach Wahl von y wird $z < x$ oder $x \le z$.

1. Fall: $\sigma < \frac{2}{3}$. Dann gilt $x = 1 - \sigma > \frac{1}{3}$ und daher $z = y - x = \sigma - (1 - \sigma) < \frac{1}{3} < x$. Konstanz der Reduktion bedeutet nun $\frac{x}{y} = \frac{y}{1}$, d.h. $0 = y^2 - x = \sigma^2 + \sigma - 1$. Die Lösung σ dieser quadratischen Gleichung mit $\frac{1}{2} < \sigma < 1$ ist $\sigma = \frac{1}{2}(\sqrt{5} - 1) = 0,618\ldots$.

2. Fall: $\sigma \ge \frac{2}{3}$. Analog zum ersten Fall folgt: $z = \sigma - (1 - \sigma) \ge \frac{1}{3} \ge x$. Also muß zur Konstanz der Reduktion $\frac{z}{y} = \frac{y}{1}$, d.h. $0 = y^2 - z = \sigma^2 - 2\sigma + 1$ gelten. Diese quadratische Gleichung besitzt offenbar keine Lösung σ mit $\frac{1}{2} < \sigma < 1$.

Der im ersten Fall gefundene Reduktionsfaktor $\sigma = 0,618\ldots$ ist also unter den getroffenen Annahmen der einzig mögliche und legt den Algorithmus fest.

Algorithmus 1.1 (Goldener Schnitt)

1. **Initialisierung:**
   $i := 0;\ a_0 := a;\ b_0 := b;$
   $x_i := a + (1-\sigma)(b_i - a_i);\ y_i := a + \sigma(b_i - a_i);$
   $fx := f(x_i);\ fy := f(y_i).$

2. **Iteration:**
   Wenn $fx < fy$, dann

   $$\begin{aligned}&a_{i+1} := a_i;\ b_{i+1} := y_i;\\ &x_{i+1} := a_{i+1} + (1-\sigma)(b_{i+1} - a_{i+1});\ y_{i+1} := x_i;\\ &fy := fx;\ fx := f(x_{i+1});\end{aligned}$$

   sonst

   $$\begin{aligned}&a_{i+1} := x_i;\ b_{i+1} := b_i;\\ &x_{i+1} := y_i;\ y_{i+1} := b_{i+1} - (1-\sigma)(b_{i+1} - a_{i+1});\\ &fx := fy;\ fy := f(y_{i+1});\end{aligned}$$

   $i := i + 1;$

3. **Abbruch:**
   Wenn $\frac{1}{2}(b_i - a_i) > \epsilon$, dann gehe nach 2.
   $\tilde{x} := \frac{1}{2}(a_i + b_i)$; Stop.

Im obigem Algorithmus wird ein Funktionswert zuviel berechnet; in der Praxis kann man die tatsächliche Berechnung des neuen fx bzw. fy zurückstellen, bis der Wert für den Vergleich benötigt wird.

Mögliche Abbruchkriterien

Wie oben beschrieben bricht der Algorithmus ab, sobald das Minimum bis auf eine vorgegebene Genauigkeit ϵ lokalisiert ist. Da nach $n+1$ Funktionsauswertungen die Länge des Restintervalls auf $\sigma^n(b-a)$ reduziert ist, kann man n in Abhängigkeit von ϵ a priori abschätzen.

Ist f auf dem Restintervall eine quadratische Funktion, so kann man den Wert des Minimums auf dem Restintervall nach unten durch

$$f(\hat{x}) \geq f_{\min} - \sigma(f_{\max} - f_{\min})$$

abschätzen. Liegt die Abweichung innerhalb einer vorgegebenen Genauigkeit, etwa wenn $f_{\max} - f_{\min} \leq \delta$ erfüllt ist, so kann man das Verfahren abbrechen.

Wenn man bereit ist, auf die Konstanz der Reduktion zu verzichten, kann man sich fragen, ob es eine Modifikation des Verfahrens gibt, das bei gleicher Anzahl von Funktionsauswertungen ein kleineres Restintervall liefert. Wir wollen mit L_n die maximale Länge eines mittels n Auswertungen in L_n auf $L_1 = 1$ reduzierbaren Intervalls bezeichnen.

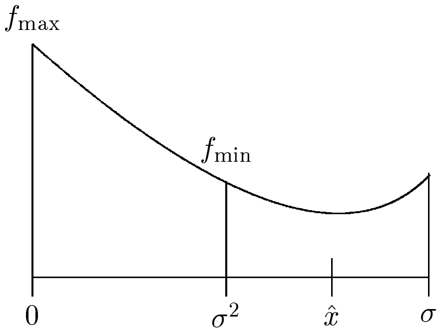

Abbildung III.3: Abbruchkriterien

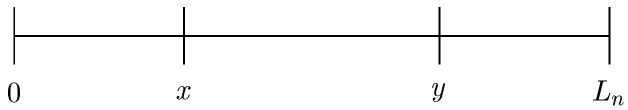

Abbildung III.4: Maximale Intervallreduktion

Durch jede der ersten beiden Stützstellen $x < y$ in L_n, etwa durch x, wird dieses Intervall in zwei Teilintervalle zerlegt. Das linke enthält höchstens $n-2$, das rechte höchstens $n-1$ der n Stützstellen. Entsprechend sind die Längen dieser Teilintervalle höchstens L_{n-2} bzw. L_{n-1}. Also gilt

$$L_n \leq L_{n-1} + L_{n-2}$$

und $L_0 = L_1 = 1$, da mindestens 2 Stützstellen zur Reduktion notwendig sind. Nach dieser Abschätzung erhalten wir die bestmögliche Reduktion, falls eine Lösung die Ungleichungen als Gleichungen erfüllt.

Fibonacci-Zahlen

Die entsprechende Rekursionsformel $F_0 := F_1 := 1, F_n := F_{n-1} + F_{n-2}$ definiert bekanntlich gerade die Fibonacci-Zahlen, deren Eigenschaften man etwa im Buch von Knuth [11] nachlesen kann. Eine Darstellung der Lösung in geschlossener Form ist

$$F_i = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^{i+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{i+1} \right)$$

In Algorithmen erweist sich die Fortsetzung $F_{-2} := 1, F_{-1} := 0$ der Lösung als hilfreich.

Umnormierung auf Intervall $[a,b]$

Die Iterationspunkte bei beliebigem Startintervall $[a,b]$ ergeben sich durch lineare Transformation mit dem Faktor $h := \frac{b-a}{F_n}$. Die Länge des Restintervalls in Iteration $i := 0, 1, \ldots, n-1$ ist $F_{n-i}h$. Im vorletzten Restintervall $[a_{n-2}, b_{n-2}]$ der Länge $2h$ fallen die Stützpunkte zusammen: $x_{n-2} = y_{n-2} = a_{n-2} + h$. Wegen $\hat{x} \in [a_{n-2}, b_{n-2}]$ gilt dann $|\hat{x} - x_{n-2}| \leq h$.

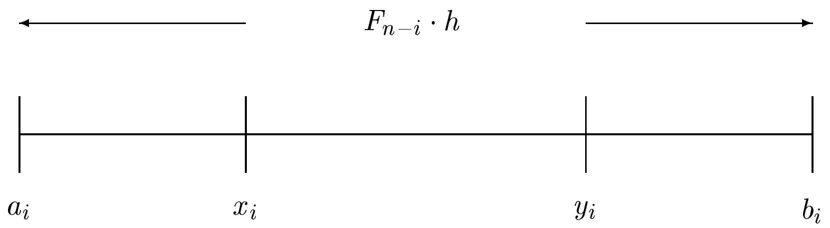

Abbildung III.5: Fibonacci-Suche

Algorithmus 1.2 (Fibonacci-Suche)

1. **Initialisierung:**
   Berechne die Fibonacci-Zahlen F_0 bis F_n.
   $i := 0$; $a_0 := a$; $b_0 := b$; $h := \frac{b-a}{F_n}$;
   $x_0 := a_0 + F_{n-2}h$; $y_0 := a_0 + F_{n-1}h$;
   $fx := f(x_0)$; $fy := f(y_0)$;

2. **Iteration:**
   Wenn $fx < fy$, dann

   $a_{i+1} := a_i$; $b_{i+1} := y_i$;
   $x_{i+1} := a_{i+1} + F_{n-i-3}h$; $y_{i+1} := x_i$;
   $fy := fx$; $fx := f(x_{i+1})$;

   sonst

   $a_{i+1} := x_i$; $b_{i+1} := b_i$;
   $x_{i+1} := y_i$; $y_{i+1} := b_{i+1} - F_{n-i-3}h$;
   $fx := fy$; $fy := f(y_{i+1})$;

   $i := i + 1$.

3. **Abbruch:**
   Wenn $i < n - 2$, dann gehe nach 2.
   $\hat{x} := x_i$; Stop.

Auch im obigen Algorithmus wird ein Funktionswert zuviel berechnet, da der im letzten Iterationschritt berechnete Funktionswert schon bekannt ist.

Ein Beispiel zur Fibonacci-Suche

Die Funktion f, definiert durch $f(x) = \sin(x-2)$, ist unimodal auf $[a,b] := [0,2]$. Gesucht wird der Minimalpunkt $\hat{x} = 2 - \frac{\pi}{2} \approx 0,429$. Die Fibonacci-Zahlen F_0 bis F_6 sind: $1, 1, 2, 3, 5, 8, 13$, d.h. bei Fibonacci-Suche mit $n = 6$ findet man ein Restintervall der Länge

$$h := \frac{b-a}{F_n} = \frac{2}{13} \approx 0,154.$$

dessen Genauigkeit hier ausreichen soll.

Die im Verfahren auftretenden Stützstellen (und Intervallgrenzen) sind alle von der Form $a + kh$ mit $k \in \{0, 1, \ldots, F_n\}$. Für eine Stützstelle $t \in [a, b]$ nennt man die entsprechende ganze Zahl $k(t) := \frac{t-a}{h}$ den Fibonacci-Index von t. Beim Start gilt $k(a_0) = 0$, $k(x_0) = F_{n-2}$, $k(y_0) = F_{n-1}$, $k(b_0) = F_n$.

In der Iteration werden die Funktionswerte $f(x_i)$ und $f(y_i)$ verglichen. Die Variable mit kleinerem Funktionswert bleibt Stützstelle, die mit größerem Funktionswert wird neue Intervallgrenze. Der Fibonacci-Index der neuen Stützstelle ergibt sich wegen der symmetrischen Lage der Stützstellen im Restintervall aus $k(x_{i+1}) - k(a_{i+1}) = k(b_{i+1}) - k(y_{i+1})$. Dies erspart bei Handrechnung das Nachschlagen in der Tabelle der Fibonaccizahlen. Ordnet man alle Werte einer Iteration zeilenweise in einer Tabelle an, so verschieben sich diese Werte bei Übergang zur nächsten Iteration nach rechts bzw. nach links ab der Position der neuen Stützstelle:

i	$k(a_i)$	$k(x_i)$	$k(y_i)$	$k(b_i)$	$f(x_i)$	$f(y_i)$
0	0	5	8	13	$-0,943$	$-0,696$
1	0	3	5	8	$-0,999$	$-0,943$
2	0	2	3	5	$-0,993$	$-0,999$
3	2	3	4	5	$-0,999$	$-0,983$
4	2	3	3	4	$-0,999$	$-0,999$

Mit $x_4 = y_4 = 0 + 3h$ bricht das Verfahren ab und es gilt: das Minimum von f ist näherungsweise gegeben durch $\hat{x} = 0,462 \pm 0,154$ ($\hat{x}$ liegt mit Sicherheit im Intervall $[a_4, b_4] = [0 + 2h, 0 + 4h]$).

Vergleich von goldenem Schnitt und Fibonacci-Suche

Aus $1 - \sigma = \sigma^2$ folgt induktiv $\sigma^n = (-1)^n(F_{n-2} - F_{n-1}\sigma)$. Aus $F_0 = 1 = F_1$ und $F_1 = 1 < \sigma^{-1} < 2 = F_2$ erhalten wir $F_2 < \sigma^{-2} \cdot (\sigma^2 + \sigma) < F_3$, also mit Hilfe der ersten Formel $F_2 < \sigma^{-2} < F_3$. Induktiv folgen für $n > 1$ die Abschätzungen:

$$\frac{1}{F_n} > \sigma^n > \frac{1}{F_{n+1}}, \qquad \lim_{n \to \infty} \frac{F_{n-1}}{F_n} = \sigma.$$

Demnach ist der Goldene Schnitt bei gleichem Aufwand von vergleichbarer kaum geringerer Genauigkeit. Auf der anderen Seite muß bei Fibonacci-Suche die gewünschte Genauigkeit a priori bekannt sein, um n zu bestimmen.

Beide Verfahren reduzieren die Intervallänge linear, d.h. es gilt $|b_{i+1} - a_{i+1}| \leq \lambda |b_i - a_i|$ für ein $\lambda \in (0, 1)$. Dabei ist der Reduktionsfaktor λ durch σ bzw. durch eine Schranke für $\frac{F_{n-1}}{F_n}$ gegeben. Nach

$$n = \left\lceil \frac{1}{\log \lambda} \left(\log(b - a) + \log \frac{1}{\epsilon} \right) \right\rceil$$

Iterationsschritten ist die Länge des Restintervalls auf weniger als ϵ reduziert. Als log wird in solchen Abschätzungen üblicherweise der Logarithmus zur Basis 2 gewählt, da dann die Logarithmen der Eingabedaten als Größe des Problems (Speicherplatz in einem Computer) interpretiert werden können.

Differenzierbare und konvexe Funktionen

Da das Minimum einer auf einem offenen Intervall $I \subseteq \boldsymbol{R}$ stetig differenzierbaren Funktion f notwendigerweise auch eine Nullstelle der Ableitung $g \equiv f'$ ist, versucht man zunächst, ein $\hat{x} \in I$ mit $g(\hat{x}) = 0$ zu finden. Kennt man ein Intervall $[a, b] \subset I$ mit

$$g(a) < 0 < g(b),$$

so folgt aus der Stetigkeit von g die Existenz einer Nullstelle in (a, b). Ist g monoton, so gibt es genau eine Nullstelle und diese ist das globale Minimum von f. Das einfachste Verfahren zur Bestimmung einer Nullstelle in I ist das Bisektionsverfahren.

Algorithmus 1.3 (Das Bisektionsverfahren)

1. **Iteration:**
   $x := \frac{a+b}{2}$; $g_x := g(x)$;
   Wenn $g_x \approx 0$, dann $\hat{x} := x$; Stop.
   Wenn $g_x < 0$, dann $a := x$;
   Wenn $g_x > 0$, dann $b := x$;

2. **Abbruch:**
   Wenn $b - a < \epsilon$, dann $\hat{x} := x$; Stop.
   Gehe nach 1.

