

LM206 - Initiation à Scilab

Contrôle Continu (1h30)

Les documents sont admis. Un seul fichier `.sci` (or `.sce`) doit être rendu. Tout fichier rendu doit porter nom et prénom de l'étudiant. Tout exercice doit être séparé des autres et toute réponse commentée. Des résultats doivent être transcrits sur papier. La présentation et la lisibilité entreront pour une part importante dans l'appréciation des copies.

Exercice 1. Construire, sans utiliser de boucle `for`, une fonction prenant en arguments d'entrée, les entiers N et M , et qui renvoie la matrice A de taille $N \times M$ dont l'élément de la ligne i et colonne j est donné par :

$$A_{ij} = (2i - 5) \times (j^3 + 9).$$

Écrire sur votre feuille la matrice obtenue pour $N = 6$ et $M = 4$.

Exercice 2. Cet exercice comporte deux questions.

1. Écrire la fonction `binome`, qui prend en arguments d'entrée un entier n et qui renvoie, sous forme d'un vecteur de taille $n + 1$, les coefficients $C_n^0, C_n^1, \dots, C_n^n$. Le coefficient C_n^k , pour $0 \leq k \leq n$, étant donné par :

$$C_n^k = \binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Remarque : les coefficients C_n^k sont bien à calculer pour k allant de 0 à n , soit $n + 1$ valeurs à stocker. Donc attention aux indices pour parcourir le vecteur stockant ces coefficients.

2. À l'aide de la fonction `binome` de la question précédente, écrire la fonction `newton`, qui prend en arguments d'entrée deux réels x et y , un entier n et qui renvoie la valeur $(x + y)^n$ donnée par :

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k$$

Calculer cette valeur, puis écrire le résultat sur votre feuille, pour

- (a) $x = 4, y = 2, n = 6$
- (b) $x = 9, y = 15, n = 3$

Exercice 3. On veut représenter la fonction $f : (x, y) \mapsto f(x, y) = \sin(\frac{x}{2}\pi) \cos((y - 1)\pi)$ sur $[0, 2] \times [1, 2]$. On définit le maillage

$$N=100; M=60; x=linspace(0,2,N); y=linspace(1,2,M)$$

1. Définir une matrice Z de taille $N \times M$ dont la composante (i, j) est égale à $f(x(i), y(j))$.
2. Utiliser la fonction `plot3d` pour écrire une instruction permettant de visualiser les résultats.

Exercice 4. On considère le fichier texte `donneesTPnote`, à télécharger depuis la page

<http://cindy-guichard.toile-libre.org/donnesTPnote>

Ce fichier contient

- le nombre de lignes N et de colonnes M d'une matrice A ,
- tous les éléments de la matrice A .

Donner les instructions Scilab permettant de lire le fichier `donneesTPnote` puis celles permettant d'écrire dans un fichier, nommé `retourTPnote`, les données suivantes :

- le nombre de lignes et de colonnes de la matrice A^{-1} (A^{-1} étant la matrice inverse de A),
- tous les éléments de la matrice A^{-1} .

Exercice 5. On considère l'évolution d'un vecteur colonne $u = (u_i)_{1 \leq i \leq 3}$ de taille 3 entre les temps n et $n + 1$ qui est donné par le système

$$\begin{cases} u_1^{(n+1)} = a_1 u_1^{(n)} + a_2 u_2^{(n)} + a_3 u_3^{(n)} + a_4 \\ u_2^{(n+1)} = b_1 u_1^{(n)} + b_2 u_2^{(n)} + b_3 u_3^{(n)} + b_4 \\ u_3^{(n+1)} = c_1 u_1^{(n)} + c_2 u_2^{(n)} + c_3 u_3^{(n)} + c_4 \end{cases} \quad (1)$$

où les vecteurs $a = (a_i)_{1 \leq i \leq 4}$, $b = (b_i)_{1 \leq i \leq 4}$, $c = (c_i)_{1 \leq i \leq 4}$, et le vecteur initial $u^{(0)} = (u_i^{(0)})_{1 \leq i \leq 3}$ sont des données du problème.

