

PROSEMINAR 2009  
Numerische Optimierung: Quasi-Newton-Verfahren  
Kapitel 8.1-8.3\*

CONSTANTIN Simone  
Ch. des Epinettes 51  
1723 Marly

19. November 2009

**Inhaltsverzeichnis**

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>Das BFGS-Verfahren</b>	<b>2</b>
2.1	Die Herleitung der DFP- und BFGS-Formeln . . . . .	2
2.2	Der BFGS Algorithmus . . . . .	5
2.3	Eigenschaften der BFGS Methode . . . . .	5
2.4	Implementation . . . . .	6
<b>3</b>	<b>Die SR1 Methode</b>	<b>6</b>
3.1	Die Herleitung der SR1-Formeln . . . . .	7
3.2	Der SR1 Algorithmus . . . . .	8
3.3	Eigenschaften der SR1 Modifikation . . . . .	8
<b>4</b>	<b>Die Broyden-Klasse</b>	<b>10</b>
4.1	Die Formeln der Broyden-Klasse . . . . .	10
4.2	Eigenschaften der Broyden-Klasse . . . . .	11

---

\*Jorge Nocedal and Stephen J. Wright: Numerical Optimization, Springer Series in Operations Research, Springer 1999.

# 1 Einleitung

In diesem Kapitel behandeln wir die Klasse der Quasi-Newton-Verfahren für kleine und mittelgrosse Probleme. Die Verfahrensklasse der unrestringierten Minimierung, wird derzeit als die effizienteste und zuverlässigste zur Lösung von Problemen nicht zu hoher Dimension angesehen.

W.C. Davidon benutzte Mitte der 50er Jahre die „coordinate descent“-Methode zur Berechnung des Optimierungsproblems. Das Computersystem stürzte jedoch immer vor Ende der Berechnung ab, da zu dieser Zeit die Rechner nicht sehr stabil waren. Also schrieb Davidon einen Algorithmus um die Iteration zu beschleunigen, den ersten Quasi-Newton-Algorithmus. Fletcher und Powell zeigten, dass dieser viel schneller und bewährter war als alle anderen bisherig bekannten Verfahren.

Die Quasi-Newton-Verfahren verwenden anstelle der exakten Hesse-Matrix der zu minimierenden Funktion eine geeignete Approximation an diese. Dadurch wird die häufig sehr aufwendige explizite Berechnung der zweiten partiellen Ableitung der Zielfunktion vermieden. Diese Approximation wird dabei von Iteration zu Iteration aufdatiert. In der Berechnung werden also keine zusätzlichen Funktions- oder Ableitungswerte benötigt. Damit ist jeder Iterationsschritt der Quasi-Newton-Verfahren weniger aufwendig und oft viel effizienter als beim Newton-Verfahren. Dennoch sind viele der Quasi-Newton-Verfahren lokal superlinear konvergent.

## 2 Das BFGS-Verfahren

Das beliebteste und erfolgreichste Quasi-Newton-Verfahren ist die BFGS Methode. Die BFGS-Formel, die praktisch zeitgleich und unabhängig voneinander von Broyden, Fletcher, Goldfarb, und Shanno, entdeckt wurde, leiten wir nun her.

### 2.1 Die Herleitung der DFP- und BFGS-Formeln

Um dieses Methode zu konstruieren, bilden wir das quadratischen Model der Zielfunktion zum Zeitpunkt des Iterationswertes  $x_k$ :

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p \quad (1)$$

wobei  $B_k \in \mathbb{R}^{n \times n}$  symmetrisch und positiv definit ist. Die Matrix  $B_k$  wird bei jeder Iteration aufdatiert. Der Wert und der Gradient dieses Models entspricht bei  $p = 0$  jeweils  $f_k$  und  $\nabla f_k$ . Als neue Iteration erhalten wir

$$x_{k+1} = x_k + \alpha_k p_k, \quad (2)$$

mit der Suchrichtung

$$p_k = -B_k^{-1} \nabla f_k. \quad (3)$$

Hierbei muss die Schrittweite  $\alpha_k$  die Wolfe-Bedingungen (cf. Kapitel 3, JULIEN Cyril)

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k, \quad (4a)$$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k, \quad (4b)$$

mit  $0 < c_1 < c_2 < 1$  erfüllen.

Diese Iteration ist sehr ähnlich zum Abstiegsverfahren von Newton. Der Unterschied besteht

darin, dass wir anstelle der exakten Hesse-Matrix der zu minimierenden Funktion die approximierete Matrix  $B_k$  benützen.

