Kapitel 5

Numerische Verfahren zur Nichtlinearen Programmierung

Nachdem wir im letzten Kapitel etwas Theorie betrieben haben, wollen wir uns nun den Algorithmen zuwenden. Die zentralen Stichworte in der Nichtlinearen Optimierung sind

- Suchrichtung und
- Schrittweite.

Im letzten Kapitel haben wir schon eine Charakterisierung von guten Suchrichtungen, nämlich die Abstiegsrichtungen, kennengelernt. Deswegen wollen wir uns hier zunächst stärker mit der Schrittweitensteuerung beschäftigen. Ist die Suchrichtung festgelegt, haben wir es mit einem Suchproblem der Funktion $f(p_0+\alpha d) : \mathbb{R}_+ \to \mathbb{R}$ zu tun.

5.1 Das allgemeine Suchverfahren

Sicherlich ist über eine Suche bei allgemeinen Funktionen $f : \mathbb{R} \to \mathbb{R}$, die uns nur durch eine Unterroutine gegeben sind, keine allgemeine Aussage möglich. Bei der Klassifizierung der Suchverfahren beschränken wir uns deshalb auf Funktionen, die ein eindeutiges Minimum haben. Die folgenden Voraussetzungen garantieren eine beweisbar richtige Suche. Die Suchverfahren oder eventuelle Modifikationen kann aber auch anwenden, wenn die Voraussetzungen nicht im strengen Sinne erfüllt sind.

Definition 5.1.1 *Sei $[a, b] \subseteq \mathbb{R}$ ein Intervall und $f : [a, b] \to \mathbb{R}$ eine Funktion. Dann heißt f* strikt unimodal auf [a,b], *wenn f genau ein lokales Minimum in $[a, b]$ hat.*

Proposition 5.1.2 *Sei $[a, b] \subseteq \mathbb{R}$ ein Intervall und $f : [a, b] \to \mathbb{R}$ eine Funktion. Dann ist f strikt unimodal genau dann, wenn für $a \leq x < y \leq b$ und $\lambda \in]0, 1[$ stets gilt:*

$$f(\lambda x + (1 - \lambda)y) < \max\{f(x), f(y)\}.$$

Beweis. Beweis durch Kontraposition. Sei $f : [x, y] \to \mathbb{R}$ eine Funktion und $\lambda \in]0, 1[, p = \lambda x + (1 - \lambda)y$ mit

$$f(p) \geq \max\{f(x), f(y)\}.$$

Wir können also ein lokales Minimum $z_1 \neq p$ von f in $[x, p]$ und ein lokales Minimum $z_2 \neq p$ von f in $[p, y]$ wählen. Da z_1 und z_2 von p verschieden sind, sind sie lokale Minima von f. Somit ist f nicht strikt unimodal. Sei nun umgekehrt $g : [a, b] \to \mathbb{R}$ nicht strikt unimodal, und seien $z_1 < z_2$ lokale Minima. Wir dürfen oBdA. annehmen (Symmetrie!), daß $g(z_1) \leq g(z_2)$. Da z_2 ein lokales Minimum von g ist, gibt es ein $1 > \varepsilon > 0$ mit $g(z_2 + \varepsilon(z_1 - z_2)) \geq g(z_2) = \max\{g(z_1), g(z_2)\}$. Somit ist $g(\varepsilon z_1 + (1 - \varepsilon)z_2) \not< \max\{g(z_1), g(z_2)\}$. □

Beispiel 5.1.3 (konvex quadratische Probleme) *Sei $Q \in \mathbb{R}^{n \times n}$ eine quadratische, symmetrische, positiv definite Matrix, $b, x_0 \in \mathbb{R}^n$ und $d \in \mathbb{R}^n \setminus \{0\}$. Dann ist die Funktion $f : \mathbb{R} \to \mathbb{R}$ definiert durch*

$$f(t) = \frac{1}{2}(x_0 + td)^\top Q(x_0 + td) + b(x_0 + td)$$

strikt unimodal auf $\mathbb{R}$, genauer gilt für $s < t$ und $\lambda \in]0, 1[$:

$$f(\lambda s + (1 - \lambda)t) < \lambda f(s) + (1 - \lambda)f(t) \leq \max\{f(s), f(t)\}.$$

Für die Gültigkeit der ersten Ungleichung in der letzten Zeile sagen wir auch: Die Funktion f ist strikt konvex.

Beweis.

$$\begin{aligned}
& \lambda f(s) + (1 - \lambda)f(t) - f(\lambda s + (1 - \lambda)t) \\
= \quad & \lambda \left(\frac{1}{2}(x_0 + sd)^\top Q(x_0 + sd) + b(x_0 + sd) \right) + (1 - \lambda) \left(\frac{1}{2}(x_0 + td)^\top Q(x_0 + td) + b(x_0 + td) \right) \\
& -\frac{1}{2} \left((x_0 + (\lambda s + (1 - \lambda)t)d)^\top Q(x_0 + (\lambda s + (1 - \lambda)t)d) \right) + b\,(x_0 + (\lambda s + (1 - \lambda)t)d)
\end{aligned}$$

$$
\begin{aligned}
&= \lambda\left(\frac{1}{2}(x_0+sd)^\top Q(x_0+sd)\right)+(1-\lambda)\left(\frac{1}{2}(x_0+td)^\top Q(x_0+td)\right) \\
&\quad -\frac{1}{2}\left(\lambda(x_0+sd)+(1-\lambda)(x_0+td)\right)^\top Q\left((\lambda(x_0+sd)+(1-\lambda)t)d\right) \\
&= \frac{1}{2}\lambda(1-\lambda)\left((x_0+sd)^\top Q(x_0+sd)-(x_0+sd)^\top Q(x_0+td)\right. \\
&\quad \left.-(x_0+td)^\top Q(x_0+sd)+(x_0+td)^\top Q(x_0+td)\right) \\
&= \frac{1}{2}\lambda(1-\lambda)(x_0+sd-x_0-td)^\top Q(x_0+sd-x_0+td) \\
&\overset{Q\,pd}{>} 0
\end{aligned}
$$

□

Bemerkung 5.1.4 *Im allgemeinen heißt eine Funktion* $f : S \to \mathbb{R}$ konvex*, wenn S konvex ist und*

$$f(\lambda x+(1-\lambda)y) \leq \lambda f(x)+(1-\lambda)f(y).$$

Dies ist genau dann der Fall, wenn für der Epigraph der Funktion epi(f) $:= \{(x, y) \in \mathbb{R}^{n+1} \mid x \in \mathbb{R}^n, y \geq f(x)\}$ *eine konvexe Menge ist. Mit Hilfe des Mittelwertsatzes der Differentialrechnung kann man zeigen, daß eine Funktion, bei der die Hesse-Matrix stets positiv definit bzw. positiv semidefinit ist, strikt konvex bzw. konvex ist.*

Bei strikt unimodalen Funktionen können wir bei der Auswertung der Funktion an zwei Punkten $x, y \in]a, b[$ feststellen, in welchem der beiden Intervalle $[a, y]$ oder $[x, b]$ das globale Minimum der Funktion im Intervall $[a, b]$ liegt.

Proposition 5.1.5 *Sei* $f : [a, b] \to \mathbb{R}$ *strikt unimodal und* $a < x < y < b$*. Dann gilt*

a) $f(x) \geq f(y) \Rightarrow \min_{[a,b]} f \in [x, b]$.

b) $f(x) \leq f(y) \Rightarrow \min_{[a,b]} f \in [a, y]$.

