

In the name of God



Sharif University of Technology

School of Electrical Engineering

Convex Optimization

Homework Nr. 1

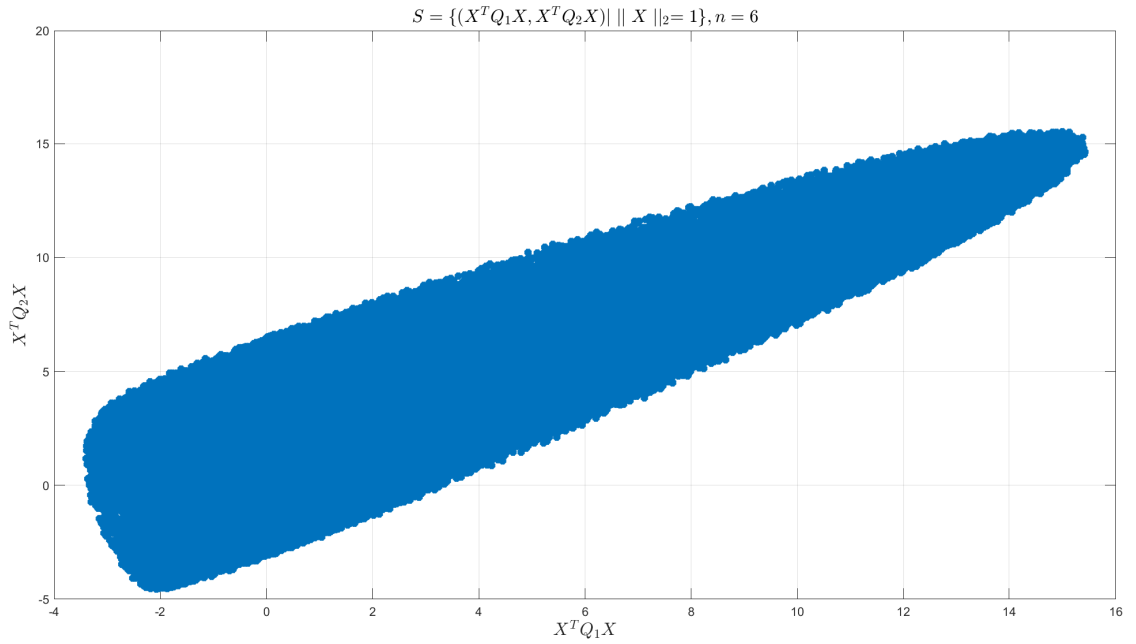
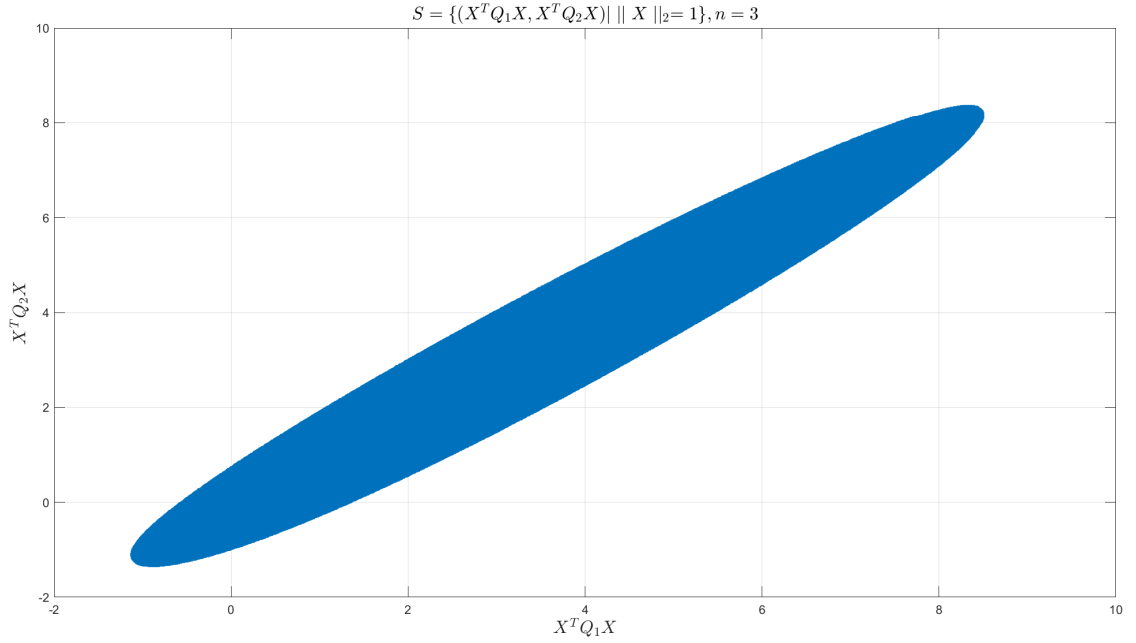
Dr. Babazadeh

