TP9 : réduction des matrices carrées

Commencer par importer les bibliothèques suivantes dans chaque fichier **Python** utilisé :

```
import numpy as np
import numpy.linalg as al
```

I. Quelques commandes à connaître

I.1. Commandes de la bibliothèque `numpy`

- `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.shape(M)` renvoie un couple (n, p) où n est le nombre de lignes de M et p est le nombre de colonnes de M. Exemple :

```
n,p = np.shape(M3)
print(n,p)
```

- `np.transpose(M)` renvoie la transposée de la matrice M. Exemple :

```
M5 = np.transpose(M1)
print(M5)
```

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

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

- La commande `M == N` renvoie une matrice $A = (A_{i,j})$ de même taille que M et N dont tous les coefficients sont des booléens. Plus précisément : $A_{i,j}$ est égal à `True` si $M_{i,j} = N_{i,j}$ et à `False` sinon. Exemple :

```
print(M6==M1)
```

I.2. Commandes de la bibliothèque `numpy.linalg`

- `al.inv(M)` renvoie l'inverse de la matrice `M` (ou un message d'erreur si `M` n'est pas inversible). Exemple :

```
M7 = np.array([[0,1,1],[-2,3,2],[1,-1,0]])
print(M7)
M8 = al.inv(M7)
print(M8)
```

  ► Retrouver le résultat affiché pour l'inverse de la matrice `M7` via l'algorithme du pivot de Gauss.

- `al.matrix_rank(M)` renvoie le rang de la matrice `M`. Exemple :

```
print(al.matrix_rank(M7))
print(al.matrix_rank(M7-np.eye(3)))
```

  ► Réécrire mathématiquement le résultat obtenu avec la commande `print(al.matrix_rank(M7-np.eye(3)))`.

  Le résultat se réécrit $\mathrm{rg}(M_7 - I) = 1$.

  ► En déduire une valeur propre de la matrice `M7` et donner la dimension du sous-espace propre associé.

  On en déduit que 1 est une valeur propre de la matrice M_7. D'après le théorème du rang :

$$3 = \dim(E_1(M_7)) + \mathrm{rg}(M_7 - I)$$

  donc $\dim(E_1(M_7)) = 2$.

- `al.matrix_power(M,n)` renvoie la matrice `M` élevée à la puissance `n`. Exemple :

```
print(al.matrix_power(M7-np.eye(3),2))
```

  ► Réécrire mathématiquement le résultat obtenu ci-dessus.

  Le résultat se réécrit $(M_7 - I)^2 = 0_{\mathscr{M}_3(\mathbb{R})}$.

  ► En déduire un polynôme annulateur de la matrice M_7.

  On en déduit que $P(X) = (X-1)^2$ est un polynôme annulateur de M_7.

- `al.solve(M,V)` renvoie la solution de l'équation $MX = V$ d'inconnue X, où M est inversible. Exemple :

```
print(al.solve(M7,np.ones([3,1])))
```

- ▶ Retrouver le résultat affiché via l'algorithme du pivot de Gauss.

- `al.eig(M)` renvoie les valeurs propres de la matrice `M` ainsi qu'un vecteur propre associé à chacune des valeurs propres. Exemple :

```
print(al.eig(M7))
```

Commentaire

Attention, **Python** n'est pas un logiciel de calcul formel. Il donne donc des valeurs approchées pour les valeurs propres et les coordonnées des vecteurs propres.

- (Hors-programme) `al.det(M)` renvoie le déterminant de la matrice carrée M. Exemple :

```
M9 = np.array([[2,3],[1,2]])
print(M9)
print(al.det(M9))
```

II. Algèbre linéaire et informatique aux concours

II.1. EDHEC 2017

On note E l'espace vectoriel des fonctions polynomiales de degré inférieur ou égal à 2 et on rappelle que la famille (e_0, e_1, e_2) est une base de E, les fonctions e_0, e_1 e_2 étant définies par :

$$\forall t \in \mathbb{R} \quad e_0(t) = 1, \quad e_1(t) = t, \quad e_2(t) = t^2$$

