DURET Guillaume
BROUSSE Léa
Groupe A

CPE Lyon – 3ETI

M-ALG : TP 5

Formes quadratiques et moindres carrés :

2017 – 2018

Exercice 2 : Méthode des moindres carrés.

Cet exercice nous propose de traiter le problème des moindres carrés pour résoudre un système d'équation linéaire. Dans un premier temps nous appliquerons cette méthode à la loi de Hooke. Nous étudierons ensuite la droite des moindres carrés.

1) Loi de Hooke.

Le problème consiste à déterminer la constante k reliant l'élongation d'un ressort à la force à laquelle il est soumis tel que F=ky.

Nous disposons des différentes valeurs de F (en N) et de y (en cm) et nous cherchons alors à déterminer k* de manière à minimiser la distance entre y et F.

a) Cette quantité à minimiser est la suivante :

$$\|F - ky\|^2 = \sum_{i=1}^{3} (F_i - kY_i)^2$$

b) Pour cela nous déterminons l'équation normale de la forme $A^T A k^* = A^T Y$

Ici, A=Y et Y=F. Ce qui donne l'équation suivante :

$$Y^T Y k^* = Y^T F \quad (1)$$

c) La résolution de l'équation (1) nous permet de déterminer k^* d'après les données du problème : $F = \begin{pmatrix} 3 \\ 5 \\ 8 \end{pmatrix}$ et $Y = \begin{pmatrix} 4 \\ 7 \\ 11 \end{pmatrix}$

F=kY $\Leftrightarrow Y^T Y k^* = Y^T F$

$$\Leftrightarrow (4 \quad 7 \quad 11) \begin{pmatrix} 3 \\ 5 \\ 8 \end{pmatrix} = (4 \quad 7 \quad 11) \begin{pmatrix} 4 \\ 7 \\ 11 \end{pmatrix} k^*$$

$$\Leftrightarrow 135 = 186\, k^*$$

$$\Leftrightarrow \ k^* = 0726\, N.m^{-1}$$

. Ainsi, nous avons déterminé la valeur de k^* de la constante k par la méthode des moins carrés. Cette méthode nous as donc permis de trouver l'approximation de la valeur de k en dépit des erreurs de mesure.

2)

Soit $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ un ensemble de N points

On souhaite approcher (a, b) tel que y=ax+b en minimisant $\sum_{i=1}^{N} (ax_i + b - \ y_i)^{\,2}$

On pose les matrices A = $\begin{pmatrix} 1 & x1 \\ \vdots & \vdots \\ 1 & xn \end{pmatrix}$, C=$\begin{pmatrix} b \\ a \end{pmatrix}$ et Y=$\begin{pmatrix} y1 \\ \vdots \\ yn \end{pmatrix}$

Ce qui permet donc d'avoir $||AC - Y||^2 = \sum_{i=1}^{N}(ax_i + b - y_i)^2$

Pour minimiser cette somme on utilise les équation normales $A^T AC = A^T Y$ avec C minimisant $||AC - Y||^2$ donc $\sum_{i=1}^{N}(ax_i + b - y_i)^2$
Ces équations sont équivalentes à

$$\begin{pmatrix} N & \sum_{i=1}^{N} x_i) \\ \sum_{i=1}^{N} x_i) & \sum_{i=1}^{N} x_i{}^2) \end{pmatrix} \begin{pmatrix} b \\ a \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N} y_i) \\ \sum_{i=1}^{N} x_i\, y_i) \end{pmatrix}$$

Il suffit donc de les résoudre afin d'obtenir la droite des moindres carrés $y = a^*x + b^*$

