
Méthodes de Krylov préconditionnées

TP2

Yousra Bouchikhi, Hamza Belkarkor

2^{me} année en Modélisation et Intelligence Artificielle, double diplôme
à l'INSA Toulouse et l'ENSEEIH - Toulouse, France

Mars 2023

Sommaire

1	Introduction	1
2	Sans Préconditionneur	2
3	Préconditionneur diagonal	3
4	Cholesky	4
4.1	Résultats de Cholesky incomplet sans fill-in	4
4.2	Cholesky incomplet avec tolérance	5
5	Conclusion	7

1 Introduction

Dans ce TP, on cherche à résoudre le système linéaire : $Ax = b$ où A est la matrice issue de la discrétisation par éléments finis d'une EDP de type elliptique. Dans ce contexte, les méthodes de Krylov constituent une alternative souvent retenue dans les grands codes de simulation. Ces méthodes (Gradient Conjugué dans le cas symétrique défini positif, GMRES non-symétrique général, ...) convergent d'autant plus vite que la matrice du système à résoudre est "proche" de la matrice identité. Afin de satisfaire à cette contrainte, des techniques de préconditionnement sont utilisées. Celles-ci consistent à résoudre un système linéaire équivalent :

$$M^{-1}Ax = M^{-1}b$$

où la matrice M est appelée préconditionneur.

Les idées qui gouvernent sa construction sont :

1. M doit être facile à construire,
2. M doit être facile à utiliser,
3. M doit être peu coûteuse en mémoire,
4. M doit être la meilleure approximation possible de A .

Parmi les préconditionneurs algébriques classiques, on compte les factorisations incomplètes de type Cholesky (pour les matrices symétriques définies positives) et LU (pour des matrices quelconques) qui permettent d'exprimer M sous la forme $M = M_1 M_2$.

Dans la suite de ce rapport nous allons voir différents types de préconditionneurs pour la résolution d'un système linéaire, ainsi qu'une comparaison entre les différents résultats obtenus avec ceux-ci.

2 Sans Préconditionneur

Toutes les matrices utilisées dans cette étude sont symétriques et définies positives ce qui justifie l'utilisation de l'algorithme du gradient conjugué dans la suite de ce rapport.

La figure ci-dessous montre l'évolution de l'erreur inverse "normwise" au cours des itérations pour 4 niveaux de raffinement.

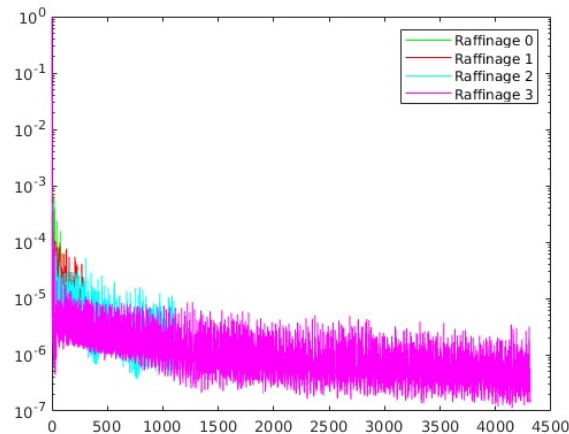


Figure 1: Évolution de l'erreur inverse en fonction des itérations

Nous n'avons pas de convergence pour les 4 niveaux de raffinement. L'erreur inverse est très instable et n'atteint pas la tolérance de 10^{-10} fixée pour ce problème.

Raffinage	Taille	Nb iter	Temps de construction	Temps de résolution	Temps total
0	155	77	4.74e-04 s	2.88e-02 s	2.93e-02 s
1	573	239	9.85e-04 s	1.77e-01 s	1.78e-01 s
2	2201	769	3.410000e-03 s	5.177691e+00 s	5.181101e+00 s
3	8625	4148	1.39e-02 s	2.42e+02 s	2.43e+02 s

Table 1: Résultats de résolution du système linéaire sans préconditionneur

Le temps de résolution du système linéaire augmente d'autant plus que la taille du système est grande, tel que le passage d'un système de taille 2201 à un autre de taille 8625 le temps de résolution a été multiplié par 100. Nous n'avons quand même pas convergence, le programme s'arrête et affiche un diagnostic de $flag = 1$.

3 Préconditionneur diagonal

Nous choisissons d'utiliser pour cette partie un preconditionneur diagonal où $M1 = I$ et $M2 = \text{diag}(A)$ et $M = M1.M2$

La figure ci-dessous montre l'évolution de l'erreur inverse "normwise" au cours des itérations pour 4 niveaux de raffinement.

