



5A ModIA

Rapport de TP

---

# Algèbre Linéaire Creuse : méthodes directes

---

*Elèves :*

Karima GHAMNIA  
Cassandra MUSSARD

*Enseignant :*

Ronan GUIVARCH

7 avril 2024

# 1 Introduction

Résoudre un système linéaire  $Ax = b$ , avec  $A$  une matrice creuse, nécessite de minimiser le nombre de valeurs non nulles dans les facteurs de matrice grâce à des stratégies de réordonnement. En effet, il est très important de garder la structure creuse de la matrice pour conserver les avantages de celle-ci (minimisation du stockage en mémoire des valeurs non nulles, accélération du temps de calcul).

L'objectif de ce TP est d'identifier la stratégie de réordonnement la plus efficace pour chaque système.

## 2 Nombre d'opérations de la phase de résolution

### 2.1 Pour une factorisation de Cholesky

La factorisation de Cholesky consiste à trouver  $L$  tel que :

$$A = LL^T$$

Avec  $A$ , une matrice symétrique définie positive, et  $L$ , une matrice triangulaire inférieure. Dans ce cas, la résolution de  $Ax = b$  revient à résoudre les 2 systèmes linéaires simples :

$$Ly = b$$

$$L^T x = y$$

La résolution d'un système linéaire triangulaire implique une opération pour chaque terme sur la diagonale et deux termes pour les autres coefficients. Le nombre total d'opérations flottantes nécessaires à la résolution des deux systèmes linéaires triangulaires est calculé comme suit :

- $L^T x = y$  :  $2nnz(L) - n$
- $Ly = b$  :  $2nnz(L) - n$
- Nombre total :  $4nnz(L) - 2n$

### 2.2 Pour une factorisation LU

Maintenant analysons le nombre d'opérations nécessaires pour la factorisation LU. On doit trouver  $U$  et  $L$  tel que :

$$A = LU$$

Pour cette factorisation on doit résoudre les systèmes suivants :

$$Ly = b$$

$$Ux = y$$

Comme précédemment les deux systèmes sont triangulaires donc le nombre d'opérations est le suivant :

- $Ux = y$  :  $2nnz(U) - n$
- $Ly = b$  :  $2nnz(L) - 2n$
- Nombre total :  $2nnz(L) - 2n + 2nnz(U) - n$

Pour le système  $Ly = b$  on a un terme en  $2n$  car sur les termes de la diagonale de  $L$  sont égaux à 1.

### 3 Résolution du système linéaire symétrique

Dans cette partie, nous testons différentes permutations pour trouver celle qui permet de diminuer le remplissage (nombre de non zéros  $nnz$ ) et le nombre d'opérations dans la résolution avec une factorisation LU et Cholesky sur différentes matrices. les matrices : mat0, mat1, mat2, mat3 et BCSSTK27.

Les différentes permutations utilisées sont les suivantes :

- Aucune.
- symamd (Approximation symétrique de l'algorithme du degré minimum)
- symrcm (Algorithme de Cuthill-McKee inverse).
- amd (Approximation de l'algorithme du degré minimum)
- colamd (Approximation par colonne de l'algorithme du degré minimum).
- colperm (Permutation des colonnes basée sur la comptage de non zéros).

### 3.1 Avec une factorisation de Cholesky

Nous appliquons la factorisation de Cholesky sur les matrices : mat0, mat1, mat2, mat3 et BCSSTK27.

