Projet de Méthodes matricielles

Décomposition PLU et résolution de systèmes d'équations linéaires

---

Vous placerez dans un répertoire nommé Votrenom_Votreprenom *vos fichiers python 2.7 (partie 1) et un fichier sagews (partie 2) ainsi qu'un fichier d'explications.*
Comme d'habitude, vous zipperez le répertoire et déposerez le fichier zip obtenu sur moodle dans la zone réservée.

Diverses méthodes de résolution de systèmes linéaires $AX = B$ s'appuient sur le fait que la matrice A puisse se décomposer en le produit de matrices particulières. On s'intéresse ici au cas particulier où A est une matrice carrée ; c'est à dire que le système a autant d'inconnues que d'équations. On peut montrer que toute matrice carrée $A \in \mathcal{M}_n(\mathbb{R})$ peut se décomposer sous la forme du produit de 3 matrices sous la forme :

$$A = PLU$$

avec :

- P matrice de permutation.
- L matrice triangulaire inférieure
- U matrice triangulaire supérieure

Précisons ce que sont les matrices de permutation.

1 Matrices de permutations

Une matrice de permutation est, par définition, une matrice carrée vérifiant les 3 conditions :

- elle ne contient que les valeurs 0 ou 1
- chaque colonne ne contient qu'un seul 1
- chaque ligne ne contient qu'un seul 1

Par exemple :

$$P = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \text{ est une matrice de permutation.}$$

On peut décrire une matrice de permutation de taille n par un n-uplet qui contient les positions de la valeur 1 dans chaque colonne. La matrice précédente peut être décrite par le quintuplet $(3, 2, 5, 1, 4)$ (qui signifie que dans le colonne 1, la valeur 1 est à la ligne 3, dans la colonne 2, la valeur 1 est dans la ligne 2, ...
De cette manière, la matrice Id_n sera représentée par le n-uplet : $(1, 2, \cdots, n)$

Lorsqu'on résout un système linéaire $AX = B$, on peut être conduit à permuter deux lignes du système. D'un point de vue matriciel, cela se traduit par le produit de A par la matrice : $Id_n - E_{i,i} + E_{j,i} - Ej, j + E_{i,j}$. Cette matrice est une matrice de permutation :
Par exemple :

Le produit de la matrice $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ par une matrice carrée A correspond à l'échange des lignes $L3 \longleftrightarrow L4$ de A.

Quelques propriétés des matrices de permutation :

1. **Déterminant** :
   On peut remarquer que le déterminant d'une matrice de permutation ne peut prendre que 2 valeurs : -1 ou 1. La valeur du déterminant d'une matrice de permutation est liée au nombre α de permutations de colonnes nécessaires pour passer de la matrice P à la matrice identité Id_n par la formule :

   $$det(P) = (-1)^\alpha$$

   Pour la matrice de notre exemple, cela donne :

matrice initiale	$(3, 2, 5, 1, 4)$
permutation $C1 \longleftrightarrow C4$ pour obtenir la bonne colonne 1	$(1, 2, 5, 3, 4)$
la col 2 est déjà bien placée, on permute $C3 \longleftrightarrow C4$ pour obtenir la bonne col 3	$(1, 2, 3, 5, 4)$
la col 4 est déjà bien placée, on permute $C4 \longleftrightarrow C5$ et on obtient la matrice Id_5	$(1, 2, 3, 4, 5)$

   en tout 3 permutations ont été nécessaires, d'où : $Det(P) = (-1)^3 = -1$
2. **Inverse** :
   Le déterminant étant non nul, toute matrice de permutation est inversible.
   On peut établir pour une matrice de permutation le résultat particulièrement agréable :

   $$P^{-1} = P^T$$

   En particulier, lorsqu'on échange 2 lignes d'une matrice, la matrice P de la permutation associée est symétrique d'où :

   $$P^{-1} = P^T = P$$
3. **Produit de deux matrices de permutation** :
   Le produit de deux matrices de permutation est encore une matrice de permutation.
4. **Produit d'une matrice par une matrice de permutation** :
   Le produit par une matrice de permutation n'est pas commutatif !

