Norwegian University of Science and Technology
Department of Engineering Cybernetics

Exercise 10
Solution

Problem 1 (30 %) The QP approximation

When developing a local SQP method, we approximate the NLP

$$\min_{x} \quad f(x) \tag{1a}$$
$$\text{s.t.} \quad c(x) = 0 \tag{1b}$$

($f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions) as the QP

$$\min_{p} \quad f_k + \nabla f_k^\top p + \frac{1}{2} p^\top \nabla_{xx}^2 \mathcal{L}_k p \tag{2a}$$
$$\text{s.t.} \quad A_k p + c_k = 0 \tag{2b}$$

at the iterate (x_k, λ_k). See Section 18.1 in the textbook.

a As explained in Section 18.1, we first replace the NLP (1) by the problem of minimizing the Lagrangean function subject to the equality constraints (2b); that is, the problem

$$\min_{x} \quad \mathcal{L}(x, \lambda) \tag{3a}$$
$$\text{s.t.} \quad c(x) = 0 \tag{3b}$$

We then linearize the constraints and make a quadratic approximation to the Lagrangean. A linearized version of constraint i at the point x_k is

$$c_i(x_k + p) \approx c_i(x_k) + \nabla c_i(x_k)^\top p = 0 \tag{4}$$

We have that $A(x)$ is the Jacobian matrix of the constraints, meaning

$$A(x)^\top = \begin{bmatrix} \nabla c_1(x), & \nabla c_2(x), & \dots, & \nabla c_m(x) \end{bmatrix} \tag{5}$$

The set of constraints can then be written in linearized form as

$$c(x_k + p) \approx c(x_k) + A(x_k)p = 0 \tag{6}$$

Similarly, a quadratic approximation of the Lagrangean at (x_k, λ_k) in x takes the form

$$
\begin{aligned}
\mathcal{L}(x_k+p,\lambda_k) &\approx \mathcal{L}(x_k,\lambda_k)+\nabla_x\mathcal{L}(x_k,\lambda_k)^\top p+\frac{1}{2}p^\top\nabla^2_{xx}\mathcal{L}_k p \\
&= \left(f_k-\lambda_k^\top c_k\right)+\left(\nabla f_k - A_k^\top\lambda_k\right)^\top p+\frac{1}{2}p^\top\nabla^2_{xx}\mathcal{L}_k p \\
&= f_k-\lambda_k^\top c_k+\nabla f_k^\top p-\lambda_k^\top A_k p+\frac{1}{2}p^\top\nabla^2_{xx}\mathcal{L}_k p \\
&= f_k+\nabla f_k^\top p-\lambda_k^\top\underbrace{(c_k+A_k p)}_{0}+\frac{1}{2}p^\top\nabla^2_{xx}\mathcal{L}_k p \\
&= f_k+\nabla f_k^\top p+\frac{1}{2}p^\top\nabla^2_{xx}\mathcal{L}_k p
\end{aligned}
\tag{7}
$$

Hence, we approximate (3) by

$$
\min_p \quad f_k+\nabla f_k^\top p+\frac{1}{2}p^\top\nabla^2_{xx}\mathcal{L}_k p \tag{8a}
$$

$$
\text{s.t.} \quad c_k+A_k p=0 \tag{8b}
$$

b The KKT conditions for the equality-constrained QP (2) are

$$
\begin{bmatrix}\nabla^2_{xx} & -A_k^\top \\ A_k & 0\end{bmatrix}\begin{bmatrix}p_k \\ l_k\end{bmatrix}=\begin{bmatrix}-\nabla f_k \\ -c_k\end{bmatrix} \tag{9}
$$

where l_k are the Lagrangean multipliers for the problem (2). Note that λ are the multipliers for the problem (1). The KKT matrix equation for an equality-constrained QP is derived in Section 16.1.

Problem 2 (40 %) Merit functions

In SQP we use a merit function to assess the quality of a step, combining the objective with a measure of constraint violation (see Section 15.4 in the textbook).

a In the conditional for the while loop. See equation (18.28) in the textbook.

b The merit functions ϕ_1, ϕ_2, and ϕ_F are

$$
\phi_1(x;\mu)=f(x)+\mu\sum_{i\in\mathcal{E}}|c_i(x)|+\mu\sum_{i\in\mathcal{I}}[c_i(x)]^-, \quad [z]^-=\max\{0,-z\} \tag{10a}
$$

$$
\phi_2(x;\mu)=f(x)+\mu\|c(x)\|_2 \tag{10b}
$$

$$
\phi_F(x;\mu)=f(x)-\lambda(x)^\top c(x)+\frac{1}{2}\mu\sum_{i\in\mathcal{E}}c_i(x)^2 \tag{10c}
$$

c The positive scalar parameter μ is a *penalty parameter*. It determines the weight we assign to constraint satisfaction (feasibility) relative to minimization of $f(x)$.

It is common that μ is small initially and grows larger as the SQP algorithm approaches the solution.

d An merit function is exact if there is a positive scalar μ^* such that for any $\mu > \mu^*$, any local solution of the NLP is a local minimizer of the merit function (Definition 15.1).

All of the three merit functions ϕ_1, ϕ_2, and ϕ_F are exact.

e The Maratos effect is the phenomenon of the merit function rejecting steps that make good progress toward a solution, which may cause the SQP algorithm to fail to converge rapidly.

The merit functions ϕ_1 and ϕ_2 suffer from the Maratos effect; the merit function ϕ_F *does not* suffer from the Maratos effect

f The merit function will generally decrease from one iteration to the next. This is because it is used in the line-search part of an SQP algorithm; the line search does not terminate until a decrease in the merit function is observed.

g The objective function *may not* decrease from one iteration to the next. This is because a step may make good progress toward a solution even though the objective function value increases. This is different from what happens in unconstrained problems, LP problems, and QP problems, where the objective function does not increase from one iteration to the next.

Problem 3 (30 %) Feasibility and local solutions

Assume that Algorithm 18.3 is used to solve an NLP problem of the form

$$\min_x \quad f(x) \tag{11a}$$
$$\text{s.t.} \quad c_i(x) = 0, \quad i \in \mathcal{E} \tag{11b}$$
$$c_i(x) \geq 0, \quad i \in \mathcal{I} \tag{11c}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c_i : \mathbb{R}^n \to \mathbb{R}$ are smooth functions.

a The initial point x_0 does not have to be feasible. This is because an SQP algorithm makes no assumptions on feasibility of x_0, and a feasible x_0 would not necessarily help the algorithm. In fact, from a feasible point that is far away from the solution would in most cases be worse than starting infeasible but close to the solution.

This is different from the SIMPLEX algorithm for LP problems and the active set method for QP problems. Both of these algorithms require feasible starting points, and finding a feasible starting point (through a Phase I problem) is generally as difficult as finding the solution to the optimization problem.

b Most iterates (x_k, λ_k) will in general be infeasible with an SQP algorithm. Requiring a feasible path could lead slower convergence. Instead, we use a merit function that ensures "moderate" infeasibility.

This is different from the SIMPLEX algorithm for LP problems and the active set method for QP problems. Neither of these algorithms produce iterates that exit the feasible region.

c The NLP problem (11) can have multiple local solutions if it is noncenvex, meaning the objective function is nonconvex, and/or one or more of the equality constraint functions are nonlinear, and/or one or more of the inequality constraint functions are nonconcave. Note that a nonconvex problem does not necessarily have multiple local solutions.

d A very simple method of looking for multiple (better) solutions is to start run the SQP algorithm many times, each time from different starting points. If the problem has many local solutions, the one found by the SQP algorithm may depend on where the SQP algorithm starts.