Beweis. Aus Symmetriegründen brauchen wir nur eine der beiden Aussagen zu zeigen. Sei also $f(x) \geq f(y)$. Für $z \in [a, x[$ sei $\lambda_z := \frac{y-x}{y-z} \in]0, 1[$. Dann ist $x = \lambda_z z + (1 - \lambda_z)y$ und, da f strikt unimodal ist, gilt $f(x) < \max\{f(z), f(y)\}$. Wegen $f(x) \geq f(y)$ gilt somit für alle $z \in [a, x[: f(z) > f(x)$. □

Aus diesem Ergebnis können sie sich als „Faustregel“ merken: Minimieren konvexer Funktionen über konvexen Mengen ist eine gutartige Aufgabenstellung.

Nach diesen Vorbereitungen sollte unser allgemeines Suchverfahren klar sein.

```
def findmin(f,a,x,b):
  fx=f(x)
  while (b-a)/(abs(b)+abs(a)) >= eps:
    x,y,fx,fy=choosepoint(f,a,x,b,fx)
    if fx >= fy:
      a=x
      x=y
      fx=fy
    else:
      b=y
  else:
    return (a+b)/2
```

Die hier angegeben Formulierung hat den Vorteil, daß die Funktion an jeder Stelle höchstens einmal ausgewertet wird. Der Aufwand in der nichtlinearen Programmierung wird oft mit der Anzahl der Funktionsauswertungen angegeben. So kann die Funktion etwa über eine Suchrichtung und eine mehrdimensionale kompliziertete Funktion gegeben sein.

5.2 Spezielle Suchverfahren

In diesem Abschnitt wollen wir zwei spezielle Varianten (Implementierungen) des allgemeinen Suchverfahrens diskutieren. Gütemaß für ein allgemeines Suchverfahren kann nur die Geschwindigkeit der Reduktion der Intervallänge sein. Ein natürlicher Gedanke ist es, binäre Suche zu implementieren. Man überlegt sich leicht, daß hierbei in zwei Schritten die Intervallänge halbiert wird. Bereits bei der Analyse des Euklidischen Algorithmus hatten wir in den Übungen gesehen, daß mit den Fibonacci-Zahlen eine bessere Abschätzung möglich ist.

Betrachten wir auch hier zwei aufeinanderfolgende Iterationen. Zunächst haben wir $a < x < y < b$ und entfernen entweder $[a, x]$ oder $[y, b]$. Im darauffolgenden Schritt wird ein z in $[x, b]$ bzw. in $[a, y]$ plaziert. Egal wie dies geschieht, es wird, falls im ersten Fall das linke Teilstück des Intervalls entfernt wird stets mindestens $[y, b]$ übrigbleiben bzw. im zweiten Fall mindestens $[a, x]$. Da man bei allgemeinen Funktionen im voraus nicht wissen kann, welche Stücke entfernt werden, haben wir somit gezeigt, daß man in einem Schritt im ungünstigsten Fall nicht mehr entfernen kann, als im nächsten Schritt übrigbleibt. Oder formaler

Proposition 5.2.1 *Sei S ein Suchverfahren und für $k \in \mathbb{Z}_+$*

$$S_k := \max\{\frac{b_k - a_k}{b - a} \mid a_k, b_k \text{ nach } k\text{-ter Iteration bei strikt unimodalen } f\}$$

dann gilt

$$S_k \leq S_{k+1} + S_{k+2}.$$

Beweis. Wir haben eben gezeigt, daß $(S_k - S_{k+1})(b-a) \leq \max\{x-a, b-y\}$ und $S_{k+2}(b-a) \geq \max\{x-a, b-y\}$. Also gilt $S_k - S_{k+1} \leq S_{k+2}$. □

Die Konsequenz der letzten Proposition ist, daß sich mittels der Fibonaccizahlen ein beweisbar bestes Suchverfahren konstruieren läßt. Sei dazu F_k die k-te Fibonaccizahl, also $F_0 = F_1 = 1$ und $F_{n+2} = F_{n+1} + F_n$. Sei die Anzahl N der Iterationen des Algorithmus vorgegeben. Für $k \leq N$ definieren wir im k-ten Schritt der Fibonaccisuche x_k, y_k wie folgt:

$$\begin{aligned} x_k &:= a_k + \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k) \\ y_k &:= a_k + \frac{F_{N+1-k}}{F_{N+2-k}}(b_k - a_k). \end{aligned}$$

Im N-ten Schritt setzen wir

$$\begin{aligned} x_N &:= y_N - \varepsilon = x_{N-1} - \varepsilon \quad \text{falls } b_N = y_{N-1}, \\ y_N &:= x_N + \varepsilon = y_{N-1} + \varepsilon \quad \text{falls } a_N = x_{N-1}, \end{aligned}$$

wobei ε die Maschinengenauigkeit ist.

Proposition 5.2.2 *a) Die Fibonaccisuche ist eine Implementierung des allgemeinen Suchverfahrens, d.h.*

$$\text{falls } a_{k+1} = x_k, \text{ so ist } x_{k+1} = y_k, \text{ falls } a_{k+1} = a_k, \text{ so ist } y_{k+1} = x_k.$$

b) Die Fibonaccisuche plaziert x und y symmetrisch in $[a, b]$, d.h. für $k < N$ gilt

$$x_k - a_k = b_k - y_k = \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k).$$

c) Für $k \leq N$ ist $b_k - a_k = \frac{F_{N+2-k}}{F_{N+1}}(b-a)$ und $b_{N+1} - a_{N+1} = \frac{1}{F_{N+1}}(b-a) + \varepsilon$

Beweis. Ist $a_{k+1} = x_k$, so ist $b_{k+1} = b_k$ und wir haben $a_{k+1} = a_k + \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k)$ und

$$\begin{aligned}
x_{k+1} &= a_{k+1} + \frac{F_{N-1-k}}{F_{N+1-k}}(b_{k+1} - a_{k+1}) \\
&= a_k + \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k) + \frac{F_{N-1-k}}{F_{N+1-k}}\left(b_k - a_k - \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k)\right) \\
&= a_k + \left(\frac{F_{N-k}}{F_{N+2-k}} + \frac{F_{N-1-k}}{F_{N+1-k}}(1 - \frac{F_{N-k}}{F_{N+2-k}})\right)(a_k - b_k) \\
&= a_k + \left(\frac{F_{N-k}}{F_{N+2-k}} + \frac{F_{N-1-k}}{F_{N+1-k}}(\frac{F_{N+1-k}}{F_{N+2-k}})\right)(a_k - b_k) \\
&= a_k + \left(\frac{F_{N-k}}{F_{N+2-k}} + \frac{F_{N-1-k}}{F_{N+2-k}}\right)(a_k - b_k) \\
&= a_k + \frac{F_{N+1-k}}{F_{N+2-k}}(a_k - b_k) = y_k
\end{aligned}$$

Im zweiten Fall berechnet man analog $y_{k+1} = x_k$ oder folgert es aus der Symmetrie (zweite Behauptung). Für diese zweite Behauptung haben wir

$$\begin{aligned}
b_k - y_k &= (b_k - a_k) - \frac{F_{N+1-k}}{F_{N+2-k}}(b_k - a_k) \\
&= (b_k - a_k)(1 - \frac{F_{N+1-k}}{F_{N+2-k}}) \\
&= (b_k - a_k)(\frac{F_{N-k}}{F_{N+2-k}}) = x_k - a_k.
\end{aligned}$$