1. Écrire sur votre feuille le système (1) sous forme matricielle $u^{(n+1)} = Au^{(n)} + r$ où A est une matrice de taille 3×3 et r un vecteur colonne de taille 3.
2. Construire la fonction Scilab (nommée `linearExo5`) qui prend en arguments d'entrée les vecteurs de données a , b , c et qui renvoie la matrice A et le vecteur r .
3. En utilisant la question précédente, coder la fonction Scilab (nommée `iterExo5`) qui prend en arguments d'entrée un entier $m > 0$, les vecteurs de données a , b , c , $u^{(0)}$ et qui renvoie le vecteur $u^{(m)}$ de taille 3.
4. Écrire sur votre feuille le vecteur $u^{(m)}$ pour
 - $m = 13$
 - $a = (1, 0.1, 3, -1)$
 - $b = (0.4, 0.25, 0, 1.2)$
 - $c = (0.7, 0, 3.1, -2)$
 - $u^{(0)} = (1, 2, 3)^t$

LM206 - Initiation à Scilab

Contrôle Continu - Corrigé

Exercice 1.

```
function A =exo1 (N,M)
// on cree une matrice qui a N lignes (qui correspond a la longueur de x
// transpose) et M colonnes (longueur de y) qui est le produit du vecteur
// colonne x avec le vecteur ligne y
    i=1:N;
    j=1:M;
    x=2.*i-5;
    y=j.^3+9;
    A=x'*y
endfunction
```

```
N=6;
M=4;
A=exo1(N,M)
```

```
A =
- 30.  - 51.  - 108.  - 219.
- 10.  - 17.  - 36.  - 73.
 10.   17.   36.   73.
 30.   51.   108.  219.
 50.   85.   180.  365.
 70.  119.  252.  511.
```

Exercice 2.

1. // La sortie de cette fonction est le vecteur C qui contient les coefficients

```
function C = binome(n)
    if n>=0 & int(n)==n then
        C = zeros(1,n+1);
        for k=0:n
            // C_n^k = n! / ( k! * (n-k) !)
            C(k+1) = factorial(n)/ (factorial(k)*factorial(n-k));
        end
    else
        disp('n doit etre un entier positif ou nul');
    end
endfunction
```

```
// Si on connait pas la fonction factorial(i) on peut ecrire la fonction
// binome de la facon suivante
```

```
function C = binome_v2(n)
    if n>=0 & int(n)==n then
        C = zeros(1,n+1);
        c_k = ones(1,n+1); // -> factoriel(0) a factoriel(n)
        for k=1:n
            c_k(k+1) = c_k(k)*(k);
        end
        for k=0:n
            // C_n^k = n! / ( k! * (n-k) !)
            C(k+1) = c_k(n+1)/ (c_k(k+1)*c_k(n-k+1));
        end
    else
        disp('n doit etre un entier positif ou nul');
    end
endfunction
```

```
// test
n = 5;
C = binome(n)
    1. 5. 10. 10. 5. 1.
C2 = binome_v2(n)
    1. 5. 10. 10. 5. 1.
```

```
2. function r = newton (x,y,n)
    // On calcul d'abord les coefficients
    C = binome(n);
    r = 0;
    for k=0:n
        r = r + C(k+1)*x^(n-k)*y^k;
    end
endfunction
```

- (a) x=4;y=2;n=6;
rNewton = newton (x,y,n)
46656.
// on peut pour verifier le resultat
rScilab = (x+y)^n
46656.
- (b) x=9;y=15;n=3;
rNewton = newton (x,y,n)
13824.
// on peut pour verifier le resultat

```
rScilab = (x+y)^n
13824.
```

Exercice 3.

```
function [f] = f(x,y)
    f = sin(1/2*x.*%pi).*cos((y-1).*%pi);
endfunction

N = 100; M = 60;
x = linspace(0,2,N);
y = linspace(1,2,M);

// On construit les matrices X et Y pour nous aider
X = x'*ones(y);
Y = ones(x')*y;

// La matrice Z est egale a la fonction f evaluee en X,Y (les matrices)
Z = f(X,Y);

// On utilise la commande plot3d en lui donnant les trois matrices
plot3d(X,Y,Z)
```

Exercice 4.

```
// On ouvre le fichier des donnees
fileDonnes=file('open','donneesTPnote','old');
// On lit le nombre de lignes et de colonnes de A
dim=read(fileDonnes,1,2) ;
// et les valeurs de A
A=read(fileDonnes,dim(1),dim(2)) ;
// On ferme le fichier
file('close',fileDonnes) ;
// On a donc
A =
    2. 3. 1. 5.
    0. 0. 1. 3.
    1. 4. 0. 2.
    1. 1. 5. 3.
// On calcule la matrice inverse de A
invA = inv(A)
// On ecrit le resultat dans le fichier retourTPnote
// On ouvre le fichier
fileRetour=file('open','retourTPnote') ;
```

