Homework 4 - Continuous Optimization

Estelle Baup, Samuel Bélisle, Cassandre Renaud

January 3, 2023

We want to solve the following problem:

$$\min_{x,y\in\mathbb{R}^n} f(x,y) \text{ subject to } h(x,y) = 0, \tag{P}$$

where f and h are specified on the homework sheet.

Question 1

The feasible set is $S = \{x, y \in \mathbb{R}^n \mid h(x,y) = 0\} = \{x, y \in \mathbb{R}^n \mid 1 - x^\top x = 0,\ 1 - y^\top y = 0,\ x^\top y = 0\}$.
It is not convex. Indeed, we will give two points z_1 and $z_2 \in S$ such that $z = \lambda z_1 + (1-\lambda) z_2 \notin S$ for a given $\lambda \in]0, 1[$. We will work with $n = 2$.

We will take $x_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ y_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and write $z_1 = (x_1, y_1)$. First we check that $z_1 \in S$.

- $1 - x_1^\top x_1 = 1 - \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1 - 1 = 0$
- $1 - y_1^\top y_1 = 1 - \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 1 - 1 = 0$
- $x_1 \top y_1 = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 0$

And we will take $z_2 = (x_2, y_2) = \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right)$. We also check that $z_2 \in S$.

- $1 - x_2^\top x_2 = 1 - \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 1 - 1 = 0$
- $1 - y_2^\top y_2 = 1 - \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1 - 1 = 0$
- $x_2^\top y_2 = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 0$

Lastly, we will take $\lambda = \frac{1}{2}$. We can compute

$$z = \lambda z_1 + (1-\lambda) z_2 = \frac{1}{2} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) + \frac{1}{2} \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) = \left(\begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}, \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \right) = (x, y)$$

But now, we have $x^\top y = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} = (\frac{1}{2})^2 + (\frac{1}{2})^2 = \frac{1}{2} \neq 0$. So the third condition of our function h does not hold at this point, hence $z \notin S$ and the set S is not convex.

By definition, LICQ holds at $x \in S$ if and only if $\nabla h_1(x), \dots \nabla h_p(x)$, and $\nabla g_i(x)$ for $i \in \mathcal{I}(x)$ are linearly independant.
Here, we do not have any constraint function g_i but we have three functions h_i:

- $h_1(x, y) = 1 - x^\top x$
- $h_2(x, y) = 1 - y^\top y$
- $h_3(x, y) = x^\top y$

If we compute their gradients, we get:

- $\nabla_x h_1(x, y) = \nabla_x \left(1 - x^\top x\right) = \nabla_x \left(1 - \sum_{i=1}^n x_i^2\right) = -2x$ and $\nabla_y h_1(x, y) = 0$

  then $\nabla h_1(x, y) = \begin{bmatrix} -2x \\ \vec{0} \end{bmatrix}$ where $\vec{0}$ is the vector $\vec{0} \in \mathbb{R}^n$.

- $\nabla_x h_2(x, y) = 0$ and $\nabla_y h_2(x, y) = \nabla_y \left(1 - y^\top y\right) = \nabla_y \left(1 - \sum_{i=1}^n y_i^2\right) = -2y$

  then $\nabla h_2(x, y) = \begin{bmatrix} \vec{0} \\ -2y \end{bmatrix}$

- $\nabla_x h_3(x, y) = \nabla_x \sum_{i=1}^n x_i \cdot y_i = y$ and $\nabla_y h_3(x, y) = \nabla_y \sum_{i=1}^n x_i \cdot y_i = x$

  then $\nabla h_3(x, y) = \begin{bmatrix} y \\ x \end{bmatrix}$

We show that these three gradients are linearly independent:

$$\lambda_1 \nabla h_1(x, y) + \lambda_2 \nabla h_2(x, y) + \lambda_3 \nabla h_3(x, y) = 0 \iff \begin{cases} -2\lambda_1 x + \lambda_3 y = 0 \\ -2\lambda_2 y + \lambda_3 x = 0 \end{cases}$$

We have that this is true without having all the $\lambda_i = 0$, if and only if $x = \lambda y$ (for some $\lambda \neq 0$ that can be deduced from the previous equations), or $y = 0$. But if $y = 0$, we have $h_2(x, y) = 1 \neq 0$ so our point is not feasible. Same with h_1 if $x = 0$. Lastly, if $x = \lambda y$ for $\lambda \neq 0, x, y \neq 0$, we get that $h_3(x, y) = x^\top y = \lambda y^\top y = \lambda ||y||^2 \neq 0$, so our point is not feasible either.
To conclude, we get that for all of our feasible points, i.e. the points in the set S, $\nabla h_1(x, y), \nabla h_2(x, y), \nabla h_3(x, y)$ are linearly independent, which means by definition that LICQ holds.

