Une méthode de Krylov par blocs : Block-GMRES
Étude Comparative avec la méthode GMRES

1 Spécification du projet

Dans de nombreuses situations, il est intéressant de travailler avec des blocs de vecteurs au lieu d'un seul. Cela permet, par exemple, d'exploiter au mieux les noyaux de calcul de bibliothèques numériques (BLAS). Il y a aussi des considérations numériques liées à la nature des problèmes physiques résolus pour lesquels il y a plusieurs systèmes à résoudre avec la même matrice mais plusieurs seconds membres. Cela peut être effectué en utilisant des généralisations par blocs des méthodes de Krylov pour lesquelles la matrice A est toujours appliquée à un groupe de vecteurs (Y. Saad, Iterative Methods for Sparse Linear Systems).

Une méthode par blocs est la méthode Block-GMRES dont l'algorithme est donnée à la fin du document 5 dans sa version qui utilise la variante modifiée du procédé d'orthogonalisation de Gram-Schmidt.

2 Développement

La partie développement du projet est de coder cette méthode Block-GMRES **préconditionnée** sous la forme d'une fonction matlab avec un profil proche de la fonction `gmres` de matlab (il manque juste le paramètre de *restart*).

```
function [X, flag, relres, iter, resvec] = gmresblock(A, B, tol, max_it, M1, M2 X0)
```

Les paramètres d'entrée sont :

- `A`, la matrice du système que l'on cherche à résoudre,
- `B`, le second membre de ce système,
- `tol`, valeur seuil de l'erreur inverse à utiliser pour détecter la convergence,
- `max_it`, le nombre d'itérations maximum,
- `M1, M2`, préconditionneur explicite en deux facteurs, à utiliser pour accélérer la convergence (signification identique à l'interface Matlab de `gmres`),
- `X0`, la vecteur initial.

Les résultats sont :

- `X`, la solution calculée du système $AX = B$,
- `flag`, statut de sortie de la routine ; ce paramètre sera égal à 0 en cas de convergence et à 1 en cas de non convergence au bout de max_it itérations.
- `relres`, la norme relative du résidu (équivalent à l'erreur inverse η_b^N)
- `iter`, le nombre d'itérations,
- `resvec`, le vecteur des normes des résidus de chaque itération.

3 Évaluation, travail à rendre

Pour ce projet, vous rendrez 2 parties :

1. **Logiciel** : codes Matlab pour l'algorithme GMRES et Block-GMRES, ainsi que le ou les codes de validation de l'implantation ; on tiendra compte de la qualité du code (commentaires, indentation, ...),
2. **Rapport** : document décrivant les réalisations et les choix d'implantations logicielles, les tests élémentaires ainsi qu'une étude comparative entre l'approche par blocs et le l'approche classique GMRES : on pourra par exemple comparer (flops, nombre d'itérations), la résolution d'un système de p seconds membres via un appel à Block-GMRES par rapport à la résolution par p appels à GMRES [1].

   En première page de ce rapport au format **pdf** figurera votre nom.

La date de remise de ces parties est fixée au **vendredi 2 juin 2015**.

4 Annexe

L'ensemble suivant des matrices de test est fourni pour mener l'étude proposée :

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
```

1. Il serait intéressant pour cette comparaison d'évaluer le nombre de flops d'une itération de chacune des deux méthodes mais aussi la mémoire nécessaire à la résolution

5 Algorithme

To solve the system with multiple right-hand sides, $AX = B$:
- A, matrix of the problem of size $n \times n$
- B, matrix of right-hand sides of size $n \times p$ (p right-hand sides),
- X, matrix of solution vectors.

Algorithm 1 Block GMRES - MGS variant

Set the initial guess $X_0 \in \mathbb{R}^{n \times p}$
$R_0 = B - AX_0$;
Compute V_1 and R_1, results of the thin QR factorization of the block of initial residuals R_0 : $R_0 = [v_1, v_2, \ldots, v_p]R_1 = V_1 R_1$ with V_1, unitary matrix of size $n \times p$ and R_1 of size $p \times p$.
j = 0
while (not convergence) **and** (not max_it reached) **do**
    $j = j + 1$
    $W_j = AV_j$
    for $i = 1, \ldots, j$ **do**
        $H_{i,j} = V_i^T W_j$
        $W_j = W_j - V_i H_{i,j}$
    end for
    Compute V_{j+1} and $H_{j+1,j}$, results of the thin QR decomposition of W_j : $W_j = V_{j+1} H_{j+1,j}$
    Solve the least-squares problem $Y_j = \arg\min \|E_1 R_1 - \bar{H}_j Y\|_F$
    Set $X_j = X_0 + U_j Y_j$
end while

where
- $\bar{H}_j = (H_{i,k})_{1 \leq i \leq j+1, 1 \leq k \leq j}$, $H_{i,k} = 0$ for $i > k + 1$,
- at step j, E_1, first p colums of $I_{(j+1)*p}$,
- $U_j = [V_1, V_2, \ldots, V_j]$,
- $\|.\|_F$ is the Frobenius norm, $\|A\|_F^2 = \sum_j \|A(:, j)\|_2^2$.