L2 M-MA
2024-2025

Programmation sous Python

Feuille 1 : Utilisation des bibliothèques scientifiques `numpy`

Exercice 1 (numpy)**.** La commande `import numpy as np` donne la possibilité d'utiliser les éléments de la bibliothèque `numpy`.

1. Expérimentez les commandes suivantes dans la console (éventuellement utiliser la commande `print` pour visualiser les résultats).
   - a = [-4,3,2,1]
   - b = [11,9,-7,5]
   - v = np.array(a)
   - A = np.array([[1, 2, 3, -4]])
   - M = np.array([[2, 0, 0, 0],a, b])
   - type(a), type(v), type(A)
   - np.shape(v)
   - np.shape(A)
   - np.shape(M), np.shape(M.T)
   - np.size(M)
   - w = v + 2*v
   - v3 = v**3
   - C = v + A
   - Mv = M.dot(v)
   - N = Mv.dot(M)
   - MAt = M.dot(A.T)
   - M.dot(A)
   - A*v
   - v.T, np.shape(v.T)
2. Que signifie `M.dot(v)` ? Quelle type de variable est le résultat ? Pourquoi la quatrième commande dans la colonne de droite ne fonctionne-elle pas ? Les vecteurs v et $v.T$ sont-ils égaux ? Comment pourrait-on en décider ?
3. Que représentent $M[0,3]$, $M[2,3]$, $M[2,:]$ et $M[1:,3]$?
4. Étudier les méthodes `np.eye(...)`, `np.zeros(...)` et `np.linspace(...)`.

Exercice 2.
1. Calculer la complexité de la fonction calculant la somme de deux matrices de taille $n \times m$.
2. Rappeler l'expression des coordonnées m_{ik} pour $1 \leq i \leq n, 1 \leq k \leq p$ de la matrice M définie par $M = AB$ où $A = (a_{ij})_{1\leq i\leq n, 1\leq j\leq m}$ et $B = (b_{jk})_{1\leq j\leq m, 1\leq k\leq p}$
3. Écrire une fonction `produit` prenant deux matrices et renvoyant leur produit (sans utiliser le produit matriciel décrit dans le premier exercice).
4. Calculer la complexité de cet algorithme en fonction de n, m et p. Donner l'expression obtenue si A et B sont de taille $n \times n$. *On pourra commencer avec la complexité du produit d'un vecteur ligne par un vecteur colonne.*

Exercice 3. On considère la matrice M du premier exercice. On veut lui appliquer le pivot de Gauss.

1. Appliquer l'algorithme de Gauss à la matrice M.
2. Que représentent du point de vue de l'algorithme de Gauss les opérations $M_1 = D_3(\frac{1}{2})\, M$, $M_2 = P_{2,1}(4)\, M_1$ et $M_3 = P_{3,1}(-11)\, M_2$, où

$$D_1(\tfrac{1}{2}) = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad P_{2,1}(4) = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ et } P_{3,1}(-11) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -11 & 0 & 1 \end{pmatrix} ?$$

3. Écrire une fonction `tmpP` qui prend en argument n, i, j et x, avec n un entier naturel non nul, i et j des entiers $1 \leq i \neq j \leq n$ et x un réel, et qui rend la matrice carrée de taille $n \times n$, $P_{i,j}(x)$ et une fonction `tmpD` qui prend comme argument deux entiers naturels non nuls $i \leq n$ et un réel x et qui renvoie $D_i(x)$.
4. Pour quels paramètres les matrices $P_{ij}(x)$ et $D_i(x)$ sont-elles inversibles ? Quand c'est le cas, calculer leur inverse.
5. Écrire une fonction `unPivot` qui prend en argument $A = (a_{ij})$ et pos, avec A une matrice et $pos = [\alpha, \beta]$ une liste définissant le pivot, et qui rend la matrice obtenue après avoir utlisé le pivot $a_{\alpha\beta}$.
6. Écrire une fonction `pivot` qui prend en argument une matrice A et qui rend une matrice échelonnée associée à A en appliquant l'algorithme du pivot de Gauss (en faisant l'hypothèse que le pivot diagonal est non-nul à chaque étape).

   On utilisera les matrices de la question 3.
7. Tester sur la matrice M.
8. (plus difficile) Modifier la fonction de la question 6 pour ne pas avoir à faire d'hypothèse sur le pivot.
9. Écrire une fonction `LU` prenant une matrice carré (qui vérifie l'hypothèse précédente sur les pivots) A et renvoie un couple (L, U) où L est triangulaire inférieure, U est triangulaire supérieure et $A = LU$
10. Soit $A \in \mathcal{M}_n(\mathbb{R})$ inversible et $b \in \mathbb{R}^n$. Utiliser la question précédente pour résoudre l'équation $Ax = b$.
11. Calculer le déterminant d'une matrice carrée grâce à la décomposition de la question précédente.
12. Calculer la complexité du calcul de déterminant de cet exercice.

Exercice 4 (déterminant).
1. Écrire une fonction `det2d` prenant une matrice M de taille 2×2 en entrée et renvoyant son déterminant.
2. Écrire une fonction `det3d` prenant une matrice M de taille 3×3 en entrée et renvoyant son déterminant, qui utilise `det2d` pour faire ses calculs.
3. Généraliser le programme pour obtenir une fonction `detNd` prenant une matrice M de taille $n \times n$ en entrée et renvoyant son déterminant par une approche récursive.
4. Calculer la complexité de la fonction `detNd`.
5. Comparer vos résultats avec la commande `det` du module `np.linalg`.

