Universität Hamburg
Fachbereich Mathematik
Dr. Matthias Voigt, Dr. Heiko Kröner

Sommersemester 2019

Optimierung – 2. Übungsblatt.

Bitte bearbeiten Sie die Übungsaufgaben in festen Zweiergruppen.
Abgabetermin: *Donnerstag, 25.04.2019 vor der Übung. Bitte senden Sie die Lösung der Programmieraufgabe bis zum 24.04.2019, 18:00 Uhr an* `heiko.kroener@uni-hamburg.de`*. Alle zur Ausführung nötigen Dateien sollten in einem Archiv mit dem Namen* `opt_blattnr_aufgabennr_name1_name2.zip` *enthalten sein.*

Aufgabe 1 (4 Punkte): Es seien eine Menge $M \subseteq \mathbb{R}^n$ und eine Funktion $f : M \to Y$ mit $Y \subseteq \mathbb{R}$ gegeben. Weiter sei $\psi : Y \to \mathbb{R}$ eine streng monoton wachsende Funktion. Zeigen Sie, daß dann

$$\min_{x \in M} \psi(f(x)) = \psi\Big(\min_{x \in M} f(x)\Big)$$

gilt (falls das Minimum auf einer der beiden Seiten existiert) und daß die lokalen und globalen Minimalpunkte von f und $\psi \circ f$ übereinstimmen.

Aufgabe 2 (8 Punkte): Bestimmen Sie für die Funktionen

a) $f(x_1, x_2) = x_1^3 - x_1 x_2 + x_2^2 - 2x_1 + 3x_2 - 4$,

b) $g(x_1, x_2) = \big(1.5 - x_1\big(1 - x_2\big)\big)^2 + \big(2.25 - x_1\big(1 - x_2^2\big)\big)^2 + \big(2.625 - x_1\big(1 - x_2^3\big)\big)^2$

alle lokalen Minimalpunkte. Berechnen Sie zunächst alle stationären Punkte von f and überprüfen Sie dann anhand weiterer Kriterien, in welchen stationären Punkten lokale Minima oder strikte lokale Minima vorliegen.

Aufgabe 3 (8 Punkte): Implementieren Sie das Verfahren von Melder und Mead. Schreiben Sie dazu eine `MATLAB`-Funktion mit dem Interface

```
[ fmin, xmin, fiters, xiters ] = neldermead( fun, x0, alpha, beta, gamma, tol )
```

und folgenden Variablenbezeichnungen:

- `fmin`, `xmin`: der minimale Funktionswert und der Minimierer;
- `fiters`, `xiters`: Arrays, die die Funktionswerte an den Iterierten und die Iterierten selbst speichern;
- `fun`: ein Function Handle zur Auswertung der Zielfunktion – d. h., wenn `f = evalfun( x )` eine weitere `MATLAB`-Funktion ist, die zu einem gegebenen `x` den zugehörigen Funktionswert `f` ausgibt, so muß `@evalfun` für `fun` eingesetzt werden;
- `x0`: Startwert der Iteration;
- `alpha`, `beta`, `gamma`: die Kontraktions-, Expansions- und Reflektionskonstanten;
- `tol`: eine Abbruchtoleranz – verwenden Sie hierzu das Kriterium von Nelder und Mead, d. h., das Verfahren bricht ab, wenn

$$\left(\frac{1}{n+1}\sum_{i=0}^{n}\big(f\big(x^{(k,i)}\big) - \bar{f}_k\big)^2\right)^{1/2} < \texttt{tol} \quad \text{mit} \quad \bar{f}_k = \frac{1}{n+1}\sum_{j=0}^{n} f\big(x^{(k,j)}\big),$$

wobei $x^{(k,j)}$ die j-te Simplexecke im k-ten Iterationsschritt bezeichnet.

Testen Sie Ihre Implementation anhand der Rosenbrock-Funktion von Übungsblatt 1 (Hinweis: Das Minimum wird bei $x_* = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$ und mit $f(x_*) = 0$ angenommen.). Verwenden Sie die Parameter

`x0 = [ -1.9; 2 ]`, `alpha = 0.5`, `beta = 2.5`, `gamma = 1` und `tol = 1e-4`.

Veranschaulichen Sie das Konvergenzverhalten der Iterierten und der Funktionswerte in je einer Graphik. Nutzen Sie hierzu die Informationen, die in `fiters` und `xiters` gespeichert sind. Zur Veranschaulichung der Niveaulinien der Zielfunktion bietet sich der Befehl `contour` an. Exportieren Sie die von Ihnen erzeugten Graphiken (mit aussagekräftigen Legenden, Achsenbeschriftungen und Bildunterschriften!) in je ein PDF-Dokument und fügen Sie diese Ihrem Code-Archiv hinzu. Vergleichen Sie Ihre Ergebnisse mit denen der `MATLAB`-Funktion `fminsearch`. Bitte versehen Sie Ihren Code auch mit aussagekräftigen Kommentaren.