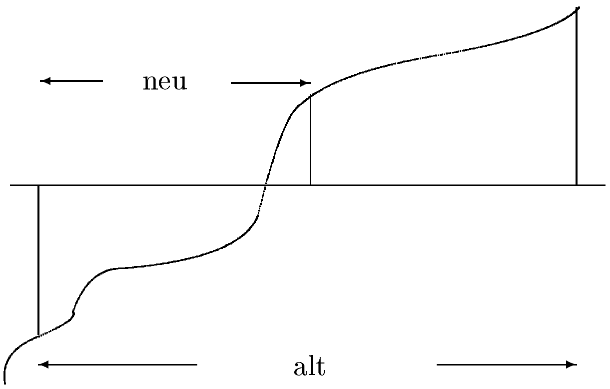

Abbildung III.6: Bisektion

Da das Bisektionsverfahren das Restintervall in jeder Iteration um den Faktor $\lambda = \frac{1}{2}$ reduziert, handelt es sich wieder um ein lineares Verfahren, das unter den angegebenen

Voraussetzungen stets eine Nullstelle von g lokalisiert. Schnellere Verfahren wie das im folgenden beschriebene Newton-Verfahren konvergieren im allgemeinen nur unter strengeren Voraussetzungen.

Sei wieder Intervall $[a,b] \subset I$ mit $g(a) < 0 < g(b)$ bekannt. Außerdem sei nunmehr $f \in C_2(I)$. Das Newton-Verfahren wird mit Hilfe der Iterationsfunktion

$$\Phi(x) := x - \frac{g(x)}{g'(x)}$$

beschrieben.

Algorithmus 1.4 (Newton-Verfahren)

1. $x_0 := b$.

2. Für $k = 0, 1, \ldots$*:*

$$x_{k+1} := \Phi(x_k);$$

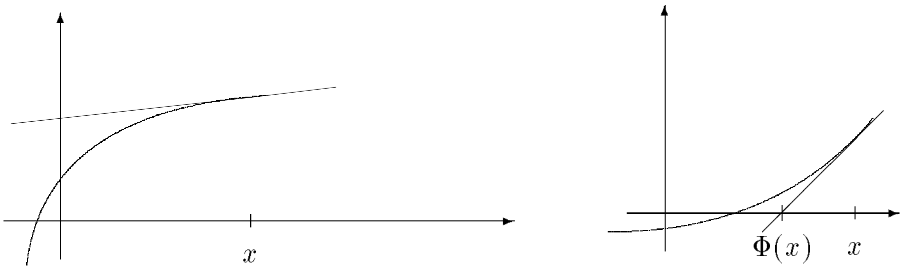

Abbildung III.7: Newton-Verfahren

Selbst wenn g monoton ist und daher eine eindeutige Nullstelle in (a,b) existiert, kann das Newtonverfahren versagen. Im folgenden betrachten wir eine wichtige Klasse von Funktionen, für die das Newton-Verfahren zum Erfolg führt.

Definition (Konvexe Funktionen): $g : I \to \boldsymbol{R}$ heißt *konvex*, falls

$$g(\lambda x + (1-\lambda)y) \leq \lambda g(x) + (1-\lambda)g(y)$$

für alle $\lambda \in (0,1)$, $x, y \in I$.

Im Abschnitt 2 werden konvexe Funktionen im $\boldsymbol{R}^n$ untersucht und auf verschiedene Weise charakterisiert. Für unsere Analyse des Newton-Verfahrens genügt uns die folgende grundlegende Eigenschaft konvexer, differenzierbarer Funktionen:

Lemma 1.1 (Gradientenungleichung) *$g : I \to \boldsymbol{R}$ sei eine konvexe, auf $(a,b) \subseteq I$ stetig differenzierbare Funktion. Dann gilt*

$$g(y) \geq g(x) + g'(x)(y-x)$$

für alle $x, y \in (a,b)$.

Beweis. Die Konvexitätsbedingung $g(\lambda y + (1-\lambda)x) \leq \lambda g(y) + (1-\lambda)g(x)$ impliziert $\frac{g(x+\lambda(y-x))-g(x)}{\lambda(y-x)}(y-x) \leq g(y) - g(x)$. Der Grenzübergang $\lambda \to 0$ führt daher auf $g'(x)(y-x) \leq g(y) - g(x)$. □

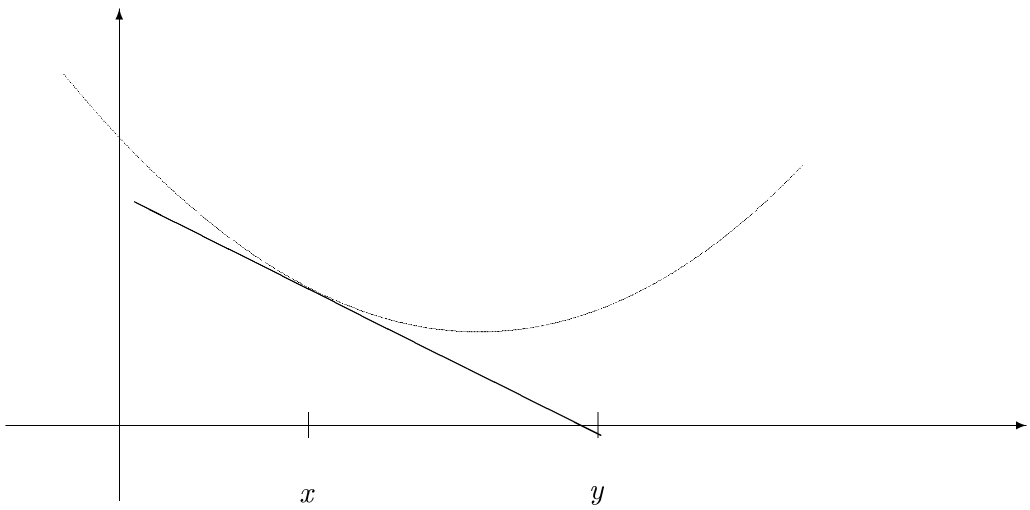

Abbildung III.8: Gradientenungleichung

Folgerung: Differenzierbare konvexe Funktionen besitzen monoton wachsende Ableitungen:

$$x < y \Rightarrow g'(x) \leq g'(y),$$

denn setzt man die Gradientenungleichung mit vertauschten Argumenten, d.h. $g(x) \geq g(y) + g'(y)(x-y)$ in die Gradientenungleichung ein, so folgt $g(y) \geq g(y) + g'(y)(x-y) + g'(x)(y-x)$. Also gilt $g'(y)(y-x) \geq g'(x)(y-x)$, woraus wegen $y - x > 0$ die Behauptung $g'(y) \geq g'(x)$ folgt.

Satz 1.2 (Globale Konvergenz des Newton-Verfahrens) *$g : I \to \boldsymbol{R}$ sei eine konvexe, auf $(a,b) \subseteq I$ stetig differenzierbare Funktion mit $g(a) < 0 < g(b)$. Dann besitzt g eine eindeutige Nullstelle $x_* \in (a,b)$, wobei $g'(x_*) > \frac{-g(a)}{b-a} > 0$, und für die im Newtonverfahren erzeugte Folge aus Algorithmus 1.4 gilt:*

1. *$(x_k) \searrow x_*$ (streng monoton fallend)*

Sei $\epsilon > 0$. Dann gilt zusätzlich:

2. *$g(x_k) \leq \epsilon$ für $2k \geq \left\lceil \log_2 \left(\frac{g'(b)g(b)}{g'(x_*)} \right) + \log_2 \frac{1}{\epsilon} \right\rceil$*
3. *$x_k - x_* \leq \epsilon$ für $2k \geq \left\lceil \log_2 \left(\frac{g'(b)g(b)}{(g'(x_*))^2} \right) + \log_2 \frac{1}{\epsilon} \right\rceil$*

Beweis. Da g stetig, gibt es nach Zwischenwertsatz eine Nullstelle $x_* \in (a,b)$. Angenommen, x'_* ist eine zweite Nullstelle. OBdA $x_* < x'_*$, d.h. $x_* = \lambda a + (1-\lambda)x'_*$ für ein $\lambda \in (0,1)$.

Da g konvex, folgt mit $0 = g(x_*) \leq \lambda g(a) + (1-\lambda)g(x'_*) = \lambda g(a) < 0$ ein Widerspruch. Im Rest des Beweises sei stets $a \leq x < x_* < y \leq b$. Aus der Eindeutigkeit der Nullstelle folgt für solche Argumente:

$$g(x) < 0 < g(y) \tag{1.2}$$

Die Gradientenungleichung liefert $g(a) \geq g(y) + g'(y)(a-y)$. Also folgt

$$g'(y) > \frac{g(y)-g(a)}{y-a} \geq \frac{-g(a)}{b-a} > 0. \tag{1.3}$$

Wegen (1.2) und (1.3) gilt $\Phi(y) < y$. Die Gradientenungleichung $g(y) + g'(y)(x_* - y) \leq g(x_*) = 0$ kann man nun durch $g'(y)$ teilen und erhält $x_* \leq y - \frac{g(y)}{g'(y)} = \Phi(y)$. Also ist die durch Algorithmus (1.4) erzeugte Newtonfolge x_k streng monoton fallend mit $x_k \geq x_*$.

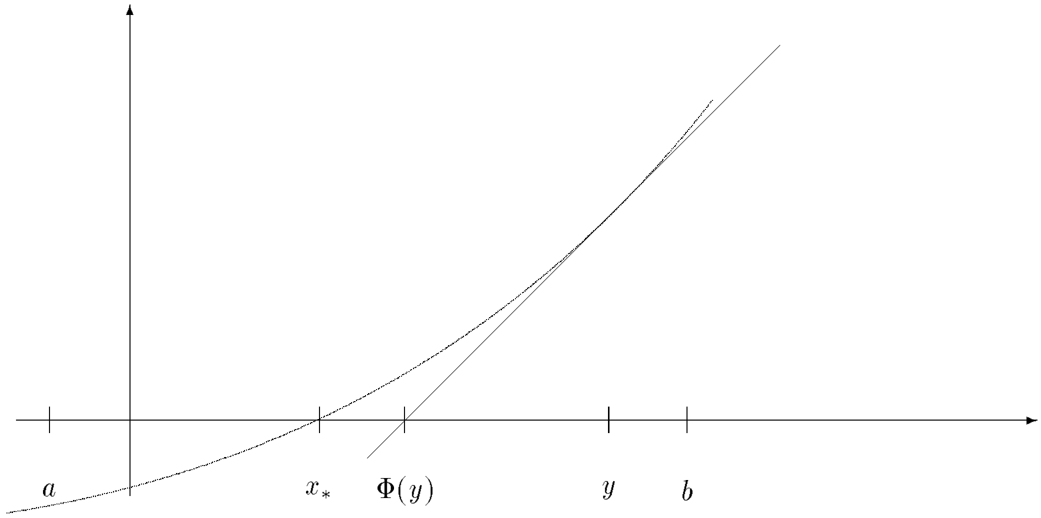

Abbildung III.9: Newton-Verfahren für konvexe Funktionen

Die Gradientenungleichung impliziert weiterhin

$$g(y) \geq g(\Phi(y)) + g'(\Phi(y))(y - \Phi(y)).$$

Mit Division durch $g(y)$ erhält man

$$1 \geq \frac{g(\Phi(y))}{g(y)} + \frac{g'(\Phi(y))}{g'(y)} =: \alpha + \beta.$$

Die Ungleichung zwischen dem geometrischen und dem arithmetischen Mittel ergibt nun $\sqrt{\alpha \cdot \beta} \leq \frac{1}{2}(\alpha+\beta) \leq \frac{1}{2}$, also $\alpha \cdot \beta \leq \frac{1}{4}$. Offensichtlich gilt

$$g\left(\Phi(y)\right) g'\left(\Phi(y)\right) \leq \frac{1}{4} g(y) g'(y),$$

woraus wir nach k Iterationen

$$g(x_k)g'(x_k) \leq (\frac{1}{4})^k g(b)g'(b)$$

erhalten. Zusammen mit $g'(x_k) \geq g'(x_*)$ liefert diese Ungleichung:

$$g(x_k) \leq (\frac{1}{4})^k \frac{g(b)g'(b)}{g'(x_*)}, \tag{1.4}$$

woraus (2.) folgt. Eine weitere Anwendung der Gradientenungleichung zeigt $g(x_k) \geq g(x_*) + g'(x_*)(x_k - x_*)$, also $\frac{g(x_k)}{g'(x_*)} \geq x_k - x_*$. Damit folgt aus (1.4) auch (3.). □

Definition (Konvergenzgeschwindigkeit): Sei (r_n) eine positiveNullfolge, d.h. $r_n \to 0, r_n > 0$. Dann heißt (r_n) *(mindestens) konvergent* von der Ordnung p, falls es ein $b > 0$ und ein $N \in \boldsymbol{N}$ gibt, so daß $r_{n+1} \leq b \cdot r_n^p$ für alle $n > N$.

(r_n) hat die *Konvergenzordnung* $p := \sup\{q > 0 \mid 0 \leq \beta_q = \limsup_{n\to\infty} \frac{r_{n+1}}{r_n^q} < \infty\}$. Dabei heißt β_q *Konvergenzfaktor* zur Ordnung q.

Beispiel:

1. Für $r_{n+1} := \frac{r_n}{2}$ mit $r_0 := 1$ gilt: $p = 1$, $\beta_1 = \frac{1}{2}$.
2. Für $r_{n+1} := r_n^2$ mit $r_0 := \frac{1}{2}$ gilt: $p = 2$, $\beta_2 = 1$.
3. Für $r_n := \frac{1}{n}$ gilt: $\beta_1 = 1$, $p = 1$.
4. Für $r_n := (\frac{1}{2})^{n^2}$ gilt: $\beta_1 = 0$,$p = 1$.

Insbesondere bezeichnet man $p = 1$, $0 < \beta_1 < 1$ als *lineare Konvergenz*, $p \geq 1, \beta_1 = 0$ als *superlineare Konvergenz* und $p = 2$ als *quadratische Konvergenz*. Die bei den bisherigen Verfahren (Goldener Schnitt, Fibonacci-Suche, Bisektionsverfahren) betrachtete Länge des Restintervalls ist jeweils eine linear konvergente Nullfolge, für die das Konvergenzverhalten unmittelbar ab $N = 1$ einsetzt. Im allgemeinen beschreibt die Konvergenzgeschwindigkeit nur das asymptotische Verhalten der Folge. Dies wird auch bei der folgenden Untersuchung des Newton-Verfahrens deutlich, die wir für lipschitzstetig differenzierbare Funktionen durchführen.

Lemma 1.3 (L-Lipschitzstetig differenzierbare Funktionen) *Sei $L > 0$ und g eine auf (a,b) stetig differenzierbare Funktion mit $|g'(y) - g'(x)| \leq L \cdot |y - x|$. Dann gilt:*

$$|g(y) - g(x) - g'(x) \cdot (y - x)| \leq \frac{1}{2}L(y-x)^2$$

für alle $x, y \in (a,b)$.

Beweis.

$$\begin{aligned} |g(y) - g(x) - g'(x)(y-x)| &\leq \int_x^y |g'(z) - g'(x)|dz \\ &\leq L\int_x^y (z-x)dz \\ &= \frac{L}{2}(y-x)^2 \end{aligned}$$

□

Mit Hilfe dieser Abschätzung kann man die Konvergenzgeschwindigkeit des Newtonverfahrens bestimmen.

