Philipps-Universität Marburg **Sommersemester 2017**
Fachbereich Mathematik
Prof. Dr. T. Surowiec, A. Kröck, A. Sieber

Übungen zur Nichtlinearen Optimierung
7. Aufgabenblatt

Aufgabe 1 (Finite Differenzen) (10)

(i) Leiten Sie die Vorwärtsdifferenzen-Formel für die Hesse-Matrix

$$\left[\nabla^2 f(x)\right]_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \approx \frac{[\nabla f(x + he_j) - \nabla f(x)]_i}{h}$$

her. Leiten Sie außerdem Formeln für den besten Wert von h sowie den Wert des Fehlers her.

(ii) Schätzen Sie die Hesse-Matrix der Funktion $f(x_1, x_2) = \exp(10x_1 + 2x_2^2)$ mit Hilfe finiter Differenzen. Bilden Sie dazu zunächst Vorwärtsdifferenzen der Gradienten, und wiederholen Sie dann Ihr Vorgehen, indem Sie Differenzen der Funktionswerte bilden.

Aufgabe 2 (Automatische Differentiation) (10)

Betrachten Sie die Funktion

$$f(x_1, x_2) = x_1^2 + 3x_1x_2 + 7x_2^2.$$

(i) Erstellen Sie den Graph zur Auswertung der Funktion f.

(ii) Nutzen Sie den Graph aus Teil (i), um eine Technik zur Auswertung der Ableitung $\nabla f(x)$ in dem Punkt $x = (4, -5)^T$ herzuleiten. Verwenden Sie den Reverse Mode der automatischen Differentiation.

Aufgabe 3 (CG-Verfahren) (15)

(i) Wenden Sie das CG-Verfahren zur Lösung des Gleichungssystems $Ax = b$ mit

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}, \text{ und } b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

an. Weisen Sie nach, dass die Vektoren $\{p_i\}$ konjugiert und die Vektoren $\{r_i\}$ orthogonal sind.

(ii) Nehmen Sie an, das CG-Verfahren wird zur Minimierung einer Funktion in drei Variablen angewandt. Initialisieren sie $r_0 = (1, -1, 2)^T$. Nehmen Sie weiter an, dass in der ersten Iteration die ersten beiden Einträge des Vektors r_1 jeweils gleich 2 sind. Bestimmen Sie die Suchrichtung des CG-Verfahrens in der zweiten Iteration.

(iii) Nehmen Sie an, dass das CG-Verfahren zur Minimierung einer Funktion eingesetzt wird. Im Verlauf der Iteration erhalten Sie die Daten $r_i = (5, 3, -1)^T$ und $p_i = (4, -2, 1)^T$. Wieso können diese Daten nicht korrekt sein?

Aufgabe 4 (Bonusaufgabe: Programmieraufgabe BFGS-Quasi-Newton-Verfahren) (15*)

Schreiben Sie ein Programm zur Minimierung einer multivariaten Funktion mit Hilfe des BFGS-Quasi-Newton-Verfahrens.
Verwenden Sie $B_0 = I$ als erste Approximation der Hesse-Matrix.

(i) Verwenden Sie zur Liniensuche die Backtracking-Strategie. Testen Sie vor dem Update der Matrix B_k, ob $y_k^T s_k > 0$ gilt. Führen Sie in dieser Iteration kein Update der Matrix B_k durch, falls diese Bedingung nicht erfüllt ist.

(ii) Akzeptieren Sie x als Lösung, falls $(1 + |f(x)|)^{-1} \|\nabla f(x)\| \leq \varepsilon = 10^{-8}$ gilt, oder die Anzahl der Iteration ITMAX = 1000 überschreitet.

(iii) Geben Sie den Startpunkt x_0 aus. Geben Sie außerdem in jeder Iteration die Suchrichtung p, die Schrittweite α und die neue Näherung der Lösung x_{k+1} aus. Geben Sie einen Hinweis aus, sofern nach ITMAX Iterationen keine Lösung gefunden wurde.

