Sequential Quadratic Programming

Zamboni Riccardo

e-Novia, Italy, ricc.zamboni@gmail.com

Keywords: SQP, Optimization, Optimal Control

Abstract

Optimal Control techniques are well understood methods which can lead to optimality with respect to a specific metrics (often L1 or L2), by exploiting a fully generative mode lvirtually without any sampling cost. More computationally plausible algorithms often make use of sequentiality, and SQP is an instance of such techniques. We will briefly dive in this method.

1 Introduction

For convenience, a typical SQP algorithm is reported. Before the algorithm, we also recall the L1-norm merit function:

$$T_1(x) = f(x) + \sum_{i=1}^{p} \sigma_i |g_i(x)| + \sum_{i=1}^{q} \tau_i \max(0, -h_i(x))$$

and its directional derivative, already simpli
ed considering that p^{k^*} is the solution to a QP featuring the linearized equality and inequality constraints:

$$\begin{aligned} D(T_1(x^k), p^{k*}) = \nabla_x f(x^k)^T p^{k*} - \sum_{i=1}^{p} \sigma_i |g_i(x^k)| \\ - \sum_{i=h_i(x^k)<0}^{q} \tau_i \nabla_x h_i(x^k)^T p^{k*} \end{aligned} \tag{1}$$

1.1 Constrained continuous NLP solver with SQP method

- **Inputs**: cost function $f : \mathcal{R}^n \leftarrow \mathcal{R}$ (and $F(x)$ for Gauss-Newton method, see below), equality constraints $g : \mathcal{R}^n \leftarrow \mathcal{R}^p$, inequality constraints $h : \mathcal{R}^n \leftarrow \mathcal{R}^q$, initial guess $x^0 \in \mathcal{R}^n$, initial values of σ_i, τ_j for the L1-norm merit function, $i = 1, \cdots, p$, $j = 1, \cdots, q$
- **Parameters**: termination tolerances $TOL_\nabla, TOL_x, TOL_f \in (0;1), TOL_{constr} > 0$, maximum number of iterations $N_{max} \in \mathcal{N}$; line-search sub-routine parameters, additional parameters according to the employed Hessian approximation method (e.g. Gauss-Newton, BFGS).
- **Initialization**: $\Delta x^0 = 1, \Delta f^0 = 1, \lambda^0 = 0, \mu^0 = 0$

Algorithm:

1. Compute $\nabla_x f(x^k), \nabla_x g(x^k), \nabla_x h(x^k)$ with a differentiation method of choice (for BFGS, these are already available except for iteration k = 0, see item 7. below.
2. Check the terminating conditions:
   - if $|\nabla \mathcal{L}(x^k, \lambda^k, \mu^k)^T p^k| \leq TOL_\nabla$ **or** $\Delta f^k \leq TOL_f$ **or** $k \geq N_{max}$ **and** $||g(x^k)||_\infty \leq TOL_{constr}$ **and** $min(h(x^k)) \geq TOL_{constr}$ **then** exit.
     In this case, either $x^* = x^k$ is a possible local minimizer (and $f(x^*)$ a local minimum), or the algorithm has ended prematurely but still returns a feasible point.
   - if $k \geq N_{max}$ **and** $||g(x^k)||_\infty > TOL_{constr}$ **and** $min(h(x^k)) < TOL_{constr}$ **then** exit.
     In this case, the algorithm has ended prematurely at an unfeasible point, or possibly the problem is not feasible.
3. Compute the QP Hessian H^k as:
   - **Constrained Gauss-Newton**:
     $H^k = 2\nabla_x F(x^k) \nabla_x F(x^k)^T$, where it is assumed that $f(x) = F(x)^T F(x)$ and $F : \mathcal{R}^n \leftarrow \mathcal{R}^N$ is provided by the user. To ensure that H^k is strictly positive definite, one can also modify it by adding a small positive quantity σ(e.g. 10^{-14}) on the diagonal.
   - **BFGS**: H^k computed with the BFGS update rule (see item 7. below), where $H^0 = I$;
4. Solve the QP:

$$\begin{aligned} \min_{p^k} \nabla_x f(x^k)^T p^k + \frac{1}{2} p^{kT} H^k p^k \\ s.t. \\ \nabla_x g(x^k)^T p^k + g(x^k) = 0 \\ \nabla_x h(x^k)^T p^k + h(x^k) \geq 0 \end{aligned} \tag{2}$$