Satz 1.4 (Lokale Konvergenz des Newton-Verfahrens) *Sei f eine auf (a,b) zweimal stetig differenzierbare Funktion mit L-lipschitzstetig differenzierbarer Ableitung $g \equiv f'$ und mit $|g'| \geq \rho > 0$. Besitzt $g(x) = 0$ eine Lösung $x_* \in (a,b)$, dann existiert ein $\eta > 0$ mit*

1. $|x_0 - x_*| < \eta \quad \Rightarrow \quad (x_k) \to x_*$
2. $|x_{k+1} - x_*| \leq \frac{L}{2\rho}(x_k - x_*)^2$

für die Newtonfolge $x_{k+1} := x_k - \frac{g(x_k)}{g'(x_k)}$.

Beweis. (Übungsaufgabe) □

Der Satz formuliert lokal, hinreichend nahe beim gesuchten Minimum x_*, eine Konvergenzaussage. Dort liefert das Newtonverfahren in jedem Schritt eine Verdopplung der Genauigkeit, d.h. eine typische Entwicklung des Abstandes $|x_{k+1} - x_*|$ ist etwa 10^{-2}, 10^{-4}, $10^{-8}, \ldots$.

Newton-Verfahren mit finiten Differenzen

Approximiert man die zur Durchführung des Newtonverfahrens explizit benötigte Ableitung von f, so erhält man eine Reihe von Varianten. Die geschickte Approximation $g'(x_k) \approx \frac{g(x_k) - g(x_{k-1})}{x_k - x_{k-1}}$ führt auf das Sekantenverfahren:

$$x_{k+1} = x_k - g(x_k)\frac{x_k - x_{k-1}}{g(x_k) - g(x_{k-1})}, \quad k \in \boldsymbol{N}.$$

wobei zur Initialisierung zwei Startpunkte x_0, x_1 gewählt werden müssen. Bei allgemeiner Approximation der Ableitung mit Hilfe des Differenzenquotienten zur Schrittweite h_c spricht man vom Newton-Verfahren mit finiten Differenzen. Der jeweils neue Iterationspunkt x_+ ergibt sich aus dem alten x_c durch

$$a_c := \frac{g(x_c + h_c) - g(x_c)}{h_c}; \quad x_+ := x_c - \frac{g(x_c)}{a_c}$$

Konvergenzabschätzung

Für eine auf (a,b) L-lipschitzstetig differenzierbare Funktion g mit $x_c, x_c + h_c \in (a,b)$ ist die Qualität der Approximation abschätzbar durch:

$$|a_c - g'(x_c)| \leq \frac{L}{2}|h_c|,$$

denn nach Lemma 1.3 gilt $|g(x_c + h_c) - g(x_c) - h_c g'(x_c)| \leq \frac{L}{2}|h_c|^2$. Ist $g(x_*) = 0$, so ergibt sich

$$\begin{aligned}
x_+ - x_* &= -\frac{g(x_c)}{a_c} + x_c - x_* \\
&= \frac{1}{a_c}\left[g(x_*) - g(x_c) - a_c(x_* - x_c)\right] \\
&= \frac{1}{a_c}\left[g(x_*) - g(x_c) - g'(x_c)(x_* - x_c) + (g'(x_c) - a_c)(x_* - x_c)\right]
\end{aligned}$$

Mit $e_+ := |x_+ - x_*|, e_c := |x_* - x_c|$ folgt $e_+ \leq \frac{1}{a_c}\left[\frac{L}{2}(x_* - x_c)^2 + \frac{L}{2}|h_c|e_c\right]$, und daher

$$e_+ \leq \frac{L}{2|a_c|}(e_c + |h_c|)e_c \tag{1.5}$$

Diese Abschätzung impliziert unter einer Reihe unterschiedlicher Annahmen Konvergenzaussagen:

Satz 1.5 (Lokale Konvergenz mit finiten Differenzen) *Sei f eine auf (a,b) zweimal stetig differenzierbare Funktion mit L-lipschitzstetig differenzierbarer Ableitung $g \equiv f'$ und mit $|g'| \geq \rho > 0$. Besitzt $g(x) = 0$ eine Lösung $x_* \in (a,b)$, dann existieren $\eta, \eta' > 0$, so daß für $0 < |h_k| \leq \eta', |x_0 - x_*| < \eta$ gilt:*

1. *x_k konvergiert mindestens linear gegen x_*,*
2. *falls $h_k \to 0$, so ist die Konvergenz mindestens superlinear,*
3. *falls $|h_k| \leq \alpha|x_k - x_*|$ oder $|h_k| \leq \alpha|g(x_k)|$ für ein $\alpha > 0$, so ist die Konvergenz mindestens quadratisch,*
4. *falls $|h_k| \leq \alpha|x_k - x_{k-1}|$ für ein $\alpha > 0$, so ist die Konvergenz mindestens quadratisch über je zwei Schritte .*

Beweis. (Übungsaufgabe), mittels (1.5) □

Bemerkung (Numerischer Vergleich und numerische Durchführung):

1. Das Newtonverfahren benötigt zwei Funktionsauswertungen (von g und g') pro Iteration und ist mindestens quadratisch konvergent. Unter den gleichen Voraussetzungen benötigt das Sekantenverfahren zwei Auswertungen (von g) in zwei Iterationen und ist wegen (4.) des vorstehenden Satzes, wenn man je zwei Iterationen zusammenfaßt, ebenfalls quadratisch konvergent. Eine genauere Konvergenzanalyse (s. Stoer [29]) zeigt sogar $p = 2.618\ldots = 1 + \frac{\sqrt{5}+1}{2}$ für Doppelschritte beim Sekantenverfahren.
2. Finite Differenzen sind numerisch nicht unproblematisch. Aus $h_k := \alpha|g(x_k)| \to 0$ folgt $x_k + h_k \to x_k$. Eine praktikable Wahl ist etwa $h_k := \sqrt{eps} \cdot \max\{\text{typ } x, |x_k|\}$. Hierbei gibt typ x die zu erwartende Größenordnung von x und eps die Maschinengenauigkeit an.
3. Im Gegensatz zur Nullstellensuche ist bei der Minimierung von Funktionen eine natürliche Qualitätskontrolle zur Hand. Falls $f(x_{k+1}) < f(x_k)$, ist die Iteration sinnvoll. Anderenfalls, falls wenigstens eine Abstiegsrichtung vorliegt, d.h. $f'(x_k)(x_{k+1} - x_k) < 0$ gilt, so stimmt die Richtung des Iterationsschritts und man kann versuchen durch, falls notwendig mehrfache, Halbierung der Schrittweite einen besseren Iterationspunkt zu finden. Ist die Richtung keine Abstiegsrichtung, so kann man in der Gegenrichtung suchen.

4. Als Abbruchkriterien kann man viele verschiedene Bedingungen verwenden, die alle mehr oder minder gut motiviert werden können. Beim Newton-Verfahren benutzt man die auftretenden relativen bzw. absoluten Änderungen:
$$\frac{|x_+ - x_c|}{\max\{\text{typ } x, |x_+|\}} < \tau_1; \quad \left|\frac{f'(x_+)}{f(x_+)} x_+\right| < \tau_2; \quad \left|\frac{f'(x_+)}{f(x_+)}\right| < \tau_3.$$

Interpolation

Sei f zweimal stetig differenzierbar auf dem offenen Intervall I. Gesucht wird wieder $f(x_*) = \min\{f(x) \mid x \in I\}$. Man kann nun f durch ein Interpolationspolynom P zu f approximieren und versuchen, dessen Minimum x_+ mit $P'(x_+) = 0, P''(x_+) > 0$ als Näherung für $\hat{x}$ zu verwenden. Das quadratische Interpolationspolynom P (zu f an der Stelle x_c) und seine Ableitung sind durch

$$P(x) = f(x_c) + f'(x_c)(x - x_c) + \frac{1}{2} f''(x_c)(x - x_c)^2$$
$$P'(x) = f'(x_c) + f''(x_c)(x - x_c)$$

gegeben (Taylorentwicklung). Löst man die zweite Gleichung nach x auf, so erkennt man, daß eine Newton-Iteration eine Nullstelle von P' liefert:

$$x_+ = x_c - \frac{f'(x_c)}{f''(x_c)}.$$

Diese Idee verwendet man in der quadratischen Interpolation iterativ. Allerdings wird statt des Taylorpolynoms das Lagrange'sche Interpolationspolynom zu drei meist äquidistanten Stützstellen benutzt:

$$f(x) \approx \sum_{i=1}^{3} f_i \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)} =: P(x)$$

Äquidistante quadratische Interpolation

Zu äquidistanten Stützstellen $x := y - h$, y, $z := y + h$ mit Schrittweite h seien die Stützwerte $f_\chi := f(\chi)$ für $\chi \in \{x, y, z\}$ bekannt. Dann ist die Nullstelle x_+ von P' gegeben durch:

$$x_+ = y + \frac{h}{2} \frac{f_x - f_z}{f_x - 2f_y + f_z}.$$

Der Nenner ist wohldefiniert, wenn die 2. Ableitung nicht verschwindet, denn

$$P''(x_+) = \frac{1}{h^2}(f_x - 2f_y + f_z),$$

woran man auch erkennt, ob ein Minimum oder Maximum bestimmt wurde. Die für ein Minimum hinreichende Bedingung $f_y < \min(f_x, f_z)$ impliziert auch $x_+ \in [x, z]$. Daher wird zunächst eine Näherungslösung y zur Schrittweite h gesucht, für die die Bedingung

$$f_y \leq f_x, f_z$$

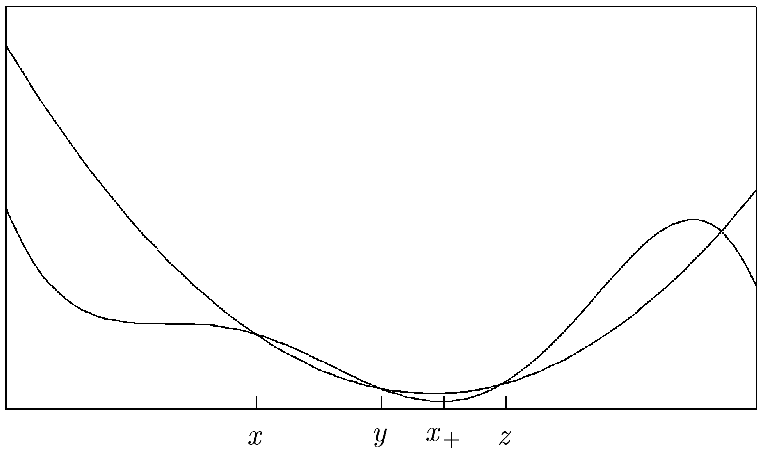

Abbildung III.10: Quadratische Interpolationsfunktionen

erfüllt ist. Dann wird x_+ berechnet und anschließend die Schrittweite h verkleinert. Kennt man näherungsweise den minimalen Funktionswert $f_* \approx f(x_*)$, und die Ableitung $f'(y)$, so kann die Schrittweite mit

$$h := 2\frac{f_* - f_y}{f'(y)} \tag{1.6}$$

automatisch gesteuert werden. Diese Formel erhält man, wenn man die Taylorentwicklung (1. Ordnung) der Ableitung

$$0 = f'(x_*) = f'(y) + f''(y)h$$

mit $(-\frac{h}{2})$ multipliziert, von der Taylorentwicklung (2. Ordnung) der Funktion

$$f_* = f(x_*) = f(y) + f'(y)h + \frac{1}{2}f''(y)h^2$$

abzieht, und die resultierende Gleichung

$$f_* = f(y) + \frac{h}{2}f'(y)$$

nach h auflöst. Im folgenden Verfahren ist die systematische Suche nach einem passenden Näherungswert y mit einer teilweisen Schrittweitensteuerung gekoppelt. Die in Gleichung (1.6) angegebene Schrittweite kann gegebenenfalls in den Schritten (1.) und (6.) verwendet werden.

Algorithmus 1.5 (Das Verfahren nach Davies, Swann und Campey)

1. Wähle Näherung y und Schrittweite h.

2. Falls $f(y) < f(y-h)$ und $f(y) < f(y+h)$, dann gehe nach Schritt 5.

   Anderenfalls wähle das Vorzeichen von h, so daß $f(y) \geq f(y+h)$.

3. $h_0 := h$;

   Solange $f(y+h) \geq f(y+2h)$: $h := 2h$;

   Falls $h = h_0$, dann $y := y + h$ und gehe nach 5.

   $h := \frac{1}{2}h$.

4. Falls $f(y+2h) \leq f(y+3h)$, dann $y := y + 2h$ und gehe nach 5.

   $y := y + 3h$.

5. $x_+ := y + \frac{h}{2} \cdot \frac{f(y-h)-f(y+h)}{f(y-h)-2f(y)+f(y+h)}$

6. Falls $f(y) > f(x_+)$, dann $y := x_+$.

   Falls Abbruchbedingung erfüllt, stop mit Ergebnis y.

   Wähle kürzere Schrittlänge h und gehe nach 2.

2 Lokale und Globale Minima

Stetig differenzierbare Funktionen bezeichnen wir wie üblich kurz als C_1-Funktionen, m-mal stetig differenzierbare als C_m-Funktionen. Ist f eine C_m-Funktion auf $X \subseteq \boldsymbol{R}^n$, so schreiben wir kurz $f \in C_m(X)$. Die Menge der inneren Punkte von X bezeichnen wir mit int$X$. Ist $f \in C_1$, so bezeichnet $\nabla f = (\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n})$ den Gradienten, ist $f \in C_2$, so bezeichnet $F = \nabla^2 f = (\frac{\partial^2 f}{\partial x_i \partial x_j})$ die Matrix der zweiten partiellen Ableitungen von f.

In diesem Abschnitt stellen wir Begriffe, Optimalitäts- und Konvergenzbedingungen für Verfahren zulässiger Richtungen zusammen, die als theoretische Grundlage zur Lösung der Aufgabe

$$\min\{f(x) \mid x \in X\}$$

nützlich sind.

Optimalitätsbedingungen

Definition (Minima): $U_\epsilon(x_*) := \{x \mid \| x - x_* \| < \epsilon\}$ heißt *ϵ-Kugel* um x_*.

1. $x_* \in X$ heißt *lokales Minimum* (von f auf X) genau dann, wenn

$$f(x_*) \leq f(x) \quad \exists\, \epsilon > 0 \quad \forall\, x \in X \cap U_\epsilon(x_*), x \neq x_*.$$

2. $x_* \in X$ heißt *globales Minimum* (von f auf X) genau dann, wenn

$$f(x_*) \leq f(x) \quad \forall\, x \in X, x \neq x_*.$$

x_* heißt *strenges lokales bzw. globales Minimum*, wenn entsprechend "$<$" gilt.

In Algorithmen werden häufig Richtungen erzeugt, in denen dann entweder die aktuelle Lösung verbessert oder deren Optimalität in der gewählten Richtung festgestellt werden muß. Natürlich sind nur solche Richtungen wesentlich, in denen zumindest nahe bei der aktuellen Lösung noch zulässige Punkte (in X) existieren.

Definition (Intervalle und zulässige Richtungen):

1. Für $a, b \in \boldsymbol{R}^n$ heißt die Menge $[a, b] := \{\lambda a + (1-\lambda)b \mid 0 \leq \lambda \leq 1\}$ *abgeschlossenes Intervall* von a bis b.

2. Sei $x \in X$. $d \in \boldsymbol{R}^n$ heißt *zulässige Richtung (bei x)* , falls ein positives λ mit $[x, x+\lambda d] \subseteq X$ existiert.