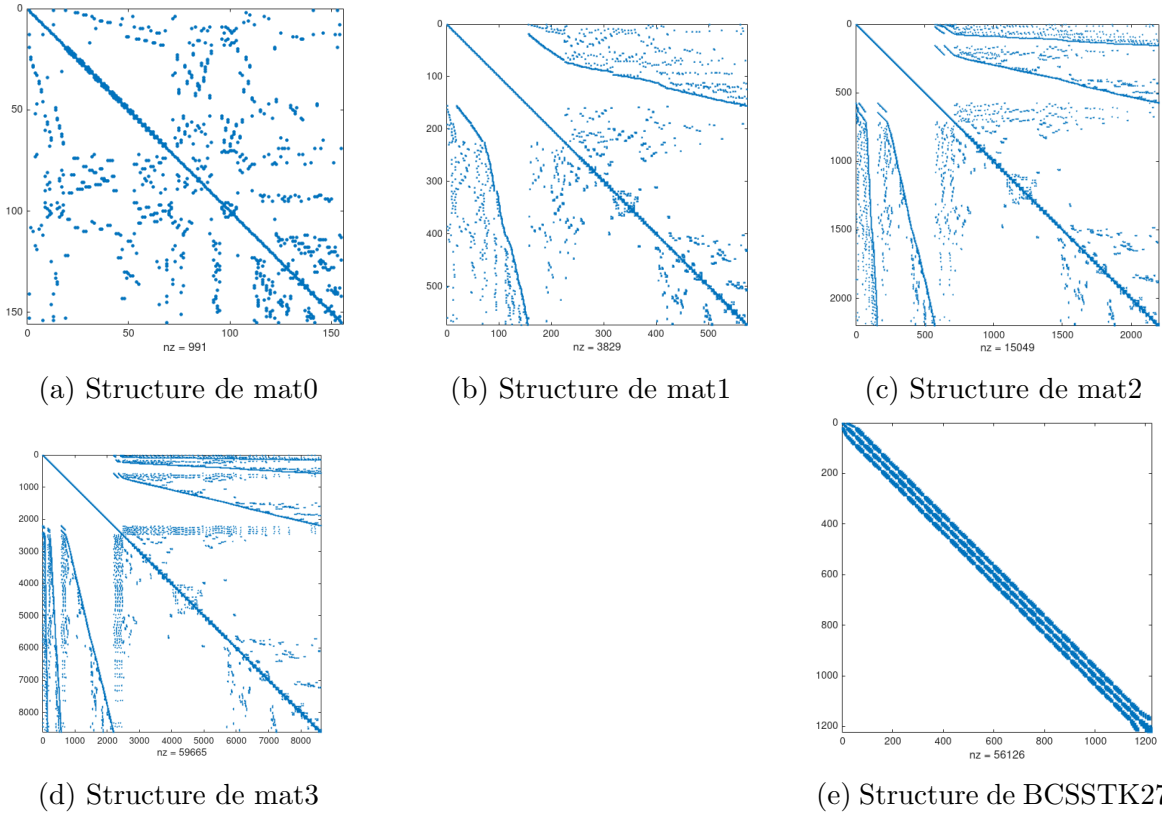


FIGURE 1 – La structure des différentes matrices étudiées avec une factorisation de Cholesky

Dans cette partie les matrices que nous avons à notre disposition sont des matrices symétriques, donc l'utilisation de la factorisation de Cholesky ainsi que des méthodes de réordonnancement basées sur des matrices symétriques est justifiée. Le tableau 5 présente le nombre d'opérations pour la résolution avec une factorisation de Cholesky sur la matrice permutée pour chacune des matrices étudiées.

Nous pouvons remarquer que chaque matrice a une permutation qui lui correspond le mieux. Par exemple, pour mat0 c'est la permutation "amd" qui donne le meilleur résultat, alors que pour BCSSTK27, aucune méthode de réordonnancement n'est efficace car c'est une matrice tridiagonale 1.

Pour évaluer la qualité de la solution obtenue, nous utilisons 2 métriques :

TABLE 1 – Nombre d'opérations pour la résolution avec une factorisation de Cholesky avec différents algorithmes de réordonnancement

Réordonnancement	mat0	mat1	mat2	mat3	BCSSTK27
aucun	15554	140754	1307170	11239574	199612
symamd	4434	27046	161702	907646	232972
symrcm	5938	41074	303202	2344106	201052
amd	4410	26958	170726	1004546	220548
colamb	5158	33998	197114	1207442	203208
colperm	12502	143854	1272602	11239574	1311532

- **Erreur directe** : Elle représente l'écart entre la solution obtenue et la solution exacte de notre problème

$$Err = \frac{||x_{exact} - x||}{||x_{exact}||}$$

- **Erreur inverse** : Elle représente l'amplitude des perturbations associée à l'erreur directe

$$Err = \frac{||b - Ax_1||}{||b||}$$

TABLE 2 – Erreurs inverses de la résolution avec une factorisation de Cholesky avec différents algorithmes de réordonnancement