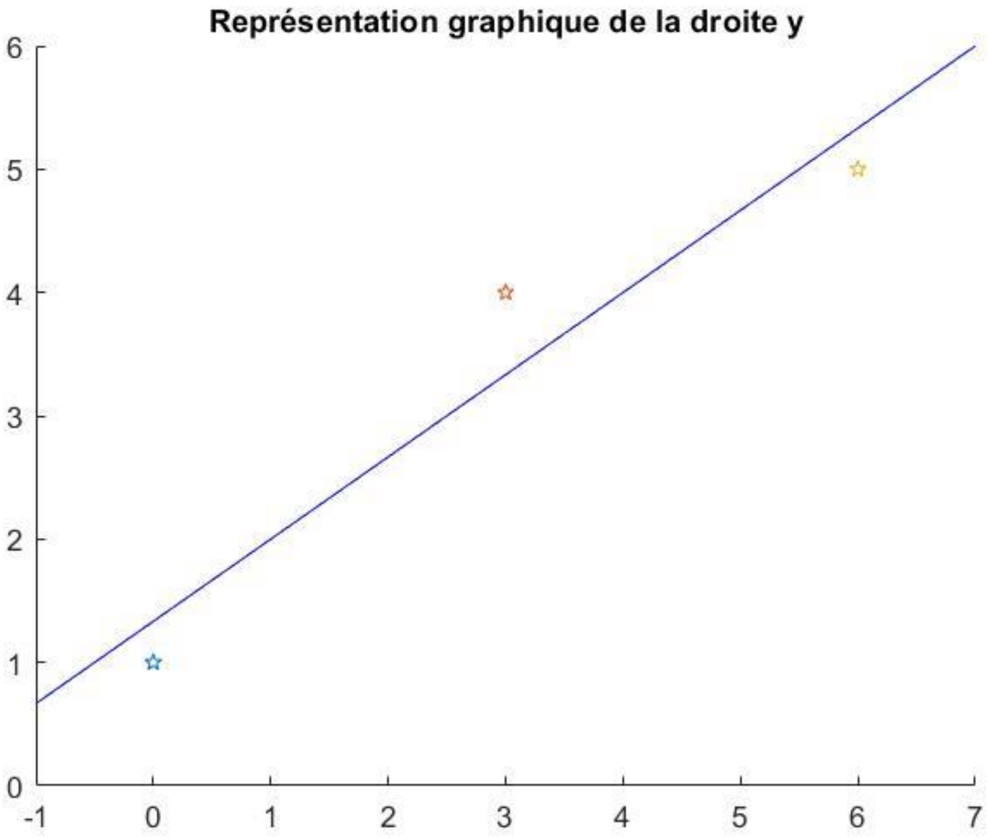
<u>Application :</u>

On applique donc cette méthode pour les trois point (0, 1), (3, 4) et (6, 5)

On pose donc les matrice A et Y et on utilise les équations normales pour déterminer C et la droite des moindres carrés

```
clear all; close all;

A=[1,0;1,3;1,6];
Y=[1;4;5];
M=A'*A;
C=(M^(-1))*(A'*Y);

x=-1:0.01:7;
y=C(2,1).*x + C(1,1); % y=ax+b

figure(1)
hold on;
plot(0,1,'p')
plot(3,4,'p')
plot(6,5,'p')
plot(x,y,'b')
axis ([-1 7 0 6])
title('Représentation graphique de la droite y');
```

Cela qui nous permet d'obtenir :



3)

On généralise ici la méthode car cette fois on ne cherche pas y forcement affine.

En effet soit $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ un ensemble de N points on veut $y = \varphi(x)$
Avec $\varphi = c_1\varphi_1 + c_2\varphi_2 + \cdots + c_p\varphi_p$ avec $\{\varphi_1, \varphi_2, \dots, \varphi_p\}$ une famille de fonction linéairement indépendante donnée.

