Chapitre 6 : Les matrices

Dans tout ce chapitre, on utilisera : **import numpy as np**
import numpy.linalg as al

1. Définir une matrice

Pour définir une matrice, on utilise **np.array**
Plus précisement, **M=np.array([[…,…],[…,…]])**

Ex : $A = \begin{pmatrix} -1 & 2 \\ 4 & 0 \end{pmatrix}$ A=np.array([[-1,2],[4,0]])

np.zeros((m,n)) définit une matrice de taille $m \times n$ ne contenant que des 0
np.ones((m,n)) définit une matrice de taille $m \times n$ ne contenant que des 1
np.eye(m,n) définit une matrice de taille $m \times n$ contenant des 1 sur la première diagonale, 0 ailleurs.
np.eye(m, n, i) défini une matrice de taille $m \times n$ contenant des 1 sur la i-ème parallèle à la première diagonale, 0 ailleurs.

Ex $I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. I3=np.eye(3,3)

Coefficient situé ligne i et colonne j : **A[i,j]**
(Attention, la numérotation des coefficients commence à 0 !)

Extraire une ligne ou une colonne :
Si A est une matrice, et i un entier $\begin{cases} \textbf{A[i]} \text{ est la i-ème ligne de A} \\ \textbf{A[:,i]} \text{ est la i-ème colonne de A} \end{cases}$

De manière générale, **A[a:b,c:d]** extrait la sous-matrice de lignes numérotées de a à $b - 1$, et de colonnes numérotées de c à $d - 1$.

Ex : Avec $A = \begin{pmatrix} -1 & 2 \\ 4 & 0 \end{pmatrix}$ Première ligne de A : A[0] Deuxième colonne de A : A[:,1]

2. Opérations sur les matrices

Opérations matricielles et opérations termes à termes :
Attention, si A et B sont deux matrices, m un réel, et n un entier naturel, les opérations :
A + B A – B m*A A*B A/B An**
sont les opérations termes à termes !

Pour le produit matriciel, il faut utiliser : **np.dot(A, B)**

Pour les autres opérations sur les matrices, nous avons besoin de la bibliothèque nump.linalg :

Puissance de A : **al.matrix_power(A, n)**

Exemple : Avec A = $\begin{pmatrix} -1 & 2 \\ 4 & 0 \end{pmatrix}$

A**2 donne : array([[1, 4], [16, 0]]) al.matrix_power(A,2) donne : array([[9, -2], [-4, 8]])

Inverse : Si A est une matrice carrée inversible, **al.inv(A)** renvoient l'inverse de A.

Résolution d'un système :
Soit A $\in \mathcal{M}_n(\mathbb{R})$ inversible et B $\in \mathcal{M}_{n,q}(\mathbb{R})$.
La solution de l'équation AX = B est donnée par : **X = al.solve(A, B)**

Remarque : X = np.dot(al.inv(A), B) est aussi possible.

Exemple : Résoudre le système : $\begin{cases} 2x - y = 4 \\ -x + 5y = 7 \end{cases}$

```
A=np.array([[2,-1],[-1,5]])
B=np.array([[4],[7]])
al.solve(A,B)                         ⇒ array([[3.], [2.]])
```

Rang : **al.matrix_rank(A)**

Transposée : **np.transpose(A)**

Valeurs propres et vecteurs propres : **al.eig(A)** renvoie une matrice avec les valeurs propres de A et une matrice avec les vecteurs propres correspondants. (Attention, les vecteurs propres sont de la forme $\begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix}$, avec $x_1^2 + \dots + x_n^2 = 1$

Exemple : Avec A = $\begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix}$:

```
A=np.array([[1,2],[0,3]])
print(al.eig(A))       renvoie :      (array([1., 3.]),
                                      array([[1.      , 0.70710678],
                                             [0.      , 0.70710678]]))
```