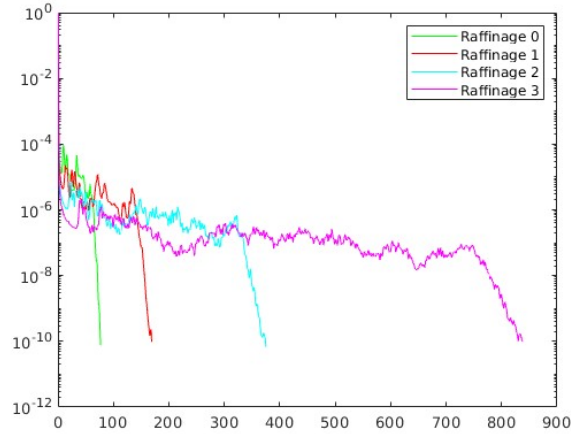


Figure 2: Évolution de l'erreur inverse en fonction des itérations

On remarque la convergence de l'algorithme avec un preconditionnement diagonal pour tous les niveaux de raffinement, cependant la convergence est lente pour le dernier niveau de raffinement de taille 8625 tel que l'erreur inverse atteint une tolérance de 10^{-10} au bout de 837 itérations.

Raffinage	Taille	Nb iter	Temps de construction	Temps de résolution	Temps total
0	155	76	3.39e-03 s	8.46e-03 s	1.18e-02 s
1	573	168	6.59e-04 s	3.49e-02 s	3.55e-02 s
2	2201	374	4.305e-03 s	8.95e-01 s	8.99e-01 s
3	8625	837	7.87e-03 s	2.33e+01 s	2.33e+01 s

Table 2: Résultats de résolution du système linéaire avec preconditionneur diagonal

Nous remarquons que plus la taille du raffinement est grande, plus le temps de construction du preconditionneur ainsi que le temps de résolution est grand. Le temps de construction ce preconditionneur reste quand même négligeable. Le preconditionneur diagonale donne des résultats en termes de précision et de temps de résolution globalement satisfaisants pour des problèmes de petite taille, mais la convergence risque d'être très lente pour des problèmes de tailles plus grandes.

4 Cholesky

4.1 Résultats de Cholesky incomplet sans fill-in

La factorisation de Cholesky incomplète est une méthode permettant de calculer une matrice triangulaire inférieure L qui satisfait $A = LL^T$, où A est une matrice symétrique et définie positive. Elle est appelée "incomplète" car seuls certains des éléments de A sont utilisés pour calculer L . Plus précisément, seuls les éléments qui ne sont pas nuls dans le motif de A sont utilisés.

Ici $M_1 = L$ et $M_2 = L^T$.

Nous pouvons voir ci-dessous la structure de la matrice A de taille 8625 et du facteur de Cholesky sur la même figure.

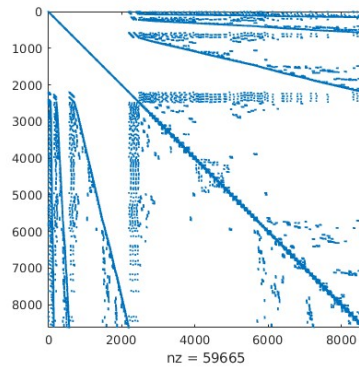


Figure 3: Structure de la matrice A de taille 8625 et du facteur de Cholesky

La figure ci-dessous montre l'évolution de l'erreur inverse "normwise" au cours des itérations pour 4 niveaux de raffinement.

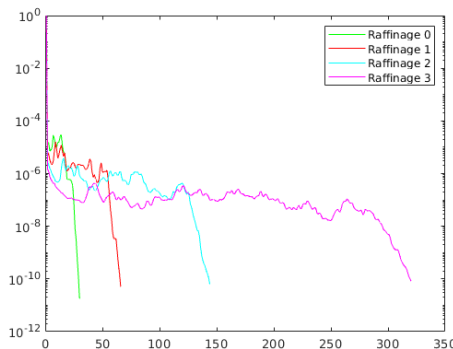


Figure 4: Évolution de l'erreur inverse en fonction des itérations

On remarque une convergence pour les 4 niveaux de raffinages, ce préconditionneur a atteint la précision de 10^{-10} et a convergé en beaucoup moins d'itérations que le préconditionneur diagonal.

La table suivante récapitule les résultats obtenus pour ce préconditionneur pour les 4 niveaux de raffinement.

Raffinage	Taille	Nb iter	Temps de construction	Temps de résolution	Temps total
0	155	29	6.99e-04 s	3.18e-02 s	3.25e-02 s
1	573	65	2.57e-04 s	9.02e-03 s	9.27e-03 s
2	2201	143	5.80e-04 s	1.05e-02 s	1.11e-02 s
3	8625	319	1.63e-03 s	1.90e-01 s	1.92e-01 s

Table 3: Résultats de Cholesky incomplet sans fill-in

4.2 Cholesky incomplet avec tolérance

La fonction *cholinc_n7* en MATLAB calcule une factorisation de Cholesky incomplète approximative pour une matrice symétrique définie positive A avec un seuil de tolérance *droptol*. Elle utilise la méthode de remplissage limité *ilutp* pour calculer une factorisation LU incomplète, puis elle calcule une matrice triangulaire inférieure L à partir de la factorisation LU incomplète. La matrice triangulaire supérieure correspondante peut être calculée en prenant la transposée de L .

