

MATH-329 Nonlinear optimization Homework 4: Duality, penalty methods

Anya-Aurore Mauron & Benoît Müller

October 4, 2023

1.

Let's write $S = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n : h(x, y) = 0\}$ the feasible set of problem (P) and define for $i = 1, 2, 3$, $h_i(x, y) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ by

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

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

$$h_3(x, y) = x^\top y$$

for all $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$. Notice that $h : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^3$ is defined by

$$h(x, y) = [h_1(x, y), h_2(x, y), h_3(x, y)]^\top.$$

We have that S is not convex. Here is a counter example: take $x \in \mathbb{R}^n$ such that $x_i = \delta_{i1}$ for all $i = 1, \dots, n$ and $y \in \mathbb{R}^n$ such that $y_i = \delta_{i2}$. We easily check that $(x, y) \in S$. Now take $x' = -x$ and $y' = -y$ in S . If S was convex, then $\frac{1}{2}(x, y) + \frac{1}{2}(x', y') = (\vec{0}, \vec{0})$ should be in S too. But it does not as $h_1(\vec{0}, \vec{0}) = h_2(\vec{0}, \vec{0}) = 1 \neq 0$.

Moreover, LICQ holds at all feasible points. We have for all $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$:

$$\nabla h_1(x, y) = [-2x, 0]^\top,$$

$$\nabla h_2(x, y) = [0, -2y]^\top,$$

$$\nabla h_3(x, y) = [y, x]^\top.$$

The vectors $\nabla h_1(x, y), \nabla h_2(x, y), \nabla h_3(x, y)$ are linearly independent: for all α_1, α_2 and α_3 in \mathbb{R}

$$\alpha_1 \nabla h_1(x, y) + \alpha_2 \nabla h_2(x, y) + \alpha_3 \nabla h_3(x, y) = 0 \quad \forall (x, y) \in S \quad (1)$$

$$\iff$$

$$-2\alpha_1 x + \alpha_3 y = 0 \quad (2)$$

$$-2\alpha_2 y + \alpha_3 x = 0. \quad (3)$$

By multiplying the equation (2) by x^\top :

$$-2\alpha_1 x^\top x + \alpha_3 x^\top y = 0 \quad \forall (x, y) \in S \iff -2\alpha_1 = 0 \iff \alpha_1 = 0, \quad (4)$$

as $x^\top x = 1$ and $x^\top y = 0$ for all $(x, y) \in S$.

By taking the transpose of equation (3) and multiplying it by y :

$$-2\alpha_2 y^\top y + \alpha_3 x^\top y = 0 \quad \forall (x, y) \in S \iff -2\alpha_2 = 0 \iff \alpha_2 = 0 \quad (5)$$

as $y^\top y = 0$ for all $(x, y) \in S$.

By taking equation (2) and $\alpha_1 = 0$:

$$\alpha_3 y = 0. \quad (6)$$

By multiplying by y^\top , we have that $\alpha_3 = 0$.

Therefore $\nabla h_1(x, y), \nabla h_2(x, y), \nabla h_3(x, y)$ are linearly independent, so LICQ holds for $(x, y) \in S$.

2.

For the relaxed problem, we can compare it to the Rayleigh quotient example (Example 8.33 in the lecture notes). Here we are considering a similar problem, the difference is that the cost function sums two functions depending on (a priori) different matrices A and B , and the set S ensures that x and y are, each one of them, of norm one.

Consider $\lambda_1^A \geq \dots \geq \lambda_n^A$ and $\lambda_1^B \geq \dots \geq \lambda_n^B$ in \mathbb{R} the ordered eigenvalues of the matrices A and B respectively. Set e_1^A, \dots, e_n^A and e_1^B, \dots, e_n^B in \mathbb{R}^n the respective orthonormal eigenvectors.

The optimal value of the relaxed problem is given by $(x, y) = (e_n^A, e_n^B) \in \mathbb{R}^n \times \mathbb{R}^n$ such that

$$f(e_n^A, e_n^B) = 1/2\lambda_n^A + 1/2\lambda_n^B$$

If $e_n^{A^\top} e_n^B = 0$ (i.e. they satisfy $h_3(e_n^A, e_n^B) = 0$), then the relaxed and target problems share the same solution. Otherwise, the target problem will have a greater value for $\min_{(x,y) \in S} f(x, y)$ since another condition is added.

3.

Let $\mu = (\mu_1, \mu_2, \mu_3)^\top \in \mathbb{R}^3$. Then, for all $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$, the Lagrangian function is given by:

$$\begin{aligned} L(x, y, \mu) &= f(x, y) + \mu^\top h(x, y) = \frac{1}{2}x^\top Ax + \frac{1}{2}y^\top By + \mu_1(1 - x^\top x) + \mu_2(1 - y^\top y) + \mu_3 x^\top y \\ &= \frac{1}{2}x^\top Ax + \frac{1}{2}y^\top By + \mu_1 - \mu_1 x^\top I_n x + \mu_2 - \mu_2 y^\top I_n y + \mu_3 x^\top y \\ &= x^\top \left(\frac{1}{2}A - \mu_1 I_n \right) x + y^\top \left(\frac{1}{2}B - \mu_2 I_n \right) y + \mu_3 x^\top y + \mu_1 + \mu_2 \\ &= \frac{1}{2}x^\top (A - 2\mu_1 I_n) x + \frac{1}{2}y^\top (B - 2\mu_2 I_n) y + \left(\frac{1}{2}\mu_3 x^\top y + \frac{1}{2}\mu_3 y^\top x \right) + \mu_1 + \mu_2 \end{aligned} \quad (7)$$

with $I_n \in \mathbb{R}^{n \times n}$ the identity matrix. The equalities are by linearity and symmetry of the scalar product. Finally the Lagrangian function can be written under matrix form:

$$\begin{aligned} L(x, y, \mu) &= \frac{1}{2} \begin{bmatrix} x^\top (A - 2\mu_1 I_n) + \mu_3 y^\top I_n & y^\top (B - 2\mu_2 I_n) + \mu_3 x^\top I_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \mu_1 + \mu_2 \\ &= \frac{1}{2} \begin{bmatrix} x^\top & y^\top \end{bmatrix} \begin{bmatrix} A - 2\mu_1 I_n & \mu_3 I_n \\ \mu_3 I_n & B - 2\mu_2 I_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \mu_1 + \mu_2 \end{aligned} \quad (8)$$

Define the matrix $M(\mu) \in \mathbb{R}^{2n \times 2n}$:

$$M(\mu) = \begin{bmatrix} A - 2\mu_1 I_n & \mu_3 I_n \\ \mu_3 I_n & B - 2\mu_2 I_n \end{bmatrix}. \quad (9)$$

4.