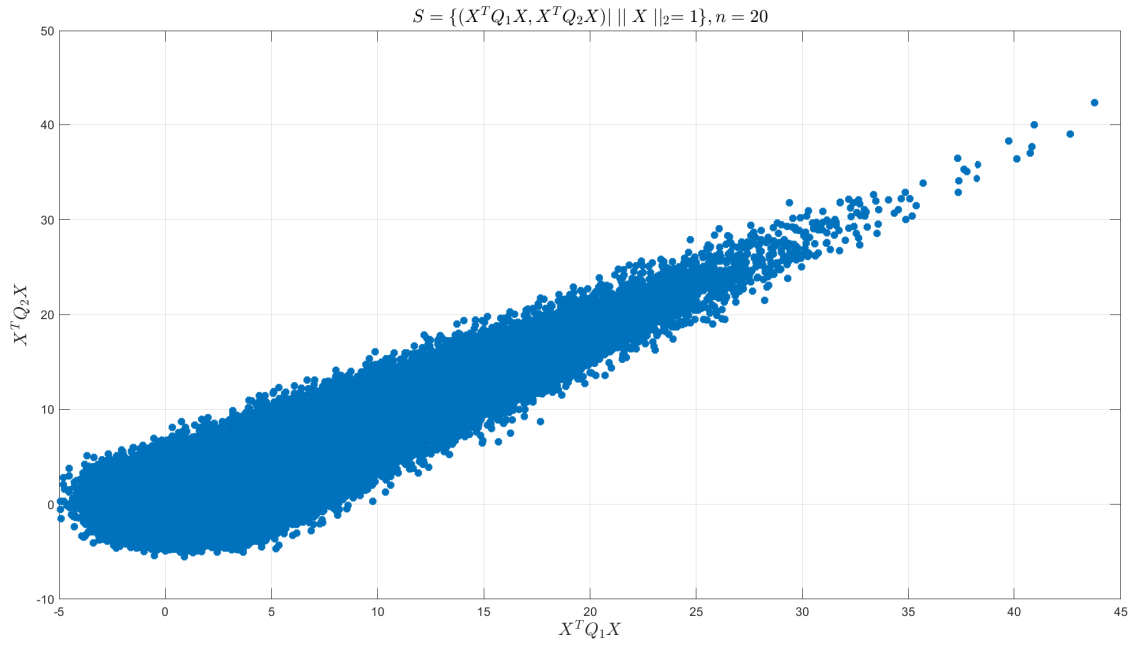
Taha Entesari

95101117

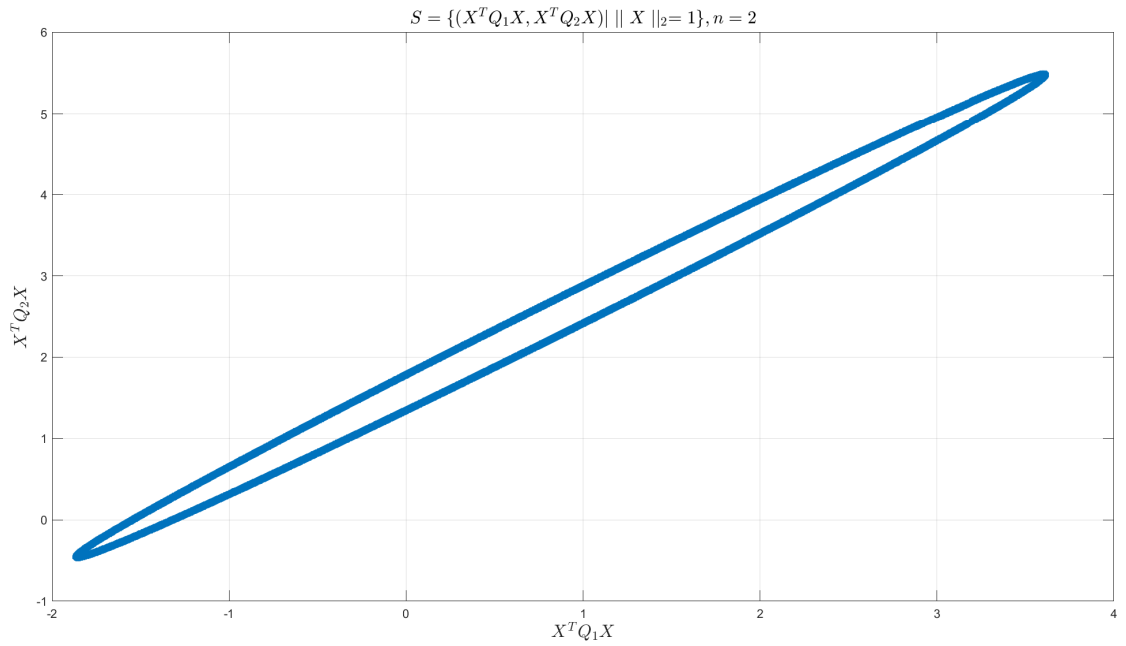
Problem 6

Below the simulation results for different values of \mathbf{n} are presented. As it can clearly be seen from the figures, the set is *convex*. These pictures are just sample results and if Q_1 and Q_2 