Optimalisering og regulering TTK4135 - Assignment 10

Ola Kristoffer Hoff

1 May, 2023

1 The QP approximation

When developing a local SQP method, we approximate the NLP

$$\min_x \ f(x) \tag{1a}$$

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

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

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

$$s.t. \quad A_k p + c_k = 0 \tag{2b}$$

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

a Explain how we arrive at the approximation and derive both the objective function and the constraint in the quadratic program (2).

The gist of how the method works is that it we linearise the NLP at a point and find the best direction from here, then we do this iteratively until we converge upon a solution.

Let's first note the definition of the Lagrangian and the Jacobian of the constraints, A.

$$\begin{aligned} \mathcal{L}(x, \lambda) &= f(x) - \lambda^T c(x) \\ A(x)^T &= \begin{bmatrix} \nabla c_1(x) & \cdots & \nabla c_m(x) \end{bmatrix} \end{aligned}$$

We now linearise the constraints:

$$\begin{aligned} c(x+p) &\approx c(x) + \nabla c(x)^T p = 0 \\ c(x+p) &\approx c(x) + A(x) p = 0 \\ c(x_k) + A(x_k) p &= 0 \\ A_k p + c_k &= 0 \end{aligned}$$

Thus we have the constraints. Now doing the same linearisation on the objective function:

$$\begin{aligned} \mathcal{L}(x+p, \lambda) &\approx \mathcal{L}(x, \lambda) + \nabla_x \mathcal{L}(x, \lambda)^T p + \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}(x, \lambda) p \\ &= \mathcal{L}_k + \nabla_x \mathcal{L}_k^T p + \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}_k p \\ &= (f_k - \lambda_k^T c_k) + (\nabla f_k - A_k^T \lambda_k)^T p + \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}_k p \\ &= f_k - \lambda_k^T c_k + (\nabla f_k^T - \lambda_k^T A_k) p + \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}_k p \\ &= f_k - \lambda_k^T c_k + \nabla f_k^T p - \lambda_k^T A_k p + \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}_k^T p \\ &= f_k + \nabla f_k^T p - \lambda_k (c_k + A_k p) + \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}_k p \\ &= f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}_k p \end{aligned}$$

We used the fact that $c_k + A_k p = 0$.

Thus we have the objective function on the correct form as well.

b State the KKT conditions for the quadratic program (2) as a matrix equation.

The matrix form is denoted as:

$$\begin{aligned} \min_x \; q(x) &\stackrel{\text{def}}{=} \frac{1}{2}x^T G x + x^T c \\ s.t \quad Ax &= b \end{aligned}$$

$$\begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix}$$

In out case we get:

$$\begin{bmatrix} \nabla^T_{xx}\mathcal{L}_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_k \end{bmatrix} = \begin{bmatrix} -\nabla f_k^T \\ -c_k \end{bmatrix}$$

2 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 Where in the SQP algorithm (Algorithm 18.3 in the textbook) do we use the merit function?

We use the merit function in the while-conditional of the algorithm.

b Write down the merit functions ϕ_1, ϕ_2, and ϕ_F (Section 15.4).

$$\begin{aligned} \phi_1(x;\mu) &= f(x) + \mu\sum_{i\in\mathcal{E}}|c_i(x)| + \mu\sum_{i\in\mathcal{I}}[c_i(x)]^- \\ \phi_2(x;\mu) &= f(x) + \mu||c(x)||_2 \\ \phi_F(x;\mu) &= f(x) - \lambda(x)^T c(x) + \frac{1}{2}\mu\sum_{i\in\mathcal{E}} c_i(x)^2 \end{aligned}$$

c What is the purpose of the parameter μ? Does it change during the course of the SQP algorithm?

Here $\mu > 0$, it's a penalty parameter. It starts of smaller and grows when needed to fulfil the relation in 18.36 [1].

d Explain the concept *exact merit function*. Which of the three merit functions are exact?

"A merit function $\phi(x;\mu)$ is exact if there is a positive scalar μ^* such that for any $\mu > \mu^*$, any local solution of the nonlinear programming problem (15.1) is a local minimizer of $\phi(x;\mu)$."[1](page. 435, Definition 15.1).

Hence, all three of the merit functions are exact.

e What is the Maratos effect? Which of the three merit functions mentioned above suffer from the Maratos effect?

"Some algorithms based on merit functions or filters may fail to converge rapidly because they reject steps that make good progress toward a solution. This undesirable phenomenon is often called the Maratos effect..."[1](page. 440).

ϕ_1 and ϕ_2 are subject to the Maratos effect due to their non-negative functions of $c(x)$. ϕ_F is not subject to the effect.

f Will the merit function generally decrease from one iteration to the next? Explain.

Yes, the way the algorithm is written it will not find a line-search direction until a decrease in the merit function is observed.

g Will the objective function generally decrease from one iteration to the next? Explain. Is this different from or the same as what we are used to from unconstrained problems, LP problems, and QP problems?

The objective function might not decrease. This is due to the fact that we don't operate directly on the objective function, but on the linearised version of it. It will however make a step in the correct direction, an thus take a shorter path "over a hill" instead of around it.

3 Feasibility and local solutions

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

$$\min_x \ f(x) \tag{3a}$$

$$s.t \quad c_i(x) = 0, \ \ i \in \mathcal{E} \tag{3b}$$

$$c_i(x) \geq 0, \ \ i \in \mathcal{I} \tag{3c}$$

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

a The user has to supply a starting point (x_0, λ_0). Is it important that x_0 is feasible? Explain. Is this different from or the same as what we are used to from LP problems and QP problems?

One of the neat things about the SQP algorithm is that it does not require feasibility for the initial point. This could result in better running due to the fact we can start closer to the actual solution, compared to a feasible point further away. This is different from the other methods we know, these need a feasible starting point.

b Are all iterates (x_k, λ_k) feasible? Explain. Is this different from or the same as what we are used to from LP problems and QP problems?

No, there is no requirement for the points to be feasible, this would force the paths to a more restricted domain leading to slower convergence. However, due to the merit function the points are some what bound to the feasible solution, meaning they don't stray of too far. Again, this is different from the other methods we have seen, since they all require that all points are feasible.

c Under which conditions can the NLP problem (3) have multiple local solutions?

There are a few conditions that would make the problem have multiple local solutions:

- The objective function is not convex.
- Has at least one non-linear equality constraint.
- Has at least one non-concave inequality constraint.

If one or more of these conditions are true, then the NLP can have multiple local solutions.

d Assume that we have a problem of the form (3) that we have solved with SQP. The solution x^* returned by the SQP algorithm does not impress you and you know the problem may have multiple local solutions. How would you try to find a better solution? Suggest a very simple method.

The simplest way is to do multiple runs all with different starting points. Thus, if there is multiple local solutions, due to probability, one will likely find other solutions. This is in essence the same as when it rains in an area. Each drop hits the surface at a "unique" point, then travels don to the local minima, and hence we have puddles (solutions).

References

[1] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Science+Business Media, LLC, 2006.