

---

# TP 10 : Matrices et résolution de systèmes linéaires

---

## 1 Introduction.

Afin de faire des calculs d'algèbre linéaire (calculs avec des matrices) nous aurons besoin de la bibliothèque `numpy.linalg`. Pour l'importer on utilise la commande :

```
import numpy.linalg as al
```

### 1.1 Construction de matrices

Tester la commande suivante dans la console :

```
1 np.array([[1,2,3],[4,5,6]])
```

▷ Que fait cette commande ?

▷ Ecrire une commande permettant de renvoyer la matrice  $A = \begin{pmatrix} 2 & 3 & -1 \\ 1 & 2 & 5 \\ 6 & 7 & 8 \\ -3 & -4 & -2 \end{pmatrix}$

Tester les commandes suivantes et expliquer ce qu'elles font :

- `np.zeros((2,3))`

- `np.zeros((4,4))`

- `np.ones((5,4))`

- `np.eye((3,3))`

- `np.eye((2,3))`

- `np.eye((4,3))`

- `np.arange(1,3,0.1)`

- `np.arange(-2,2,0.4)`

- `np.arange(3,1,0.1)`

## 1.2 Opérations sur les matrices.

▷ Créer 4 variables A, B, C et D telles que :

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, \quad B = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

▷ Tester les commandes suivantes dans la console et commenter :

- `A+C`

- `A-C`

- `A*B`

- `np.dot(A,B)`

- `np.dot(B,A)`

- `D**2`

- `D**3`

- `np.transpose(A)`

- `al.inv(D)`

- `al.inv(A)`

- `F=np.array([[1,1],[2,2]]), al.inv(F)`

## 2 Extraction d'éléments d'une matrice

On cherche ici à extraire un des coefficients ou une partie d'une matrice.

- ▷ Commencer par créer une variable `E` contenant la matrice  $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix}$
- ▷ Tester les commandes suivantes et les commenter.

- `E[1,1]`

`E[0,1]`

`E[1,4]`

`E[0,:]`

`E[:,2]`

`E[0:2,1:3]`

— Anticiper le résultat du code ci-dessous :

E[1:2,0:2]

### 3 Dimensions de matrices

▷ Changer les valeurs de A,B et C pour

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

▷ Tester les commandes et commenter `np.shape(A)`, `np.shape(B)`, `np.shape(C)`

### 4 Application aux résolutions de systèmes linéaires

On cherche à résoudre un système linéaire de la forme

$$\begin{cases} a_{1,1}x_1 + \dots + a_{1,n}x_n = y_1 \\ a_{2,1}x_1 + \dots + a_{2,n}x_n = y_2 \\ \dots \\ a_{n,1}x_1 + \dots + a_{n,n}x_n = y_n \end{cases}$$

Les  $a_{i,j}$  et les  $y_i$  étant donnés.

On rappelle que ce système peut s'écrire sous forme matricielle :

$$AX = Y$$

Avec  $A = \begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,n} \end{pmatrix}$ ,  $X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$  et  $Y_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ .

Alors si la matrice est inversible l'unique solution est

$$X = A^{-1}Y$$

▷ Rappeler quelle commande permet d'inverser une matrice  $M$

▷ A l'aide de Python résoudre le système

$$\begin{cases} x + 3y = 1 \\ 4x + 5y = 2 \end{cases}$$

▷ A l'aide de Python résoudre le système

$$\begin{cases} x + 2y + 3z = 1 \\ x - 3y + 2z = 3 \\ 8x + 2y - 4z = -2 \end{cases}$$

**Exercice 1.** Vous vous renseignez sur un concours, pour lequel les coefficients des épreuves sont inconnus. Vous parvenez à récupérer les résultats de trois candidats, rassemblés dans le tableau suivant :

| Candidat | A  | B  | C  |
|----------|----|----|----|
| LV2      | 7  | 11 | 11 |
| Anglais  | 12 | 6  | 16 |
| Maths    | 6  | 10 | 14 |
| Moyenne  | 8  | 9  | 14 |

Quels sont les coefficients de chaque épreuve ? Y a-t-il unicité de la solution ? Pourquoi ?