Define $\gamma_1 \geq \dots \geq \gamma_{2n} \in \mathbb{R}$ and $a_1, \dots, a_{2n} \in \mathbb{R}^{2n}$ the eigenvalues, respectively the corresponding orthonormal eigenvector basis of the matrix $M(\mu)$. We have two cases:

- if $\gamma_{2n} \geq 0$: as μ is fixed, the quantity $L_D(\mu)$ attains its infimum at $(x, y) = (\vec{0}, \vec{0})$. Indeed, suppose by contradiction that there exists $(x', y') \in \mathbb{R}^n \times \mathbb{R}^n$ such that

$$L(x', y', \mu) < L(\vec{0}, \vec{0}, \mu). \quad (10)$$

We can write (x', y') as a linear combination of a_1, \dots, a_{2n} :

$$(x', y') = \sum_{i=1}^{2n} \beta_i a_i$$

with $\beta_i \in \mathbb{R}$ for $i = 1, \dots, 2n$.

Then :

$$\begin{aligned} L(x', y', \mu) &= \frac{1}{2} \left(\sum_{i=1}^{2n} \beta_i a_i \right)^\top M(\mu) \left(\sum_{j=1}^{2n} \beta_j a_j \right) + \mu_1 + \mu_2 = \frac{1}{2} \sum_{i=1}^{2n} \beta_i^2 a_i^\top M(\mu) a_i + \mu_1 + \mu_2 \\ &= \frac{1}{2} \sum_{i=1}^{2n} \beta_i^2 \gamma_i + \mu_1 + \mu_2 \geq \mu_1 + \mu_2 = L(\vec{0}, \vec{0}, \mu) \end{aligned} \quad (11)$$

as $\{a_i\}_{i=1}^{2n}$ is an orthonormal basis and $\gamma_i \geq 0$ for all $i = 1, \dots, 2n$. We have a contradiction here with Equation (10). Therefore the infimum is finite:

$$L_D(\mu) = \mu_1 + \mu_2$$

By the hint, $M(\mu)$ being positive semi-definite (PSD) implies that its diagonal blocks are PSD as well. In other words $A - 2\mu_1 I_n$ and $B - 2\mu_2 I_n$ are PSD. In particular, their smallest eigenvalue is bigger or equal to zero. With the former notation:

$$\begin{aligned} \lambda_n^A - 2\mu_1 \geq 0 &\iff \mu_1 \leq \frac{1}{2} \lambda_n^A \\ \lambda_n^B - 2\mu_2 \geq 0 &\iff \mu_2 \leq \frac{1}{2} \lambda_n^B. \end{aligned} \quad (12)$$

Therefore the function $L_D(\mu)$ is bounded by:

$$L_D(\mu) \leq \frac{1}{2} (\lambda_n^A + \lambda_n^B). \quad (13)$$

- if $\gamma_{2n} < 0$: the infimum is not finite, as we could take $(x, y) = t a_{2n}$, $t \in \mathbb{R}$. Then we would obtain:

$$\lim_{t \rightarrow \infty} L(x, y, \mu) = \lim_{t \rightarrow \infty} \frac{1}{2} t^2 \gamma_{2n} + \mu_1 + \mu_2 = -\infty. \quad (14)$$

In short, we have:

$$L_D(x, y, \mu) = \begin{cases} \mu_1 + \mu_2 & \text{if } \gamma_{2n} \geq 0 \\ -\infty & \text{otherwise.} \end{cases} \quad (15)$$

5.

By equation (13), the dual problem

$$\max_{\mu \in \mathbb{R}^3} L_D(\mu) \tag{16}$$

attains its optimal value at $\frac{1}{2}(\lambda_n^A + \lambda_n^B)$, if and only if $M(\mu)$ is positive semi-definite.

We have a necessary condition in equation (12). If this condition is satisfied, then the optimal value is attained by $\mu_1 = \frac{1}{2}\lambda_n^A$ and $\mu_2 = \frac{1}{2}\lambda_n^B$.

Let's find μ so that $M(\mu)$ is PSD and that the optimal value is attained. In order for the dual function $L_D(\mu) = \mu_1 + \mu_2$ to attain the optimal value $\frac{1}{2}(\lambda_n^A + \lambda_n^B)$, we necessarily have $\mu_1 = \frac{1}{2}\lambda_n^A$ and $\mu_2 = \frac{1}{2}\lambda_n^B$. We notice that with $\mu_3 = 0$, $M(\mu)$ is PSD :

For all $x, y \in \mathbb{R}^n$:

$$\begin{aligned} [x^\top \quad y^\top] \begin{bmatrix} A - 2\mu_1 I_n & \mu_3 I_n \\ \mu_3 I_n & B - 2\mu_2 I_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} &= [x^\top \quad y^\top] \begin{bmatrix} A - 2\frac{1}{2}\lambda_n^A I_n & 0_n \\ 0_n & B - 2\frac{1}{2}\lambda_n^B I_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ &= \underbrace{x^\top (A - \lambda_n^A I_n) x}_{\geq 0} + \underbrace{y^\top (B - \lambda_n^B I_n) y}_{\geq 0} \geq 0. \end{aligned} \tag{17}$$

by definition of the eigenvalues of A and B .

Thus with $\mu = (\mu_1, \mu_2, \mu_3) = (\frac{1}{2}\lambda_n^A, \frac{1}{2}\lambda_n^B, 0)$, the matrix $M(\mu)$ is PSD. We have found a sufficient condition, so that the optimal value is attained in the feasible set. To conclude, the optimal value is $\frac{1}{2}(\lambda_n^A + \lambda_n^B)$.

