Nonlinear optimization - Homework 4

Tuomas Porkamaa, 1030671

December 2021

Problem 4.1

Solutions given in HW4_skeleton.ipynb

Problem 4.2

Primal problem:

$$\begin{aligned} \text{(P): min. } & c^T x + \frac{1}{2} x^T Q x \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned} \tag{1}$$

Dual problem of (1):

$$\begin{aligned} \text{(D): max. } & b^T v - \frac{1}{2} x^T Q x \\ \text{s.t. } & A^T v + u - Qx = c \\ & u \geq 0 \end{aligned} \tag{2}$$

(a)

KKT-conditions of (1):

$$\begin{aligned} & Ax = b,\ x \geq 0 \\ & A^T v + u - Qx = c,\ u \geq 0 \\ & u^T x = 0 \end{aligned} \tag{3}$$

(b)

Barrier problem for (1):

$$\begin{aligned} \text{(BP): min. } & c^T x + \frac{1}{2} x^T Q x - \mu \sum_{i=1}^{n} ln(x_i) \\ \text{s.t. } & Ax = b \end{aligned} \tag{4}$$

KKT-conditions of (BP):

$$
\begin{aligned}
&Ax = b,\ x > 0 \\
&A^T v + u - Qx = c \\
&XUe = \mu e
\end{aligned} \tag{5}
$$

where $U = diag(u)$, $X = diag(x)$ and $e = (1, ..., 1)^T$.

(c)

Newton system based on KKT-conditions of (BP):

$$
\begin{bmatrix} A & 0^T & 0 \\ -Q & A^T & I \\ \overline{U} & 0^T & \overline{X} \end{bmatrix} \begin{bmatrix} d_x \\ d_v \\ d_u \end{bmatrix} = - \begin{bmatrix} r_p \\ r_d \\ \overline{XU}e - \mu e \end{bmatrix} \tag{6}
$$

where r_p represent the primal residuals $r_p = A\overline{x} - b$ and r_d represent the dual residuals $r_d = A^T\overline{v} + \overline{u} - Q\overline{x} - c$.

Newton system (6) can be further represented as a system of equations, from where the update formulas for directions d_x, d_v and d_u can be derived.

$$
\begin{aligned}
Ad_x &= -r_p \\
-Qd_x + A^T d_v + d_u &= -r_d \\
\overline{U}d_x + \overline{X}d_u &= \mu e - \overline{XU}e
\end{aligned} \tag{7}
$$

First, we can derive an expression for d_u using the second equation.

$$
\begin{aligned}
-Qd_x + A^T d_v + d_u &= -r_d \\
d_u &= -r_d + Qd_x - A^T d_v
\end{aligned} \tag{8}
$$

Next, we can derive an expression for d_x by applying the above expression for d_u to the third equation of (7)

$$
\begin{aligned}
\overline{U}d_x + \overline{X}d_u &= \mu e - \overline{XU}e \\
\overline{U}d_x + \overline{X}(-r_d + Qd_x - A^T d_v) &= \mu e - \overline{XU}e \\
(\overline{U} + \overline{X}Q)d_x &= \mu e - \overline{XU}e + \overline{X}r_d + \overline{X}A^T d_v \\
d_x &= (\overline{U} + \overline{X}Q)^{-1}(\mu e - \overline{XU}e + \overline{X}r_d + \overline{X}A^T d_v)
\end{aligned} \tag{9}
$$

.

Lastly, we can derive d_v by applying (9) to the first equation of (7).

$$
\begin{aligned}
Ad_x &= -r_p \\
A(\overline{U} + \overline{X}Q)^{-1}(\mu e - \overline{XU}e + \overline{X}r_d + \overline{X}A^T d_v) &= -r_p \\
A(\overline{U} + \overline{X}Q)^{-1}\overline{X}A^T d_v &= -r_p - A(\overline{U} + \overline{X}Q)^{-1}(\mu e - \overline{XU}e + \overline{X}r_d) \\
d_v &= (A(\overline{U} + \overline{X}Q)^{-1}\overline{X}A^T)^{-1}(-r_p - A(\overline{U} + \overline{X}Q)^{-1}(\mu e - \overline{X}Ue + \overline{X}r_d))
\end{aligned} \tag{10}
$$

The code produced following convergence plot. Based on the visual interpretation, it looks like the IPM converged successfully to the inside-feasible-region minima, which is located at the corner of the feasible region that is closest to the outside-feasible-region minima of the quadratic function