- produit à gauche : $P * A$
  Prenons un exemple : la permutation $(4, 2, 1, 3)$

$$PA = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} a1 & a2 & a3 & a4 \\ b1 & b2 & b3 & b4 \\ c1 & c2 & c3 & c4 \\ d1 & d2 & d3 & d4 \end{pmatrix} = \begin{array}{c} \begin{pmatrix} c1 & c2 & c3 & c4 \\ b1 & b2 & b3 & b4 \\ d1 & d2 & d3 & d4 \\ a1 & a2 & a3 & a4 \end{pmatrix} \end{array} \begin{array}{c} L3 \\ L2 \\ L4 \\ L1 \end{array}$$

  Chaque ligne est globalement conservée, mais leur ordre change :
  · sur la ligne 1 on trouve maintenant la ligne 3 : 3 est la position de 1 ds $(4, 2, 1, 3)$
  · la ligne 2 est inchangée : 2 est la position de 2 dans $(4, 2, 1, 3)$
  · sur la ligne 3 on trouve maintenant la ligne 4 : 4 est la position de 3 ds $(4, 2, 1, 3)$
  · sur la ligne 4 on trouve maintenant la ligne 1 : 1 est la position de 4 ds $(4, 2, 1, 3)$
- produit à droite : $A * P$
  Pour les mêmes matrices :

$$AP = \begin{pmatrix} a1 & a2 & a3 & a4 \\ b1 & b2 & b3 & b4 \\ c1 & c2 & c3 & c4 \\ d1 & d2 & d3 & d4 \end{pmatrix} * \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} a4 & a2 & a1 & a3 \\ b4 & b2 & b1 & b3 \\ c4 & c2 & c1 & c3 \\ d4 & d2 & d1 & d3 \end{pmatrix}$$

  Chaque colonne est globalement conservée, mais leur ordre change : $(4, 2, 1, 3)$ donne directement l'ordre des colonnes
  · sur la colonne 1 on trouve la colonne 4 de A
  · sur la colonne 2 on trouve la colonne 2 de A
  · sur la colonne 3 on trouve la colonne 1 de A
  · sur la colonne 4 on trouve la colonne 3 de A

En conclusion, pour calculer le produit d'une matrice A par une matrice de permutation, aucun calcul n'est nécessaire, il suffit de réécrire la matrice A en échangeant ses lignes ou ses colonnes.

2 Matrices de transvection

Une transvection est l'opération qui consiste dans une matrice à remplacer une ligne par la somme de cette ligne et du produit d'une autre ligne par un réel :

$$Li \longleftarrow Li + \alpha Lj \ \textbf{avec} \ \ i \neq j$$

Dans la suite, on notera $t(i, \alpha, j)$ cette transvection.
C'est précisément ce type d'opération qu'on exécute lorsqu'on résout un système linéaire. D'un point de vue matriciel, cette opération se traduit par le produit à gauche par la matrice : $Id_n + \alpha E_{i,j}$ où $E_{i,j}$ est la matrice élémentaire. Ce type de matrice possède des propriétés remarquables :

1. **Déterminant :** il vaut 1 car on a une matrice triangulaire dont tous les éléments diagonaux sont égaux à 1.

2. **Inverse :** soit un couple (i,j) avec $i \neq j$

$$(Id_n + \alpha E_{i,j}) * (Id_n - \alpha E_{i,j}) = Id_n - \alpha^2 E_{i,j} * E_{i,j} = Id_n$$

Ce qui prouve que la matrice inverse d'une transvection $t(i,\alpha,j)$ est la transvection $t(i,-\alpha,j)$

3. **Produit d'une matrice par une matrice de transvection** :
Le produit par une matrice de transvection n'est pas commutatif !
   - produit à gauche : $T * A$
     Comme on l'a déjà vu, multiplier à gauche par $T(i,\alpha,j)$ a pour effet de modifier la ligne Li d'indice i de M :
$$Li \longleftarrow Li + \alpha Lj$$
   - produit à droite : $A * T$
     La transvection agit alors sur les colonnes de la matrice$M$, la colonne Ci d'indice i devient :
$$Ci \longleftarrow Ci + \alpha Cj$$

En conclusion, pour calculer le produit d'une matrice A par une matrice de transvection, aucun calcul n'est nécessaire, il suffit de réécrire la matrice A en modifiant une ligne ou une colonne.

