

# Calculs matriciels et polynomiaux avec Python

Lycée Richelieu  
Rueil Malmaison

avril 2026

L'objectif de cette séance est de répondre aux questions en complétant le script présent dans le même répertoire que ce sujet. Si la question requiert de rédiger en langage naturel, utiliser les commentaires pour répondre.

À la fin, on déposera le script dans ce répertoire en ligne avec le mot de passe `TP_matrices_2026` et on veillera à ce que le nom du fichier permette d'identifier le candidat. Par exemple `monscript.py` n'est pas explicite tandis que `script_matrices_polynomes_pa_fournie.py` l'est.

On rappelle que le script doit pouvoir être exécuté sans erreur de compilation par l'examineur. Si certaines des fonctions conduisent à des erreurs, il faut commenter les lignes correspondantes.

## 1 Polynômes

Dans Python, on codera un polynôme comme une liste de coefficients correspondant aux *degrés croissants*. Par exemple, pour définir  $P = 2X^4 + 3X^3 + X + 1$ , on tapera :

```
P=[1,1,0,3,2]
```

L'un des problèmes avec les polynômes résidera dans la gestion des derniers coefficients nuls. Par exemple les deux polynômes suivants sont en fait les mêmes :

```
P1=[1,1,0,3,2]
```

```
P2=[1,1,0,3,2,0,0]
```

On règle ce problème en écrivant la première fonction.

- ☞ *Algorithme n° 0*: Écrire une fonction nommée `Epure` qui :
  - prend pour argument un polynôme  $P$ ;
  - renvoie le même polynôme sans les 0 finaux.
- ☞ *Algorithme n° 1*: Écrire une fonction nommée `Degre` qui :
  - prend pour argument un polynôme  $P$ ;
  - renvoie le degré de  $P$ , avec pour convention  $-1$  si  $P$  est nul.
- ☞ *Algorithme n° 2*: Écrire une fonction nommée `SommePoly` qui :
  - prend pour argument deux polynômes  $P$  et  $Q$ ;
  - renvoie le polynôme de  $P + Q$ .
- ☞ *Algorithme n° 3*: Écrire une fonction nommée `ProduitPoly` qui :
  - prend pour argument deux polynômes  $P$  et  $Q$ ;
  - renvoie le polynôme de  $P \times Q$ .
- ☞ *Algorithme n° 4*: Écrire une fonction nommée `Monome` qui :
  - prend pour argument un entier  $n$ ;
  - renvoie le polynôme de  $X^n$ .

- ☞ *Algorithme n° 5:* Écrire une fonction nommée `ProduitExt` qui :
  - prend pour argument un polynôme  $P$  et un nombre  $a$ ;
  - renvoie le polynôme de  $aP$ .
- ☞ *Algorithme n° 6:* Écrire une fonction nommée `Division` qui :
  - prend pour argument deux polynômes  $P$  et  $D$ ;
  - renvoie un couple de polynôme  $(Q, R)$  correspondant au quotient et au reste de la division de  $P$  par  $D$ .

## 2 Matrices

Dans Python, on codera une matrice comme une liste de listes. Par exemple, pour définir  $A = \begin{pmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 2 \end{pmatrix}$ , on tapera :

```
A=[[1 ,2] ,[1 ,3] ,[1 ,2]]
```

- ☞ *Algorithme n° 7:* Écrire une fonction nommée `Taille` qui :
  - prend pour argument une matrice  $A$ ;
  - renvoie le couple  $(n, p)$  correspondant aux nombres de lignes et de colonnes de  $A$ .
- ☞ *Algorithme n° 8:* Écrire une fonction nommée `SommeMat` qui :
  - prend pour argument deux matrices  $A$  et  $B$ ;
  - renvoie la somme  $A + B$  ou bien un message d'erreur si la somme est impossible.
- ☞ *Algorithme n° 9:* Écrire une fonction nommée `Identite` qui :
  - prend pour argument un entier  $n$ ;
  - renvoie la matrice  $I_n$ .
- ☞ *Algorithme n° 10:* Écrire une fonction nommée `ProduitMat` qui :
  - prend pour argument deux matrices  $A$  et  $B$ ;
  - renvoie le produit  $A \times B$  ou bien un message d'erreur si le produit est impossible.