Ici on veut donc minimiser $\sum_{i=1}^{N}(\varphi(x_i) - y_i)^2$

Pour cela on pose les matrices : A = $\begin{pmatrix} \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{pmatrix} = \varphi$, C=$\begin{pmatrix} c1 \\ \vdots \\ cn \end{pmatrix}$ et Y=$\begin{pmatrix} y1 \\ \vdots \\ yn \end{pmatrix}$

Ce qui permet donc d'avoir $||AC - Y||^2 = \sum_{i=1}^{N}(\varphi(x_i) - y_i)^2$

On utilise ensuite les équation normales $A^T AC = A^T Y$ pour minimiser la somme et d'obtenir φ s'approchant au mieux des différents points

<u>Application :</u>

On applique cette méthode pour les points (0, 3), (1, 2), (2, 4) et (3, 4)
avec $\varphi_1(x) = 1, \varphi_2(x) = x, \varphi_3(x) = x^2$ et p=3

On crée donc les matrices A et Y affin d'utiliser les équations normales $A^T AC = A^T Y$

```
A2=[1,0,0;1,1,1;1,2,4;1,3,9];

Y2=[3;2;4;4];

M2=A2'*A2 % A'*A
M3=A2'*Y2 % A'*Y
```

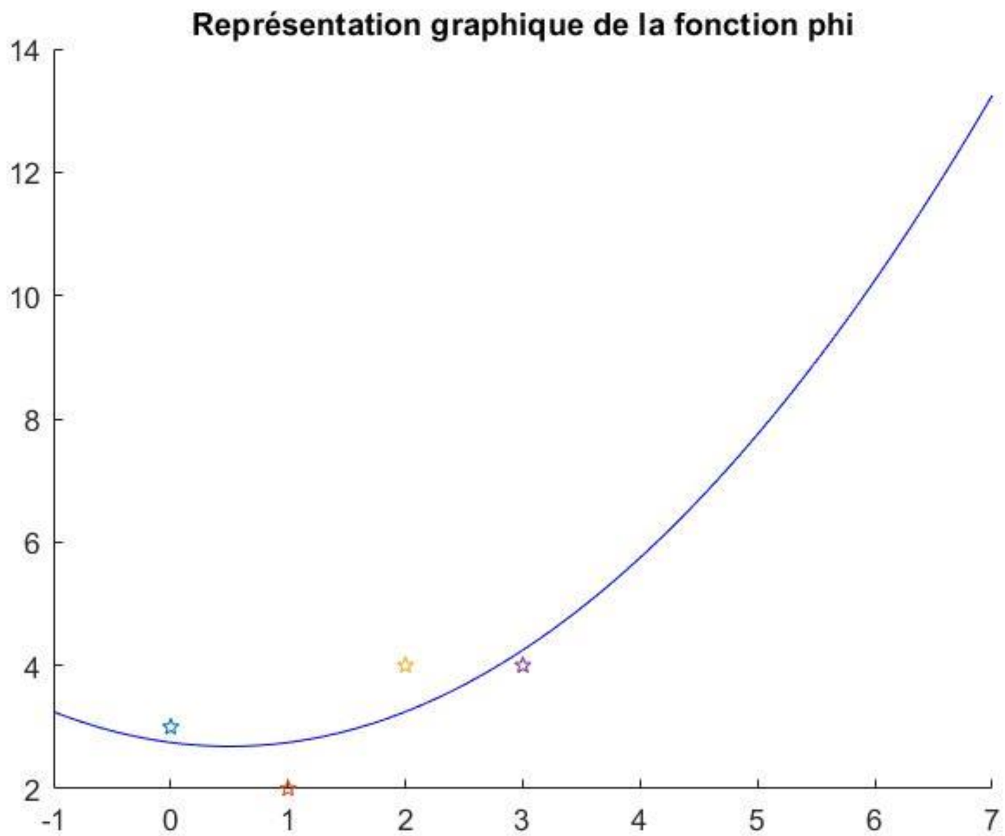
```
M2 =

     4     6    14
     6    14    36
    14    36    98

M3 =

    13
    22
    54
```

