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

Universität Ulm
Institut für Numerische Mathematik
Sommersemester 2013

Nichtlineare Optimierung und optimale Steuerung - Übungsblatt 6

(Besprechung: Montag, 27. Mai 2013)

Aufgabe 20 *(Schrittweitensteuerung von Armijo)*

Sei $f : \mathbb{R}^n \to \mathbb{R}$ mit $f \in C^1(\mathbb{R})$, sowie $\sigma \in (0,1)$ und $\beta \in (0,1)$ fest. Dann wählt man die Schrittweite t nach der Armijoregel für die Startschrittweite $s \in \mathbb{R}^n$ und der Abstiegsrichtung $d \in \mathbb{R}^n$ im Punkt $x \in \mathbb{R}^n$ wie folgt:

$$t_A := \max\{s\beta^\ell \,:\, \ell = 0, 1, \ldots \,:\, f(x + s\beta^\ell d) \le f(x) + \sigma s \beta^\ell \nabla f(x)^T d\}.$$

Siehe dazu auch Paragraph 5.3 im Skript dazu.

a) Implementieren Sie die Armijo-Schrittweitenstrategie in `MATLAB` in einer Datei `armijo.m` mit folgendem Funktionsaufruf

```
function t = armijo(f,gradf,x,d,s,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 $x \in \mathbb{R}^n$
- `d` - die Suchrichtung $d \in \mathbb{R}^n$
- `s` - die Startschrittweite $s > 0$
- `sigma,beta` - die Konstanten $\sigma \in (0,1)$ und $\beta \in (0,1)$ für die Armijoregel

Zurückgegeben werden soll eine Schrittweite `t` $= t_A$, welche die Armijo-Bedingung erfüllt.

b) Schreiben Sie ein Skript `test_armijo.m` welches die Armijoschrittweite für die Rosenbrock Funktion $f : \mathbb{R}^2 \to \mathbb{R}$

$$f(x_1, x_2) := (1 - x_1)^2 + 100(x_2 - x_1^2)^2$$

mit folgenden Eingabeparametern

i) $x = (1.7, 1.5)^T, d = (-1, 0)^T, s = 4, \sigma = 0.1$ und $\beta = 0.5$
ii) $x = (0, 0)^T, d = (1, 0)^T, s = 1, \sigma = 0.1$ und $\beta = 0.5$

berechnet.

c) Plotten Sie die Rosenbrock-Funktion und diskutieren Sie die numerischen Ergebnisse.

Aufgabe 21 *(Innere-Punkte-Verfahren)*

a) Implementieren Sie ein Inneres-Punkte-Verfahren in `MATLAB` in einer Datei `interior_point.m` mit folgendem Funktionsaufruf

```
function [x,lambda,s,fx,nit] = interior_point(A,b,c,x,lambda,s,tol,sigma,alpha),
```

mit den Parametern

- `A` - die Matrix $A \in \mathbb{R}^{m \times n}$
- `b` - die rechte Seite $b \in \mathbb{R}^m$ der Gleichungsbeschränkung $Ax = b$
- `c` - der Kostenvektor $c \in \mathbb{R}^n$
- `x` - der Anfangswert $x_0 \in \mathbb{R}^n$
- `lambda` - der Anfangswert $\lambda_0 \in \mathbb{R}^m$
- `s` - der Anfangswert $s_0 \in \mathbb{R}^n$
- `tol` - die Toleranz für die Abschätzung der dualen Lücke
- `sigma` - der Zentrierungsparameter $\sigma \in [0, 1]$
- `alpha` - der Dämpfungsparameter $\alpha \in (0, 1]$ in der Schrittweitensteuerung

Zurückgegeben werden soll das Optimum `(x,lambda,s)` $= (x^*, \lambda^*, s^*)$, der optimale Funktionswert `fx` und die Anzahl der Iterationsschritte `nit`.

b) Laden Sie die Dateien `test_interiorpoint.m` und `output_ip.txt` von der Vorlesungshomepage herunter und testen Sie damit ihre Funktion. Die Datei `output_ip.txt` enthält die korrekte Ausgabe des Skriptes.

c) Schreiben Sie ein Skript `test_interiorpoint2.m`, das folgende lineare Programm löst

$$
\textbf{(LP)} \begin{cases}
\min\limits_{x_1,x_2,x_3} \; -5x_1 - 4x_2 - 6x_3 \\
\text{s.t.} \quad \begin{array}{rcrcrcl}
x_1 & - & x_2 & + & x_3 & \leq & 20 \\
3x_1 & + & 2x_2 & + & 4x_3 & \leq & 42 \\
3x_1 & + & 2x_2 & & & \leq & 30 \\
 & & & & x_1, x_2, x_3 & \geq & 0
\end{array}
\end{cases}
$$

mit von Ihnen gewählten Startwerten. Verifizieren Sie das Ergebnis, indem Sie das Optimierungsproblem ebenfalls mit `AMPL` lösen.