Ce TP vous invite à programmer la méthode du pivot de Gauss et la méthode de Gauss-Jordan.

1 Rappel : Pivot de Gauss

L'objectif est de résoudre le système $AX = B$, d'inconnue X avec A et B données du problème.
Pour rappel, les 2 opérations autorisées sont les suivantes :

— Échange de 2 lignes.

— Ajout d'un multiple d'une ligne à une autre.

L'algorithme du pivot de gauss fonctionne alors comme suit : on transforme A en matrice triangulaire supérieure en sélectionnant pour la colonne 1 un coefficient non nul (en général le plus grand en valeur absolue). On remonte cette ligne en première dans la matrice, puis on annule le premier coefficient de toutes les autres lignes en y ajoutant le bon multiple de cette ligne. On répète le processus avec les colonnes suivantes : Pour la colonne i, on choisit un coefficient non nul dans cette colonne entre les lignes i et n. On remonte cette ligne en $i$ème place et on annule le ième coefficient des lignes $i+1$ à n en ajoutant un multiple de la i-ème ligne. Si A était inversible, il y a toujours un coefficient non nul par colonne et donc on obtient ainsi une matrice A' triangulaire supérieure avec aucun 0 sur la diagonale. On effectue en même temps les mêmes calculs à B pour obtenir B'. Voir exemple ci-dessous

$$A, B = \begin{pmatrix} 1 & 3 & 4 & 1 \\ 2 & 4 & 3 & 4 \\ 1 & 1 & 2 & 5 \\ 1 & 4 & 5 & 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 3 \\ 3 \end{pmatrix} \to \begin{pmatrix} 2 & 4 & 3 & 4 \\ 1 & 3 & 4 & 1 \\ 1 & 1 & 2 & 5 \\ 1 & 4 & 5 & 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \\ 3 \\ 3 \end{pmatrix} \to \begin{pmatrix} 2 & 4 & 3 & 4 \\ 0 & 1 & 2.5 & -1 \\ 0 & -1 & 0.5 & 3 \\ 0 & 2 & 3.5 & 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \\ 2 \\ 2 \end{pmatrix} \to \begin{pmatrix} 2 & 4 & 3 & 4 \\ 0 & 2 & 3.5 & 0 \\ 0 & 1 & 2.5 & -1 \\ 0 & -1 & 0.5 & 3 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \\ 0 \\ 2 \end{pmatrix}$$

$$\to \begin{pmatrix} 2 & 4 & 3 & 4 \\ 0 & 2 & 3.5 & 0 \\ 0 & 0 & 0.75 & -1 \\ 0 & 0 & 2.25 & 3 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \\ -1 \\ 3 \end{pmatrix} \to \begin{pmatrix} 2 & 4 & 3 & 4 \\ 0 & 2 & 3.5 & 0 \\ 0 & 0 & 2.25 & 3 \\ 0 & 0 & 0.75 & -1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \\ 3 \\ -1 \end{pmatrix} \to \begin{pmatrix} 2 & 4 & 3 & 4 \\ 0 & 2 & 3.5 & 0 \\ 0 & 0 & 2.25 & 3 \\ 0 & 0 & 0 & -2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \\ 3 \\ -2 \end{pmatrix}$$

Une fois la matrice triangulaire, le système peut s'écrire $A'X = B'$ avec A' triangulaire supérieure et B' obtenue par cette méthode. Le système peut être résolu de manière itérative : x_n apparaît seul dans une équation. Une fois x_n connu, x_{n-1} apparaît seul avec x_n dans une équation, on continue ainsi, car chaque variable x_i apparaît dans une équation avec uniquement des variables x_j avec $j > i$ puisqu'il n'y a pas de 0 sur la diagonale de A'.

2 Fonctions de bases

Les matrices que vous manipulerez seront des variables globales afin de pouvoir simplement les modifier.

Question 0

Écrire une fonction *affiche_ mat* (resp. *affiche_ vect*) de type *void* qui prend en entrée une matrice A (resp. un vecteur X) et qui affiche A (resp. X). Ces fonctions seront utiles pour effectuer des tests. Les essayer en rentrant les matrices A et B donnés dans l'exemple ci-dessus.

Question 1

Écrire une fonction *mult_ vec* de type *void* qui prend en entrée une matrice carrée A et deux vecteurs X et Y et qui calcul dans Y le produit AX.

Question 2

Écrire une fonction *mult_ mat* de type *void* qui prend en entrée 3 matrice carrée A, X et Y et qui calcul dans Y le produit AX.

3 Pivot de Gauss

Question 3

Écrire une fonction *trouve_max* de type *int* qui prend en entrée une matrice A et un entier i et qui renvoie un indice plus grand que i contenant le plus grand coefficient de la colonne i (parmi les lignes de i à n).

Question 4

Écrire une fonction *echange_ligne* de type *void* qui prend en entrée une matrice A et un vecteur B et deux entiers $l1$ et $l2$ et qui échange les lignes $l1$ et $l2$ dans A et dans B.

Question 5

Écrire une fonction *mult_scal* de type *void* qui prend en entrée une matrice A et un vecteur B, deux entiers $l1$ et $l2$ et double s et qui remplace la ligne L^A_{l1} de A par $L^A_{l1} + s * L^A_{l2}$, de même, cette fonction remplace la ligne L^B_{l1} de B par $L^B_{l1} + s * L^B_{l2}$.

Question 6

Écrire une fonction *pivot* de type *void* qui prend en entrée une matrice A et un vecteur B et qui calcule A' et B' de sorte que $AX = B$ si et seulement si $A'X = B'$ avec A' triangulaire supérieure.

Question 7

Écrire une fonction *gauss* de type *void* qui prend en entrée une matrice A et deux vecteurs B et X, de sorte qu'après un appel à cette fonction, X soit une solution de $AX = B$.

4 Inversion de matrices

Si l'on dispose de $N >> n$ vecteurs $B_1, \ldots, B_N$, et qu'on souhaite résoudre $AX_i = B_i$ pour $X_1, \ldots, X_N$ des inconnues, notre méthode nous force à recalculer un pivot pour chaque B_i alors que les calculs effectués sont souvent les mêmes.On va donc calculer l'inverse de A à l'aide du pivot de Gauss afin de pouvoir trouver les vecteurs X_i simplement en calculant $X_i = A^{-1}B_i$.

Question 8

En utilisant le pivot de gauss que vous avez déjà écrit, écrire une fonction *inverse* de type *void* qui prend en entrée deux matrices carrées A et $A1$ et qui calcul A^{-1} et le stock dans $A1$. On pourra calculer les inverses des différentes colonnes de I_n.

Question 9

La méthode précédente effectue plusieurs fois certains calculs. Une solution efficace est d'effectuer les calculs de la Section 2 directement sur l'identité et non sur ses colonnes une à une. Écrire une fonction *inverse_rapide* de type *void* qui prend en entrée deux matrices carrées A et $A1$ et qui calcul A^{-1} et le stock dans $A1$ en appliquant cette méthode. On pourra s'aider de fonctions auxiliaires inspirées de celles de la Section 2.