UCLouvain

LINMA1170 Analyse Numérique
Décomposition LU et conditionnement
Rapport

Devoir 2

Élève :
Pierre Vandermeersch

Enseignant :
Jean-François Remacle

11 mars 2024

1 Question 1

1.1 Nombre de conditionnement

Soit $f : X \mapsto Y$ avec X, Y deux espaces vectoriels normés. Le conditionnement du problème que f définit, où X est l'espace des données et Y l'espace des solutions, peut être vu comme l'analyse de la sensibilité de f par rapport à des petites perturbations de $\partial x \in X$. On étudie donc les variations de $\partial f = f(x + \partial x) - f(x)$.
Le nombre de conditionnement κ est défini comme le plus grand facteur du rapport à la limite, lorsque la perturbation ∂x tends vers 0, entre la variation relative de f c'est-à-dire $\frac{\|\partial f\|}{\|f(x)\|}$ et la variation relative de x défini comme $\frac{\|\partial x\|}{\|x\|}$. On a donc

$$\kappa = \lim_{\Delta \to 0} \sup_{\|\partial x\| \leq \Delta} \left(\frac{\|\partial f\| / \|f(x)\|}{\|\partial x\| / \|x\|} \right). \tag{1}$$

1.2 Moindres carrés

Dans le problème des moindres carrés, le conditionnement de x par rapport à A $\kappa_{A \to x}$ est donné par [1]

$$\kappa_{A \to x} = \kappa(A) + \frac{\kappa(A)^2 \tan \theta}{\eta} \tag{2}$$

et le conditionnement de x par rapport à b $\kappa_{b \to x}$ est donné par

$$\kappa_{b = \to x} = \frac{\kappa(A)}{\eta \cos \theta} \tag{3}$$

avec $\eta = \frac{\|A\| \|x\|}{\|Ax\|}$ et $\theta = \cos^{-1} \frac{\|Ax\|}{\|b\|}$.

1.3 Analyse numérique

Numériquement, on peut vérifier ces définitions en approximant la limite tel que décrite dans (1) avec un $\epsilon \approx 10^{-10}$ et un ∂x dans un grand nombre p de directions aléatoires différentes ($p = 1000$ dans ce cas-ci) [2].
Tel que montré dans la figure 1, on peut vérifier que les équations (2) et (3) donnent bien une bonne valeur de (1) puisque l'erreur entre elles reste relativement petite même pour des matrices A et b de tailles croissantes. On peut également remarquer que pour des matrices aléatoires A et b le problème est bien conditionné avec $\kappa_{A \to x}, \kappa_{b \to x} << 10^6$.

2 Question 2

2.1 Algorithme Décomposition LU

Soit $A \in \mathbb{R}^{m \times n}$ avec $m \geq n$, l'algorithme pour transformer A en LU avec $L \in \mathbb{R}^{m \times m}$ et $U \in \mathbb{R}^{m \times n}$. L'algorithme de la décomposition LU sans pivot s'écrit en Python (en utilisant Numpy) tel que

1. cf. Numerical Linear Algebra de Nick Trefethen p.131
2. Voir conditionnement.py

```python
1  import numpy as np
2  m, n = A.shape
3  L = np.eye(m)
4  U = A.copy()
5
6  for k in range(m - 1): # Boucle exterieure
7      for i in range(k + 1, m): # Boucle interieure
8          factor = U[i, k] / U[k, k]
9          L[i, k] = factor
10         U[i, k:] -= (factor * U[k, k:])
11
12 return L, U
```

2.2 Complexité

Tout d'abord, on note que la boucle extérieure tourne $m-1$ fois. La boucle intérieure tourne $m-1-(k+1)=m-k-2$. Dans cette boucle intérieure, on a une division et $m-k$ multiplication-soustraction, soit $m-k+1$ flops. Etant donné qu'on s'interesse à la limite lorsque m tends vers l'infini, on peut dire que $m-1 \approx m$, $m-k-2 \approx m-k$ et $m-k+1 \approx m-k$. On a donc

$$
\begin{aligned}
\#\text{ flops} &= \sum_{k=1}^{m} \left(\sum_{j=k+1}^{m} m-k \right) && (4)\\
&= \sum_{k=1}^{m} (m-k)(m-k) = \sum_{k=1}^{m} m^2 - 2km + k^2 && (5)\\
&= \sum_{k=1}^{m} m^2 + \sum_{k=1}^{m} (-2km) + \sum_{k=1}^{m} k^2 && (6)\\
&= m^3 - 2m\frac{m(m+1)}{2} + \frac{m(m+1)(2m+1)}{6} && (7)\\
&\approx m^3 - m^3 + \frac{(m^2+m)(2m+1)}{6} \approx \frac{1}{3}m^3. && (8)
\end{aligned}
$$

On a donc une complexité en $\mathcal{O}(\frac{1}{3}m^3)$ pour la décomposition LU sans pivot [3]. On peut le vérifier numériquement à l'aide de Python (voir figure 2).

2.3 Parallèlisation

La décomposition LU est parallélisable au niveau de la ligne 10 du code ci-dessus. En effet, la multiplication-soustraction peut être parallèlisé avec chaque thread qui soustrait $factor * U[k, j]$ à $U[i, j]$ pour $j \in [k, m]$.

3. A noter que si on considère la multiplication-soustraction comme 2 flops et pas une, on a $\mathcal{O}(\frac{2}{3}m^3)$.

