

---

Dans ce devoir, nous nous intéressons à la densité des matrices issues d'une décomposition LU, sans pivotage. Les matrices à factoriser sont celles générées par le modèle éléments finis `ccore.py`.

### Format plein

Soit une matrice hermitienne  $A \in \mathbb{C}^{n \times n}$  de faible densité<sup>1</sup>.

Nous vous demandons d'étudier la densité de sa décomposition LU. Pour ce faire, écrivez une fonction PYTHON3 `LUfactorize(A)` qui effectue en place cette décomposition, où `A` est un `numpy.array`; cette fonction ne retourne donc rien.

Au moyen d'une expérience numérique, évaluez la complexité temporelle pour résoudre un système linéaire  $Ax = b$  via une décomposition LU.

### Format creux

Soit une matrice carrée  $A \in \mathbb{C}^{n \times n}$  de faible densité.

Implémentez une fonction PYTHON3 `CSRformat(A)` effectuant la conversion d'une matrice (format plein) en format CSR. Cette fonction reçoit un `numpy.array` `A` et retourne le tuple `(sA, iA, jA)` composé de trois `numpy.array`. Ce tuple contient respectivement les entrées non nulles de `A`, les débuts (et fins) de chaque ligne, les indices des colonnes correspondantes. En d'autres mots :

- `sA[s]` donne le  $s$ -ème élément non nul de `A`
- `iA[i]` donne l'indice dans `sA` du premier élément non nul sur la ligne  $i$
- `jA[j]` donne l'indice de la colonne contenant le  $j$ -ème élément non nul de `A`

Évaluez les performances de votre convertisseur.

Ensuite, écrivez une fonction PYTHON3 `LUcsr(sA, iA, jA)` effectuant la décomposition LU d'une matrice sous format CSR<sup>2</sup>. Cette fonction retourne le tuple `(sLU, iLU, jLU)` correspondant à la décomposition LU en place, sous format CSR.

Par ailleurs, nous vous demandons d'étudier l'influence de l'algorithme `reverse Cuthill McKee` sur la densité de la décomposition LU. Il vous faut donc écrire une fonction PYTHON3 `RCMK(iA, jA)` qui retourne `r`, un `numpy.array` représentant le vecteur de permutation.

Évaluez la complexité temporelle pour résoudre un système linéaire  $Ax = b$  via la décomposition LU *creuse*, avec et sans l'algorithme `RCMK`. Concluez de ses performances vis-à-vis de votre expérience pour la LU pleine.

---

1. Telle que celles produites par le modèle `ccore.py`.

2. A aucun moment une matrice de format plein ne doit être construite.

## Consignes <sup>3</sup>

Ce devoir est absolument **individuel**.

Un rapport de quatre pages (càd deux rectos et deux versos) de texte et figures est à remettre lors du cours du 6 décembre. Les implémentations PYTHON sont à soumettre dans le même temps (8h max) sur Moodle : (i) votre fichier `lu.py` comprenant vos fonctions `LUfactorize`, `CSRformat`, `LUcsr` ainsi que vos scripts réalisant les expériences demandées. Le `.pdf` de votre rapport, ainsi que le `.tex` (figures) sont également à soumettre sur Moodle. Le tout est compressé dans un dossier `.zip` (~~`.rar`~~). Nous vous rappelons que vos travaux seront soumis à un logiciel anti-plagiat.

Le rapport ne doit pas contenir de page de garde, seulement une entête reprenant au moins le nom de l'auteur. Le rapport ne doit pas spécialement contenir de code source. La langue de rédaction est le français. Le rapport doit être réalisé avec L<sup>A</sup>T<sub>E</sub>X, avec la `documentclass article` [11pt] en `pagestyle plain`. Veillez à fournir des implémentations lisibles, dûment commentées, avec des noms de variables explicites.

Les implémentations sont à réaliser en PYTHON 3.6. Les bibliothèques admises sont :

- NUMPY
- MATPLOTLIB
- TIME

Toute autre bibliothèque ne sera pas acceptée ; **SCIPY est donc prohibée**.

---

3. Tout non respect des consignes entraîne une pénalité plus ou moins importante selon l'irrégularité commise.