Question 2

Our two first constraints $h_1(x, y) = 1 - x^\top x = 0$ and $h_2(x, y) = 1 - y^\top y = 0$ can be rephrased $||x|| = ||y|| = 1$.
We notice that there is no constraint on x and y at the same time, i.e. a constraint that tells us what x must be related to y and vice-versa. Similarly, we can notice that our target function doesn't have a part where x and y are mixed.
It means that if we optimized x and y separately, the optimum found will be the optimum for our relaxed problem as well. So we rewrite our relaxed problem as:

$$\min_{\substack{x \in \mathbb{R}^n \\ ||x|| = 1}} \frac{1}{2} x^\top A x + \min_{\substack{y \in \mathbb{R}^n \\ ||y|| = 1}} \frac{1}{2} y^\top B y$$

The two problems are solved identically as the only thing changing is the matrix. To solve them, we first recall example 2.14 from the notes:
If A is a symmetric linear map with eigenvalues $\lambda_1, ..., \lambda_n$, then $\forall u \in \mathcal{E}$, we have

$$\lambda_{\min} ||u||^2 \leq \langle u, A(u) \rangle \leq \lambda_{\max} ||u||^2$$

and by rewriting the scalar product we get

$$\lambda_{\min}||u||^2 \leq u^\top A u \leq \lambda_{\max}||u||^2$$

Applied to our problems (let's look at the first as they are similar), knowing that we optimize on vector of norm 1, we get that for every feasible x, we have

$$\frac{1}{2}\lambda_{\min} \leq \frac{1}{2}x^\top A x \leq \frac{1}{2}\lambda_{\max}$$

It tells us that our optimal value can't be lower that $\frac{1}{2}\lambda_{\min}$. If we find an x that attains this bound, we will know that it is the optimal value. So let's find it.
As $\lambda_{\min}$ is an eigenvalue, it means that $\exists v_{\min} \in \mathbb{R}^n, v_{\min} \neq 0$ such that $Av_{\min} = \lambda_{\min} v_{\min}$. We take $x = \frac{v_{\min}}{||v_{\min}||}$, which is feasible since it has norm 1, and then we have

$$x^\top A x = \frac{1}{||v_{\min}||^2} v_{\min}^\top (A v_{\min}) = \frac{1}{||v_{\min}||^2} v_{\min}^\top (\lambda_{\min} v_{\min}) = \frac{\lambda_{\min}}{||v_{\min}||^2} v_{\min}^\top v_{\min} = \frac{\lambda_{\min}}{||v_{\min}||^2} \cdot ||v_{\min}||^2 = \lambda_{\min}$$

Therefore our lower bound is attained for this x, which means it is the optimal value.
We denote $\lambda_{\min}(A)$ and $\lambda_{\min}(B)$ the minimal eigenvalues for the matrices A and B. From our previous reasoning, we have that $\frac{\lambda_{\min}(A)}{2}$ and $\frac{\lambda_{\min}(B)}{2}$ are our optimal values, and so the optimal value for our relaxed problem is $\frac{\lambda_{\min}(A)+\lambda_{\min}(B)}{2}$.

Now, what does that say about our target problem?
Because our relaxed problem is our target problem minus a constraint, we know its optimal value must be lower than the optimal value of the target problem, as every feasible solution of our target problem is feasible in our relaxed problem. Therefore, we can say that our target problem has a lower bound of $\frac{\lambda_{\min}(A)+\lambda_{\min}(B)}{2}$. However, we don't know (yet) if it is the optimal value or not.

Question 3

Let's find an expression for the Lagrangian function $L(x, y, \mu)$. We denote I_n for the identity matrix in $\mathbb{R}^{n\times n}$.

$$\begin{aligned}
L(x, y, \mu) &= f(x, y) + \mu^\top h(x, y) \\
&= \frac{1}{2}x^\top A x + \frac{1}{2}y^\top B y + \begin{bmatrix}\mu_1 & \mu_2 & \mu_3\end{bmatrix} \begin{bmatrix} 1 - x^\top x \\ 1 - y^\top y \\ x^\top y \end{bmatrix} \\
&= \frac{1}{2}x^\top A x + \frac{1}{2}y^\top B y + \mu_1(1 - x^\top x) + \mu_2(1 - y^\top y) + \mu_3 x^\top y \\
&= \frac{1}{2}\left(x^\top A x - 2\mu_1 x^\top x + y^\top B y - 2\mu_2 y^\top y + 2\mu_3 x^\top y\right) + \mu_1 + \mu_2 \\
&= \frac{1}{2}\left(x^\top (A - 2\mu_1 I_n) x + y^\top (B - 2\mu_2 I_n) y + x^\top (\mu_3 I_n) y + y^\top (\mu_3 I_n) x\right) + \mu_1 + \mu_2 \\
&= \frac{1}{2}\begin{bmatrix}x^\top & y^\top\end{bmatrix} \begin{bmatrix} A - 2\mu_1 I_n & \mu_3 I_n \\ \mu_3 I_n & B - 2\mu_2 I_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \mu_1 + \mu_2
\end{aligned}$$

As A and B are symmetric by assumptions, and I_n is also symmetric, we see that the above matrix is symmetric.

Question 4

By definition, $L_D(\mu) = \inf_{x,y\in\mathbb{R}^n} L(x,y,\mu)$. Using the previous question, and denoting $M_\mu := \begin{bmatrix} A - 2\mu_1 I_n & \mu_3 I_n \\ \mu_3 I_n & B - 2\mu_2 I_n \end{bmatrix}$, we find:

$$\begin{aligned}
L_D(\mu) &= \inf_{x,y\in\mathbb{R}^n} L(x,y,\mu) \\
&= \inf_{x,y\in\mathbb{R}^n} \left\{ \frac{1}{2} \begin{bmatrix} x^\top & y^\top \end{bmatrix} M_\mu \begin{bmatrix} x \\ y \end{bmatrix} + \mu_1 + \mu_2 \right\} \\
&= \inf_{x,y\in\mathbb{R}^n} \left\{ \frac{1}{2} \begin{bmatrix} x^\top & y^\top \end{bmatrix} M_\mu \begin{bmatrix} x \\ y \end{bmatrix} \right\} + \mu_1 + \mu_2 \\
&= \begin{cases} \mu_1 + \mu_2 & \text{if } M_\mu \succeq 0 \\ -\infty & \text{otherwise} \end{cases}
\end{aligned}$$

