Algèbre linéaire (matrices) - niveau 2

I. Savoir utiliser l'algorithme du pivot de Gauss pour déterminer une base d'un « sous-espace propre »

Exercice 1

Pour tout $A \in \mathscr{M}_3(\mathbb{R})$ et pour tout $\lambda \in \mathbb{R}$, on note : $E_\lambda(A) = \{X \in \mathscr{M}_{3,1}(\mathbb{R}) \mid AX = \lambda \cdot X\}$.

Commentaire

- On peut démontrer que, pour tout $\lambda \in \mathbb{R}$, $E_\lambda(A)$ est un espace vectoriel.
  Pour ce faire, on démontre que $E_\lambda(A)$ est un sous-espace vectoriel de $\mathscr{M}_{3,1}(\mathbb{R})$.
  On laisse la preuve en exercice.
- Savoir déterminer une base de $E_\lambda(A)$ constitue un exercice incontournable en $2^{\text{ème}}$ année. Cette question est présente dans `TOUTES` les épreuves de mathématiques. C'est d'ailleurs une excellente nouvelle car cette question se résout très facilement : il s'agit simplement de savoir résoudre un système linéaire homogène ! En effet :

$$\begin{array}{rcl} X \in E_\lambda(A) & \Leftrightarrow & AX = \lambda \cdot X \\ & \Leftrightarrow & AX - \lambda \cdot X = 0_{\mathscr{M}_{3,1}(\mathbb{R})} \\ & \Leftrightarrow & \big(A - \lambda \cdot I_3\big) \cdot X = 0_{\mathscr{M}_{3,1}(\mathbb{R})} \end{array}$$

1. Dans la suite de l'exercice, on note $A = \begin{pmatrix} 1 & 0 & 2 \\ 0 & -2 & 1 \\ 0 & 0 & 3 \end{pmatrix}$.

a) Déterminer $E_1(A)$.

b) Déterminer $E_{-2}(A)$.

c) Déterminer $E_3(A)$.

2. On considère la matrice $P = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & \frac{1}{5} \\ 0 & 0 & 1 \end{pmatrix}$.

a) Démontrer que P est inversible et déterminer P^{-1}.

b) Démontrer que la matrice définie par $D = P^{-1}AP$ est diagonale.
Déduire de cette écriture une écriture de la matrice A en fonction des matrices P et D.

c) Montrer que pour tout $n \in \mathbb{N}$, $A^n = PD^nP^{-1}$.

3. La formule de la question ***2.c)*** est-elle vérifiée pour $n = -1$?

Exercice 2

On considère la matrice $A = \begin{pmatrix} 3 & -2 & 2 \\ -1 & 4 & -2 \\ 0 & 4 & -1 \end{pmatrix}$.

Pour tout $\lambda \in \mathbb{R}$, on note $E_\lambda(A) = \{X \in \mathscr{M}_{3,1}(\mathbb{R}) \mid AX = \lambda \cdot X\}$.

1. ***a)*** Déterminer $E_1(A)$.

b) Déterminer $E_2(A)$.

c) Déterminer $E_3(A)$.

2. On considère la matrice $P = \begin{pmatrix} 1 & -2 & 1 \\ -1 & 3 & -1 \\ -2 & 4 & -1 \end{pmatrix}$.

a) Démontrer que P est inversible d'inverse $P^{-1} = \begin{pmatrix} 1 & 2 & -1 \\ 1 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix}$.

b) Démontrer que la matrice définie par $D = P^{-1}AP$ est diagonale.
Déduire de cette écriture une écriture de la matrice A en fonction des matrices P et D.

c) Montrer que pour tout $n \in \mathbb{N}$, $A^n = PD^nP^{-1}$.

II. Puissances d'une matrice via le binôme de Newton

Exercice 3
On considère les matrices suivantes :

$$A = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 3 & -1 \\ 1 & 1 & 1 \end{pmatrix}, \qquad P = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \qquad J = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Pour tout $\lambda \in \mathbb{R}$, on note $E_\lambda(A) = \{X \in \mathscr{M}_{3,1}(\mathbb{R}) \mid AX = \lambda X\}$.