By weak duality, for all $\mu \in \mathbb{R}^3$ and $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$:

$$L_D(\mu) \leq L_P(x, y).$$

One of the consequences is that

$$\max_{\mu \in \mathbb{R}^3} L_d(\mu) \leq \min_{(x,y) \in \mathbb{R}^n \times \mathbb{R}^n} L_P(x, y) = \min_{(x,y) \in S} f(x, y). \tag{18}$$

Thus

$$\frac{1}{2}(\lambda_n^A + \lambda_n^B) = \max_{\mu \in \mathbb{R}^3} L_d(\mu) \leq \min_{(x,y) \in S} f(x, y). \tag{19}$$

In light of the second question, we have that the relax problem attains the same optimal value of $\frac{1}{2}(\lambda_n^A + \lambda_n^B)$. We have found that the target problem has the same optimal value.

6.

The hypothesis of the strong duality problem are:

- a) The primal problem has a global minimizer $(x^*, y^*) \in S$;
- b) There exists a valid Lagrange multiplier μ^* for (x^*, y^*) ;
- c) The function $(x, y) \mapsto L(x, y, \mu^*)$ is convex.

Let's verify each one of them.

- a) As the cost function $f(x, y)$ is continuous and the search space is compact, then f admits a global minimum (x^*, y^*) . Indeed S is compact as it is bounded: for all $(x, y) \in S$, $\|(x, y)\| \leq 5$ as $\|x\|^2, \|y\|^2 = 1$. More precisely:

$$\|(x, y)\|^2 = \sum_{i=1}^n x_i^2 + \sum_{j=1}^n y_j^2 = \|x\|^2 + \|y\|^2 = 2 \leq 25. \quad (20)$$

Also S is closed as h_1, h_2 and h_3 are continuous.

- b) Suppose $(x^*, y^*) \in S$ is a global minimizer of $f(x, y)$ on S . It is a stationary point. Since LICQ holds in S by question 1, it holds at (x^*, y^*) . Thus there exists a KKT Lagrange multiplier $\mu^* \in \mathbb{R}^3$ for (x^*, y^*) .
- c) The function $L(x, y, \mu^*)$ is convex if and only if its hessian is positive semi-definite. First, let's make the change the variables:

$$z^\top = [x^\top y^\top] \text{ for } (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \quad (21)$$

with $z \in \mathbb{R}^{2n}$.

The Lagrangian function becomes:

$$L(z, \mu) = \frac{1}{2} z^\top \begin{bmatrix} A - 2\mu_1 I_n & \mu_3 I_n \\ \mu_3 I_n & B - 2\mu_2 I_n \end{bmatrix} z + \mu_1 + \mu_2. \quad (22)$$

We recognize a quadratic function with respect to z . From previous homeworks, we know that the hessian is given by

$$\nabla_z^2 L(z, \mu^*) = \begin{bmatrix} A - 2\mu_1^* I_n & \mu_3^* I_n \\ \mu_3^* I_n & B - 2\mu_2^* I_n \end{bmatrix}. \quad (23)$$

The hessian $\nabla_z^2 L(z, \mu^*)$ being PSD implies that its block matrices are (and not the reverse). That means that $A - 2\mu_1^* I_n$ and $B - 2\mu_2^* I_n$ are PSD. With the same reasoning as in question 4 (see equation 12), we have that

$$\mu_1^* \leq \frac{1}{2} \lambda_n^A \quad \mu_2^* \leq \frac{1}{2} \lambda_n^B. \quad (24)$$

So if we have $\mu_1^* > \frac{1}{2} \lambda_n^A$ or $\mu_2^* > \frac{1}{2} \lambda_n^B$, the hessian is not PSD, and therefore $L(z, \mu^*)$ is not convex (necessary condition).

But if the condition in equation (24) is verified, it does not imply that the hessian is PSD. We would need more information. We would directly need to prove that the hessian $\nabla_z^2 L(z, \mu^*)$ is PSD, in order to show convexity.

7.

First we compute the jacobian of the augmented Lagrangian $L_\beta(z, \mu)$ with respect to x , by using linearity and the matrix product derivation formula; we use the notation $z = (x, y)$:

$$\begin{aligned} D_x(L_\beta(z, \mu)) &= D_x f(z) + \mu^\top D_x h(z) + \frac{\beta}{2} D_x (h(z)^\top h(z)) \\ &= (\nabla_x f(z))^\top + \mu^\top D_x h(z) + \beta h(z)^\top D_x h(z) \\ &= (Ax)^\top + (\mu + \beta h(z))^\top D_x h(z), \end{aligned}$$

where we use (0) a vector of the same dimensions of x and y with null coordinates. We compute then the jacobian of h :

$$D_x h(z) = D_x(1 - x^\top x, 1 - y^\top y, x^\top y)^\top = (-2x, (0), y)^\top.$$

We can now put it in the formula for $D_x(L_\beta(z, \mu))$ and take the transpose to have the gradient :

$$\nabla_x(L_\beta(z, \mu)) = Ax + (-2x, (0), y)(\mu + \beta h(z)).$$

By the same arguments and the symmetry of the situation, we find similar computations for

$$\nabla_y L_\beta(z, \mu) = By + ((0), -2y, x)(\mu + \beta h(z)).$$

8.

The following code is a function which takes as input x, y, μ, β and returns $L_\beta(x, y, \mu)$, $\nabla_x L_\beta(x, y, \mu)$ and $\nabla_y L_\beta(x, y, \mu)$. In addition to that, we add two output for the value of $f(z)$ and $h(z)$, it will be useful later.

```

1 function [L, Gx, Gy, f, h] = lagrangian(x,y,mu,beta)
2 % Question 8.
3 % Compute quantites related to
4 % L_beta(z,mu) = f(z) + <\mu,h(z)> + beta/2||h(z)||^2,
5 % the Augmented Lagrangian of the problem min_S f,
6 % where f(x,y) = 0.5<x,Ax> + 0.5<y,By>,
7 % and S={z:h(z)=0}
8 %
9 % INPUTS :
10 %      (x,y) = z, the point of the space
11 %      mu = the lagrangian parameter
12 %      beta = the cost paramter
13 % OUTPUTS :
14 %      L = the evaluation
15 %      Gx and Gy = the gradients with respect to x and y
16 %      f = the evaluation of the initial function to minimize
17 %      h = the evaluation of the equality constraint function
18 %
19 % If mu is the vector null, it can be put as a scalar.
20
21 A=[ 2 4 4 5 9
22     4 4 8 7 5
23     4 8 2 8 5
24     5 7 8 6 5
25     9 5 5 5 2];
26 B=[ 4 8 3 6 8
27     8 4 4 5 2
28     3 4 2 8 5
29     6 5 8 4 7
30     8 2 5 7 8];
31 f= (x'*A*x + y'*B*y)/2;
32 h= [ 1-x'*x ; 1-y'*y ; x'*y ];
33 L= f + (mu + (beta/2)*h)'*h;
34 Dx= [ -2*x , zeros(5,1) , y ];
35 Dy= [ zeros(5,1) , -2*y , x ];
36 res= mu + beta*h;
37 Gx= A*x + Dx*res;
38 Gy= B*y + Dy*res;
39 end

```

9.