Die Menge aller zulässigen Richtungen (bei x) wird mit $D(x)$ bezeichnet.

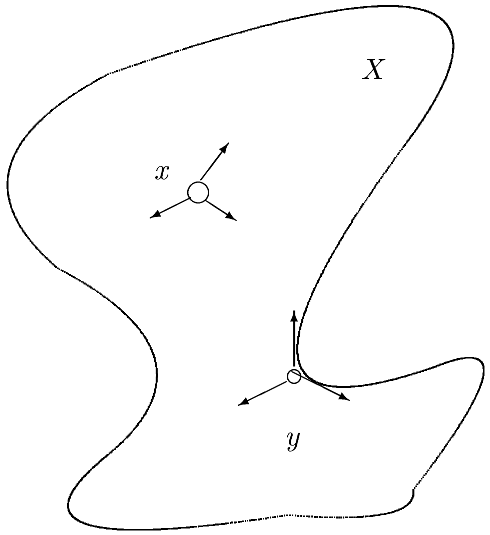

Abbildung III.11: Zulässige Richtungen

Zunächst soll an die aus der Analysis bekannten Optimalitätsbedingungen erinnert werden.

Satz 2.1 (Notwendige Bedingungen) *x_* sei lokales Minimum von $f \in C_1(X)$.*

1. *Da $f \in C_1$, so gilt:* $\quad \nabla f(x_*)d \geq 0 \quad \forall\, d \in D(x_*)$

2. *Ist $f \in C_2$, so gilt außerdem:* $\quad \nabla f(x_*)d = 0 \Rightarrow d^T F(x_*)d \geq 0 \quad \forall\, d \in D(x_*)$

Beweis. zu (1.): Zu $d \in D(x_*)$ mit $\nabla f(x_*)d < 0$ sei $x(\lambda) := x_* + \lambda d \in X$ für $\lambda \in [0, \bar{\lambda}], \bar{\lambda} > 0$. Dann besitzt die Funktion $g(\lambda) := f(x(\lambda))$ auf $[0, \bar{\lambda}]$ ein lokales Minimum in $\lambda = 0$ mit

Ableitung $g'(0) = \nabla f(x_*)d < 0$. Nun ergibt sich aus der Taylorentwicklung (1. Ordnung) der Ableitung für $\lambda > 0$

$$\frac{g(\lambda) - g(0)}{\lambda} = g'(0) + \frac{o(\lambda)}{\lambda},$$

ein Widerspruch, da bekanntlich $\frac{o(\lambda)}{\lambda} \to 0$ für $\lambda \to 0$. Für positive λ ist die linke Seite stets nichtnegativ, die rechte Seite jedoch wird für hinreichend kleine positive λ negativ.

Zu (2.): Hier gilt nun zusätzlich $g'(0) = 0, g''(0) = d^T F(x_*)d < 0$. Daher lautet die Taylorentwicklung (2. Ordnung) nun

$$g(\lambda) - g(0) = \frac{1}{2} g''(0)\lambda^2 + o(\lambda^2)$$

und man erhält einen analogen Widerspruch. □

Eine $n{\cdot}n$-Matrix A heißt bekanntlich *positiv semidefinit*, falls $d^T Ad \geq 0$ für alle $d \in \boldsymbol{R}^n$, und *positiv definit*, falls $d^T Ad > 0$ für alle $d \in \boldsymbol{R}^n \backslash \{0\}$.

Korollar 2.2 *Sei $x_* \in intX$. Ist $f \in C_1$, so gilt $\nabla f(x_*) = 0$. Ist $f \in C_2$, so ist zusätzlich $F(x_*)$ positiv semidefinit.*

Satz 2.3 (Hinreichende Bedingungen) *Sei $x_* \in intX, f \in C_2$. Falls $\nabla f(x_*) = 0$ und $F(x_*)$ positiv definit, so ist x_* strenges lokales Minimum von f auf X.*

Beweis. Da F stetig, ist $F(x)$ positiv definit auf $U_\epsilon(x_*)$ für hinreichend kleine $\epsilon > 0$. Die Taylorentwicklung (2. Ordnung) liefert $f(x_*+d) - f(x_*) = \frac{1}{2} d^T F(y)d$ für ein $y \in [x_*, x_*+d]$. Da nun $F(y)$ positiv definit für alle d mit $\| d \| < \epsilon$, folgt hieraus $f(x_* + d) > f(x_*)$ für alle d mit $\| d \| < \epsilon$. □

Korollar 2.4 *Sei $x_* \in intX, f \in C_2, F(x_*)$ nicht singulär. Dann gilt: x strenges lokales Minimum von f auf X genau dann, wenn $\nabla f(x_*) = 0 und F(x_*)$ positiv semidefinit.*

Beweis. Nach Korollar 2.2 sind die Bedingungen notwendig. Sie sind auch hinreichend, da $F(x_*)$ positiv semidefinit und nicht singulär, also positiv definit ist. □

Konvexe Funktionen

Zunächst wird die Definition der Konvexität auf die hier betrachteten Funktionen ausgedehnt.

Definition (Konvexe Mengen und Funktionen):

1. Sei $X \subseteq \boldsymbol{R}^n$. Dann heißt X konvex, falls $[a, b] \subseteq X$ für alle $a, b \in X$.
2. Sei $X \subseteq \boldsymbol{R}^n$. Dann heißt

$$convX := \left\{ \sum_{i=1}^{m} \lambda_i x_i | m \in \boldsymbol{N}, x_i \in X, \lambda_i \in \boldsymbol{R}_+, \sum_{i=1}^{m} \lambda_i = 1 \right\}$$

   die *konvexe Hülle* von X.

3. Sei X konvex. Dann heißt $f : X \to \boldsymbol{R}$ konvex (auf X), falls

$$f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y)$$

für alle $\lambda \in [0,1]$, $x, y \in X$.

Bemerkung:

1. $convX$ ist konvex.
2. Sei $X \subseteq Y$,Y konvex. Dann ist $convX \subseteq Y$. D. h. $convX$ ist die kleinste konvexe Obermenge von X.
3. Sei X konvex. Dann ist $convX = X$.

Beweis. (Übungsaufgabe)

Offenbar ist eine Funktion konvex, wenn sie eingeschränkt auf die abgeschlossenen Intervalle ihres konvexen Definitionsbereiches konvex ist.

Lemma 2.5 *Sei f konvex (auf X). Dann sind die Niveaumengen*

$$A_\alpha := \{x \in X \mid f(x) < \alpha\} \quad , \quad B_\alpha := \{x \in X \mid f(x) \le \alpha\}$$

konvex für alle $\alpha \in \boldsymbol{R}$.

Beweis. Für A_α: Seien $x, y \in A_\alpha, \lambda \in (0,1)$. Dann folgt $f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y) < \alpha$. Für B_α ist der Beweis analog. □

Bemerkung: Sei f konvex (auf X). Es gilt:

$$\forall x \in Y \subset X : f(x) \le \alpha \Rightarrow \forall x \in convY : f(x) \le \alpha$$

.

Denn: $Y \subset B_\alpha$, B_α konvex $\Rightarrow convY \subset B_\alpha$.

Eine überraschende Konsequenz der Konvexität zeigt der folgende Satz.

Satz 2.6 (Stetigkeit) *Sei f konvex (auf X). Dann ist f stetig auf $intX$.*

Beweis. $\bar{x} \in \mathrm{int}X$ beliebig. OBdA $\bar{x} = 0$, $f(\bar{x}) = 0$. (Nötigenfalls kann man sonst die Funktion $\tilde{f}(y) := f(y) - f(\bar{x})$ betrachten). Zu zeigen ist daher nur: $\| x \| < \delta(\epsilon) \Rightarrow |f(x)| < \epsilon$.

Sei $W_\rho := \{x \mid \| x \|_\infty \le \rho\} \subseteq X$ und sei M das Maximum von f auf den endlich vielen Ecken des Würfels W_ρ. Jeder Punkt des Würfels ist Konvexkombination gewisser Ecken V_ρ ($convV_\rho = W_\rho$). Da B_M konvex ist und die Ecken enthält, gilt $W_\rho \subseteq B_M$, d.h. $f|_{W_\rho} \le M$.

Für x mit $\| x \| = \alpha \le \rho$ gilt $u = \frac{\rho}{\alpha} x \in W_\rho$. Also ist $f(\frac{\rho}{\alpha} x) \le M$. Nun liegt x im Interval $[0, u]$. Da f konvex, folgt $f(x) = f\left((1 - \frac{\alpha}{\rho})0 + \frac{\alpha}{\rho} u\right) \le \frac{\alpha}{\rho} M$.

Analog erhält man aus $\| -x \| = \alpha$ die Abschätzung $f(-x) \le \frac{\alpha}{\rho} M$. Da f konvex und $0 \in [-x, x]$ liegt, folgt $0 = f(0) \le \frac{1}{2} f(x) + \frac{1}{2} f(-x) \le \frac{1}{2}(f(x) + \frac{\alpha}{\rho} M)$. Also gilt auch $f(x) \ge -\frac{\alpha}{\rho} M$.

Zusammen ergibt sich $|f(x)| \leq \frac{M}{\rho}\alpha = \frac{M}{\rho} \parallel x \parallel$, woraus man die Stetigkeit im Ursprung abliest. □

Auch Differenzierbarkeitseigenschaften folgen aus der Konvexität, allerdings nur in Richtungen.

Definition (Richtungsableitungen): Sei f eine Funktion auf X, $\bar{x} \in X$, $d \in D(\bar{x})$. Falls der Grenzwert

$$\lim_{\lambda \to 0^+} \frac{f(\bar{x} + \lambda d) - f(\bar{x})}{\lambda}$$

existiert, so heißt er *Richtungsableitung* von f bei $\bar{x}$ in Richtung d und wird mit $f'(\bar{x}; d)$ bezeichnet.

Satz 2.7 (Richtungsdifferenzierbarkeit) *Sei f konvex (auf X), $d \in D(\bar{x})$. Dann existiert $f'(\bar{x}; d)$ und es gilt*

$$f'(\bar{x}; d) = \inf_{\lambda > 0} \frac{f(\bar{x} + \lambda d) - f(\bar{x})}{\lambda}.$$

Beweis. Sei $\lambda > \mu > 0$ hinreichend klein. Dann folgt aus der Konvexität

$$f(\bar{x} + \mu d) = f(\frac{\mu}{\lambda}(\bar{x} + \lambda d) + (1 - \frac{\mu}{\lambda})\bar{x}) \leq \frac{\mu}{\lambda} f(\bar{x} + \lambda d) + (1 - \frac{\mu}{\lambda}) f(\bar{x}),$$

also nach elementarer Umformung

$$\frac{f(\bar{x} + \mu d) - f(\bar{x})}{\mu} \leq \frac{f(\bar{x} + \lambda d) - f(\bar{x})}{\lambda}.$$

Offenbar ist die Funktion $g(\lambda) := \frac{1}{\lambda}(f(\bar{x} + \lambda d) - f(\bar{x}))$ für $\lambda \to 0$ nicht wachsend. Läßt man den uneigentlichen Grenzwert $-\infty$ zu, so existiert $\lim_{\lambda \to 0^+} g(\lambda) = \inf_{\lambda > 0} g(\lambda)$ in jedem Fall. □

Die Konvexitätsungleichung und die Gradientenungleichung schließen die zugrundeliegende konvexe Funktion in jeder zulässigen Richtung in einer Schere ein. Die obere Abschätzung durch die Konvexität wird dabei immer enger, je kürzer die betrachtete Richtung ist (woraus wieder die Stetigkeit folgt).

Korollar 2.8 *Seien $\bar{x}, \bar{x} + d \in X$. Dann gilt für alle $\lambda \in [0, 1]$:*

$$f(\bar{x}) + \lambda f'(\bar{x}; d) \leq f(\bar{x} + \lambda d) \leq f(\bar{x}) + \lambda(f(\bar{x} + d) - f(\bar{x}))$$

Bemerkung: Die Glattheitseigenschaften konvexer Funktionen erlauben es, das eindimensionale Newton-Verfahren auf einem Intervall auch ohne explizite Annahme der stetigen Differenzierbarkeit zu definieren. Seine globale Konvergenz kann ähnlich wie im Beweis von Satz 1.2 nachgewiesen werden.

Das folgende Lemma zeigt, daß jeder Punkt einer konvexen Menge mit inneren Punkten über innere Punkte erreicht werden kann.

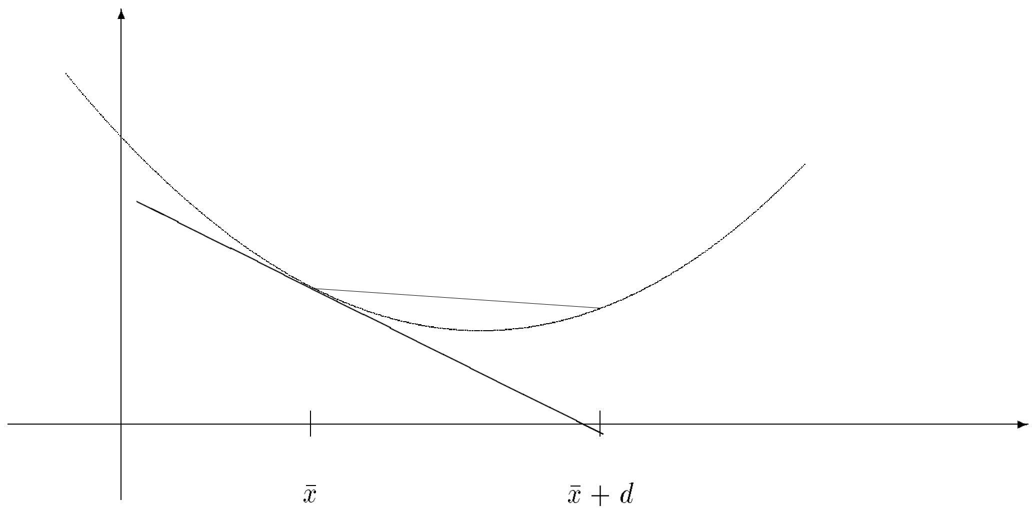

Abbildung III.12: Einschließung konvexer Funktionen

Lemma 2.9 (Erreichbarkeit) *Sei X konvex. Dann gilt $[\tilde{x}, x) \subseteq intX$ für alle $\tilde{x} \in intX$, $x \in X$.*

Beweis. Sei x' ein beliebiger Punkt in $(\tilde{x}, x)$, d.h. $x' = \lambda\tilde{x} + (1-\lambda)x$ mit $\lambda \in (0,1)$. Nach $\tilde{x}$ aufgelöst ergibt sich $\tilde{x} = \frac{1}{\lambda}(x' - (1-\lambda)x)$. Sei $U_\delta(\tilde{x}) \subseteq X$ und $y \in U_{\delta\lambda}(x')$. Sei $z := \frac{1}{\lambda}(y - (1-\lambda)x)$. Dann gilt $\| \tilde{x} - z \| = \frac{1}{\lambda} \| x' - y \|$, also $z \in U_\delta(\tilde{x})$. Da mit $z, x \in X$ auch die Konvexkombination y in X liegt, gilt $U_{\delta\lambda}(x') \subseteq X$. $\square$

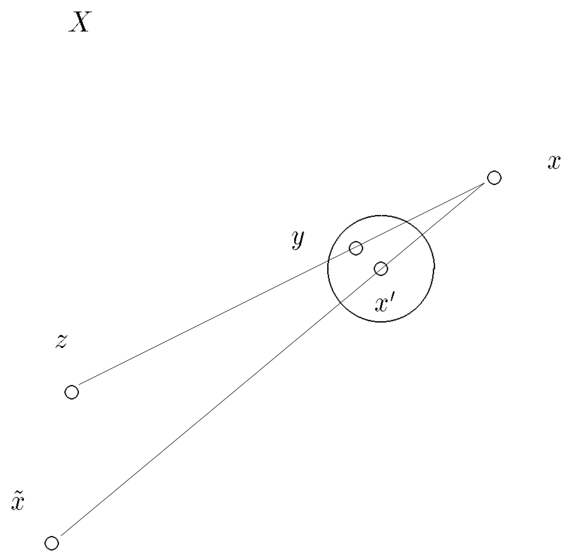

Abbildung III.13: Erreichbarkeitslemma

Das Erreichbarkeitslemma ermöglicht es, die Charakterisierung zweiter Ordung der Konvexität im folgenden Satz unter schwachen Voraussetzungen zu formulieren.

