

Gradients conjugués

Minh-Phuong Tran - 51091600 - Décembre 2018

1 Introduction

La méthode des gradients conjugués est une méthode itérative résolvant les systèmes symétriques et définies positives. En arithmétique exacte, cette méthode converge en au plus m , la taille du système ce qui est une propriété intéressante. Néanmoins, comme il ne sera pas possible de travailler sans erreurs d'arrondi, cette propriété ne pourra pas être utilisée. Il faudra alors trouver une autre borne pour la vitesse de convergence.

Deuxièmement, l'utilisation d'un préconditionneur ILU0 peut être associée au gradients conjugués et son effet sera étudié dans les sections suivantes.

NB : Toutes les analyses se font par rapport aux systèmes fournis par le fichier `ccore.py` sauf indication contraire.

2 Conditionnement et spectre

Tout d'abord, dans le cas du maillage du `ccore.py` rappelons les paramètres ayant un impact sur le conditionnement.

- La taille de l'entrefier, avec lequel le conditionnement évolue en $\frac{1}{x}$
- Le raffinement du maillage
- La perméabilité du noyau (proche de 0 : en $\frac{1}{x}$, loin de 0 : linéairement)

Dans le reste de cet article, le conditionnement sera modifié avec la taille de l'entrefier et les analyses se feront en fonction de ce paramètre.

Le préconditionnement est un outil intéressant pour les systèmes mal conditionnés. En effet, lorsqu'appliqué à un système, le nouveau système obtenu est beaucoup mieux conditionné. De sorte que l'on préfère résoudre $M^{-1}Ax = M^{-1}b$ que le système initial $Ax = b$ avec M le préconditionneur. Pour cela, il faut que $\kappa(M^{-1}A) \ll \kappa(A)$. Le préconditionneur qui sera utilisé dans le reste de l'article est le préconditionneur ILU0 (appliquant une décomposition LU incomplète au système).

Ainsi, avec la Figure 1, le conditionnement du système avec préconditionneur ILU0 est nettement meilleure que sans et la différence est d'autant plus marquante que le système de base est mal conditionné. Le spectre des valeurs propres du système avec préconditionneur est également beaucoup moins large que le système de base. L'utilisation d'un préconditionneur ILU0 est donc justifiée, d'autant plus que la méthode des gradients conjugués converge plus rapidement lorsque le spectre des valeurs propres est "regroupé" (*clustered*) (sauf autour de 0)[1] ce qui est bien donné par ILU0 [2].

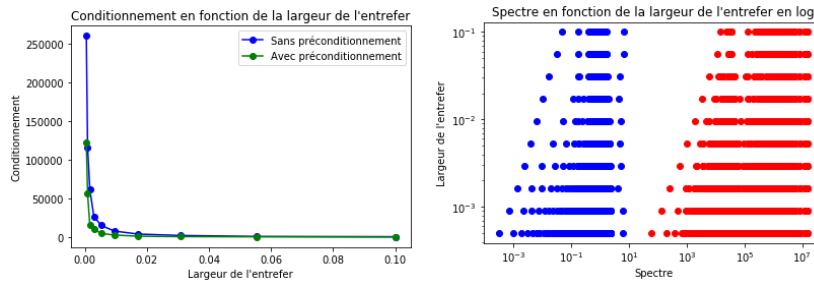


FIGURE 1: Spectre et conditionnement avec et sans préconditionnement en fonction de la largeur de l'entrefier

3 Convergence

La convergence de la méthode des gradients conjugués peut être mesurée par son nombre d'itérations comme montré sur la Figure 2. Sur le premier graphique, la convergence évolue en $\frac{1}{x}$ avec la largeur de l'entrefier. On observe grâce à ce graphique que lorsque la largeur de l'entrefier devient de plus en plus petite, c'est-à-dire que le problème est de moins en moins bien conditionné, il devient beaucoup plus intéressant de travailler avec un préconditionneur qui permet de réduire drastiquement le nombre d'itérations, que sans. La relation directe entre la convergence et le conditionnement est montrée sur le 2e graphique. En effet, le logarithme de la convergence évolue linéairement avec le logarithme du conditionnement. Ce qui veut dire que leur relation est de la sorte : $\text{Nb d'itérations} = C * (\text{conditionnement})^a$. Ici l'exposant a peut être obtenu comme étant la pente du 2e graphique et C comme étant la pente du 3e graphique. Il est

intéressant de voir que l'exposant α est fort semblable (voir le même) pour les 2 cas de figure (avec et sans préconditionnement) mais que C est bien inférieur dans le cas avec préconditionnement.

