OPTIMIZATION - OPTIMALISATIE

EXERCISE SESSION 7: SEQUENTIAL QUADRATIC PROGRAMMING (SQP) METHODS

Exercise 1 (**Equality Constrained SQP).**

Consider the problem:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\textbf{minimize}} \quad & f(x) \\ \textbf{subject to} \quad & h(x) = 0 \end{aligned}$$

with $f\colon \mathbb{R}^n \to \mathbb{R}$, $h\colon \mathbb{R}^n \to \mathbb{R}^m$.

Newton Lagrange Method: We use Lagrange multipliers $\lambda \in \mathbb{R}^m$ to handle the equality constraints:

$$L(x, \lambda) = f(x) + \lambda^T h(x).$$

The optimality conditions are given by:

$$\begin{aligned} 0 &= \nabla_x L(x, \lambda) = \nabla_x f(x) + \nabla_x h(x)\lambda \\ 0 &= \nabla_\lambda L(x, \lambda) = h(x). \end{aligned}$$

Applying Newton's method on the KKT system yields for a given x_k and λ_k:

$$\begin{pmatrix} \nabla^2_{xx} L_k & \nabla h_k \\ \nabla h_k^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla f_k \\ -h_k \end{pmatrix}. \tag{1.1}$$

By solving the linear system (1.1), we obtained Δx and λ. The next iterate is calculated by

$$\begin{aligned} x_{k+1} &= x_k + \Delta x, \\ \lambda_{k+1} &= \lambda. \end{aligned}$$

QP Interpretation: Consider the quadratic program:

$$\begin{aligned} \underset{\Delta x}{\textbf{minimize}} \quad & \Delta x^T \nabla f_k + \frac{1}{2} \Delta x^T \nabla^2_{xx} L_k \Delta x \\ \textbf{subject to} \quad & \nabla h_k^T \Delta x + h_k = 0 \end{aligned} \tag{1.2}$$

Introduce Lagrange multipliers λ:

$$L(\Delta x, \lambda) = \Delta x^T \nabla f_k + \frac{1}{2} \Delta x^T \nabla^2_{xx} L_k \Delta x + \lambda^T (\nabla h_k^T \Delta x + h_k)$$

and the optimality conditions are:

$$\begin{aligned}\nabla f_k + \nabla^2_{xx} L_k \Delta x + \nabla h_k \lambda &= 0\\ \nabla h_k^T \Delta x + h_k &= 0.\end{aligned}$$

Rewriting this as a linear system, we have:

$$\begin{pmatrix} \nabla^2_{xx} L_k & \nabla h_k \\ \nabla h_k^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla f_k \\ -h_k \end{pmatrix}. \tag{1.3}$$

Since this KKT system (1.3) is identical to the one we obtained in (1.1), we can use a QP solver to directly solve (1.2) instead of solving (1.1) ourselves.

Tasks:

(1a) An example solution to Exercise 4.2 is provided in `minimize_eq_sqp.py`. Convert this file to use the QP solver `OSQP` interfaced through the package Casadi, instead of solving for (1.1). A description of the Casadi `conic` function can be found in https://web.casadi.org/docs/#quadratic-programming.
You will need to get λ from the output of `conic`. As before, use BFGS to approximate the Hessian $\nabla^2_{xx} L$.

(1b) Verify your modification by using your solver on the problem from last time:

$$\begin{aligned} \underset{x,y}{\textbf{minimize}} \quad & \frac{1}{2}\left(x^2 + \left(\frac{y}{2}\right)^2\right) \\ \textbf{subject to} \quad & y = (x-1)^2 - x + 3 \end{aligned}$$

The file `s7_ex1_eq_sqp.py` sets up the problem for you.

(1c) Solve the problem first with line search, then with full step. What is the difference?

(1d) Add Powell's trick from Section 12.4 of the script [1] to your algorithm. Solve the same problem (both with and without line search) and compare the plots.

Exercise 2 (Inequality Constrained SQP).