   Denote with $p^{k*}, \tilde{\lambda}^{k*}, \tilde{\mu}^{k*}$ the obtained solution and Lagrange multipliers, and compute

$$\begin{aligned} \Delta\lambda^k = (\tilde{\lambda}^{k*} - \lambda^k) \\ \Delta\mu^k = (\tilde{\mu}^{k*} - \mu^k) \end{aligned} \tag{3}$$

   with $\lambda^0, \mu^0 = 0$.

5. Update the weights σ_i, τ_i for the merit function $T_1(x)$ as:

$$\begin{aligned} \sigma_i &= max(|\tilde{\lambda}^{k*}|, \frac{\sigma_i^- + |\tilde{\lambda}^{k*}|}{2}) \\ \tau_i &= max(|\tilde{\mu}^{k*}|, \frac{\tau_i^- + |\tilde{\mu}^{k*}|}{2}) \end{aligned} \tag{4}$$

   for all $i = 1, \cdots, p$ and all$j = 1, \cdots, q$, where σ_j^-, τ_j^- are the previous values.

6. Compute the directional derivative $D(T_1(x^k); p^{k*})$ as in (3), and carry out the back-tracking line search using the merit function $T_1(x)$ and directional derivative $D(T_1(x^k); p^{k*})$ to compute t^k. Then update:

$$\begin{aligned} x^{k+1} &= x^k + t^k p^{k*} \\ \lambda^{k+1} &= \lambda^k + t^k \Delta\lambda^k \\ \mu^{k+1} &= \mu^k + t^k \Delta\mu^k \end{aligned} \tag{5}$$

7. Only for BFGS:

   - Compute the gradients $\nabla_x f(x^{k+1})$, $\nabla_x g(x^{k+1})$, $\nabla_x h(x^{k+1})$ and:

$$\begin{aligned} \nabla_x \mathcal{L}(x^{k+1}, \lambda^{k+1}, \mu^{k+1}) &= \nabla_x f(x^{k+1}) \\ -\nabla_x g(x^{k+1})\lambda^{k+1} &- \nabla_x h(x^{k+1})\mu^{k+1} \\ \nabla_x \mathcal{L}(x^k, \lambda^{k+1}, \mu^{k+1}) &= \nabla_x f(x^k) \\ -\nabla_x g(x^k)\lambda^{k+1} &- \nabla_x h(x^k)\mu^{k+1} \end{aligned} \tag{6}$$

     Then compute

$$\begin{aligned} y = (\nabla_x \mathcal{L}(x^{k+1}, \lambda^{k+1}, \mu^{k+1}) \\ -\nabla_x \mathcal{L}(x^k, \lambda^{k+1}, \mu^{k+1})) \\ s = (x^{k+1} - x^k) \end{aligned} \tag{7}$$

   - If $y^T s \leq \gamma s^T H^k s$ (where $\gamma \in (0, 1)$), then compute $\tilde{y} = y + \frac{\gamma s^T H^k s - s^T y}{s^T H^k s - s^T y}(H^k s - y)$
   - Compute $H^{k+1} = H^k - \;+\;$ or using $\tilde{y}$ instead of y, see previous step))S

8. Update the relative changes of optimization variables and cost function, $\Delta x + k$, $\Delta f + k$ as $\Delta x^k = \frac{||x^{k+1} - x^k||}{max(\epsilon, ||x^k||)}$, $\Delta f^k = \frac{|f(x^{k+1}) - f(x^k)|}{max(\epsilon, |f(x^k)|}$.

9. set k = k + 1, go to 1.

This algorithm either terminates with the value x^* of a possible local minimizer, or it stops due to too little progress (in either the relative change of local minimizer, or the relative improvement of the cost function) or too many iterations. In addition, the corresponding cost function value $f(x^*)$ and the number of computed iterations k are available as outputs.