Nichtlineare Optimierung
3. Rechnerübungsblatt

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik
Prof. Dr. Stefan Ulbrich
Hannes Meinlschmidt

WS 2011-2012
9.12.2011

Rechnerübung

Aufgabe R1 (Inexaktes CG-Newton-Verfahren)

(a) Implementieren Sie das allgemeine Abstiegsverfahren (Algorithmus 3) in der Version von Aufgabe G1, Übungsblatt 7 (Algorithmus 1 dort) in `matlab`, bestimmen Sie also die Suchrichtung mit dem inexakten Newton-CG-Verfahren und die Schrittweite nach der Armijo-Regel. Verwenden Sie als Konstanten

$$\alpha = 10^{-3} \quad \text{und} \quad \nu = 10^{-2}.$$

Nutzen Sie zur Bestimmung der Schrittweiten Ihre bereits programmierte Funktion `armijo` von der ersten Rechnerübung. Beachten Sie hierbei, dass die Schrittweiten-Bestimmung nach Armijo auch hier mit $\gamma \in (0, 0.5)$ statt $\gamma \in (0, 1)$ aufgerufen werden soll. Der Funktionskopf des Verfahrens sollte folgendermaßen aussehen:

```
function [xn] = cgnewt(x0,fgH,tol,maxit),
```

wobei wie immer `x0` der Startpunkt, `fgH` eine Funktion, die Funktionswert, Gradient und Hessematrix zurückliefert, `tol` eine Abbruchtoleranz und `maxit` die maximale Anzahl durchzuführender Iterationen sein soll.

In jeder Iteration des Abstiegsverfahrens sollte die aktuelle Iteration, der Funktionswert, die Norm des Gradienten und die Schrittweite ausgegeben werden. Beim CG-Verfahren zur Bestimmung der Suchrichtung soll die Anzahl der benötigten Iterationen und das Restresiduum der Newtongleichung ausgegeben werden.

(b) Testen Sie Ihr Programm an den bereits bekannten Funktionen

- $f_1(x_1, x_2) = x_1^2 + \alpha x_2^2$, mit verschiedenen $\alpha \geq 1$ (z.B. mit $\alpha = 10$ und Startwert $x_0 = (10, 20)$),
- der Rosenbrock-Funktion $f_2(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ und verschiedenen Startwerten $x_0 \neq (1, 1)$,

sowie der Funktion

- $f_5(x_1, x_2) = x_1^4 - 2x_2^2 + x_2^4$ mit Startwert $x_0 = (0.5, 0.5)$.

Als weiterer Test ist zudem ein etwas komplizierteres Problem verfügbar, das *Minimalflächenproblem*:

$$\min_z \int_\Omega \sqrt{1 + \|\nabla z(x)\|^2}\, dx, \quad z = z_b \text{ auf } \partial\Omega.$$

Hier wird eine Funktion $z : \Omega \to \mathbb{R}$ gesucht, die auf dem Rand $\partial\Omega$ eines gegebenen Gebietes Ω mit einer gegebenen Funktion z_b übereinstimmt.

Dieses Problem wird diskretisiert auf einem 100×100-Gitter berechnet. Um eine Lösung berechnen zu lassen, laden Sie die Datei `minsurf.zip` von der Veranstaltungshomepage herunter und kopieren Sie die Dateien `runminsurfcgnewt.m` und `Fminsurf.m` in das Verzeichnis, in der auch die Funktion `cgnewt` liegt. Rufen Sie dort dann `runminsurfcgnewt` in `matlab` auf.

Zum Vergleich des CG- mit dem üblichen Newton-Verfahren, wie in der zweiten Rechnerübung implementiert, passen Sie den Funktionskopf Ihres dort programmierten Newton-Verfahrens auf folgendes an:

```
function [xn] = newt(x,fgH,tol,maxit).
```

Entpacken Sie auch die Datei `runminsurfnewt.m` aus dem Archiv `minsurf.zip` ins selbe Verzeichnis wie Ihr Newton-Verfahren, und rufen Sie die Funktion in Matlab auf.

(c) Vergleichen Sie die Ergebnisse, die Anzahl der Iterationen und die Laufzeit des Newton-CG-Verfahrens mit dem globalisierten Newton-Verfahren vom zweiten Rechnerübungsblatt.

Hinweis: Verwenden Sie hier die `matlab`-Befehle `tic`,`toc` und `cputime`.