We want to minimize with respect to x and z , so to use optimization functions, we write a short function `zlagrangian` that do the same as `lagrangian` but take x and y in one concatenated vector z , and return the two gradients in one concatenated vector Gz :

```

1 function [L, Gz] = zlagrangian(z,mu,beta)
2 % Question 8.
3 % compute the same as lagrangian.m but all with
4 % x and y concatenated in z.
5     [L, Gx, Gy,~,~] = lagrangian(z(1:5), z(6:end), mu, beta);
6     Gz= [Gx; Gy];
7 end

```

Now we are able to implement the code that minimize the augmented Lagrangian function for one β , it use Matlab's optimization function `fminunc`:

```

1 function [x, y, L]=minimiser(x0, y0, mu,beta)
2 % minimize the cost function F_beta implemented in
3 % lagrangian.m with fminunc and use of gradient.
4
5 L = @(z) zlagrangian(z,mu,beta);
6 options = optimoptions('fminunc','SpecifyObjectiveGradient',true,'Display', 'none');
7 [z, L] = fminunc(L,[x0;y0],options);
8 x= z(1:5);
9 y= z(6:end);
10 end

```

We can finally use this function for some particular values:

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Question 9
3 % Minimize the augmented Lagrangian function
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5 close all
6 clear
7 % Data:
8 beta = 1.42;
9
10 % Initialisation:
11 mu = [1;2;-3];
12 x0 = [1;0;-1;2;1];
13 y0 = [1;2;0;1;2];
14
15 % Computation:
16 [x, y, L] = minimiser(x0, y0, mu,beta);
17
18 % Display:
19 disp('Optimisation ended at the position')
20 x=x
21 y=y
22 disp('with value of objective function')
23 L.beta=L

```

The values we get are in the command are:

```

1 Optimisation ended at the position
2 x =
3     -1.3695
4     -0.2535
5     -0.0107
6      0.0825
7      1.4721
8 y =
9     -1.2693
10      0.9636
11     -1.0786
12      0.7410
13      0.5569
14 with value of objective function
15 L.beta =
16     -26.4171

```

10.

We implement the quadratic penalty method for this problem:

```

1 function [x,y,X,Y,F,H]=QuadraticPenalty(x,y,Beta)
2 % The Quadratic Penalty method for the problem min.S f.
3 % INPUTS :
4 %         (x,y) = z, the point of the space
5 %         Beta = A vector with the values of beta at each step,
6 %               determines the number of iterations
7 % OUTPUTS :
8 %         L = the evaluation
9 %         Gx and Gy = the gradients with respect to x and y
10 %         f = the evaluation of the initial function to minimize
11 %         h = the evaluation of the equality constraint function
12 iter=length(Beta);
13 X=zeros(length(x),iter);
14 Y=X;
15 F=zeros(1,iter);
16 H=zeros(3,iter);
17
18 for j=1:iter
19     [x,y,~] = minimiser(x, y,0,Beta(j));
20     X(:,j)=x;
21     Y(:,j)=y;
22     [~, ~, ~, F(j), H(:,j)] = lagrangian(x,y,0,Beta(j));
23 end
24 end

```

We use it for some random initialisation:

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Question 10
3 % Use of QuadraticPenalty
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5 close all
6 clear
7 % Initialisation:

```



```

8  Beta = 2.^(0:8);
9  x0 = 10*randn(5,1);
10 y0 = 10*randn(5,1);
11
12 % Computation:
13 [x,y,X,Y,F,H]=QuadraticPenalty(x0,y0,Beta);
14
15 % Result values:
16 BH= Beta.*H;
17 normBH= sum(BH.^2,1);
18 steps= BH(:,1:end-1) - BH(:,2:end);
19 diff= sqrt(sum(steps.^2 , 1));
20 normH= sqrt(sum(H.^2,1));
21
22 %% Display evolution of beta h(x,y)
23 figure
24 semilogy(diff, '-.')
25 hold on
26 semilogy(normBH, '-.')
27 title('Question 10. Quadratic penalty method, evolution of $\beta h(x,y)$',...
28       'interpreter','latex')
29 legend('$\\beta_{k+1} h(z_{k+1}) - \beta_k h(z_k) \\$',...
30        '$\\beta_k h(z_k) \\$', 'interpreter','latex')
31 xlabel('step ($k$)', 'interpreter','latex')
32 hold off
33
34 %% Display Evolution of ||h(x,y)||
35 figure
36 semilogy(normH, '-.')
37 hold on
38 semilogy(exp(-(1:9)), '-.')
39 title('Question 10. Quadratic penalty method, evolution of $||h(x,y)||$',...
40       'interpreter','latex')
41 xlabel('step ($k$)', 'interpreter','latex')
42 legend('$\\|h(z_k)\\|$', '$e^{-k}$', 'interpreter','latex')
43
44 %% Display evolution of h(x,y) and beta h(x,y)
45 figure
46 plot(H, '-.', 'color', '#0072BD')
47 hold on
48 plot(BH, '-.', 'color', 'r')
49 title('Question 10. Quadratic penalty method, evolution of $h(x,y)$ and $$\beta h(x,y)$$',...
50       'interpreter','latex')
51 legend('$h_1(x,y)$', '$h_2(x,y)$', '$h_2(x,y)$', '$\beta h_1(x,y)$',...
52        '$\beta h_2(x,y)$', '$\beta h_3(x,y)$', 'interpreter','latex')
53 xlabel('step ($k$)', 'interpreter','latex')
54 hold off

```

The data we get is the following. The values of x are stored in the columns of the variable X:

1	X =									
2		-1.3391	-1.0365	-0.8481	-0.7350	-0.6708	-0.6360	-0.6177	-0.6083	-0.6036
3		-0.4804	-0.4026	-0.3517	-0.3248	-0.3114	-0.3050	-0.3019	-0.3003	-0.2996
4		0.4334	0.3792	0.3503	0.3393	0.3364	0.3362	0.3365	0.3368	0.3370
5		-0.1131	-0.1000	-0.0978	-0.0989	-0.1007	-0.1022	-0.1031	-0.1036	-0.1038
6		1.4538	1.1276	0.9235	0.8012	0.7318	0.6941	0.6743	0.6642	0.6591

The values of y are stored in the columns of the variable Y:

1	Y =									
2		-0.9626	-0.7603	-0.6284	-0.5507	-0.5075	-0.4845	-0.4727	-0.4666	-0.4636
3		0.7537	0.5928	0.4911	0.4314	0.3984	0.3809	0.3718	0.3673	0.3649
4		-1.2185	-0.9649	-0.8132	-0.7252	-0.6767	-0.6508	-0.6374	-0.6306	-0.6272
5		0.9559	0.7630	0.6450	0.5764	0.5383	0.5180	0.5074	0.5020	0.4992

The values of $f(x, y)$ are stored in the columns of the variable F:

1	F =									
2		-27.0394	-16.6912	-11.5137	-8.9232	-7.6269	-6.9781	-6.6536	-6.4912	-6.4100

The values of $h(x, y)$ are stored in the columns of the variable H:

1	H =									
2		Columns 1 through 5								
3		-3.3381e+00	-1.6619e+00	-8.2810e-01	-4.1261e-01	-2.0572e-01				
4		-2.9969e+00	-1.5051e+00	-7.5479e-01	-3.7863e-01	-1.8985e-01				
5		7.5882e-01	3.8896e-01	2.0016e-01	1.0270e-01	5.2319e-02				
6		Columns 6 through 9								
7		-1.0267e-01	-5.1275e-02	-2.5621e-02	-1.2807e-02					
8		-9.5114e-02	-4.7614e-02	-2.3823e-02	-1.1916e-02					
9		2.6462e-02	1.3317e-02	6.6809e-03	3.3463e-03					

The values of $\beta h(x, y)$ are stored in the columns of the variable BH:

1	BH =									
2		-3.3381	-3.3239	-3.3124	-3.3009	-3.2916	-3.2853	-3.2816	-3.2795	-3.2785
3		-2.9969	-3.0102	-3.0192	-3.0291	-3.0376	-3.0436	-3.0473	-3.0493	-3.0504
4		0.7588	0.7779	0.8006	0.8216	0.8371	0.8468	0.8523	0.8552	0.8566

We comment briefly the behavior of $\|h(x, y)\|$ and $\beta h(x, y)$: The code "question10" we wrote, display two plots:

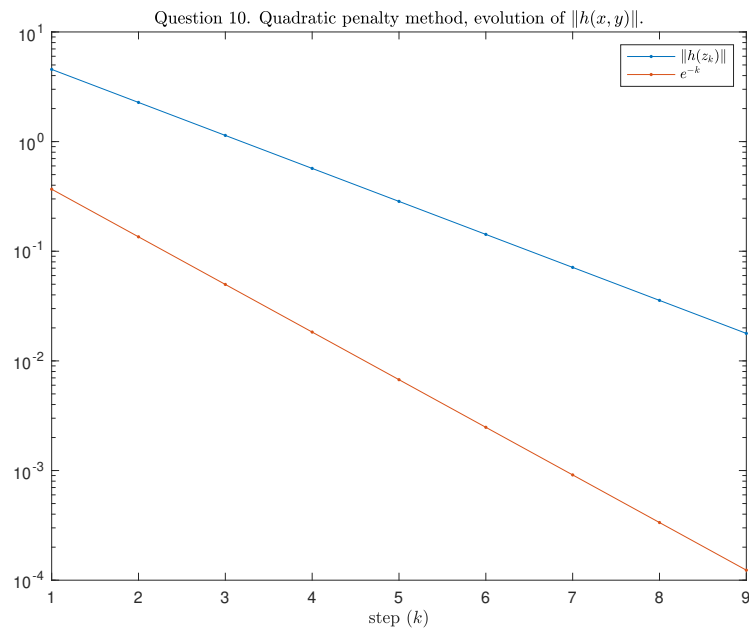


Figure 1

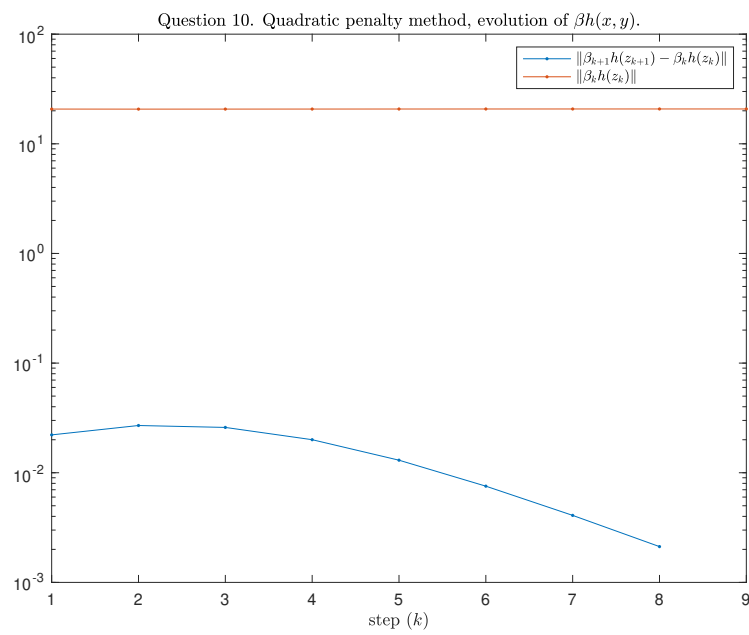


Figure 2

As we can see, the norm of h decrease very fast, exponentially in k . This show that the current position go near the feasible set. However, if we take the β into account to watch $\beta h(x, y)$ which is the step between two consecutive μ , we see that its value is very stable in the space. It's norm seem constant, and more than that, the length of the steps it makes, are small: absolutely, and relatively to its norm. This mean that it really doesn't change.