Réordonnancement	mat0	mat1	mat2	mat3	BCSSTK27
aucun	2.4309e+03	7.9555e+03	1.0638e+04	1.7381e+04	2.7433e+07
symamd	1.2042e-12	7.7481e-12	3.0990e-11	1.0873e-10	4.4940e-14
symrcm	1.4392e-12	8.7598e-12	4.2393e-11	1.7213e-10	3.5509e-14
amd	1.5298e-12	5.5497e-12	2.7188e-11	1.1690e-10	3.6027e-14
colamd	1.7170e-12	8.0905e-12	3.0935e-11	1.2809e-10	3.8606e-14
colperm	9.3378e-13	6.9890e-12	2.9388e-11	1.2655e-10	4.0717e-14

D'après le tableau 2 qui représente les erreurs inverses obtenues avec les différents reordonnancement sur les différentes matrices étudiées, nous pouvons estimer que les

solutions obtenues avec les méthodes de réordonnancement sont assez correctes (erreur très faible considérée comme erreur machine), comparées aux solutions obtenues sans permutation.

### 3.2 Avec une factorisation LU

Nous appliquons la factorisation LU sur les matrices : hydcar20, pde225 5e-1 et piston.

En effet, lorsque nous avons essayé de faire la factorisation de Cholesky pour ces 3 matrices nous nous sommes rendus compte que le paramètre flag de la fonction chol était différent de 0. Cela signifie que les matrices ne remplissent pas les conditions d'utilisation de la factorisation de Cholesky (symétrique définie positive).

Dans cette partie nous avons décidé de ne pas afficher les matrices obtenues après permutation car nous rencontrons des problèmes au niveau de matlab car ces matrices sont assez grandes.

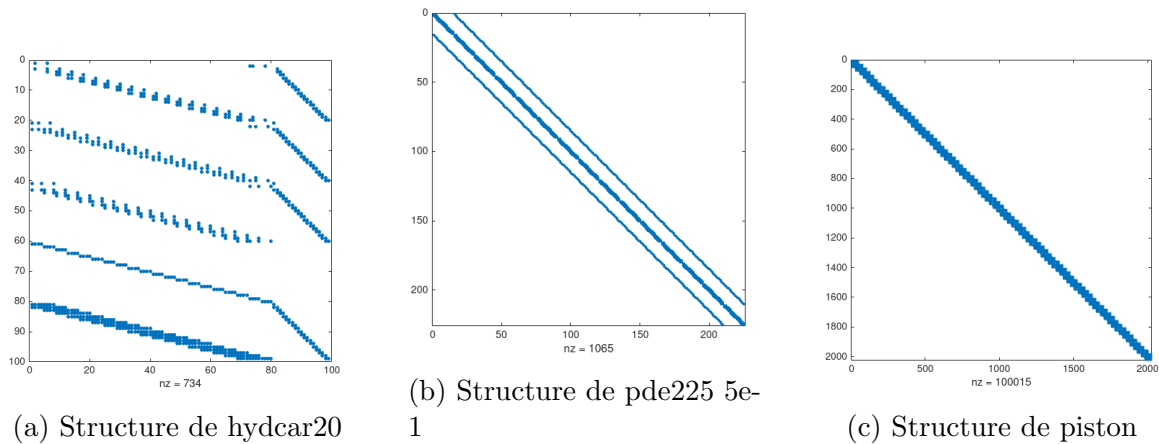


FIGURE 2 – La structure des différentes matrices étudiées avec une factorisation LU

Dans un premier temps nous allons analyser le nombre de non zéros des 3 matrices :

TABLE 3 – Nombre d'opérations pour la résolution avec une factorisation LU avec différents algorithmes de réordonnancement

	hydcar20	pde225 5e-1	piston
Nb de non zéros	734	1065	100015

Nous voyons que la matrice piston est celle qui contient le plus de non zéros suivie de la matrice pde et hydcars.