Donc résoudre les équations normales est équivalent à résoudre le système :

$$\begin{cases} 4c_1 + 6c_2 + 14c_3 = 13 \\ 6c_1 + 14c_2 + 36c_3 = 22 \\ 14c_1 + 36c_2 + 98c_3 = 54 \end{cases}$$

On résout donc ce système et obtenons finalement la fonction φ

```
C2=(M2^(-1))*(A2'*Y2);

phi=C2(1,1)*1+C2(2,1).*x+C2(3,1).*(x.^2) ;

figure(2)

hold on;
plot(0,3,'p')
plot(1,2,'p')
plot(2,4,'p')
plot(3,4,'p')
plot(x,phi,'b');
title('Représentation graphique de la fonction phi');
```



Exercice n°5 : équation de la chaleur

On cherche dans cet exercice à déterminer la température aux cours du temps et la températures final (stable) d'une plaque modélisée par une grille n x n.

Pour cela on numérote les points de la grille de 1 à n² en balayant de gauche à droite et de haut en bas.

On crée ensuite la matrice d'incidence M qui correspond en fait à la matrice du système de n² équations et n² inconnues : $T(i) = \frac{\Sigma \, Temperature \; des \; points \; voisins}{Nombre \; de \; points \; voisins=4}$

On remarque en effet que les coefficients de la matrice d'incidence M_{ij} correspondent à ¼ quand la case de la grille i est voisin de la case j.

On construit donc cette matrice par cette méthode :

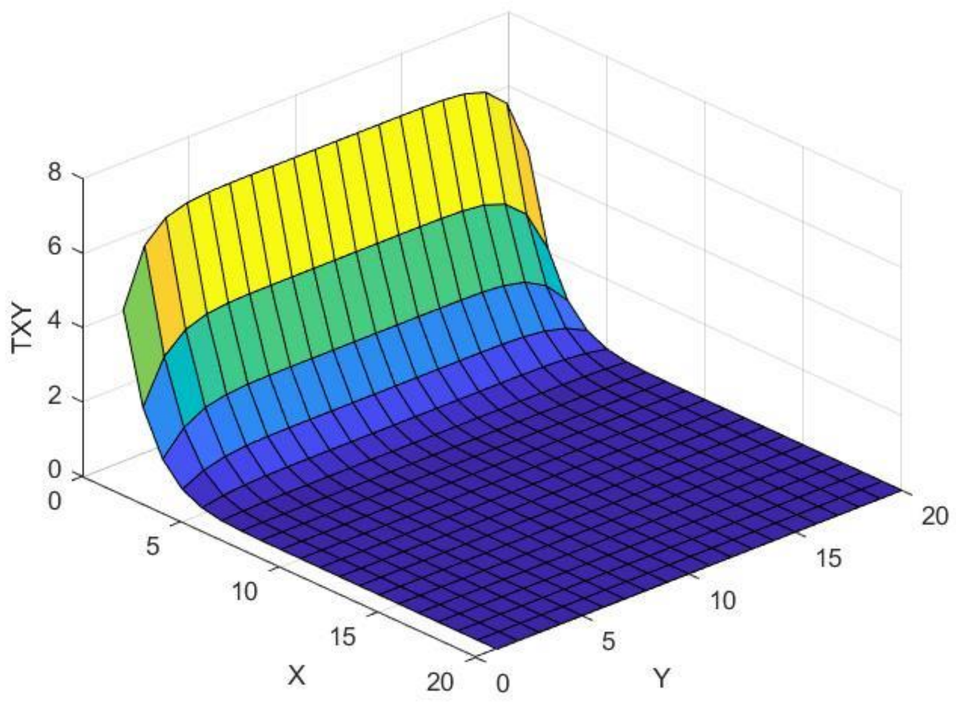
```
clear all;clc; close all;
n=20;
A=zeros(n,n);

for i=1:n
    for j=1:n
        A(i,j)=n*(i-1)+j; % on constitue la matrice auxiliaire qui numerote les cases
    end
end

M=zeros(n^2,n^2);
for i=1:n-1 % Pour tout sous block m/n on fait M_mn = 1 et M_nm = 1

    for j=1:n
        m=A(i,j);
        n2=A(i+1,j);
        M(m,n2)=1;
        M(n2,m)=1;
    end
end
for i=1:n % Pour tout sous block m n on fait M_mn = 1 et M_nm = 1
    for j=1:n-1
        m2=A(i,j);
        n3=A(i,j+1);
        M(m2,n3)=1;
        M(n3,m2)=1;
    end
end

M2=0.25*M; % matrice d'incidence
```