Indeed, if M_μ is not positive semidefinite, there exists a negative eigenvalue $\lambda < 0$. We can consider its associated eigenvector $v = (v_x, v_y) \in \mathbb{R}^{2n}$ (with $v_x, v_y \in \mathbb{R}^n$; by simplicity we take v such that $||v|| = 1$), i.e. we have $M_\mu v = \lambda v$. Then for all $n \in \mathbb{N}$, we have $(nv)^\top M_\mu (nv) = n^2 v^\top \lambda v = n^2 \lambda$, so by looking at the sequence $(nv)_{n\in\mathbb{N}}$, we have that $L(nv_x, nv_y, \mu)$ converges to $-\infty$.
Hence we can write the dual as:

$$\max_{\mu\in\mathbb{R}^3} \{\mu_1 + \mu_2\} \quad \text{subject to } \begin{bmatrix} A - 2\mu_1 I_n & \mu_3 I_n \\ \mu_3 I_n & B - 2\mu_2 I_n \end{bmatrix} \succeq 0 \tag{D}$$

Question 5

We know that M_μ is symmetric for all $\mu \in \mathbb{R}^3$. If μ is a solution of the dual problem, then the matrix M_μ associated with the values of μ should be positive semidefinite. This implies in particular that the diagonal blocks would be positive semidefinite too, i.e. $A - 2\mu_1 I_n \succeq 0$ and $B - 2\mu_2 I_n \succeq 0$. The two latter conditions are equivalent to $2\mu_1 \leq \lambda_{\min}(A)$ and $2\mu_2 \leq \lambda_{\min}(B)$, where $\lambda_{\min}(A)$ and $\lambda_{\min}(B)$ are the smallest eigenvalues of A and B respectively.
Hence, if we focus only on satisfying the conditions $A - 2\mu_1 I_n \succeq 0$ and $B - 2\mu_2 I_n \succeq 0$, the optimal value for (D) is $\frac{\lambda_{\min}(A)+\lambda_{\min}(B)}{2}$.
In fact, this is the optimal value even for the whole condition $M_\mu \succeq 0$. Indeed, let's focus on the relaxed problem for a bit. Call P' the relaxed primal, and D' the relaxed dual. By question 2, we know that $\frac{\lambda_{\min}(A)+\lambda_{\min}(B)}{2}$ is the lower bound on the optimal value for P'. Now observe that the relaxed dual function is in fact

$$L_{D'}(\mu) = \inf_{x,y\in\mathbb{R}^n} \left\{ \frac{1}{2} \begin{bmatrix} x^\top & y^\top \end{bmatrix} \begin{bmatrix} A - 2\mu_1 I_n & 0 \\ 0 & B - 2\mu_2 I_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \right\} + \mu_1 + \mu_2,$$

This comes directly from the fact that if we remove the third constraint $h_3(x,y) = 0$, then we no longer have any μ_3.

By weak duality for the relaxed problem, and denoting $S' = \{(x,y) \mid h_1(x,y) = 0, h_2(x,y) = 0\}$ the feasible set of P', we get:

$$\max_{\mu\in\mathbb{R}^3} L_{D'}(\mu) \leq \min_{(x,y)\in\mathbb{R}^n\times\mathbb{R}^n} L_{P'}(x,y) = \min_{(x,y)\in S'} f(x,y) = \frac{\lambda_{\min}(A) + \lambda_{\min}(B)}{2}$$

Moreover, we know $\max_{\mu\in\mathbb{R}^3} L_D(\mu) \leq \max_{\mu\in\mathbb{R}^3} L_{D'}(\mu)$ as if we go from D' to D we add a constraint, and the maximum can only go lower. Then putting this together with the previous equality gives:

$$\max_{\mu\in\mathbb{R}^3} L_D(\mu) \leq \frac{\lambda_{\min}(A) + \lambda_{\min}(B)}{2}$$

Finally, we see that taking $\mu^* = \left(\frac{\lambda_{\min}(A)}{2}, \frac{\lambda_{\min}(B)}{2}, 0\right)$ in the dual D gives $M_{\mu^*} \succeq 0$ as both diagonal blocks will be positive semidefinite (and the anti-diagonal blocks are zero), and $L_D(\mu^*) = \frac{\lambda_{\min}(A) + \lambda_{\min}(B)}{2}$. So the upper bound is attained and is indeed the optimal value of the dual problem.

Question 6

To use the strong duality theorem, we need to satisfy the two following assumptions:

(A1) the primal problem (P) admits a KKT point $(x^*, y^*) \in S$ with valid Lagrange multipliers $\mu^* \in \mathbb{R}^3$

(A2) the function $(x, y) \mapsto L(x, y, \mu^*)$ is convex

First, let's prove that (A1) holds.

We will use the theorem 8.26 of the lecture notes, as we know by question 1 that LICQ holds at all feasible points. It remains to show that there exists a local minimum of f in S.

