Optimization for Engineers

2. Lab Exercise

28.05.2018
Summer Term 2018

Dr. Johannes Hild
Department Mathematik
Friedrich-Alexander-Universität
Erlangen-Nürnberg

Assignment 1: Conjugate Gradient Algorithm - 5 Credits

Complete the conjugate gradient algorithm in the template *conjugateGradient.m*, for solving $Ax_s = b$

a) Input: $A \in \mathbb{R}^{n\times n}$; b, $x_0 \in \mathbb{R}^n$; $\varepsilon > 0$.

b) Set $x_k \leftarrow x_0$, $r_k \leftarrow Ax_k - b$ and $d_k \leftarrow -r_k$.

c) While $||r_k|| > \varepsilon$ do

i) Set $\rho_k \leftarrow d_k^\top A d_k$.

ii) Set $t_k \leftarrow -\frac{r_k^\top d_k}{\rho_k}$.

iii) Set $x_k \leftarrow x_k + t_k d_k$.

iv) Set $r_k \leftarrow r_k + t_k A d_k$.

v) Set $\beta_k \leftarrow \frac{r_k^\top A d_k}{\rho_k}$

vi) Set $d_k \leftarrow -r_k + \beta_k d_k$.

d) Return $x_s \leftarrow x_k$.

Hint: Use **norm(x)** for $||x||$.
Test the algorithm with the command **sheet02Script(1);**

Assignment 2: Newton's Method - 5 Credits

Complete Newton's method in the template *newtonsMethod.m*, for minimizing $f : \mathbb{R}^n \to \mathbb{R}$ with Hessian evaluation:

a) Input: $f \in \mathcal{C}^2$; $x_0 \in \mathbb{R}^n$; $\varepsilon > 0$.

b) Set $x_k \leftarrow x_0$.

c) While $||\nabla f(x_k)|| > \varepsilon$ do:

i) Solve $\nabla^2 f(x_k) d_k = -\nabla f(x_k)$ for d_k using **conjugateGradient.m**.

ii) If d_k is a descent direction compute t_k by calling **backtrackingLineSearch.m** for f at x_k along d_k.

iii) Else throw a warning, set d_k to steepest descent and use **bisectionLineSearch.m** for t_k.

iv) Set $x_k \leftarrow x_k + t_k d_k$.

d) Return $x_s \leftarrow x_k$.

Hints:

a) Use **getHessian(f_handle,x_k)** to get the Hessian matrix at x_k.

b) For the conjugateGradient call only provide A and b.

c) Throw a warning with **warning('yourPersonalWarningMessage');**

d) Test the algorithm with the command **sheet02Script(2);**

Evaluation and Upload

Hand in the following files (unzipped) to StudOn using the **Exercises** object again:

a) **conjugateGradient.m**

b) **newtonsMethod.m**