Prof. Dr. Karsten Urban
M.Sc. Florian Kunstmann
Institut für Numerische Mathematik
Universität Ulm

Übungsblatt 10

Besprechung 13.01.2016.

Aufgabe 28 (Primales und Duales Problem) (2 Punkte)

Gegeben sei das *primale lineare Problem* der Form:

$$\begin{cases} \langle c, x\rangle \to \max \\ \text{Nebenbedingung: } Ax = b, \quad x \geq 0. \end{cases} \tag{1}$$

Dann lautet das *duale Problem* zu (1):

$$\begin{cases} \langle b, y\rangle \to \min \\ \text{Nebenbedingung: } A^T y \geq c. \end{cases} \tag{2}$$

Zeigen Sie, dass das duale Problem zu (2) wieder (1) ist.

Aufgabe 29 (Logarithmische Barrieren) (4 Punkte)

Zeigen Sie, dass die *logarithmische Barriere* Φ_μ strikt konvex auf $\mathcal{F}_P^0$ bzw. $\tilde{\Phi}_\mu$ strikt konkav auf $\mathcal{F}_D^0$ ist.

Aufgabe 30 (Strikt zulässige Punkte) (4+1 Punke)

Gegeben seien stetige Funktionen $h_i : \mathbb{R}^n \to \mathbb{R},\ i = 1, \dots, l$ und definiere $h := (h_1, \dots, h_l)^T$. Man betrachte nun die Menge $\mathcal{F} := \{x \in \mathbb{R}^n \ :\ h(x) \leq 0\}$, sowie das *strikte Innere*

$$\mathcal{F}^0 := \{x \in \mathbb{R}^n \ :\ h(x) < 0\}$$

und den *offenen Kern* $\mathring{\mathcal{F}} := \mathcal{F} \setminus \partial\mathcal{F}$ von $\mathcal{F}$. Zeigen Sie:

(i) Sind die Funktionen h_i konvex und ist $\mathcal{F}^0 \neq \emptyset$, so ist $\mathcal{F}^0 = \mathring{\mathcal{F}}$.

(ii) Finden Sie stetige Funktionen h_i derart, dass $\mathcal{F} \neq \emptyset$, aber $\mathcal{F}^0 \neq \mathring{\mathcal{F}}$.

Aufgabe 31 (SQP-Verfahren) (5 Punkte)

Gegeben sei folgendes Optimierungsproblem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0 \\ & g(x) \leq 0 \end{aligned}$$

mit einer beliebigen Funkion $f : \mathbb{R}^n \mapsto \mathbb{R}$, sowie m Ungleichungsbeschränkungen, d.h. $g : \mathbb{R}^n \mapsto \mathbb{R}^m$ und p Gleichungsbeschränkungen, also $h : \mathbb{R}^n \mapsto \mathbb{R}^p$.

a) Implementieren Sie in `Matlab` ein (lokales) SQP-Verfahren mit $H_k := \nabla^2 f(x_k)$ **ohne** Schrittweitensteuerung und **ohne** Nutzung einer Penalty-Funktion zur Lösung obiger Programme mit folgendem Funktionsaufruf

```
[x,lambda,mu,fx,nit] = sqp(f,gradf,hessf,g,jacobig,h,jacobh,x,lambda,mu)
```

mit den Parametern

- `f` - die Funktion $f : \mathbb{R}^n \mapsto R$ als *function handle*
- `gradf` - Der Gradient von f als *function handle*
- `hessf` - die Hessematrix von f als *function handle*
- `g` - die Funktion $g : \mathbb{R}^n \mapsto R^m$ als *function handle*
- `jacobig` - die Jacobi Matrix von g als *function handle*
- `h` - die Funktion $h : \mathbb{R}^n \mapsto R^p$ als *function handle*
- `jacobih` - die Jacobi Matrix von h als *function handle*
- `x,lambda,mu` - die Startwerte $(x_0, \lambda_0, \mu_0) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^m$

Zur Lösung des Quadratischen Programms können Sie Ihre Implementierung der Active-Set-Methode oder die Funktion `quadprog` verwenden. Prüfen Sie für den Abbruch der Funktion die KKT-Bedingungen (z.B. wie in Aufgabe 21).

b) Schreiben Sie ein Skript `test_sqp.m`, dass das Optimierungsproblem mittels `sqp`-Funktion löst und ver-

$$\begin{array}{rrcl} \min\limits_{x \in \mathbb{R}^2} & f(x) & := & (x_1 - \sqrt{2})^2 + (x_2 + \sqrt{2})^2 + 5 \\ \text{s.t.} & x_1^2 + x_2^2 & \geq & 1 \\ & \frac{1}{2}x_1^2 + x_2^2 & \leq & 9 \end{array}$$

wenden Sie den folgenden Startwert $x_0 = (0.1, 2.5)^T$ und $\mu = (1, 1)^T$

c) Diskutieren Sie ihre numerischen Ergebnisse und ermitteln Sie, an welchen Stellen bei allgemeinen Optimierungsaufgaben Probleme au