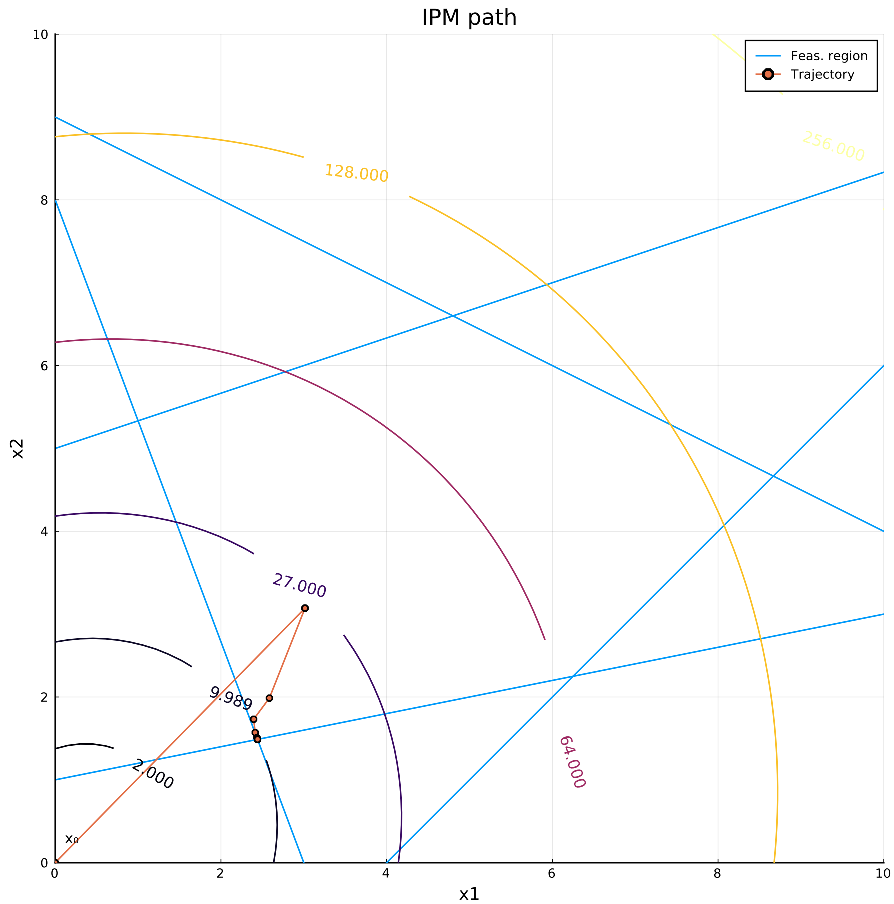

Figure 1: Convergence trajectory of IPM

Problem 4.3

(a)

Implementation given in HW4_skeleton.ipynb. The given parameters $\mu = 10$ and $\Delta_k = 1$ seemed to work fine, and they provided quite fast convergence in iteration-specific subproblem and overall sense. The overall problem was solved in 5 iterations, and the average number of iterations for subproblem was 5.6.

I also experimented with other parametrizations via decreasing or increasing μ and Δ_k respectively, until I noted some deviant behaviour. By decrasing the μ all to way to $\mu = 2$, the algorithm still converges with 5 iterations, but the subproblem iteration counts were smaller, on average 4.8. Alternatively, while increasing μ we can observe that the subproblem iterations also increase by some amount. By decrasing Δ_k, we can see that the algorithm requires a lot more iterations to converge after certain value (e.g. $\Delta_k < 0.2$), which is expected behaviour because the new direction vector will have very little space to change its value. When Δ_k was set to relatively small value (e.g. [0.25 1.0]) we can see that the temporal solution x_k never leaves the feasible region. Alternatively, even very aggressive increase of Δ_k didn't seem to affect the convergence iteration count (it stays at 5), but the subproblem specific iterations steadily increased as well. Overall, in terms of iteration counts until converge, I found good parameter settings as $\mu = 2$ and $\Delta_k \in [0.25, 1.0]$.

(b)

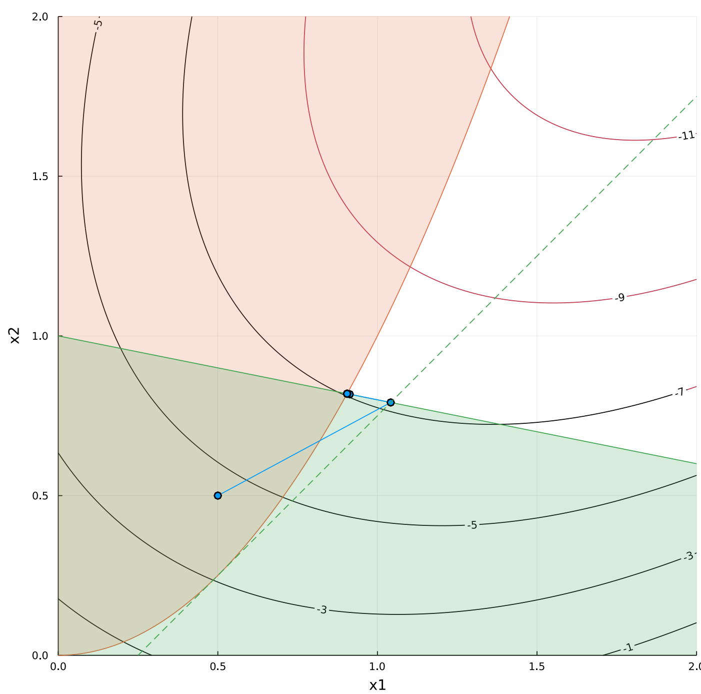

Figure 2: Feasible region and $l_1 - SQP$ iterations using $\mu = 10$ and $\Delta_k = 1$.