Universität Konstanz
Fachbereich Mathematik und Statistik
Prof. Dr. Stefan Volkwein
Tim Seger

14. Mai 2009

Optimierung
2. Übungsblatt

Alle Aufgaben sind zum Vortragen in den Übungen am Dienstag, 26. Mai 2009, vorzubereiten. Die mit ✏ gekenntzeichneten Aufgaben sind auch schriftlich auszuarbeiten und bis Freitag, 22. Mai 2009, 10 Uhr in den **Briefkasten Nr. 16** abzugeben.

Aufgabe 2.1 ✏ Sei $f : \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar und $X \subset \mathbb{R}^n$ konvex. Wir betrachten das Optimierungsproblem

$$\min f(x) \text{ unter der Nebenbedingung } x \in X. \tag{1}$$

Zeigen Sie:

a) Ist f konvex auf X, so ist die Lösungsmenge von (1) konvex (evtl. leer).

b) Ist f strikt konvex auf X, so besitzt (1) höchstens eine Lösung.

Aufgabe 2.2 Wir betrachten das allgemeine Abstiegsverfahren (Algorithmus 3.4) für die Funktion $f : \mathbb{R} \to \mathbb{R}$, $f(x) = x^2$ mit Startpunkt $x^{(0)} := 1$ und den Richtungen bzw. Schrittweiten

a) $d_k := -1$, $\lambda_k := \left(\frac{1}{2}\right)^{k+2} \quad (k \in \mathbb{N}_0)$,

b) $d_k := (-1)^{k+1}$, $\lambda_k := 1 + \frac{3}{2^{k+2}} \quad (k \in \mathbb{N}_0)$.

Überprüfen Sie, dass diese Wahl der Parameter für $k \in \mathbb{N}_0$ zu einer Verminderung der Zielfunktion führt. Stellen Sie dazu die durch den Algorithmus generierte Folge $x^{(k)}$ mithilfe direkter Rechnungen oder Induktion nur in Abhängigkeit von k dar. Bestimmen Sie jeweils $\lim_{k\to\infty} f(x_k)$ und vergleichen Sie mit dem Minimum von f. Wo liegt der Fehler[1]?

Aufgabe 2.3

a) Betrachten Sie die Funktion $f(x) = (x_1 + x_2^2)^2$ am Punkt $x = (1, 0)^T$. Zeigen Sie, dass $d = (-1, 1)^T$ eine Abstiegsrichtung ist und finden Sie alle Minimalstellen des Problems $\min_{\alpha>0} f(x + \alpha d)$.

b) Betrachten Sie die Funktion $f(x) = 3x_1^4 - 4x_1^2x_2 + x_2^2$. Zeigen Sie, dass $x_0 := (0, 0)^T$ ein stationärer Punkt von f ist und dass f auf jeder Ursprungsgeraden ein striktes lokales Minimum bei x_0 besitzt. Zeigen Sie, dass x_0 jedoch keine lokale Minimalstelle von f ist.

Bitte wenden!

[1] Eine Skizze der Funktion und das Einzeichnen der Lage der ersten Folgeglieder kann hilfreich sein.

Aufgabe 2.4 Programmieraufgabe (in Zweiergruppen zu bearbeiten)

Implementieren Sie die ARMIJO-Schrittweitenwahl aus der Vorlesung in Matlab. Erstellen Sie dazu eine Datei `armijo.m` und verwenden Sie als erste Zeile `function t = armijo(fhandle, x0, d, t0, sigma, beta)`. Dabei bezeichnet `fhandle` das Handle auf eine Funktion (`help function_handle` und `doc function_handle` Beispiel: `@sin`), `x0` den Ausgangspunkt, `d` eine Abstiegsrichtung, `t0` die Startschrittweite und `alpha` und `beta` die in der Vorlesung eingeführten Parameter des Algorithmus. Zurückgegeben werden soll eine Schrittweite `t`, welche die Armijo-Bedingung erfüllt.

Testen Sie die Funktion an folgenden Funktionen und Parametern (schreiben Sie hierzu einzelne Funktionen `rosenbrock.m` o.ä., die Sie dann in einem kurzen „Hauptprogramm" aufrufen. Tipp: Hier ist es günstig, wenn mit dem Funktionwert auch der Wert der Ableitung abgerufen werden kann):

a) an der ROSENBROCK-Funktion aus Aufgabe 1.1 mit `x0=[1.7;1.5], d=[-1;0], t0=4, alpha=0.1` und `beta=0.5`.

b) an der ROSENBROCK-Funktion aus Aufgabe 1.1 mit `x0=[0;0], d=[1;0], t0=1`, `alpha=0.1` und `beta=0.5`.

c) an der skalaren Funktion

$$\varphi(x) := \omega(c_1)\sqrt{(1-x)^2 + c_2^2} + \omega(c_2)\sqrt{x^2 + c_1^2},$$

mit $\omega(c) := \sqrt{1+c^2} - c, c_1 := 0.01, c_2 := 0.001,$ `x0=0, d=1, t0=1, beta=0.5`. Variieren Sie `alpha` zwischen 0.01 und 0.5. Was beobachten Sie?

Implementieren Sie das allgemeine Abstiegsverfahren (3.4) mit der Richtungswahl $d_k := -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$ und der ARMIJO-Schrittweitenstrategie. Erstellen Sie eine Datei `gradiverf.m` und verwenden Sie

```
function X = gradiverf(fhandle, x0, epsilon, s, alpha, beta)
```

als erste Zeile. Dabei bezeichnen `x0` den Startpunkt, `epsilon` den Parameter für das Abbruchkriterium $\|\nabla f(x_k)\| < \varepsilon$ und `s, alpha` und `beta` die Parameter für die ARMIJO-Regel. Zurückgegeben werden soll eine Matrix `X = [x0; x1; x2;...]` , welche den gesamten Iterationsverlauf enthält.

Testen Sie ihr Programm an folgenden Funktionen:

a) an der Funktion $\cos(x)$ mit `x0=1.1656, epsilon=`10^{-3}`, t0=1, alpha=`10^{-2} und `beta=0.5`.

b) an der ROSENBROCK-Funktion mit `x0=[1;-0.5], epsilon=`10^{-2}`, t0=1, alpha=`10^{-2} und `beta=0.5`.

HINWEISE

- Der Programmtext ist angemessen zu kommentieren. (Was beschreibt die Variable? Worüber läuft die Schleife? Was wird an dieser Stelle berechnet?)
- Bei Funktionen sollte am Anfang als Kommentar erklärt werden, was die Funktion berechnet, welche Größen zurückgegeben werden und was die Eingabeparameter sind.
- Fragen zur Aufgabe können per Email an Tim.Seger@uni-konstanz.de gestellt werden.
- Abgabe der Programmieraufgabe bis Freitag, 22. Mai 2009, 10 Uhr an Tim.Seger@uni-konstanz.de.