



Projet d'Introduction aux Méthodes de Krylov

Variantes de la méthode GMRES à espace mémoire contraint

1 Spécification du projet

Le but de ce projet est d'implanter en Matlab les variantes redémarrée (i.e. GMRES(m)) et à "fenêtre glissante" (i.e. DQGMRES(m)) de la méthode GMRES préconditionnée. Pour ces deux algorithmes le critère d'arrêt sera basé sur une erreur inverse; on cherchera autant que possible à ne pas recalculer explicitement le résidu associé à chaque itéré. La signature de chacune de ces fonctions Matlab sera la suivante :

```
[x, flag, relres, iter, resvec] = variantGMRES(A, b, m, tol, max_it, M1, M2, x0)
```

où `variantGMRES` est soit `restartedGMRES`, soit `dqGMRES`. Les paramètres de sortie sont :

- **x**, la solution calculée du système $Ax = b$,
- **flag**, statut de sortie de la routine; ce paramètre sera égal à zéro en cas de convergence et à 1 en cas de non convergence au bout de `max_it` itérations.
- **relres**, la norme relative du résidu à la convergence ($\equiv \eta_b^N(x_{iter})$),
- **iter**, le nombre d'itérations,
- **resvec**, le vecteur des normes des résidus de chaque itération.

Les paramètres d'entrée sont :

- **A**, la matrice du système que l'on cherche à résoudre,
- **b**, le second membre de ce système,
- **m**, le paramètre de restart (pour `restarted_gmres`) ou la taille de la fenêtre d'orthogonalisation (pour `dqgmres`),
- **tol**, valeur seuil de l'erreur inverse à utiliser pour détecter la convergence,
- **max_it**, le nombre d'itérations maximum,
- **M1**, **M2**, préconditionneur implicite en deux facteurs, à utiliser pour accélérer la convergence (signification identique à l'interface Matlab de `gmres`, $M1 \times M2$ proche de A),
- **x0**, la vecteur initial.

2 Documentation

Les variantes à espace mémoire contraint de GMRES sont décrites dans le livre de référence de Y. Saad dont la page web est donnée sous Moodle. Dans la seconde édition de *Iterative methods for sparse linear systems*, les deux méthodes sont abordées à la page 199 du *pdf*.

3 Organisation du développement et de la validation

En TP, vous avez développé et validé le code d'un solveur qui permet d'utiliser la méthode FOM ou la méthode GMRES.

En partant de ce code, nous vous proposons d'avancer par étape :

1. codez et validez le solveur `gmres` préconditionné à gauche (algorithme donné sous moodle). Son interface sera identique à celle des variantes, même si le paramètre m est inutile. Vous pourrez valider ce solveur en le comparant avec le `gmres` de Matlab : les résultats devraient être les mêmes (superposition des normes du résidus sur la même figure),
2. codez et validez la variante `restarted_gmres`,
3. codez et validez la variante `dqgmres` en ne mettant en place que la partie orthogonalisation incomplète et en gardant la factorisation QR Matlab et le calcul de $x_m = x_0 + V_m y_m$,
4. codez et validez la version finale de `dqgmres` en utilisant les rotations de Givens pour résoudre le problème aux moindres carrés et en calculant x_m par la récurrence courte.

Pour chacune de ces étapes, gardez la version du code du solveur ainsi que celle des tests de validation.

4 Évaluation, travail à rendre

Pour ce projet, vous rendrez 2 parties :

1. **Logiciel** : codes Matlab pour chacune des étapes de développement (solveur + validation) ; on tiendra compte de la qualité du code (commentaires, indentation, ...),
2. **Rapport** : document décrivant, les méthodes (principes, conséquences sur les calculs/algo-rithmes), les réalisations et les choix d'implantations logicielles, les tests élémentaires ainsi qu'une petite étude comparative entre ces deux approches. En première page de ce rapport au format **pdf** figurera votre nom.

La date de remise du projet est fixée au **mardi 22 mai 2018, 12H**. Cela se fera par un dépôt sous Moodle.

Annexe

L'ensemble suivant des matrices de test est fourni pour mener l'étude proposée (auxquelles vous pouvez rajouter les matrices du problème d'EDP du TP2) :

Nom	Taille	Source
BFW398A	398	Bounded finline dielectric waveguide
BWM200	200	Chemical engineering
GRE1107	1107	Simulation studies in computer systems
HOR131	434	Flow in networks
ORSIRR1	1030	Oil reservoir simulation

Pour chacun de ces cas tests un fichier ".mtx" contenant les données et un fichier ".m" permettant de lire ces données sont fournis. Afin de lire ces matrices, il suffit d'utiliser la commande Matlab suivante : `[A,matname] = matrice.`

Ci-dessous un exemple d'appel :

```
>> [A,matname]=bfw398a;
>> matname
matname =
BFW398A
>> size(A)
ans =
398 398
```