Ensuite nous allons calculer le nombre d'opérations pour la résolution avec une factorisation LU sur la matrice permutée pour chacune des matrices.

TABLE 4 – Nombre d'opérations pour la résolution avec une factorisation LU avec différents algorithmes de réordonnement

Réordonnement	hydcars20	pde225 5e-1	piston
aucun	6119	12881	345831
symamd	6529	6537	407633
symrcm	9925	9605	329383
amd	6057	6517	342355
colamd	2639	8657	450887
colperm	7359	28229	993661

On remarque que pour la matrice hydcars le meilleur algorithme de réordonnement est colamd. Pour la matrice pde225 c'est l'algorithme amd et enfin pour la matrice piston c'est l'algorithme symrcm.

Pour la matrice piston c'est un réordonnement symétrique qui permet de diminuer au maximum le nombre d'opérations.

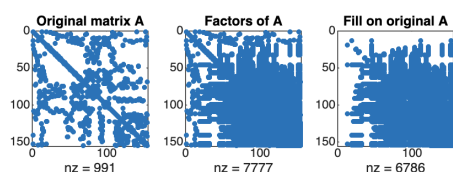
TABLE 5 – Erreurs de la norme inverse sur la solution de la résolution avec une factorisation LU avec différents algorithmes de réordonnement

Réordonnement	hydcars20	pde225 5e-1	piston
aucun	5.3053e-15	4.6159e-16	4.8581e-10
symamd	1.001288e+00	4.671751e+00	1.023499e+00
symrcm	1.004594e+00	5.730556e-01	9.407606e-01
amd	1.002024e+00	4.298578e+00	1.040268e+00
colamd	1.000497e+00	3.816767e+00	1.014061e+00
colperm	1.006331e+00	1.406792e+00	1.006815e+00

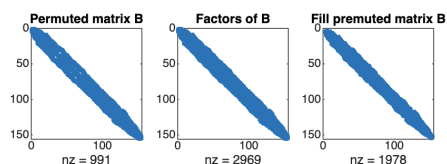
On remarque cette fois que l'erreur est assez élevée pour chacune des permutations. Pour chaque matrice l'erreur la plus faible est obtenue sans aucun réordonnancement.

## 4 Conclusion

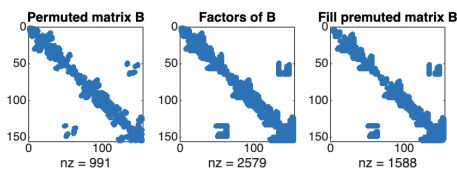
## 5 Annexe



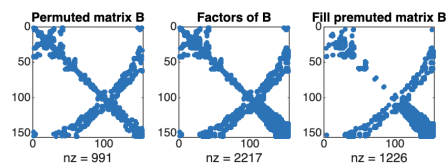
(a) Mat0 sans permutation



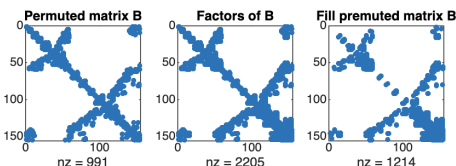
(c) Fillin de mat0 avec permutation symamd



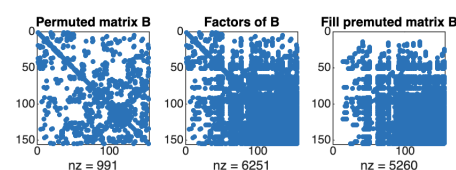
(e) Fillin de mat0 avec permutation colamd



(b) Fillin de mat0 avec permutation symamd



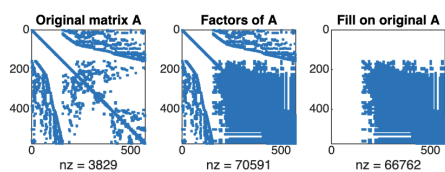
(d) Fillin de mat0 avec permutation amd



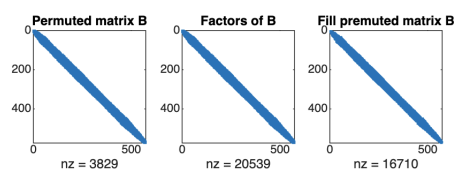
(f) Fillin de mat0 avec permutation colperm

