

Grundlagen der Optimierung Übung 4

1. Bestimme Gradienten und Hessematrix folgender Funktionen. Zeichne die Funktionen auf den angegebenen Gebieten mittels der Matlab-Funktionen `mesh` und `contour`. (Es ist günstig, geeignete Niveaus vorzugeben.)

(a) $f(x) = \frac{1}{2}x^T Qx + b^T x + d$ für $b = \begin{pmatrix} 5 \\ 1 \end{pmatrix}$, $d = 3$ und $Q =$

(i) $\begin{pmatrix} 4 & -2 \\ -2 & 2 \end{pmatrix}$, (ii) $\begin{pmatrix} 4 & -2 \\ -2 & 1 \end{pmatrix}$, (iii) $\begin{pmatrix} 4 & -2 \\ -2 & -2 \end{pmatrix}$, (iv) $\begin{pmatrix} -4 & -2 \\ -2 & -2 \end{pmatrix}$

auf $[-10, 10] \times [-10, 10]$.

(2 Punkte)

(b) $g(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ (Rosenbrockfunktion) auf $[-2, 2] \times [-1, 3]$. (2 Punkte)

2. (a) Bestimme die Extrema der Funktion $f(x) = x_1^3 + x_2^3 - 3x_1x_2$ und deren Art. (2 Punkte)

(b) Besitzt die Funktion $f(x) = (x_1 - x_2^2)(2x_1 - x_2^2)$ im Punkt $x = (0, 0)$ ein lokales Minimum (Begründung)? (2 Punkte)

3. Für die Matrix $A \in \mathbb{R}^{m \times n}$ mit $m > n$ und $\text{rang} A = n$ sei eine Zerlegung $A = QR$ bekannt, wobei $Q \in \mathbb{R}^{m \times n}$ mit $Q^T Q = I_n$, und $R \in \mathbb{R}^{n \times n}$ eine obere Dreiecksmatrix ist. Außerdem sei ein Vektor $b \in \mathbb{R}^m$ gegeben. Bestimme eine bestmögliche Lösung des überbestimmten Gleichungssystems $Ax = b$, d.h. eine optimale Lösung der Aufgabe $\min_x \|Ax - b\|$ (Nachweis der Optimalität!).

(4 Punkte)

4. Entwickle und implementiere in Matlab eine line-search-Funktion der Form

```
[npoint, nval, ngrad]=linesearch(@fun, cpoint, dir, maxa, cval, cgrad, c1, c2).
```

Für eine stetig differenzierbare Funktion $f: \mathbb{R}^n \rightarrow \mathbb{R}$ soll sie für $f(x_c + \alpha p)$, $\alpha \in [0, \bar{\alpha}]$ einen Punkt zurückliefern, der die Wolfe Bedingungen mit den Parametern c_1 und c_2 erfüllt. Die Übergabeparameter sind

<code>fun</code>	für f ,	<code>cpoint</code>	für x_c ,
<code>dir</code>	für die Abstiegsrichtung p ,	<code>maxa</code>	für $\bar{\alpha}$,
<code>cval</code>	für $f(x_c)$,	<code>cgrad</code>	für $\nabla f(x_c)$.

Rückgabewerte sind der neue (Wolfe-) Punkt, der neue Funktionswert und der neue Gradient. Alle Vektoren/Punkte sind Spaltenvektoren. `fun` soll ein Handle auf eine beliebige Funktion der Form

```
[val, grad]=function_name(point)
```

sein (help feval, doc feval, help function_handle, doc function_handle).

Schicke die erzeugten Dateien an `susanna.reiss@mathematik.tu-chemnitz.de`
(Betreff: HA-Optimierung)! (8 Punkte)

Programmiervorschlag:

Sei $x(\alpha) := x_c + \alpha p$. Starte mit $\alpha = \bar{\alpha}$. Erfüllt $x(\bar{\alpha})$ die Wolfe-Bedingungen, so fertig. Erfüllt $x(\bar{\alpha})$ Armijo, aber nicht die Curvature-Bedingung, so gib trotzdem $x(\bar{\alpha})$ zurück ($\bar{\alpha}$ könnte zu klein gewählt sein, und im gegebenen Intervall kein Wolfe-Punkt existieren). Erfüllt aber $x(\bar{\alpha})$ nicht die Armijo-Bedingung, so finde mittels Backtracking $\alpha \leftarrow \gamma \alpha$ mit z.B. $\gamma = 0.5$ einen Punkt $x(\alpha)$ der die Armijo-Bedingung erfüllt. Ist $x(\alpha)$ ein Wolfe-Punkt, so fertig. Sonst suche einen Wolfe-Punkt mit Intervallhalbierung in $[\alpha, \gamma^{-1}\alpha]$: Ist der Mittelpunkt ein Wolfe-Punkt, so fertig, erfüllt er nur Armijo, so unterteile rechten Teil, erfüllt er nicht Armijo, so unterteile linken Teil, usw.