Satz 2.10 (Charakterisierung differenzierbarer konvexer Funktionen)
Sei $X \subseteq \boldsymbol{R}^n$ konvex, $f : \boldsymbol{R}^n \to \boldsymbol{R}$.

1. *(Gradientenungleichung):*
   Sei $f \in C_1$. Dann ist f konvex auf X genau dann, wenn $f(y) \geq f(x) + \nabla f(x)(y - x)$ für alle $x, y \in X$.

2. *Sei $f \in C_2$ und $\mathrm{int}X \neq \emptyset$. Dann ist f konvex auf X genau dann, wenn $F(x)$ positiv semidefinit für alle $x \in X$.*

Beweis. (zu 1.) Sei f konvex, d.h. $f(\alpha y + (1 - \alpha)x) \leq \alpha f(y) + (1 - \alpha) f(x)$, für $1 > \alpha > 0$. Nach elementaren Umformungen ergibt sich $\frac{1}{\alpha}\left(f(x + \alpha(y - x)) - f(x)\right) \leq f(y) - f(x)$, woraus für $\alpha \to 0_+$ die Gradientenungleichung folgt: $\nabla f(x)(y - x) \leq f(y) - f(x)$. Andererseits gelte die Gradientenungleichung. Für $x, y \in X$, $z := \lambda x + (1 - \lambda)y \in [x, y] \subseteq X$ ergeben sich für x, z bzw. y, z:

$$f(x) \geq f(z) + \nabla f(z)(x - z)$$

$$f(y) \geq f(z) + \nabla f(z)(y - z)$$

Multipliziert man die erste Ungleichung mit λ, die zweite mit $(1 - \lambda)$ und addiert beide, so ergibt sich

$$\lambda f(x) + (1 - \lambda) f(y) \geq f(z) + \nabla f(z)\left[\lambda x + (1 - \lambda)y - z\right],$$

wobei die eckige Klammer wegen der Definition von z verschwindet.

(zu 2.) Die Taylorentwicklung (1. Ordnung) liefert ein $\tilde{x} \in [x, y]$ mit

$$f(y) = f(x) + \nabla f(x)(y - x) + \frac{1}{2}(y - x)^T F(\tilde{x})(y - x) \tag{2.1}$$

Da F positiv semidefinit, folgt aus dieser Gleichung die Gradientenungleichung, und aus dieser nach (1.) die Konvexität von f.

Sei andererseits F nicht positiv semidefinit, etwa

$$(y - \bar{x})^T F(\bar{x})(y - \bar{x}) < 0 \tag{2.2}$$

für ein $\bar{x} \in X$ und für ein $y \in \boldsymbol{R}^n$.

OBdA $\bar{x} \in \mathrm{int}X$. Falls $\bar{x} \notin \mathrm{int}X$, so gilt wegen $\hat{x} \in \mathrm{int}X$ nach dem Erreichbarkeitslemma $[\hat{x}, \bar{x}] \subseteq \mathrm{int}X$. Aus Stetigkeitsgründen folgt aus der Ungleichung 2.2

$$(y - x)^T F(x)(y - x) < 0 \quad \forall\, x \in U(\bar{x}) \cap X.$$

Man kann daher in $U(\bar{x}) \cap [\hat{x}, \bar{x}]$ ein passendes neues $\bar{x}$ wählen.

Die Stetigkeit von F und die Ungleichung (2.2) ermöglichen nun die Wahl einer Umgebung $V(\bar{x}) \subseteq X$ mit

$$(y - \bar{x})^T F(\tilde{x})(y - \bar{x}) < 0 \quad \forall\, \tilde{x} \in V(\bar{x}). \tag{2.3}$$

Wir wählen jetzt ein $\bar{y} \in (\bar{x}, y) \cap V(\bar{x})$, d.h. nahe genug bei $\bar{x}$, so daß $[\bar{x}, \bar{y}] \subseteq X$. Ersetzt man dann y in der Ungleichung (2.3) durch $\bar{y}$, bleibt diese gültig. Daher gilt insbesondere

$$(\bar{y} - \bar{x})^T F(\tilde{x})(\bar{y} - \bar{x}) < 0 \quad \forall\, \tilde{x} \in [\bar{x}, \bar{y}].$$

Einsetzen in die Taylorentwicklung (2.1) für $x := \bar{x}, y := \bar{y}$, liefert die Ungleichung $f(y) < f(x) + \nabla f(x)(y - x)$, d.h. die Gradientenungleichung gilt nicht und daher ist nach (1.) f nicht konvex. □

Definition (Pseudokonvexe Funktionen): X konvex, $f \in C_1$. f heißt *pseudokonvex* auf X, falls für alle $x, y \in X$ gilt:

$$\nabla f(x)(y - x) \geq 0 \Rightarrow f(y) \geq f(x)$$

f heißt *pseudokonkav*, wenn $-f$ pseudokonvex ist, d. h. wenn für alle $x, y \in X$ gilt:

$$\nabla f(x)(y - x) \leq 0 \Rightarrow f(y) \leq f(x)$$

Die Gradientenungleichung zeigt, daß konvexe C_1-Funktionen pseudokonvex sind. Die Umkehrung ist im allgemeinen falsch. Wir betrachten als Beispiel die Funktion f mit $f(x) = x + x^3, f'(x) = 1 + 3x^2, f''(x) = 6x$. Da $f'' < 0$ für $x < 0$, ist f nicht konvex. Andererseits ist f pseudokonvex, denn für alle reellen x, y gilt:

$$f'(x)(y - x) \geq 0 \Rightarrow y - x \geq 0 \Rightarrow f(y) \geq f(x).$$

Satz 2.11 *Sei X konvex, f pseudokonvex. Dann ist $x_* \in X$ globales Minimum von f auf X genau dann, wenn $\nabla f(x_*)d \geq 0$ für alle $d \in D(x_*)$.*

Beweis. Nach Satz 2.1 ist die Bedingung notwendig. Andererseits sei $y \in X$ mit $y \neq x_*$. Dann gilt $y - x_* \in D(x_*)$, und daher ist nach Voraussetzung $\nabla f(x_*)(y - x_*) \geq 0$. Da f pseudokonvex, folgt $f(y) \geq f(x_*)$. □

Offenbar ist für pseudokonvexe Funktionen lokale Optimalität identisch mit globaler Optimalität. Abstiegsverfahren, mit deren Hilfe systematisch entsprechende Punkte gesucht werden, die die notwendige Bedingung des Satzes 2.1 erfüllen, lassen sich in folgender Weise schematisch formulieren.

Algorithmus 2.1 (Abstiegsverfahren)

1. ***Initialisierung***
   Finde $x \in X$.

2. ***Wahl der Abstiegsrichtung***
   Falls $\nabla f(x)d \geq 0 \quad \forall\, d \in D(x)$, dann stop.
   Wähle $d \in D(x)$ mit $\nabla f(x)d < 0$.

3. ***Wahl der Schrittweite***
   Wähle $\tilde{x}$ mit $f(\tilde{x}) \approx \min\{f(x + \lambda d) \mid x + \lambda d \in X, \lambda > 0\}$;
   $x := \tilde{x}$ und gehe nach 2.

Konvergenz

Bei Verfahren der Nichtlinearen Optimierung spricht man von Konvergenz, wenn entweder die nach endlich vielen Iterationsschritten bestimmte Lösung x_k oder alle Häufungspunkte der erzeugten Folge das Optimalitätskriterium, z.B. $\nabla f(x)d \geq 0$ für alle $d \in D(x)$, erfüllen. Ist M die Menge der Punkte des $\boldsymbol{R}^n$, die das Optimalitätskriterium erfüllen, so spricht man von *Konvergenz gegen* M. Da der endliche Fall, $x_k \in M$ für ein $k \in \boldsymbol{Z}_+$, trivial ist, wird meist nur untersucht, welche Häufungspunkte auftreten können.

Um Konvergenz gegen kritische Punkte zu erreichen, müssen die Wahl der Abstiegsrichtung und die Wahl der Schrittweite aufeinander abgestimmt werden. Die naheliegende Wahl der optimalen Schrittweite, d.h. die Bestimmung des exakten globalen Optimums $\tilde{x}$ in Schritt 3., ist aus numerischen Gründen meist nicht möglich.

Armijo-Suche (1. Ordnung) Sei $f \in C_1$, $0 < \alpha < 1$. Im Iterationsschritt k sei $\nabla f(x_k)d_k < 0$, $d_k \in D(x_k)$. Dann wähle $\lambda_k := 2^{-i(k)} \geq 0$ mit minimalen $i(k) \geq 0$,d. h. λ_k maximal, so daß

$$f(x_k + \lambda_k d_k) - f(x_k) \leq \alpha \cdot \lambda_k \nabla f(x_k) d_k. \tag{2.4}$$

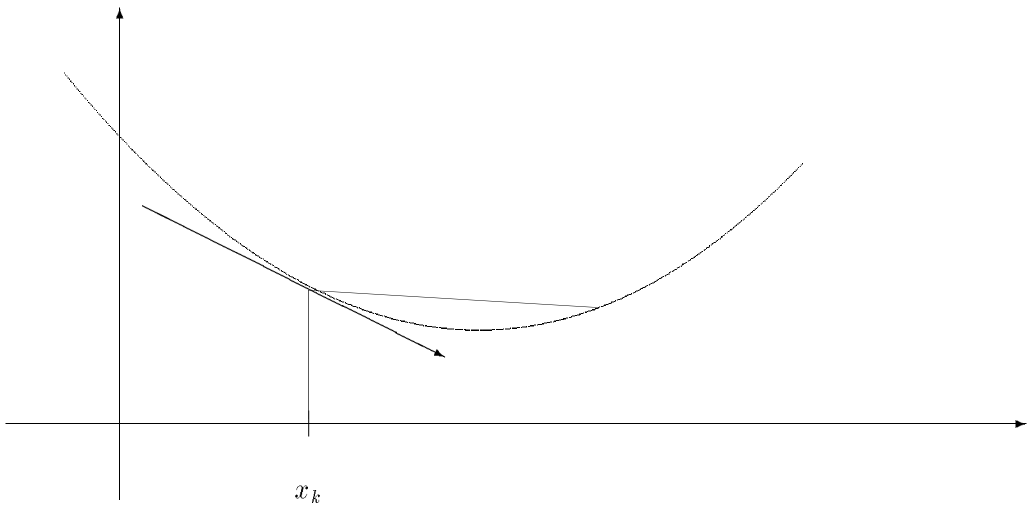

Abbildung III.14: Armijosuche (1. Ordnung). Sei $g(\lambda) := f(x_k+\lambda d_k)$. Hier genügt $i(k) = 2$, denn $g(\frac{1}{4}) \leq g(0) + \alpha\frac{1}{4}g'(0)$.

Unter den Voraussetzungen der Armijosuche ist der Index $i(k)$ wohldefiniert, da $0 < \alpha < 1$. Die Folge der erzeugten Funktionswerte $f(x_k)$ ist wegen (2.4) streng monoton fallend. Konvergiert eine Teilfolge $\{x_k\}_{k\in K}$ gegen $\bar{x}$, so konvergiert die Folge aller Funktionswerte gegen $f(\bar{x})$. Verwendet man die Armijosuche in Abstiegsverfahren, so läßt sich zeigen, daß konvergente Teilfolgen der erzeugten Iterationspunktefolge gegen kritische Punkte konvergieren.

Lemma 2.12 (Armijo-Abstieg 1. Ordnung) *Die Voraussetzungen der Armijo-Suche seien erfüllt und die Teilfolge $\{x_k\}_{k\in K}$ konvergiere gegen $\bar{x}$. Falls für geeignete Konstanten $\beta, \delta, \gamma > 0$:*

1. $\frac{\nabla f(x_k)d_k}{\|\nabla f(x_k)\|\|d_k\|} \leq -\delta \quad \forall\, k \in K,$
2. $\beta \parallel \nabla f(x_k) \parallel \leq \parallel d_k \parallel \leq \gamma \quad \forall\, k \in K,$

so gilt $\nabla f(\bar{x}) = 0$.

Beweis. 1. Fall: Für ein $\bar{\lambda}$ gilt: $\lambda_k \geq \bar{\lambda}$ für alle $k \in K$. Da für alle k die Ungleichung (2.4) und $\nabla f(x_k)d_k < 0$ gilt, folgt die Abschätzung

$$\begin{aligned} f(\bar{x}) - f(x_0) &= \sum_{k=0}^{\infty}(f(x_{k+1}) - f(x_k)) \\ &\leq \alpha \sum_{k=0}^{\infty} \lambda_k \nabla f(x_k)d_k \leq \alpha\bar{\lambda} \sum_{k\in K} \nabla f(x_k)d_k. \end{aligned}$$

Da die Summe beschränkt ist, bilden die Summanden eine Nullfolge. Wegen (1.) und (2.) lassen sich diese weiter abschätzen:

$$\nabla f(x_k)d_k \leq -\delta \parallel \nabla f(x_k) \parallel\parallel d_k \parallel \leq -\delta\beta \parallel \nabla f(x_k) \parallel^2 .$$

Wegen der Stetigkeit von Gradient und Norm folgt aus $\parallel \nabla f(x_k) \parallel \to 0$ unmittelbar $\parallel \nabla f(\bar{x}) \parallel = 0$.

