

Corrigé 14

Exercice 1

1.a) On définit le lagrangien $\forall \vec{x}, \vec{\lambda} \in \mathbb{R}^n$

$$\mathcal{L}(\vec{x}, \vec{\lambda}) = f(\vec{x}) - \sum_{i=1}^n \lambda_i (\vec{x} - \vec{c})_i$$

et on a

$$\partial_x \mathcal{L}(\vec{x}, \vec{\lambda}) = A\vec{x} - \vec{b} - \vec{\lambda}.$$

Les conditions KKT sont : $\exists \vec{\lambda}^* \in \mathbb{R}^n$ tel que

$$\begin{cases} A\vec{x}^* - \vec{b} - \vec{\lambda}^* &= 0, \\ \vec{x}^* - \vec{c} &\geq 0, \\ \vec{\lambda}^* &\geq 0, \\ (\vec{x}^* - \vec{c})_i \lambda_i^* &= 0 \quad i = 1, \dots, n. \end{cases}$$

On introduit une variable supplémentaire $\vec{s} \in \mathbb{R}^n$ tel que $\vec{x}^* - \vec{c} = \vec{s}$ et le problème devient : on cherche $\vec{x}^* \in \mathbb{R}^n$, $\vec{\lambda}^* \in \mathbb{R}^n$ et $\vec{s} \in \mathbb{R}^n$ tels que

$$\begin{cases} A\vec{x}^* - \vec{b} - \vec{\lambda}^* &= 0, \\ \vec{x}^* - \vec{c} - \vec{s} &= 0, \\ s_i \lambda_i^* &= 0 \quad i = 1, \dots, n, \\ \vec{\lambda}^* &\geq 0, \\ \vec{s} &\geq 0. \end{cases}$$

1.b) Le fichier est complété de la manière suivante :

```
function sol=intzero(n)
function [x_new,lambda_new,s_new]=quadip(n,eps)
% quadratic functional with inequality constraints solved with interior point method
% min 1/2 x^T A x - b^T x under the constraints x >= c
% the KKT conditions are (a slack variable s is added)
% Ax-b-lambda=0
% x-c-s=0
% s_i lambda_i = 0 i=1,...,n
% s>=0 and lambda>=0
% Newton method is used for the 3 first conditions
A = (n+1)*(n+1)*(sparse(2:n,1:n-1,-1,n,n)
    + sparse(1:n,1:n,2,n,n) + sparse(1:n-1,2:n,-1,n,n));
b = -ones(n,1);
c = -0.05*ones(n,1);
x_old=zeros(n,1);
s_old=max(eps,ones(n,1));
lambda_old=max(eps,ones(n,1));
for iter=1:100
    mat1 = horzcat(A,-speye(n),sparse(n,n));
    mat2 = horzcat(speye(n),sparse(n,n),-speye(n));
    mat3 = horzcat(sparse(n,n),sparse(1:n,1:n,s_old,n,n),sparse(1:n,1:n,lambda_old,n,n));
    mat = vertcat(mat1,mat2,mat3);
    rhs1 = A*x_old-b-lambda_old;
    rhs2 = x_old-c-s_old;
    rhs3 = sparse(n,1);
    for i=1:n
        rhs3(i)=lambda_old(i)*s_old(i);
    end
    rhs = vertcat(rhs1,rhs2,rhs3);
    sol=mat\rhs;
    x_new=x_old-sol(1:n);
    plot(x_new);pause();
    lambda_new=max(eps,lambda_old-sol(n+1:2*n));
    s_new=max(eps,s_old-sol(2*n+1:3*n));
    discrep=norm(x_new-x_old)/norm(x_new);
    printf('iter: %d Discrepancy: %f \n',iter,discrep);
    x_old=x_new;
```

```

lambda_old=lambda_new;
s_old=s_new;
if (discrep<0.0001)
    break
end
end
end
end

```

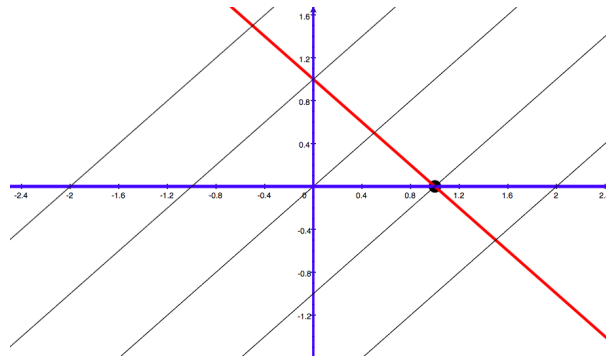
1.c) On obtient les résultats suivants :

n	ϵ	nombre d'itérations	n	ϵ	nombre d'itérations
19	0.01	21	9	0.0001	7
19	0.001	12	19	0.0001	9
19	0.0001	9	39	0.0001	9
			79	0.0001	8
			159	0.0001	8

On observe que le nombre d'itérations dépend pas de n .