Cas particulier : produit de certaines transvections : soient deux transvections correspondant à des opérations élémentaires utilisées dans l'algorithme du pivot de Gauss lorsqu'on fait apparaître des zéros sous un pivot :
$t(i,\alpha,j)$ et $t(k,\alpha,l)$ avec $j \neq k$
alors le produit matriciel : $E_{i,j}E_{k,l} = \delta(j,k)E_{i,l}$ est nul, d'où :

$$(Id_n + \alpha E_{i,j}) * (Id_n + \beta E_{k,l}) = Id_n + \alpha E_{i,j} + \beta E_{k,l} + \alpha\beta E_{i,j}E_{k,l} = Id_n + \alpha E_{i,j} + \beta * E_{k,l}$$

Le produit de 2 transvections n'est pas une transvection ! Mais le produit de 2 matrices de transvection est très facile à écrire : à partie de Id_n, il suffit de placer les deux coefficients α et β.
Par exemple :

$$M = \begin{pmatrix} 1 & 0 & 0 \\ \mathbf{2} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mathbf{5} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \mathbf{2} & 1 & 0 \\ \mathbf{5} & 0 & 1 \end{pmatrix}$$

ce qui entraîne pour la matrice inverse :

$$M^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\mathbf{5} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\mathbf{2} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -\mathbf{2} & 1 & 0 \\ -\mathbf{5} & 0 & 1 \end{pmatrix}$$

ou encore :

$$N = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \mathbf{a} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \mathbf{b} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \mathbf{c} & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \mathbf{a} & 1 & 0 & 0 \\ \mathbf{b} & 0 & 1 & 0 \\ \mathbf{c} & 0 & 0 & 1 \end{pmatrix}$$

$$N^{-1} = [\begin{pmatrix}1&0&0&0\\ \mathbf{a}&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix} * \begin{pmatrix}1&0&0&0\\0&1&0&0\\ \mathbf{b}&0&1&0\\0&0&0&1\end{pmatrix} * \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\ \mathbf{c}&0&0&1\end{pmatrix}]^{-1} =$$

$$\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\-\mathbf{c}&0&0&1\end{pmatrix} * \begin{pmatrix}1&0&0&0\\0&1&0&0\\-\mathbf{b}&0&1&0\\0&0&0&1\end{pmatrix} * \begin{pmatrix}1&0&0&0\\-\mathbf{a}&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix} = \begin{pmatrix}1&0&0&0\\-\mathbf{a}&1&0&0\\-\mathbf{b}&0&1&0\\-\mathbf{c}&0&0&1\end{pmatrix}$$

Conséquence pour le pivot de Gauss :
Pour faire apparaître des zéros sous le pivot il suffit de multiplier à gauche par une matrice de la forme :

$$E_j = \begin{pmatrix}1 & & & & \\ & & 1 & & 0 & \\ & & \alpha_{j+1,j} & 1 & & \\ & 0 & \vdots & & \ddots & \\ & & \alpha_{n,j} & & & 1\end{pmatrix}$$

dont l'inverse s'écrit sans calculs, en rajoutant quelques signes moins !

$$\begin{pmatrix}1 & & & & \\ & & 1 & & 0 & \\ & & -\alpha_{j+1,j} & 1 & & \\ & 0 & \vdots & & \ddots & \\ & & -\alpha_{n,j} & & & 1\end{pmatrix}$$

3 Décomposition PLU

On peut faire apparaître les matrices P, L et U grâce à l'algorithme du pivot de Gauss.
U est la plus facile à obtenir car il s'agit de la matrice échelonnée classique obtenue grâce à cet algorithme. L'obtention de P et de L demande plus de travail ; ces 2 matrices s'obtiennent comme produits de diverses matrices de permutations et de transvections. D'où le travail préparatoire réalisé dans les paragraphes précédents...