Die letzte Behauptung zeigen wir mittels vollständiger Induktion. Die Verankerung besagt $b-a = b-a$ ist also sicher wahr. Wegen der eben gezeigten Symmetrie gilt nun

$$\begin{aligned}
b_{k+1} - a_{k+1} &= b_k - x_k \\
&= b_k - a_k - \frac{F_{N-k}}{F_{N+2-k}}(b_k - a_k) \\
&= (b_k - a_k)(1 - \frac{F_{N-k}}{F_{N+2-k}}) \\
&= \frac{F_{N+1-k}}{F_{N+2-k}}(b_k - a_k) \\
&\overset{I.V.}{=} \frac{F_{N+1-k}}{F_{N+2-k}} \frac{F_{N+2-k}}{F_{N+1}}(b - a) \\
&= \frac{F_{N+1-k}}{F_{N+1}}(b - a).
\end{aligned}$$

Schließlich ist $b_{N+1} - a_{N+1} = \frac{1}{2}(b_N - a_N) + \varepsilon = \frac{2}{2F_{N+1}}(b - a) + \varepsilon.$ □

Satz 5.2.3 *Sei S wie in Proposition 5.2.1. Dann gilt:*

$$S_N \geq \frac{1}{F_{N+1}}.$$

Also kann kein Suchverfahren bei fest vorgewählter Schrittzahl eine stärkere Reduktion des Suchintervalls garantieren.

Beweis. Offensichtlich ist $S_{N-1} \leq 2S_N$. Also haben wir für $k = 1, 2 : S_{N+1-k} \leq F_k S_N$. Wir beweisen nun per Induktion, daß diese Aussage für $k \leq N+1$ gilt. Nach Proposition 5.2.1 haben wir für $k \in 3, \ldots, N+1$

$$S_{N+1-k} \leq S_{N+1-(k-1)} + S_{N+1-(k-2)} \overset{i.V.}{\leq} F_{k-1}S_N + F_{k-2}S_N = F_k S_N.$$

Die Behauptung folgt nun aus $S_0 = 1$. □

Die Fibonacci-Suche hat zwei Nachteile:

a) man muß im Voraus wissen, wieviel Iterationen man machen will, bzw. wie klein das Suchintervall werden soll. Im allgemeinen wird man jedoch auch die Funktionswerte miteinbeziehen.

b) man muß eine Tabelle der Fibonaccizahlen bereitstellen. Diese sind entweder in Gleitkommadarstellung nur angenähert oder man muß Langzahlarithmetik verwenden.

Statt dessen betrachtet man direkt das Verhalten des Quotienten benachbarter Fibonacci-Zahlen (vgl. Übung 3.Serie):

$$\begin{aligned}
\frac{F_{n+1}}{F_n} &= \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}}{\left(\frac{1+\sqrt{5}}{2}\right)^{n} - \left(\frac{1-\sqrt{5}}{2}\right)^{n}} \\
&= \frac{\left(\frac{1+\sqrt{5}}{2}\right) - \left(\frac{1-\sqrt{5}}{2}\right)\left(\frac{1-\sqrt{5}}{1+\sqrt{5}}\right)^{n}}{1 - \left(\frac{1-\sqrt{5}}{1+\sqrt{5}}\right)^{n}} \\
&\overset{n\to\infty}{\to} \frac{1+\sqrt{5}}{2}.
\end{aligned}$$

Die Zahl $\frac{1+\sqrt{5}}{2}$ ist auch als *Goldener Schnitt* bekannt.

> Goldener Schnitt Gesetz zur Konstruktion harmonischer Proportionen. Beim G.S. verhält sich das längere Teilstück einer Strecke zum kürzeren Teilstück wie das längere Teilstück zur gesamten Strecke. Am häufigsten kommen die Verhältniszahlen 3:5, 5:8, 8:13 und 13:21 zur Anwendung. Beim typografischen Gestalten läßt sich der G.S. auf das Verhältnis von Abständen, Schriftgrößen, Seitenproportionen etc. anwenden.
> Quelle: Kleines Glossar Typographie und Layout im Desktop-Publishing, Zusammengestellt von Jürgen F. Schopp Universität Tampere, Finnland.

Im Grenzwert wird also aus der Fibonacci-Suche die *Goldener-Schnitt-Suche*. Wir plazieren x so, daß x, y symmetrisch in $[a, b]$ liegen und a, x, y ein Goldener Schnitt ist. Setzen wir also $l_1 = x - a, l_2 = y - a$, so haben wir aus Symmetriegründen $l_1 + l_2 = b - a$. Damit wir einen Goldenen Schnitt erhalten, müssen x, y so gewählt werden, daß

$$\frac{l_1}{l_2} = \frac{l_2 - l_1}{l_1}. \tag{5.1}$$

Da nun $l_1 = b - a - l_2$ ist, erhalten wir hieraus:

$$l_2^2 + (b-a)l_2 - (b-a)^2 = 0. \tag{5.2}$$

Diese quadratische Gleichung hat genau eine positive Nullstelle, nämlich

$$\frac{\sqrt{5}-1}{2}(b-a) = \left(\frac{1+\sqrt{5}}{2}\right)^{-1}(b-a).$$

Also wird y an dieser Stelle plaziert und auch a, y, b ist ein Goldener Schnitt. Wegen $1 - \frac{\sqrt{5}-1}{2} = \frac{3-\sqrt{5}}{2}$ erhalten wir als Vorschrift für die Goldener Schnitt-Suche.

$$\begin{aligned} x_k &:= a_k + \frac{3-\sqrt{5}}{2}(b_k - a_k) \\ y_k &:= a_k + \frac{\sqrt{5}-1}{2}(b_k - a_k). \end{aligned}$$

In Pseudocode erhalten wir:

```
leftfak=(3-sqrt(5))/2
rightfak=(sqrt(5)-1)/2

def choosepoint(f,a,x,b,fx):
  if b-x <= x-a:
    return a+leftfak*(b-a),x,f(a+leftfak*(b-a)),fx
  else:
    return x,a+rightfak*(b-a),fx,f(a+rightfak*(b-a))

def findmin(f,a,x,b):
  fx=f(x)
  while (b-a)/(abs(b)+abs(a)) >= eps:
    x,y,fx,fy=choosepoint(f,a,x,b,fx)
    if fx >= fy:
      a=x
      x=y
      fx=fy
    else:
      b=y
  else:
    return (a+b)/2
```

Bemerkung 5.2.4 *Wir haben hier nur Verfahren angesprochen, welche die Stetigkeit der Funktion ausnutzen. Es gibt einige Verfahren, welche Differenzierbarkeitsinformation ausnutzen, also etwa lineare oder quadratische Annäherung, auf die wir hier aber nicht näher eingehen wollen. Allerdings kann man die eindimensionale Variante des Newton-Verfahrens, das wir im übernächsten Abschnitt diskutieren als Beispiel heranziehen.*

5.3 Koordinatensuche, Methode des steilsten Abstiegs

In diesem Abschnitt wollen wir uns der Frage geeigneter Suchrichtungen zuwenden. Dabei wollen wir in unseren Untersuchungen Nebenbedingungen vernachlässigen. Die hier vorgestellten Algorithmen sind allerdings eher „prinzipiell" zu verstehen und haben sich in der Praxis als nicht besonders effizient erwiesen. Wir werden auf diese Problematik später noch etwas näher eingehen.

