DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MATHÉMATIQUES DE L'APPRENTISSAGE PROFOND
DEVOIR 2

10 mars 2022

Directives : *Ce devoir doit être complété pour jeudi le 31 mars. Il doit être déposé sur le site Moodle du cours en un seul fichier, en format pdf. Vous devez utiliser le langage Julia.*

Modules Julia

Pour ce devoir, vous aurez besoin des modules `Plots`, `LinearAlgebra`, `MatrixDepot` et `IterativeSolvers`.

Fonctions Julia

Vous aurez besoin de la fonction `arnoldi.jl`, qui contient une implémentation de la méthode d'Arnoldi. Elle est disponible sur le site du cours. Pour y accéder, utiliser la commande

```
include("arnoldi.jl");
```

Pour l'appeler, utiliser la commande

```
Q,H = arnoldi(A,u,m)
```

Avec ce que nous avons vu en classe, et avec les commentaires en début de fonction, l'utilisation de cette fonction ne devrait pas vous causer de problèmes.

1. On veut comparer la base de Krylov avec la base calculée par la méthode d'Arnoldi. Nous allons travailler avec le système linéaire suivant :

   ```
   lambda = @. 10 + (1:100)
   A = triu(rand(100,100),1) + diagm(0=>lambda)
   b = rand(100);
   ```

   A est une matrice triangulaire supérieure de dimensions $n \times n = 100 \times 100$.

   Comme nous avons vu en classe, nous ne voulons pas utiliser la base de Krylov complète. Nous voulons plutôt travailler avec un sous-espace de dimension $m \ll n$.

   (a) Construire les matrices de Krylov K_m pour $m = 1, \ldots, 30$. En fait, vous n'avez qu'à construire K_{30}, puis vous n'utiliserez que les colonnes voulues. Puisqu'il peut y avoir de grandes variations dans l'ordre de grandeur des termes $A^k\boldsymbol{b}$, normalisez chaque colonne calculée afin d'éviter les instabilités numériques.

   (b) On veut ensuite exprimer la solution de $A\boldsymbol{x} = \boldsymbol{b}$ dans les bases formées par les colonnes des K_m. On doit donc approcher $\boldsymbol{x}$ au sens des moindres carrés :

   $$\min_{\boldsymbol{x}\in \boldsymbol{C}(K_m)} \|A\boldsymbol{x} - \boldsymbol{b}\|^2 = \min_{\boldsymbol{z}\in\mathbb{R}^m} \|(AK_m)\boldsymbol{z} - \boldsymbol{b}\|^2.$$

   Faites la résolution au sens des moindres carrés de ce problème pour $\boldsymbol{x} \in \boldsymbol{C}(K_m)$, où $m = 1, \ldots, 30$. L'opérateur « \ » (la barre oblique inversée) de Julia bascule vers la résolution au sens des moindres carrés pour les matrices rectangulaires. Calculer la norme du résidu $\boldsymbol{b} - A\boldsymbol{x}$ pour chaque m, que vous illustrerez dans un graphique à l'aide de la commande `plot`, dans un repère semi-logarithmique. Qu'observez-vous ?

   (c) Reprendre la sous-question précédente, mais en utilisant la base orthonormale donnée par la fonction du fichier `arnoldi.jl`. On se rendra à une base de dimension $m = 60$. Qu'observez-vous ?

2. Pour de grandes valeurs de n, le coût de la méthode GMRES en calculs et en espace mémoire utilisé devient prohibitif. Dans ce cas, on peut utiliser la version « redémarrée » (« *restarted* ») de la méthode, GMRES(k). Après k étapes de la méthode GMRES, l'algorithme est redémarré en utilisant $\boldsymbol{x}_k$ comme approximation initiale.
   (a) Comparer le nombre d'opérations (on veut un ordre de grandeur) et l'utilisation de la mémoire des méthodes GMRES et GMRES(k) pour k fixé, pour des valeurs de n de plus en plus grandes.
   (b) Donner un exemple pour lequel on peut s'attendre à ce que la méthode GMRES(k) converge en presqu'aussi peu d'itérations que la méthode GMRES (et donc en beaucoup moins d'opérations).
   (c) Donner un exemple pour lequel la méthode GMRES(k) ne convergera probablement pas, alors que la méthode GMRES convergera.

3. On veut expérimenter l'utilisation de la version redémarrée de la méthode GMRES. On va d'abord chercher une matrice provenant d'une discrétisation de l'équation de Poisson chez « Matrix Depot », puis on construit le système linéaire associé :

```
d = 50;
A = d^2*matrixdepot("poisson",d);
b = ones(n);
```

   Vous pourriez être intéressés de voir le profil de la matrice avec la commande `spy(A)`.

   On veut comparer la méthode GMRES avec la méthode redémarrée GMRES(k) pour $k = 20, 40, 60$ et 120 (le nombre maximal d'itérations qu'on acceptera). On utilisera la commande

```
x,hist = gmres(A,b,reltol=1e-8,restart=k,maxiter=120,log=true,verbose=true);
```

   Faire un graphe des itérées dans un repère semi-logarithmique. Commenter ce que vous observez.

Steven Dufour