Homework 4 - Continuous Optimization

~~Estelle~~ Ezra Baup, Samuel Bélisle, Cassandre Renaud

December 29, 2022

> Faudra pas oublier de mettre nos noms "comme il faut". J'ai juste voulu éviter de renforcer ma tite tête avec ton nom *officiel* à chaque fois que je lis cette ligne.
>
> -Sam

holà les gus
ce devoir, Sam'enchante pas, vous?
(un peu cass-é comme blague je l'avoue, mais gardez ez-poir en moi please)

We want to solve the following problem:

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

where f and h are specified on the homework sheet.

Question 1

NOTE/TODO je suis pas du tout sure que cette notation est très belle ducoup dites si vous voulez changer. Aussi c'est très moche les produit scalaires "horizontaux" mais plus lisible? jsp

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

We will take $z_1 = (x_1, y_1) = ((1,0),(0,1))$. First we check that $z_1 \in S$.

- $1 - x_1^\top x_1 = 1 - \left\langle \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix} \right\rangle = 1 - 1 = 0$
- $1 - y_1^\top y_1 = 1 - \begin{pmatrix}0 & 1\end{pmatrix} \begin{pmatrix}0\\1\end{pmatrix} = 1 - 1 = 0$
- $x_1 \top y_1 = \langle (1,0),(0,1) \rangle = 0$

And we will take $z_2 = (x_2, y_2) = ((0,1),(1,0))$. we also check that $z_2 \in S$.

- $1 - x_2^\top x_2 = 1 - \langle (0,1),(0,1) \rangle = 1 - 1 = 0$
- $1 - y_2^\top y_2 = 1 - \langle (1,0),(1,0) \rangle = 1 - 1 = 0$
- $x_2 \top y_2 = \langle (0,1),(1,0) \rangle = 0$

Lastly, we will take $\lambda = \frac{1}{2}$. Now we can compute our $z = \lambda z_1 + (1-\lambda)z_2$

$$z = \lambda z_1 + (1-\lambda)z_2 = \frac{1}{2}((1,0),(0,1)) + \frac{1}{2}((0,1),(1,0)) = ((\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})) = (x,y)$$

But if we compute $x^\top y = \left\langle (\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2}) \right\rangle = (\frac{1}{2})^2 + (\frac{1}{2})^2 = \frac{1}{2} \neq 0$, so the third condition of our function h does not hold on this point, hence our set 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:
TODO : revoir gradient : les calculs sont justes et après la matrice est fausse, jsp si ça change les résultats pour le lin indep

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

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

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

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

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

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

They are linearly independent:
$\lambda_1 \nabla h_1(x,y) + \lambda_2 \nabla h_2(x,y) + \lambda_3 \nabla h_3(x,y) = 0 \iff -2\lambda_1 x + \lambda_3 y = 0$ and $\lambda_3 x - 2\lambda_2 y = 0$. We have that this is true without having all the $\lambda_i = 0$, if and only if $x = \lambda y$ (for some lambda 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 $\lambda, x, y \neq 0$, we get that $h_3(x,y) = x^\top y = \lambda y^\top y = \lambda ||y|| \neq 0$, 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 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:

$$\min_{x\in\mathbb{R}^n:||x||=1} \frac{1}{2}x^\top Ax + \min_{y\in\mathbb{R}^n:||y||=1} \frac{1}{2}y^\top By$$

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

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

and by rewriting the scalar product we get

$$\lambda_{\text{min}}||u||^2 \le u^\top Au \le \lambda_{\text{max}}||u||^2$$

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

$$\frac{1}{2}\lambda_{\text{min}} \le \frac{1}{2}x^\top Ax \le \frac{1}{2}\lambda_{\text{max}}$$

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

$$x^\top Ax = \frac{1}{||v_{\text{min}}||^2}v_{\text{min}}^\top\left(Av_{\text{min}}\right) = \frac{1}{||v_{\text{min}}||^2}v_{\text{min}}^\top(\lambda_{\text{min}}v_{\text{min}}) = \frac{\lambda_{\text{min}}}{||v_{\text{min}}||^2}v_{\text{min}}^\top v_{\text{min}} = \frac{\lambda_{\text{min}}}{||v_{\text{min}}||^2}\cdot||v_{\text{min}}||^2 = \lambda_{\text{min}}$$

So, this x, plotted in our function, would give us $\frac{1}{2}\lambda_{\text{min}}$, which means our bound is attained, which means it is the optimal value.
We denote $\lambda_{\text{min}}(A)$ and $\lambda_{\text{min}}(B)$ our minimal eigenvalues for the matrices A and B. From our previous reasoning, we have that $\frac{\lambda_{\text{min}}(A)}{2}$ and $\frac{\lambda_{\text{min}}(B)}{2}$ are our optimal values, and so the optimal value for our relaxed problem is $\frac{\lambda_{\text{min}}(A)+\lambda_{\text{min}}(B)}{2}$.

