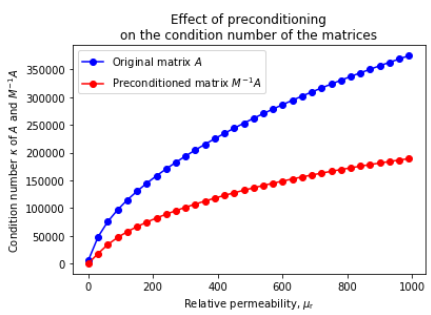
Introduction

Pour ce devoir, il était demandé d'écrire des fonctions en langage Python permettant de résoudre efficacement un système d'équations linéaires issu d'un modèle d'éléments finis appelé `ccore` en utilisant un algorithme de gradients conjugués. La résolution est donc faite de façon itérative, et l'influence du préconditionneur `ILU0` est également montrée. Une attention particulière est apportée à l'étude de l'évolution du conditionnement de la matrice, facteur très important pour la vitesse de convergence du solveur.

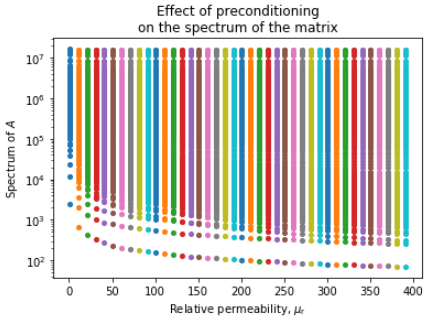
1 Effet du préconditionneur

Pour la première question, il était demandé d'observer l'effet du préconditionnement sur le conditionnement de la matrice, ainsi que sur le spectre des valeurs propres de celle-ci. On appelle $A \in \mathbb{C}^{n\times n}$ la matrice initiale, et $M \in \mathbb{C}^{n\times n}$ le préconditionneur `ILU0` utilisé. Le système que l'on résout devient alors $M^{-1}Ax = M^{-1}b$ au lieu de $Ax = b$. Le nombre d'itérations nécessaires (pour une tolérance donnée) pour la méthode des gradients conjugués évolue en $\mathcal{O}\left(\sqrt{\kappa}\right)$, où κ dénote le conditionnement de la matrice utilisée (A ou $M^{-1}A$). L'algorithme `ILU0` va calculer la matrice M telle que $\kappa(M^{-1}A) \ll \kappa(A)$ et $\|M^{-1}A - I\| \approx 0$. On peut voir dans le livre de référence [1] pp. 298–300 que les Théorèmes 38.3–5 affirment qu'afin de diminuer le nombre d'itérations nécessaires pour converger, nous pouvons jouer sur le spectre des valeurs propres de la matrice, Λ. En effet, un « bon » spectre possède au moins l'une des deux qualités suivantes : ses valeurs propres sont fortement groupées et/ou son conditionnement κ (le rapport $|\lambda_{\max}| \,/\, |\lambda_{\min}|$ pour une matrice hermitienne) est faible. En affichant l'évolution du conditionnement de la matrice en fonction de l'évolution de l'un des trois paramètres (largeur de l'entrefer, raffinement du maillage et perméabilité magnétique du noyau, chaque fois en fixant les deux autres), on observe que le conditionnement de la matrice préconditionnée est inférieur à celui de la matrice initiale pour les plages de valeurs choisies et les matrices issues du modèle `ccore`. Par souci de place, seul le graphe en fonction de la perméabilité relative est montré à la Figure 1a, bien que le résultat soit similaire lorsque l'on joue sur les autres paramètres.

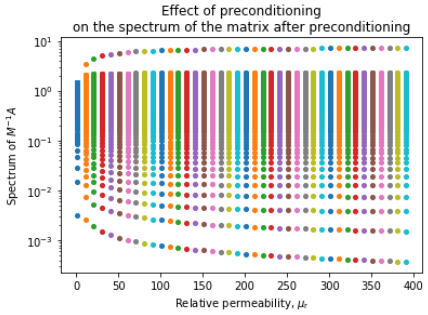
Regardons ensuite l'effet du préconditionnement sur le spectre des valeurs propres de la matrice. Comme dit ci-dessus, un « bon » spectre est un spectre groupé autour de l'unité. Afin d'observer ceci, il est intéressant de représenter le support du spectre, comme sur les graphes des Figures 1b et 1c. Comme les graphes sont semi-logarithmiques, il suffit de comparer les échelles des graphes pour voir que sur la Figure 1c les valeurs sont beaucoup mieux groupées et se trouvent beaucoup plus près de 1, ce qui confirme que le conditionnement est bien plus faible pour la matrice préconditionnée dans le cas d'un changement de perméabilité relative.



(a) Évolution de κ en fonction de la perméabilité relative pour une matrice avec et sans préconditionnement.



(b) Évolution de Λ en fonction de la perméabilité relative pour une matrice non préconditionnée.



(c) Évolution de Λ en fonction de la perméabilité relative pour une matrice préconditionnée.

FIGURE 1 – Différents graphes pertinents pour l'analyse de conditionnement en section 1.

2 Convergence de la méthode itérative

Pour la seconde partie du devoir, il était demandé d'analyser en détail la convergence de la méthode itérative. Afin de faire cela, commençons par un développement théorique.

