MTH8211

Laboratoire 2

Dominique Orban

Rappel : les rapports de laboratoire sont strictement personnels.

Rendez vos réponses sous forme d'un carnet Jupyter (extension `ipynb`). Pour ce faire, exécutez les commandes suivantes à l'invite Julia :

```
pkg> add IJulia  # utiliser ] pour entrer en mode "package"
julia> using IJulia  # utiliser DEL pour en sortir
julia> notebook(detached=true, dir=".")  # une fenêtre de votre browser devrait apparaître
```

Laboratoire 2

Ce laboratoire consiste à implémenter la factorisation suivante d'une matrice carrée hermitienne :

$$A = LDL^*,$$

où

- L est triangulaire inférieure unitaire, c'est-à-dire que sa diagonale n'est composée que de 1 ;
- D est diagonale.

On se concentre sur la factorisation dense, i.e., A est donnée sous forme de tableau 2D.

Partie 1 (5 pts)

Si on suppose A définie positive, on peut garantir l'existence de cette factorisation sans devoir pivoter. Elle est en fait équivalente à la factorisation de Cholesky.

1. Dans ce cas de figure, expliquer pourquoi D sera également définie positive ;
2. rédiger l'algorithme qui permet de découvrir les éléments de L et D colonne par colonne ;

3. lire la documentation des types `Symmetric` et `Hermitian` ;
4. implémenter cet algorithme en Julia sous forme d'une fonction `(L, D) = ldl(A)` qui prend en entrée une matrice de type `Symmetric` ou `Hermitian` et qui renvoie `L` de type `UnitLowerTriangular` et `D` de type `Diagonal` ;
5. écrire une fonction `x = solve(L, D, b)` permettant de résoudre le système $Ax = b$ sur base de cette factorisation ;
6. tester votre implémentation sur des systèmes réels symétriques et complexes hermitiens.

Votre implémentation doit fonctionner correctement si un seul des deux triangles de A est représenté explicitement, comme par exemple

```
A = rand(10, 10)
A = Symmetric(triu(A' * A))      # triangle supérieur
A = Symmetric(tril(A' * A), :L)  # triangle inférieur
```

Partie 2 (1 pt)

La factorisation LDL^* existe également pour d'autres matrices sans pivoter, telles que les matrices de la forme

$$\begin{bmatrix} M & A^* \\ A & -N \end{bmatrix},$$

où M et N sont hermitiennes définies positives et A est arbitraire (A peut être rectangulaire). Ces matrices sont dites *quasi définies*. Dans ce cas de figure, D n'est plus définie positive, mais votre implémentation fournit toujours la bonne factorisation. Tester votre implémentation sur des systèmes quasi définis.

Partie 3 (4 pts)

1. Implémenter une variante de votre factorisation qui n'alloue aucune nouvelle mémoire et qui mémorise les éléments de L et D par-dessus les éléments de A. Cette nouvelle fonction renvoie une seule valeur `LD`, du même type que `A`.
2. Ajuster votre fonction de résolution d'un système à cet objet `LD`, lui donner la signature `x = solve(LD, b)` et tester votre implémentation.

Bon travail !

Dominique Orban