Alle Verfahren benutzen line search als Unterroutine. Bei unseren theoretischen Überlegungen in diesem Abschnitt wollen wir von der idealisierten Vorstellung ausgehen, daß der line search stets das Minimum findet.

Das simpelste Verfahren ist die sogenannte *Koordinatenabstiegsmethode*:

Sei $\bar{x} \in \mathbb{R}^n$ gegeben. Wir fixieren alle Koordinaten bis auf die i-te und lösen

$$\min_{x_i \in \mathbb{R}} f(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{i-1}, x_i, \bar{x}_{i+1}, \ldots, \bar{x}_n).$$

Ist x^* die optimale Lösung dieses Subproblems, so setzen wir $\bar{x} = x^*$, wählen eine andere Koordinate und fahren fort. Wir erhalten also folgenden Algorithmus (hier enden leider unsere Möglichkeiten, mit python „ausführbaren Pseudocode“ zu erzeugen):

while $\|x - x\_old\| > \varepsilon$:
  for i in [n]:
    x_old[i]=x[i]
    $\lambda = \text{argmin}_{\lambda} f(x + \lambda e_i)$ # löse mit line search
    $x = x + \lambda e_i$

Da die Auswahl der Koordinaten zyklisch erfolgt, nennt man obige Methode *zyklisches Abstiegsverfahren*. Werden die Koordinaten in der Reihenfolge $1, 2, \ldots, n-1, n, n-1, \ldots, 2, 1, 2, \ldots$ abgearbeitet, so trägt das Verfahren den Namen „Aitken double sweep method“. Nutzt man zusätzlich Differenzierbarkeitsinfomation aus und wählt stets die Koordinate mit dem größten Absolutwert im Gradienten, so erhalten wir das *Gauß-Southwell-Verfahren*.

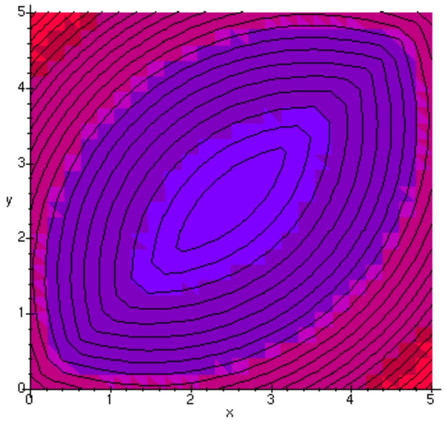
Die dargestellten Verfahren scheinen sinnvoll. Konvergenz gegen „etwas Sinnvolles“ kann man jedoch nur garantieren, wenn f differenzierbar ist. Betrachten wir zunächst ein Beispiel.

Beispiel 5.3.1 *Sei die stetige (!) Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ wie folgt definiert:*

$$f(x,y) := \begin{cases} (x+y-5)^2 + (x-y-2)^2 & \textit{falls} \quad x \leq y \\ (x+y-5)^2 + (x-y+2)^2 & \textit{falls} \quad x > y. \end{cases}$$

Wir führen eine Koordinatensuche beginnend in $(0,0)$ durch. Suchen wir in x-Richtung stellen wir zunächst fest, daß das Minimum in positive Richtung x-Richtung zu suchen ist. Also müssen wir zunächst die Funktion $(x-5)^2+(x+2)^2$ minimieren. Aus Symmetriegründen oder mittels nachrechnen findet man das Minimum in $x = 1.5$. In y Richtung minimieren wir also nun die Funktion $(y-3.5)^2 + (-y+3.5)^2$ für $y \leq 1.5$ und $(y-3.5)^2+(-y-0.5)^2$ für $y > 1.5$. Diese Funktion hat ihr Minimum in $y = 1.5$. Wieder in x-Richtung betrachten wir nun die Funktion $(x-3.5)^2+(x-3.5)^2$ für $x \leq 1.5$ und $(x-3.5)^2 + (x+0.5)^2$ für $x > 1.5$. Diese ist minimal in $x = 1.5$. In y-Richtung $(y-3.5)^2 + (-y-0.5)^2$ für $1.5 \leq y$ und $(y-3.5)^2 + (-y+3.5)^2$ für

$1.5 > y$ *erhalten wir die gleiche Funktion. Die Koordinatensuche terminiert also mit dem Wert* $4 + 4 = 8$*, das Minimum liegt aber in* $(2.5, 2.5)$ *mit dem Wert 4.*



Ist hingegen f differenzierbar, so können wir zeigen:

Satz 5.3.2 *Ist* $f : \mathbb{R}^n \to \mathbb{R}$ *stetig differenzierbar und ist* $(x_i)_{i\in\mathbb{N}}$ *eine Folge, die von einem Koordinatensuchverfahren erzeugt wird, so konvergiert jede konvergente Teilfolge von* $(x_{i_j})_{j\in\mathbb{N}}$ *gegen ein* x^* *mit* $\nabla f(x^*) = 0$.

Beweis. Angenommen $\nabla f(x^*) \neq 0$. Sei dann $(y_i)_{i\in\mathbb{N}}$ ein Teilfolge von $(x_{i_j})_{j\in\mathbb{N}}$, bei der stets in Richtung e_{i_0} mit $\nabla f(x^*)_{i_0} \neq 0$ gesucht wird. Eine solche Teilfolge gibt es offensichtlich in allen drei Varianten des Algorithmus. Da $-f(x^* - te_{i_0})'(0) = f(x^* + te_{i_0})'(0) = \nabla f(x^*)_{i_0} \neq 0$ ist, gibt es ein $\alpha \neq 0$ mit $f(x^* + te_{i_0}) < f(x^*)$ für alle $t \in]0, \alpha]$ bzw. $t \in [\alpha, 0[$. Da f stetig ist, gibt es ein ε mit $f(x) < f(x^*)$ für alle $x \in U_\varepsilon(x^* + \alpha e_{i_0})$. Sei nun y_j ein Folgenelement mit $y_j \in U_\varepsilon(x^*)$. Dann ist $y_j + \alpha e_{i_0} \in U_\varepsilon(x^* + \alpha e_{i_0})$ und somit $f(y_j + \alpha e_{i_0}) < f(x^*)$ also gilt auch für den Nachfolger x^j von y_j in der Folge $f(x^j) < f(x^*)$. Dies widerspricht aber der Tatsache, daß $(f(x_i))_{i\in\mathbb{N}}$ monoton fallend ist und gegen $f(x^*)$ konvergiert. □

Koordinatenabstiegsvefahren haben sich in der Praxis nur in ganz wenigen Spezialfällen (z.B. Probleme mit Rechtecknebenbedingungen) bewährt, i.a. ist ihr Konvergenzverhalten schlecht, so daß selbst Probleme mit geringer Variablenzahl kaum gelöst werden können. Es scheint, daß die einzige einigermaßen erfolgreiche Variante die *Methode von Rosenbrock* ist, bei der in jeder Iteration ein neues „Koordinatensystem“ gewählt wird.

Als nächstes wollen wir in Richtung des Gradienten suchen.

Methode des steilsten Abstiegs:

while $\|\nabla f(x)\| > \varepsilon$:
  $\lambda = \operatorname{argmin}_\lambda f(x - \lambda \nabla f(x))$ # löse mit line search
  $x = x - \lambda \nabla f(x)$.

