Exercises for Lecture Course on Numerical Optimal Control (NOC)
Albert-Ludwigs-Universität Freiburg – Winter Term 2025 / 26

# Exercise 3: Equality Constrained Optimization

Prof. Dr. Moritz Diehl, Andrea Zanelli, Dimitris Kouzoupis, Florian Messerer, Yizhen Wang, Armin Nurkanović

In this sheet we will build on the previous exercise by implementing a Newton-type algorithm for equality constrained problems and looking into linear independence constraint qualification.

1. **Newton method for equality constrained problems.** Consider the following equality constrained optimization problem:

$$\min_{x,\, y \in \mathbb{R}} \quad \overbrace{\frac{1}{2}(x-1)^2 + \frac{1}{2}(10(y-x^2))^2 + \frac{1}{2}x^2}^{=:\, f(x,y)} \tag{1a}$$

$$\text{s.t.} \quad \underbrace{x + (1-y)^2}_{=:\, g(x,y)} = 0. \tag{1b}$$

In this exercise we will implement a simple Newton-type algorithm that can be used to solve problem (1).

(a) Compute on paper the gradients of $f$ and $g$ and their Hessian.

(b) Write on paper the Karush-Kuhn-Tucker (KKT) conditions for problem (1). Are these conditions necessary for optimality? Are they sufficient?

(c) In the provided template implement $f$ and $g$ and their Jacobians and Hessians as CasADi functions.

(d) The KKT conditions derived in (b) can be written in compact form as

$$r(w) = 0, \tag{2}$$

where $w := (x, y, \lambda)$ and $\lambda$ is the Lagrange multiplier associated with the equality constraint $g(x, y) = 0$. Using the template provided, implement the following Newton-type method:

$$w_{k+1} = w_k - M^{-1} r(w_k), \tag{3}$$

where $M_k \approx \nabla r(w_k)$ is an approximation of the Jacobian of $r$. One block of $\nabla r(w_k)$ corresponds to the Hessian of the Lagrangian of (1). Test your implementation with two different Hessian approximations: i) $B_k = \rho I_2$ for different values of $\rho$ and ii) the exact Hessian $B_k = \nabla^2 f(x_k, y_k) + \lambda \nabla^2 g(x_k, y_k)$. Initialize the iterates at $w_0 = (1, -1, 1)$ and run the algorithm for $N = 100$ iterations. Plot the iterates in the $x$−$y$ space. When using the fixed Hessian approximation, does the algorithm converge for $\rho = 100$? And for $\rho = 600$?

2. **Linear independence constraint qualification.** Consider the problem of finding the optimal way of throwing a ball as far as possible within a fixed time window $[0, T]$. The dynamics of the system in two dimensional space can be modeled by the differential equation

$$\begin{aligned}
\dot{p}_y &= v_y, \\
\dot{p}_z &= v_z, \\
\dot{v}_y &= -(v_y - w) \left\| v - [w,\, 0]^T \right\| d, \\
\dot{v}_z &= -v_z \left\| v - [w,\, 0]^T \right\| d - g,
\end{aligned}$$

with state $x = (p_y, p_z, v_y, v_z)$, where $p_y$ and $p_z$ represent the $y$ and $z$ coordinate of the ball respectively and $v_y$ and $v_z$ the components of its velocity. The ball is subject to drag force with drag coefficient $d$, side wind $w$ and gravitational acceleration $g$. The initial state is given by $x(0) = \bar{x}_0$, where $\bar{x}_0 = (p_y^0, p_z^0, v_y^0, v_z^0)$. In order to throw the ball, we need to pick the initial velocity $v^0 = (v_y^0, v_z^0)$, for a given initial position. Denote by $p_y^{\mathrm{f}}(v^0), p_z^{\mathrm{f}}(v^0)$ the final position of the ball after simulating forward over time $T$ for the initial velocity $v^0$, using the RK4 scheme for integration. Furthermore, the final position of the ball should be above ground, which has a flat region and a region with slope $\alpha$. We can formulate our goal as the optimization problem

$$\begin{aligned}
\min_{v^0} \quad & - p_y^{\mathrm{f}}(v^0) && \text{(4a)} \\
\text{s.t.} \quad & -p_z^{\mathrm{f}}(v^0) \leq 0, && \text{(4b)} \\
& -\alpha(p_y^{\mathrm{f}}(v^0) - 10) - p_z^{\mathrm{f}}(v^0) \leq 0, && \text{(4c)} \\
& \|v\|_2^2 \leq \bar{v}^2. && \text{(4d)}
\end{aligned}$$

The slope $\alpha$ is an externally set parameter of the optimization problem and may take values as $\alpha \in [-1, 1]$.

(a) Implement the differential equation of the system in the provided `ballistic_dynamics.m`.

(b) Using the provided template, implement and solve the optimization problem for different values of $\alpha \in [-1, 1]$. The template shows how the NLP constructed by CasADi can be parametrized by $\alpha$. Like this the value of $\alpha$ can be conveniently changed without needing to construct a new NLP each time. Plot the normalized gradients of the constraints as three vectors. What happens when $\alpha = 0$? For $\alpha \geq \bar{\alpha}$ for some $\bar{\alpha} \geq 0$ the problem becomes infeasible. What happens to the three vectors as $\alpha$ approaches $\bar{\alpha}$? It is not required to compute the exact value of $\bar{\alpha}$.