The assumptions of theorem 10.10 are true: $\beta_k \rightarrow \infty$, z_k are assumed critical points thanks to fminunc. They seem to converge to a point z^* , that is mostly feasible thanks to the fact that h is almost zero, LICQ holds thanks to question 1, and so MFCQ holds too. We see that the conclusion is confirmed, $\beta_k h(z_k)$ is bounded and the point to which it tends to a, is a valid pair of Lagrange multipliers in z^* , which is stationary.

Notice that f actually increase during the process, but it's because the method move outside the feasible set (most of the time because it is a meagre set), and tends to it.

The following graphic resume the values of h and βh :

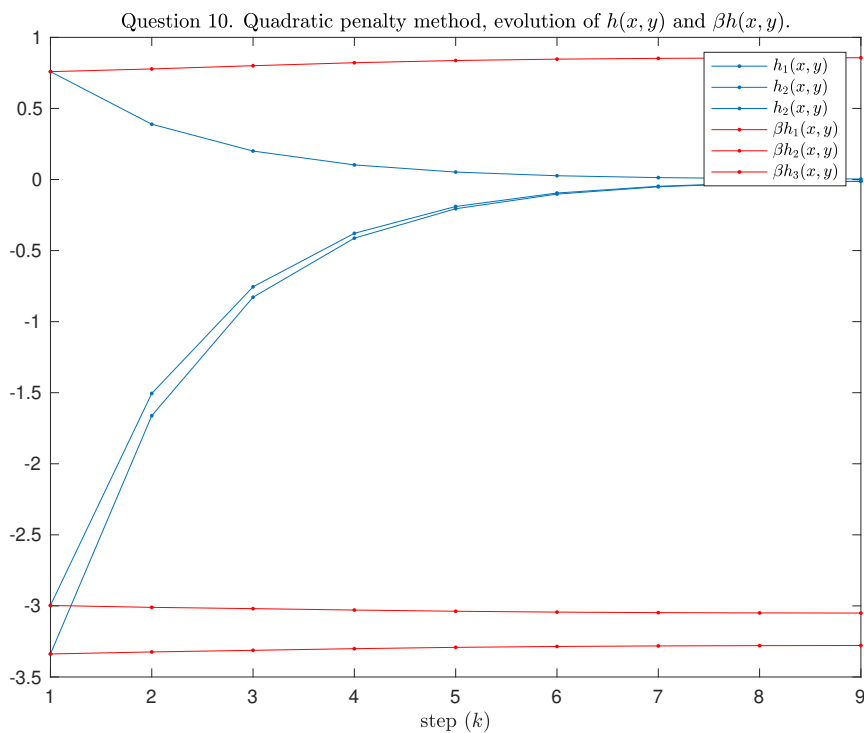


Figure 3

11.

We implement the augmented Lagrangian method for this problem:

```

1 function [x,y,X,Y,F,H,Mu] = AugmentedLagrangian(x,y,mu,Beta)
2 % The Augmented Lagrangian method for the problem min_S f.
3 % INPUTS :
4 %         (x,y) = z, the point of the space

```

```

5 %           mu = the initial lagrangian parameter
6 %           Beta = A vector with the values of beta at each step,
7 %           determines the number of iterations
8 % OUTPUTS :
9 %           (x,y) = the final minimizer
10 %          X and Y = The successive x and y, stored in collumns
11 %          F = the successive evaluations of f stored in collumns
12 %          H = the successive evaluations of h stored in collumns
13 %          Mu = the successive values of mu stored in collumns
14
15 iter=length(Beta);
16 X=zeros(length(x),iter);
17 Y=X;
18 F=zeros(1,iter);
19 H=zeros(3,iter);
20 Mu=H;
21
22 for j=1:iter
23     [x,y,~] = minimiser(x, y,mu,Beta(j));
24     X(:,j)=x;
25     Y(:,j)=y;
26     [~, ~, ~, F(j), H(:,j)] = lagrangian(x,y,0,Beta(j));
27     Mu(:,j)=mu;
28     mu = mu + Beta(j)*H(:,j);
29 end
30 end

```

We use it for some random initialisation:

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Question 11
3  % Use of AugmentedLagrangian
4  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5  close all
6  clear
7  % Initialisation:
8  mu=0;
9  Beta = 2.^(0:8);
10 x0 = 10*randn(5,1);
11 y0 = 10*randn(5,1);
12 iter=9;
13
14 % Computation:
15 [x,y,X,Y,F,H,Mu]=AugmentedLagrangian(x0,y0,mu,Beta);
16
17 % Result values:
18 BH= Beta.*H;
19 normBH= sum(BH.^2,1);
20 steps= BH(:,1:end-1) - BH(:,2:end);
21 diff= sqrt(sum(steps.^2 , 1));
22 normH= sqrt(sum(H.^2,1));
23
24 %% Display Evolution of beta h(x,y)
25 figure
26 semilogy(diff, '-.')
27 hold on
28 semilogy(normBH, '-.')
29 title('Question 11: Augmented Lagrangian method, evolution of $\beta h(x,y)$','interpreter','latex')
30 legend('$\beta_{k+1} h(z_{k+1}) - \beta_k h(z_k)$','$\beta_k h(z_k)$','interpreter','latex')
31 xlabel('step ($k$)','interpreter','latex')
32 hold off

```