2. Fall: $\{\lambda_k\}_{k\in K'} \to 0$ für eine Teilfolge $K' \subseteq K$. Für $2 \cdot \lambda_k$ ist die Ungleichung 2.4 nicht erfüllt. Mit dem Satz von Taylor (1. Ordnung) findet man:

$$\begin{aligned} \nabla f(x_k)d_k 2\lambda_k + o(\parallel d_k \parallel 2\lambda_k) &= f(x_k + d_k 2\lambda_k) - f(x_k) \\ &> \alpha 2\lambda_k \nabla f(x_k)d_k. \end{aligned}$$

Elementare Umformungen liefern mit Hilfe von (1.):

$$\frac{o(\parallel d_k \parallel 2\lambda_k)}{\parallel d_k \parallel 2\lambda_k} > \frac{(\alpha - 1)\nabla f(x_k)d_k}{\parallel d_k \parallel} \geq (1 - \alpha)\delta \parallel \nabla f(x_k) \parallel \geq 0.$$

Für $k \to \infty$ konvergiert der Nenner der linken Seite wegen (2.) gegen 0. Daher konvergiert die linke Seite gegen 0. Also muß $\parallel \nabla f(\bar{x}) \parallel = 0$ gelten. □

3 Gradienten- und Newtonverfahren

In diesem Abschnitt untersuchen wir für einmal oder zweimal stetig differenzierbare Funktionen $f : \boldsymbol{R}^n \to \boldsymbol{R}$ zwei einfache Verfahren zur Lösung der Aufgabe

$$\min\{f(x) \mid x \in \boldsymbol{R}^n\}. \tag{3.1}$$

Gesucht werden dabei statt der globalen Minima Lösungen, die die notwendigen Bedingungen erster oder, bei C_2-Funktionen, zweiter Ordnung für lokale Minima (siehe Satz 2.1) erfüllen. Eines der ältesten Verfahren, das bereits 1847 von Cauchy zur Lösung von Gleichungssystemen formuliert wurde, ist das folgende Abstiegsverfahren.

Algorithmus 3.1 (Methode des steilsten Abstiegs)

1. *Wähle* $x_0 \in \boldsymbol{R}^n$.

2. *Für* $k = 0, 1, \ldots$:
$$d_k := -\nabla f(x_k)^T; \quad x_{k+1} := x_k + \lambda_k d_k.$$

Wegen der Wahl der Abstiegsrichtung nennt man das Verfahren auch Gradientenverfahren.

Wahl der Schrittweite λ_k

1. **(Optimale Schrittweite)**. Man versucht das globale Optimum in der jeweiligen Richtung zu bestimmen:
$$\alpha := f(x_k + \lambda_k d_k) := \min_{\lambda \geq 0} f(x_k + \lambda d_k).$$
Dann verschwindet bei x_{k+1} die entsprechende Richtungsableitung $\nabla f(x_{k+1})^T d_k$, d.h. sukzessive Suchrichtungen sind orthogonal. Hier kann man Konvergenz gegen kritische Punkte untersuchen.

2. **(ϵ-optimale Schrittweite)**. Man akzeptiert Lösungen, deren Funktionswert nahe bei dem des globalen Optimums liegt:
$$f(x_k + \lambda_k d_k) \leq \alpha + \epsilon.$$
Hier kann man nur Konvergenz gegen "approximativ kritische" Punkte, d.h. Punkte mit $\| \nabla f(x) \| \leq \epsilon$ erwarten.

3. **(Armijo-Suche 1. Ordnung)**. In den Voraussetzungen der Armijo-Suche (siehe Lemma 2.12) sind mit $\beta = \delta = 1$ zwei der drei Ungleichungen erfüllt, denn
$$\frac{\nabla f(x_k) d_k}{\| \nabla f(x_k) \|_2 \| d_k \|_2} = \frac{- \| d_k \|_2^2}{\| d_k \|_2 \| d_k \|_2} = -1 \quad \text{und} \quad \| \nabla f(x_k) \| = \| d_k \| .$$

Für die praktikable Armijo-Suche wollen wir einen Konvergenzsatz formulieren, dessen Voraussetzungen in typischer Weise die noch fehlende Ungleichung $\| d_k \| \leq \gamma$ im Lemma 2.12 erzwingen.

Satz 3.1 *Bei Schrittweitenwahl nach Armijo ist das Gradientenverfahren konvergent, falls* $N(x_0) := \{x \mid f(x) \leq f(x_0)\}$ *beschränkt ist.*

Beweis. Wegen der Stetigkeit von f ist die Niveaumenge $N(x_0)$ kompakt. Da die Armijosuche fallende Funktionswerte $f(x_k)$ garantiert, gehören alle Iterationspunkte zu $N(x_0)$. Wir betrachten eine konvergente Teilfolge, oBdA. $x_k \to \bar{x}$. Wie bereits oben festgestellt sind zwei der drei Ungleichungen in den Voraussetzungen des Lemmas 2.12 erfüllt für $\delta = \beta = 1$. Außerdem ist ∇f auf dem Kompaktum $N(x_0)$ beschränkt, etwa
$$\| \nabla f(x) \| \leq \gamma \quad \forall\, x \in N(x_0)$$
für ein $\gamma > 0$. Wegen $d_k = -\nabla f(x_k)^T$ ist auch die dritte Ungleichung des Lemmas 2.12 erfüllt. □

Quadratische Funktionen

Sei Q eine reell symmetrische, positiv definite $n \cdot n$-Matrix und sei $b \in \boldsymbol{R}^n$. Dann ist

$$f(x) := \frac{1}{2} x^T Q x - b^T x$$

eine quadratische Funktion.

Mit Hilfe einer zugehörigen quadratischen Funktion können wir eine beliebige Funktion $g \in C_2$ in der Nähe ihres lokalen Minimums $\bar{x}$ modellieren. Bricht man die Taylorentwicklung von g nach dem Term 2. Ordnung ab, so gilt:

$$g(x) \approx g(\bar{x}) + b^T(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T Q (x - \bar{x}),$$

wobei $b^T := \nabla g(\bar{x}) = 0$ und $Q := G(\bar{x})$ positiv semidefinit.

Eigenschaften quadratischer Funktionen

Eine quadratische Funktionen f läßt sich besonders einfach minimieren.

1. Das Minimum x_* ist gegeben durch die Gleichung $Qx_* = b$, denn $\nabla f(x) = x^T Q - b^T = (Qx - b)^T$.

2. Der minimale Funktionswert ist $f(x_*) = \frac{1}{2}(Q^{-1}b)QQ^{-1}b - b^T Q^{-1} b = -\frac{1}{2} b^T Q^{-1} b$.

3. Das quadratische Modell der Taylorentwicklung (siehe oben) ist exakt, d.h.

   $$f(x) = f(x_*) + \frac{1}{2}(x - x_*)^T Q (x - x_*).$$

   Daher kann man zur Vereinfachung die rein quadratische Funktion $\tilde{f}(x) := f(x) - f(x_*)$ untersuchen, die dasselbe Minimum besitzt.

4. Das globale Minimum in einer Richtung $d \neq 0$ kann explizit durch x angegeben werden. Die Funktion $h : \boldsymbol{R} \to \boldsymbol{R}$ mit $h(\lambda) := f(x + \lambda d)$ ist eine konvexe C_1-Funktion. Das Minimum λ_* ist gegeben durch die Bedingung $0 = \frac{dh}{d\lambda}(\lambda_*) = \nabla f(x + \lambda_* d) d = [Q(x + \lambda_* d) - b]^T d$. Auflösung ergibt

   $$\lambda_* = -\frac{(Qx - b)^T d}{d^T Q d}$$

   .

Damit läßt sich das Gradientenverfahren für quadratische Funktionen explizit vereinfachen.

Algorithmus 3.2 (Steilster Abstieg für quadratische Funktionen)

1. *Wähle* $x_0 \in \boldsymbol{R}^n$.

2. *Für* $k = 0, 1, \ldots$*:*

$$d_k := -(Qx_k - b); \quad x_{k+1} := x_k + \frac{d_k^T d_k}{d_k^T Q d_k} d_k.$$

Beispiele zum Konvergenzverhalten

Die Konvergenzeigenschaften sind je nach Form der quadratischen Funktion sehr unterschiedlich:

1. Für $f(x) = \frac{1}{2}x^T x$ findet man das Optimum im ersten Iterationsschritt, denn $x_1 = x_0 + \frac{x_0^T x_0}{x_0^T x_0}(-x_0) = 0$.

2. Für $f(x) = x^2 + 100y^2$ konvergiert die Folge der Iterationspunkte linear und es gilt $f(x_{k+1}) \approx 0{,}63 \cdot f(x_k)$ für alle $k = 0, 1, \ldots$.

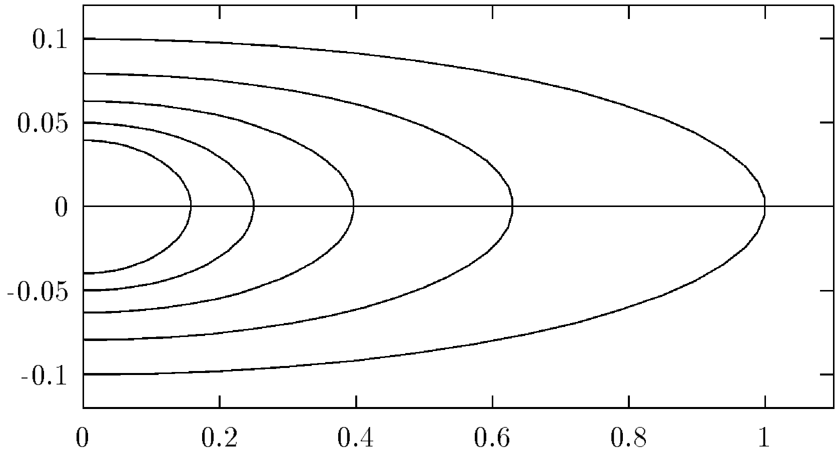

Abbildung III.15: Konvergenz für quadratische Funktionen

Die langsame Konvergenz beim zweiten Beispiel erweist sich als typisch für das Gradientenverfahren. Eine entsprechende Abschätzung liefert der folgende Satz. Hier verwenden wir die bekannte Tatsache, daß eine positiv definite $n \cdot n$-Matrix eine Basis aus paarweise orthonormalen Eigenvektoren zu ihren n positiven reellen Eigenwerten besitzt.

Satz 3.2 *Sei f quadratisch, seien $0 < \mu_1 \leq \ldots \leq \mu_n$ die Eigenwerte von Q, und sei x_* Minimum von f. Dann gilt für die mit Algorithmus 3.2 erzeugte Folge*

$$f(x_{k+1}) - f(x_*) \leq \frac{(\frac{\mu_n}{\mu_1} - 1)^2}{(\frac{\mu_n}{\mu_1} + 1)^2}(f(x_k) - f(x_*)).$$

Beweis. Modulo einer Koordinatenverschiebung können wir oBdA annehmen (siehe Eigenschaften quadratischer Funktionen 3.), daß $b = 0$ und daher $f(x_*) = 0$, $x_* = 0$ und $\nabla f(x) = x^T Q = (Qx)^T$ gelten. Aus der globalen Minimierung in den untersuchten Richtungen folgt:

$$(\dagger) \quad f(x_{k+1}) \leq f(x_k - \lambda Q x_k) \quad \forall\, \lambda \geq 0.$$

Die paarweise orthonormalen Eigenvektoren $a_1, a_2, \ldots, a_n$ zu den Eigenwerten $\mu_1, \mu_2, \ldots, \mu_n$ der reell symmetrischen, positiv definiten Matrix Q bilden eine Orthonormalbasis.

Ist $x_k := \sum_{i=1}^n \alpha_i a_i$ die Koordinatendarstellung von x_k, so ist der Funktionswert gegeben durch

$$(*) \quad f(x_k) = \frac{1}{2} x_k^T Q x_k = \frac{1}{2} \sum_{i=1}^n (\alpha_i)^2 \mu_i.$$

Die Koordinatendarstellung des Argumentes in (†) ist

$$x_k - \lambda Q x_k = \sum_{i=1}^n \alpha_i a_i - \lambda \sum_{i=1}^n \alpha_i Q a_i = \sum_{i=1}^n \alpha_i (1 - \lambda \mu_i) a_i.$$

Daher ergibt sich analog zu (∗) für spezielle Wahl der Schrittweite $\bar{\lambda} = \frac{2}{\mu_n + \mu_1}$ der Funktionswert

$$f(x_k - \bar{\lambda} Q x_k) = \frac{1}{2} \sum_{i=1}^n \alpha_i^2 (1 - \bar{\lambda}\mu_i)^2 \mu_i = \frac{1}{2} \sum_{i=1}^n \alpha_i^2 \left[\frac{\mu_n + \mu_1 - 2\mu_i}{\mu_n + \mu_1} \right]^2 \mu_i.$$

Da $\max_i \, (\mu_n + \mu_1 - 2\mu_i)^2 = (\mu_n - \mu_1)^2$, gilt

$$f(x_{k+1}) \leq f(x_k - \bar{\lambda} Q x_k) \leq \left(\frac{\mu_n - \mu_1}{\mu_n + \mu_1} \right)^2 f(x_k).$$

□

Bemerkung (Konvergenzverhalten):

1. Das Verfahren ist bezogen auf die Funktionswerte $r_k := f(x_k) - f(x_*)$ linear konvergent, da $\left(\frac{\mu_n - \mu_1}{\mu_n + \mu_1}\right)^2 < 1$.

2. Es ist praktisch unbrauchbar, falls $\mu_n \gg \mu_1$, da für $r := \frac{\mu_n}{\mu_1} \to \infty$ folgt $(\frac{r-1}{r+1})^2 \to 1$.

3. Die Konvergenzanalyse überträgt sich lokal in der Nähe eines Minimums auf den nichtquadratischen Fall. Mit $x_k \to x_*$ gilt $F(x_k) \to F(x_*) =: Q$, so daß die Eigenwerte von $F(x_*)$ das Konvergenzverhalten nahe x_* bestimmen.

4. Ein weiterer Nachteil des Verfahrens ist seine Skalierungsabhängigkeit. *Skalierung der Koordinaten*, gegeben etwa durch $y_i := \alpha_i x_i$ für $i = 1, \ldots, n$, ist eine spezielle lineare Koordinatentransformation. Nach Koordinatenwechsel $y := H^{-1} x$ mit regulärer Matrix H wird die Funktion g mit $g(y) := f(Hy)$ minimiert. Die zugehörigen Gradienten sind gegeben durch $\nabla f(Hy) H$. In y-Koordinaten ist die neue Suchrichtung $-H^T \nabla f(Hy)^T$. In x-Koordinaten lautet die so gefundene neue Suchrichtung daher $-HH^T \nabla f(x)^T$. Nicht nur die Suchrichtung ist sehr abhängig von den gewählten Koordinaten, auch die Eigenwerte können sich entscheidend ändern, es gilt $G(y) = H^T F(Hy) H$. Durch Skalierung der Koordinaten werden insbesondere die alten Eigenwerte μ_i einer quadratischen Funktion in die Eigenwerte $\frac{\mu_i}{\alpha_i^2}$ transformiert.

Quadratische lokale Approximation

Wir betrachten nun nochmals die abgebrochene Taylorreihe als quadratische Approximation der zu minimierenden Funktion, wobei wir jetzt bei $\bar{x}$, dem momentanen Iterationspunkt, entwickeln:

$$f(x) \approx f(\bar{x}) + \nabla f(\bar{x})(x-\bar{x}) + \frac{1}{2}(x-\bar{x})^T F(\bar{x})(x-\bar{x}) =: Q(x)$$

Falls $F(\bar{x})$ positiv definit ist, so ist durch $x_* - \bar{x} = F(\bar{x})^{-1}[-\nabla f(\bar{x})^T]$ das globale Minimum x_* der quadratischen Funktion $Q(x)$ gegeben. Für nichtlineare C_2-Funktionen kann man x_* als neuen Iterationspunkt betrachten. Das daraus abgeleitete Verfahren ist das

Algorithmus 3.3 (Newton-Verfahren)

1. *Wähle* $x_0 \in \boldsymbol{R}^n$.
2. *Für* $k = 0, 1, \ldots$*:*

$$F(x_k)d_k := -\nabla f(x_k)^T; \quad x_{k+1} := x_k + d_k.$$

Falls $F(x_k)$ positiv definit ist, so liegt wegen

$$\nabla f(x_k)d_k = -\nabla f(x_k)F^{-1}(x_k)\nabla f(x_k)^T$$

eine Abstiegsrichtung vor. Im allgemeinen wird dies aber für nichtlineare Funktionen nicht gelten, so daß globale Konvergenz nur unter zusätzlichen Annahmen zu erzwingen ist. Zur Verbesserung des Verfahrens kann man auch statt der festen Schrittweite $\lambda = 1$ eine Schrittweitensteuerung einbauen. Wir untersuchen zunächst das lokale Konvergenzverhalten. Zur Formulierung eines typischen Konvergenzsatzes benötigen wir Matrixnormen.

