Prof. Dr. Dirk Lebiedz
M.Sc. Pascal Heiter

Universität Ulm
Institut für Numerische Mathematik
Sommersemester 2013

Nichtlineare Optimierung und optimale Steuerung - Übungsblatt 8

(Besprechung: Montag, 10. Juni 2013)

Aufgabe 28 *(Normalität)*

Zeigen Sie, dass für das Standardproblem der nichtlinearen Optimierung im $\mathbb{R}^n$ folgende Äquivalenz gilt:

$$\bar{x} \text{ erfüllt LICQ} \quad \Leftrightarrow \quad \bar{x} \text{ ist normal.}$$

Aufgabe 29 *(Konvergenz des Newtonverfahrens)*

Gegeben sei folgendes Optimierungsproblem

$$\min_{x\in\mathbb{R}} f(x) := x \arctan x - \frac{1}{2}\log(x^2+1).$$

In Aufgabe 27 d) haben Sie die Divergenz des Newtonverfahrens für den Startwerte $x_0 = 4$ bemerkt.

a) Implementieren Sie das Newtonverfahren in `MATLAB` in einer Datei `newtonMethod.m` unter Verwendung der Armijo-Regel zur Schrittweitensteuerung (siehe Aufgabe 20) mit folgendem Funktionsaufruf

```
[X,fx,nit] = newtonMethod(f,gradf,hessf,x,tol,sigma,beta)
```

mit den Parametern

- `f` - die Funktion $f : \mathbb{R}^n \to \mathbb{R}$ als *function handle*
- `gradf` - der Gradient von f als *function handle*
- `hessf` - die Hesse-Matrix von f als *function handle*
- `x` - der Startpunkt $x \in \mathbb{R}^n$
- `tol` - die Toleranz für das Abbruchkriterium
- `sigma,beta` - die Konstanten $\sigma \in (0,1)$ und $\beta \in (0,1)$ für die Armijo-Regel

Zurückgegeben werden soll der gesamte Pfad, d.h. `X` $= [x_0, x_1, x_2, ..., x^*]$, der optimale Funktionswert `fx` $= f(x^*)$ und die Anzahl der Iterationsschritte `nit`.

b) Schreiben Sie ein Skript `test_newton_ls.m`, dass das gegebene Optimerungsproblem löst und verwenden Sie $x_0 = 4$ als Startwert und `tol` $= 10^{-12}$. Diskutieren Sie die numerischen Ergebnisse. Was fällt auf?

Aufgabe 30 *(Quasi-Newton-Verfahren)*

Die Wahl der Suchrichtung d im Punkt x_k ist charakteristisch für numerische Optimierungsalgorithmen. Dabei haben Sie die folgenden zwei Verfahren bereits kennen gelernt:

- Gradientenverfahren: $I_{n\times n}d = -\nabla f(x_k)$
- Newtonverfahren: $\nabla^2 f(x_k)d = -\nabla f(x_k)$.

Ein Quasi-Newton-Verfahren verzichtet explizit auf die Hessematrix und approximiert diese in jedem Schritt. Demnach wird das Gleichungssystem

$$W_k d = -\nabla f(x_k)$$

gelöst, eine zulässige Schrittweite t bestimmt und $x_{k+1} = x_k + td$ berechnet. Danach wird eine Aktualisierung der Approximation der Hessematrix vorgenommen, durch folgende Rechenvorschrift nach Broyden-Fletcher-Goldfarb-Shanno

$$W_{k+1} = W_k + \frac{y_k y_k^T}{y_k^T s} - \frac{W_k(ss^T)W_k}{s^T W_k s} \qquad \textbf{(BFGS)}$$

mit $y_k := \nabla f(x_{k+1}) - \nabla f(x_k)$ und $s = td$. Die Wahl von W_0 ist dabei beliebig. Wählt man $W_0 = I_{n\times n}$, so entspricht der erste Schritt genau dem ersten Schritt des Gradientenverfahrens.

a) Implementieren Sie das Quasi-Newtonverfahren in `MATLAB` in einer Datei `quasiNewtonMethod.m` unter Verwendung der Armijo-Regel zur Schrittweitensteuerung (siehe Aufgabe 20) mit folgendem Funktionsaufruf

```
[X,fx,nit] = quasiNewtonMethod(f,gradf,x,tol,sigma,beta)
```

mit den Parametern

- `f` - die Funktion $f : \mathbb{R}^n \to \mathbb{R}$ als *function handle*
- `gradf` - der Gradient von f als *function handle*
- `x` - der Startpunkt $x \in \mathbb{R}^n$
- `tol` - die Toleranz für das Abbruchkriterium
- `sigma,beta` - die Konstanten $\sigma \in (0,1)$ und $\beta \in (0,1)$ für die Armijo-Regel

Zurückgegeben werden soll der gesamte Pfad, d.h. `X` $= [x_0, x_1, x_2, ..., x^*]$, der optimale Funktionswert `fx` $= f(x^*)$ und die Anzahl der Iterationsschritte `nit`. Wählen Sie für die Approximation der Hessematrix oben genannte Vorschrift (BFGS) und setzen Sie $W_0 = I_{n\times n}$.

b) Ändern Sie ihre Methode

```
[X,fx,nit] = gradientMethod(f,gradf,x,sigma,beta,tol)
```

in `gradientMethod.m` derart ab, sodass ebenfalls der gesamte Pfad, d.h. `X` $= [x_0, x_1, x_2, ..., x^*]$ zurück gegeben wird.

c) Laden Sie die Datei `compareMethods.m` von der Vorlesungshomepage herunter und testen Sie das Skript mit folgenden Parametern

- $\sigma = 0.5$, $\beta = 0.5$ und $x_0 = (1.9, 2)^T$
- $\sigma = 0.9$, $\beta = 0.2$ und $x_0 = (1.9, 2)^T$
- $\sigma = 0.1$, $\beta = 0.5$ und $x_0 = (0, 2)^T$.

Diskutieren Sie die numerischen Ergebnisse und erläutern Sie die Stärken und Schwächen der einzelnen Verfahren.

Mehr Informationen zur Vorlesung und den Übungen finden Sie auf

`http://www.uni-ulm.de/mawi/mawi-numerik/lehre/sommersemester-2013/nouos.html`