Ayant la matrice d’incidence, on crée la matrice B qui caractérise les températures extérieures fixe et avançons dans le temps jusqu’au régime permanant.

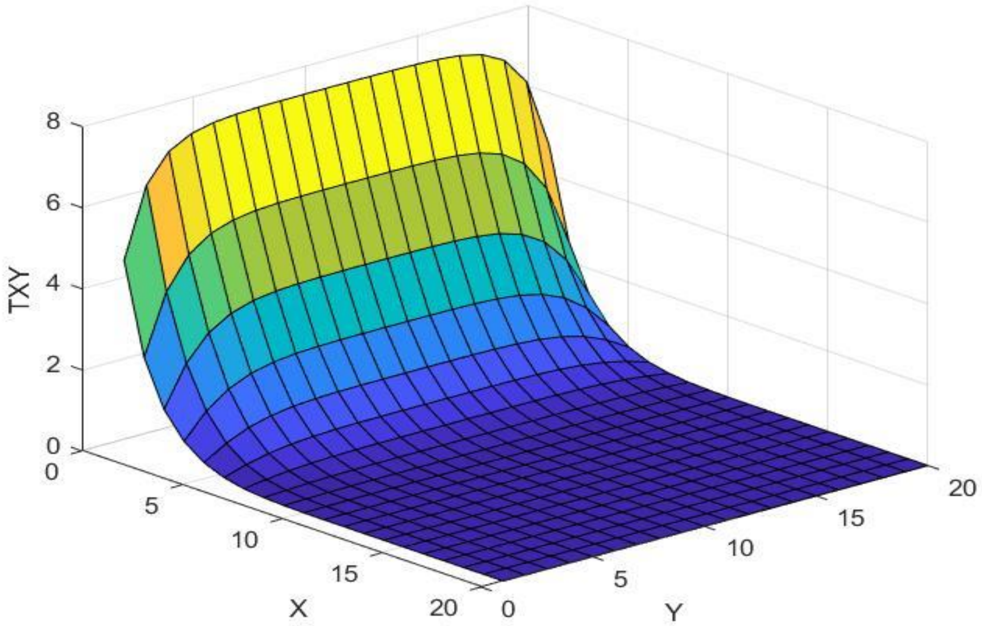
En effet on a $T_{n+1} = M * T_n + B$

```
B=zeros(n^2,1);

B(1:n,1)=10/4;

T=zeros(n^2,1);

for k=0:5
    T=M2*T+B;
end

TXY=reshape(T,n,n);

[X,Y]=meshgrid(1:n);

surf(X,Y,TXY)
xlabel('X')
ylabel('Y')
zlabel('TXY')
```

Pour N=10 iterations :



Pour N=20 iterations :



Pour N=20 la température de la plaque à attend son régime permanent.

<u>Remarque :</u> On aurait pu aussi déterminer le régime permanant en résolvant $T \quad = M * T \quad + B$ ce qui nous aurai fait obtenir un point fixe.
De plus par la méthode de résolution avec un point fixe d'une suite arithmético-géométrique on peut facilement obtenir la température directement en fonction du temps.

<u>Exercice 6 :</u> Systèmes d'équations différentielles dans $\mathbb{R}$

Tout d'abord, nous commençons pat effectuer un travail préliminaire sur les exponentielles de matrice pour notamment déterminer les fonctions Matlab correctement associées.