Matrixnormen

Im folgenden ist mit $\| \cdot \|$ stets die Euklidische Norm $\| \cdot \|_2$ bezeichnet. Eine aus der euklidischen Norm abgeleitete Matrixnorm ist gegeben durch

$$\| A \| := \max_{\|x\|=1} \| Ax \| .$$

Offenbar gilt $\| A \| = \max_{x \neq 0} \sqrt{\frac{x^T A^T A x}{x^T x}} = \sqrt{\lambda_{\max}(A^T A)}$, wobei $\lambda_{\max}$ der maximale Eigenwert von $A^T A$ ist.

Diese Matrixnorm ist *verträglich* mit der euklidischen Norm, d.h. es gilt

$$\| Ax \| \leq \| A \| \cdot \| x \|,$$

und sie ist *submultiplikativ*, d.h. es gilt

$$\| AB \| \leq \| A \| \cdot \| B \| .$$

Auf diese kanonische Weise kann man zu jeder Vektornorm passende Matrixnormen ableiten.

Satz 3.3 (Lokal quadratische Konvergenz des Newton-Verfahrens) *Sei $X \subseteq \boldsymbol{R}^n$ offen, $\bar{x} \in X$ und sei $f \in C_2(X)$ mit*

1. *$\nabla f(\bar{x}) = 0$, $F(\bar{x})$ regulär,*
2. *es gibt ein $\mu \geq 0$ und eine Umgebung $U(\bar{x}) \subseteq X$, so daß für alle $x \in U(\bar{x})$ gilt*

$$\| F(x) - F(\bar{x}) \| \leq \mu \| x - \bar{x} \| .$$

Dann gibt es ein $\epsilon > 0$, so daß das Newtonverfahren in $U_\epsilon(\bar{x})$ mindestens quadratisch gegen $\bar{x}$ konvergiert.

Beweis. Wähle $U(\bar{x})$ in (2.) klein genug, so daß für alle $x \in U(\bar{x})$ die Norm der Inversen beschränkt, d.h. $\| F^{-1}(x) \| \leq C$ für eine Konstante $C > 0$, und die Iterationsfunktion $g(x) := x - F^{-1}(x) \nabla f(x)^T$ wohldefiniert ist. Um $g(x) - \bar{x}$ abschätzen zu können, untersuchen wir zunächst den Gradienten. Da $\nabla f(\bar{x}) = 0$, gilt

$$-\nabla f(x)^T = \underbrace{-\nabla f(x)^T + F(\bar{x})x}_{=:\,h(x)} + \underbrace{\nabla f(\bar{x})^T - F(\bar{x})\bar{x}}_{=-h(\bar{x})} - F(\bar{x})(x - \bar{x}).$$

Die Ableitung der hier definierten Funktion h ist $-F(x) + F(\bar{x})$. Aus dem Mittelwertsatz erhalten wir daher die Abschätzung

$$\| h(x) - h(\bar{x}) \| \leq \| x - \bar{x} \| \cdot \sup_{y \in [x, \bar{x}]} \| F(y) - F(\bar{x}) \| \leq \mu \| x - \bar{x} \|^2 .$$

Nun gilt

$$\begin{aligned} F(x)(g(x) - \bar{x}) &= \underbrace{F(x)(g(x) - x)}_{=-\nabla f(x)^T} + F(x)(x - \bar{x}) \\ &= h(x) - h(\bar{x}) + [F(x) - F(\bar{x})](x - \bar{x}). \end{aligned}$$

Multiplikation mit F^{-1}, Anwendung der vorbereiteten Abschätzung und Ausnutzen der Dreiecksungleichung liefern

$$\begin{aligned} \| g(x) - \bar{x} \| &\leq \| F^{-1}(x) \| \cdot [\, \| h(x) - h(\bar{x}) \| + \| F(x) - F(\bar{x}) \| \cdot \| x - \bar{x} \| \,] \\ &\leq C \cdot 2\mu \| x - \bar{x} \|^2 . \end{aligned}$$

Wähle $\epsilon > 0$ hinreichend klein, so daß $0 \leq \epsilon \cdot 2\mu C =: \alpha < 1$ und $U_\epsilon(\bar{x}) \subseteq U(\bar{x})$. Wähle $x_0 \in U_\epsilon(\bar{x})$. Damit erhalten wir

$$\| x_1 - \bar{x} \| = \| g(x_0) - \bar{x} \| \leq 2\mu C \| x_0 - \bar{x} \|^2 < \alpha \| x_0 - \bar{x} \| < \alpha\epsilon,$$

woraus per Induktion ("iterativ") folgt

$$\| x_k - \bar{x} \| < \alpha^k \epsilon \quad \forall\, k \geq 0.$$

Also gilt $x_k \to \bar{x}$. Diese Folge ist mindestens quadratisch konvergent, denn

$$\| x_{k+1} - \bar{x} \| = \| g(x_k) - \bar{x} \| \leq 2\mu C \| x_k - \bar{x} \|^2 .$$

□

Das lokal quadratisch und damit schnell konvergente Newton-Verfahren verlangt einen wesentlich höheren Aufwand als das Gradientenverfahren. Pro Iteration muß neben dem Gradienten die Matrix $F(x)$ berechnet oder approximiert, und das Gleichungssystem $F(x)d := \nabla f(x)^T$ gelöst werden. Darüberhinaus bleibt, wie oben bereits angesprochen, die globale Konvergenz unsicher. Beide Schwierigkeiten versucht man mit geeigneten Modifikationen aus dem Weg zu räumen. Dazu sei G eine stetige Matrixfunktion, die durch $x \mapsto G(x)$ definiert ist, wobei $G(x)$ reell, symmetrisch und positiv definit für alle x.

Algorithmus 3.4 (Modifizierte Newton-Verfahren)

1. *Wähle* $x_0 \in \boldsymbol{R}^n$.
2. *Für* $k = 0, 1, \ldots$:

$$G(x_k)d_k := -\nabla f(x_k)^T; \quad x_{k+1} := x_k + \lambda_k d_k.$$

Bemerkung:

1. Die Schrittweite λ_k wird mit Hilfe geeigneter eindimensionaler Minimierungsverfahren gewählt.

2. Die Matrixfunktion wird als Näherung von F gewählt, d.h. $G(x) \approx F(x)$. Für $G \equiv I$ bzw. $G \equiv F$ erhält man formal das Gradienten-, bzw. das Newton-Verfahren, vorausgesetzt F ist positiv definit. Das "vereinfachte" Newton-Verfahren erhält man für $G(x) \equiv F(x_0)$. Praktikabel sind flexible Verfahren, bei denen etwa zunächst Gradientenschritte und dann für einen Zyklus von Schritten vereinfachte Newton-Iterationen durchgeführt werden. Verschlechtert sich das Konvergenzverhalten im Zyklus, wird F erneut approximiert. Wird quadratische Konvergenz erkennbar, wird die Schrittweite auf $\lambda = 1$ fixiert.

3. Ist $f \in C_2(X)$, und $F(x)$ positiv definit auf X, so ist f streng konvex. Dann besitzt f ein eindeutig bestimmtes strenges Minimum, etwa $\bar{x}$. Bei $\bar{x}$ konvergiert das Newton-Verfahren lokal quadratisch.

4. Modifizierte Newton-Verfahren sind stets Abstiegsverfahren, da mit G auch G^{-1} positiv definit sind, woraus $\nabla f(x)d = -\nabla f(x)G(x)^{-1}\nabla f^T(x) < 0$ folgt. Bestimmt man die Schrittweite mit Armijo-Suche und ist die Niveaumenge $N(x_0) := \{x \mid f(x) \leq f(x_0)\}$ beschränkt, so erhält man globale Konvergenz wie in Satz 3.1.

4 Konjugierte Richtungen

Auch in diesem Abschnitt werden wir zunächst wieder quadratische Minimierungsprobleme untersuchen. Sei also Q eine reelle, symmetrische, positiv definite Matrix und sei $f : \boldsymbol{R}^n \to \boldsymbol{R}$ definiert durch $f(x) := \frac{1}{2}x^T Qx - b^T x$ mit Minimum $x_* = Q^{-1}b$ und Ableitung $\nabla f(x)^T = Qx - b =: g(x)$.

Definition (Konjugierte Richtungen): Sei $k \in \boldsymbol{N}$. Vektoren $d_0, \ldots, d_k \in \boldsymbol{R}^n \setminus \{0\}$ heißen *konjugiert*, falls $d_i^T Qd_j = 0$ für alle $0 \le i < j \le k$. Zur Abkürzung sei $\rho_i := d_i^T Qd_i$ für $0 \le i \le k$.

Konjugierte Vektoren sind immer linear unabhängig, denn es gilt für α mit $\alpha_i \neq 0$: $d_i^T Q \left(\sum \alpha_j d_j\right) = \rho_i \alpha_i \neq 0$.

Lemma 4.1 *Sind $d_0, \ldots, d_{n-1}$ konjugiert, so gilt*

$$Q^{-1} = D := \sum_{k=0}^{n-1} \frac{1}{\rho_k} d_k d_k^T, \quad x_* = \sum_{k=0}^{n-1} \frac{1}{\rho_k} d_k d_k^T b = \sum_{k=0}^{n-1} \frac{d_k^T b}{\rho_k} d_k$$

Beweis. Die Formel für Q^{-1} impliziert die für x_*. Es genügt daher zu zeigen, daß für alle $x \in \boldsymbol{R}^n$ gilt: $DQx = x$. Für $x = \sum_{i=0}^{n-1} \alpha_i d_i$ findet man $d_k^T Qx = \sum_{i=0}^{n-1} \alpha_i d_k^T Qd_i = \alpha_k \rho_k$. Daher gilt

$$DQx = \sum_{k=0}^{n-1} \frac{1}{\rho_k} d_k d_k^T Qx = \sum_{k=0}^{n-1} \frac{1}{\rho_k} d_k (\alpha_k \rho_k) = x$$

$\square$

Für bekannte konjugierte Richtungen $d_0, \ldots, d_{n-1}$ bietet es sich daher an, sukzessive in diesen Richtungen die Funktion f zu minimieren.

Algorithmus 4.1 (Verfahren konjugierter Richtungen)

1. *Wähle $x_0 \in \boldsymbol{R}^n$.*

2. *Für $k = 0, 1, \ldots, n-1$; $d_0, \ldots, d_{n-1}$ konjugiert:*

$$\lambda_k := -\frac{1}{\rho_k}(Qx_k - b)^T d_k; \quad x_{k+1} := x_k + \lambda_k d_k.$$

Die gewählte Schrittweite λ_k ist optimal. Als Abbruchkriterium kann man das Verschwinden des Gradienten, d.h. $g_k := g(x_k) \approx 0$, testen.

Der von Vektoren $p_0, \ldots, p_k \in \boldsymbol{R}^n$ aufgespannte Unterraum wird in üblicher Weise bezeichnet:

$$< p_0, \ldots, p_k > := \{\sum_{i=0}^{k} \alpha_i p_i \mid \alpha_i \in \boldsymbol{R}; i = 0, \ldots, k\}.$$

Lemma 4.2 *Für $k = 1, \ldots, n$ gilt im Verfahren konjugierter Richtungen*

1. *g_k orthogonal auf $T_k := < d_0, \dots, d_{k-1} >$,*

2. *x_k ist globales Minimum von f auf $T_k + x_0$,*

3. *$x_* = x_n$.*

Beweis. Aus (1.) folgt (2.), da $g_k^T = \nabla f(x_k)$. Aus (2.) folgt (3.), da $T_n + x_0 = \boldsymbol{R}^n$. Zu zeigen ist also nur noch (1.).

Für $i < k$ gilt $x_k = x_i + \sum_{j=i}^{k-1} \lambda_j d_j$, also auch $Qx_k - b = Qx_i - b + \sum_{j=i}^{k-1} \lambda_j Q d_j$. Multiplikation von links mit d_i^T ergibt

$$d_i^T g_k = d_i^T g_i + \sum_{j=i}^{k-1} \lambda_j d_i^T Q d_j = d_i^T g_i + \lambda_i \rho_i .$$

Wegen $\lambda_i = -\frac{1}{\rho_i} g_i^T d_i$ verschwindet die rechte Seite für alle $i < k$, d.h. $g_k \perp T_k$. □

Lemma 4.3 (Gram-Schmidt-Orthogonalisierung) *Sind für ein r mit $1 \leq r \leq n$ die Vektoren $p_0, \dots, p_{r-1}$ linear unabhängig, so sind die durch $d_0 := p_0$, und für $k = 0, \dots, r-2$ durch:*

$$d_{k+1} := p_{k+1} - \sum_{i=0}^{k} \frac{p_{k+1}^T Q d_i}{\rho_i} d_i$$

definierten Vektoren konjugiert und es gilt

$$< p_0, \dots, p_{r-1} > = < d_0, \dots, d_{r-1} > .$$

Beweis. Der Beweis wird induktiv über r geführt. Der Induktionsanfang ist trivial wegen $< d_0 > = < p_0 >$. Für den Induktionsschluß $r \to r+1$ sei $j < r$. Dann gilt

$$d_j^T Q d_r = d_j^T Q p_r - \sum_{i=0}^{r-1} \frac{p_r^T Q d_i}{\rho_i} d_j^T Q d_i = 0,$$

d.h. d_r ist konjugiert zu d_j für $j < r$. Nun gilt nach Induktionsannahme $< d_0, \dots, d_{r-1} > = < p_0, \dots, p_{r-1} >$, und daher nach Rekursionsformel

$$d_r \in < d_0, \dots, d_{r-1}, p_r > = < p_0, \dots, p_r > .$$

Damit folgt $< d_0, \dots, d_r > \subseteq < p_0, \dots, p_r >$, wobei "=" gelten muß, da die d_i linear unabhängig sind. □

Mit Hilfe der Gram-Schmidt-Orthogonalisierung kann man also aus jeder Basis eine Basis aus konjugierten Richtungen erzeugen.

Wahl der p_k

Die Freiheit bei der Wahl der Basis nutzt man aus, um die p_k sukzessive so zu bestimmen, daß $g_k^T d_k$ jeweils möglichst klein ist. Zur Normierung dieser Größe sei zunächst $\| p_k \|= 1$ festgelegt. Im ersten Schritt ist der schon aus den Gradientenverfahren bekannte steilste Abstieg minimal, d.h. $d_0 := p_0 := -\frac{g_0}{\|g_0\|}$.

