Exercises for Lecture Course on Numerical Optimal Control (NOC)
Albert-Ludwigs-Universität Freiburg – Summer Term 2017

Exercise 3: Equality Constrained Optimization

Prof. Dr. Moritz Diehl, Andrea Zanelli, Dimitris Kouzoupis

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:

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

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

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.

(1 point)

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

(1 point)

(c) Implement f and g as CasADi expressions and use them to define two CasADi functions. It is possible to compute the Hessian and Jacobian of expressions as follows:

```
x = MX.sym('x',2,1);
expr = sin(x(1))*x(2);
j = jacobian(expr,x);
h = hessian(expr,x);
```

The obtained expressions can be in turn used to define CasADi functions that return evaluations of the Hessian and Jacobian of an expression:

```
j_fun = Function('j_fun',{x,y},{j});
h_fun = Function('h_fun',{x,y},{h});
```

Implement CasADi functions that take x and y as inputs and return the Jacobians and Hessians of f and g.

Remark: the function `hessian` *only accepts scalar expressions as an input.*

(1 point)

(d) The KKT conditions derived at point (a) can be written in compact form as

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

where $w := [x, y, \lambda]^T$ and λ 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 $M \approx \nabla r(w^k)$ is an approximation of the exact Jacobian of r. Test your implementation with two different Hessian approximations: i) $B = \rho \boldsymbol{I_2}$ for different values of ρ and ii) $B = \nabla^2 f(x^k, y^k) + \lambda \nabla^2 g(x^k, y^k)$. Initialize the iterates at $w^0 = [1, -1, 1]^T$ 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 $\rho = 600$?

(3 points)

2. **Linear independence constraint qualification.** Consider the problem of finding the optimal way of throwing a ball such that progress in the horizontal coordinate after a fixed time T is maximized. The dynamics of the system can be modeled in two dimensions by the following differential equation:

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

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. In order to achieve the desired goal, we formulate the following optimization problem:

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

where $v := [v_y(0),\ v_z(0)]^T$ are the decision variables and $p(T; v)$ is the output of an RK4 integrator that discretizes the dynamics of the system. Additional constraints have been added to the formulation that represent the requirement that the ball has to be above the ground at time T and that it has to be in the half-space defined by the linear constraint (4c), where $\alpha \in [-1,\ 1]$ is a fixed parameter.

(a) Create a MATLAB function that takes the initial velocity of the ball as an input and returns the final position at time T and use it to generate a CasADi expression for $p(T; v)$. Use $N = 100$ equidistant intermediate steps and $T = 0.5\,\text{s}$. Set $d = 0.1\,\text{m}^{-1}$ and $w = 2\,\text{m/s}$.

(1 point)

(b) Using the provided template, solve the optimization problem for different values of $\alpha \in [-1, 1]$ and plot the normalized gradients of the constraints as three vectors. Fix $\bar{v} = 15\,\text{m/s}$ and $p(0) = [0,\ 0]^T$. 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 α approaches $\bar{\alpha}$? It is not required to compute the exact value of $\bar{\alpha}$.

(3 points)

This sheet gives in total 10 points