(iv) Testen Sie ihr Programm anhand der nachstehenden Funktionen;

$$
\begin{aligned}
f_a(x) &= x_1^2 + x_2^2 + x_3^2, & x_0 &= (1, 1, 1)^T \\
f_b(x) &= x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2, & x_0 &= (0, 0)^T \\
f_c(x) &= 100(x_2 - x_1^2)^2 + (1 - x_1)^2, & x_0 &= (-1.2, 1)^T \\
f_d(x) &= (x_1 + x_2)^4 + x_2^2, & x_0 &= (2, -2)^T \\
f_e(x) &= (x_1 - 1)^2 + (x_2 - 1)^2 + \eta(x_1^2 + x_2^2 - 0.25)^2, & x_0 &= (1, -1)^T.
\end{aligned}
$$

Verwenden Sie für die Funktion f_e die Parameter $\eta \in \{1, 10, 100\}$. Für diese Funktion wird die Konditionszahl der Hesse-Matrix in der Lösung mit wachsendem Parameter η größer. Beschreiben Sie, wie dies die Performance des Algorithmus beeinflusst.

(v) Sind Ihre Ergebnisse konsistent mit der Theorie der Quasi-Newton-Verfahren?

Hinweise: Versehen Sie Ihr Programm mit sinnvollen Kommentaren. Programme, die nicht kompilieren, erhalten keine Punkte.

Aufgabe 5 (Bonusaufgabe: Programmieraufgabe CG-Verfahren) (15*)

Ein wesentliche Vorteil des CG-Verfahrens zur Lösung von Gleichungssystemen $Ax = b$ ist, dass die Matrix A nur in Form von Matrix-Vektor-Produkten vorkommt. Daher muss nur die Abbildung $v \mapsto Av$ implementiert werden, nicht aber die Matrix A selbst.

Implementieren Sie das CG-Verfahren der Vorlesung, wobei die Matrix-Vektor-Produkte in einer Untermethode berechnet werden. Schreiben Sie eine solche Untermethode für die Block-Tridiagonalmatrix

$$B = \frac{1}{h^2} \begin{pmatrix} T & -I_n & & & \\ -I_n & T & -I_n & & \\ & \ddots & \ddots & \ddots & \\ & & -I_n & T & -I_n \\ & & & -I_n & T \end{pmatrix} \in \mathbb{R}^{n^2 \times n^2},$$

bestehend aus jeweils n Blockmatrizen der Größe $n \times n$ pro Zeile und Spalte. Die Blockmatrizen T der Hauptdiagonale sind dabei von der Form

$$T = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

Dabei müssen die Vektoren nicht zwangsläufig in einem linearen Feld (Vektor) gespeichert werden, sondern können auch direkt in einem rechteckigen Feld (vollbesetzte Matrix) gespeichert werden.

Testen Sie das Verfahren zur Lösung des Gleichungssystems $Ax = b$ mit $b = (1, \ldots, 1)^T \in \mathbb{R}^{n^2}$ für verschiedene Werte $h \in \{0.1, 0.05, 0.02\}$. Dabei gilt für die Schrittweite $h = \frac{1}{n+1}$. Verwenden Sie die Abbruchbedingung $\|r_i\|_2 \leq 10^{-6}$. Plotten Sie Ihre Näherungslösungen. Matlab-Nutzer sollten dazu die Methode *surf* verwenden.

Hinweise: Versehen Sie Ihr Programm mit sinnvollen Kommentaren. Programme, die nicht kompilieren, erhalten keine Punkte.

Abgabe der schriftlichen Übungen: Dienstag, 20. Juni 2017, vor der Vorlesung.

Abgabe der Programmieraufgaben: Dienstag, 27. Juni 2017, vor der Vorlesung, per Email an kroecka@mathematik.uni-marburg.de