TD – 01

Exercices d'applications

Exercice 0 – Création de matrices

Question 1 *Créer les matrices suivantes :* $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 \\ 1 & 6 & 1 & 1 \end{pmatrix}$ *et* $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 6 \end{pmatrix}$.

Question 2 *Créer une matrice 3x3 dont les valeurs vont de 0 à 8.*

Question 3 *Créer une matrice 3x3x3 avec des valeurs aléatoires.*

Question 4 *Créer une matrice 10x10 de 0 avec des 1 sur les bords*

Question 5 *Créer une matrice 8x8 de 0 et de 1 ayant la « forme » d'un jeu d'échec.*

Question 6 *Créer une matrice 8x8 de booléens ayant la « forme » d'un jeu d'échec.*

Question 7 *Normer une matrice aléatoire de taille 5x5* ($z_i = \dfrac{z_i - z_{min}}{z_i - z_{max}}$

Question 8 *Calculer le produit matriciel d'une matrice aléatoire 5x3 et 3x2.*

Question 9 *Soit un tableau de dix éléments numéroter de 0 à 10. Modifier ce tableau pour que les valeurs comprises entre 3 et 8 soient remplacées par leur opposé.*

Question 10 *Créer une matrice 5x5 dont chaque ligne est de la forme [0,1,2,3,4].*

Exercice 1 – Calcul de somme

Pour cet exercice, on prend $n = 1000000$. On pourra augmenter ou diminuer cette valeur en fonction de la machine utilisée.

1. Calculer $\sum_{i=0}^{n} i$ sans utiliser `numpy`.
2. Chronométrer le temps nécessaire pour le calcul précédent, par exemple en utilisant `time.clock()`.
3. Utiliser un tableau `numpy` et la méthode `sum` pour calculer à nouveau la somme proposée.
4. Comparer le temps de calcul avec la méthode précédente.

Exercice 2 – Produit de Wallis

On peut justifier que : $\pi = 2\prod_{n=1}^{+\infty} \frac{4n^2}{4n^2-1}$ appelé le *produit de Wallis.*

1. Écrire une fonction itérative, d'argument n, calculant : $2\prod_{i=1}^{n} \dfrac{4i^2}{4i^2-1}$.
2. Écrire une fonction utilisant un tableau `numpy`, effectuant le même calcul.
3. Comparer le temps de calcul de ces deux fonctions.

Exercice 3

1. Définir une matrice aléatoire `a` de taille 50×50.
2. Déterminer la valeur $\max_{i,j} |a_{i,j+1} - a_{i,j}|$.

Exercice 4

1. Définir une matrice aléatoire de flottants `a` de taille 50×50.
2. Compter le nombre de valeurs inférieures à 0.5.
3. Remplacer toutes les valeurs inférieures à 0.5 par 0, et celles strictement supérieures à 0.5 par 1.

Exercice 5

D'après exemple 3.15 p 28 « Algèbre linéaire », Robert C. Dalang, Amel Chaabouni.

On s'intéresse au système linéaire suivant :

$$\left\{\begin{array}{ccccccccc} x_1 & + & 2x_2 & + & x_3 & + & x_4 & = & 0 \\ & & x_2 & + & 2x_3 & + & x_4 & = & 0 \\ x_1 & & & + & x_3 & + & 2x_4 & = & 1 \\ 2x_1 & + & x_2 & & & + & x_4 & = & 0 \end{array}\right.$$

1. Vérifier qu'il n'y a qu'une solution à ce système.

2. En utilisant `np.linalg.solve`, déterminer cette solution.
3. Vérifier le résultat obtenu en utilisant un produit matriciel.
4. Construire la matrice `m` de format 4×5, dont les colonnes sont successivement les colonnes de `a` et `b`.
5. Appliquer à `m` la méthode du pivot de Gauss pour résoudre « à la main » le système proposé.

Exercice 6

1. Créer une matrice aléatoire de taille 5×15, constituées d'entiers et l'afficher.
2. Mettre à zéro tous les éléments de la première ligne de cette matrice.
3. Déterminer la moyenne des éléments de cette matrice.
4. Construire le vecteur dont les composantes sont les moyennes des lignes de la matrice.
5. Déterminer la moyenne des éléments de ce vecteur.