Anstatt  $B_k$  bei jeder Iteration neu zu berechnen, schlug Davidon vor die Matrix auf einfache Weise aufzudatieren. Angenommen wir würden eine neue Iteration  $x_{k+1}$  bilden und ein neues quadratische Model der Form

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p$$

aufstellen.

Welche Bedingungen müssen wir an  $B_{k+1}$  stellen? Eine sinnvolle Anforderung an  $B_{k+1}$  ist, dass der Gradient von  $m_{k+1}$  dem Gradienten der Zielfunktion  $f$  an den letzten zwei Iterationen  $x_k$  und  $x_{k+1}$  entspricht. D.h. wir haben zwei Bedingungen die erfüllt werden müssen:

1.  $\nabla m_{k+1}$  entspricht  $\nabla f_k$ :  
diese Bedingung wird für  $p = -\alpha_k p_k$  erfüllt,  
i.e.  $\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k$
2.  $\nabla m_{k+1}$  entspricht  $\nabla f_{k+1}$ :  
diese Bedingung wird für  $p = 0$  erfüllt.

Daraus folgt

$$B_{k+1} \alpha_k p_k = \nabla f_{k+1} - \nabla f_k \tag{5}$$

Um die Schreibweise zu vereinfachen, bestimmen wir folgende Notationen für  $s_k$  und  $y_k$ :

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f_{k+1} - \nabla f_k \tag{6}$$

Setzen wir diese Definitionen in die Gleichung (5) ein, erhalten wir

$$B_{k+1} s_k = y_k \tag{7}$$

Diese Formel nennt man die Sekantengleichung oder die Quasi-Newton-Gleichung.

Die Sekantengleichung (7) erfordert, dass die positiv definite und symmetrische Matrix  $B_{k+1}$ ,  $s_k$  in  $y_k$  abbildet. Dies ist nur möglich, wenn  $s_k$  und  $y_k$  die Krümmungsbedingung

$$s_k^T y_k > 0, \tag{8}$$

erfüllen. Die Sekantengleichung (7) hat immer eine Lösung  $B_{k+1}$  wenn die Krümmungsbedingung (8) erfüllt wird.

Doch wann genau erfüllen wir die Krümmungsbedingung (8)? Angenommen  $f$  ist eine strikt konvexe Funktion, dann gilt die Ungleichung (8) für beliebige Punkte  $x_k$  und  $x_{k+1}$ . Jedoch für nicht konvexe Funktionen wird diese Bedingung nicht immer erfüllt. Daher beschränken wir das Abstiegsverfahren zur Bestimmung von  $\alpha$ . Die Krümmungskondition wird garantiert erfüllt, wenn wir dem Abstiegsverfahren die Wolfe-Konditionen (4a) und (4b) aufzwingen. Denn aus (6) und (4b) gilt  $\nabla f_{k+1}^T s_k \geq c_2 \nabla f_k^T s_k$ . Daraus folgt

$$y_k^T s_k \geq (c_2 - 1) \alpha_k \nabla f_k^T p_k, \tag{9}$$

wobei  $c_2 < 1$  und  $p_k$  ist die Abstiegsrichtung. Der Term  $(c_2 - 1) \alpha_k \nabla f_k^T p_k$  wird positiv sein, also ist die Krümmungsbedingung erfüllt.

Um  $B_{k+1}$  eindeutig zu bestimmen, führen wir eine zusätzliche Bedingung ein, die erfüllt werden muss. Wir wählen die Matrix  $B_{k+1}$ , welche die Krümmungsbedingung (8) erfüllt, die am nächsten der Matrix  $B_k$  ist. D.h. wir lösen das Problem

$$\min_B \|B - B_k\| \quad \text{unterliegt} \quad (10a)$$

$$B = B^T, \quad B s_k = y_k \quad (10b)$$

wobei  $B_k$  die positiv definite, symmetrische Matrix ist und  $s_k, y_k$  die Bedingung (8) erfüllen. Um das Minimisationproblem (10a),(10b) einfach zu lösen, benützen wir eine Gewichtsmatrix und die Frobenius-Norm

$$\|A\|_W \equiv \|W^{\frac{1}{2}} A W^{\frac{1}{2}}\|_F \quad (11)$$

wobei  $\|\cdot\|_F$  durch  $\|C\|_F^2 = \sum_{i=1}^n \sum_{j=1}^n c_{ij}^2$  definiert ist und das Gewicht  $W$  ist eine beliebige Matrix, die  $W y_k = s_k$  erfüllt.

Wir wählen  $W = \bar{G}_k^{-1}$ , wobei

$$\bar{G}_k = \left[ \int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) d\tau \right]$$

der Durchschnitt der hessischen Matrix ist, für die die Sekantenrelation selbst erfüllt ist.