TODO On devrait comparer avec le "target problem"

Question 3

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

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

As A and B are symmetric by assumptions, 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:

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

Hence we can write the dual as:

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

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, by question 2,

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

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:

$$\begin{aligned}\frac{\partial}{\partial x_i} f(x, y) &= \frac{1}{2}\frac{\partial}{\partial x_i} x^\top A x = \frac{1}{2}\frac{\partial}{\partial x_i}\sum_{j,k=1}^{n} x_j A_{ik} x_k = \frac{1}{2}\frac{\partial}{\partial x_i}\left(\sum_{\substack{j,k=1 \\ j,k\neq i}}^{n} x_j A_{ik} x_k + A_{ii} x_i^2 + x_i \sum_{\substack{j=1 \\ j\neq i}}^{n} (A_{ij} + A_{ji}) x_j\right) \\ &= \frac{1}{2}\left(2A_{ii}x_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} (A_{ij} + A_{ji}) x_j\right) = \frac{1}{2}\left(2A_{ii}x_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} 2A_{ij} x_j\right) = \sum_{j=1}^{n} A_{ij} x_j = row_i(A) \cdot x\end{aligned}$$

As $\frac{\partial}{\partial x_i} f(x, y)$ is the i-th element of $\nabla_x f(x, y)$, we get 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

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

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 indeed 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
```

Those functions use 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 `main.m`, we defined a function handle which allow 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:
   -3.1138    1.0000
   -3.5622    2.0000
    2.3865   -3.0000
```

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

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

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
   LBeta(x, y, mu) = -6.3694
and h(x, y), beta*h(x, y):
   -0.0128   -3.2785
   -0.0119   -3.0504
   -0.0033   -0.8566

which have norms
    0.0178    4.5593
```

We notice that the norme of $h(x, y)$ is divided by two at each iterations, since $\beta h(x, y)$ stays almost constant. This confirms—as we have seen in the lecture notes; we could hope—that $h(x_k) \approx \frac{\mu^*}{\beta_k}$. This seems indeed to be the case, as we will see in question 11.
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:

```
1  mu = [0; 0; 0];
2  % 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 + ".")
13     disp('We found x and y:')
14     disp([z(1:n), z(n+1:end)])
15     disp("with value f(x, y) = " + f(z))
16     disp("   LBeta(x, y, mu) = " + LBeta(z, mu, beta, A, B))
17     disp('and h(x, y), new mu:')
18     disp([hz, mu])
19     disp('which have norms')
20     disp([vecnorm(hz), vecnorm(mu)])
21
22     beta = 2 * beta; % Updating
23 end
```

Note that the only difference is the update of μ at line 10. 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
   LBeta(x, y, mu) = -6.3288
and h(x, y), new mu:
   -0.0000   -3.2774
    0.0000   -3.0515
    0.0000   -0.8582

which have norms
    0.0000    4.5595

Our obtained value for the Dual problem is mu_1 + mu_2 = -6.3288
```

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 we know is μ^*— 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 ± the same.

Question 12

Our program finds $\mu = \begin{pmatrix} -3.2774 \\ -3.0515 \\ \pm 0.8582 \end{pmatrix}$. We will work with $+0.8582$ for simplicity.

This makes the Lagrangian function be

$$\begin{aligned} L : (x, y) \mapsto & f(x, y) - 3.2774(1 - x^\top x) - 3.0515(1 - y^\top y) + 0.8582 x^\top y \\ & = f(x, y) + 3.2774 x^\top x + 3.0515 y^\top y + 0.8582 x^\top y - 6.3288 \end{aligned}$$

We can also write this $\frac{1}{2} \begin{bmatrix} x^\top & y^\top \end{bmatrix} \begin{bmatrix} A - 2\mu_1 I_n & \mu_3 I_n \\ \mu_3 I_n & B - 2\mu_2 I_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - 6.3288$, using question 3.