Université Lyon 1 Année 2020-2021
Master Mathématiques Générales $1^{\text{ère}}$ année
Logiciels Scientifiques

TP2 – Bases de Python 2

Pour tout le TP, on pourra entrer les commandes

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

1 Tableaux et matrices

Exercice 1

1. Écrire la matrice

$$A = \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix}$$

Calculez son déterminant ainsi que son inverse.

2. Entrez les vecteurs

$$b_1 = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix} \quad b_2 = \begin{bmatrix} 32.1 \\ 22.9 \\ 33.1 \\ 30.9 \end{bmatrix}$$

et résolvez les sytèmes linéaires $Ax = b_1$ et $Ax = b_2$ (utiliser `np.linalg.solve`). Qu'y a-t-il d'étonnant ? Calculer le conditionnement de A (utiliser `np.linalg.cond`).

Exercice 2

Tester les commandes `np.prod`, `np.sum`, `np.cumprod`, `np.cumsum`, `np.triu` et `np.tril` sur la matrice

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 3 & 6 & 9 \\ 4 & 3 & 2 \end{bmatrix}$$

Exercice 3

1. Définir le vecteur $V = [0, 1, 2, 3, \ldots, 49, 50]$. Définir le vecteur W contenant les cinq premiers éléments de V, et le vecteur X contenant les cinq premiers et les cinq derniers éléments. Définir ensuite le vecteur $Z = [0, 2, 4, \ldots, 48, 50]$ à partir de V.
2. Définir une matrice M aléatoire à trois lignes et sept colonnes (utiliser `np.random.rand`). Combien de nombres dans cette matrice sont plus grand que $0,5$? Où sont-ils situés ? (utiliser `np.nonzero` ou `np.argwhere`).
3. Construire alors la matrice P obtenue à partir de la matrice M en remplaçant tous les nombres de M inférieurs ou égal à $0,5$ par -2, et ceux supérieurs à $0,5$ par 4.

Exercice 4

1. Définir C une matrice nulle de taille $n = 6$.
2. Grâce à une double boucle sur i et j, remplir la matrice C de telle sorte que, pour $1 \leq i \leq n$ et $1 \leq j \leq i$, on ait $C_{i-1,j-1} = \binom{i}{j}$ en utilisant la relation de récurrence

$$C_i^j = C_{i-1}^{j-1} + C_{i-1}^j$$

3. En utilisant `np.mod`, construire la matrice B de taille n telle que

$$B_{i,j} = \begin{cases} 1 \text{ si } C_{i,j} \text{ est impair,} \\ 0 \text{ sinon.} \end{cases}$$

Exercice 5

1. On considère $n = 10$
   (a) Construire la matrice du Laplacien

$$A_n = \begin{bmatrix} 2 & -1 & 0 & \ldots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & 2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \ldots & 0 & -1 & 2 \end{bmatrix}$$

   en utilisant `np.diag` et `np.ones`.

   (b) Vérifier que vous savez montrer mathématiquement que les valeurs propres de A sont les $\lambda_k = 4\sin^2\left(\frac{k\pi}{2(n+1)}\right)$, $1 \leq k \leq n$ avec pour vecteur propre associé $p_k = \left[\sin\left(\frac{jk\pi}{(n+1)}\right)\right]_{1\leq j\leq n}$.

   (c) Calculer les valeurs propres et les vecteurs propres numériques de A (`np.linalg.eig`).

   (d) Calculer le conditionnement de la matrice.

   (e) Ré-ordonner le vecteur des valeurs propres (`np.sort`) de A et déterminer l'erreur numérique maximale commise sur les valeurs propres.

2. Recommencer pour $n = 50$, $n = 100$ et $n = 1000$. Que peut-on constater sur le conditionnement ?

---

2 Programmation

Exercice 1 - Division euclidienne

Sans utiliser de fonction prédéfinie, écrire une fonction `division` qui effectue la division euclidienne d'un entier a par un entier b, et renvoie le quotient q et le reste r de cette division.

— Faire une version *fonction*, où le couple (q, r) est renvoyé dans un tableau (avec la commande `return`)

— Faire une version *procédure* où le résultat de la division est imprimé à l'écran (avec la commande `print`)

Exercice 2 - Permutation circulaire

1. Écrire une fonction `permute_1step_right` prenant une liste en entrée, et qui effectue une permutation circulaire des éléments d'une liste à n éléments de un pas vers la droite.
2. Même exercice en effectuant une permutation circulaire de deux pas vers la gauche.

Exercice 3 - Exponentiation rapide

1. Sans utiliser la fonction `**`, écrire une fonction récursive `puiss(a,n)` calculant a^n.
2. En utilisant le fait que, selon la parité de n, on a $a^{2n} = \left(a^2\right)^n$ ou $a^{2n+1} = a \cdot \left(a^2\right)^n$, écrire une fonction récursive `puiss_rapide(a,n)`.
   $\hookrightarrow$ *Un entier n modulo p s'écrit* `n%p`.
3. Utiliser une variable globale `cpt` dans les deux fonctions pour compter le nombre de multiplications effectuées et comparer les deux méthodes.
   $\hookrightarrow$ *Définir* `cpt` *comme une variable globale de sorte à pouvoir l'utiliser en dehors de la fonction. Attention, elle doit être initialisée à chaque appel de la fonction.*

Exercice 4 - Tri par insersion

Le tri par insertion consiste à prendre les éléments d'un tableau un à un et de les insérer à la bonne place dans un tableau déjà trié, comme pour trier des cartes au fur et à mesure qu'elles sont distribuées.

1. Écrire une fonction `tri_insertion(T)` qui implémente le tri par insertion.
2. Tester la fonction sur un tableau T de taille n contenant des entiers aléatoires entre 0 et n (commande `np.random.randint`).

---

3 Figures

Exercice 1 - Courbes en 2D

On considère les fonctions $f(x) = \sin(x)$ et $g(x) = x\cos(x)$ pour $x \in [0, 2]$.

1. Définir un tableau X contenant une discrétisation en $n = 50$ points de $[0, 2]$ (utiliser `np.linspace`), puis définir deux vecteurs F et G contenant les valeurs de f et g en ces points.
2. Tracer F et G en fonction de X sur un même graphe avec des couleurs et des types de traits différents (voir l'aide).
3. Nommer les axes de la figure, donner un titre à la figure et à chaque courbe.

Exercice 2 - Courbes polaires

En utilisant la fonction `plt.polar`, tracer la courbe du trifolium $r_1 = 1 + \cos(\beta)$, pour $\beta \in [-\pi, \pi]$.

En ajoutant l'option `projection="polar"` à la fonction `plt.subplot`, tracer les courbes suivantes sur la même figure

— Cardioïde $r_1 = 1 + \cos(\beta)$, pour $\beta \in [-\pi, \pi]$.

— Limaçon $r_2 = 3/2 + \cos(\beta)$, pour $\beta \in [-\pi, \pi]$.

— Rosace $r_3 = \sin(6\beta)$, pour $\beta \in [-\pi, \pi]$.

— Spirale d'Archimède $r_4 = 2\beta$, pour $\beta \in [0, 4\pi]$.

Donner un titre à chaque courbe.