Mithilfe der gewichteten Matrix  $W$  und der Frobenius-Norm erhalten wir die eindeutig bestimmte Lösung von (10a),(10b)

$$(DFP) \quad B_{k+1} = (I - \gamma_k y_k s_k^T) B_k (I - \gamma_k s_k y_k^T) + \gamma_k y_k y_k^T, \quad (12)$$

mit  $\gamma_k = \frac{1}{y_k^T s_k}$  und  $y_k = \bar{G}_k \alpha_k p_k = \bar{G}_k s_k$ .

Dies ist die sogenannte Davidon-Fletcher-Powell-Formel. Sie wurde 1959 von Davidon vorgeschlagen und studiert, ausgebaut und veröffentlicht von Fletcher und Powell.

Sei nun  $H_k$  die inverse Matrix von  $B_k$ , i.e.  $H_k = B_k^{-1}$ . Dies ist sehr hilfreich für die Implementation der Methode, weil die Berechnung der Suchrichtung  $p_k$  in (3) durch eine einfache Matrix-Vektor-Multiplikation erfolgt. Mithilfe der inversen Hessian-Approximation  $H_k$  können wir  $B_k$  in (12) aktualisieren und erhalten

$$(DFP) \quad H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}. \quad (13)$$

Die DFP-Formel ist effizient, wurde aber von der BFGS-Formel abgelöst. Die BFGS-Formel wird als die Effektivste aller Quasi-Newton-Formel betrachtet. Anstatt Bedingungen an die hessische Approximationsmatrix  $B_k$  zu stellen, wie wir dies in (10a) und (10b) getan haben, stellen wir diese an deren inverse Matrix  $H_k$ :

$$\min_H \|H - H_k\| \quad (14a)$$

$$H = H^T, \quad H y_k = s_k. \quad (14b)$$

Die aufdatierte Approximation  $H_{k+1}$  muss symmetrisch, positiv definit sein und die Sekantengleichung  $H_{k+1} y_k = s_k$  erfüllen. Wiederum arbeiten wir mit einer beliebigen Gewichtsmatrix  $W$ , die  $W, W s_k = y_k$  erfüllt, und der Frobenius-Norm. Angenommen  $W$  wird durch den

Durchschnitt der hessischen Matrix  $\bar{G}_k$  gegeben, dann erhalten wir die eindeutig bestimmte Lösung

$$(BFGS) \quad H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \quad (15)$$

wobei

$$\rho_k = \frac{1}{y_k^T s_k}. \quad (16)$$

Doch wie wählen wir  $H_0$ ? Es existiert keine allgemein gültige Formel, um diese Matrix zu bestimmen. Wir können ganz einfach die Identitätsmatrix oder ein Mehrfaches dieser nehmen, oder durch spezielle Informationen die approximierte inverse hessische Matrix erstellen.

## 2.2 Der BFGS Algorithmus

**Algorithmus 1** (Das BFGS Verfahren). Gegeben sei der Startpunkt  $x_0$ , die tolerante Konvergenz  $\epsilon > 0$ , die approximierte inverse Hesse-Matrix  $H_0$ ;

$k \leftarrow 0$ ;

**while**  $\|\nabla f_k\| > \epsilon$  **do**

Berechne die Suchrichtung  $p_k = -H_k \nabla f_k$ ;

Setze  $x_{k+1} = x_k + \alpha_k p_k$ , wobei  $\alpha_k$  mit dem Abstiegsverfahren berechnet wird und die Wolfe Bedingungen (4a) und (4b) erfüllt;

Definiere  $s_k = x_{k+1} - x_k$  und  $y_k = \nabla f_{k+1} - \nabla f_k$ ;

Berechne  $H_{k+1}$  anhand von (15);

$k \leftarrow k + 1$ ;

**end while**

Jede Iteration benötigt nur  $O(n^2)$  Operationen, im Gegensatz zum Newton-Verfahren ( $O(n^3)$  Operationen). Der Algorithmus ist sehr stabil und er besitzt eine superlineare Konvergenz. Er bietet jedoch keine quadratische Konvergenz wie das Newton-Verfahren. Ein weiterer Vorteil der BFGS Methode ist es, dass sie keine zweiten Ableitungen berechnet.

## 2.3 Eigenschaften der BFGS Methode

Die BFGS Methode konvergiert superlinear. Das Minimierungsproblem (14a) und (14b), aus welchem wir die BFGS-Formel herleiten, erfordert nicht explizit, dass die approximierte Hesse-Matrix positiv definit ist. Doch wann ist  $H_{k+1}$  positiv definit?

