TP 21 - Pivot de Gauss

Objectif : Programmer l'inverse d'une matrice par la méthode du pivot de Gauss ; puis donner une adaptation pour résoudre un système $AX = B$.
On testera les fonctions sur la matrice inversible suivante :

$$A = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 0 & 0 & 2 & -1 \\ -1 & 1 & -1 & 2 \\ 1 & 0 & 0 & -1 \end{pmatrix}$$

Exercice 1 Compléter la fonction `echligne(M,i,j)` qui renvoie une matrice égale à la matrice M où les lignes i et j ont été échangées :

A compléter

```
def echligne(M,i,j):
    B=1*M
    B[i,:],B[j,:]=...
    return(B)
```

Quelle est la matrice affichée par :

```
B=echligne(A,1,2)
B=echligne(B,2,3)
print(B)
```

$$B = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

Remarque : On rappelle que `B=M` créer un alias (deux noms pour une même variable, mais `B=1*M` créer une copie.

Exercice 2 Compléter la fonction `pivot(M,i)` qui cherche la position du coefficient $|a_{j,i}|$ maximal dans la colonne i sur les lignes $j \geq i$.

pivot sur la ième colonne

```
def pivot(M,i):
    n,j=...
    for k in range(.........):
        ...
        ...
    return(j)
```

Que retourne l'instruction `pivot(A,1)` ?

Exercice 3 Compléter la fonction `elimine(M,i,j)` qui renvoie une matrice égale à la matrice M où la ligne j a été modifiée en utilisant le pivot en position (i,i) de ligne i : le coefficient en (j,i) a été annulé.

```
def elimine(M,i,j):
    B=1*M
    B[j]=...
    return(B)
```

Quelle est la matrice affichée par :

```
B=elimine(A,0,2)
B=elimine(B,0,3)
print(B)
```

$$B = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

Remarque : On peut utiliser `M[i,:]` ou `M[i]` pour obtenir la ième ligne de la matrice `M` ; en effet, un tableau est une *liste* de lignes.

➢ La fonction suivante normalise les coefficients diagonaux :

```
def normalise(M):
    B=1*M
    n=len(M)
    for i in range(n):
        B[i,:]=M[i,:]/M[i,i]
    return(B)
```


Lycée Saint-Exupery – maffre.saintexupery@gmail.com – MPSI

Exercice 4 Compléter la fonction `Gauss(M,B)` qui effectue la réduction du pivot de Gauss sur la matrice `M` concaténée à la matrice `B`. Suivant le choix de `B`, il sera donc possible de résoudre un système ou simplement d'inverser une matrice.

```
1 def Gauss(M,B=False):
2 # verification que M est carree
3     t=...
4     assert len(t)==2 and t[0]==t[1],
5         'matrice non carree'
6     n=t[0]
7 # si B==False prendre B=In ->calcul de l'inverse
8     if type(B)==bool:
9         B=np.identity(n)
10 # verification que B est de la bonne taille
11     tb=np.shape(B)
12     assert ...,"B n'est pas compatible"
13 # construction de M|B
14     T=np.zeros((.....))
15     T[:,:n]=...
16     ...
17 # travail sur les lignes
18     for i in range(n):
19         p=pivot(T,i)
20 # detecter si M est inversible
21         assert abs(T[p,i])>1e-14,
22                 'matrice non inversible'
23     # positionnement du pivot
24         if p!=i:
25             ...
26     # elimination sur la colonne
27         ...
28             if ...:
29                 T=elimine(T,i,j)
30     # normalisation des coeff diagonaux
31     T=normalise(T)
32     return(T)
```

Quel est l'aspect à la matrice `T`, lors de l'appel `Gauss(A)`, après l'exécution de l'instruction à :

1. la ligne 16 :

2. la ligne 25, pour `i=1` :

3. la ligne 29, pour `i=1` :

4. la ligne 31 :

Remarque : Il existe dans numpy la fonction `concatenate` :
- `np.concatenate((A,B),axis=1)`, `A` et `B` doivent avoir le même nombre de lignes, les lignes de A et B sont concaténées.
- `np.concatenate((A,B),axis=0)`, idem sur les colonnes. La valeur `axis=0` est celle par défaut.

Exercice 5 En utilisant la fonction gauss, donner le script d'une fonction `inverse(M)` qui retourne l'inverse de `M` ou un message d'erreur si elle ne l'est pas.

```
1 def inverse(M):
2     ...
3     ...
4     return(.............)
```

Quel est l'instruction qui permet d'afficher le produit $A * A^{-1}$?

Exercice 6 Donner le script permettant de résoudre les systèmes suivants :

$$AX = Y \text{ avec } Y \in \left\{ \begin{pmatrix} 1 \\ 0 \\ 2 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

```
Y=...
...
print('Les solutions sont : X= ')
print(.........)
print('Verification : AX = ')
print(.........)
```