Auch hier weisen wir die Konvergenz nach:

Satz 5.3.3 *Sei $f : \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar und sei $(x_i)_{i \in \mathbb{N}}$ eine konvergente Teilfolge einer von der Methode des steilsten Abstiegs erzeugten Punktfolge. Dann konvergiert (x_i) gegen einen stationären Punkt x^*, d.h. $\nabla f(x^*) = 0$.*

Beweis. Angenommen $\nabla f(x^*) \neq 0$. Da $-\nabla f(x^*)$ dann eine Abstiegsrichtung ist, gibt es ein α mit $f(x - \alpha \nabla f(x^*)) < f(x^*)$. Da f stetig ist, gibt es ein $\varepsilon > 0$ mit $f(x) < f(x^*)$ für alle $x \in U_\varepsilon(x - \alpha \nabla f(x^*)$. Da f stetig differenzierbar ist, gilt $\lim_{n\to\infty} \nabla f(x_n) = \nabla f(x^*)$. Da außerdem $x_i \to x^*$ geht, können wir N_0 so wählen, daß

$$\|x_{N_0} - x^*\| < \frac{\varepsilon}{2} \text{ und } \|\nabla f(x_{N_0}) - \nabla f(x^*)\| < \frac{\varepsilon}{2|\alpha|}.$$

Dann ist aber

$$\begin{aligned}
\|x_N - \alpha \nabla f(x_N) - (x^* - \alpha \nabla f(x^*))\| &= \|x_N - x^* - (\alpha \nabla f(x_N) - \alpha \nabla f(x^*))\| \\
&\leq \|x_N - x^*\| + \|\alpha \nabla f(x_N) - \alpha \nabla f(^*))\| \\
&= \|x_N - x^*\| + |\alpha| \|\nabla f(x_N) - \nabla f(^*))\| \\
&< \frac{\varepsilon}{2} + |\alpha| \frac{\varepsilon}{2|\alpha|} = \varepsilon
\end{aligned}$$

und der Widerspruch folgt wie eben. □

Obwohl dieses Verfahren lokal die „beste“ Richtung benutzt ist sein Konvergenzverhalten eher mäßig. Eine Analyse der Konvergenzrate ist etwas aufwendiger. Wir verweisen auf Luenberger S. 149–154 und zitieren:

Satz 5.3.4 *Sei $f : \mathbb{R}^n \to R$ zweimal stetig differenzierbar und x^* ein relatives Minimum von f. Sei ferner die Hessematrix $\nabla^2 f(x^*)$ positiv definit mit größtem Eigenwert $\lambda_1 > 0$ und kleinstem Eigenwert $\lambda_n > 0$. Ist dann $(x_i)_{i \in \mathbb{N}}$ eine von dem Gradientenabstiegsverfahren erzeugte, gegen x^* konvergente Folge, dann konvergiert die Folge der Zielfunktionswerte $(f(x_i))_{i \in \mathbb{N}}$ linear gegen $f(x^*)$ mit einer Konvergenzrate von höchstens $\left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right)$.*

Bemerkung 5.3.5 *Die Gradienten aufeinanderfolgender Iterationspunkte stehen senkrecht aufeinander, d.h. es gilt stets:*

$$\nabla f(x^k) \nabla f(x^{k+1})^\top = 0.$$

Beweis. $0 = f(x_k + t \nabla f(x_k))'(\lambda) = \nabla f(x_k) \nabla f(x_{k+1})^\top$. □

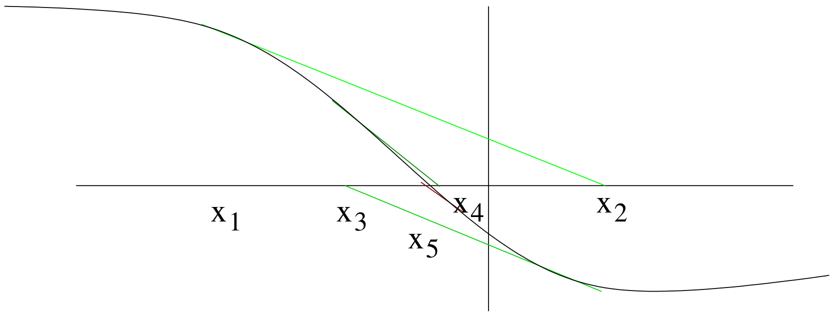
Bemerkung 5.3.6 *Bei der Benutzung von Ableitungen in numerischen Algorithmen nähert man diese üblicherweise nur an, d.h. der Gradient $\nabla f(x^*)$ wird etwa angenähert durch den Ausdruck*

$$\frac{1}{h}(f(x^* + he_1) - f(x^*), \ldots, f(x^* + he_n) - f(x^*)).$$

5.4 Newtonverfahren

Der Hauptvorteil des Newtonverfahrens ist, daß das lokale Konvergenzverhalten deutlich besser als beim line search ist.

Vielleicht kennen Sie das Newton-Verfahren zur Bestimmung der Nullstelle einer Funktion noch aus der Schule: $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$. Hierbei wird iterativ die Funktion lokal durch eine lineare Funktion angenähert.



Relaxieren wir nun die Suche nach einem lokalen Minimum zu der Suche nach einem stationären Punkt, so erhalten wir die Vorschrift:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}.$$

Hier wird also die Funktion f lokal durch die quadratische Funktion $q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2$ approximiert und für den nächsten Iterationspunkt deren eindeutiger stationärer Punkt berechnet.

In dieser Form und Interpretation können wir das Newton-Verfahren direkt auf die Situation einer Funktion $f : \mathbb{R}^n \to \mathbb{R}$ übertragen. Wir erhalten dann folgendes Verfahren:

while $\|x - x\_old\| > \varepsilon$:
  $x\_old = x$
  $x = x - \left(\nabla^2 f(x)\right)^{-1} \nabla f(x)^\top$

Im allgemeinen können bei dieser Vorgehensweise schon bei der Bestimmung einer Nullstelle im Eindimensionalen Schwierigkeiten auftreten, nämlich, daß die Ableitung Null wird, weil man sich einem stationären Punkt nähert. Im allgemeinen können folgende Probleme auftreten

a) Im Laufe des Verfahrens kann die Hesse-Matrix singulär oder schlecht konditioniert werden.

b) Es kann passieren, daß $f(x_{k+1}) > f(x_k)$ ist.

c) Die Folge der generierten Punkte kann gegen einen Sattelpunkt konvergieren.

Wir werden nun aber nachweisen, daß lokal das Konvergenzverhalten sehr gut ist.

Satz 5.4.1 *Sei $f : \mathbb{R}^n \to \mathbb{R}$ zweimal stetig differenzierbar, sei $\nabla f(x^*) = 0$, $\nabla^2 f(x^*)$ regulär und x_1 ein Startpunkt, so daß es δ_1, δ_2 gibt mit $\delta_1\delta_2 < 1$ und für alle x mit $\|x - x^*\| < \|x_1 - x^*\|$ gilt:*

a) $\| \left(\nabla^2 f(x)\right)^{-1} \|_2 \leq \delta_1$,

b) $\dfrac{\|\nabla f(x^*) - \nabla f(x) - \nabla^2 f(x)(x - x^*)\|}{\|x - x^*\|} \leq \delta_2.$

Dann konvergiert das Newtonverfahren gegen x^*.

