Technische Universität Chemnitz
Prof. Dr. C. Helmberg

Chemnitz, 25. April 2003

Optimierung 1
Übung 2

1. Sei $\{f_i(x) = a_i^T x + b_i : i \in J\}$ eine beliebige Familie linearer Funktionen mit $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$ für $i \in J$. Zeige: $f(x) := \sup_{i \in J} f_i(x)$ ist konvex auf $\mathbb{R}^n$.

2. Im inexakten line search Problem (approximiere $\min_{\alpha \geq 0} f(x+\alpha p)$, p Abstiegsrichtung) werden meist die Wolfe Bedingungen ($0 < c_1 < c_2 < 1$)

$$\begin{array}{rcll} f(x+\alpha p) & \leq & f(x) + c_1 \alpha \nabla f(x)^T p & \text{(Armijo Bedingung)} \\ \nabla f(x+\alpha p)^T p & \geq & c_2 \nabla f(x)^T p & \text{(Krümmungs-Bedingung)} \end{array}$$

als Garant für hinreichenden Abstieg verwendet. Ermittle für $f : \mathbb{R}^2 \to \mathbb{R}$, $f(x) = 10\cos(3\|x\|^2)e^{-\|x\|^2} + \frac{1}{2}x^T Ax + b^T x$ mit
$A = \begin{pmatrix} 21 & 8 \\ 8 & 9 \end{pmatrix}$, $b = \begin{pmatrix} 4 \\ -8 \end{pmatrix}$ in $\bar{x} = \begin{pmatrix} 0.8 \\ -0.9 \end{pmatrix}$
eine Abstiegsrichtung und bestimme graphisch in einem Matlab-Plot die zugehörigen Intervalle der Schrittlängen, die die Wolfe Bedingungen für $c_1 = 0.1, c_2 = 0.2$ erfüllen. Wie könnte man die Intervalle numerisch bestimmen?

3. Zeige: Ist $f : \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar, $p \in \mathbb{R}^n$ eine Abstiegsrichtung von f in x und ist f auf dem Halbstrahl $\{x + \alpha p : \alpha > 0\}$ nach unten beschränkt, dann gibt es für $0 < c_1 < c_2 < 1$ Schrittlängenintervalle, auf denen die Wolfe Bedingungen erfüllt sind.

4. Eine Funktion $f : S \to \mathbb{R}$ mit $S \subset \mathbb{R}^n$ konvex heißt *streng quasikonvex* auf S, wenn für alle $x, y \in S$ mit $x \neq y$

$$f(\alpha x + (1-\alpha)y) < \max\{f(x), f(y)\}, \quad \forall \alpha \in (0,1).$$

Zeige: Ist $f : \mathbb{R} \to \mathbb{R}$ stetig und streng quasiconvex auf $[a,b]$ dann hat $\min_{a \leq x \leq b} f(x)$ eine eindeutige Optimallösung x^* und für $\lambda, \mu \in \mathbb{R}$ mit $a \leq \lambda < \mu \leq b$ gilt:

   (i) Aus $f(\lambda) \leq f(\mu)$ folgt $x^* \in [a, \mu]$,
   (ii) aus $f(\lambda) \geq f(\mu)$ folgt $x^* \in [\lambda, b]$.

5. Die Goldene-Schnitt Suche zur Minimierung einer streng quasikonvexen Funktion $f : \mathbb{R} \to \mathbb{R}$ auf $[a_1, b_1]$ vergleicht in Iteration k die Funktionswerte in

$$\begin{array}{ll} \lambda_k = a_k + (1-t)(b_k - a_k), & k = 1, 2, \ldots \\ \mu_k = a_k + t(b_k - a_k), & k = 1, 2, \ldots \end{array}$$

wobei $t = \frac{-1+\sqrt{5}}{2} \approx 0.618$, und setzt a_{k+1} und b_{k+1} wie in 4 (i) und (ii).

   a) Zeige: In jeder Iteration gilt
      (i) $b_k - \lambda_k = \mu_k - a_k$,
      (ii) entweder ist $\lambda_{k+1} = \mu_k$ oder $\mu_{k+1} = \lambda_k$.
   b) Erstelle einen Algorithmus zur Suche mit dem Goldenen Schnitt.

c) Wende das Verfahren auf $\min -e^{-(x-2)^2/10}+1$ mit $a_1=-1$, $b_1=4$ an.

6. Folgendes Beispiel findet sich in George B. Dantzig[1], Linear Programming and Extensions, Princeton University Press, 1998 (Neuauflage des Klassikers von 1963), Seite 117:
Formulate as a linear programming problem: Suppose six foods listed below have calories, amounts of protein, calcium, vitamin A, and costs per pound purchased as shown. In what amounts should these foods be purchased in order to meet exactly the daily equivalent per person shown in the last column at minimum cost? How is the model modified if the daily requirements may be exceeded; if the requirements except for calories may be exceeded?

	Contents and Costs Per Pound Purchased						Daily	
	Bread	Meat	Potatoes	Cabbage	Milk	Gelatin	Requirements	
Calories	1254	1457	318	46	309	1725	3000	
Protein	39	73	8	4	16	43	70	(grams)
Calcium	418	41	42	141	536	–	800	(mg.)
Vitamin A	–	–	70	860	720	–	500	(I.U.)
Cost	\$ 0.30	\$ 1.00	\$ 0.05	\$ 0.08	\$ 0.23	\$ 0.48	Minimum	

Erstelle ein allgemeines Diätmodell mit AMPL für den exakten Fall und löse es für diese Daten.

AMPL-Syntax:
Mehrdimensionale Parameterfelder:
`param` coeffname { set1 `,` set2 `,` ...}`;`
Referenzieren der Parameter:
coeffname `[`i1`,`i2`,`...`]`;`
Nebenbedingungsfamilien:
`subject to` ineqname {i1 `in` set1`,` i2 `in` set2`,` ...}`:` ...`;`
Wertzuweisung an mehrdimensionale Parameter:
2-dimensional:
`param` coeffname`:` set2_1 set2_2 ... set2_n `:=` set1_1 coef11 coeff12 ... coeff1n set1_2 coeff21 ...
2D transponiert:
`param` coeffname`(tr)` `:` set1_1 set1_2 ... set1_n `:=` set2_1 coef11 coeff21 ...
Allgemein gibt es viele Varianten; eine für dünn besetzte Felder ist:
`param` coeffname `default` defaultwert `:`
set1_1 set2_1 ... setk_1 coeff11...1
set1_1 set2_1 ... setk_2 coeff11...2
set1_2 set2_1 ... setk_2 coeff21...2
...

[1] Erfinder der Simplex Methode.