1. Démontrer que la matrice $A - 2I$ est non inversible. Déterminer $E_2(A)$.

2. Démontrer que P est inversible et déterminer son inverse P^{-1}.

3. Démontrer que la matrice $\Delta = P^{-1}AP$ est triangulaire supérieure et trouver $\alpha \in \mathbb{R}$ tel que :

$$\Delta = \alpha I + J$$

4. Soit $n \in \mathbb{N}$.

a) Calculer J^2, puis J^k pour tout entier $k \geqslant 2$.

b) À l'aide de la formule du binôme de Newton, exprimer Δ^n en fonction de I et J.

c) En déduire l'expression matricielle de A^n.

III. Manipulations de matrices en Python

Testez les commandes suivantes sur votre ordinateur.

- `np.array(L)` prend en paramètre une liste de listes `L` et renvoie la matrice dont les lignes sont les listes contenues dans `L`. Exemple :

```
M1 = np.array([[1,2,3],[4,5,6],[7,8,9]])
print(M1)
```

- `np.zeros([n,m])` renvoie la matrice de taille `n` $\times$ `m` remplie de 0. Exemple :

```
M2 = np.zeros([2,3])
print(M2)
```

- `np.ones([n,m])` renvoie la matrice de taille `n` $\times$ `m` remplie de 1. Exemple :

```
M3 = np.ones([4,2])
print(M3)
```

- `np.eye(n)` renvoie la matrice identité d'ordre `n`. Exemple :

```
M4 = np.eye(3)
print(M4)
```

- `np.dot(M,N)` renvoie la matrice $A = MN$ (produit matriciel). Exemple :

```
M6 = np.dot(M1,M4)
print(M6)
```

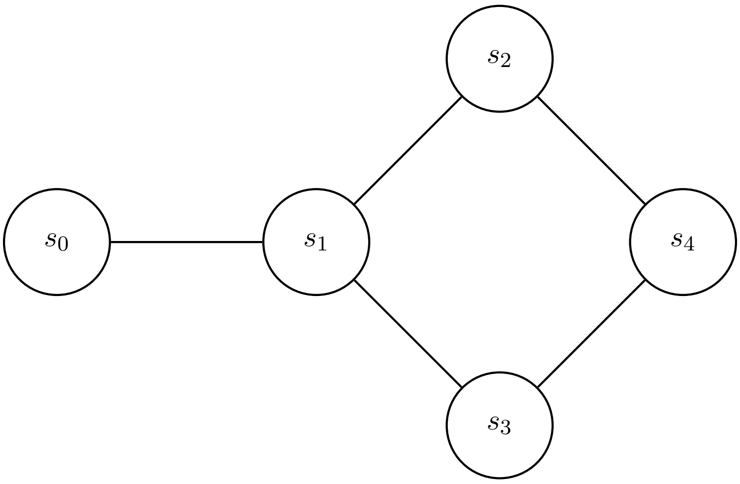
Exercice 4
On rappelle que la commande `rd.randint(a, b, [n, m])` renvoie une matrice aléatoire de taille $n \times m$ dont chaque coefficient est choisi indépendamment et uniformément dans $[\![a, b-1]\!]$.
Expliquer ce que fait le script **Python** suivant.

```
import numpy as np
import numpy.random as rd

A = np.array([[1,1,1], [0,1,1], [0,0,1]])
B = np.ones([3,3]) - np.eye(3)
C = rd.randint(-3, 4, [3,3])
D = np.dot(np.dot(A,B),C)
print('A = '), print(A)
print('B = '), print(B)
print('C = '), print(C)
print('D = '), print(D)
```

Exercice 5

On considère le graphe G non orienté suivant :



1. Rappeler la définition de la matrice d'adjacence du graphe G. On la notera M.
2. Quelle commande doit-on écrire pour définir la matrice M en **Python** ?
3. Soient n un entier naturel non nul, i un entier de $[\![1,5]\!]$ et j un entier de $[\![1,5]\!]$.
   Rappeler sans justification l'interprétation du coefficient situé à la ligne i et à la colonne j dans la matrice M^n.