Lycée Louis Thuillier | Année 2025-2026
BCPST2

TP 9 - Réduction d'une matrice et d'un endomorphisme

Pour ce TP, nous avons besoin des bibliothèques suivantes :

NUMPY

`import numpy as np`
`np.array()` — Transforme une liste en matrice numpy
`np.zeros([n, m])` — Créé la matrice nulle de taille $n \times m$
`np.eye(n)` — Créé la matrice identité de taille n
`np.diag(L)` — Créé la matrice diagonale dont les termes diagonaux sont les éléments de la liste L
`np.transpose(M)` — Renvoie la transposée de M
`np.dot(M, P)` — Renvoie le produit matriciel MP

NUMPY.LINALG

`import numpy.linalg as la`
`la.inv(M)` — Renvoie l'inverse de la matrice M si elle est inversible
`la.matrix_rank(M)` — Renvoie le rang de M

1 Réduction d'une matrice

On pose $A = \begin{pmatrix} 3 & 0 & -1 \\ 2 & 4 & 2 \\ -1 & 0 & 3 \end{pmatrix} \in \mathcal{M}_3(\mathbb{R})$. On veut calculer A^n pour tout $n \in \mathbb{N}$ de différentes manières.

1.1 Calculer A^n sans réduction

1. En utilisant la fonction `array` du module `numpy`, définir la matrice A en Python.
2. Écrire une fonction itérative (avec un `for`) nommée `Puissance(n)` qui renvoie A^n.
3. Écrire une fonction récursive (avec un `if` et un appel à la fonction au rang précédent) nommée `PuissanceRec(n)` qui renvoie A^n.

1.2 Déterminer à la main des valeurs propres et des vecteurs propres pour A

4. (a) Montrer avec Python que 4 est une valeur propre de A en utilisant la notion de rang.
   (b) Qu'en déduisez-vous concernant la dimension de $E_4(A)$?
5. On veut déterminer avec Python si un vecteur $X \in \mathcal{M}_{3,1}(\mathbb{R})$ est un vecteur propre de A.
   (a) À quelles conditions sur les vecteurs AX et X a-t-on X vecteur propre de A ?
   (b) Comment lier cette information au rang de la matrice constituée en ligne des vecteurs AX et X ?
   (c) Écrire une fonction `EstVP(X)` qui prend en entrée un vecteur $X = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \in \mathcal{M}_{3,1}(\mathbb{R})$ défini en Python par `np.array([a,b,c])` et qui renvoie `True` si X est un vecteur propre de A, `False` sinon. Pour créer la matrice qui contient en ligne les vecteurs X et Y, on utilise la syntaxe `np.array([X,Y])`.

(d) Tester cette fonction sur le vecteur $U = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$ pour confirmer qu'il s'agit d'un vecteur propre de A.

6. (a) Écrire une fonction `DeterValP(X)` qui prend en entrée un vecteur $X \in \mathcal{M}_{3,1}(\mathbb{R})$ et qui renvoie la valeur propre associée si X est un vecteur propre de A, `False` sinon.
   (b) Tester cette fonction sur le vecteur U pour déterminer une deuxième valeur propre de A.
7. Connaît-on le spectre de A ? La matrice A est-elle diagonalisable ?
8. (a) En utilisant la fonction `EstVP`, montrer que les vecteurs $V = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ et $W = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$ sont des vecteurs propres de A et déterminer à quelle valeur propre ils sont associés.
   (b) En déduire une matrice inversible P et une matrice diagonale D telles que $A = PDP^{-1}$.
   (c) Que vaut alors A^n pour tout $n \in \mathbb{N}$?
   (d) Écrire une fonction `PuissanceDiago(n)` qui utilise la formule obtenue en question 8c pour calculer A^n.

1.3 Utiliser une fonction Python pour diagonaliser A

NUMPY.LINALG
`import numpy.linalg as la`
`la.eigvals(M)` — Renvoie la liste des valeurs propres de M
`la.eig(M)` — Renvoie un couple L, P où L est la liste des valeurs propres de M et P la matrice de passage associée

9. Comment utiliser les fonctions décrites dans l'aide Python pour retrouver les matrices D et P déterminées en question 8b ?