1) Préliminaire sur les exponentielles de matrice

a) Dans un premier temps, nous vérifions la différence en les fonctions `exp()` et `expm()`. Pour cela nous prenons une matrice A quelconque de dimension 2x2.

```
A=[2,3;4,6];
exp(A)
 ans =

     7.3891   20.0855
    54.5982  403.4288

expm(A)
  ans =
     1.0e+03 *

      0.7460    1.1175
      1.4900    2.2360
```

La fonction exp() applique la fonction exponentielle à chaque composant de la matrice A, tandis que la fonction expm() applique la fonction exponentielle à la matrice A.

b) Nous allons maintenant vérifier que cette même fonction expm (A) coïncide avec $\sum_{n=0}^{\infty} \frac{1}{n!} A^n$ (1)

Pour cela, on construit la matrice correspondante à la relation (1) par une boucle for.

```
ex=zeros(2,2) ;
for k=0 :100
    Ex=ex+(1/factorial(k))*A^k ;
end
ex
  ex =
     1.0e+03 *

      0.7460    1.1175
      1.4900    2.2360
```

Le résultat obtenu est identique à celui de expm(A).

c) Cette fois, nous allons vérifier que exp(A*B) n'est pas égal à expm(A)*expm(B), à condition que les matrices A et B ne commutent pas.

Nous prenons :

A=[1 1 ; 8 6] ;

B= [5 9 ; 18 18]

```
expm(A).*expm(B)
```

```
ans =
   1.0e+13 *

    0.1769    0.2034
    2.3507    2.7134
```

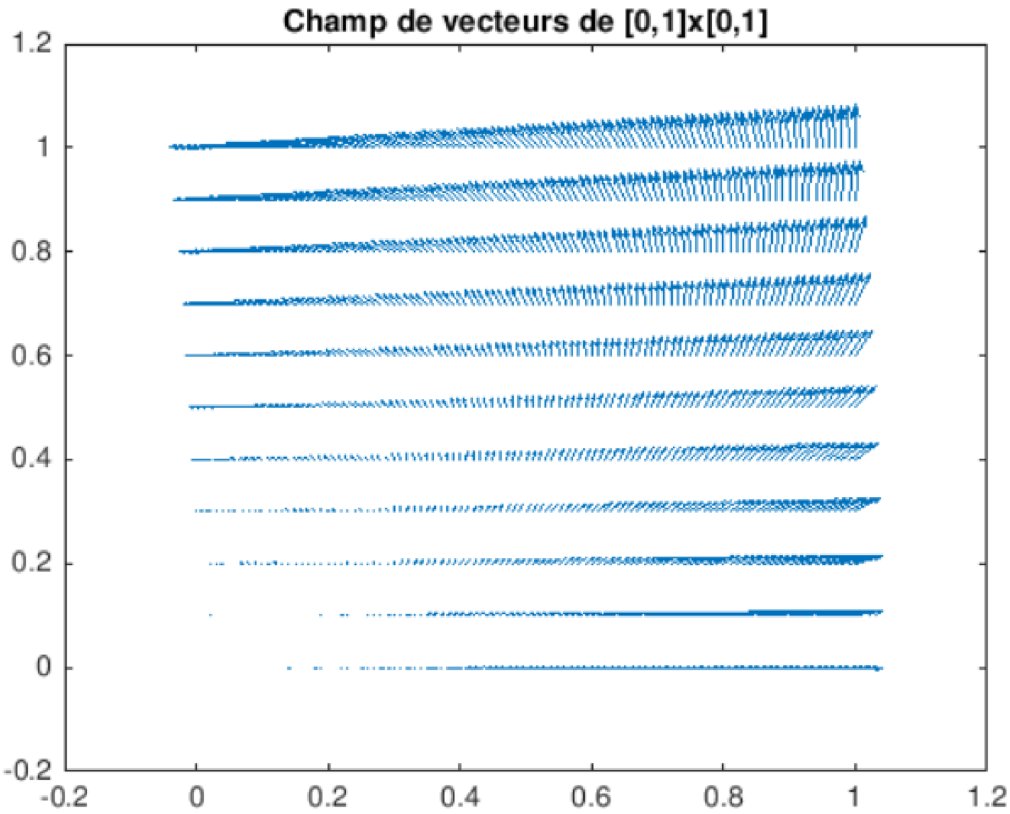
```
expm(A*B)
```

```
ans =
   1.0e+85 *

    0.6681    1.0176
    4.4475    6.7740
```