Beweis.

$$\begin{aligned}
\|x_{k+1} - x^*\| &= \|x_k - \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k) - x^*\| \\
&= \|(x_k - x^*) - \left(\nabla^2 f(x_k)\right)^{-1} (\nabla f(x_k) - \nabla f(x^*))\| \\
&= \| \left(\nabla^2 f(x_k)\right)^{-1} \left(\nabla f(x^*) - \nabla f(x_k) - \left(\nabla^2 f(x_k)\right)(x^* - x_k)\right) \| \\
&\leq \| \left(\nabla^2 f(x_k)\right)^{-1} \| \| \left(\nabla f(x^*) - \nabla f(x) - \left(\nabla^2 f(x_k)\right)(x^* - x_k)\right) \| \\
&\leq \delta_1\delta_2\|x^* - x_k\| \leq \|x^* - x_k\|.
\end{aligned}$$

Wir sagen, die Methode ist kontraktiv. Also bildet $(\|x^* - x_k\|)_{k\in\mathbb{N}}$ eine streng monoton fallende nichtnegative Folge, die somit konvergieren muß. Falls diese Folge gegen Null konvergiert, sind wir fertig. Angenommen also die Folge konvergierte gegen einen Wert > 0. Da die Werte der x_k betraglich beschränkt sind, können wir eine konvergente Teilfolge $(x_{k_i})_{i\in\mathbb{N}}$ wählen, konvergiere diese gegen y_1. Da die Abbildung $x \mapsto x - \left(\nabla^2 f(x)\right)^{-1} \nabla f(x)$ mit ∇f und $\nabla^2 f$ stetig ist, konvergiert die Folge

$(x_{k_i+1} = x_{k_i} - \left(\nabla^2 f(x_{k_i})\right)^{-1} \nabla f(x_{k_i}))_{i\in\mathbb{N}}$ gegen $y_2 = y_1 - \left(\nabla^2 f(y_1)\right)^{-1} \nabla f(y_1))$. Also ist y_2 ein anderer Häufungspunkt des Algorithmus. Somit gilt auch $\|x^* - y_1\| = \|x^* - y_2\|$. Andererseits folgt mit der gleichen Rechnung wie eben: $\|x^* - y_1\| < \|x^* - y_2\|$. Widerspruch. □

Da das Newton-Verfahren aus einer quadratischen Annäherung an die Funktion abgeleitet ist, erwarten wir lokal quadratische Konvergenz:

Satz 5.4.2 *Sei $f : \mathbb{R}^n \to \mathbb{R}$ viermal stetig differenzierbar, x^* ein Punkt mit $\nabla f(x^*) = 0$ und $\nabla^2 f(x^*)$ regulär. Sei $(x_k)_{k\in N}$ eine vom Newton-Verfahren erzeugte, gegen x^* konvergente Folge, also $x_{k+1} = N(x_k) := x_k - (\nabla^2 f(x^*))^{-1} \nabla f(x^*)^\top$. Dann konvergiert die Folge quadratisch.*

Beweis. Die Funktion $N(x)$ ist als Verknüpfung zweimal stetig differenzierbarer Funktionen zweimal stetig differenzierbar. Wir berechnen den Eintrag $(JN(x))_{ij}$ der Jacobischen. Dafür bezeichnen wir die ik-te Komponentenfunktionen der Matrix $G(x) = \left(\nabla^2 f(x)\right)^{-1}$ mit $G_{ik}(x)$. Dann ist

$$\begin{aligned}
JN(x))_{ij} &= \frac{\partial}{\partial x_j}\left(x_i - \sum_{k=1}^n G_{ik}(x)\frac{\partial f}{\partial x_k}(x)\right) \\
&= \delta_{ij} - \sum_{k=1}^n \frac{\partial G_{ik}}{\partial x_j}(x)\frac{\partial f}{\partial x_k}(x) - \sum_{k=1}^n G_{ik}(x)\frac{\partial^2 f}{\partial x_k \partial x_j}(x) \\
&= \delta_{ij} - \frac{\partial}{\partial x_j}(G_{i.})(x)\nabla f(x)^\top - (G_{i.})(\nabla^2 f(x))_{.j}) \\
&= -\frac{\partial}{\partial x_j}(G_{i.})(x)\nabla f(x)^\top,
\end{aligned}$$

wobei

$$\delta_{ij} = \begin{cases} 1 \text{ falls } i = j \\ 0 \text{ sonst} \end{cases}.$$

Da nach Annahme $\nabla f(x^*) = 0$ ist, gilt insbesondere $JN(x^*) = 0$ und wir erhalten

$$\begin{aligned}
|x_{k+1} - x^*| &= |N(x_k) - N(x^*)| \\
&\leq |JN(x^*)(x_k - x^*)| + \frac{1}{2}|N''(\bar{x})||x_k - x^*|^2.
\end{aligned}$$

Hierbei nutzen wir den Satz von Taylor und den Mittelwertsatz, also ist $\bar{x}$ ein Punkt auf der Verbindungslinie von x_k nach x^*. Wir bitten die etwas laxe Schreibweise zu entschuldigen. Z.B. ist $N''(x)$ ein Gebilde, bei dem bei der „Multiplikation“ mit zwei Vektoren einen Vektor erhält. Man mache sich die Situation im Eindimensionalen klar.

Wir erhalten also mit einer Abschätzung c des Terms $\frac{1}{2}|N''(\bar{x})|$ in der Nähe von x^* nach oben:

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

$\square$

Leider konvergiert dieses Verfahren nicht unbedingt. Man kann es auf verschiedene Arten modifizieren, um Konvergenz zu erzwingen. Wir wollen hier eine Möglichkeit darstellen.

Dazu betrachten wir allgemein Verfahren, bei denen die Iterationsvorschrift durch

$$x_{k+1} = x_k - \alpha_k M_k \nabla f(x_k)^\top$$

mit einem Suchparameter α_k und einer positiv definiten Matrix M_k gegeben ist. Dann ist in erster Näherung (bei der Entwicklung in erster Näherung bleiben quadratische und höhere Terme „übrig“)

$$\begin{aligned} f(x_{k+1}) &= f(x_k) + \nabla f(x_k)(x_{k+1} - x_k) + O(|x_{k+1} - x_k|^2) \\ &= f(x_k) - \alpha_k \nabla f(x_k) M_k \nabla f(x_k)^\top + O(\alpha_k^2). \end{aligned}$$

Nahe bei Null dominiert der in α_k lineare Term, also garantiert die positive Definitheit von M_k, daß $M_k \nabla f(x_k)^\top$ eine Abstiegsrichtung ist. Für $M_k = I$ erhalten wir das steilste Abstiegsverfahren. Genauso wie dort kann man auch globale Konvergenz nachweisen. Nahe bei einem lokalen Minimum mit positiv definiter Hessematrix erhalten wir ein parametrisiertes Newtonverfahren.

Nun ist die Hessematrix bei zweimal stetig differenzierbaren Funktionen stets symmetrisch, also gibt es nach Satz 2.4.3 eine orthogonale Matrix Q und eine Diagonalmatrix D mit $\nabla^2 f(x_k) = Q^\top DQ$, wobei auf der Diagonale von D die Eigenwerte von $\nabla^2 f(x_k)$ stehen. Die Diagonaleinträge d_{ii} ersetzt man nun durch $\max\{\delta, d_{ii}\}$, wobei $\delta > 0$ ein Steuerungsparameter ist. Nahe bei dem lokalen Minimum sind die Eigenwerte alle $\geq \delta$ und die Methode wird zum Newtonverfahren.