Pour commencer, plaçons nous dans un cas particulier :

3.1 si lors de l'application de Gauss-Jordan les pivots sont bien placés

Par exemple :

$$A = \begin{pmatrix} \boxed{4} & -9 & 2 \\ 2 & -4 & 4 \\ -1 & 2 & 2 \end{pmatrix}$$

Dans la première colonne, on utilise le pivot 4 pour faire apparaître deux 0 sous le pivot. Cela revient à multiplier successivement à gauche par les matrices des 2 transvections suivantes : $L2 \longleftarrow L2 - \frac{1}{2}L1$ et $L3 \longleftarrow L3 + \frac{1}{4}L1$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{4} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} A = \begin{pmatrix} \boxed{4} & -9 & 2 \\ 0 & \boxed{\dfrac{1}{2}} & 3 \\ 0 & -\frac{1}{4} & \frac{5}{2} \end{pmatrix}$$

On poursuit en faisant apparaître un 0 sous le pivot $\frac{1}{2}$ de la colonne 2 en multipliant à gauche par la matrice de la transvection $L3 \longleftarrow L3 + \frac{1}{2}L2$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{4} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} A = \begin{pmatrix} \boxed{4} & -9 & 2 \\ 0 & \boxed{\dfrac{1}{2}} & 3 \\ 0 & 0 & \boxed{4} \end{pmatrix}$$

On multiplie à gauche par l'inverse de chaque matrice :

$$A = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{4} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} 4 & -9 & 2 \\ 0 & \frac{1}{2} & 3 \\ 0 & 0 & 4 \end{pmatrix}$$

d'où :

$$A = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -\frac{1}{4} & -\frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} 4 & -9 & 2 \\ 0 & \frac{1}{2} & 3 \\ 0 & 0 & 4 \end{pmatrix}$$

d'où le résultat recherché :

$$A = PLU \text{ avec } P = Id_3,\ \ L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -\frac{1}{4} & -\frac{1}{2} & 1 \end{pmatrix} \text{ et } U = \begin{pmatrix} 4 & -9 & 2 \\ 0 & \frac{1}{2} & 3 \\ 0 & 0 & 4 \end{pmatrix}$$

Revenons maintenant au cas général :

3.2 si dans certaines colonnes, le pivot n'est pas à sa place...

2 possibilités :

— soit il n'y a pas de pivot dans la colonne courante : dans ce cas on ne fait rien et on poursuit avec la colonne suivante

— soit un échange de 2 lignes est nécessaire pour bien positionner le pivot : dans ce cas, il suffit de multiplier à gauche par une matrice de permutation comme on l'a vu précédemment.

Ainsi l'algorithme de Gauss Jordan se traduit matriciellement par une suite de multiplications (à gauche) de A par
P_1 puis E_1 pour la colonne 1 , ensuite par P_2 puis par E_2 pour la colonne 2 etc jusqu'à la colonne $n-1$.
On obtient donc :

$$U = E_{n-1}P_{n-1}\cdots E_2P_2E_1P_1A$$

On pose alors :

$$P = P_{n-1}\cdots P_2P_1 \text{ et } L = P^{-1}\big(E_{n-1}P_{n-1}\cdots E_2P_2E_1P_1\big)^{-1}$$

soit encore :

$$L = P^{-1}\big(P_1^{-1}E_1^{-1}P_2^{-1}E_2^{-1}\cdots P_{n-1}^{-1}E_{n-1}^{-1}\big)$$

L s'exprime donc comme le produit de matrices inverses de permutations ou de transvections décrites au début de ce document.
Vérification :
on a $L = P^{-1}\big(E_{n-1}P_{n-1}\cdots E_2P_2E_1P_1\big)^{-1} \Rightarrow PL = \big(E_{n-1}P_{n-1}\cdots E_2P_2E_1P_1\big)^{-1}$
$\Rightarrow E_{n-1}P_{n-1}\cdots E_2P_2E_1P_1 = L^{-1}P^{-1}$

$$\text{d'où : } U = E_{n-1}P_{n-1}\cdots E_2P_2E_1P_1A \Rightarrow U = L^{-1}P^{-1}A \Rightarrow PLU = A$$

3.3 Exemple

Soit la matrice :

$$A = \begin{pmatrix} \boxed{2} & 4 & 2 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 2 & 1 & 3 \\ 1 & 2 & 2 & 1 \end{pmatrix}$$