**Behauptung 2** (Positive Definitheit von  $H_{k+1}$ ). Ist  $H_k$  positiv definit, dann auch  $H_{k+1}$ .

*Beweis.* Aus (9) wissen wir, dass  $y_k^T s_k$  positiv ist, also sind die Formel (15) und (16) wohldefiniert. Für jeden Vektor  $z \neq 0$  haben wir

$$z^T H_{k+1} z = w^T H_k w + \rho_k (z^T s_k)^2 \geq 0,$$

mit  $w = z - \rho_k y_k (s_k^T z)$ .

$w^T H_k w + \rho_k (z^T s_k)^2$  ist nur dann 0, wenn  $s_k^T z = 0$ . In diesem Falle müsste gelten  $w = z \neq 0$  und daraus folgt, dass  $z^T H_{k+1} z > 0$ . Folglich ist  $H_{k+1}$  positiv definit.  $\square$

Das BFGS-Verfahren, angewandt auf die quadratische Funktion, hat viele interessante Eigenschaften. Diese werden wir im allgemeinen Fall unter dem Kapitel (4) sehen.

Es ist sinnvoller uns die Frage zu stellen, in welchem Fall das BFGS-Verfahren schlechte Resultate liefert. Also was wenn bei einer Iteration die Matrix  $H_k$  eine schlechte Approximation ist, können wir diese korrigieren? Wenn  $y_k^T s_k$  sehr klein ist (jedoch immer noch positiv), dann wird  $H_{k+1}$  sehr gross. Ist diese Eigenschaft sinnvoll? Gibt es Rundungsfehler, die möglicherweise nützliche Informationen für die approximierten Quasi-Newton-Matrix löschen?

Diese Fragen wurden analytisch und versuchsweise studiert. Heutzutage wissen wir, dass die BFGS-Formel sehr effektive Eigenschaften der Selbstkorrektur besitzen. Zudem ist die DFP Methode weniger effizient bei der Korrektur einer schlechten Approximation der Hesse-Matrix. Eine weitere Eigenschaft ist, dass die DFP- und die BFGS-Formeln zueinander duale Aufdatierungsformeln sind, i.e. indem man die Tripel  $(H, s, y)$ ,  $(B, y, s)$  vertauscht, erhält man die jeweils entgegengesetzte Formel.

## 2.4 Implementation

Um eine effiziente Implementation zu erhalten, müssen wir den Algorithmus (1) erweitern. Das Abstiegsverfahren, das die Wolfe-Bedingungen (4a) und (4b) erfüllt, sollte zuerst immer mit der Schrittweite  $\alpha_k = 1$  berechnet werden. Diese Schrittweite wird meistens akzeptiert und dabei wird superlineare Konvergenz über dem ganzen Algorithmus erzeugt. Für die Konstanten  $c_1$  und  $c_2$  in den Wolfe-Bedingungen werden meistens die Werte  $c_1 = 10^{-4}$ ,  $c_2 = 0.9$  benutzt. Als Anfangsmatrix  $H_0$  wird oft ein Vielfaches der Identitätsmatrix benutzt, i.e.  $H_0 = \beta I$ . Es gibt jedoch keine allgemeine Regel um dieses  $\beta$  zu bestimmen. Wählen wir  $\beta$  zu gross, dann wird unser erster Schritt  $p_0 = -\beta g_0$  zu lang. Bei einigen Programmen definiert der Benutzer  $\delta$  für die Norm der ersten Evaluation und setzt dann  $H_0 = \delta \|g_0\|^{-1} I$  um die Norm zu erhalten. Eine weitere Möglichkeit besteht darin, die Startmatrix erst nach dem ersten Verfahrensschritt, jedoch noch vor der ersten BFGS Aufdatierung zu bestimmen. D.h. der provisorische Wert  $H_0 = I$  wird durch

$$H_0 \leftarrow \frac{y_k^T s_k}{y_k^T y_k} I, \quad (17)$$

ersetzt und bevor wir die BFGS-Formel (15) mit (16) aufdatieren, erhalten wir  $H_1$ .

Wir können das BFGS-Verfahren mit der hessischen Approximationsmatrix  $B_k$  anstelle  $H_k$  schreiben und erhalten

$$(BFGS) \quad B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k - k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \quad (18)$$

Eine effiziente Implementation dieser Annäherung speichert  $B_k$  nicht explizit, sondern die Cholesky-Faktorisierung  $L_k D_k L_k^T$  dieser Matrix. Eine Formel zur Aufdatierung der Faktoren  $L_k$  und  $D_k$  können wir aus (18) herleiten. Diese Berechnungen werden in  $O(n^2)$  Operationen durchgeführt, also erhalten wir hierbei ähnliche Berechnungskosten wie im Algorithmus 1. Erfahrungen mit der Berechnung dieser Art haben gezeigt, dass diese Methode keine speziellen Vorteile erbringt.

