



3. Aufgabenblatt des Rechnerpraktikums zur „Nichtlinearen Optimierung“

Aufgabe P5 (BFGS-Verfahren)

Implementieren Sie das folgende globalisierte BFGS-Verfahren in `Matlab`:

Algorithmus GLOBALISIERTES BFGS-VERFAHREN

Initialisierung: Wähle $\gamma \in (0, 0.5)$ und $\theta \in (\gamma, 1)$. Wähle einen Startpunkt $x_0 \in \mathbf{R}^n$ und eine symmetrische, positiv definite Matrix $B_0 = H_0^{-1} \in \mathbf{R}^{(n,n)}$.

Für $k = 0, 1, \dots$:

1. Falls $\nabla f(x_k) = 0$: STOP mit Ergebnis x_k .
2. Berechne $s_k = -B_k \nabla f(x_k)$.
3. Bestimme eine Schrittweite $\sigma_k > 0$ nach der Powell-Wolfe-Regel.
4. Setze $x_{k+1} = x_k + \sigma_k s_k$.
5. 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 $d_k = x_{k+1} - x_k$, $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ (siehe auch Skript).

Wählen Sie

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

und verwenden Sie – wie bisher – eine relaxierte Abbruchbedingung an die Norm des Gradienten. Verwenden Sie Ihre erweiterte `armijo`-Funktion aus Aufgabe P4 zur Bestimmung der Powell-Wolfe-Schrittweiten.

Testen Sie ihr Verfahren an den Funktionen f_1 und f_2 aus Aufgabe P3. Vergleichen Sie die Ergebnisse, die Anzahl der Iterationen und die Laufzeit dieses Programms mit dem Programm aus Aufgabe P3.

Hinweis: Verwenden Sie die `Matlab`-Befehle `tic`, `toc` und `cputime` zur Bestimmung der Laufzeit.

Aufgabe P6 (Optimization Toolbox)

Nutzen Sie die Implementation des BFGS-Verfahrens in der `Matlab Optimization Toolbox` zur Minimierung der Funktionen f_1 und f_2 aus Aufgabe P3:

Die Funktion `[x, fval, exitflag] = fminunc(f, x0, options)` optimiert die Funktion `f` mit Start in `x0` unter Verwendung der Optionen `options`.

Die Rückgabe enthält den optimalen Punkt `x`, den Funktionswert `fval=f(x)` und einen Statuswert `exitflag`.

Mit `options = optimset('param(1)', value(1), ...)` werden in `options` die Parameter `param(i)` auf `value(i)` gesetzt.

Mit `options = optimset(options, 'param1', value1, ...)` können Parameter in `options` geändert werden.

Nutzen Sie folgende Parameter:

- `'LargeScale' : 'off'`
- `'Display' : 'iter'` – Ausgabe erfolgt je Iteration
- `'GradObj' : 'on'` – Benutze Gradienten im zweiten Rückgabe-Argument der Funktion
- `'HessUpdate' : 'bfgs'` – Benutze BFGS-Update

Vergleichen Sie `fminunc` mit Ihrer Implementation aus Aufgabe P5 im Hinblick auf Laufzeit und Anzahl der Iterationen.

Weitere Informationen zur `Matlab Optimization Toolbox`:

<http://www.mathworks.de/access/helpdesk/help/toolbox/optim/>