

# T.P. V - Matrices

Code Capytale : aa20-2467643

## I - Ce qu'il faut savoir

### Solution de l'exercice 1.

```
import numpy as np

n = 10
I = np.eye(3) # Contient la matrice identité : A^0
A = np.array([[1, 2, 1], [1/2, 1, 2], [1, 1/2, 1]]) # Contient la matrice A
B = np.dot(A, A) # Contient la matrice A^2

for k in range(3, n+1): # k varie de 3 à n
    C = 3 * B + 9/4 * I # Contient la matrice A^k
    I = A # Contient la matrice A^(k-2)
    A = B # Contient la matrice A^(k-1)
    B = C # Contient la matrice A^(k)

print(B) # Affiche A^n = A^(10)
```

□

### Solution de l'exercice 2.

#### 1.

```
import numpy as np

def g(x): # Définition de la fonction g
    return 2 * x - 1 + np.log(x/(x+1))

a = 0.5 # initialisation de la valeur de a
b = 1 # initialisation de la valeur de g

while b - a > 10**(-2) : # s'arrête dès que b - a <= 10^(-2)
    m = (a + b)/2 # calcule le milieu de [a, b]
```

```
if g(a) * g(m) <= 0: # si g change de signe entre a et m
    b = m
else: # si g change de signe entre m et b
    a = m

print(a, b)
```

#### 2.

```
import numpy as np
import matplotlib.pyplot as plt

U = np.zeros((51, 1)) # Crée un vecteur contenant uniquement des 0

for n in range(1, 51):
    U[n] = (2 * n - 1) - g(n) # U[n] contient la valeur de u_n

X = np.arange(0, 51) # Liste des abscisses : entiers de 0 à 50
S = np.cumsum(U) # S[i] contient u_0 + u_1 + ... + u_i

plt.figure()
plt.plot(X, S, '+')
plt.show()
```

Le graphique représente la suite de points de coordonnées  $(n, S_n)_{0 \leq n \leq 50}$ ,

où  $S_n = \sum_{i=0}^n u_i$ .

On peut conjecturer que la suite  $(S_n)$  tend vers  $+\infty$  et donc que la série de terme général  $u_n$  diverge. □

## II - Suites récurrentes

**Solution de l'exercice 3.** On note  $U_n = \begin{pmatrix} u_n \\ v_n \end{pmatrix}$ .

```
import numpy as np

n = 20
A = np.array([[ -2, 1], [3, 0]]) # Crée la matrice A
C = np.array([0, 1]) # Crée la matrice U_0

for k in range(1, n+1): # k varie de 1 à 20
    C = np.dot(A, C) # C contient A U_{k-1} = U_k

print(C[0], C[1]) # Affiche u_20 et v_20
```

□

**Solution de l'exercice 4.** On note  $U_n = \begin{pmatrix} u_n \\ v_n \end{pmatrix}$ .

```
import numpy as np

n = 12
A = np.array([[1, 1], [2, 0]]) # Crée la matrice A
C = np.array([1, 0]) # Contient la matrice U_0

for k in range(1, n+1): # k varie de 1 à 12
    C = np.dot(A, C) # C contient U_k

print(C[0], C[1]) # Affiche u_12 et v_12
```

□

**Solution de l'exercice 5.**

```
import numpy as np

A = np.array([[4, -6, 2], [2, -4, 2], [-2, 2, 0]]) # Crée la matrice A
U = [0, 1, 2] # Contient la valeur de U_0
for i in range(1, 11): # i varie de 1 à 10
    U = np.dot(A, U) # U contient U_k

print(U[0]) # Affiche la valeur de a_10
```

□

**Solution de l'exercice 6.**

1.

```
n = 20
v = -1 # Contient la valeur de u_0
u = 1 # Contient la valeur de u_1

for i in range(2, n+1): # i varie de 2 à n
    a = u # a contient la valeur de u_{i-1}
    u = 3 * u + 2 * v - 4 # contient la valeur de u_i
    v = a # contient la valeur de u_{i-1}

print(u)
```

2.

```
n = 20
X = np.array([[1], [-1]]) # contient X_0
A = np.array([[3, 2], [1, 0]]) # crée la matrice A
B = np.array([[4], [0]]) # crée la matrice B

for i in range(1, n+1): # i varie de 1 à 20
    X = np.dot(A, X) + B # contient la valeur de X_k

print(X[1]) # affiche la seconde composante de X_20, soit a_20
```

□

**Solution de l'exercice 7.**

```
import numpy as np

A = np.array([[1, 0, 0], [0, 1, 2], [2, 0, 1]]) # crée la matrice A
u = np.zeros((11, 1)) # contient 10 zéros
v = np.zeros((11, 1)) # contient 10 zéros
w = np.zeros((11, 1)) # contient 10 zéros
u[0] = 1 # la première composante de u contient u_0
v[0] = 0 # la première composante de u contient v_0
w[0] = 2 # la première composante de u contient w_0
X = np.array([1, 0, 2])
for i in range(1, 11):
    X = np.dot(A, X)
    u[i] = 1
    v[i] = X[1]
    w[i] = X[2]
```

```
X = np.arange(0, 11)

plt.figure()
plt.plot(X, u, 'r.') # Trace la suite u avec des points rouges
plt.plot(X, v, 'go') # Trace la suite v avec des points verts
plt.plot(X, w, 'b+') # Trace la suite w avec des + bleus
plt.show()
```

□