## 3 Die SR1 Methode

In der BFGS und DFP Formel weicht die aufdatierte Matrix  $B_{k+1}$  oder  $H_{k+1}$  durch eine Rang-2-Matrix von  $B_k$  bzw.  $H_k$  ab. Es existiert jedoch eine einfachere Rang-1-Modifikation, welche die Symmetrie der Matrix beibehält und zur selben Zeit die Sekantengleichung (7)

erfüllt. Im Gegenteil zur Rang-2-Modifikation garantiert das symmetrische Rang-1-Verfahren, kurz SR1, keine positive Definitheit der Matrix. Viele numerische Resultate beziehen sich auf den SR1-Algorithmus, darum leiten wir diesen her.

### 3.1 Die Herleitung der SR1-Formeln

Die SR1-Modifikation wird allgemein geschrieben als

$$B_{k+1} = B_k + \sigma v v^T,$$

wobei  $\sigma = \pm 1$  und  $\sigma, v$  sind so gewählt, dass  $B_{k+1}$  die Sekantengleichung  $B_{k+1}s_k = y_k$  erfüllt. Ersetzen wir  $B_{k+1}$  durch  $B_k + \sigma v v^T$  in der Sekantengleichung, i.e.  $(B_k + \sigma v v^T)s_k = y_k$ , erhalten wir

$$B_k s_k + [\sigma v^T s_k] v = y_k. \quad (19)$$

Weil  $[\sigma v^T s_k]$  ein Skalar ist, ist  $v$  ein Vielfaches von  $y_k - B_k s_k$ , i.e.  $v = \delta(y_k - B_k s_k)$  für ein Skalar  $\delta$ . Setzen wir  $v$  in (19) ein, erhalten wir

$$\sigma \delta^2 [s_k^T (y_k - B_k s_k)] (y_k - B_k s_k) = (y_k - B_k s_k). \quad (20)$$

Diese Gleichung wird jedoch nur dann erfüllt, wenn wir die Parameter  $\sigma$  und  $\delta$  wie folgt wählen

$$\sigma = \text{sign} [s_k^T (y_k - B_k s_k)], \quad \delta = \pm |s_k^T (y_k - B_k s_k)|^{-\frac{1}{2}}.$$

Die einzige symmetrische Rang-1-Modifikationsformel, die die Sekantengleichung erfüllt ist wird somit gegeben durch

$$(SR1) \quad B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}. \quad (21)$$

Die entsprechende Formel für die approximierte inverse hessische Matrix  $H_k$  lautet

$$(SR1) \quad H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}. \quad (22)$$

Wenn  $B_k$  oder  $H_k$  positiv definit ist, muss dies nicht auch der Fall sein für  $B_{k+1}$  oder  $H_{k+1}$ . Für  $B_k$  oder  $H_k$  gibt es drei verschiedene Fälle, welche wir nun anhand der Matrix  $B_k$  betrachten:

1.  $(y_k - B_k s_k)^T \neq 0$ : es existiert eine eindeutig bestimmte Rang-1-Modifikation gegeben durch (21), die die Sekantengleichung (7) erfüllt;
2.  $y_k = B_k s_k$ : es existiert eine eindeutig bestimmte Rang-1-Modifikation gegeben durch  $B_{k+1} = B_k$ , die die Sekantengleichung (7) erfüllt;
3.  $y_k \neq B_k s_k$  und  $(y_k - B_k s_k)^T = 0$ : es existiert keine eindeutig bestimmte Rang-1-Modifikation welche die Sekantengleichung (7) erfüllt, aufgrund von (20).

In der Praxis wurde beobachtet, dass die SR1 Methode gut ausgeführt wird, wenn der Nenner bei der Modifikation möglichst klein gehalten wird. Um den Abbruch der SR1 Methode zu verhindern benützt man zur Implementation oft eine Regel:

Die Aufdatierung von (21) wird nur dann durchgeführt, wenn gilt

$$|s_k^T (y_k - B_k s_k)| \geq r \|s_k\| \|y_k - B_k s_k\| \quad (23)$$

wobei  $r \in (0, 1)$  ist klein (z.B.  $r = 10^{-8}$ ). Wird (23) nicht eingehalten, dann setzen wir  $B_{k+1} = B_k$ .

### 3.2 Der SR1 Algorithmus

Für den SR1 Algorithmus benutzen wir als Bezugssystem die Trust-Region. Dies aus dem einfachen Grund, dass wir so die hessische Approximationsmatrix nicht verändern müssen, damit sie genügend positiv definit ist.