Exercice 5 (valeurs propres et inverse).
1. La commande `eig` du module `np.linalg` permet de récupérer les valeurs propres d'une matrice carrée et les vecteurs propres qui leur sont associés. Effectuer quelques tests pour en comprendre le fonctionnement.
2. Construire P et D dans l'écriture $M = PDP^{-1}$ (on prendra des exemples où M est diagonalisable). On aura besoin de chercher la commande de `np.linalg` permettant d'inverser une matrice.
3. Résoudre le système
$$\begin{cases} x - y + 2z = 3 \\ -x + 2y + 3z = -7 \\ -y + z = 1. \end{cases}$$

Exercice 6 (rang).
1. Chercher une commande dans le module `np.linalg` permettant de déterminer le rang d'une matrice.
2. Effectuer quelques tests avec les matrices
$$M = \begin{pmatrix} 1 & -1 & 2 & 1 & 2 \\ -1 & 2 & 3 & -4 & 1 \\ 0 & -1 & 1 & 0 & 0 \end{pmatrix} \quad \text{et} \quad N = \begin{pmatrix} 1 & 2 & 3 & 16 & 0 & 17 \\ -2 & -1 & -3 & -14 & 0 & -16 \\ -1 & -2 & -3 & -16 & 0 & -17 \end{pmatrix}$$
ou toute autre matrice de votre choix (voir les feuilles d'exercices des cours d'algèbre linéaire).

Exercice 7 (Cayley-Hamilton).
1. Écrire une fonction `puissMat(M, n)` qui prend une matrice carrée M et un entier n et qui renvoie la matrice M^n.
2. Améliorer cette fonction en utilisant un algorithme récursif de type « diviser pour régner ».

3. Calculer la complexité de ces deux algorithmes (on pourra supposer que la taille de la matrice M est une puissance de 2 et admettre que le résultat obtenu est vrai quelle que soit la taille).
4. Comparer les temps de calcul de vos fonctions avec celles de `np.linalg.matrix_power` pour des grandes valeurs de n.
5. Comparer également les temps de calcul pour $n = 2$. Comment expliquer ce phénomène ?
6. Soient M une matrice de taille 2×2 et N une matrice de taille 3×3. Montrer que le polynôme caractéristique de M s'écrit
$$\chi_M = X^2 - \mathrm{tr}(M)X + \det(M).$$
et celui de N s'écrit [1]
$$\chi_N = (-1)^3 \left(X^3 - \mathrm{tr}(N)X^2 + \left(\frac{\mathrm{tr}(N)^2 - \mathrm{tr}(N^2)}{2} \right) X - \det(N) \right)$$
7. *(plus difficile)* Soient A une matrice de taille $n \times n$ et $\chi_A = (-1)^n \left(X^n + \sum_{i=0}^{n-1} a_i X^i \right)$ son polynôme caractéristique. Montrer que $a_0 = (-1)^n \det(A)$, $a_{n-1} = -\mathrm{tr}(A)$ et $a_{n-2} = \frac{\mathrm{tr}(A)^2 - \mathrm{tr}(A^2)}{2}$.
On pourra utiliser le lien entre les valeurs propres d'une matrice et son polynôme caractéristique, et utiliser les formules de Viète. On pourra aussi montrer que $\mathrm{tr}(A^2)$ *est la somme des carrés des valeurs propres de* A.
8. Le théorème de Cayley-Hamilton stipule que toute matrice carrée annule son polynôme caractéristique. Vérifier cette propriété à l'aide de votre fonction `puissMat` dans quelques cas simples (cf. question 6).

Exercice 8 (polynôme caractéristique)**.** Soient M une matrice de taille n et $\chi_M = (-1)^n \sum_{i=0}^{n} a_i X^i$ son polynôme caractéristique.

1. Soient $v_0, \ldots, v_n \in \mathbb{R}$ des réels distincts. Notons w_i l'évaluation de χ_M en v_i pour $0 \leq i \leq n$. Décrire ces $n+1$ égalités comme une égalité matricielle
$$C \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} w_0 \\ \vdots \\ w_n \end{pmatrix}$$
avec $C \in \mathcal{M}_{n+1,n+1}(\mathbb{R})$ puis calculer le déterminant de cette matrice.
2. Construire un vecteur $\mathbf{v}$ de taille $n+1$ contenant les valeurs de χ_M évalué en $n+1$ points aléatoirement choisis (on fera l'hypothèse (raisonnable !) que les points obtenus sont distincts). On utilisera les outils de `np.random`.
3. Construire la matrice C de la question précédente pour la famille $\mathbf{v}$. Déterminer les coefficients du polynôme caractéristique de M.
4. Vérifier le théorème de Cayley-Hamilton pour des matrices de grandes tailles.

1. On notera que cela a un sens si le corps de base est $\mathbb{Q}$, $\mathbb{R}$ ou $\mathbb{C}$ mais n'en a pas si celui-ci est $\mathbb{Z}/2\mathbb{Z}$