In the name of God

Sharif University of Technology

School of Electrical Engineering

# Convex Optimization

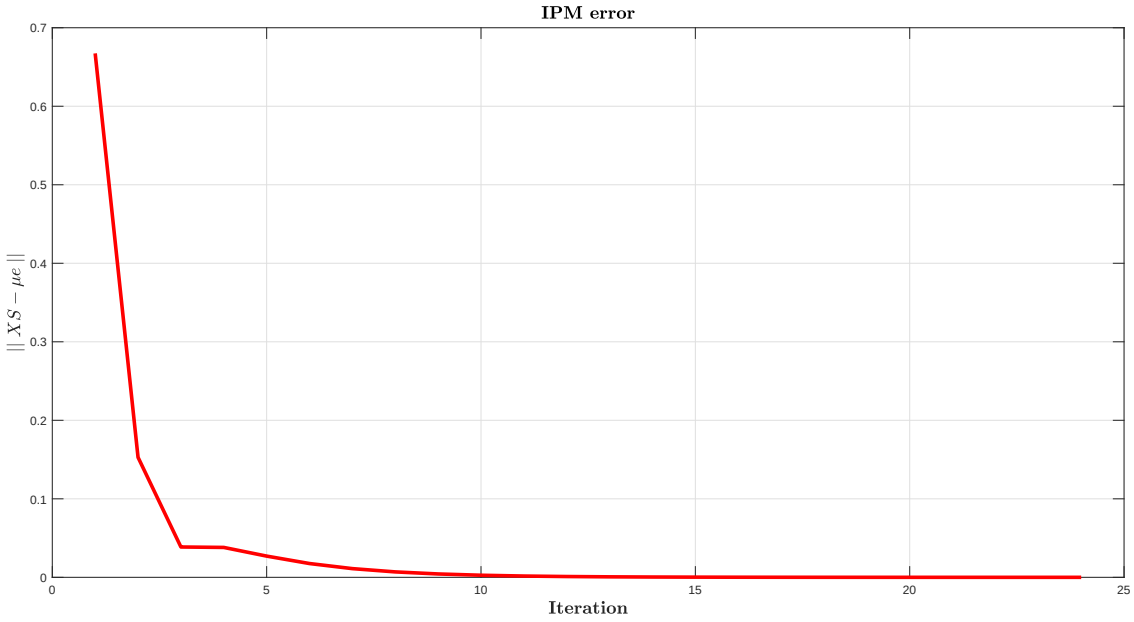
# HW Nr. 4, MATLAB

Dr. Babazadeh

Taha Entesari

95101117

I was unable to type parts a to c of this homework due to numerous tasks and have sent beside this file the scanned version. The required code is available in the appendix of this pdf file. The `.m` file is also available in the folder should you want to run the code for yourself.
A sample run of the code results in the following plot: Having run the code




several times I noticed that there also existed cases in which though CVX resulted in a bounded finite optimum value, the implemented method would diverge, But these may be extreme cases with matrices that might be very close to being singular.

## Appendix: MATLAB Code

```
clear
clc
close all
n=5;
m=2;
Q=abs(randn(n,n));
Q=Q+Q';
Q=Q+diag(max(max(Q))+diag(Q));
Q=Q/max(max(Q));
A=randn(m,n);
b=randn(m,1);
c=randn(n,1);
alpha0=0.99;
sigma=.2;
k=0;
x=abs(randn(n,1)/rand());
y=zeros(m,1);
s=abs(randn(n,1)/rand());
mu=x'*s/n;
zetap=b-A*x;
zetad=c-A'*y-s+Q*x;
epsp=10^(-5);
```

```
epsd=10^(-5);
eps0=10^(-5);
% CVX
cvx_begin
variable w(n)
minimize(1/2*w'*Q*w+c'*w)
subject to
A*w==b;
w>=0;
cvx_end
% Interior point method
while (norm(zetap)/(1+norm(b))>epsp || ...
        norm(zetad)/(1+norm(c))>epsd ||...
        x'*s/n/(1+abs(c'*x+1/2*x'*Q*x))>eps0)
    mu=sigma*mu;
    AA=[A, zeros(m,m), zeros(m,n);...
        -Q, A', eye(n);...
        diag(s), zeros(n,m), diag(x)];
    BB=[b-A*x;c+Q*x-A'*y-s;sigma*mu*ones(n,1)-diag(x)*diag(s)*ones(n,1)];
    res=AA\BB;
    dx=res(1:n);
    dy=res(n+1:n+m);
    ds=res(n+m+1:end);
    alphap=100;
    while any(x+alphap*dx<0)
        alphap=alphap/4
    end
    alphad=100;
    while any(s+alphad*ds<0)
        alphad=alphad/4
    end
    alphap=alpha0*alphap;
    alphad=alpha0*alphad;
    x=x+alphap*dx;
    y=y+alphad*dy;
    s=s+alphad*ds;
    zetap=b-A*x;
    zetad=c-A'*y-s+Q*x;
    k=k+1
    X(k)=1/2*x'*Q*x+c'*x;
    error(k)=norm((diag(x)*diag(s)-mu)*ones(n,1));
end
cvx_optval
X(end)
plot(error,'r','LineWidth',3)
grid on
title('\textbf{IPM error}','interpreter','latex','FontSize',15);
xlabel('\textbf{Iteration}','interpreter','latex','FontSize',15);
ylabel('\textbf{$\mid\mid XS-\mu e\mid\mid$}','interpreter','latex','FontSize'
    ,15);
```