Université de la Polynésie française | Licence 2 | 2021-2022

Programmation analyse matricielle

Feuille de TP 1

Le langage utilisé est le *Python 3*. *Les vecteurs seront représentés sous forme de liste. Les matrices seront représentées sous forme de listes de listes.*

Exercice 1:

1. Tester avec Python $3 + 10^{-15} == 3$ puis $3 + 10^{-16} == 3$. Commenter.
2. Tester avec Python $1.1 + 1.1 == 2.2$ puis $1.1 + 1.1 + 1.1 == 3.3$. Commenter.

On pourra regarder la documentation Python concernant les “bizarreries” du calcul avec les nombres flottants à l'adresse `https://docs.python.org/fr/3/tutorial/floatingpoint.html`.

Pour éviter ces problèmes, on remplacera dans nos TP le test d'égalité $x == y$ entre deux flottants par un test $|x-y| < prec$ où prec est une précision fixée par l'utilisateur et inférieure à la précision machine 10^{-16} (inutile bien entendu si x et y sont de type entiers !).

Exercice 2:

1. Entrer successivement $a = 0$, $b = a$ puis $a = 1$. Que vaut a, que vaut b ?
2. Entrer successivement $A = [0, 1]$, $B = A$ puis $A[0] = 1$. Que vaut A ? Que vaut B ? Interpréter.
3. Écrire une fonction **copie_liste**(A) qui permet d'éviter ce problème de “copie superficielle”.
4. Écrire une fonction analogue **copie_matrice**(A) pour les matrices. Vérifier sur un exemple que la matrice B=**copie_matrice**(A) n'est pas modifiée lorsque vous modifiez A.

*Remarque : la fonction **deepcopy** du module **copy** permet aussi de faire une ”copie profonde” des listes ou des listes de listes. Pour l'utiliser, vous pouvez importer les fonctionnalités du module copy en tapant en préambule **from copy import ***. Cependant, vous devez savoir programmer (vite) ces fonctions de base **copie_liste** et **copie_matrice**.*

Exercice 3: Écrire une fonction **saisie()** qui demande la taille d'une matrice et qui renvoie la matrice saisie par l'utilisateur.

*Remarque : inutile d'utiliser cette fonction **saisie** dans vos TP pour rentrer une matrice donnée.*

Exercice 4: Écrire une fonction **affichage(**M**)** qui affiche la matrice M sous la forme d'un tableau. On séparera les coefficients d'une même ligne par une tabulation.

Exercice 5: Écrire une fonction **identite(**n**)** qui retourne la matrice identité I_n.

Exercice 6: Écrire une fonction **trace(**M**)** qui calcule la trace d'une matrice M. Tester votre fonction sur la matrice identité I_5.

Exercice 7: Écrire une fonction **addition_matrice**(M, N) qui renvoie la somme deux matrices M et N. Prévenir l'utilisateur en cas d'impossibilité d'additionner les deux matrices et renvoyer 0.

Exercice 8: Écrire une fonction **multiplie_scalaire**(M, a) qui multiplie la matrice M par le scalaire a.

Exercice 9: Écrire une fonction **multiplier_ligne**(M, i, a) qui renvoie la matrice obtenue à partir de la matrice M en multipliant sa i-ème ligne par a.

Exercice 10: Écrire une fonction **permuter_lignes**(M, i, j) qui renvoie la matrice obtenue à partir de la matrice M en permutant les lignes i et j.

Exercice 11: Écrire une fonction **transvection**(M, i, j, a) qui renvoie la matrice obtenue à partir de la matrice M en mettant dans la ligne i la somme de a fois la ligne j avec la ligne i.

Exercice 12: Faire les trois exercices précédents mais avec les colonnes.

Exercice 13: Écrire une fonction **transposition**(M) qui renvoie la matrice transposée de M.

Exercice 14: Écrire une fonction **produit_matrice**(M, N) qui renvoie le produit des matrices M et N par la méthode naïve. On préviendra l'utilisateur si le produit est impossible et on renverra FALSE.

Exercice 15: Écrire une fonction **compare_matrice**(M, N) qui compare deux matrices M et N. On retournera TRUE si elles sont égales, FALSE sinon. On tiendra compte des conclusions de l'exercice 1.

Exercice 16: Écrire une fonction **verifie_symetrique(M)** qui vérifie qu'une matrice M à coefficients réels est symétrique.

On testera les matrices $\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 6 \end{pmatrix}$, $\begin{pmatrix} 1 & 3 & 8 \\ 3 & 7 & 11 \\ 8 & 11 & -4 \end{pmatrix}$, $\begin{pmatrix} 1 & 3 & 8 \\ 3 & 7 & 11 \\ -8 & 11 & -4 \end{pmatrix}$.

Exercice 17: Écrire une fonction **verifie_antisymetrique(M)** qui vérifie qu'une matrice M à coefficients réels est anti-symétrique.

On testera les matrices $\begin{pmatrix} 0 & 2 & 3 \\ -2 & -3 & 6 \end{pmatrix}$, $\begin{pmatrix} 0 & 3 & 8 \\ -3 & 0 & 11 \\ -8 & -11 & 0 \end{pmatrix}$, $\begin{pmatrix} 1 & 3 & 8 \\ -3 & 7 & 11 \\ -8 & -11 & -4 \end{pmatrix}$.

Exercice 18: Écrire une fonction **verifie_orthogonale(M)** qui vérifie qu'une matrice M à coefficients réels est orthogonale.

On testera les matrices $\begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix}$, $\begin{pmatrix} \frac{1}{2} & \sqrt{3}/2 & 0 \\ \sqrt{3}/2 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$.