Consider the inequality constrained problem:

$$\begin{aligned} \underset{x}{\textbf{minimize}} \quad & f(x) \\ \textbf{subject to} \quad & h(x) = 0 \\ & g(x) \leq 0 \end{aligned}$$

[1] https://p.cygnus.cc.kuleuven.be/bbcswebdav/pid-30052757-dt-content-rid-307696315_2/xid-307696315_2.

with $f\colon \mathbb{R}^n \to \mathbb{R}$, $h\colon \mathbb{R}^n \to \mathbb{R}^m$, and $g\colon \mathbb{R}^n \to \mathbb{R}^l$.
Introduce Lagrange multipliers $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^l$ to handle the constraints:

$$L(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x).$$

The optimality conditions are:

$$\begin{aligned}
0 &= \nabla f(x) + \nabla h(x)\lambda + \nabla g(x)\mu \\
0 &= h(x) \\
0 &\geq g(x) \\
0 &\leq \mu \\
0 &= \mu^T g(x).
\end{aligned}$$

In the equality-constrained case we linearized and solved the optimality conditions either by solving (1.1) by hand, or by solving (1.2) with a QP solver. It is a pain to solve something with inequalities by hand, so here we let the QP solver do the work. The equivalent QP is:

$$\begin{aligned}
\underset{\Delta x}{\textbf{minimize}} \quad & \Delta x^T \nabla f_k + \frac{1}{2}\Delta x^T \nabla^2_{xx} L_k \Delta x \\
\textbf{subject to} \quad & \nabla h_k^T \Delta x + h_k = 0 \\
& \nabla g_k^T \Delta x + g_k \leq 0
\end{aligned} \tag{2.1}$$

Tasks:

(2a) Add inequality constraints to your SQP solver by completing the example code given in `minimize_sqp.py`. You will need to get λ and μ from the output of `conic`. Use BFGS with Powell's trick to approximate the Hessian $\nabla^2_{xx} L$. For the line search merit function, use:

$$\varphi_1(x) := f(x) + c\left(\|h(x)\|_1 + \|g^+(x)\|_1\right)$$

and

$$\nabla \varphi_1(x_k)^T d_k = \nabla f(x_k)^T d_k - c\left(\|h(x_k)\|_1 + \|g^+(x_k)\|_1\right),$$

where

$$g^+(x) = \mathbf{max}(0, g(x)) \qquad \text{(pointwise)}.$$

Use as stopping criterion:

$$\|\nabla f_k + \nabla h_k \lambda_k + \nabla g_k \mu_k\|_\infty \leq \epsilon$$

and

$$\left\| \begin{pmatrix} h_k \\ g_k^+ \end{pmatrix} \right\|_\infty \leq \epsilon.$$

(2b) Verify your modification by solving

$$\begin{aligned}
\underset{x,y}{\textbf{minimize}} \quad & \frac{1}{2}\left(x^2 + \left(\frac{y}{2}\right)^2\right) \\
\textbf{subject to} \quad & y = (x-1)^2 - x + 3 \\
& 0 \leq 2x - 0.4x^2 - y
\end{aligned}$$

The file `s7_ex2_sqp.py` sets up the problem for you.

(2c) Try solving the problem with both line search and full step. What is the difference?

Exercise 3 (Hanging chain, one last time).

We will solve the hanging chain one last time. In this case we will consider a completely inelastic chain where the distance between links is fixed, and the chain is partially resting on an inclined table with slope 0.15:

$$\begin{aligned}
\underset{x_1..x_N, y_1..y_N}{\textbf{minimize}} \quad & \sum_{k=1}^{N} y_k \\
\textbf{subject to} \quad & x_1 = -1 \\
& x_N = 1 \\
& y_1 = 1 \\
& y_N = 1 \\
& (x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2 = r^2, \quad k = 1 \ldots N-1 \\
& y_k \geq 0.15 x_k + 0.3, \quad k = 1 \ldots N
\end{aligned}$$

with $r = 1.4 \cdot 2/N$.

Tasks:

(3a) Complete the file `s7_ex3_sqp_chain.py` to solve this problem. Start by trying to solve this for small N, and see how large you can make N. Be extremely careful about the initial guess. You can use as initial guess `x0=np.linspace(-1,1,N)`, `y0=np.ones(1,N)`.

(3b) Try different initial guesses like `y0=1+0.2*np.cos(x0*pi/2)`. Can you explain this behavior?

(3c) Remove the convex constraint

$$y_k \geq 0.15 x_k + 0.3, \quad k = 1 \ldots N$$

and add the non-convex constraint

$$y_k \geq -0.6 x_k^2 + 0.15 x_k + 0.5, \quad k = 1 \ldots N$$

and solve the problem again.