```

33
34 %% Display Evolution of ||h(x,y)||:
35 figure
36 semilogy(normH, '-.')
37 hold on
38 semilogy(exp(-(1:9)), '-.')
39 title('Question 11: Augmented Lagrangian method, evolution of  $\|h(x,y)\|$ ','interpreter','latex')
40 xlabel('step ($k$)','interpreter','latex')
41 legend('$\|h(z_k)\|$', '$e^{-k}$','interpreter','latex')
42
43 %% Display Evolution of mu:
44 figure
45 plot(Mu, '-.')
46 title('Question 11: Augmented Lagrangian method, evolution of  $\mu$ ','interpreter','latex')
47 xlabel('step ($k$)','interpreter','latex')
48 legend('$\mu_1$', '$\mu_2$', '$\mu_3$', 'interpreter','latex')

```

The data we get is the following. The values of x are stored in the columns of the variable X:

1	X =								
2		-1.3358	-0.5924	-0.6019	-0.5987	-0.5988	-0.5988	-0.5988	-0.5988
3		-0.4893	-0.2933	-0.2961	-0.2988	-0.2988	-0.2988	-0.2988	-0.2988
4		0.4352	0.3231	0.3316	0.3371	0.3372	0.3372	0.3372	0.3372
5		-0.1078	-0.0961	-0.1019	-0.1041	-0.1041	-0.1041	-0.1041	-0.1041
6		1.4521	0.6471	0.6571	0.6538	0.6539	0.6539	0.6539	0.6539

The values of y are stored in the columns of the variable Y:

1	Y =								
2		0.9671	0.4863	0.4603	0.4609	0.4605	0.4605	0.4605	0.4605
3		-0.7491	-0.3815	-0.3622	-0.3629	-0.3626	-0.3626	-0.3626	-0.3626
4		1.2137	0.6202	0.6228	0.6237	0.6237	0.6237	0.6237	0.6237
5		-0.9633	-0.4858	-0.4960	-0.4963	-0.4965	-0.4965	-0.4965	-0.4965
6		-0.3185	-0.1626	-0.1445	-0.1450	-0.1447	-0.1447	-0.1447	-0.1447

The values of $f(x, y)$ are stored in the columns of the variable F:

1	F =								
2		-27.0282	-6.3463	-6.3356	-6.3290	-6.3289	-6.3288	-6.3288	-6.3288

The values of $h(x, y)$ are stored in the columns of the variable H:

1	H =									
2		Columns 1 through 5								
3		-3.3335e+00	3.0649e-02	-2.0728e-03	4.0643e-04	-8.6815e-06				
4		-2.9989e+00	-2.9100e-02	2.1782e-03	-4.0581e-04	8.6800e-06				
5		-7.5578e-01	-3.4339e-02	-7.6795e-03	-3.2636e-04	-2.4175e-05				
6		Columns 6 through 9								
7		2.4628e-07	-5.2901e-08	5.6256e-08	-2.3243e-08					
8		-2.4609e-07	5.2757e-08	-5.6157e-08	2.3193e-08					
9		-1.5097e-07	-5.5143e-08	7.2996e-09	9.6474e-09					

The values of μ are stored in the columns of the variable Mu:

1	Mu =								
2		0	-3.3335	-3.2722	-3.2805	-3.2772	-3.2774	-3.2774	-3.2774
3		0	-2.9989	-3.0571	-3.0484	-3.0516	-3.0515	-3.0515	-3.0515
4		0	-0.7558	-0.8245	-0.8552	-0.8578	-0.8582	-0.8582	-0.8582

In the he code "question11" we wrote, we display again the two same plots as in question 10, to be able to compare them:

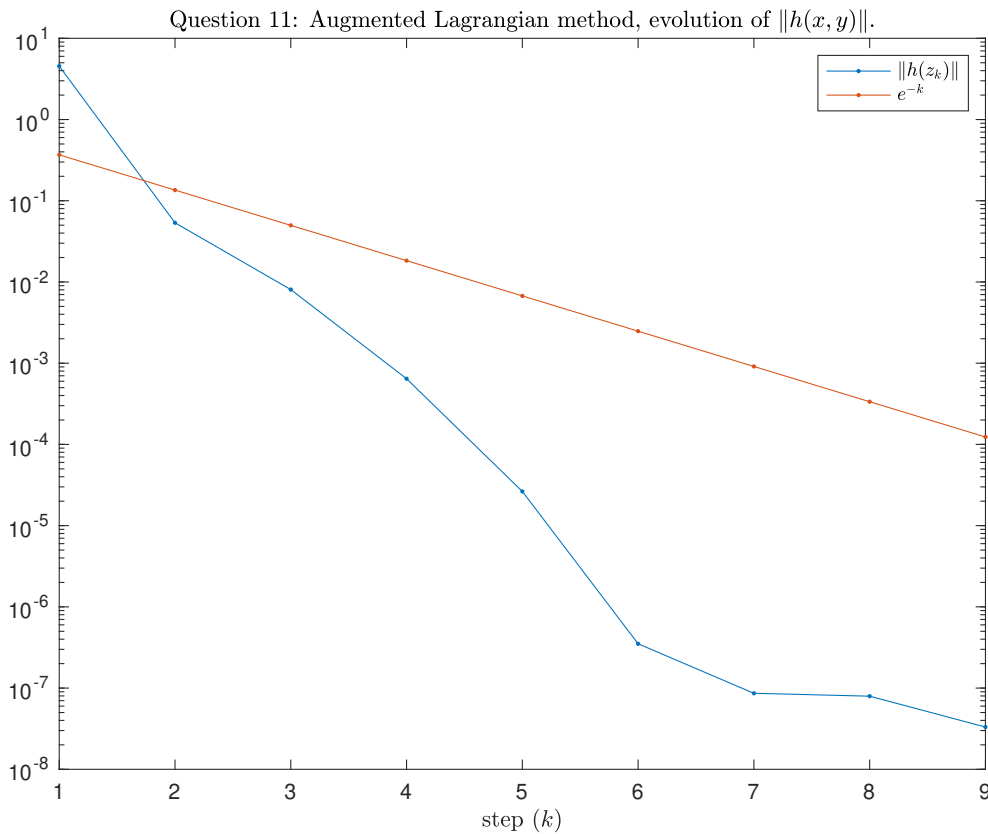


Figure 4

Here, the norm of h decrease very fast, exponentially in k , even faster and smaller than before. We attain the seventh negative power of ten, versus the second for the penalty method. This show that the current position of h goes faster and nearer to the feasible set. If we look at $\beta h(x, y)$, we see that its value is not constant. Its norm decreases, as well as the length of the steps it makes. But if they are absolutely small, they are bigger than the own norm of H , it does change direction of search. The quantity $\beta h(x, y)$ is just the step between to consecutive μ , which converges indeed.

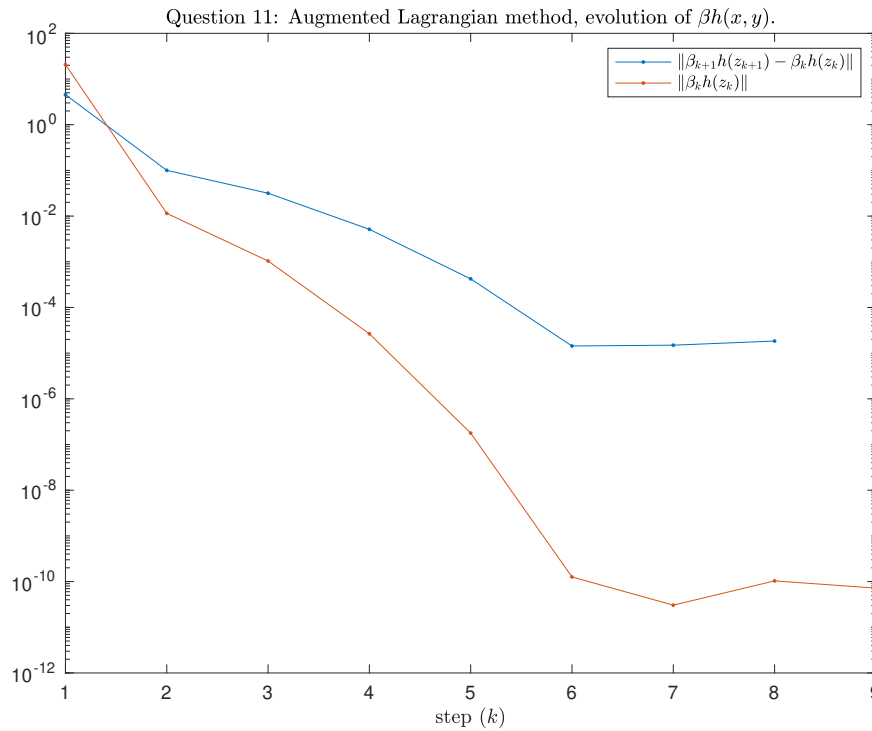


Figure 5

12.

We use (24) to show that it is not convex: $\mu(1) - 0.5 \cdot \min(\text{eig}(A))$ and $\mu(2) - 0.5 \cdot \min(\text{eig}(B))$ give 0.2872 and 0.1195, so the hessian is not PSD. As a matter of fact, with question 6 and (23) we can quantify convexity on the positivity of the hessian matrix

$$\nabla_z^2 L(z, \mu) = \begin{bmatrix} A - 2\mu_1 I_n & \mu_3 I_n \\ \mu_3 I_n & B - 2\mu_2 I_n \end{bmatrix}.$$

We check numerically too if the Lagrangian is convex by plotting the images of random lines:

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Question 12
3  % Convexity of the Lagrangian
4  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5  A=[ 2 4 4 5 9
6      4 4 8 7 5
7      4 8 2 8 5
8      5 7 8 6 5
9      9 5 5 5 2];
10 B=[ 4 8 3 6 8
11      8 4 4 5 2
12      3 4 2 8 5
13      6 5 8 4 7
14      8 2 5 7 8];
    
```



```

15 mu = [-3.27737103718143; -3.05147230163484; -0.858177313718625]
16 % By quadratic function formula and eigenvalues:
17 H= [ A-2*mu(1)*eye(5) , mu(3)*eye(5) ;
18      mu(3)*eye(5)      , B-2*mu(2)*eye(5) ];
19 disp('The smallest eigenvalue of the hessian of the Lagrangian is')
20 lambda = min(eig(H))
21
22 % By random lines:
23 t=linspace(-5,5,100);
24 for j=1:1000
25     v=rand(10,1);
26     v=v/norm(v);
27     L=zeros(1,100);
28     for k=1:100
29         [L(k), ~] = zlagrangian(t(k)*v,mu,0);
30     end
31     plot(L)
32     hold on
33 end
34 hold off