Observe that $S = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n \mid h(x, y) = 0\}$ is a closed set as a preimage of the closed set $\{0\} \subset \mathbb{R}$ by the continuous function h. Moreover, we have $S \subset \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n \mid h_1(x, y) = 0, h_2(x, y) = 0\} = \partial B_1(0) \times \partial B_1(0)$, where $B_1(0) = \{z \in \mathbb{R}^n \mid ||z|| < 1\}$ is the unit ball in $\mathbb{R}^n$. Hence S is bounded. So we deduce that it is compact (as it is closed and bounded in $\mathbb{R}^n \times \mathbb{R}^n = \mathbb{R}^{2n}$).

Now f is continuous on the compact set S, thus it must attains its (global) minimum in it, say at a point (x^*, y^*). The theorem 8.26 implies that (x^*, y^*) is a KKT point (as LICQ holds in particular at this point). Let $\mu^* \in \mathbb{R}^3$ be valid Lagrange multipliers for (x^*, y^*).

Consider now the condition (A2). It is equivalent to $\begin{bmatrix} A - 2\mu_1^* I_n & \mu_3^* I_n \\ \mu_3^* I_n & B - 2\mu_2^* I_n \end{bmatrix}$ being positive semi-definite. Now for this latter condition to be satisfied, we would need information about μ^*. But we can't expect to have those informations before solving the problem, as we need to solve it to find (x^*, y^*).

Question 7

Here, we will calculate the gradient of the augmented Lagrangian function. In the first section, we present un-requested details, and then we present the results.

Details

We will first compute the gradient with respect to x. ∇_y will be done similarly.
First of all, we know, by linearity, that

$$\nabla_x L_\beta(x, y, \mu) = \nabla_x f(x, y) + \nabla_x \mu^\top h(x, y) + \frac{\beta}{2} \nabla_x ||h(x)||^2$$

We will compute each of these expressions:

We know that $\nabla_x f(x, y) = Ax$

Now for the second one, we can develop $\mu^\top h(x, y) = \mu_1(1 - x\top x) + \mu_2(1 - y^\top y) + \mu_3 x^\top y$, so we have:

$$\begin{aligned} \nabla_x \mu^\top h(x, y) &= \nabla_x(\mu_1(1 - x\top x)) + \nabla_x(\mu_2(1 - y^\top y)) + \nabla_x(\mu_3 x^\top y) \\ &= \mu_1 \nabla_x(1 - x\top x) + \mu_3 \nabla_x(x^\top y) \\ &= \mu_1(-2x) + \mu_3 y \\ &= -2\mu_1 x + \mu_3 y \end{aligned}$$

As $\frac{\partial}{\partial x_i}(x^\top x) = \frac{\partial}{\partial x_i}\sum_{i=1}^n x_i^2 = 2x_i$ and $\frac{\partial}{\partial x_i}(x^\top y) = \frac{\partial}{\partial x_i}\sum_{i=1}^n x_i y_i = y_i$

Now for the third part we will use the fact that $\nabla ||x||^2 = 2x$ as $x^\top x = ||x||^2$ and we've just computed it before. Plus we know the chain rule: $\frac{\partial}{\partial x_i} f \circ g(x) = \nabla f(g(x)) \cdot \frac{\partial}{\partial x_i} g(x)$. We can use it to compute our ∇ :

$$\frac{\partial}{\partial x_i}\frac{\beta}{2}||h(x,y)||^2 = \beta h(x,y) \cdot \frac{\partial}{\partial x_i}(h(x,y))$$

which gives us the formula

$$\begin{aligned}
\nabla_x \frac{\beta}{2}||h(x,y)||^2 &= \beta \nabla_x h(x,y) \cdot h(x,y) \text{ where } \nabla_x h(x,y) \text{ is a n} \times 3 \text{ matrix} \\
&= \beta \begin{bmatrix} -2x & \vec{0} & y \end{bmatrix} \cdot \begin{bmatrix} 1 - x^\top x \\ 1 - y^\top y \\ x^\top y \end{bmatrix} \\
&= \beta \begin{bmatrix} 2(x^\top x - 1)x_1 + (x^\top y)y_1 \\ \vdots \\ 2(x^\top x - 1)x_n + (x^\top y)y_n \end{bmatrix} \\
&= \beta \left(2(x^\top x - 1)x + (x^\top y)y\right)
\end{aligned}$$

Results

Putting together our three calculations, we get

$$\begin{aligned}
\nabla_x L_\beta(x,y,\mu) &= Ax - 2\mu_1 x + \mu_3 y + \beta\left(2(x^\top x - 1)x + (x^\top y)y\right) \\
&= Ax + 2(\beta(x^\top x - 1) - \mu_1)x + (\beta \cdot x^\top y + \mu_3)y
\end{aligned}$$

Similarly, we get

$$\begin{aligned}
\nabla_y L_\beta(x,y,\mu) &= By - 2\mu_2 y + \mu_3 x + \beta\left(2(y^\top y - 1)y + (x^\top y)x\right) \\
&= (\beta \cdot x^\top y + \mu_3)x + By + 2(\beta(y^\top y - 1) - \mu_2)y
\end{aligned}$$

Question 8

We wrote code that takes as input $z = [x^\top, y^\top]^\top$, μ, β (and also A and B) and returns $L_\beta(z,\mu)$ and $\nabla_z L_\beta(z,\mu)$. Our function is named `LBetaValAndGrad` as it calls the two sub-functions we are showing here: `LBeta` and `LBetaGrad`

```
function val = LBeta(z, mu, beta, A, B)
    hz = h(z);
    val = f(z, A, B) + mu' * hz;
    if beta ≠ 0 % Just to avoid calculating the norm if not necessary...
        val = val + beta * sum(hz .* hz)/2;
        % = val + beta * vecnorm(hz)^2 /2;
    end
end
```

```
function grad = LBetaGrad(z, mu, beta, A, B)
    n = length(z)/2;
    x = z(1:n);
```

```
    y = z(n+1:end);

    C1 = 2 * (beta * (x'*x - 1) - mu(1));
    C2 = 2 * (beta * (y'*y - 1) - mu(2));
    C3 = beta * x' * y + mu(3);

    grad = [A*x + C1*x + C3*y; C3*x + B*y + C2*y];
end
```

The first function uses the functions `f` and `h` which are a direct implementation of their definition.