change, we will get different results but the set still remains convex for all symmetric Q .





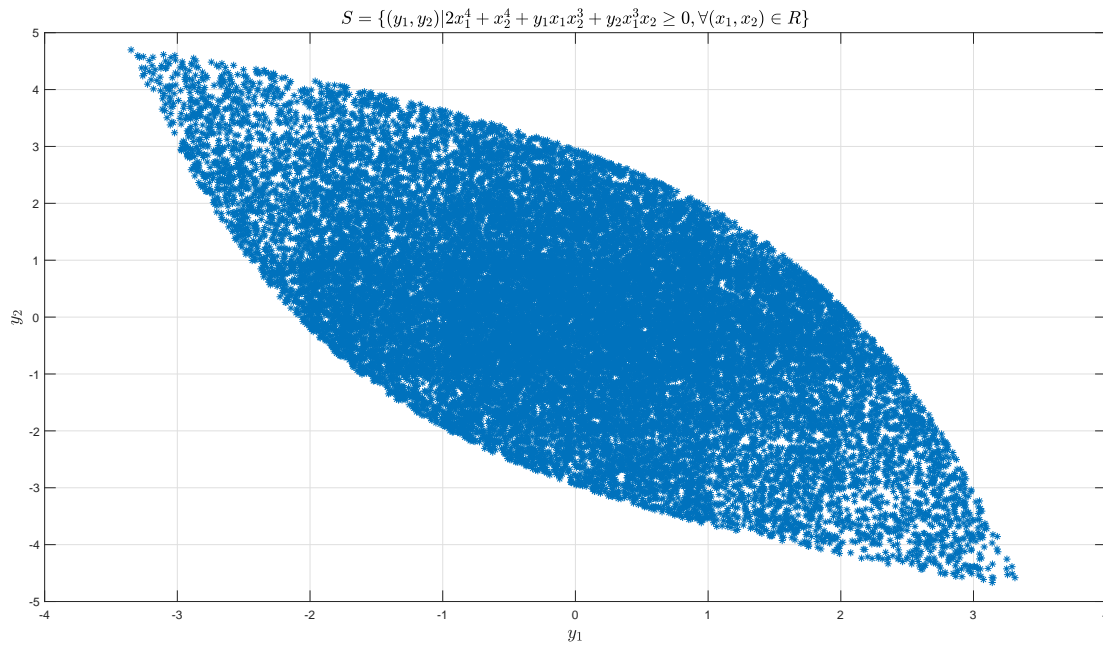
As it is mentioned in the question, the matrices Q_1 and Q_2 need to have $n > 2$. For $n=2$, the set defined by the points is the boundary of a convex set. A sample figure is shown below:



Problem7

Part a

For this problem, first 4 vectors are randomly generated. These vectors are y_1, y_2, x_1, x_2 . Each vector has elements in each quarter of R^2 plane. Using a for loop the required condition is checked and the plot is sketched using only the points that satisfy the inequality for all the generated x points. The below figure is the result and as it can be seen, the set is convex.



Part b

Consider the given equation for a constant vector (x_1, x_2) , the equation is writable as $A^T y + b \geq 0$ and is thus the set of the intersection of hyperplanes and thus is convex.

Problem 8

Part b

Below are a few results for different k . The first image is for $k=1$. The second

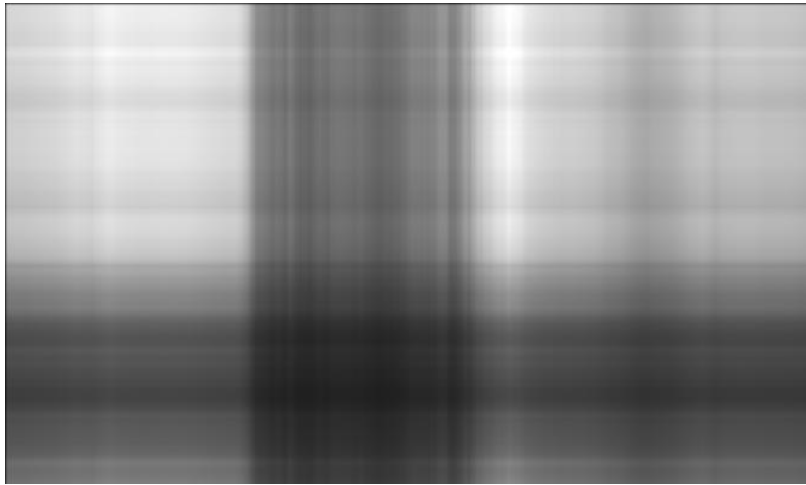


image is for $k=6$. The third image is for $k=21$. The fourth image is for $k=50$.



The final image is for $k=200$. It is evident that we can reconstruct the original



signal using the k singular values and the error drops pretty quickly. For example, the following errors were recorded using MATLAB. The 1st column of this

	1	2
1	1	68.5536
2	6	19.0365
3	11	12.7701
4	16	9.5193
5	21	7.4982
6	26	5.9728
7	31	5.2671
8	36	4.4721
9	50	3.1763
10	100	1.4725
11	200	0.5263
12	400	1.2504e-12

country is the value of k and the data on the 2nd column is the corresponding error.

Though it is probably not a part of the course, it is important to notice that though we have used a few number of singular values to *compress* the image, this does not mean that the resultant *.jpg* image will also take up less space. This is due to the encoding of the *jpg* format. Quite the opposite was seen, i.e. through using 50 singular values, we get a file that takes up more space

that the original black and white image. This can somewhat be understood. By using the singular value decomposition, it is possible that for example the background of the image becomes more detailed as in having less of an equal shape. Due to the nature of the *jpg* encoding format, this can result in a larger file.

Part c

Looking at the final figure for the error rate, $k=50$ would seem like a reasonable choice since the image has most of the detail as the original image and the error is also small.