Hinweis: Die Idee im SQP-Verfahren ist, das Inkrement d als Lösung einer lokalen quadratischen Approximation zu berechnen:

$$\begin{array}{rl} \min\limits_{d} & \frac{1}{2} d^T f''(x) d + \nabla f(x)^T d \\ \text{unter} & g(x) + \nabla g(x)^T d \leq 0 \\ & h(x) + \nabla h(x)^T d = 0 \end{array}$$

Aufgabe 32 (Innere-Punkte-Verfahren) (5 Punkte)

Wir betrachten Algorithmus 4.5.1 (Innere-Punkte-Verfahren) im Skript. Da die Matrix DF_0 viele Null-Einträge besitzt, überlegen wir uns zunächst, dass das Gleichungssystem in Schritt 3.) effizient gelöst werden kann, indem man folgende Herangehensweise betrachtet:
Sei $r := XSe - \sigma\tau e$, $D^2 := S^{-1}X$ und betrachte das Gleichungssystem (Schritt 3.) im Algorithmus 4.5.1):

$$DF_0 \cdot \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \cdot \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ XSe - \sigma\tau e \end{pmatrix}, \tag{3}$$

Auflösen der ersten Zeile nach Δs, die dritte Zeile nach Δx und Einsetzen in die zweite Zeile ergibt somit:

$$\begin{aligned} \Delta y &= -\left(AD^2A^T\right)^{-1} AS^{-1}r = -\left(AD^2A^T\right)^{-1} A(Xe - \sigma\tau S^{-1}e), \\ \Delta s &= -A^T \Delta y, \\ \Delta x &= S^{-1}r - S^{-1}X\Delta s = (Xe - \sigma\tau S^{-1}e) - S^{-1}X\Delta s. \end{aligned}$$

(i) Zeigen Sie, dass das so berechnete Tripel $(\Delta x, \Delta y, \Delta s)^T$ eine Lösung von (3) ist.

(ii) Wieviele Gleichungssysteme gibt es hier nun zu lösen?

(iii) Wir bemerken weiter, dass die Matrix AD^2A^T symmetrisch und im Falle $x, s > 0$ positiv definit ist, da die Matrix A vollen Rang besitzt. Das bedeutet, dass wir hier eine Cholesky-Zerlegung verwenden können. Welche Dimension besitzt AD^2A^T?

(iv) Laden Sie das Material von der Homepage.

(v) Ergänzen Sie die Funktion `[x y s iter] = InteriorPointMethod(c,A,b,x,y,s,TOL)` derart, dass ein Innere-Punkte-Verfahren entsteht.

(vi) Testen Sie Ihre Funktion mit Hilfe der Datei `test.m`. Was tut hierbei die Matlab-Funktion `linprog(c,-eye(4),zeros(4,1),A,b);`?

Aufgabe 33 (Kurz-Schritt-Algorithmus) (4 Punkte)

Eine weitere Möglichkeit, das Innere-Punkte-Verfahren zu implementieren, ist der sogenannte Kurz-Schritt-Algorithmus, welcher wie folgt lautet: Sei $\alpha := 1$ und $\sigma := 1 - \frac{2}{5\sqrt{n}}$. Betrachte dann den so entstehenden Algorithmus:
Wir definieren zunächst $\mathcal{N}_2(\beta) := \left\{(x,y,s) \in \mathcal{F}^0 \;:\; \|XSe - \frac{1}{n}\langle x,y\rangle\| \leq \beta\frac{1}{n}\langle x,y\rangle\right\}$. Sei $\left(x^{(0)}, y^{(0)}, s^{(0)}\right) \in \mathcal{N}_2(\frac{1}{2})$, $\varepsilon > 0$ und $k = 0$ gegeben. Definiere $\tau_0 := \frac{1}{n}\langle x,s\rangle = \frac{1}{n}x^Ts$.

WHILE $\tau > \varepsilon$

1. Löse (3) und erhalte $(\Delta x, \Delta y, \Delta s)^T$
2. Setze $\left(x^{(k+1)}, y^{(k+1)}, s^{(k+1)}\right)^T = \left(x^{(k)}, y^{(k)}, s^{(k)}\right)^T - (\Delta x, \Delta y, \Delta s)^T$
3. $\tau_{k+1} = \sigma \cdot \tau_k$
4. $k = k + 1$

END

(i) Implementieren Sie den Kurz-Schritt-Algorithmus. Ihre Funktion sollte wie folgt aufgerufen werden: `[x y s iter] = InteriorPointMethod_ShortStep(c,A,b,x,y,s,TOL)`.

(ii) Vergleichen Sie die beiden Verfahren mit Hilfe von `test.m`.

(iii) Worin liegen die wesentlichen Unterschiede der beiden Verfahren? Diskutieren Sie auch die Bedingungen an die Startwerte.