Traitement de la colonne 1 : $P_1 = Id_4$ et $E_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ -1/2 & 0 & 1 & 0 \\ -1/2 & 0 & 0 & 1 \end{pmatrix}$

$E_1P_1A = \begin{pmatrix} \boxed{2} & 4 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 1 \end{pmatrix}$

Traitement de la colonne 2 : pas de pivot dans cette colonne, on ne fait rien $P_2 = E_2 = Id_4$
Traitement de la colonne 3 : on place le pivot en échangeant les lignes 2 et 4 grâce à la permutation $p_3 = (1,4,3,2)$ de matrice $P_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$. Une fois le pivot placé, on constate que le reste de la colonne est nul donc rien de plus à faire : $E_3 = Id_4$.
On a :

$$P_3E_3E_1P_1A = \begin{pmatrix} \boxed{2} & 4 & 2 & 0 \\ 0 & 0 & \boxed{1} & 1 \\ 0 & 0 & 0 & \boxed{3} \\ 0 & 0 & 0 & 0 \end{pmatrix} = U$$

Traitement de la colonne 4 : dans cet exemple, il se trouve que le pivot est déjà bien placé mais d'une manière générale quelque soit le contenu de la dernière colonne on est certain d'avoir une matrice triangulaire supérieure.
On obtient les matrices recherchées :

$$P = P_3P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$L = P^{-1}P_1^{-1}E_1^{-1}P_3^{-1}E_3^{-1} = P_3 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 1/2 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & 1 \end{pmatrix} P_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1 \\ 1/2 & 0 & 1 & 0 \\ 1/2 & 1 & 0 & 0 \end{pmatrix} P_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 1/2 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & 1 \end{pmatrix}$$

d'ù la décomposition :

$$A = \begin{pmatrix} 2 & 4 & 2 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 2 & 1 & 3 \\ 1 & 2 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 1/2 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boxed{2} & 4 & 2 & 0 \\ 0 & 0 & \boxed{1} & 1 \\ 0 & 0 & 0 & \boxed{3} \\ 0 & 0 & 0 & 0 \end{pmatrix} = PLU$$

Pour une matrice carrée A de taille n, l'algorithme général est le suivant :

```
lig = 0 indice de la ligne courante
col = 0 indice de la colonne courante
P = Id initialisation de P
L = Id initialisation de L
Tant que lig < n et col < n
        Chercher dans la colonne col à partir de la ligne lig un coefficient non nul appelé pivot
        Si on n'en a pas trouvé :
                col++
        Sinon soit lp la ligne du pivot
                P_aux = MatEch(lig, lp) matrice qui échange les deux lignes
                A = P_aux * A Echanger les lignes lig et lp dans A
                P = P_aux * P Mise à jour de P
                L = L * P_aux^(-1) Mise à jour de L
                pivot = A[lig][col]
                Pour toutes les lignes i allant de lig + 1 à n - 1 :
                        A = T(i, -A[i][col]/pivot, lig) * A Faire apparaître 0 à la place de A[i][col]
                        L = L * T(i, -A[i][col]/pivot, lig)^(-1) Mise à jour de L
                lig + +
                col + +
L = P^(-1) * L
U = A
```

Dans cet algorithme, les produits de matrice se font exclusivement avec les fonctions spécifiques écrites précédemment :
ProduitPermutG,ProduitPermutD,ProduitTransvectionG et ProduitTransvectionD.

4 Application de la décomposition PLU :

Cette décomposition peut être efficace pour calculer le déterminant d'une matrice , pour calculer son inverse ou bien pour résoudre des systèmes d'équations linéaires.

4.1 Utilité de la décomposition $A = PLU$ pour la résolution de systèmes

Lorsqu'on doit résoudre le sytème d'équations linéaires $AX = B$ pour plusieurs valeurs du second memnre B, il est opportun de factoriser A sous forme PLU en effet le système initial se ramène à la résolution de 2 systèmes triangulaires :
La résolution du système $AX = B$ équivaut à $PLUX = B$ se ramène à la résolution des 2 systèmes linéaires triangulaires :

- On résout pour commencer le système triangulaire inférieur :$LY = P^T B$ d'inconnue Y
- puis, une fois connu Y, il reste à résoudre le système triangulaire supérieur d'inconnue $X : UX = Y$