Question 9

We have used the Matlab `fminunc` function. We set it to use the 'Quasi-Newton' sub-algorithm as it is the default parameter, it performs as well as Trust-Region in our situation, and it is a little quicker. We have created a function `minXY` which calls the precedent function with our choices for the parameters.
Here is our implementation.

```
function [z, zval] = minXY(mu, beta, A, B, z0, verbose)
% Our function that runs fminunc with desired parameters,
    % to minimize LBeta with fixed mu and beta.
% Input:
    % mu            : Fixed vector of size (3, 1)
    % beta          : Fixed cost factor.
    % A and B       : The defined matrices, size (n, n)
    % z0            : Initial point.
    % verbose       : Boolean indicating what the algorithm should print.
% Output:
    %   z           : The final z found with the given algorithm.
    %   zval        : It's value.
    options = optimoptions('fminunc');
    options.SpecifyObjectiveGradient = true;    % indicate gradient is provided
    %options.Algorithm = 'trust-region';        % We could also use TR...
    if verbose % add display
        options.Display = 'iter-detailed';
    else
        options.Display = 'none';
    end
    [z, zval] = fminunc(@(z) LBetaValAndGrad(z, mu, beta, A, B), z0, options);
end
```

In the 19-th line of the `main.m`, we defined a function handle which allows us to invoke `minXY` easily, and which does not output anything in the console:
`silentMinXY = @(mu, beta, z0)minXY(mu, beta, A, B, z0, 0);`
This function with $\beta = 1.42$, $\mu = [1, 2, -3]^\top$ and initial guess $z_0 = [1, 0, -1, 2, 1, 1, 2, 0, 1, 2]^\top$ gives after 50 iterations

```
Found x and y are
   -1.3695   -1.2693
   -0.2535    0.9636
   -0.0107   -1.0786
    0.0825    0.7410
    1.4721    0.5569

with value f(x, y) = -28.9565    and LBeta(x, y, mu) = -26.4171
and h(x, y), mu:       which have norms 5.2991, 3.7417
   -3.1138    1.0000
   -3.5622    2.0000
    2.3865   -3.0000
```

One can see the full listing in the Appendix.

Note that the results would have been little bit different if we were using 'trust-region' as a sub-algorithm for the function `fminunc`.

Question 10

The part of the `main` running the quadratic penalty method is the following:

```
mu = [0; 0; 0];
z0 = randn(2*n, 1);
z = z0;
beta = 1;
for i = 1:N
    [z, zval] = silentMinXY(mu, beta, z);

    % Display
    disp(newline + "Iteration " + i + " with beta = " + beta + ". We found x and y:")
    disp([z(1:n), z(n+1:end)])
    disp("with value f(x, y) = " + f(z) + "    and LBeta(x, y, mu) = " + LBeta(z, mu, beta, ...
        A, B))
    hz = h(z);
    disp("and h(x, y), beta*h(x, y):       which have norms " + vecnorm(hz) + ", " + beta * ...
        vecnorm(hz))
    disp([hz, beta * hz])

    beta = 2 * beta; % Updating
end
```

We listed the complete output in the Appendix, but here is what the final iteration leaves:

```
Iteration 9 with beta = 256. We found x and y:
    0.6036   -0.4636
    0.2996    0.3649
   -0.3370   -0.6272
    0.1038    0.4992
   -0.6591    0.1458

with value f(x, y) = -6.41    and LBeta(x, y, mu) = -6.3694
and h(x, y), beta*h(x, y):       which have norms 0.01781, 4.5593
   -0.0128   -3.2785
   -0.0119   -3.0504
   -0.0033   -0.8566
```

We notice that the norm of $h(x, y)$ is divided by two at each iterations, since $\beta h(x, y)$ stays almost constant. This reassures us that we could hope, as we have seen in the lecture notes, that $h(x_k) \approx \frac{\mu^*}{\beta_k}$. However, we will indeed have a convergence of this sort, as we will see in question 11, but not really to μ^*, as we will see in question 12.

We see that after 9 iterations, the points x and y are still *far* from $h(x, y) = 0$. That's why we consider the augmented Lagrangian method.