Exercice 2

2.a) L'image suivante représente le cas $n = 2$. La ligne rouge représente la contrainte $x_1 + x_2 = 1$ et les lignes noires représentent les droites $-x_1 + x_2 = cste$. En vert on voit le minimum cherché : $x_1 = 1, x_2 = 0$.



Pour n quelconque la solution est $x_1 = 1, x_2 = \dots = x_n = 0$.

2.b) Soit $f(\vec{x}) = \vec{c}^T \vec{x}$ avec $\vec{c}^T = [-1, 1, \dots, 1]$. On définit le lagrangien $\forall \vec{\lambda}^* \in \mathbb{R}^n, \mu \in \mathbb{R}$

$$\mathcal{L}(\vec{x}, \vec{\lambda}, \mu) = f(\vec{x}) - \sum_{i=1}^n x_i \lambda_i - \mu(x_1 + \dots + x_n - 1).$$

On obtient

$$\partial_x \mathcal{L}(\vec{x}, \vec{\lambda}, \mu) = \vec{c} - \vec{\lambda} - \mu \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

et les conditions KKT sont : $\exists \vec{\lambda}^* \in \mathbb{R}^n, \exists \mu^* \in \mathbb{R}$ tels que

$$\begin{cases} \vec{c} - \vec{\lambda}^* - \mu^* \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = 0, \\ x_1^* + \dots + x_n^* - 1 = 0, \\ x_i^* \lambda_i^* = 0 \quad i = 1, \dots, n, \\ \vec{x}^* \geq 0, \\ \vec{\lambda}^* \geq 0. \end{cases}$$

2.c) Le fichier est complété de la manière suivante :

```

function x_new=linprogip(n,eps)
% linear programming solved with interior point method
% min c^T x under the constraints A x = b and x>=0
% the KKT conditions are
%   c-A^T mu -lambda = 0
%   A x -b = 0
%   x_i lambda_i = 0 i=1,...,n
%   x>=0 and lambda>=0
% Newton method is used for the 3 first conditions
p=1;
A=ones(p,n);
b=sparse(p,1);
b(1)=1;
c=ones(n,1);
c(1)=-1;
x_old=max(eps,zeros(n,1));
lambda_old=max(eps,zeros(n,1));
mu_old=zeros(p,1);
for iter=1:10
    mat1 = horzcat(sparse(n,n),-speye(n,n),-A');
    mat2 = horzcat(A,sparse(p,n),sparse(p,p));
    mat3 = horzcat(sparse(1:n,1:n,lambda_old,n,n),sparse(1:n,1:n,x_old,n,n),sparse(n,p));
    mat = vertcat(mat1,mat2,mat3);
    rhs1 = c-A'*mu_old-lambda_old;
    rhs2 = A*x_old-b;
    rhs3 = sparse(n,1);
    for i=1:n
        rhs3(i)=x_old(i)*lambda_old(i);
    end
    rhs = vertcat(rhs1,rhs2,rhs3);
    sol=mat\rhs;
    x_new=max(eps,x_old-sol(1:n));
    lambda_new=max(eps,lambda_old-sol(n+1:2*n));
    mu_new=mu_old-sol(2*n+1:2*n+p);
    discrep=norm(x_new-x_old)/norm(x_new);
    printf ("iter: %d Discrepancy: %f \n",iter,discrep);
    x_old=x_new;
    lambda_old=lambda_new;
    mu_old=mu_new;
    if (discrep<0.001)
        break
    end
end
end

```

2.d) On obtient les résultats suivants :

n	ϵ	nombre d'itérations	n	ϵ	nombre d'itérations
19	0.01	5	9	0.0001	4
19	0.001	4	19	0.0001	4
19	0.0001	4	39	0.0001	4
			79	0.0001	4
			159	0.0001	4

On observe que le nombre d'itérations dépend pas de n .