2.1 Borne théorique en arithmétique exacte

Une première borne qui semble intéressante est celle donnée par le Théorème 38.2 dans le livre de référence : « *pour une matrice symétrique définie positive, la méthode des gradients conjugués converge en un nombre d'itérations inférieur à la taille du système* ». Dans notre cas, cela revient à dire que l'algorithme

1. Trefethen, Lloyd N. & Bau, David III. (1997). *Numerical Linear Algebra*. Philadelphia : PA, SIAM.

devra faire au plus n itérations. Cependant, ce n'est pas ce qu'on observe sur le graphe de la Figure 2a, qui donne le nombre d'itérations nécessaires afin d'obtenir une erreur en-dessous de la tolérance en fonction de n. Ceci s'explique par le fait que la borne théorique n'est valable qu'en arithmétique exacte, or le calcul sur ordinateur se fait en virgule flottante. Cette borne a donc peu d'intérêt en analyse numérique pratique. On note en passant que la relation entre n et le nombre d'itérations paraît sublinéaire sur le graphe.

2.2 Borne théorique en fonction du conditionnement

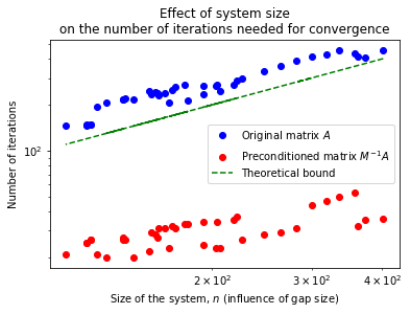
Une autre borne donnée dans le livre est celle du Théorème 38.5, qui dit que « *pour une tolérance donnée, la convergence vers une solution acceptable se fera en* $\mathcal{O}(\sqrt{\kappa}) = \mathcal{O}(\kappa^{1/2})$ *itérations pour* κ *suffisamment grand (mais pas trop grand)* ». Remarquons qu'il s'agit d'une borne théorique supérieure, et qu'en pratique il est possible d'observer un exposant inférieur pour κ. On s'attend donc à ce que le graphe du nombre d'itérations ξ en fonction du conditionnement de la matrice (avec les deux axes en logarithmique) présente une pente $c \leq 1/2$ (car $\xi \leq \alpha\kappa^c \iff \log\xi \leq \log\alpha + c\log\kappa$). C'est bien ce qu'on observe sur le graphe de la Figure 2b. Il est intéressant de remarquer que pour les matrices préconditionnées, comme le spectre est beaucoup mieux groupé (Figures 1b et 1c), la pente est visiblement plus faible que pour les matrices directement issues de `ccore`. Cette expérience indique que si on veut résoudre le système linéaire rapidement, il faut préconditionner la matrice A de sorte à obtenir un κ plus faible et un spectre plus groupé.

2.3 Norme des résidus

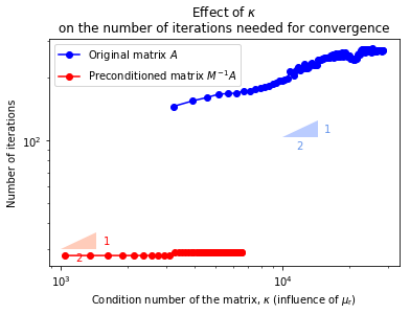
Étudions ensuite l'évolution des résidus au fur et à mesure des itérations du solveur. Par le Théorème 38.5, la convergence attendue est linéaire, car

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n, \quad \text{où} \quad \|x\|_A \triangleq \sqrt{x^T A x}\,.$$

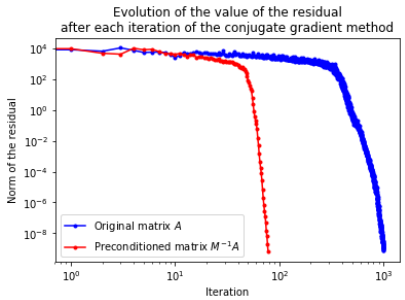
En pratique, on observe dans le cas des matrices `ccore` le résultat du graphe 2c, qui donne l'évolution de la norme du résidu à l'itération courante. On remarque que la résolution se fait en deux « étapes » : les quelques premières itérations n'améliorent que peu la solution (mais de façon linéaire), jusqu'à atteindre un seuil qui dépend des paramètres. Une fois ce seuil atteint, les itérations améliorent toujours de façon linéaire la norme des résidus, mais la pente devient plus négative. Cet effet n'est probablement pas très important, car il semble disparaître pour des matrices plus grandes.



(a) Évolution du nombre d'itérations en fonction de n, en jouant sur la largeur de l'entrefer. La borne est celle de la section 2.1.



(b) Évolution du nombre d'itérations en fonction de κ. Les triangles servent à visualiser la borne supérieure sur la pente ($c = 1/2$).



(c) Évolution de la norme du résidu à l'itération courante. La convergence devient plus rapide à partir du seuil, mais cet effet n'est pas réellement important.

FIGURE 2 – Différents graphes pertinents pour l'analyse de convergence en section 2.

Conclusion

Les systèmes (symétriques) définis positifs tels que ceux issus du modèle `ccore` se rencontrent très fréquemment dans le monde de la physique et de l'ingénierie. Grâce aux méthodes itératives comme celle des gradients conjugués (basée sur l'itération d'Arnoldi), il est possible de profiter des propriétés intéressantes de ces matrices afin de diminuer fortement le temps nécessaire pour résoudre un système linéaire les contenant.

En faisant ce devoir, il est devenu apparent que trouver un bon préconditionneur qui ne soit pas trop difficile à calculer est primordial afin de pouvoir espérer une convergence en peu d'itérations, mais que choisir celui-ci n'est pas aussi simple que cela n'en a l'air. Le préconditionneur `ILU0` semble donner un bon compromis entre facilité de calcul et diminution du nombre d'itérations.