MS-E2122 - Nonlinear optimization **DEADLINE:**
Homework 5 **Thursday 13.12.2018 23:55**

Please submit your answers to https://mycourses.aalto.fi/mod/assign/view.php?id=391886 before the given deadline.

Homework 5.1 Application of the Conjugate Gradient Method

Implement the Conjugate Gradient method introduced in Lecture 8 using exact line search (e.g., univariate Newton's method with $\lambda = 0$ as the initial value). Compare the performance of the Conjugate Gradient method on the following two functions from Project Work 1:

(a) $f(x_1, x_2) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$ with starting point $(x_1, x_2) = (7, 3)$

(b) $f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$ with starting point $(x_1, x_2) = (1, 1.5)$

Report the number of steps in both cases and briefly discuss if there are any theoretical justification for these numbers. **Make sure to also submit your Conjugate Gradient implementation as a Julia code file.**

Homework 5.2 Interior-Point Method for Quadratic Problems

Consider the following quadratic optimization problem with equality constraints:

$$(P) \;:\; \text{minimize} \;\; c^\top x + \frac{1}{2}x^\top Q x \tag{1}$$
$$\text{subject to} \;\; Ax = b \tag{2}$$
$$x \geq 0 \tag{3}$$

with variables $x \in \mathbf{R}^n$. We assume that $Q \in \mathbf{R}^{n\times n}$ is a symmetric positive semidefinite matrix, and that $c \in \mathbf{R}^n$, $A \in \mathbf{R}^{m\times n}$, and $b \in \mathbf{R}^m$. The dual problem of (1) – (3) is

$$(D) \;:\; \text{maximize} \;\; b^\top v - \frac{1}{2}x^\top Q x \tag{4}$$
$$\text{subject to} \;\; A^\top v + u - Qx = c \tag{5}$$
$$u \geq 0 \tag{6}$$

with dual variables $v \in \mathbf{R}^m$ and $u \in \mathbf{R}^n$. We want to solve the problem (1) – (3) with a similar primal-dual interior-point method as described in Lecture 10 for LP problems.

(a) Write the KKT conditions of the problem (1) – (3)

(b) Formulate the barrier problem for (1) – (3) using the logarithmic barrier function. Write the KKT conditions of the barrier problem using the same notation as in Lecture 10, slide 15

(c) Write the Newton system based on the KKT conditions of part (b), and derive update formulas for directions d_v, d_x, and d_u similarly as in Lecture 10, slides 19 – 20.

Hint: First derive a formula for d_v, then for d_x which uses the already computed d_v, and finally for d_u which uses the already computed d_v and d_x. This allows us to solve the Newton directions efficiently without explicit matrix inversions. Compare with Lecture 10, slide 20 where the order is d_v, d_u, and d_x instead.

(d) Using the update rules derived in part (c), modify this skeleton file to solve problems of the form (1) – (3) with the interior point method. Basically, you have to modify the function which computes the Newton directions based on the formulas derived in part (c). Solve the problem instance given in the skeleton file.

(e) **Remember to also return your Julia code file when submitting your answers.**

Homework 5.3 Sequential Quadratic Programming

Consider the quadratic optimization problem studied in Exercise 11.2:

$$\begin{aligned}
\underset{x}{\text{minimize}} \quad & f(x) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 && (7)\\
\text{subject to} \quad & g_1(x) = x_1^2 - x_2 \le 0 && (8)\\
& g_2(x) = x_1 + 5x_2 - 5 \le 0 && (9)\\
& g_3(x) = -x_1 \le 0 && (10)\\
& g_4(x) = -x_2 \le 0 && (11)
\end{aligned}$$

(a) Solve the problem (7) – (11) using the l_1-SQP variant described in Lecture 11, slide 27. Use $x^1 = (0.5, 0.5)^\top$ as the initial primal solution and $u^1 = (0, 0, 0, 0)^\top$ as the initial dual solution. You can implement the l_1-SQP variant by modifying the Julia code of the basic SQP presented in Exercise 11. Try to find suitable values for the penalty parameter μ and the trust region parameter Δ^k to obtain convergence. You can keep the trust region parameter Δ^k constant even though it has the index k.

(b) Plot the feasible region defined by the inequalities (8) – (11) and the progress of the l_1-SQP iterations in the same plot.

(c) **Remember to also return your Julia code file when submitting your answers.**