Question 11

The part of the `main` running the augmented Lagrangian method is the following:

```
mu = [0; 0; 0];
% z0 = randn(2*n, 1);    % We start with the same z0 as for QPM.
```

```
3  z = z0;
4  beta = 1;
5  for i = 1:N
6      [z, zval] = silentMinXY(mu, beta, z);
7
8      hz = h(z);
9      mu = mu + beta * hz; % Updating
10
11     % Display
12     disp(newline + "Iteration " + i + " with beta = " + beta + ". We found x and y:")
13     disp([z(1:n), z(n+1:end)])
14     disp("with value f(x, y) = " + f(z) + "    and LBeta(x, y, mu) = " + LBeta(z, mu, beta, ...
           A, B))
15     disp("and h(x, y), new mu:       which have norms " + vecnorm(hz) + ", " + vecnorm(mu))
16     disp([hz, mu])
17
18     beta = 2 * beta; % Updating
19 end
```

Note that the only difference is the update of μ at line 9. Also, we decided not to choose a new z_0 randomly, but to reuse the same starting point as with the QPM of question 10. The final iteration leaves:

```
Iteration 9 with beta = 256. We found x and y:
    0.5988   -0.4605
    0.2988    0.3626
   -0.3372   -0.6237
    0.1041    0.4965
   -0.6539    0.1447

with value f(x, y) = -6.3288    and LBeta(x, y, mu) = -6.3288
and h(x, y), new mu:       which have norms 3.2811e-08, 4.5595
   -0.0000   -3.2774
    0.0000   -3.0515
    0.0000   -0.8582
```

Again, we listed the complete outputs in the Appendix.

This time, $h(x, y)$ goes to 0 really quickly. Moreover, we see that our estimation of μ quickly stabilizes to a fixed value which is indeed the one $\beta h(x, y)$ was converging to with the quadratic penalty method.

Note that depending on the starting points, the sign of the values may differ. But with all our tests, the numbers were more or less the same.

Question 12

Our program finds $\begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} = \begin{pmatrix} -3.2774 \\ -3.0515 \\ \pm 0.8582 \end{pmatrix}$. The sign of the last value varies, but this is not important for the rest of our calculations.

We know from question 3 that the Lagrangian is

$$L(x, y, \mu) = \frac{1}{2} \begin{bmatrix} x^\top & y^\top \end{bmatrix} \begin{bmatrix} A - 2\mu_1 I_n & \mu_3 I_n \\ \mu_3 I_n & B - 2\mu_2 I_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \mu_1 + \mu_2.$$

We know that for a fixed μ, this function is convex if only if the matrix—that we named M_μ in question 4—is positive semi-definite. In our main, the following lines are calculating the minimum eigenvalue of M_μ and may output a warning in the console.

```
    disp("And therefore, M_mu is not semi-positive definite,")
    disp("and mu is not feasible in (D).")
```

```
else
    disp("≤> M_mu is semi-positive definite.")
end

disp(newline + "Note that the theoretical maximal value for (D) is:")
```

As this warning is triggered by our value of μ, we conclude that $L(x, y, \mu)$ is not convex for this μ, and also that μ is not feasible in (D). That is bad news, since our hope was that μ^k goes to μ^* as our algorithm progresses; but instead we have μ, which is the limit[1] of μ^k, that is not feasible in (D) and therefore not μ^*. Note that if μ was feasible, then $\mu_1 + \mu_2 = -6.3288$ would be a lower bound for the value $f(x, y)$ from the weak duality theorem, and we could conclude that our found (x, y) (which reached this value) is a global minimum. Note that the optimal value for (D) is bounded by $\frac{\lambda_{\min}(A)+\lambda_{\min}(B)}{2} = -6.7355$, therefore there is no μ that will show that our found (x, y) is a global minimum.

Now we consider the strong duality theorem. We have seen in question 6 that the first assumption holds. However, we can't know if the second one holds before actually solving the problem. We also know that if the theorem does not hold, then there will be a gap between the optimal values for (P) and (D). With our found (x, y), we see that there is a gap with the optimal value of (D), but we can't know if we truly found an optimal point for (P), or if there is a better solution—hiding somewhere and really difficult to find—with no gap and a feasible μ^*, which would make the Lagrangian convex. Thus, we can't know that the theorem holds, nor be sure that it does not hold. However, we are pretty close of a no-gap solution, and the fact that every execution of our algorithm leaves to the same (four, considering axial symmetries) solutions is reassuring in thinking that no better solution exists.

[1] μ should be the limit of our calculated sequence μ^k. In practice, we stop at μ^9. Two things motivate this choice. First, the value of μ^k rapidly converged, and, as the associated value $h(x^k)$ is really close to zero at that point, the value of μ^k should stay unchanged. But secondly, as β increases if we do too much iterations, small calculation residues in $h(x^k)$ will be drastically amplified and will change μ^k to reach extreme values.

