

LINMA1170 – Devoir 1: Résolution de systèmes linéaires

Mattéo Couplet, Thomas Leyssens, Prof. Jean-François Remacle

Vous avez vu en cours l'algorithme de la décomposition LU, qui permet de résoudre des systèmes linéaires. Dans ce devoir (individuel), on vous demande d'implémenter l'algorithme classique de décomposition LU ainsi que les substitutions avant et arrière pour résoudre un système linéaire correspondant. Vous écrirez un script de test pour vérifier que votre implémentation est correcte, et vous analyserez la complexité asymptotique de vos algorithmes.

1. On vous fournit les *headers* `matrix.h` et `lu.h`. Chaque fichier contient les spécifications de deux fonctions à implémenter (celles de `matrix.h` ont déjà été faites en TP). Écrivez `matrix.c` et `lu.c` en définissant les fonctions demandées. **Ne changez pas les déclarations des fonctions fournies : nous les testerons automatiquement pour vérifier que vos codes sont corrects !** Vous êtes cependant encouragés à définir d'autres fonctions dans les modules, ou même d'autres modules si nécessaire : une bonne organisation du code sera valorisée.
2. Écrivez un script de test qui vérifie que vos codes sont corrects : par exemple, définissez une matrice A et un vecteur b aléatoires, et vérifiez que $A = L * U$ et que $Ax = b$. Expliquez dans votre rapport ce que fait votre script de test et comment le compiler et l'exécuter.
3. Quelle est la complexité asymptotique de vos algorithmes ? Démontrez cette complexité en mesurant le temps pour des matrices de taille croissante, et en traçant les données sur un graphe log-log (taille, temps), avec Excel ou `matplotlib` par exemple. Une bonne idée est d'écrire les données dans un fichier de la manière suivante :

```
$ ./monProgramme > mesDonnées.txt
```

puis d'exploiter les données avec Excel ou Python. Commenter les résultats obtenus. À nouveau, montrez comment compiler et exécuter votre code.

Délivrables :

- Les codes que vous avez écrits pour répondre aux questions.
- Un rapport de maximum 2 pages.