2) Nous allons tracer au point (x,y) le champ de vecteurs de [0,1]x[0,1] qui associe le vecteur $(u,v)=(x^2-y^2, 2xy)$.

```
[x,y]=meshgrid(0:0.01:1,0:0.1:1);
u=x.^2-y.^2;
v=2.*x.*y;
figure (1)
quiver(x,y,u,v) %affichage du champ de vecteurs
title ('Champ de vecteurs de [0,1]x[0,1]')
```

Nous obtenons la figure suivante :

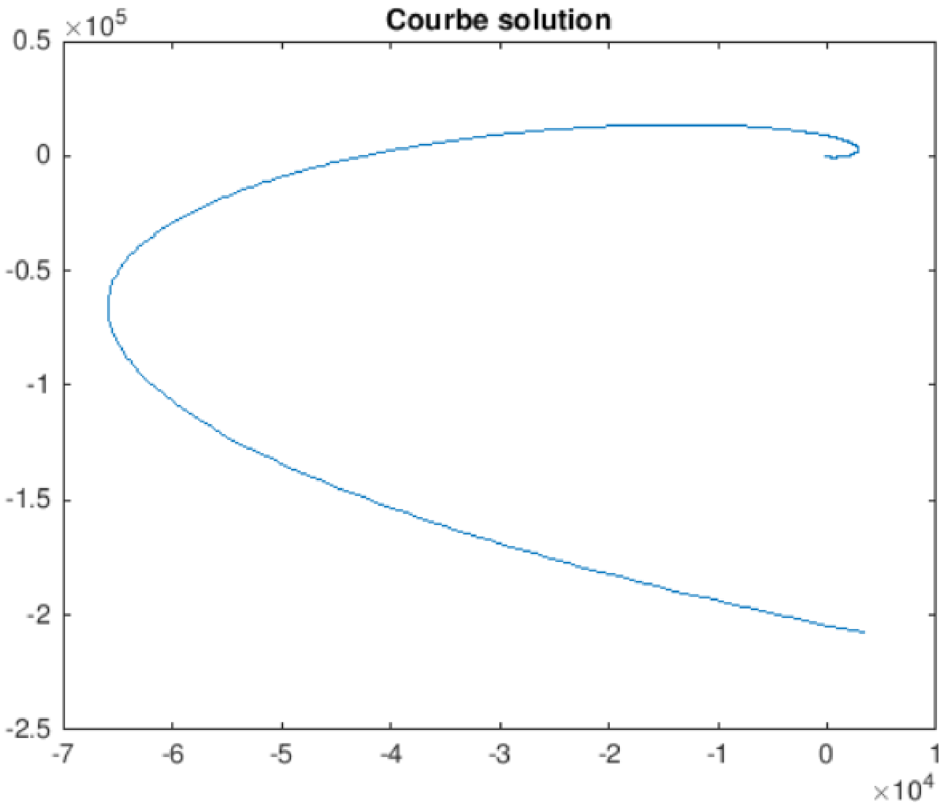


Cependant, il semblerait que notre champ de vecteur devrait être tourné de 90° par rapport à ce qu'il est.