La figure ci-dessous montre l'évolution de l'erreur inverse "normwise" au cours des itérations pour 4 niveaux de raffinement et avec une tolérance de 0.1.

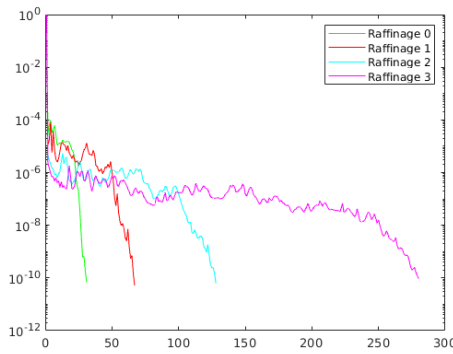


Figure 5: Évolution de l'erreur inverse en fonction des itérations - Tolérance = 0.1

On remarque une convergence pour les 4 niveaux de raffinages, ce préconditionneur a atteint la précision de 10^{-10} et a convergé en beaucoup moins d'itérations que les précédents. La table suivante récapitule les résultats obtenus pour ce préconditionneur pour les 4 niveaux de raffinement.

Raffinage	Taille	Nb iter	Temps de construction	Temps de résolution	Temps total
0	155	30	2.371e-03 s	1.536e-02 s	1.773e-02 s
1	573	66	1.583e-03 s	8.844e-03 s	1.043e-02 s
2	2201	127	3.179e-03 s	9.599e-03 s	1.278e-02 s
3	8625	279	6.100e-03 s	1.118e-01 s	1.172e-01 s

Table 4: Résultats de Cholesky incomplet avec une tolérance de 0.1

NB: Les deux méthodes convergent pour tous les niveaux de raffinement.

Les figures ci-dessous montrent l'influence du threshold sur la vitesse de convergence.

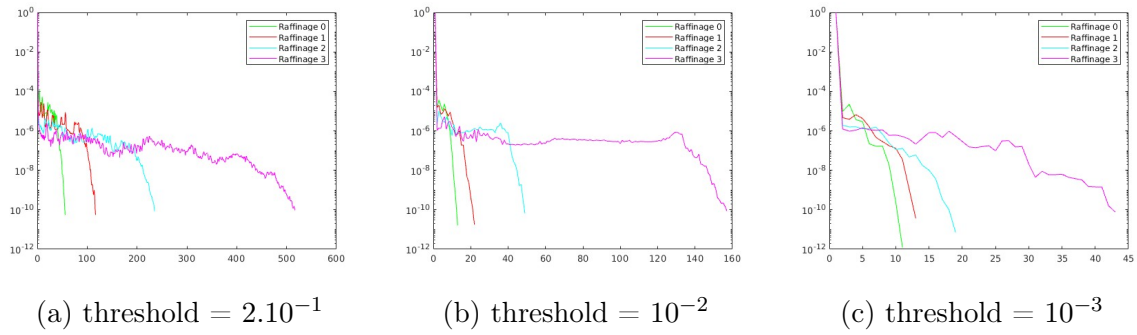


Figure 6: Évolution de l'erreur inverse pour 4 niveaux de raffinement et pour différentes valeurs de threshold

Nous remarquons que plus petit est le seuil choisi plus rapide est la convergence de l'algorithme avec préconditionneur de Cholesky incomplet avec threshold, cependant d'après les résultats obtenus (voir code matlab) nous remarquons que plus le threshold est petit plus on s'éloigne des facteurs complets de A car dans ce cas $LL^T A$ s'éloigne de I (matrice identité).

5 Conclusion

Dans ce rapport, nous avons testé trois préconditionneurs différents pour l'algorithme du gradient conjugué, à savoir la matrice diagonale, la factorisation de Cholesky sans remplissage (ou "no-fill" en anglais) et la factorisation de Cholesky avec seuil de remplissage. Nous avons testé ces préconditionneurs sur des systèmes linéaires de tailles croissantes et avons mesuré les temps de calcul nécessaires pour obtenir une solution précise.

Nos résultats montrent que la factorisation de Cholesky avec seuil de remplissage est le préconditionneur le plus efficace pour les systèmes linéaires de grande taille. En effet, ce préconditionneur a permis d'obtenir des temps de calcul significativement plus courts que les autres préconditionneurs testés, tout en garantissant une précision suffisante pour les applications considérées.

En revanche, pour les systèmes linéaires de petite taille, les préconditionneurs diagonal et Cholesky sans remplissage peuvent offrir des temps de calcul plus courts que la factorisation de Cholesky avec seuil de remplissage. Cela s'explique par le fait que la factorisation de Cholesky avec seuil de remplissage implique un coût de calcul supplémentaire lié au choix du seuil de remplissage optimal.

En conclusion, le choix du préconditionneur dépend de la taille du système linéaire considéré. Pour les systèmes de grande taille, la factorisation de Cholesky avec seuil de remplissage est le préconditionneur le plus efficace, tandis que pour les systèmes de petite taille, le préconditionneur diagonal ou la factorisation de Cholesky sans remplissage peuvent être préférables.