In den Iterationen für $0 \leq k \leq n-2$ ist die Wahl der zu minimierenden Größe eingeschränkt durch die Rekursion. Es gilt

$$g_{k+1}^T d_{k+1} = g_{k+1}^T p_{k+1} - \sum_{i=0}^{k} \frac{p_{k+1}^T Q d_i}{\rho_i} g_{k+1}^T d_i,$$

wobei die letzte Summe wegen $g_{k+1}^T d_i = 0$ verschwindet. Daher wird der Ausdruck minimal für $p_{k+1} := -\frac{g_{k+1}}{\|g_{k+1}\|}$. Da im allgemeinen $p_{k+1} \neq d_{k+1}$, ist im allgemeinen in den Iterationen kein steilster Abstieg mehr möglich. Ohne Berücksichtigung der Normierung kann man die Richtungen also durch

(4.1) $$p_k := -g_k \text{ für } k = 0, \ldots, n-1$$

angeben. Sind diese Richtungen nun auch noch linear unabhängig? Dies kann natürlich nur gelten, wenn keiner der Gradienten verschwindet, da sonst das Verfahren vorzeitig mit der Optimallösung abbricht.

Lemma 4.4 *Verwendet man sukzessive die durch Lemma 4.3 aus den Richtungen in Gleichung (4.1) erzeugten Vektoren im Verfahren 4.1 und verschwindet in keinem Schritt der Gradient, so gilt:*

1. $g_0, \ldots, g_{n-1}$ *paarweise orthogonal,*
2. $< g_0, \ldots, g_k > = < d_0, \ldots, d_k >$ *für* $k = 0, \ldots, n-1$,
3. $d_{k+1} = -g_{k+1} + \frac{1}{\rho_k}(g_{k+1}^T Q d_k) d_k$.

Beweis. (1.) und (2.) lassen sich simultan zeigen: da $g_0 = -d_0$ folgt aus Lemma 4.2 $g_1 \perp g_0$, und daher aus Lemma 4.3 $< g_1, g_0 > = < d_1, d_0 >$. Aus Lemma 4.2 folgt wieder $g_2 \perp < g_1, g_0 >$ usw. Sei $0 \leq j \leq n-2$. Es gilt

(4.2) $$g_{j+1} - g_j = Q(x_{j+1} - x_j) = \lambda_j Q d_j,$$

da nach 2.) $g_{j+1} \neq g_j$ und somit $\lambda_j \neq 0$ gilt. Die Orthogonalität der Gradienten liefert daher für $0 \leq j < k \leq n-2$

$$0 = g_{k+1}^T (g_{j+1} - g_j) = \lambda_j g_{k+1}^T Q d_j.$$

Wegen $\lambda_j \neq 0$ gilt $g_{k+1}^T Q d_j = 0$, woraus folgt

$$d_{k+1} = -g_{k+1} - \sum_{i=0}^{k} \frac{-g_{k+1}^T Q d_i}{\rho_i} d_i = -g_{k+1} + \frac{g_{k+1}^T Q d_k}{\rho_k} d_k.$$

□

Nun können wir für quadratische Funktionen das vollständige Verfahren von Hesteness, Stiefel (1952) beschreiben.

Algorithmus 4.2 (Konjugierte Gradienten bei quadratischen Funktionen)

1. Wähle $x_0 \in \boldsymbol{R}^n$; $\quad g_0 := Qx_0 - b$.
   Falls $g_0 = 0$, dann stop.
   $d_0 := -g_0$; $\quad k := 0$.
2. $\lambda_k := -\frac{g_k^T d_k}{\rho_k}$; $\quad x_{k+1} := x_k + \lambda_k d_k$.
3. $g_{k+1} := g_k + \lambda_k Q d_k$;
   falls $g_{k+1} = 0$, dann stop.
4. $\beta_k := \frac{g_{k+1}^T Q d_k}{\rho_k}$; $\quad d_{k+1} := -g_{k+1} + \beta_k d_k$;
   $k := k+1$; $\quad$ gehe nach 2.

Bemerkung (Zur Konvergenz):

1. Man erzeugt lokal unter allen konjugierten Abstiegsrichtungen die mit dem steilsten Abstieg.
2. Bei etwa doppeltem Aufwand pro Iteration im Vergleich zum Gradientenverfahren erhält man bei quadratischen Funktionen Konvergenz in n Iterationen.
3. Rundungsfehler führen allerdings auf $g_n \neq 0$. Dann kann man für $k \geq n$ weitere Iterationen anschließen und abbrechen, wenn $\| g_k \| \leq \epsilon$.

Im folgenden Lemma werden einige äquivalente Formeln bereitgestellt, die teilweise die notwendigen Berechnungen vereinfachen. Im wesentlichen dienen sie allerdings zur Modifikation des Verfahrens bei nichtquadratischen Funktionen.

Lemma 4.5 (Rechenalternativen) *Für $k = 0, 1, \ldots, n-1$ gilt:*

1. $\lambda_k = \frac{1}{\rho_k} g_k^T g_k$,

2. $\beta_k = \frac{1}{\rho_k} g_{k+1}^T Q d_k = \frac{(g_{k+1} - g_k)^T g_{k+1}}{g_k^T g_k} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$.

Beweis. zu 1.): Im Verfahren konjugierter Gradienten wurden bereits $d_k = -g_k + \beta_{k-1} d_{k-1}$ und $\lambda_k = -\frac{g_k^T d_k}{\rho_k}$ benutzt. Also gilt

$$-g_k^T d_k = g_k^T g_k - \beta_{k-1} g_k^T d_{k-1} = g_k^T g_k.$$

zu 2.): Unter Verwendung der Gleichung (4.2) finden wir

$$g_{k+1}^T Q d_k = \frac{1}{\lambda_k} g_{k+1}^T (g_{k+1} - g_k) = \frac{\rho_k}{g_k^T g_k} g_{k+1}^T (g_{k+1} - g_k) = \rho_k \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}.$$

□

Für nichtquadratische Funktionen ersetzt man $g_k \sim \nabla f(x_k)^T$ und $Q \sim F(x_k)$ und berechnet x_{k+1} mit näherungsweise optimaler Schrittweite in Abstiegsrichtung. Die verschiedenen Formeln für β_k liefern dann verschiedene bekannte Verfahren für nichtquadratische Funktionen:

1. **Daniel, 1967:** $\beta_k := \frac{g_{k+1}^T F(x_{k+1}) d_k}{d_k^T F(x_{k+1}) d_k}$,
2. **Polak-Ribière, 1969:** $\beta_k := \frac{(g_{k+1} - g_k)^T g_{k+1}}{g_k^T g_k}$,
3. **Fletcher-Reeves, 1964:** $\beta_k := \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$.

Algorithmus 4.3 (Konjugierte Gradienten bei nichtlinearen Funktionen)

1. Wähle $x_0 \in \boldsymbol{R}^n$; $\quad g_0 := \nabla f(x_0)$.
   Falls $g_0 = 0$, dann stop.
   $d_0 := -g_0$; $\quad k := 0$.

2. Bestimme (optimale) Schrittweite λ_k in Richtung d_k;
   $x_{k+1} := x_k + \lambda_k d_k$.

3. $g_{k+1} := \nabla f(x_{k+1})$;
   falls $g_{k+1} = 0$, dann stop;
   falls $k = n - 1$, dann gehe nach 6.

4. Berechne β_k; $\quad d_{k+1} := -g_{k+1} + \beta_k d_k$.

5. Falls $g_{k+1}^T d_{k+1} > 0$, dann gehe nach 6.;
   $k := k + 1$; $\quad$ gehe nach 2.

6. $x_0 := x_{k+1}$; $\quad d_0 := -g_{k+1}$;
   $k := 0$; $\quad$ gehe nach 2.

Konvergenz

1. Globale Konvergenz wird unter den üblichen Voraussetzungen durch den nach n Iterationen eingeschobenen neuen Gradientenschritt erzwungen. Dieser startet einen neuen Zyklus von n Iterationen mit konjugierten Gradienten.

2. Bei geeigneten Voraussetzungen wird quadratische Konvergenz wie beim Newtonverfahren erreicht. Dabei entsprechen n Iterationen einem Newtonschritt.

Index

adjazent, 60
Algorithmus
  Abstiegsverfahren, 136
  Bellman-Ford-Verfahren, 88
  Bestimmung maximaler Branchings, 80
  Breadth-First-Search, 82
  Davis, Swann und Campey, 129
  Depth-First-Search, 83
  Dijkstra-Verfahren, 86
  Fibonacci-Suche, 118
  Goldener Schnitt, 116
  Gradientenverfahren, 139
    für quadratische Funktionen, 140
  Kleene-Verfahren, 95
  konjugierte Gradienten bei nichtlin. Funktionen, 150
  konjugierte Gradienten bei quadratischen Funktionen, 149
  konjugierte Richtungen, 146
  Newton-Verfahren, 121
    mehrdimensional, 143
    modifiziert, 145
  Newton-Verfahren mit finiten Differenzen, 125
  revidiertes Simplexverfahren, 26
  Simplexverfahren, 20
  Verfahren von Floyd, 98
  Verfahren von Kruskal, 73
  Verfahren von Prim, 71
Armijo-Suche, 137
Ausengrad, 66

Basis, 14
Basislösung, 14
  zulässige Basislösung, 14
Basisvariable, 14
Baum, 63
  gerichteter Baum, 64
Blands Regel, 37
Blaue Regel, 68
Branching, 75

Darstellungskoeffizienten, 15
duales Problem, 49

Ecken, 10
  entartete Ecke, 10
Einselement, 91
Expandieren, 79

Gerüst, 63
  gerichtetes Gerüst, 65
  minimales Gerüst, 68
Grad, 66
Gradientenungleichung, 135
Graph, 60
  einfacher Graph, 60
  gerichteter Graph, 60
  stark zusammenhängender Graph, 63
  ungerichteter Graph, 60
  zusammenhängender Graph, 63

Innengrad, 66
Innengrad-Bedingung, 75
Inzidenzvektor, 68

konjugierte Richtungen, 146
Konvergenz
  lineare Konvergenz, 124
  quadratische Konvergenz, 124
  superlineare Konvergenz, 124
Konvergenzfaktor, 124

Konvergenzgeschwindigkeit, 124
Konvergenzordnung, 124
konvex, 121
konvexe Hülle, 131
Kreis, 62

lexikographische Ordnung, 34

M-Methode, 31
Matching, 99
    perfektes Matching, 99
Matrixnorm, 143
Mehrphasenmethode, 29
Minimum
    globales Minimum, 129
    lokales Minimum, 129
    streng lokales Minimum, 130
    streng globales Minimum, 130
Monoid, 84

Nichtbasis, 14
Nichtbasisvariable, 14
Nullelement, 91

optimales Paar, 50

Polyeder, 10
    Seiten von Polyedern, 10
    spitzer Polyeder, 10
positiv definit, 131
positiv semidefinit, 131
primales Problem, 49
pseudokonkav, 136
pseudokonvex, 136

reduzierte Kostenkoeffizienten, 16
Richtungsableitung, 133
Rote Regel, 68
Rundreiseproblem, 105

Schlinge, 60
Schlupfvariable, 12
Schnitt, 63
Schrumpfen, 78
Semiring, 91
    abgeschlossener Semiring, 92
    Bool'scher Semiring, 92
Skalierung, 142
stabiler Zustand, 46
Standardform, 14
stochastischer Zustand, 46
submultiplikativ, 143

Teilgraph, 63
    kritischer Teilgraph, 76
transitive Hülle, 91

unimodal, 114
Untergraph, 63

Wald, 68
    aufspannender Wald, 68
Weg, 62
    einfacher Weg, 62
    Euler'scher Weg, 65
    Länge des Weges, 62
Wurzel, 64

zulässige Richtung, 130
Zuordnung, 99
Zusammenhangskomponenten, 63
Zweiphasenmethode, 26

Literaturverzeichnis

[1] V. Chviátal. *Linear Programming.* W.H.Freeman and Company, New York, 1983.

[2] T. H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* MIT Press, Cambridge, Ma., 1990.

[3] S. Even. *Graph Algorithms.* Computer Science Press, 1979.

[4] R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, Chichester, 1987.

[5] M. Gondran and M. Minoux. *Graphs and Algorithms.* John Wiley & Sons, Chichester/New York, 1984.

[6] M. Groetschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer, Berlin, 1988.

[7] R. Horst. *Nichtlineare Optimierung.* Carl Hanser, Muenchen/Wien, 1979.

[8] R. Horst and H. Tuy. *Global Optimization.* Springer, Berlin, second edition, 1992.

[9] Jr. J. E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall Inc., 1983. Prentice-Hall Series in Computational Mathematics.

[10] D. Jungnickel. *Graphen, Netzwerke und Algorithmen.* Bibliographisches Institut, Mannheim, 1987.

[11] D.E. Knuth. *The Art of computer programming, Vol. 1: Fundamental Algorithms.* Series in Computer Science and Information Processing. Addison-Wesley, Reading, 1969.

[12] E.L. Lawler, J.K. Lenstra, A.H.G Rinnooy Kan, and D.B. Shmoys, editors. *The Travelling Salesman Problem.* John Wiley & Sons, Manchester/New York, 1985.

[13] Eugene L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart and Winston, New York, 1976.

[14] J. Leeuven, v. *Algorithms and Complexity.* Elsevier, Amsterdam, 1990.

[15] T. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout.* Teubner, Stuttgart, 1990.

[16] K. Mehlhorn. *Graph Algorithms and NP-Completeness*, volume 2 of *Data Structures and Algorithms.* Springer, Berlin, 1984.

[17] K. Mehlhorn. *Multi-dimensional Searching and Computational Geometry*, volume 3 of *Data Structures and Algorithms.* Springer, Berlin, 1984.

[18] M. Minoux. *Mathematical Programming. Theory and Algorithms.* John Wiley & Sons, Chichester, 1986.

[19] B. M. E. Moret and H. D. Shapiro. *Algorithms from P to NP.* Benjamin Cummings, Redwood City, 1991.

[20] K.G. Murty. *Linear Programming.* John Wiley & Sons, New York/Chichester, 1983.

[21] K.G. Murty. *Linear Complementarity, Linear and Nonlinear Programming.* Heldermann, Berlin, 1988.

[22] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley & Sons, New York, 1988.

[23] G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd. *Optimization.* North-Holland, Amsterdam, 1989.

[24] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization (Algorithms and Complexity).* Prentice-Hall, New Jersey, 1982.

[25] R. Gary Parker and Ronald L. Rardin. *Discrete Optimization.* Academic Press, Boston, 1988.

[26] H.M. Salkin and K. Mathur. *Foundations for Integer Programming.* North-Holland, New York, 1989.

[27] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, Chichester/New York, 1986.

[28] Peter Spellucci. *Numerische Verfahren der nichtlinearen Optimierung.* Birkhaeuser, Basel, 1993.

[29] J. Stoer. *Einfuehrung in die Numerische Mathematik I.* Springer, Berlin, 1989.

[30] Robert E. Tarjan. Finding optimum branchings. *Networks*, 7:25–35, 1977.

[31] Uwe Zimmermann. *Linear and Combinatorial Optimization in Ordered Algebraic Structures.* North-Holland, Amsterdam, New-York, Oxford, Tokyo, 1981.