On considère l'application φ qui, à toute fonction P de E, associe la fonction, notée $\varphi(P)$, définie par :

$$\forall x \in \mathbb{R},\ (\varphi(P))(x) = \int_0^1 P(x+t)\ dt$$

1. ***a)*** Montrer que φ est linéaire.

b) Déterminer $(\varphi(e_0))(x)$, $(\varphi(e_1))(x)$ et $(\varphi(e_2))(x)$ en fonction de x, puis écrire $\varphi(e_0)$, $\varphi(e_1)$ et $\varphi(e_2)$ comme combinaison linéaire de e_0, e_1 et e_2.

c) Déduire des questions précédentes que φ est un endomorphisme de E.

2. ***a)*** Écrire la matrice A de φ dans la base (e_0, e_1, e_2). On vérifiera que la première ligne de A est :

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \end{pmatrix}$$

b) Justifier que φ est un automorphisme de E.

c) La matrice A est-elle diagonalisable ?

3. Compléter les commandes **Python** suivantes pour que soit affichée la matrice A^n pour une valeur de n entrée par l'utilisateur :

```
n = int(input('Rentrez la valeur de n :'))
A = ---
print(---)
```

4. ***a)*** Montrer par récurrence que, pour tout entier naturel n, il existe un réel u_n tel que l'on ait :

$$A^n = \begin{pmatrix} 1 & \frac{n}{2} & u_n \\ 0 & 1 & n \\ 0 & 0 & 1 \end{pmatrix}$$

Donner u_0 et établir que : $\forall n \in \mathbb{N},\ u_{n+1} = u_n + \dfrac{1}{6}(3n+2)$.

b) En déduire, par sommation, l'expression de u_n pour tout entier n.

c) Écrire A^n sous forme de tableau matriciel.

II.2. EDHEC 2022

On note $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $J = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, $K_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, $K_2 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $K_3 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$, $K_4 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ et on rappelle que (K_1, K_2, K_3, K_4) est la base canonique de $\mathscr{M}_2(\mathbb{R})$.
On considère l'application φ qui, à toute matrice M de $\mathscr{M}_2(\mathbb{R})$, associe :

$$\varphi(M) = JM - MJ$$

1. Montrer que φ est un endomorphisme de $\mathscr{M}_2(\mathbb{R})$.

2. ***a)*** Exprimer $\varphi(K_1)$, $\varphi(K_2)$, $\varphi(K_3)$ et $\varphi(K_4)$ comme combinaisons linéaires de K_1, K_2, K_3, K_4.

b) Expliquer comment est construite la matrice A de φ dans la base (K_1, K_2, K_3, K_4) puis expliciter A.

c) Montrer que A est diagonalisable.

3. ***a)*** Déterminer le rang de A, puis donner une base de $\mathrm{Im}(\varphi)$.

b) En déduire la dimension de $\mathrm{Ker}(\varphi)$, puis montrer que (I, J) est une base de $\mathrm{Ker}(\varphi)$.

4. ***a)*** Calculer A^2 puis montrer que $A^3 - 4A = 0$.

b) En déduire les valeurs propres possibles de A.

5. En **Python**, la commande `r=al.matrix_rank(M)` renvoie dans la variable `r` le rang de la matrice M. On a saisi :

```
A = np.array([[0,-1,1,0],[-1,0,0,1],[1,0,0,-1],[0,1,-1,0]])
r1 = al.matrix_rank(A-2*np.eye(4))
r2 = al.matrix_rank(A+2*np.eye(4))
print(f'r1={r1}')
print(f'r2={r2}')
```

Python a renvoyé `r1=3` et `r2=3`.
Que peut-on conjecturer quant aux valeurs propres non nulles de φ et à la dimension des sous-espaces propres associés ?

6. ***a)*** Résoudre les systèmes $AX = 2X$ et $AX = -2X$, avec $X = \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix}$.

b) Déterminer le spectre de A ainsi que les sous-espaces propres de A.