TP4 : Matrice inverse et résolution de système

1 Matrices inverses et déterminant en Python

Rappel 1. *Soit A une matrice carrée de taille $n \in \mathbb{N}^\star$. On dit que A est inversible si et seulement s'il existe une matrice notée A^{-1} telle que*

$$A \times A^{-1} = A^{-1} \times A = I_n \quad (\star)$$

où I_n est la matrice identité de taille n.
La matrice A^{-1} est appelée matrice inverse de A.
Dans l'égalité $(\star)$ de la définition, il suffit d'avoir $A^{-1} \times A = I_n$ ou $A \times A^{-1} = I_n$, l'un impliquant l'autre.

Exercice 1 : Soient les matrices suivantes :

$$A = \begin{pmatrix} 2 & 1 & 0 \\ -2 & -1 & 1 \\ 2 & 0 & 4 \end{pmatrix},\ B = \begin{pmatrix} -2 & -2 & 0.5 \\ 5 & 4 & -1 \\ 1 & 1 & 0 \end{pmatrix},\ C = \begin{pmatrix} 1 & 1 & -1 \\ 2 & 3 & 1 \\ -1 & 0 & 2 \end{pmatrix},$$

$$D = \begin{pmatrix} 1 & -1 & 4 \\ 2 & 2 & 1 \end{pmatrix},\ E = \begin{pmatrix} 1 & 2 & -2 \\ 3 & 1 & 1 \\ 1 & -3 & 5 \end{pmatrix},$$

1. Vérifier que B est bien la matrice inverse de A en calculant le produit matriciel $A \times B$.
2. Utiliser la commande *det* pour calculer les déterminants de A et de B. Commentez.
3. Utiliser la commande *inv* pour déterminer la matrice inverse de C. Vérifier le résultat en calculant $dot(C, inv(C))$.
4. Utiliser les commandes *inv* pour déterminer la matrice inverse de D. Commentez.
5. Utiliser les commandes *inv* et *det* pour déterminer la matrice inverse de E et son déterminant. Commentez.

Exercice 2 : effet des arrondis sur les matrices inverses
Soient les matrices suivantes :

$$A = \begin{pmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{pmatrix},\ B = \begin{pmatrix} 1 & 0.5 & 0.33 \\ 0.5 & 0.33 & 0.25 \\ 0.33 & 0.25 & 0.2 \end{pmatrix}$$

1. Calculer les matrices inverses de A et B avec Python. Commentez le résultat abtenu.
2. Calculer les déterminants de A et de B.
3. Dans la matrice B nous avons arrondi la fraction $\frac{1}{3}$ avec une précision de 10^{-2}. Quelle précision faut-il prendre pour que les coefficients des inverses de A et B soient proches à 10^{-2} près ? (tester à tâton)

2 Transposée d'une matrice

Définition 1. *Soit $A \in \mathcal{M}_{n,p}(\mathbb{R})$ une matrice.*
On appelle matrice transposée de A la matrice de taille $p \times n$, notée ${}^tA$ telle que ${}^tA(i,j) = A(j,i)$.

Exemple : Soit $A = \begin{pmatrix} 1 & 5 \\ -2 & 8 \\ 3 & 1 \end{pmatrix}$ et $B = \begin{pmatrix} -2 & 3 & 9 \\ 1 & 3 & -2 \\ 7 & -3 & 2 \end{pmatrix}$. on obtient :

$$^tA = \begin{pmatrix} 1 & -2 & 3 \\ 5 & 8 & 1 \end{pmatrix} \quad \text{et} \quad ^tB = \begin{pmatrix} -2 & 1 & 7 \\ 3 & 3 & -3 \\ 9 & -2 & 2 \end{pmatrix}$$

Remarque : Faire la transposée d'une matrice revient à faire une symétrie des coefficients par rapport à la diagonale ou, autrement dit à échanger les lignes contre les colonnes.

