

Outils fondamentaux pour les maths : résolution systèmes linéaires

Dans ce module, nous utiliserons une version **3.x** du langage **python**.



Tous les TPs sont à faire sous GNU/Linux (pas en machine virtuelle, bootez sous GNU/Linux !)

1. Pivot de Gauss

On souhaite écrire une fonction python qui se comporte ainsi :

```
M = [[3,2,1], [2,-1,1], [1,3,2]]
Y = [1,6,-1]
print(pivot_de_Gauss(M, Y))
```

renvoie :

```
([[3, 2, 1], [0.0, -2.33333333333333, 0.333333333333337], [0.0, 0.0, 2.0]], [1,
5.33333333333333, 4.000000000000001])
```

Voici un code, simplifié et à trou de cette fonction. Complétez-le et testez avec d’autres systèmes

```
def pivot_de_Gauss(M,Y):

    for ipiv in range(len(M)-1): ❶

        if M[ipiv][ipiv] == 0:
            for i in range(ipiv + 1, len(M)): ❷

                if M[i][ipiv] != 0:
                    (M[ipiv], M[i]) = .... ❸
                    (Y[ipiv], Y[i]) = ....
                    break

            pivot = float(M[ipiv][ipiv])

            if pivot == 0: ❹
                continue

            for i in range(ipiv + 1, len(M)): ❺

                coeff = M[i][ipiv] / pivot
                for j in ..... : ❻
                    M[i][j] -= coeff * .....
                    Y[i] -= coeff * .....

    return M,Y
```

- ❶ Potentiels pivots
- ❷ S’il est nul, on en cherche un en dessous
- ❸ Echanger les lignes \$ipiv\$ et \$i\$ dans \$M\$ et \$Y\$
- ❹ On n’a pas trouvé de pivot non nul!
- ❺ Pour toutes les lignes sous le pivot :
- ❻ Modifier la ligne \$i\$ à l’aide du pivot

2. Avec les librairies

La librairie numpy contient des fonctions essentielles pour traiter les tableaux, les matrices et les opérations d’algèbre linéaire avec Python. Elle permet de faire du calcul numérique.

Sympy est une librairie de calcul symbolique (d’où le nom). Elle permet de faire du calcul formel comme développer réduire une expression avec des variables.

```
import numpy as np
import sympy as sp
```

Faire les exercices 4.7 et 4.9 avec python.