Le but de ce TP est de parvenir à calculer la décomposition QR d'une matrice. Vous rendrez ce TT sous forme d'un unique fichier "[nom]_[prenom].c" par mail à *nacim.oijid@univ-lyon1.fr* le 03 février 2023 au plus tard.

La clarté du code et les commentaires feront partie de la notation. Il est recommandé de tester les différentes fonctions.

1 Méthode de Gram-Schmidt et décomposition QR

1.1 Méthode de Gram-Schmidt

Dans ce TP, on se place dans $\mathbb{R}^n$ muni du produit scalaire canonique $\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$.

Contrairement à d'habitude, on stockera les matrices sous forme de liste de colonnes et nom de liste de lignes : si M est une matrice, $M[i]$ doit être la i-ème colonne de M.

L'algorithme de Gram-Schmidt consiste à transformer une base $(x_1, \dots, x_n)$ de $\mathbb{R}^n$ en l'unique base orthonormale $(e_1, \dots, e_n)$ telle que pour $1 \leq i \leq n$, $(x_1, \dots, x_i)$ et $(e_1, \dots, e_i)$ génèrent le même espace, et $(e_i, x_i) > 0$.

On rappelle qu'une base orthonormale vérifie pour $1 \leq i < j \leq n$, $\langle e_i, e_i \rangle = 1$ et $\langle e_i, e_j \rangle = 0$.

Les différentes bases de $\mathbb{R}^n$ seront stockées sous forme de matrices.

1.2 Décomposition QR

Soit M une matrice. Une décomposition QR de M est l'écriture de M sous forme de deux matrices Q et R tel que $M = QR$, $Q = (a_{i,j})$ soit orthonormale (i.e. ${}^tQQ = I_n$) et R soit triangulaire supérieure. Si $Q = (e_1, \dots, e_n)$ est obtenue en appliquant la méthode de Gram-Schmidt à M, on a alors $R = (r_{i,j}) = \langle (e_i, a_j) \rangle$.

2 Préliminaires

Comme lors du TP sur la méthode de Strassen, vous êtes invités à trouver vous les vous-même les fonctions auxiliaires si necessaires que vous souhaitez utiliser.

Comme d'habitude, on posera $N = 8$ à l'aide d'un $\#define$ et on ne manipulera que des matrices de taille N. Les matrices seront stockées sous forme de double**, et les fonctions de manipulations de matrices seront de type void.

Question 1 Écrire une fonction *affiche_matrice* qui prend en entrée une matrice et l'affiche.

Question 2 Écrire une fonction *produit_scalaire* de type double, qui prend en entrée deux vecteurs et calcul leur produit scalaire.

Question 3 Écrire une fonction *est_orthonormale* de type int qui prend en entrée une matrice et renvoie 1 si la matrice est orthonormale, sinon on renverra 0. Pour rappel, une matrice est orthonormale si son inverse est égale à sa transposée.

Question 4 Écrire une fonction *test_QR* de type int qui prend en entrée trois matrices M, Q et R et renvoie 1 si (Q, R) est la décomposition QR de la matrice M, sinon on renverra 0. On se souvient qu'il faut vérifier $M = QR$, Q orthonormale et R triangulaire supérieure.

3 Algorithmes

Question 4 Écrire une fonction *Gram_Schmidt* qui prend en entrée une matrice M et une matrice Q et qui stock dans Q la matrice obtenue en appliquant la méthode de Gram-Schmidt à M.

Question 5 Écrire une fonction *decompose* qui prend en entrée une matrice M, matrice Q et une matrice R et qui stock dans Q et R les matrices obtenues en effectuant la décomposition QR de M.

Question 6 Écrire une fonction *resout* qui prend en entrée une matrice M et un vecteur b et calcul x tel que $Mx = b$. en utilisant la décomposition QR de M.