5.5 Verfahren der konjugierten Richtungen

Der Ansatz der konjugierten Richtungen ist ein weiterer Versuch, die Vorteile des steilsten Abstiegsverfahrens (globale Konvergenz) mit denen des Newton-Verfahrens (Ausnutzung von Information zweiter Ordnung) zu verknüpfen. Die Ideen und Eigenschaften lassen sich besser am Spezialfall eines quadratischen Programms studieren. Wir betrachten also zunächst folgendes Problem:

Sei $Q \in \mathbb{R}^{n \times n}$ eine quadratische, symmetrische, positiv definite Matrix und $b \in \mathbb{R}^n$. Wir betrachten das Minimierungsproblem

$$\min g(x) = \frac{1}{2} x^\top Q x + b^\top x.$$

Eine Möglichkeit dies Problem zu lösen, wäre die notwendigen Bedingungen zu betrachten und

$$\nabla g(x)^\top = Qx - b = 0 \tag{5.3}$$

zu lösen. Wir wollten aber näher an einer Richtungssuche bleiben und das aufwendige Invertieren der Matrix umgehen. Nach einem line search hat man die Optimierungsaufgabe optimal auf einem 1-dimensionalen affinen Teilraum des Lösungsraumes bewältigt. Wir werden nun iterativ Suchrichtungen konstruieren, so daß die Dimension des Teilraumes, auf dem das Problem optimal gelöst ist, stets um eins wächst. Dafür definieren wir zunächst.

Definition 5.5.1 *Sei $Q \in \mathbb{R}^{n \times n}$ eine quadratische, symmetrische Matrix. Dann heißen zwei Vektoren $d_1, d_2 \in \mathbb{R}^n$* Q-orthogonal, Q-konjugiert *oder auch kurz* konjugiert, *wenn $d_1^\top Q d_2 = 0$ gilt. Eine endliche Menge von Vektoren $d_1, \ldots, d_k \in \mathbb{R}^n$ heißt* Q-orthogonal, *wenn sie paarweise konjugiert sind.*

Bilden $d_1, \ldots, d_n$ eine Orthonormalbasis von Eigenvektoren einer symmetrischen Matrix, so sind sie Q-orthogonal und orthogonal im euklidischen Sinne. Im allgemeinen fallen die Begriffe nicht zusammen. Für den positiv definiten Fall gilt allerdings:

Proposition 5.5.2 *Sei $Q \in \mathbb{R}^{n \times n}$ eine quadratische, symmetrische, positiv definite Matrix und $d_1, \ldots, d_k \in \mathbb{R}^n \setminus \{0\}$ Q-orthogonal. Dann sind diese Vektoren linear unabhängig.*

Beweis. Sei $\sum_{i=1}^k \alpha_i d_i = 0$. Dann ist auch $0 = d_j^\top Q 0 = \sum_{i=1}^k \alpha_i d_j^\top Q d_i = \alpha_j d_j^\top Q d_j$. Da Q positiv definit ist schließen wir, daß $\alpha_j = 0$ für $j = 1, \ldots, k$ gelten muß. □

Haben wir also in unserer Aufgabenstellung der quadratischen Optimierung konjugierte Richtungen $d_1, \ldots, d_n$ gegeben, so ist die Optimallösung x^* eine Linearkombination dieser Vektoren:

$$x^* = \sum_{i=1}^n \alpha_i d_i.$$

Aus dieser Gleichung können wir unter Berücksichtigung von Gleichung (5.3) herleiten:

$$\alpha_i = \frac{d_i^\top Q x^*}{d_i^\top Q d_i} = \frac{d_i^\top b}{d_i^\top Q d_i}. \tag{5.4}$$

Somit können wir x^* durch Skalar- und Matrixprodukte ausrechnen:

$$x^* = \sum_{i=1}^{n} \frac{d_i^\top Q x^*}{d_i^\top Q d_i} d_i. \tag{5.5}$$

Wir können diese Linearkombination allerdings erst berechnen, wenn wir eine konjugierte Basis haben. Dann ist es allerdings leicht, auf von einer Teilmenge der d_i aufgespannten Unterraum das quadratische Problem zu lösen.

Satz 5.5.3 *Seien Q, b wie eben, $d_1, \ldots, d_n$ Q-orthogonal und $x_0 \in \mathbb{R}^n$. Bezeichne B_i den von $d_1, \ldots, d_i$ an x_0 aufgespannten affinen Unterraum*

$$B_i := \{x_0 + \sum_{j=1}^{i} \lambda_j d_j \mid \lambda_j \in \mathbb{R}\} = \{y \in \mathbb{R}^n \mid d_k^\top (y - x_0) = 0, k = i+1, \ldots, n\}.$$

Seien nun $x_1, \ldots, x_n$ definiert durch

$$x_k := x_{k-1} - \frac{x_{k-1}^\top Q d_k - b^\top d_k}{d_k^\top Q d^k} d_k. \tag{5.6}$$

Dann ist x_k die Optimallösung des Problems

$$\min_{x \in B_k} \frac{1}{2} x^\top Q x - b^\top x.$$

Insbesondere löst x_n das volle quadratische Problem.

Beweis. Wir zeigen dies mittels vollständiger Induktion. Für $k = 0$ ist die Behauptung offensichtlich richtig. Sei also $k > 0$. Da die Nebenbedingungen hier linear sind, sind die Kuhn-Tucker Bedingungen äquivalent dazu, daß $\nabla g(x_k) = Q x_k - b$ senkrecht auf $d_1, \ldots, d_k$ steht. Nun ist aber

$$x_k^\top Q d_k - b^\top d_k = x_{k-1}^\top Q d_k - \frac{x_{k-1}^\top Q d_k - b d^k}{d_k^\top Q d^k} d_k^\top Q d_k - b d_k = 0 \tag{5.7}$$

und für $j < k$.

$$\begin{aligned} x_k^\top Q d_j - b^\top d_j &= x_{k-1}^\top Q d_j - \frac{x_{k-1}^\top Q d_k - b d_k}{d_k^\top Q d_k} d_k^\top Q d_j - b d_j \\ &= x_{k-1}^\top Q d_j - b d_j \\ &= \nabla g(x_{k-1}) d_j \overset{I.V.}{=} 0. \end{aligned}$$

□

Ein großer Nachteil der bisherigen Überlegungen ist, daß sie stets davon ausgehen, daß eine konjugierte Basis bekannt ist. Im folgenden werden wir diese dynamisch erzeugen.

Methode der konjugierten Gradienten Sei $x_0 \in \mathbb{R}^n$ und $d_1 = -g_0 = b - Qx_0$. Iterativ berechnen wir

$$\begin{aligned} x_k &= x_{k-1} - \frac{g_{k-1}^\top d_k}{d_k^\top Q d_k} d_k && (5.8)\\ g_k &= Qx_k - b && (5.9)\\ d_{k+1} &= -g_k + \frac{g_k^\top Q d_k}{d_k^\top Q d_k} d_k. && (5.10) \end{aligned}$$

Wenn wir nun nachweisen, daß $d_1, \ldots, d_n$ Q-konjugiert sind, so erfüllt das Verfahren offensichtlich die Voraussetzungen von Satz 5.5.3. Dies wird mit dem folgenden Satz getan:

Satz 5.5.4 *Die in (5.8, 5.9, 5.10) defnierte Methode der konjugierten Gradienten erfüllt die Voraussetzungen von Satz 5.5.3. Insbesondere gilt, falls das Verfahren nicht in x_k terminiert, daß*

a) $\mathit{lin}(\{g_0, g_1, \ldots, g_{k-1}\}) = \mathit{lin}(\{g_0, Qg_0, Q^2 g_0, \ldots, Q^k g_0\})$,

b) $\mathit{lin}(\{d_0, d_1, \ldots, d_k\}) = \mathit{lin}(\{g_0, Qg_0, Q^2 g_0, \ldots, Q^k g_0\})$,

c) $d_k^\top Q d_i = 0$ *für* $i < k$,

d) $-\dfrac{g_{k-1}^\top d_k}{d_k^\top Q d_k} = \dfrac{g_{k-1}^\top g_{k-1}}{d_k^\top Q d_k}$,

e) $\dfrac{g_k^\top Q d_k}{d_k^\top Q d_k} = \dfrac{g_k^\top g_k}{g_{k-1}^\top g_{k-1}}$.