Dans la suite, on se restreint à un système triangulaire pour lequel tous les éléments diagonaux $a_{i,i}$ c'est à dire tous les pivots sont nopn nuls.
Un tel système triangulaire admet une solution unique , c'est un système de Cramer et l'algorithme de résolution est très simple :

1. **Cas d'une matrice triangulaire T supérieure** : système $TX = B$ avec

$$T = \begin{pmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n} \\ 0 & t_{2,2} & & t_{2,n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & t_{n,n} \end{pmatrix}, X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \text{ et } B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

   On résout les équations en commençant par la dernière et en remontant :

$$\begin{aligned} x_n &= \frac{b_n}{t_{n,n}} \\ x_{n-1} &= \frac{b_{n-1} - t_{n-1,n}x_n}{t_{n-1,n-1}} \\ &\vdots \\ x_{n-k} &= \frac{b_{n-k} - \sum_{j=n-k+1}^{n} t_{n-k,j}x_j}{t_{n-k,n-k}} \end{aligned}$$

   c'est à dire que, pour tout indice i compris entre 1 et n : $x_i = \frac{b_i - \sum_{j=i+1}^{n} t_{i,j}x_j}{t_{i,i}}$

   L'algorithme de résolution est donc :

$X[n-1] = B[n-1]/T[n-1][n-1]$
Pour i allant de n-2 à 0 :
    Calculer la somme $S = T[i][i+1] * X[i+1] + \cdots + T[i][n-1] * X[n-1]$
    $X[i] = (B[i] - S)/T[i][i]$

2. **Cas d'une matrice triangulaire T inférieure :** système $TX = B$ avec

$$T = \begin{pmatrix} t_{1,1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ t_{n-1,n} & & t_{n-1,n-1} & 0 \\ t_{n,1} & \cdots & t_{n,n-1} & t_{n,n} \end{pmatrix}, X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

On résout les équations en commençant par la première et en descendant :

$$\begin{aligned} x_1 &= \frac{b_1}{t_{1,1}} \\ x_2 &= \frac{b_2 - t_{2,1}x_1}{t_{2,2}} \\ &\vdots \\ x_i &= \frac{b_i - \sum_{j=1}^{i-1} t_{i,j}x_j}{t_{i,i}} \end{aligned}$$

L'algorithme de résolution est donc :

$X[0] = B[0]/T[0][0]$
Pour i allant de 1 à n-1 :
    Calculer la somme $S = T[i][0] * X[0] + \cdots + T[i][i-1] * X[i-1]$
    $X[i] = (B[i] - S)/T[i][i]$

Par exemple, pour résoudre le système $AX = B$ avec $A = \begin{pmatrix} 1 & 1 \\ 2 & -2 \end{pmatrix}$ et $B = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$

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

- On résout d'abord le système : $LY = P^T B$, d'inconnue Y :
  $\begin{pmatrix} 2 & 0 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} y1 \\ y2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \Rightarrow \begin{cases} y1 = \frac{3}{2} \\ y2 = \frac{1-y1}{2} = -\frac{1}{4} \end{cases}$
- Et, pour finir, le système : $UX = Y$:
  $\begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x1 \\ x2 \end{pmatrix} = \begin{pmatrix} y1 = \frac{3}{2} \\ y2 = -\frac{1}{4} \end{pmatrix} \Rightarrow \begin{cases} x1 = \frac{3}{2} + x2 = \frac{5}{4} \\ x2 = -\frac{1}{4} \end{cases}$

5 Travail à réaliser

Mise en garde : Vous ne devez jamais utiliser les fonctions de calcul de produit, de déterminant ou d'inverse vues en TP. L'objet de la décomposition PLU est d'éviter l'utilisation de ces algorithmes complexes !