FIGURE 3 – Fillin de mat0 avec différentes méthodes de réordonnancement

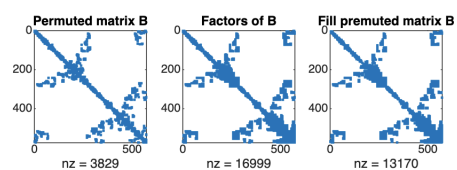




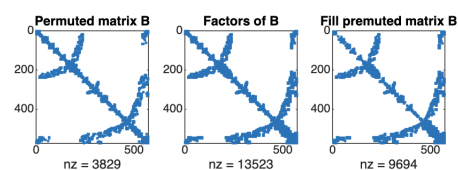
(a) Mat1 sans permutation



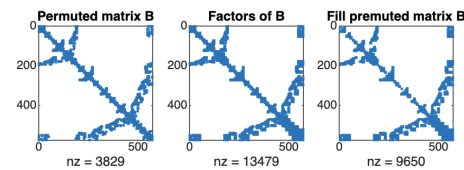
(c) Fillin de mat1 avec permutation symrcm



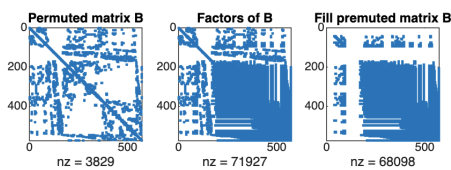
(e) Fillin de mat1 avec permutation colamd



(b) Fillin de mat1 avec permutation symamd

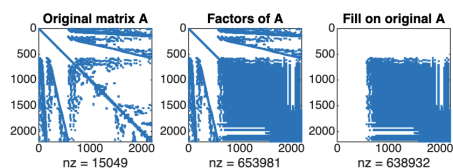


(d) Fillin de mat1 avec permutation amd

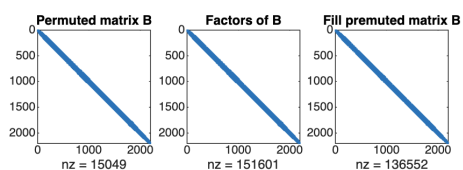


(f) Fillin de mat1 avec permutation colperm

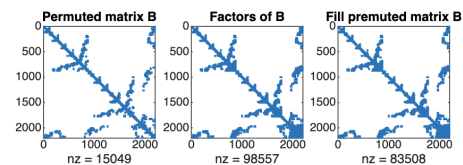
FIGURE 4 – Fillin de mat1 avec différentes méthodes de réordonnancement



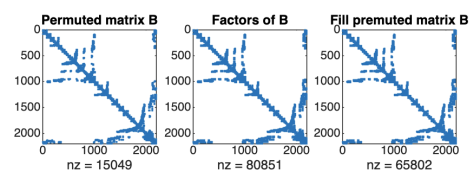
(a) Mat2 sans permutation



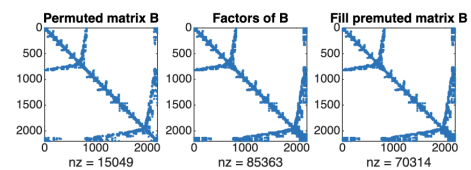
(c) Fillin de mat2 avec permutation symrcm



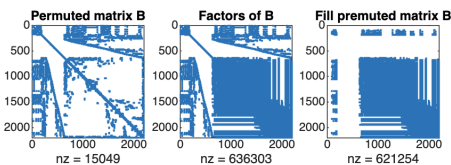
(e) Fillin de mat2 avec permutation colamd