**Algorithmus 3** (Das SR1 Trust-Region Verfahren). Gegeben sei der Startpunkt  $x_0$ , die tolerante Konvergenz  $\epsilon > 0$ , die approximierte inverse Hesse-Matrix  $B_0$ , der Radius der Trust-Region  $\Delta_0$ , die Parameter  $\eta \in (0, 10^{-3})$  und  $r \in (0, 1)$ ;

$k \leftarrow 0$ ;

**while**  $\|\nabla f_k\| > \epsilon$  **do**

Berechne  $s_k$  mithilfe der Berechnung des Unterproblems

$$\min_s \nabla f_k^T s + \frac{1}{2} s^T B_k s \quad \text{gemäss } \|s\| \leq \Delta_k.$$

Berechne

$$\begin{aligned} y_k &= \nabla f(x_k + s_k) - \nabla f_k, \\ \text{ared} &= f_k - f(x_k + s_k) \quad (\text{aktuelle Reduktion}) \\ \text{pred} &= -(\nabla f_k^T s_k + \frac{1}{2} s_k^T B_k s_k) \quad (\text{erwartete Reduktion}); \end{aligned}$$

**if**  $\text{ared}/\text{pred} > \eta$  **then**

$$x_{k+1} = x_k + s_k$$

**else**

$$x_{k+1} = x_k;$$

**end if**

**if**  $\text{ared}/\text{pred} > 0.75$  **then**

**if**  $\|s_k\| \leq 0.8\Delta_k$  **then**

$$\Delta_{k+1} = \Delta_k$$

**else**

$$\Delta_{k+1} = 2\Delta_k;$$

**end if**

**else if**  $0.1 \leq \text{ared}/\text{pred} \leq 0.75$  **then**

$$\Delta_{k+1} = \Delta_k$$

**else**

$$\Delta_{k+1} = 0.5\Delta_k$$

**end if**

**if** wird (23) eingehalten **then**

benütze (21) um  $B_{k+1}$  zu berechnen, auch wenn gilt  $x_{k+1} = x_k$ .

**else**

$$B_{k+1} \leftarrow B_k;$$

**end if**

$k \leftarrow k + 1$ ;

**end while**

### 3.3 Eigenschaften der SR1 Modifikation

Zu den grössten Vorteilen der SR1 Modifikation ist, dass sie die hessische Matrix sehr gut approximiert. Betrachten wir nämlich eine quadratische Funktion mit der Schrittweite  $\alpha_k = 1$

erhalten wir

$$p_k = -H_k \nabla f_k, \quad x_{k+1} = x_k + p_k. \quad (24)$$

Daraus folgt dass  $p_k = s_k$ .

**Satz 4.** *Angenommen  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  ist die streng konvexe quadratische Funktion  $f(x) = b^T x + \frac{1}{2} x^T A x$ , wobei  $A$  symmetrisch und positiv definit. Wenn  $\forall k$  gilt  $(s_k - H_k y_k)^T \neq 0$ , dann konvergiert für jeden beliebigen Startpunkt  $x_0$  und für jede beliebige symmetrische Anfangsmatrix  $H_0$  die Iterationen  $\{x_k\}$ , erstellt durch die SR1 Methode (22), (24), zum Minimierer in mindestens  $n$  Schritten.*

*Ferner gilt, wenn nach  $n$  Schritten die Suchrichtungen  $p_i$  linear unabhängig sind, dann  $H_n = A^{-1}$ .*

*Beweis.* Die SR1 Modifikation ist aufgrund der Voraussetzung  $(s_k - H_k y_k)^T y_k \neq 0, \forall k$  immer wohldefiniert. Durch Induktion zeigen wir dass gilt

$$H_k y_j = s_j \quad \text{für } j = 0, \dots, k-1, \quad (25)$$

i.e. wir verlangen, dass die Sekantengleichung nicht nur für die letzte Suchrichtung gilt, sondern für alle Suchrichtungen. Die SR1 Modifikation erfüllt laut Definition die Sekantengleichung, also haben wir  $H_1 y_0 = s_0$ . Angenommen (25) gilt für  $k > 1$ , also zeigen wir, dass es ebenfalls für  $k+1$  gilt. Wir haben

$$(s_k - H_k y_k)^T y_j = s_k^T y_j - y_k^T (H_k y_j) = s_k^T y_j - y_k^T s_j \quad \forall j < k, \quad (26)$$

weil für die quadratische Funktion  $y_i = A s_i$  gilt, erhalten wir  $s_k^T y_j - y_k^T s_j = 0$ .

