# Méthodes GMRES et FOM

## 1 Implantation des méthodes GMRES et FOM

On cherche à résoudre le système linéaire : $Ax = b$.

Pour cela on considère les méthodes GMRES et FOM, pour lesquelles on s'intéressera aux historiques de convergence de l'erreur inverse *normwise*

$$\eta_b^N(x_m) = \frac{\|b - Ax_m\|}{\|b\|}.$$

Les itérations seront stoppées lorsque $\eta_b^N(x_m)$ sera inférieur à un seuil $\epsilon$ ou que l'on aura dépassé un nombre maximum d'itérations.

Ces méthodes seront implantées sous la forme d'une unique fonction Matlab.

$$[x, flag, relres, iter, resvec] = krylov(A, b, x0, tol, maxit, type)$$

Les paramètres de cette fonctions seront :

1. `A`, la matrice du système que l'on cherche à résoudre
2. `b`, le second membre de ce système
3. `x0`, le vecteur initial
4. `tol`, le seuil demandé
5. `maxit`, le nombre maximum d'itérations
6. `type`, 0 pour FOM, 1 pour GMRES

Cette fonction renverra les mêmes résultats que l'appel complet à `gmres` de Matlab :

1. `x`, la solution
2. `flag` qui indique si la méthode a convergé (voir code)
3. `relres`, la norme relative du résidu ($\equiv \eta_b^N(x_m)$)
4. `iter`, le nombre d'itérations
5. `resvec`, le vecteur des normes des résidus de chaque itération

## 2 Algorithme GMRES

On rappelle dans les grandes lignes l'algorithme GMRES pour lequel on cherche une solution $x_m$ dans l'espace de Krylov de dimension $m$, $x_0 + \mathcal{K}_m$. L'algorithme utilise le procédé modifié de Gram-Schmidt pour construire la base orthonormée.

**Algorithm 1** GMRES - MGS variant

```
1:  Set the initial guess x_0
2:  r_0 = b − A x_0 ; β = ‖r_0‖
3:  v_1 = r_0/‖r_0‖ ;
4:  for j = 1, 2, ..., m do
5:      w_j = A v_j
6:      for i = 1 to j do
7:          h_{i,j} = v_i^T w_j
8:          w_j = w_j − h_{i,j} v_i
9:      end for
10:     h_{j+1,j} = ‖w_j‖
11:     if h_{j+1,j} = 0 then
12:         goto 16
13:     end if
14:     v_{j+1} = w_j / h_{j+1,j}
15: end for
16: Solve the least-squares problem y_j = arg min ‖β e_1 − H̄_j y‖
17: Set x_j = x_0 + V_j y_j and r_j = b − A x_j
```

## Remarques

1. On rappelle aussi que cet algorithme **doit** être adapté pour s'arrêter lorsque l'itéré courant vérifie le critère de convergence (erreur inverse relative inférieure à une certaine tolérance) ou lorsque le nombre d'itérations est supérieur au nombre d'itérations maximum accepté (paramètres de la fonction `krylov`).
2. La résolution du problème de moindre-carrés dans GMRES et celle du système linéaire dans FOM seront effectuées en utilisant la commande Matlab "\" (`help \`).
3. Enfin, n'oublions pas le résultat concernant l'estimation de la norme du résidu (à la fois pour FOM et pour GMRES) qui évite de calculer le résidu à chaque itération (à mettre en place une fois que cela fonctionne correctement avec un vrai calcul du résidu). Il vous faudra peut-être adapter la phase de résolution pour les deux algorithmes en n'utilisant plus le "\".

## 3 Tests

Vous testerez votre fonction sur les matrices fournies en commençant par la matrice `mat1` puis `pde225_5e-1` et en terminant par `hydcar20` (commande Matlab **load**).

Vous prendrez comme second membre $b = (1, 2, 3, \cdots)^T$.

Vous pourrez visualiser l'historique de convergence de l'erreur inverse avec la commande **semilogy** et, pour GMRES, le comparer avec celui de la fonction Matlab `gmres`.

## 4 Rendu (en binôme)

Déposez sous Moodle dans une archive dont le nom comporte les noms des deux membres du binôme et comportant

1. les codes développés (fichiers tp.m et krylov.m),
2. un petit rapport montrant le bon fonctionnement des solveurs (tableaux du nombre d'itérations, de la précision obtenue) ainsi que les figures de l'évolution de la norme du résidu et la comparaison par rapport à la fonction `gmres` de Matlab.

   Vous pourrez faire varier la tolérance demandée et voir à partir de quelle valeur les problèmes deviennent difficiles à résoudre.

Ce rendu est à déposer sous Moodle.

## Matlab

- `help`
- `zeros`
- `norm`
- `for`

Inutile de considérer que certains objets ont besoin d'avoir un indice pour les différencier d'une itération à l'autre : on peut par exemple utiliser la même variable pour tous les $w_j$ et tous les $x_j$.

Par contre les $v_j$ et $h_{ij}$ sont bien à mémoriser dans les objets $V$ et $H$ présentés en séance de CTD.