Méthodes de Krylov préconditionnées

Dans tous les exercices, on s'intéressera au coût mémoire, à la vitesse de convergence (nombre d'itérations) et au temps de calcul associé aux choix algorithmiques (choix du préconditionneur) retenus pour la résolution d'un système linéaire

$$Ax^* = b$$

1 Préambule

Pour la résolution de systèmes linéaires creux de très grande taille, les méthodes itératives constituent une alternative aux méthodes directes (factorisation) dans le cas où la taille des facteurs devient trop importante (au regard de la mémoire disponible sur l'ordinateur cible), ou lorsque les temps de factorisation deviennent trop élevés ou encore lorsque la contrainte de précision numérique peut être relâchée (schéma non-linéaire, imprécision sur les données).

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.

Ceci est illustré en particulier par la borne suivante qui peut être établie pour la vitesse de convergence du gradient conjugué (voir cours) :

$$\|x_k - x^*\|_A \leq 2 \cdot \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|x_0 - x^*\|_A.$$

où $\kappa(A) = \dfrac{\lambda_{\max}}{\lambda_{\min}}$ et $\|x\|_A^2 = x^T A x$.

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 (matrice symétrique définie positive) et LU (matrice quelconque) qui permettent d'exprimer M sous la forme $M = M_1.M_2$.

2 Résolution d'un système linéaire associé à une matrice issue de la discrétisation d'une EDP

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 (voir fichier pdf explicatif).

Vous commencerez par vérifier que les matrices `mat0`, `mat1`, `mat2` et `mat3` sont symétriques définies positives ce qui justifie que l'algorithme de résolution de système linéaire que nous allons utiliser est le Gradient Conjugué (Préconditionné).

En utilisant la méthode du gradient conjugué (`pcg`), vous étudierez la vitesse de convergence pour différentes tailles du système linéaire (lorsque le maillage est raffiné).

Vous tracerez en particulier, l'historique de convergence de l'erreur inverse *normwise*

$$\eta_b^N(x_k) = \frac{\|r_k\|}{\|b\|} = \frac{\|b - Ax_k\|}{\|b\|}$$

où r_k est le résidu associé à l'itéré x_k de la k^{ieme} itération.

Vous utiliserez différents préconditionneurs :

1. aucun (préconditionneur identité),
2. préconditionneur de Jacobi (préconditionneur diagonal),
3. préconditionneur issu de la factorisation incomplète de Cholesky sans remplissage ($IC('0')$).
4. préconditionneur issu de la factorisation incomplète de Cholesky (IC) avec «threshold» (threshold = $1.e^{-3}$, $5e^{-4}$ et $1e^{-4}$ par exemple).

Pour les préconditionneurs issus de la factorisation incomplète, vous afficherez la structure de A et celle du facteur de Cholesky sur la même figure.

3 Fournitures

1. les fichiers `tubeG.m, tubeB.m, tubeF.m` et `setupC.m` de définition de l'EDP ainsi qu'un fichier pdf explicatif (utilisation de la PDE Toolbox de Matlab)
2. **le fichier `cholinc_n7.m` pour la factorisation incomplète de Cholesky avec «threshold».**
3. le fichier `tp.m` canevas à compléter.

Déposez sous Moodle le fichier tp.m complété ainsi qu'un petit rapport reprenant les différents résultats pour différentes tailles de problèmes :

1. historiques de convergence en fonction des préconditionneur, du threshold,
2. nombres d'itérations, temps (calcul du préconditionneur, résolution) (**tableaux ?**)

Vous pourrez aussi étudier le compromis threshold / remplissage des facteurs du préconditionneur / temps (construction, résolution) pour les préconditionneurs à base de factorisation de Cholesky avec threshold.

Ce rendu est à déposer sous Moodle pour le vendredi 22 Mars.

4 Fonctions Matlab qui peuvent être utiles

— **semilogy**
— **spy**
— **eye, diag, triu**
— **pcg**
— **ichol**, pour la factorisation incomplète de Cholesky sans remplissage
— **cholinc_n7** pour la factorisation incomplète de Cholesky avec threshold