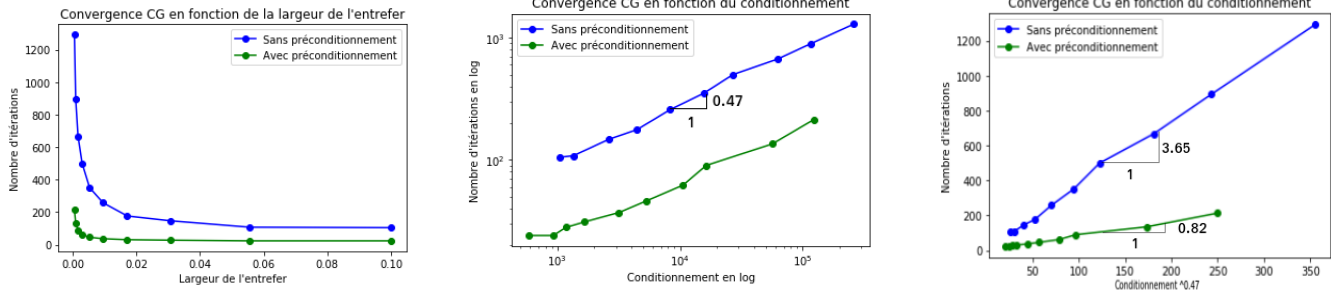


FIGURE 2: Convergence en fonction du conditionnement

Enfin, une borne sur les erreurs à chaque itération et sur la vitesse de convergence peut être trouvée grâce au théorème 3.5 du livre Numerical Linear Algebra [3] (pour des κ larges et un A symétrique, défini positif avec κ défini avec la 2-norm) :

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq 2 * \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^n \sim 1 - \frac{2}{\sqrt{\kappa}}$$

Avec $\|v\|_A = (v^T A v)^{1/2}$. Cela signifie que *pour des k larges mais pas trop larges, la convergence vers une certaine tolérance peut être obtenue en au plus $O(\sqrt{\kappa}) = O(\kappa^{0.5})$ itérations*[3].

Or dans notre cas, la relation entre le nombre d'itérations et le conditionnement a été trouvé précédemment (pour le cas sans préconditionnement) :

$$\text{Nb d'itérations} = 3.65 * (\text{conditionnement})^{0.47}$$

L'exposant 0.47 est bien inférieur à 0.5 (pareillement dans le cas avec préconditionnement). La borne sur la vitesse de convergence est vérifiée pour les maillages générées par le ccore.py.

Ensuite en calculant les erreurs à chaque itération et pour chaque système avec κ , la borne des erreurs est également vérifiée sur la Figure 3, avec en bleu $2 * \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^n$, et en rouge $\frac{\|e_n\|_A}{\|e_0\|_A}$:

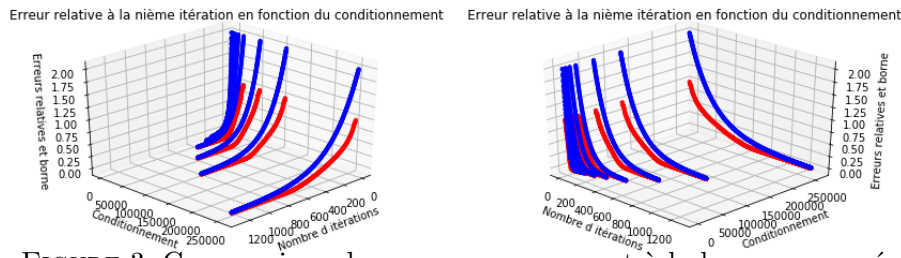


FIGURE 3: Comparaison des erreurs par rapport à la borne annoncée

4 Conclusion

Au terme de cette analyse, l'utilisation du préconditionneur ILU0 sur les systèmes mal conditionnés est justifiée puisqu'elle permet une convergence plus rapide. La borne théorique sur les erreurs et la vitesse de convergence du théorème 3.5 est également vérifiée pour les maillages du ccore.py. Cela ne signifie pas pour autant qu'elle est vérifiée sur n'importe quel système en effet sur les systèmes générés par la fonction creatDF, ce n'est pas le cas où $\text{Nb d'itérations} = C * (\text{conditionnement})^{0.54}$ et où l'utilisation d'un préconditionneur avec des matrices de tailles faibles (et de conditionnement κ faible) donne une convergence plus lente.

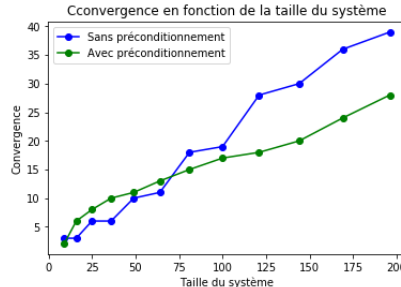


FIGURE 4: Convergence CG en fonction de la taille du système

Références

- [1] Emory College of Arts and Science, Department of Mathematics and Computer Science, Preconditioning Techniques for Large Linear Systems : A Survey, [http ://www.mathcs.emory.edu/ benzi/Web_papers/survey.pdf](http://www.mathcs.emory.edu/benzi/Web_papers/survey.pdf)
- [2] Technical University of Munich, Parallel Numerics, [https ://www5.in.tum.de/lehre/vorlesungen/parnum/WS11/PARNUM_8.pdf](https://www5.in.tum.de/lehre/vorlesungen/parnum/WS11/PARNUM_8.pdf)
- [3] Trefethen, Bau, Numerical Linear Algebra, Siam, 1997.