

GMRES préconditionné

Implémentez une fonction PYTHON3 `csrGMRES(sA,iA,jA,b,rtol,prec)` qui résout le système linéaire $Ax = b$ avec l'algorithme itératif de GMRES et qui retourne le tuple (x,res) où x est la solution du système.

Les `numpy.array` `sA`, `iA`, `jA` servent au stockage de la matrice du système en format CSR, `res` est un `np.array` contenant les normes des résidus obtenus à chaque itération, `rtol` est la tolérance relative prescrite sur la norme de ce résidu (critère de convergence), et `prec` est un paramètre booléen qui indique si l'algorithme préconditionné est utilisé ou non. Lorsque le paramètre `prec = True`, votre fonction utilisera l'algorithme GMRES préconditionné par la factorisation incomplète ILU(0) de la matrice A .

Vous vous baserez sur l'algorithme GMRES vu au cours et sur le livre de référence. Pour le préconditionnement, vous vous baserez sur l'algorithme donné dans le fichier `notes_precon.pdf` disponible sur Moodle.

Le seul produit matrice×vecteur de l'algorithme est

$$v = M^{-1}Aq_n$$

au début de l'itération d'Arnoldi. Pour être efficace, il faut faire le produit $z = Aq_n$ en format CSR, et ensuite résoudre le système $Mv = z$ en utilisant la décomposition ILU(0) de M et en effectuant une forward- puis une back-substitution (Trefethen, p152). On trouve facilement sur internet comment effectuer efficacement un produit matrice×vecteur en format CSR.

Pour la matrice rectangulaire \tilde{H} et la résolution du système aux moindres carrés, vous pouvez travailler en format plein et utiliser la décomposition QR de numpy. Le plus simple est de définir une matrice \tilde{H} de taille 200×201 (par exemple) et d'étudier les 200 premières itérations de l'algorithme.

Les itérations GMRES sont de plus en plus lentes car on résout par les moindres carrés un système rectangulaire de plus en plus grand. Il est habituel, à un moment donné (après 200 itérations par exemple) de prendre comme x_0 la solution à laquelle on est arrivé et de redémarrer l'algorithme du début. C'est là le rôle du vecteur u_0 dans `notes_precon.pdf`. Cette procédure s'appelle restart. On ne demande pas de l'implémenter. Vous pouvez donc considérer que $u_0 = 0$.

Analyse

- Ecrivez soigneusement l'algorithme de GMRES préconditionné en utilisant les notations du livre de référence, et mettez-le en annexe à votre rapport. Précisez en dessous de l'algorithme les dimensions des différents vecteurs et matrices utilisés à l'itération n . Écrivez l'algorithme

en version symbolique, comme il est fait dans les encadrés "Algorithm x.y" du livre de référence.

- Résolvez les systèmes linéaires générés par le modèle `ndt.py` au moyen de votre fonction `csrGMRES`.
- Analysez la convergence dans les 4 régimes qui ont été définis et pour deux maillages différents : un petit (quelques centaines d'inconnues) et un plus grand (quelques milliers d'inconnues).
- Évaluez quantitativement l'effet du préconditionnement sur le taux de convergence.

Consignes

Ce devoir est absolument **individuel**.

Un rapport de trois pages (un recto et un verso, et une page supplémentaire pour l'algorithme demandé). contenant texte et figures est à remettre pour le vendredi 20 décembre 2019. Le `.pdf` de ce rapport, ainsi que le fichier `.tex` (figures) et les fonctions PYTHON demandées dans ce devoir sont à soumettre sur Moodle. Le tout est compressé dans un dossier unique `.zip` (~~`.rar`~~). Il n'y a pas de remise en format papier pour ce dernier devoir. Nous vous rappelons que vos implémentations et votre rapport sont soumis à un logiciel anti-plagiat.

Le rapport ne doit pas contenir de page de garde, seulement une entête reprenant au moins le nom de l'auteur. La langue de rédaction est le français. Le rapport doit être réalisé avec L^AT_EX, avec la `documentclass article [11pt]` en `pagestyle plain`. Veillez à fournir des implémentations lisibles, dûment commentées, avec des noms de variables explicites.

Les implémentations sont à réaliser en PYTHON 3. Les librairies admises sont :

- NUMPY
- MATPLOTLIB
- OS
- TIME

Toute autre librairie (p.ex. SCIPY) n'est pas acceptée.

Bon travail !