Beweis. Wir zeigen zunächst die ersten drei Behauptungen mittels vollständiger Induktion, wobei die Verankerung klar ist. Sei also $k > 0$. Dann ist $g_k = Qx_k - b = g_{k-1} - \frac{g_{k-1}^\top d_k}{d_k^\top Q d_k} Q d_k$. Nach Induktionsvoraussetzung sind

$$g_{k-1}, Qd_k \in \operatorname{lin}(\{g_0, Qg_0, Q^2 g_0, \ldots, Q^k g_0\}),$$

somit gilt dies auch für g_k. Andererseits ist $g_k \notin \operatorname{lin}(\{g_0, Qg_0, Q^2 g_0, \ldots, Q^{k-1} g_0) = \operatorname{lin}(d_1, \ldots, d_k)$, da ansonsten das Minimum der Zielfunktion innerhalb B_k liegen würde und somit nach Induktionsvoraussetzung für die Konjugiertheit der Richtungen in x_k angenommen würde. Somit wäre dann $g_{k+1} = 0$ und der Algorithmus terminiert in x_k im Widerspruch zur Voraussetzung. Die zweite Behauptung folgt nun sofort aus der Formel (5.10) und der ersten.

Nun ist $d_{k+1}^\top Qd_i = -g_k^\top Qd_i + \frac{g_k^\top Qd_k}{d_k^\top Qd_k} d_k^\top Qd_i$. Für $i = k$ evaluiert man den Ausdruck zu Null, für $i < k$ sind beide Summanden Null, der zweite nach Induktionsvoraussetzung und der erste, weil $Qd_i \in \operatorname{lin}(d_1, \ldots, d_{i+1})$ und g_k senkrecht auf diesem Raum steht (wegen Satz 5.5.3).

Schließlich berechnen wir

$$-g_{k-1}^\top d_k = g_{k-1}^\top g_{k-1} + \frac{g_k^\top Qd_k}{d_k^\top Qd_k} g_{k-1}^\top d_{k-1},$$

wobei der zweite Summand Null ist, da g_{k-1} senkrecht auf B_{k-1} steht. Aus (5.9) schließen wir

$$Qd_k = -\frac{d_k^\top Qd_k}{g_{k-1}^\top d_k}(g_k - g_{k-1}). \tag{5.11}$$

Wegen *a*), *b*) und Satz 5.5.3 ist $g_{k-1}^\top g_k = 0$ und somit $g_k^\top Qd_k = -\frac{d_k^\top Qd_k}{g_{k-1}^\top d_k} g_k^\top g_k$. $\square$

Zum Abschluß der Vorlesung wollen wir zwei mögliche Erweiterungen auf nichtquadratische Probleme diskutieren. Eine naheliegende ist die Methode der quadratischen Approximation, bei der wir stets die Matrix Q durch die momentan aktuelle Hessematrix ersetzen.

Wir betrachten also nun wieder ein nichtlineares Optimierungsproblem

$$\min f(x).$$

Quadratische Approximation Sei $x_0 \in \mathbb{R}^n$ und $d_1 = -g_0 = \nabla f(x_0)$.

While Abbruchbedingung noch nicht erfüllt
For $i = 1, \ldots, n$:

$$x_k = x_{k-1} - \frac{g_{k-1}^\top d_k}{d_k^\top \nabla^2 f(x_{k-1}) d_k} d_k$$

$g_k = \nabla f(x_k)$

if $k \neq n$:

$$d_{k+1} = -g_k + \frac{g_k^\top \nabla^2 f(x_{k-1}) d_k}{d_k^\top Qd_k} d_k$$

else:

$x_0 = x_n$.

Der Vorteil dieser Methode ist, daß man keinen expliziten line search durchführen muß. Dennoch hat dieser Ansatz zwei gravierende Nachteile. Zum einen ist die ständige Neuberechnung der Hessematrix sehr aufwendig, zum anderen ist die Methode in dieser Form im allgemeinen nicht global konvergent.

Der folgende Ansatz von Fletcher und Reeves berücksichtigt diese Nachteile, indem er einerseits in jedem Schritt einen line search durchführt und andererseits die Hessematrix durch die Identität abschätzt.

Methode nach Fletcher-Reeves Sei $x_0 \in \mathbb{R}^n$ und $d_1 = -g_0 = \nabla f(x_0)$.

While Abbruchbedingung noch nicht erfüllt
For $i = 1, \ldots, n$:
$\alpha_k = \operatorname{argmin} f(x_{k-1} + \alpha_k d_k)$ # löse mit line search
$x_k = x_{k-1} + \alpha_k d_k$
$g_k = \nabla f(x_k)$
if $k \neq n$:
$$d_{k+1} = -g_k + \frac{g_k^\top g_k}{g_{k-1}^\top g_{k-1}} d_k$$
else:
$x_0 = x_n$.

Die globale Konvergenz dieses Verfahrens wird dadurch sichergestellt, daß einerseits der Wert der Zielfunktion nie steigt, da ein line search durchgeführt wird, und andererseits alle n Schritte ein Gradientenabstiegsschritt durchgeführt wird. Dies ist ein Beispiel für einen *Spacer Step*. Ganz allgemein kann man in Abstiegsverfahren durch „Untermischen“ von unendlich vielen Schritten eines global konvergenten Algorithmus globale Konvergenz erzwingen. Genauer gilt:

Satz 5.5.5 (Spacer Step Theorem) *Sei A eine stetige Funktion und sei $x_{k+1} = A(x_k)$ die Iterationsvorschrift eines auf der Menge $X \subseteq \mathbb{R}^n$ global gegen ein Elemente der Menge Γ (z.B. die Menge der stationären Punkte) konvergenten Verfahrens und f eine stetige Funktion mit*

$$f(A(x_k)) < f(x_k) \text{ für alle } x \in X \setminus \Gamma \tag{5.12}$$

und

$$f(A(x_k)) \leq f(x_k) \text{ für alle } x \in \Gamma. \tag{5.13}$$

Sei nun $(y_n)_{n\in\mathbb{N}}$ eine Folge, $\mathcal{K} \subseteq \mathbb{N}$ eine unendliche Indexmenge mit $y_{n+1} = A(y_n)$ falls $n \in \mathcal{K}$ und $f(y_{n+1}) \leq f(y_n)$ für alle $n \in N$. Dann konvergiert jede konvergente Teilfolge von $(y_k)_{k\in\mathcal{K}}$, gegen ein Element in Γ.

Beweis. Sei $(z_i)_{i \in \mathbb{N}}$ ein solche Folge und $\lim_{i \to infty} z_i = x^*$. Da A stetig ist, konvergiert die Folge $(A(z_i))_{i \in \mathbb{N}}$ gegen $A(x^*)$. Wegen (5.12) und (5.13) konvergiert $f(y_n)$ gegen $f(x^*) = f(A(x^*))$. Letzteres impliziert aber wegen (5.12) $x^* \in \Gamma$. $\square$

Ende der Vorlesung