3) Nous allons afficher la courbe solution de système en suivant la méthode décrite dans l'énoncé. Nous choisissons comme conditions initiales le vecteurs X0=[5 ;8].

```
X0=[5;8]; %Conditions initiales
M = [1,-1;1,1];
X=X0;
for t = 0:0.01:10
     Xt = expm(t*M)*X0;
      X = [X, Xt];
end
figure(1);
plot(X(1,:), X(2,:)); %Affichage (x(t),y(t))
title ('Courbe solution et champ de vecteurs')
```

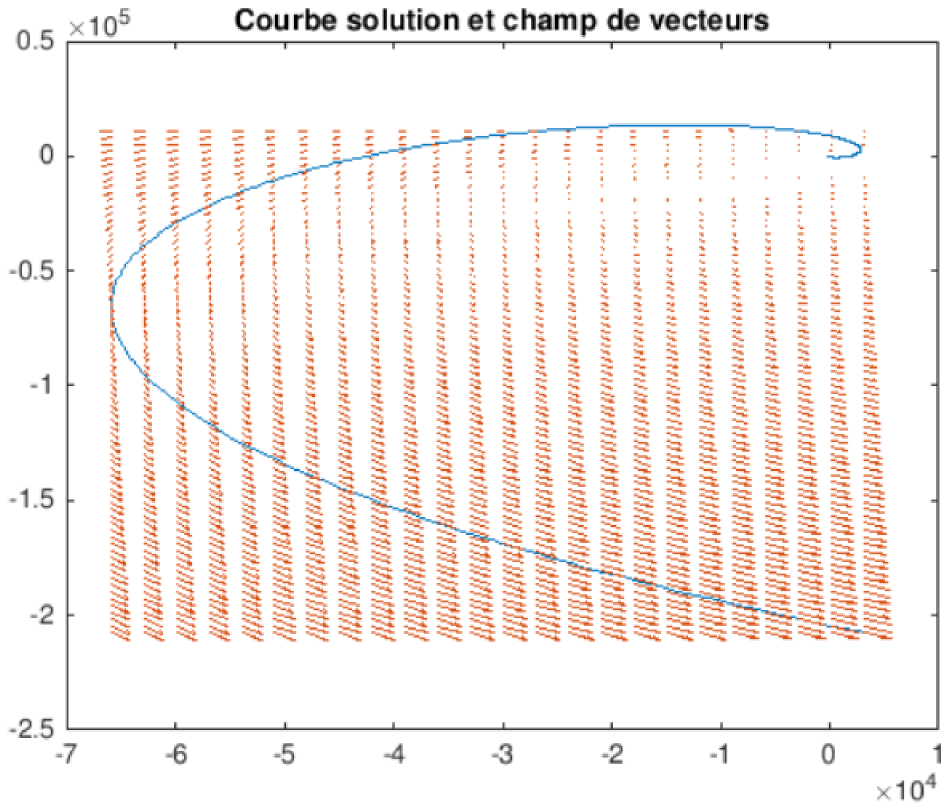
Nous obtenons la courbe suivante :



b) Nous recopions le script donné dans l'énoncé en le complétant que nous ajoutons à la suite du script précédent.

```
xmin=min(X(1,:))-0.5;
xmax=max(X(1,:))+0.5;
ymin=min(X(2,:))-0.5;
ymax=max(X(2,:))+0.5;
step=3000;
x=xmin:step:xmax;
y=ymin:step:ymax;
u=zeros(length(x),length(y));
v=zeros(length(x),length(y));

for i=1:length(x)
    for j=1:length(y)
        Xij=[x(i);y(j)];
        V=M*Xij;
        u(i,j)=V(1);
        v(i,j)=V(2);
    end
end
hold on;
[a,b]=meshgrid(x,y);
quiver(a,b,u',v'); %Affichage du champ des vecteurs
```

Nous obtenons la figure suivante :



Le champ de vecteurs cette fois-ci est dans le bon sens, contrairement à la figure obtenue à la question 2).

Nous avons modifié la valeur du `step` qui était donnée puisqu'elle ne plaisait pas à Matlab et exécuter le code trop lentement.