Daraus folgt  $H_{k+1} y_j = H_k y_j = s_j, \forall j < k$ . Aus der Sekantengleichung wissen wir dass gilt  $H_{k+1} y_k = s_k$ . und (25) für  $k+1$ . Also gilt (25) für alle  $k$ .

Führt der Algorithmus  $n$  Schritte aus und sind die Schritte  $\{s_j\}$  linear unabhängig erhalten wir

$$s_j = H_n y_j = H_n A s_j, \quad \text{für } j = 0, \dots, n-1.$$

Daraus folgt dass  $H_n A = I$ , also  $H_n = A^{-1}$ . Folglich ist der Schritt bei  $x_n$  der Newton-Schritt und die nächste Iteration  $x_{n+1}$  die Lösung. Somit ist auch der Algorithmus hier beendet.

Sind jedoch die Schritte  $\{s_j\}$  linear abhängig, können wir  $s_k$  als Linearkombination

$$s_k = \xi_0 s_0 + \dots + \xi_{k-1} s_{k-1}, \quad (27)$$

schreiben, wobei  $\xi_i$  Skalare sind. Aus (27) und (25) erhalten wir

$$\begin{aligned} H_k y_k &= H_k A s_k \\ &= \xi_0 H_k A s_0 + \dots + \xi_{k-1} H_k A s_{k-1} \\ &= \xi_0 H_k y_0 + \dots + \xi_{k-1} H_k y_{k-1} \\ &= \xi_0 s_0 + \dots + \xi_{k-1} s_{k-1} \\ &= s_k. \end{aligned}$$

Weil  $y_k = \nabla f_{k+1} - \nabla f_k$  und  $s_k = p_k = -H_k \nabla f_k$  erhalten wir  $H_k (\nabla f_{k+1} - \nabla f_k) = -H_k \nabla f_k$ . Ist nun  $H_k$  nicht singulär, dann ist  $\nabla f_{k+1} = 0$ . Daraus folgt, dass  $x_{k+1}$  der Lösungspunkt ist.  $\square$

**Satz 5.** *Angenommen  $f$  ist zweimal stetig differenzierbar, die hessische Matrix beschränkt und Lipschitz-stetig an der Umgebung eines Punktes  $x^*$ . Sei  $\{x_k\}$  eine Folge von Iterationen s.d.  $x_k \rightarrow x^*$  für ein  $x^* \in \mathbb{R}^n$ . Angenommen (23) gilt für alle  $k$  und für einige  $r \in (0, 1)$  und die Schritte  $s_k$  sind gleichmässig linear unabhängig. Dann erfüllen die durch die SR1 modifizierten Matrizen  $B_k$*

$$\lim_{x \rightarrow \infty} \|B_k - \nabla^2 f(x^*)\| = 0.$$

## 4 Die Broyden-Klasse

Bis jetzt haben wir die BFGS, DFP und SR1 Verfahren beschrieben. Mit der sogenannten Broyden-Formel erhalten wir eine ganze Klasse von Quasi-Newton-Verfahren.

### 4.1 Die Formeln der Broyden-Klasse

Die Broyden-Klasse ist eine Familie, die mit der folgenden Formel rechnet:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \phi_k (s_k^T B_k s_k) v_k v_k^T, \quad (28)$$

wobei  $\phi_k$  ein Skalar ist und

$$v_k = \left[ \frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k} \right]. \quad (29)$$

Setzen wir nun in (28)  $\phi_k = 0$  dann erhalten wir die BFGS-Formel und bei  $\phi_k = 1$  die DFP-Formel. Also gehören das BFGS und das DFP Verfahren zur Broyden-Klasse. Dadurch können wir (20) auch als Linearkombination dieser beiden Formeln schreiben und erhalten

$$B_{k+1} = (1 - \phi_k) B_{k+1}^{BFGS} + \phi_k B_{k+1}^{DFP}.$$

Viel Aufmerksamkeit wurde der sogenannten eingeschränkten (auch konvexen) Broyden-Klasse zugewandt. Hierbei wird  $\phi_k$  auf das Intervall  $[0, 1]$  beschränkt. Wenn wir dies an einer quadratischen Funktion anwenden, erhalten wir den Satz 6. Einfachheitshalber nehmen wir an, dass jede Iteration die Form

$$p_k = -B_k^{-1} \nabla f_k, \quad x_{k+1} = x_k + p_k \quad (30)$$

besitzt.

