Einführung in die Numerische Mathematik

Sommersemester 2015
Prof. Dr. Jochen Garcke
Patrick Diehl

universitätbonn

Übungsblatt 7. Abgabe am **02.06.2015 vor der Vorlesung**.

Aufgabe 1. Wenden Sie den Algorithmus 7 zur Bestimmung aktiven Indizes des folgenden Optimierungsproblem an

$$\min \left\{ f(x) = \frac{1}{2}(x_1^2 + x_2^2) + 2x_1 + x_2 \;\middle|\; \begin{array}{l} g_1(x) = -x_1 - x_2 \\ g_2(x) = x_2 - 2 \\ g_3(x) = x_1 + x_2 - 5 \\ g_4(x) = -x_1 + x_2 - 2 \\ g_5(x) = x_1 - 5 \end{array} \right.$$

und geben Sie in einer Tabelle x^k, $\tilde{A}(x^k)$, s^{k+1}, $\lambda_1, \ldots, \lambda_6$ an. Verwenden Sie als Startwert $(5,0)^T$ und $\lambda = (0,0,0,0,0,0)^T$.

(6 Punkte)

Aufgabe 2. Zeigen Sie, dass wenn bei einem quadratischen Minimierungsproblem im k-tem Iterationsschritt von Algorithmus 7 bei der Bestimmung der aktiven Indizes die Vektoren a_i mit $i \in \tilde{A}(x^k) \cup \{m+1, \ldots, m+p\}$ linear unabhängig sind, dann sind auch die Vektoren a_i mit $i \in \tilde{A}(x^{k+1}) \cup \{m+1, \ldots, m+p\}$ linear unabhängig.

(6 Punkte)

Aufgabe 3. Sei $f \in C^2(\mathbb{R}^d, \mathbb{R}), h \in C^2(\mathbb{R}^n, \mathbb{R})$ und $(x^{\otimes}, \mu^{\otimes}) \in \mathbb{R}^d \times \mathbb{R}^p$ ein KKT-Punkt des zugehörigen Optimierungsproblems.

Zeigen Sie, dass wenn die Gradienten $\nabla h_1(x), \ldots, \nabla h_p(x)$ linear unabhängig sind und $y^T \nabla_x^2 L(x,y) y > 0$ für alle $y \in \mathbb{R}^d \setminus \{0\}$ mit $\nabla H(x)^T y = 0$ gilt, die Matrix

$$M(x^{\otimes}, \mu^{\otimes}) := \begin{pmatrix} \nabla_x^2 L(x^{\otimes}, \mu^{\otimes}) & \nabla H(x^{\otimes}) \\ \nabla H(x^{\otimes})^T & 0 \end{pmatrix}$$

invertierbar ist.

(6 Punkte)

Programmieraufgabe 1. (Auswahl der aktiven Indizes)

Implementieren Sie den Algorithmus 7 zur Auswahl der aktiven Indizes in python. Wenden Sie den Algorithmus auf das Problem aus Aufgabe 1 an und geben Sie die Werte für x^k, $\tilde{A}(x^k)$, s^{k+1}, $\lambda_1, \ldots, \lambda_6$ aus. Für die Verifikation ihrer Implementierung können Sie die Ausgabe mit der von Hand berechneten Tabelle aus Aufgabe 1 vergleichen.
Hinweise:

- Verwenden Sie die Datenstruktur *set*[1]
- Schauen Sie sich die API des Pythonpakets *numpy.linalg* an. Dort finden Sie viele Funktionen[2], die Sie im Algorithmus benötigen.

[1] `https://docs.python.org/2/library/sets.html`
[2] `http://docs.scipy.org/doc/numpy/reference/routines.linalg.html`

(12 Punkte)

Abgabe am 08.06.2015 oder 09.06.2015 im CIP-Pool. Weitere Hinweise finden Sie auf der Webseite.

Programmieraufgabe 2. (SQP-Verfahren)

Implementieren Sie in python das SQP-Verfahren und wenden Sie es auf folgendes Optimierungsproblem an

$$\min \left\{ f(x) = 3(x_1 - 2)^2 + 2(x_2 - 3)^2 \,\middle|\, \begin{array}{l} g_1(x) = x_1^2 - x_2 \\ g_2(x) = x_1^2 + x_2^2 - 1 \end{array} \right.$$

mit den Startwert $x^0 = (\frac{1}{2}, 1)^T$ und $\lambda^0 = (0, 0)^T$. Verwenden Sie zur Wahl der aktiven Indizes den Algorithmus aus der ersten Programmieraufgabe.

Hinweis:
Für die Suche eines zulässigen Wertes für das QP-Teilproblem können Sie den simplex Algorithmus verwenden. Dieser ist in *scipy*[3] 1.15.0 verfügbar. Falls Sie eine ältere Version von scipy verwenden, dann können Sie den Algorithmus auch auf der Webseite runterladen.

(6 Punkte)

Abgabe am 08.06.2015 oder 09.06.2015 im CIP-Pool. Weitere Hinweise finden Sie auf der Webseite.

[3] `http://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.optimize.linprog.html`