2 Réduction d'un endomorphisme

Soit $n \geq 1$. On pose φ l'application telle que :

$$\forall P \in \mathbb{R}_n[X],\ \varphi(P) = XP' + P.$$

10. Montrer que φ est un endomorphisme de $\mathbb{R}_n[X]$.

En Python, le polynôme $P = a_0 + a_1X + a_2X^2$ sera représenté par la liste $[a_0, a_1, a_2]$, le polynôme $Q = \sum_{k=0}^{n} a_kX^k$ par la liste $[a_0, a_1, \ldots, a_n]$. Pensez bien à tester toutes les fonctions sur un polynôme simple, par exemple $P = X^2 + 3X + 2$, représenté par la liste $[2, 3, 1]$.

11. Écrire une fonction `Derive(L)` qui prend en entrée la liste `L` correspondant au polynôme $P \in \mathbb{R}_n[X]$ et qui renvoie la liste correspondant au polynôme dérivé P'.
12. Écrire une fonction `Multiplie(L)` qui prend en entrée la liste `L` correspondant au polynôme $P \in \mathbb{R}_n[X]$ et qui renvoie la liste correspondant au polynôme XP.
13. Écrire une fonction `Phi` qui prend en entrée la liste `L` correspondant au polynôme $P \in \mathbb{R}_n[X]$ et qui renvoie la liste correspondant au polynôme $\varphi(P)$.
14. Déterminer les images des polynômes de la base canonique par φ.

15. Soit $\ell \in \mathbb{N}$.
    (a) Déterminer l'image des polynômes de la base canonique par φ^ℓ.
    (b) En déduire l'image d'un polynôme $P \in \mathbb{R}_n[X]$ quelconque par φ^ℓ.
    (c) Écrire une fonction `PuissancePhi(l,L)` qui prend en entrée un entier `l` et la liste `L` correspondant au polynôme $P \in \mathbb{R}_n[X]$ et qui renvoie la liste correspondant au polynôme $\varphi^\ell(P)$.

3 Un algorithme de détermination de valeurs propres : l'algorithme de la puissance itérée

Il est légitime de se demander comment Python réussit à déterminer les valeurs propres et les vecteurs propres lorsqu'on utilise la fonction `la.eig`.
Il existe plusieurs algorithmes de détermination de valeurs propres et vecteurs propres. Le plus simple est l'algorithme de la puissance itérée. Il permet d'approcher la plus grande valeur propre (en valeur absolue) d'une matrice et un vecteur propre associé. On travaille sur la matrice $A \in \mathcal{M}_3(\mathbb{R})$ définie en partie 1. L'algorithme fonctionne selon le principe suivant :

— On choisit un vecteur au hasard : X_0

— On calcule $Y_1 = AX_0$.

— On calcule $X_1 = \dfrac{Y_1}{\|Y_1\|}$.

— On continue : Y_2, X_2, etc

— On s'arrête lorsque X_n et X_{n+1} sont très proches. Ici, on va choisir $\|X_n - X_{n+1}\| \leq 10^{-4}$.

Le vecteur X_n est alors une approximation d'un vecteur propre de A, et on peut trouver la valeur propre associée en reproduisant le raisonnement mis en place dans notre fonction `DeterValP`.

On rappelle que pour un vecteur $Y = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \in \mathcal{M}_{3,1}(\mathbb{R})$, on a $\|Y\| = \sqrt{a^2 + b^2 + c^2}$.

16. Écrire une fonction `norm(X)` qui prend en entrée un vecteur $X \in \mathcal{M}_{3,1}(\mathbb{R})$ et qui renvoie le réel $\|X\|$.
17. Comment peut-on piocher au hasard un vecteur X_0 avec Python ?
18. Écrire un programme `AlgoPuissance(A)` qui prend en entrée une matrice $A \in \mathcal{M}_3(\mathbb{R})$, qui applique l'algorithme de la puissance itérée et qui renvoie une approximation de la plus grande valeur propre (en valeur absolue) de la matrice et un vecteur propre associé.
19. Tester ce programme sur la matrice A définie en partie 1. Le résultat est-il cohérent avec ceux de la partie 1 ?