Penalty method (problema esempio)

```
global Q c A b eps;
%dati
Q = [ 1 0 ; 0 2 ] ; c = [ -3 ; -4 ] ;
A = [-2 1 ; 1 1 ; 0 -1 ]; b = [ 0 ; 4 ; 0 ];
%variabili del metodo
tau = 0.1 ; eps0 = 5 ; tolerance = 1e-6 ;
%inizializzazione
x = [0;0]; eps = eps0; iter = 0; SOL=[];
while true
   [x,pval] = fminunc(@p_eps,x);   %calcolo x in Pe
   infeas = max(A*x-b); %vedo quando è il ottenuto è piccolo Ax<=b
   SOL=[SOL;iter,eps,x',infeas,pval]; %addo nuova riga alla sol dove vedo
     %valori ottenuti nella nuova iterazione
   if infeas < tolerance
      break
   else
      eps = tau*eps;   iter = iter + 1 ;
   end
end
fprintf('\t iter \t eps \t x(1) \t x(2) \t max(Ax-b) \t pval \n');
SOL %print soluzione
function v= p_eps(x)  %penalized function
   global Q c A b eps;
   v = 0.5*x'*Q*x + c'*x ;
   for i = 1 : size(A,1)
      v = v + (1/eps)*(max(0,A(i,:)*x-b(i)))^2 ; %qui è al quadrato
   end
end
```

Exact Penalty method (problema esempio)

```
global Q c A b eps;
%% data
Q = [ 1 0 ; 0 2 ] ; c = [ -3 ; -4 ] ;
A = [-2 1 ; 1 1 ; 0 -1 ]; b = [ 0 ; 4 ; 0 ];
tau = 0.5 ;eps0 = 4 ; tolerance = 1e-6 ;
eps = eps0; x0 = [0;0]; iter = 0; SOL=[];
while true
   [x,pval] = fminunc(@p_eps,x0);
   infeas = max(A*x-b);
   SOL=[SOL;iter,eps,x',infeas,pval];
   if infeas < tolerance
      break
   else
      eps = tau*eps; iter = iter + 1 ;
   end
end
fprintf('\t iter \t eps \t x(1) \t x(2) \t max(Ax-b) \t pval \n');
SOL
function v= p_eps(x)  % exact penalty method
   global Q c A b eps;
   v = 0.5*x'*Q*x + c'*x ;
   for i = 1 : size(A,1)
      v = v + (1/eps)*(max(0,A(i,:)*x-b(i))) ;
```

```
   end
end
```

logarithmic barrier (problema esempio)

```
global Q c A b eps;
Q = [ 1 0 ; 0 2 ] ; c = [ -3 ; -4 ] ;
A = [-2 1 ; 1 1 ; 0 -1 ]; b = [ 0 ; 4 ; 0 ];
delta = 1e-3 ; tau = 0.5 ; eps1 = 1 ; x0 = [ 1 ; 1 ];
x = x0; eps = eps1 ; m = size(A,1) ; SOL=[];
while true
   [x,pval] = fminunc(@logbar,x);
   gap = m*eps;
   SOL=[SOL;eps,x',gap,pval];
   if gap < delta
      break
   else
      eps = eps*tau;
   end
end
fprintf('\t eps \t x(1) \t x(2) \t gap \t  pval \n\n');
SOL
function v = logbar(x) %% logarithmic barrier function
   global Q c A b eps
   v = 0.5*x'*Q*x + c'*x ;
   for i = 1 : length(b)
      v = v - eps*log(b(i)-A(i,:)*x) ;
   end
end
```