```

This gives the following plot:

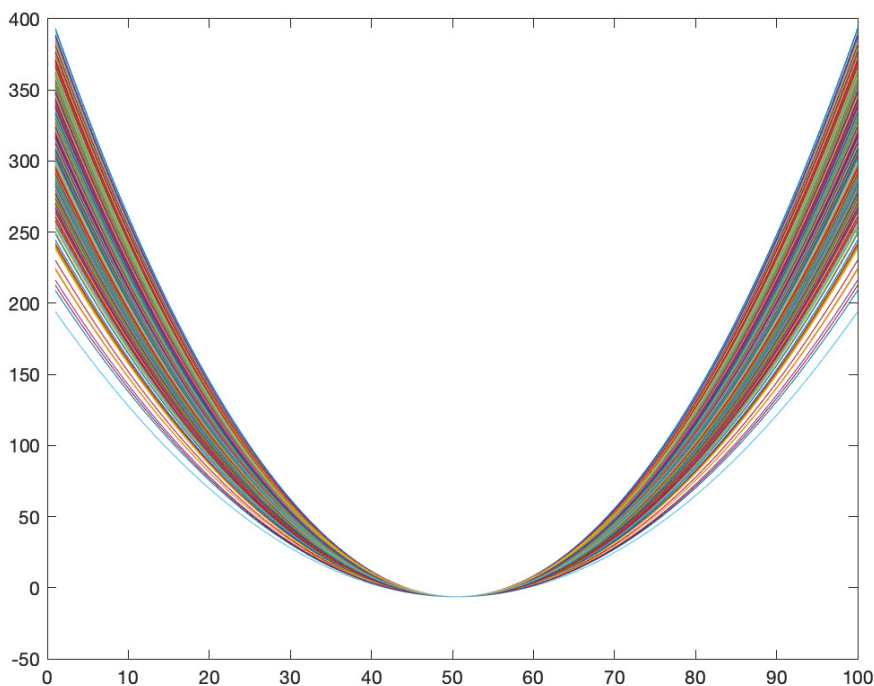


Figure 6

We see that all these curves are convex and that however, the matrix is not semi positive definite. The range of eigenvalues goes from -1.1210 to 34.5590. This mean that the function is almost convex, except from the direction to the eigenvector associated to the eigenvalue -1.1210 (which is the only significant

negative value).

The question 5 showed us in (19) that

$$\frac{1}{2}(\lambda_n^A + \lambda_n^B) \leq \min_{(x,y) \in S} f(x,y), \quad (25)$$

where λ_n^A and λ_n^B are the smallest eigenvalues of A and B . We compute their mean with $(\min(\text{eig}(A)) + \min(\text{eig}(B))) / 2$ in MATLAB and obtain -6.7355. This is pretty near under -6.4100 and -6.3288 that we obtained for the two methods of the last questions. This should give a hint that we are near the best solution and that we cannot go down much more. However, this has to be nuanced because we don't know if this difference is significantly big or small for the problem, and we must have lost precision due to finite precision machine. The hypothesis of the strong duality theorem are not satisfy, so we cannot conclude on strong duality, although it seems to be satisfied numerically.