(b) Fillin de mat2 avec permutation symamd

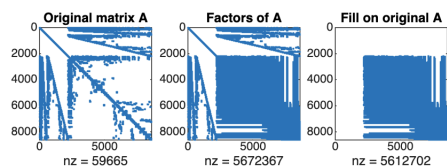


(d) Fillin de mat2 avec permutation amd

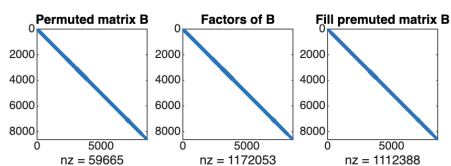


(f) Fillin de mat2 avec permutation colperm

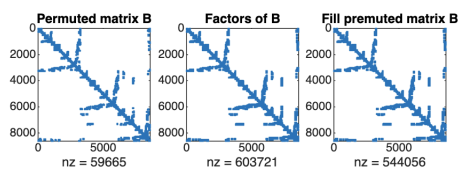
FIGURE 5 – Fillin de mat2 avec différentes méthodes de réordonnancement



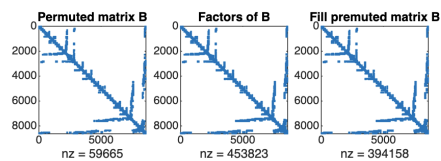
(a) Mat3 sans permutation



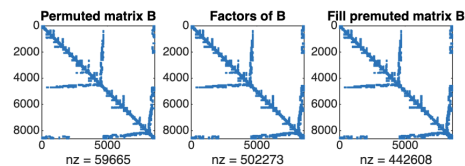
(c) Fillin de mat3 avec permutation symrcm



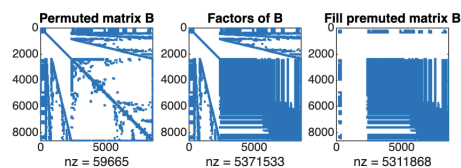
(e) Fillin de mat3 avec permutation colamd



(b) Fillin de mat3 avec permutation symamd

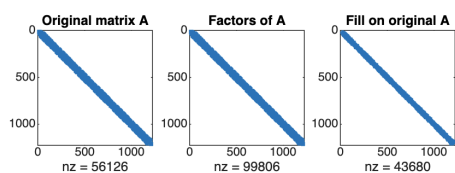


(d) Fillin de mat3 avec permutation amd

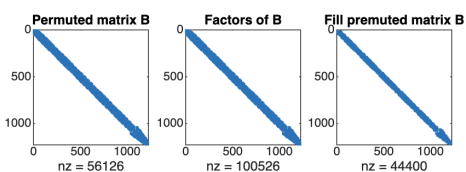


(f) Fillin de mat3 avec permutation colperm

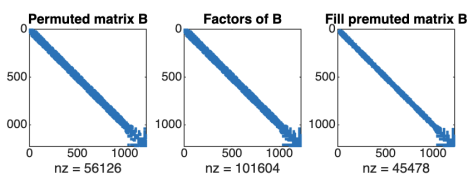
FIGURE 6 – Fillin de mat3 avec différentes méthodes de réordonnancement



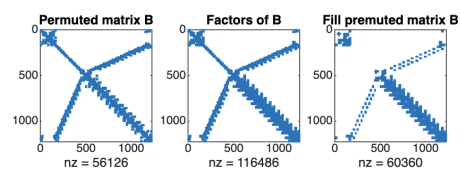
(a) Bcsstk\_25 sans permutation



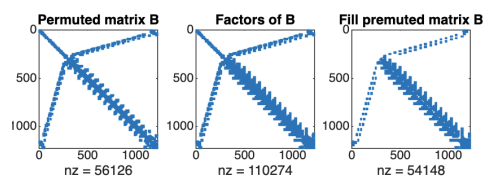
(c) Fillin de Bcsstk\_25 avec permutation symm



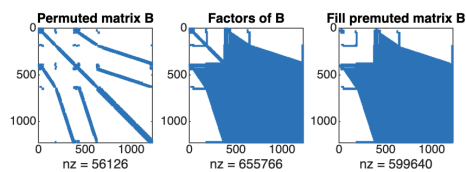
(e) Fillin de Bcsstk\_25 avec permutation colamd



(b) Fillin de Bcsstk\_25 avec permutation symamd



(d) Fillin de Bcsstk\_25 avec permutation amd



(f) Fillin de Bcsstk\_25 avec permutation colperm

FIGURE 7 – Fillin de Bcsstk\_25 avec différentes méthodes de réordonnancement