**Satz 6.** *Angenommen  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  ist die streng konvexe quadratische Funktion  $f(x) = b^T x + \frac{1}{2} x^T A x$ , wobei  $A$  symmetrisch und positiv definit. Sei  $x_0$  ein beliebiger Startpunkt für die Iteration (30) und sei  $B_0$  die positiv definite, symmetrische Startmatrix. Angenommen die Matrizen  $B_k$  werden anhand der Broyden-Formel (28) aufdatiert mit der Eigenschaft  $\phi_k \in [0, 1]$ .*

*Definiere  $\lambda_1^k \leq \lambda_2^k \leq \dots \leq \lambda_n^k$  als die Eigenwerte der Matrix*

$$A^{\frac{1}{2}} B_k^{-1} A^{\frac{1}{2}}. \quad (31)$$

*Dann gilt für alle  $k$*

$$\min\{\lambda_i^k, 1\} \leq \lambda_i^{k+1} \leq \max\{\lambda_i^k, 1\}, \quad i = 1, \dots, n. \quad (32)$$

*Zu beachten ist, dass wenn  $\phi_k \notin [0, 1]$ , dann gilt die Eigenschaft (32) nicht.*

Dieser Satz besagt uns, dass wenn die Eigenwerte  $\lambda_i^k$  der Matrix (31) alle gleich 1 sind, dann ist die Quasi-Newton-Approximation  $B_k$  identisch der Hesseschen Matrix  $A$  der quadratischen Zielfunktion. Dies wäre die ideale Situation, also hoffen wir, dass unsere Eigenwerte möglichst nahe von 1 sind. (32) besagt uns, dass die Eigenwerte  $\{\lambda_i^k\}$  monoton, jedoch nicht streng monoton gegen 1 konvergieren.

## 4.2 Eigenschaften der Broyden-Klasse

Wenn  $B_k$  positiv definit ist und wenn gilt  $y_k^T s_k > 0$ ,  $\phi_k \geq 0$ , wobei  $\phi_k \in [0, 1]$ , dann ist für eine Aktualisierung mit der Broyden-Klasse die Matrix  $B_{k+1}$  ebenfalls positiv definit. Weiter wissen wir, wenn die Hesse-Matrix  $B_0$  symmetrisch positiv ist und für alle  $k$  gilt  $s_k^T y_k > 0$  und  $\phi_k > \phi_k^c$ , wobei

$$\phi_k^c = \frac{1}{1 - \mu_k}, \quad \text{mit } \mu_k = \frac{(y_k^T B_k^{-1} y_k)(s_k^T B_k s_k)}{(y_k^T s_k)^2}, \quad (33)$$

gilt. Dann sind alle Matrizen  $B_k$  die mithilfe einer Broyden-Formel berechnet wurden symmetrisch und positiv definit.

Wenden wir die Broyden-Klasse mit dem exakten Abstiegsverfahren an einer quadratischen Funktion an, erhalten wir einige Eigenschaften, die unter anderem im folgenden Satz beschrieben werden.

**Satz 7.** *Sei  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  die streng konvexe quadratische Funktion, wobei  $x_0$  ein beliebiger Startpunkt und  $B_0$  eine positiv definite, symmetrische Startmatrix ist. Angenommen wir wenden eine Methode der Broyden-Klasse an  $f$  an. Vorausgesetzt wird, dass  $\alpha_k$  die exakte Schrittweite und  $\phi_k \geq \phi_k^c$  für alle  $k$  ist. Dann gelten folgende Aussagen:*

- i Die iterierten Werte konvergieren zur Antwort in höchstens  $n$  Iterationen.*
- ii Die Sekantengleichung wird für alle Suchrichtungen erfüllt, i.e.*

$$B_k s_j = y_j \quad j = k - 1, \dots, 1.$$

- iii Ist die Startmatrix  $B_0 = I$ , dann sind die iterierten Werte identisch zu denen, die wir beim Verfahren der konjugierten Gradienten (CG-Verfahren) erhalten. Insbesondere haben wir konjugierte Suchrichtungen, i.e.  $s_i^T A s_j = 0$  für  $i \neq j$ . wobei  $A$  die hessische Matrix der quadratischen Funktion ist;*

- iv Werden  $n$  Iterationen durchgeführt, haben wir  $B_{n+1} = A$ .*

## Literatur

- [1] Jorge Nocedal and Stephen J. Wright: Numerical Optimization, Springer Series in Operations Research, Springer 1999.
- [2] Webseite des Proseminars: <http://perso.unifr.ch/ales.janka/numeroptim>.
- [3] Carl Geiger und Christian Kanzow: Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben, Springer-Lehrbuch, Springer 1999.
- [4] P. Spellucci: Numerische Verfahren der nichtlinearen Optimierung, Internationale Schriftreihe zur numerischen Mathematik: Lehrbuch, Birkhäuser 1993.