Appendix

Complete listing of the call of fminunc in question 9

```
===========================================
Start of one fminunc with fixed mu and beta
                                                        First-order
 Iteration  Func-count       f(x)        Step-size       optimality
     0           1          178.82                             82.2
     1           2         10.1162       0.0121625             17.7
     2           3        -4.20962               1             12.7
     3           4        -15.0808               1             5.64
     4           5        -17.2193               1             6.58
     5           6        -19.7316               1             6.67
     6           7        -21.3684               1             5.65
     7           8        -21.9965               1             2.56
     8           9        -22.4082               1             2.17
     9          10        -22.9733               1             1.42
    10          11        -23.0828               1            0.947
    11          12        -23.1327               1            0.782
    12          13        -23.1999               1            0.884
    13          14        -23.2755               1             1.05
    14          15        -23.3273               1            0.745
    15          16        -23.3533               1            0.522
    16          17        -23.3677               1            0.486
    17          18        -23.3878               1            0.485
    18          19        -23.4311               1             0.81
    19          20        -23.5459               1              1.7
                                                        First-order
 Iteration  Func-count       f(x)        Step-size       optimality
    20          22        -23.6747        0.402734             3.25
    21          23        -23.9004               1             3.18
    22          24         -24.136               1              2.7
    23          26        -24.3296             0.5             1.35
    24          27        -24.4815               1            0.818
    25          28        -24.5246               1            0.755
    26          29        -24.6162               1            0.793
    27          30        -24.7419               1             1.06
    28          31         -24.925               1             1.33
    29          32        -25.0605               1             2.47
    30          33        -25.2445               1              2.3
    31          34         -25.481               1             3.21
    32          35        -25.7183               1             1.71
    33          36        -25.9075               1              1.9
    34          37        -26.1991               1             1.26
    35          38         -26.341               1            0.724
    36          39        -26.3684               1            0.644
    37          40        -26.3901               1            0.254
    38          41        -26.3943               1            0.249
    39          42        -26.4027               1            0.219
                                                        First-order
 Iteration  Func-count       f(x)        Step-size       optimality
    40          43        -26.4078               1            0.229
    41          44         -26.411               1            0.185
    42          45        -26.4127               1            0.123
    43          46        -26.4145               1            0.119
    44          47        -26.4161               1            0.106
    45          48        -26.4169               1           0.0649
    46          49        -26.4171               1           0.0159
    47          50        -26.4171               1          0.00199
    48          51        -26.4171               1         0.000491
    49          52        -26.4171               1         0.000166
    50          53        -26.4171               1         2.59e-05
```

```

Optimization completed: The first-order optimality measure, 3.112651e-07, is less
than options.OptimalityTolerance = 1.000000e-06.

Found x and y are
   -1.3695   -1.2693
   -0.2535    0.9636
   -0.0107   -1.0786
    0.0825    0.7410
    1.4721    0.5569

with value f(x, y) = -28.9565    and LBeta(x, y, mu) = -26.4171
and h(x, y), mu:       which have norms 5.2991, 3.7417
   -3.1138    1.0000
   -3.5622    2.0000
    2.3865   -3.0000
```

Complete listing of the QPM of question 10

```
=====================================
Start of the Quadratic penalty method

Iteration 1 with beta = 1. We found x and y:
    1.3349   -0.9717
    0.4912    0.7564
   -0.4391   -1.2122
    0.1089    0.9557
   -1.4509    0.3235

with value f(x, y) = -27.0433    and LBeta(x, y, mu) = -16.69
and h(x, y), beta*h(x, y):       which have norms 4.5504, 4.5504
   -3.3330   -3.3330
   -3.0037   -3.0037
   -0.7585   -0.7585


Iteration 2 with beta = 2. We found x and y:
    1.0365   -0.7603
    0.4026    0.5928
   -0.3792   -0.9649
    0.1000    0.7630
   -1.1276    0.2499

with value f(x, y) = -16.6912    and LBeta(x, y, mu) = -11.5126
and h(x, y), beta*h(x, y):       which have norms 2.2757, 4.5513
   -1.6619   -3.3239
   -1.5051   -3.0102
   -0.3890   -0.7779


Iteration 3 with beta = 4. We found x and y:
    0.8481   -0.6284
    0.3517    0.4911
   -0.3503   -0.8132
    0.0978    0.6450
   -0.9235    0.2035

with value f(x, y) = -11.5137    and LBeta(x, y, mu) = -8.9226
and h(x, y), beta*h(x, y):       which have norms 1.1382, 4.5528
   -0.8281   -3.3124
   -0.7548   -3.0192
   -0.2002   -0.8006
```

```
Iteration 4 with beta = 8. We found x and y:
    0.7350   -0.5507
    0.3248    0.4314
   -0.3393   -0.7252
    0.0989    0.5764
   -0.8012    0.1761

with value f(x, y) = -8.9232    and LBeta(x, y, mu) = -7.6266
and h(x, y), beta*h(x, y):       which have norms 0.56935, 4.5548
   -0.4126   -3.3009
   -0.3786   -3.0291
   -0.1027   -0.8216


Iteration 5 with beta = 16. We found x and y:
    0.6708   -0.5075
    0.3114    0.3984
   -0.3364   -0.6767
    0.1007    0.5383
   -0.7318    0.1610

with value f(x, y) = -7.6269    and LBeta(x, y, mu) = -6.9781
and h(x, y), beta*h(x, y):       which have norms 0.28479, 4.5566
   -0.2057   -3.2916
   -0.1899   -3.0376
   -0.0523   -0.8371


Iteration 6 with beta = 32. We found x and y:
    0.6360   -0.4845
    0.3050    0.3809
   -0.3362   -0.6508
    0.1022    0.5180
   -0.6941    0.1530

with value f(x, y) = -6.9781    and LBeta(x, y, mu) = -6.6536
and h(x, y), beta*h(x, y):       which have norms 0.14243, 4.5579
   -0.1027   -3.2853
   -0.0951   -3.0436
   -0.0265   -0.8468


Iteration 7 with beta = 64. We found x and y:
    0.6177   -0.4727
    0.3019    0.3718
   -0.3365   -0.6374
    0.1031    0.5074
   -0.6743    0.1489

with value f(x, y) = -6.6536    and LBeta(x, y, mu) = -6.4912
and h(x, y), beta*h(x, y):       which have norms 0.071229, 4.5586
   -0.0513   -3.2816
   -0.0476   -3.0473
   -0.0133   -0.8523


Iteration 8 with beta = 128. We found x and y:
    0.6083   -0.4666
    0.3003    0.3673
   -0.3368   -0.6306
    0.1036    0.5020
   -0.6642    0.1468

