Correction TP3 Python ECG2

Exercice 1

1)commandes à effectuer sur la console :

```
A=np.array([[1,-3],[2,-7]])
B=np.array([[1,1],[-1,0]])
C=np.transpose(np.dot(A,B))
D=np.dot(np.transpose(B),np.transpose(A))
print(C,D)
```

Le programme renvoie : $C = D = \begin{pmatrix} 4 & 9 \\ 1 & 2 \end{pmatrix}$.

Exercice 2

1)(S) s'écrit alors matriciellement sous la forme $AX = B$ avec :

$A = \begin{pmatrix} 1 & -2 & -4 \\ 2 & 0 & 3 \\ -3 & 5 & 9 \end{pmatrix}$, $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ et $B = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$.

2)commandes à effectuer sur la console :

```
A=np.array([[1,-2,-4],[2,0,3],[-3,5,9]])
B=np.array([[2],[1],[3]])
S=np.dot(al.inv(A),B)
print(al.inv(A))
print(S)
```

Le programme renvoie : $A^{-1} = \begin{pmatrix} 15 & 2 & 6 \\ 27 & 3 & 11 \\ -10 & -1 & -4 \end{pmatrix}$.

La solution du système (S) est alors donnée par $X = A^{-1}B = \begin{pmatrix} 50 \\ 90 \\ -33 \end{pmatrix}$.

> Remarque
> La commande `al.solve(A,B)` donne directement la solution de (S).

Exercice 3

1)$A = \begin{pmatrix} \frac{1}{6} & \frac{1}{6} \\ 1 & \frac{1}{3} \end{pmatrix}$.

3)Sur la console :

```
A=np.array([[1/6,1/6],[1,1/3]])
C0=np.array(([1],[2]))
C100=np.dot(al.matrix_power(A,100),C0)
print("u100=",C100[0,0])
print("v100=",C100[1,0])
```

Le programme renvoie : $u_{100} \approx 1.96 \times 10^{-18}$ et $v_{100} \approx 5.90 \times 10^{-18}$.
On peut conjecturer que $\lim\limits_{n \to +\infty} u_n = 0$ et $\lim\limits_{n \to +\infty} v_n = 0$.

Exercice 4
Sur la console :

```
A=np.array([[0,1,1],[0,2,0],[-2,1,3]])
U=np.array([[1],[0],[1]])
np.dot(A,U)
array([[1],[0],[1]])
```

On voit donc que $AU = U$, ce qui prouve que U est vecteur propre de A associé à 1.

Exercice 5
D'après le THM1 du chapitre 4, λ est valeur propre de A si et seulement si $A - \lambda I$ n'est pas inversible.
Or, les matrice inversibles de $\mathscr{M}_n(\mathbf{R})$ sont celles de rang n, d'où l'équivalence mentionnée...

Sur la console :

```
A=np.array([[0,1,0],[1,0,1],[1,1,1]])
I=np.eye(3)
al.matrix_rank(A+I)
2
```

Le rang de $A + I$ est inférieur à 3, donc -1 est valeur propre de A.

Exercice 6

1)$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, B = \begin{pmatrix} 5 \\ 6 \end{pmatrix}, C = \begin{pmatrix} 7 & 8 \end{pmatrix}, V = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 7 & 8 \end{pmatrix}$ et $H = \begin{pmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{pmatrix}$.

2)$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$, $L = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$ et $C = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$.

Par concaténations, $B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ et $D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}$.

3)programme :

```
import numpy as np
C=np.ones((3,1))
Z=np.zeros((3,4))
H=np.hstack((C,Z))
L=np.ones((1,5))
D=np.vstack((L,H,L))
```

Exercice 7

$B = A + {}^tA$ avec $A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$. Donc $B = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$.

${}^tB = {}^t\left(A + {}^tA\right) = {}^tA + {}^t\left({}^tA\right) = {}^tA + A = B$.

B est symétrique donc diagonalisable.

Exercice 8

programme :

```
def f(a,b,n):
    A=b*np.ones((n,n))+(a-b)*np.eye(n)
    return A
```

Exercice 9

Programme :

```
def g(n):
    M=np.zeros((n,n))
    for i in range(n):
        M[i,:]=(i+1)*np.ones((1,n))
    return M
```

Exercice 10

1)Exemple de matrice stochastique : $A = \begin{pmatrix} 0.2 & 0.8 \\ 0.3 & 0.7 \end{pmatrix}$.

2)programme :

```
def stochastique(A):
    for i in range(len(A)):
        for j in range(len(A)):
            if A[i,j]<0:
                return False
        if sum(A[i,:])!=1:
            return False
    return True
```

Exercice 11

1)programme :

```
import numpy as np
def f(A):
    c=0
    for i in range(len(A)):
        for j in range(len(A)):
            if A[i,j]==0:
                c=c+1
    return c
```

2)programme :

```
def g(A):
    for i in range(len(A)):
        for j in range(len(A)):
            if i!=j and A[i,j]!=0:
                return False
    return True
```

Exercice 12

1)La fonction renvoie T^p.