Algorithmus 1 : Globalisiertes BFGS-Verfahren

Wähle $\gamma \in (0, 0.5)$ und $\theta \in (\gamma, 1)$. Wähle einen Startpunkt $x_0 \in \mathbb{R}^n$ und eine symmetrische, positiv definite Matrix $B_0 = H_0^{-1} \in \mathbb{R}^{n \times n}$;
for $k = 0, 1, \ldots$: **do**
  if $\nabla f(x_k) = 0$ **then**
    STOP mit Ergebnis x_k;
  end
  Berechne $s_k = -B_k \nabla f(x_k)$;
  Bestimme eine Schrittweite $\sigma_k > 0$ nach der Powell-Wolfe-Regel;
  Setze $x_{k+1} = x_k + \sigma_k s_k$;
  Berechne $B_{k+1} = H_{k+1}^{-1}$ nach dem inversen BFGS-Update

$$B_{k+1} = B_k + \frac{(d_k - B_k y_k) d_k^T + d_k (d_k - B_k y_k)^T}{y_k^T d_k} - \frac{(d_k - B_k y_k)^T y_k}{(y_k^T d_k)^2} d_k d_k^T,$$

  mit $d_k = x_{k+1} - x_k$ und $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ (siehe auch Skript);
end

Aufgabe R2 (Globalisiertes BFGS-Verfahren)
Implementieren Sie das globalisierte BFGS-Verfahren (Algorithmus 1) in `matlab`.
Wählen Sie

$$\gamma = 0.001, \quad \theta = 0.9 \quad \text{und} \quad H_0^{-1} = I,$$

und verwenden Sie im Algorithmus, wie in den vorigen Rechnerübungen gelernt, eine relaxierte Abbruchbedingung an die Norm des Gradienten. Verwenden Sie die Funktion zur Bestimmung der Powell-Wolfe-Schrittweite vom zweiten Rechnerübungsblatt.
Testen Sie Ihr Verfahren an den Funktionen f_1 und f_2 aus der vorigen Aufgabe. Vergleichen Sie auch die Anzahl der Iterationen und die Laufzeit des BFGS-Algorithmus mit dem globalisierten- und dem CG-Newton-Verfahren für diese Funktionen.

Hausübung

Aufgabe H1 (Charakterisierung der Lösung des TR-Problems) (4 Punkte)
Sei $H \in \mathbb{R}^{n,n}$ symmetrisch und $c \in \mathbb{R}^n$. Für die quadratische Funktion $q : \mathbb{R}^n \to \mathbb{R}$,

$$q(s) = c^T s + \frac{1}{2} s^T H s,$$

betrachten wir für $\Delta > 0$ das Trust-Region-Problem

$$\min q(s) \quad \text{s.t. } \|s\|_2 \leq \Delta. \tag{TP}$$

Zeigen Sie:

(a) Das Trust-Region-Problem (TP) besitzt eine Lösung.

(b) Falls für ein $\bar{s} \in \mathbb{R}^n$ ein $\lambda \geq 0$ existiert, so dass die Bedingungen
  i. $(H + \lambda I)\bar{s} = -c$,
  ii. $(H + \lambda I)$ positiv semidefinit,
  iii. $(\|\bar{s}\|_2 - \Delta)\lambda = 0$, d.h. es gilt entweder $\|\bar{s}\|_2 = \Delta$ oder $\|\bar{s}\|_2 < \Delta$ und $\lambda = 0$,

  erfüllt sind, so ist $\bar{s}$ eine Lösung des Trust-Region-Problems (TP).
  Hinweis: Betrachten Sie als Hilfe die Funktion $\tilde{q}(s) := c^T s + \frac{1}{2} s^T (H + \lambda I) s = q(s) + \frac{\lambda}{2} \|s\|_2^2$.

Bemerkung: Wir werden noch sehen, dass die Bedingungen unter (b) tatsächlich auch notwendige Optimalitätsbedingungen sind.

Aufgabe H2 (Lösen des TR-Problems) (6 Punkte)
Wir betrachten nochmals das Problem TP aus Aufgabe H1, mit den gegebenen Werten $\Delta = \sqrt{2}$,

$$c = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad \text{und} \quad H = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

(a) Geben Sie die notwendigen und hinreichenden Optimalitätsbedingungen aus H1 konkret an.

(b) Berechnen Sie $s(\lambda) := -(H + \lambda I)^{-1} c$, wobei $\lambda > \max\{0, -\lambda_{\min}(H)\}$ und $p(\lambda) := \|s(\lambda)\|_2$.

(c) Bestimmen Sie die Optimallösung des Trust-Region-Problems. Zeichen Sie die Funktionen $p(\lambda) - \Delta$ und $\frac{1}{p(\lambda)} - \frac{1}{\Delta}$ als Funktionen in λ auf einem geeigneten Intervall. Was fällt Ihnen auf?

Aufgabe H3 (Beispiel zu Trust-Region-Modellen) (4 Punkte)

Sei $f : \mathbb{R}^2 \to \mathbb{R}$ gegeben durch einen kleinen Bruder der Rosenbrock-Funktion, nämlich

$$f(x_1, x_2) = 10(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Das quadratische Modell dieser Funktion ist gegeben durch

$$q(s) := f(x) + \nabla f(x)^T s + \frac{1}{2} s^T \nabla^2 f(x) s.$$

Zeichnen Sie jeweils die Höhenlinien des quadratischen Modells im Punkt $x_1 = (0, -1)^T$ und $x_2 = (0, 0.5)^T$ (z.B. mit `matlab`) und fügen Sie die Trust Regions, die durch $\Delta \in \{0.5, 1, 1.5, 2\}$ gegeben sind, ein. Geben Sie anhand der Bilder an, wo sich die Lösung des jeweiligen TR-Problems befindet. Vergleichen Sie auch die tatsächlichen Höhenlinien von f und die Trust Regions um x_2.