```

```
with value f(x, y) = -6.4912    and LBeta(x, y, mu) = -6.41
and h(x, y), beta*h(x, y):       which have norms 0.035618, 4.5591
   -0.0256   -3.2795
   -0.0238   -3.0493
   -0.0067   -0.8552


Iteration 9 with beta = 256. We found x and y:
    0.6036   -0.4636
    0.2996    0.3649
   -0.3370   -0.6272
    0.1038    0.4992
   -0.6591    0.1458

with value f(x, y) = -6.41    and LBeta(x, y, mu) = -6.3694
and h(x, y), beta*h(x, y):       which have norms 0.01781, 4.5593
   -0.0128   -3.2785
   -0.0119   -3.0504
   -0.0033   -0.8566
```

Complete listing of the ALM of question 11

```
========================================
Start of the augmented Lagrangian method

Iteration 1 with beta = 1. We found x and y:
    1.3349   -0.9717
    0.4912    0.7564
   -0.4391   -1.2122
    0.1089    0.9557
   -1.4509    0.3235

with value f(x, y) = -27.0433    and LBeta(x, y, mu) = 4.0166
and h(x, y), new mu:       which have norms 4.5504, 4.5504
   -3.3330   -3.3330
   -3.0037   -3.0037
   -0.7585   -0.7585


Iteration 2 with beta = 2. We found x and y:
    0.5927   -0.4849
    0.2931    0.3805
   -0.3230   -0.6199
    0.0962    0.4859
   -0.6474    0.1617

with value f(x, y) = -6.339    and LBeta(x, y, mu) = -6.3265
and h(x, y), new mu:       which have norms 0.052296, 4.5535
    0.0302   -3.2727
   -0.0264   -3.0566
   -0.0336   -0.8256


Iteration 3 with beta = 4. We found x and y:
    0.6017   -0.4603
    0.2962    0.3623
   -0.3318   -0.6228
    0.1020    0.4960
   -0.6569    0.1446

with value f(x, y) = -6.3353    and LBeta(x, y, mu) = -6.3287
and h(x, y), new mu:       which have norms 0.0079021, 4.5591
```

```
   -0.0019   -3.2803
    0.0020   -3.0485
   -0.0074   -0.8552


Iteration 4 with beta = 8. We found x and y:
    0.5987   -0.4609
    0.2988    0.3629
   -0.3371   -0.6237
    0.1041    0.4963
   -0.6538    0.1449

with value f(x, y) = -6.329    and LBeta(x, y, mu) = -6.3288
and h(x, y), new mu:       which have norms 0.00063213, 4.5594
    0.0004   -3.2773
   -0.0004   -3.0516
   -0.0003   -0.8578


Iteration 5 with beta = 16. We found x and y:
    0.5988   -0.4605
    0.2988    0.3626
   -0.3372   -0.6237
    0.1041    0.4965
   -0.6539    0.1447

with value f(x, y) = -6.3289    and LBeta(x, y, mu) = -6.3288
and h(x, y), new mu:       which have norms 2.5921e-05, 4.5595
   -0.0000   -3.2774
    0.0000   -3.0515
   -0.0000   -0.8582


Iteration 6 with beta = 32. We found x and y:
    0.5988   -0.4605
    0.2988    0.3626
   -0.3372   -0.6237
    0.1041    0.4965
   -0.6539    0.1447

with value f(x, y) = -6.3288    and LBeta(x, y, mu) = -6.3288
and h(x, y), new mu:       which have norms 3.6134e-07, 4.5595
    0.0000   -3.2774
   -0.0000   -3.0515
   -0.0000   -0.8582


Iteration 7 with beta = 64. We found x and y:
    0.5988   -0.4605
    0.2988    0.3626
   -0.3372   -0.6237
    0.1041    0.4965
   -0.6539    0.1447

with value f(x, y) = -6.3288    and LBeta(x, y, mu) = -6.3288
and h(x, y), new mu:       which have norms 8.9348e-08, 4.5595
   -0.0000   -3.2774
    0.0000   -3.0515
   -0.0000   -0.8582


Iteration 8 with beta = 128. We found x and y:
    0.5988   -0.4605
    0.2988    0.3626
   -0.3372   -0.6237
```

```
    0.1041    0.4965
   -0.6539    0.1447

with value f(x, y) = -6.3288    and LBeta(x, y, mu) = -6.3288
and h(x, y), new mu:       which have norms 7.7183e-08, 4.5595
    0.0000   -3.2774
   -0.0000   -3.0515
    0.0000   -0.8582


Iteration 9 with beta = 256. We found x and y:
    0.5988   -0.4605
    0.2988    0.3626
   -0.3372   -0.6237
    0.1041    0.4965
   -0.6539    0.1447

with value f(x, y) = -6.3288    and LBeta(x, y, mu) = -6.3288
and h(x, y), new mu:       which have norms 3.2811e-08, 4.5595
   -0.0000   -3.2774
    0.0000   -3.0515
    0.0000   -0.8582
```

The informations and comments we made our program output, according to the final results of ALM.

```
========================================
Final result's interpretation
Our (hopefully feasible) obtained value for the Dual problem (D) is mu_1 + mu_2 = -6.3288
But 2 * mu(1) > lambda_min(A): -6.5548 > -7.1291
But 2 * mu(2) > lambda_min(B): -6.1029 > -6.3419
hence, lambda_min(M_mu) = -1.121.
And therefore, M_mu is not semi-positive definite,
and mu is not feasible in (D).

Note that the theoretical maximal value for (D) is:
(lambda_min(A) + lambda_min(B))/2 = -6.7355
```