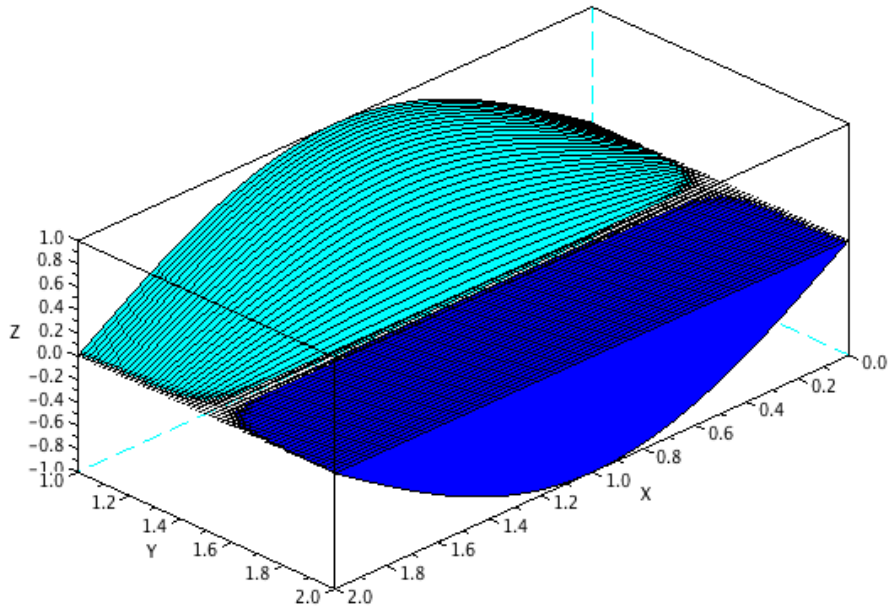


Figure 1: Ex. 3.2

```
// On ecrit d'abord la taille de la matrice inverse
write(fileRetour,size(invA));
// et ensuite la matrice inverse
write(fileRetour,invA) ;
// On ferme le fichier
file('close',fileRetour) ;
```

Fichier retourTPnote :

```
4 4
0.78125 - 0.9375 - 0.59375 0.03125
- 0.21875 0.0625 0.40625 0.03125
- 0.140625 - 0.03125 0.046875 0.234375
0.046875 0.34375 - 0.015625 - 0.078125
```

Exercice 5.

1. On a $u^{(n+1)} = Au^{(n)} + r$ avec $A = \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix}$ et $r = \begin{pmatrix} a_4 \\ b_4 \\ c_4 \end{pmatrix}$.

```

2. function [A,r] = linearExo5 (a,b,c)
    A = zeros(3,3) ; r = zeros(3,1) ;
    A(1,:) = a(1:3) ; r(1) = a(4) ;
    A(2,:) = b(1:3) ; r(2) = b(4) ;
    A(3,:) = c(1:3) ; r(3) = c(4) ;
endfunction

```

```

3. function [u] = iteExo5 (a,b,c,u0,m)
    [A,r] = linearExo5 (a,b,c)
    u = u0 ;
    for i=1:m
        u = A*u + r
    end
endfunction

```

Et si on remarque que $\forall n \geq 0, u^{(n)} = A^n u^{(0)} + \left(\sum_{k=0}^{n-1} A^k \right) r$, on peut aussi faire

```

function [u] = iteExo5_v2 (a,b,c,u0,m)
    [A,r] = linearExo5 (a,b,c)
    u = A^m * u0 ;
    for i=0:m-1
        u = u + A^i*r ;
    end
endfunction

```

```

4. m = 13 ;
a = [1 .1 3 -1 ] ;
b = [.4 .25 0 1.2 ] ;
c = [.7 0 3.1 -2] ;
u0 = [1 ; 2 ; 3 ] ;
um = iteExo5 (a,b,c,u0,m)
um =
    82443279.
    9181126.9
    77791375.
um_bis = iteExo5_v2 (a,b,c,u0,m)
um_bis =
    82443279.
    9181126.9
    77791375.

```