ADMM (Alternating Direction Method of Multipliers)

Let us consider the following opt-prob.

$$\boxed{\begin{array}{ll} \min\limits_{x,z} & f(x)+g(z) \\ s.t. & Ax+Bz=c \end{array}}$$

Augmented Lagrangian: $\boxed{L_\rho(x,z;y)=f(x)+g(z)+y^T(Ax+Bz-c)+\left(\frac{\rho}{2}\right)\|(Ax+Bz-c)\|_2^2}$

THE ADMM ALGO: (Assume: f and g are cvx fcns)

$$\boxed{x^{k+1} := \operatorname*{argmin}_x L_\rho\left(x,z^k,y^k\right)}$$

$$\boxed{z^{k+1} := \operatorname*{argmin}_z L_\rho\left(x^{k+1},z,y^k\right)}$$

$$\boxed{y^{k+1} := \boxed{y^k} + \rho\left(Ax^{k+1}+Bz^{k+1}-c\right)}$$

Example: $\boxed{\begin{array}{ll} \min\limits_{x,z} & x^TQ_xx+z^TQ_zz \\ s.t. & Ax+Bz=c \end{array}}$, $\boxed{\begin{array}{c} x\in\mathbb{R}^{10} \\ z\in\mathbb{R}^{20} \\ c\in\mathbb{R}^5\rightarrow y\in\mathbb{R}^5 \end{array}}$, $\boxed{\begin{array}{l} Q_x\succcurlyeq 0 \\ Q_z\succcurlyeq 0 \end{array}}$

```
%% ADMM problem
clear all,close all,clc;yalmip('clear');
% REQUIRED TOOLS: YALMIP,MOSEK
%% problem construction
x_real=[1:1:10]'; x_dim=length(x_real);
z_real=[1:1:20]'; z_dim=length(z_real);
rng(123)
Qx=randi([-10,10],x_dim,x_dim);
Qx=Qx'*Qx;

Qz=randi([-10,10],z_dim,z_dim);
Qz=Qz'*Qz;

A=randi([-10,10],5,x_dim);
B=randi([-10,10],5,z_dim);
c=A*x_real+B*z_real;
rho=0.001;
y_dim=length(c);
```

```
%% the LOOP
x_val=ones(x_dim,1);z_val=ones(z_dim,1);y_val=ones(y_dim,1);
z_k=z_val;   y_k=y_val;
N=1e2; % # iterations
x_cost_history=zeros(1,N);
z_cost_history=zeros(1,N);
constraint_cost_history=zeros(1,N);
for ii=1:1:N
    rho=1/ii;
    [x_kp1] = minimize_x(z_k,y_k,Qx,Qz,A,B,c,rho);
    [z_kp1] = minimize_z(x_kp1,y_k,Qx,Qz,A,B,c,rho);
    y_kp1=y_k + rho*(A*x_kp1+B*z_kp1-c);

    z_k=z_kp1;    y_k=y_kp1;

    x_cost_history(ii)=x_kp1'*Qx*x_kp1;
    z_cost_history(ii)=z_kp1'*Qz*z_kp1;
    constraint_cost_history(ii)=norm(A*x_kp1+B*z_kp1-c,2);
end
```

```
function [x_kp1] = minimize_x(z_k,y_k,Qx,Qz,A,B,c,rho)
% INPUTS z_k,y_k,Qx,Qz,A,B,c,rho
% OUTPUTS x_kp1
x=sdpvar(size(Qx,1),1);
z=z_k;
y=y_k;
Lp_xz_y=(x'*Qx*x)+(z'*Qz*z)+y'*(A*x+B*z-c)+(rho/2)*norm((A*x+B*z-c),2)^2;
% DIAGNOSTIC = OPTIMIZE(Constraint,Objective,options)
options = sdpsettings('solver','mosek');
DIAGNOSTIC = optimize([],Lp_xz_y,options);
x_kp1=value(x);
end
```

```
function [z_kp1] = minimize_x(x_kp1,y_k,Qx,Qz,A,B,c,rho)
% INPUTS xkp1,y_k,Qx,Qz,A,B,c,rho
% OUTPUTS z_kp1
z=sdpvar(size(Qz,1),1);
x=x_kp1;
y=y_k;
Lp_xz_y=(x'*Qx*x)+(z'*Qz*z)+y'*(A*x+B*z-c)+(rho/2)*norm((A*x+B*z-c),2)^2;
% DIAGNOSTIC = OPTIMIZE(Constraint,Objective,options)
options = sdpsettings('solver','mosek');
DIAGNOSTIC = optimize([],Lp_xz_y,options);
z_kp1=value(z);
end
```

```
%% PRINTING THE RESULTS
disp('==============================');
disp(['norm(x_real-x_kp1,2) : ',num2str(norm(x_real-x_kp1,2))]);
disp(['norm(z_real-z_kp1,2) : ',num2str(norm(z_real-z_kp1,2))]);
disp(['cost x_real:',num2str(x_real'*Qx*x_real)]);
disp(['cost z_real:',num2str(z_real'*Qz*z_real)]);
disp(['cost x:',num2str(x_kp1'*Qx*x_kp1)]);
disp(['cost z:',num2str(z_kp1'*Qz*z_kp1)]);
disp(['ineq cost:',num2str(norm(A*x_kp1+B*z_kp1-c,2))]);
disp('==============================');
fig_1=figure(1); fig_1.Color=[1,1,1];
plot(1:1:N,x_cost_history,'r.'); hold on;
plot(1:1:N,z_cost_history,'b.'); hold on;
plot(1:1:N,constraint_cost_history,'k.'); hold on;
xlabel('iter');
ylabel('cost values');
legend('cost(x)','cost(z)','cost(CONSTRAINT)');
fig_1.CurrentAxes.FontSize=15;
```