Code Appendix

```
1 %% Q6
2 clc
3 close all
4 clear
5 for n=2
6     figure('units','normalized','outerposition',[0 0 1 1])
7     q1=rand(n,n);
8     q1=.5*(q1+q1')*5;
9     q2=rand(n,n);
10    q2=.5*(q2+q2')*5;
11    %q1=q1/norm(q1);
12    %q2=q2/norm(q2);
13    tries=500000;
14    parfor i=1:tries
15        fprintf('%d\n',i);
16        x=rand(n,1);
17        x=(-1).^randi(2,n,1).*x/norm(x);
18        %x=randn(n,1);
19        %x=x/norm(x);
20        S1(i)=x'*q1*x;
21        S2(i)=x'*q2*x;
22    end
23    plot(S1,S2,'*', 'LineWidth',3);
24    grid on
25    title(['\textbf{$S = \{(X^TQ_1X,X^TQ_2X) | \mid\mid\{X\}\mid\mid_2 = 1\}, n = '$
        , num2str(n), '$\}$'], 'interpreter','latex','FontSize',15)
26    xlabel('\textbf{$X^TQ_1X$}', 'interpreter','latex','FontSize',15)
27    ylabel('\textbf{$X^TQ_2X$}', 'interpreter','latex','FontSize',15)
28    txt='q1%d';
29    txt=sprintf(txt,n);
30    saveas(gca,txt,'epsc')
31 end
32
33
34
35 %% Q7
36 clc
37 clear
38 close all
39 tic
40 n=20000;
41 range=10;
42 y1=[rand(1,n),-rand(1,n),rand(1,n),-rand(1,n)].*randi(range,1,4*n);
43 y2=[rand(1,n),-rand(1,n),-rand(1,n),rand(1,n)].*randi(range,1,4*n);
44 x1=[rand(1,n),-rand(1,n),rand(1,n),-rand(1,n)].*randi(range,1,4*n);
45 x2=[rand(1,n),-rand(1,n),-rand(1,n),rand(1,n)].*randi(range,1,4*n);
46 a=2*x1.^4+x2.^4;
47 b=x1.*x2;
48 %
49 res=zeros(4*n,1);
50 for i=1:4*n
51
52     fprintf('%d\n',i);
53     res(i)=all(a+b.*(y1(i)*x2.^2+y2(i)*x1.^2)>=0);
54 end
55 res=res==1;
56 toc
57 figure('units','normalized','outerposition',[0 0 1 1])
```

```

58 plot(y1(res'),y2(res'),'*')
59 title('\textbf{$S = \{(y_1, y_2) | 2x_1^4+x_2^4+y_1x_2^3+y_2x_1^3x_2 \geq 0, \}$}$', 'interpreter', 'latex', 'FontSize',15)
60 xlabel('\textbf{$y_1$}', 'interpreter', 'latex', 'FontSize',15)
61 ylabel('\textbf{$y_2$}', 'interpreter', 'latex', 'FontSize',15)
62 grid on
63 saveas(gca, 'q7', 'eps');
64
65 %% Q8
66 clear
67 clc
68 close all
69 A=imread('HajiFirouz.jpg');
70 A=im2double(A);
71 A=rgb2gray(A);
72 [u,sigma,v]=svd(A);
73 w=1;
74 for k=[1:5:40,50,100,200,400]
75     uk=u(:,1:k);
76     vk=v(:,1:k);
77     sigma_k=sigma(1:k,1:k);
78     Ak=uk*sigma_k*vk';
79     figure();
80     subplot(2,1,1)
81     imshow(A)
82     txt='The value of the cost function for k = %d is %f\n';
83     txt=sprintf(txt,k,norm(A-Ak));
84     title(txt)
85     subplot(2,1,2)
86     imshow(Ak)
87     title(['Compressed image for k = ', num2str(k)], 'FontSize',15)
88     txt='ReconstructedHajik=%d.jpg';
89     txt=sprintf(txt,k);
90     imwrite(Ak,txt);
91     err(w)=norm(Ak-A);
92     w=w+1;
93 end

```