On obtenait directement ce résultat par la commande `al.matrix_power(T,p)`.

2)Prenons par exemple $T = \begin{pmatrix} 1/2 & 1 & 3 \\ 0 & -1/2 & 5 \\ 0 & 0 & 1/3 \end{pmatrix}$.

On obtient $T^{10} = \begin{pmatrix} 0,000976 & 0 & 0,0518 \\ 0 & 0,000976 & -0,00575 \\ 0 & 0 & 0,000017 \end{pmatrix}$.

Les coefficients de T^{10} sont proches de zéro.

3)a)Soit $\mathscr{P}(p)$ la proposition : « il existe un réel b_p tel que $T^p = \begin{pmatrix} a^p & b_p \\ 0 & c^p \end{pmatrix}$ où $0 \leq b_p \leq pa^{p-1}b$ ».

$\mathscr{P}(0)$ est vraie car $T^0 = I$. On prend alors $b_0 = 0$. L'égalité et la double inégalité sont vérifiées.

Soit $p \in \mathbf{N}$. Supposons $\mathscr{P}(p)$ vraie, montrons que $\mathscr{P}(p+1)$ est vraie.

$T^{p+1} = T^pT = \begin{pmatrix} a^p & b_p \\ 0 & c^p \end{pmatrix}\begin{pmatrix} a & b \\ 0 & c \end{pmatrix} = \begin{pmatrix} a^{p+1} & ba^p + cb_p \\ 0 & c^{p+1} \end{pmatrix}$.

Posons $b_{p+1} = ba^p + cb_p$.

On a $b_p \geq 0$ par hypothèse de récurrence. De plus, a, b et c sont positifs ou nuls donc $b_{p+1} \geq 0$.

De plus, $b_p \leq pa^{p-1}b$ par hypothèse de récurrence donc $cb_p \leq cpa^{p-1}b$.

De plus, $cpa^{p-1}b \leq pa^p b$ car $c \leq a$.
Par recollement d'inégalités : $cb_p \leq pa^p b$.
On déduit que $a^p b + cb_p \leq a^p b + pa^p b$, soit $b_{p+1} \leq (p+1)a^p b$.
Donc $\mathscr{P}(p+1)$ est vraie.

b)Comme $0 < a < 1$, on a $\lim\limits_{p\to+\infty} a^p = 0$ et, par croissances comparées, $\lim\limits_{p\to+\infty} pa^{p-1}b = 0$.

Comme $0 \leq b_p \leq pa^{p-1}b$, le théorème d'encadrement donne $\lim\limits_{p\to+\infty} b_p = 0$.

Enfin, $0 < c \leq a < 1$ donne $\lim\limits_{p\to+\infty} c^p = 0$.

Tous les coefficients de T^p tendent vers 0, donc $\lim\limits_{p\to+\infty} T^p = 0$.

4)Soit T une matrice triangulaire diagonalisable de $\mathscr{M}_2(\mathbf{R})$.

Posons $T = \begin{pmatrix} a & b \\ 0 & c \end{pmatrix}$.
T est triangulaire donc ses valeurs propres sont a et c.
T est diagonalisable donc il existe une matrice D diagonale portant sur sa diagonale les valeurs propres de T et une matrice P inversible telles que $T = PDP^{-1}$.
Par récurrence, on obtient $\forall p \in \mathbf{N}, T^p = PD^pP^{-1}$.

Posons $D = \begin{pmatrix} a & 0 \\ 0 & c \end{pmatrix}$, $P = \begin{pmatrix} d & e \\ f & g \end{pmatrix}$ et $P^{-1} = \begin{pmatrix} i & j \\ k & l \end{pmatrix}$.

On déduit pour tout $p \in \mathbf{N}$:

$$\begin{aligned} T^p &= \begin{pmatrix} d & e \\ f & g \end{pmatrix}\begin{pmatrix} a^p & 0 \\ 0 & c^p \end{pmatrix}\begin{pmatrix} i & j \\ k & l \end{pmatrix} \\ &= \begin{pmatrix} da^p & ec^p \\ fa^p & gc^p \end{pmatrix}\begin{pmatrix} i & j \\ k & l \end{pmatrix} \\ &= \begin{pmatrix} ida^p + kec^p & jda^p + elc^p \\ ifa^p + kgc^p & jfa^p + lgc^p \end{pmatrix}. \end{aligned}$$

Par hypothèse, $-1 < a < 1$ et $-1 < c < 1$. Donc $\lim\limits_{p\to+\infty} a^p = 0$ et $\lim\limits_{p\to+\infty} c^p = 0$.

On déduit que $\lim\limits_{p\to+\infty} T^p = 0$.