Optimization for Engineers

3. Lab Exercise

11.06.2018
Summer Term 2018

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

Assignment 1: Inexact Newton-CG Algorithm - 5 Credits

Complete the CG descent direction subroutine in the template *CGDirection.m*:

a) Input: $f \in \mathcal{C}^1$; $x_k \in \mathbb{R}^n$; $h > 0$.

b) Set $x_j \leftarrow 0$, $d_j \leftarrow -\nabla f(x_k)$, $r_j \leftarrow \nabla f(x_k)$, $\eta_k \leftarrow \min(\frac{1}{2}, ||\nabla f(x_k)||)$.

c) For $j \leftarrow 1, \ldots, n$ do

i)

$$\tilde{\nabla}^2 f d_j \leftarrow \begin{cases} 0 & \text{if } ||d_j|| < h \\ \frac{||d_j||}{h}(\nabla f(x_k + \frac{h}{||d_j||} d_j) - \nabla f(x_k)) & \text{else} \end{cases}$$

ii) Set $\rho_j \leftarrow d_j^\top \tilde{\nabla}^2 f d_j$.

iii) If $\rho_j \leq 0$: If $j == 1$ return $d_k \leftarrow d_j$, else return $d_k \leftarrow x_j$.

iv) Set $t_j \leftarrow -\frac{r_j^\top d_j}{\rho_j}$, set $x_j \leftarrow x_j + t_j d_j$, set $r_j \leftarrow r_j + t_j \tilde{\nabla}^2 f d_j$.

v) If $||r_j|| < \eta_k ||\nabla f(x_k)||$ return $d_k \leftarrow x_j$.

vi) Set $\beta_j \leftarrow \frac{r_j^\top \tilde{\nabla}^2 f d_j}{\rho_j}$

vii) Set $d_j \leftarrow -r_j + \beta_j d_j$.

d) Return $d_k \leftarrow x_j$.

Hints:

a) **zeros(n,1)** generates a $n \times 1$-vector of zeros.

b) Test the algorithm with the command **sheet03Script(1);**

Assignment 2: Levenberg-Marquardt Method - 5 Credits

Complete the Levenberg-Marquardt algorithm in *levMarq.m*:

a) Input: $R \in \mathcal{C}^1$; $x_0 \in \mathbb{R}^n$; $\alpha_0 > 0$; $\beta > 1$, $\varepsilon > 0$.

b) Set $x_k \leftarrow x_0$, $\alpha_j \leftarrow \alpha_0$.

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

i) Solve $\left(J_k^\top J_k + \alpha_j E_n\right) d_k = -\nabla f(x_k)$ with *conjugateGradient.m*.

ii) If $f(x_k + d_k) < f(x_k)$ accept $x_k \leftarrow x_k + d_k$, reset $\alpha_j \leftarrow \alpha_0$.

iii) Else increase $\alpha_j \leftarrow \beta \alpha_j$.

d) Return $x_s \leftarrow x_k$.

Hints:

a) Use the anonymous functions **R=@(y)getErrorVector(error_handle,y)** and **J=@(y)getErrorJacobian(error_handle,y)**.

b) Remember $f(x_k) = \frac{1}{2} R(x_k)^\top R(x_k)$ and $\nabla f(x_k) = J(x_k)^\top R(x_k)$.

c) E_n is the unit matrix of size n. Use **eye(n)** in Matlab.

d) Test the algorithm with the command **sheet03Script(2);**

Evaluation and Upload

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

a) *CGDirection.m*

b) *levMarq.m*