Exercice 3

1. Quelle fonction Python permet d'obtenir la transposée d'une matrice. La tester pour les matrices de l'exemple précédent.
2. À l'aide de deux boucles for imbriquées, écrire une fonction Python qui prend en entrée une matrice et qui renvoie la matrice transposée.

3 Lien entre systèmes et matrices inverses

Rappels : Un système linéaire à n équations et p inconnues se note de manière matricielle :

$$\begin{cases} a_{1,1}x_1 & +a_{1,2}x_2 & \dots & +a_{1,p}x_p & = b_1 \\ a_{2,1}x_1 & +a_{2,2}x_2 & \dots & +a_{2,p}x_p & = b_2 \\ \dots \\ a_{n,1}x_1 & +a_{n,2}x_2 & \dots & +a_{n,p}x_p & = b_n \end{cases} \Leftrightarrow \quad AX = B$$

où : $A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,p} \\ a_{2,1} & a_{2,2} & \dots & a_{2,p} \\ \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,p} \end{pmatrix}$, $X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_p \end{pmatrix}$ et $B = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix}$

Exemple :

Le système $\begin{cases} 2x_1 & +3x_2 & -x_3 & = 7 \\ -2x_1 & +x_2 & +3x_3 & = -3 \\ x_1 & +x_2 & -x_3 & = 3 \end{cases}$ se note $\begin{pmatrix} 2 & 3 & -1 \\ -2 & 1 & 3 \\ 1 & 1 & -1 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ -3 \\ 3 \end{pmatrix}$

Méthode : Pour résoudre un système linéaire il suffit de multiplier l'inverse de A par B :

$$AX = B \Leftrightarrow A^{-1}AX = A^{-1}B \Leftrightarrow Id \times X = A^{-1}B \Leftrightarrow X = A^{-1}B$$

Exercice 4

1. Utiliser Python pour trouver la solution du système :

$$\begin{cases} 2x_1 & +3x_2 & -x_3 & = 7 \\ -2x_1 & +x_2 & +3x_3 & = -3 \\ x_1 & +x_2 & -x_3 & = 3 \end{cases}$$

2. Le système suivant admet-il une unique solution ?

$$\begin{cases} 2x_1 & -3x_2 & -x_3 & = 2 \\ -2x_1 & +x_2 & +4x_3 & = -1 \\ 2x_1 & -5x_2 & +2x_3 & = 3 \end{cases}$$

4 Exercices complémentaires

Exercice 5
En remarquant que

$$3 = 3 \times 10^0$$
$$33 = 3 \times 10^0 + 3 \times 10^1$$
$$333 = 3 \times 10^0 + 3 \times 10^1 + 3 \times 10^2$$
$$3333 = 3 \times 10^0 + 3 \times 10^1 + 3 \times 10^2 + 3 \times 10^3$$
$$\ldots$$

Ecrire une fonction, nommée precision, qui prend en entrée un entier n et qui renvoie le nombre "33...3" dont l'écriture est composée de n chiffres 3 les uns à côté des autres.
Par exemple, precision(4) renvoie 0.3333.

Exercice 6
Ecrire une fonction, nommée construcB, qui prend en entrée un entier n, et qui renvoie la matrice B de l'exercice 2 précédent où $1/3$ est approché avec une précision de 10^{-n}

Par exemple, construcB(4) renvoie la matrice $B = \begin{pmatrix} 1 & 0.5 & 0.3333 \\ 0.5 & 0.3333 & 0.25 \\ 0.3333 & 0.25 & 0.2 \end{pmatrix}$

Exercice 7
Tester le script ci-dessous. Observer le résultat. Que se passe-t-il ?

```
A=array([[1,1/2,1/3], [1/2,1/3,1/4],[1/3,1/4,1/5]])
N=15
X=zeros([1,N-2])
Y=zeros([1,N-2])

for k in range(2,N):
    B=inv(construcB(k))
    X[0,k-2]=k
    Y[0,k-2]=norm(inv(A)-B)


print(log(Y))
print(X)

plot(log(X[0,:]),log(Y[0,:]))
show()
```