5.1 Matrices de permutation :

Ecrire les fonctions suivantes :

- PermutationAleatoire qui prend un entier $n > 0$ et renvoie le n-uplet d'une permutation aléatoire.
- MatricePermutation qui prend en paramètre une permutation sous forme de n-uplet et renvoie la matrice de la permutation associée.
- PermutationAssociée qui prend une matrice de permutation et renvoie le n-uplet associé. C'est la fonction réciproque de MatricePermutation
- EstPermutation qui prend une matrice et renvoie un booléen vrai si et seulement s'il s'agit d'une matrice de permutation.
- MatricePermutationInverse qui prend une matrice de permutation et renvoie sa matrice inverse (c'est à dire sa matrice transposée).
- ProduitPermutG qui pour une matrice de permutation P et une matrice M renvoie la matrice produit : $P * M$
- ProduitPermutD qui pour une matrice de permutation P et une matrice M renvoie la matrice produit : $M * P$

5.2 Transvection :

Ecrire les fonctions suivantes :

- MatriceTransvection qui prend en paramètre les caractéristiques d'une transvection $t(i, \alpha, j)$ et une taille n et renvoie la matrice carrée de taille n de la transvection associée.
- EstTransvection qui prend une matrice et renvoie un booléen vrai si et seulement s'il s'agit d'une matrice de transvection.
- MatriceTransvectionInverse qui prend une matrice de transvection et renvoie sa matrice inverse.
- ProduitTransvectionG qui pour une matrice de transvection T et une matrice M renvoie la matrice produit : $T * M$.
- ProduitTransvectionD qui pour une matrice de transvection T et une matrice M renvoie la matrice produit : $M * T$.

5.3 Décomposition PLU :

Ecrire les fonctions suivantes :

- TriangulaireSup qui prend une matrice et renvoie un booléen vrai si et seulement si la matrice est triangulaire supérieure.
- TriangulaireInf qui prend une matrice et renvoie un booléen vrai si et seulement si la matrice est triangulaire inférieure.
- DecompositionPLU qui prend une matrice carrée A et renvoie 3 matrices P , L et U solutions du problème de décomposition en PLU.

Tester en utilisant SAGE , votre décomposition sera peut être différente du résultat retourné par SAGE, mais c'est normal car il n'y a pas unicité de la décomposition...

5.4 Application de la décomposition PLU :

1. Calcul du déterminant :
   Ecrire la fonction DeterminantPLU qui prend une matrice carrée M, et renvoie son déterminant. Le calcul doit de faire à partir de la décomposition $M = PLU$ correspondante.
2. Résolution de systèmes de Cramer :
   Ecrire les fonctions :
   · ResolutionTriSupCramer(A,B) qui renvoie la solution du système $AX = B$ où A est une matrice triangulaire supérieure.
   · ResolutionTriInfCramer(A,B) qui renvoie la solution du système $AX = B$ où A est une matrice triangulaire inférieure.
   · ResolutionCramer(A,B) qui renvoie la solution d'un système de Cramer $AX = B$
   Tester avec un système de 5 équations à 5 inconnues $AX = B$ de votre choix pour lequel la décomposition PLU donnée par SAGE est différente de la votre. Résoudre successivement le système avec les 2 décompositions et constater que la solution est la même.

6 Notation

- Codage : 14 pts
  — Permutations : 4
  — Transvections : 3
  — Décomposition PLU : 5
  — Applications : 4 (2 points bonus)
- Tests : vous rendrez 2x4=8 fichiers. Pour chaque partie, un fichier de fonctions et un fichier de tests :Permutations.py, TestsPermutations.py, Transvections.py, TestsTransvections.py, Decomposition.py, TestsDecomposition.py, Applications.py, TestsApplications.py
  Au maximum la note de codage peut être diminuée de 5 pts selon la qualité des tests unitaires.
- Forme : 2pts
  Explications dans le code quand cela est nécessaire, cartouches pour les fonctions (décrivant les Entrées, les Sorties et l'action accomplie), choix des identificateurs...
- Document texte explicatif : 4pts
  Il doit être au format pdf, contenir 4 pages maximum. Dans ce document, vous préciserez :
  1. qui a codé fait quoi ? comment avez-vous travaillé ensemble ?
  2. qu'est ce qui a été traité ? testé ? qu'est ce qui est fonctionnel ?
  3. quelles sont les limitations du programme : temps d'exécution, taille des matrices...
  4. les problèmes (de tous ordres) rencontrés.
  5. toute autre chose que vous jugerez utile.

◇ ◇ ◇◇ ◇ ◇◇