3 Question 3

On va procéder par induction sur n. Soit $A \in \mathbb{R}^{m\times n}$, $m \geq n$, de rang n. Pour $n = 1$, on a $A^{(1)} \in \mathbb{R}^m$ et on suppose que le mineur principal de $A^{(1)}$ soit positif, alors

$$A^{(1)} = \begin{pmatrix} \alpha_{11} \\ a_{m1} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 \\ \frac{a_{m1}}{\alpha_{11}} \end{pmatrix}}_{L^{(1)}} \cdot \underbrace{\alpha_{11}}_{U^{(1)}}. \tag{9}$$

Le mineur positif de $A^{(1)}$ est $\alpha_{11} > 0$. Donc $L^{(1)}$ et $U^{(1)}$ existent bien et sont uniques. A noter que $a_{m1} \in \mathbb{R}^{m-1}$. Maintenant, supposons que ce soit vrai pour $n = k$. C'est-à-dire, si les mineurs principaux de $A^{(k)}$ sont positifs, alors ils existent $L^{(k)}$ et $U^{(k)}$ tel que $A^{(k)} = L^{(k)}U^{(k)}$. Prenons désormais $n = k + 1$. Si $A^{(k+1)} = L^{(k+1)}U^{(k+1)}$, elle est de la forme

$$\begin{pmatrix} A_{11} & a_{k1} \\ a_{1k}^T & \alpha_{11} \\ A' & a' \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ l_{1k}^T & 1 \\ L' & l' \end{pmatrix} \begin{pmatrix} U_{11} & u_{k1} \\ 0 & u' \end{pmatrix} \tag{10}$$

qui donne le système d'équations

$$\begin{cases} L_{11}u_{k1} = a_{k1}, \\ u' = \alpha_{11} - l_{1k}^T u_{k1}, \\ l' = \frac{a' - L'u_{k1}}{u'}. \end{cases} \tag{11}$$

Par l'hypothèse inductive, L_{11}, l_{1k}^T, L' existent et sont uniques.

1. L_{11} n'est pas singulière, alors $u_{k1} = L_{11}^{-1}a_{k1}$ existe et est unique.
2. On sait aussi que l_{1k}^T, α_{11} existent et sont uniques donc $u' = \alpha_{11} - l_{1k}^T u_{k1}$ existe et est unique aussi. Par hypothèse, tous les mineurs principaux de $A^{(k+1)}$ sont positifs donc $\det(A^{(k)}) > 0$ et donc $u' \neq 0$.
3. Et enfin, puisque u' existe, est unique et non nul, $l' = \frac{a' - L'u_{k1}}{u'}$ existe et est unique.

On a donc bien une factorisation LU de $A^{(k+1)}$ sans pivotage. Par le principe d'induction, cela tient pour tout $A \in \mathbb{R}^{m\times n}$, $m \geq n$, de rang n dont tous les mineurs principaux sont positifs.

4 Question 4

Etant donné que A^TA est symétrique définie positive, on peut utiliser la factorisation de Cholesky pour résoudre le problème des moindres carrés. La complexité de l'algorithme de Cholesky est de $\mathcal{O}(\frac{1}{6}n^3)$ [4] pour une matrice $A \in \mathbb{R}^{m\times n}$. L'algorithme de Cholesky requiert donc 2 fois moins d'opérations que la décomposition LU. On peut vérifier ce gain numériquement tel que dans la figure 3.

4. Comme pour la décomposition LU, si l'on considère les multiplications-soustraction comme deux flops distinctes alors la complexité est de $\mathcal{O}(\frac{1}{3}n^3)$

5 Graphes

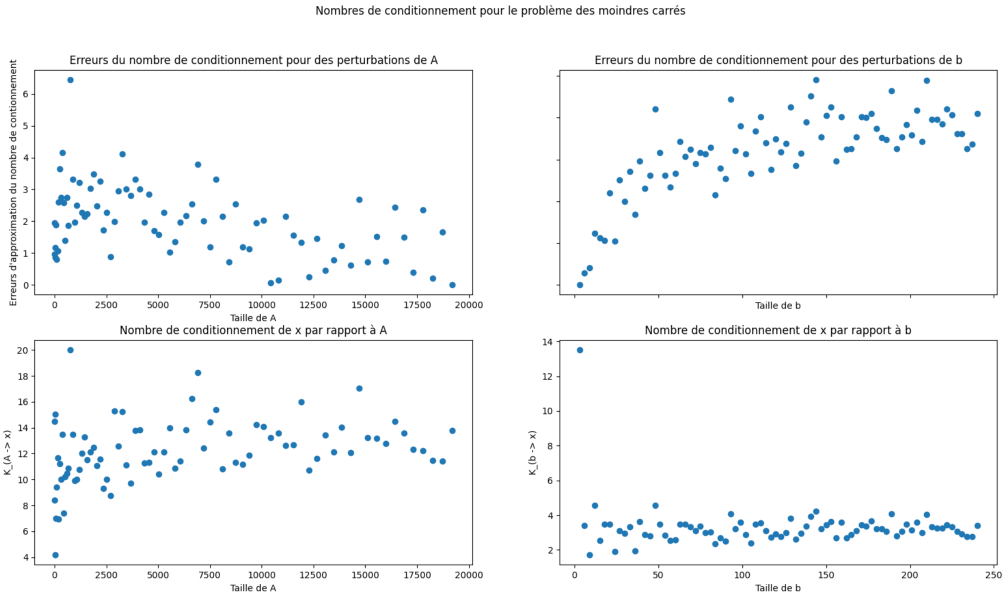

FIGURE 1 – Graphe du conditionnement de x en fonction de A et b ainsi que l'erreur entre les définitions (2), (3) avec (1).

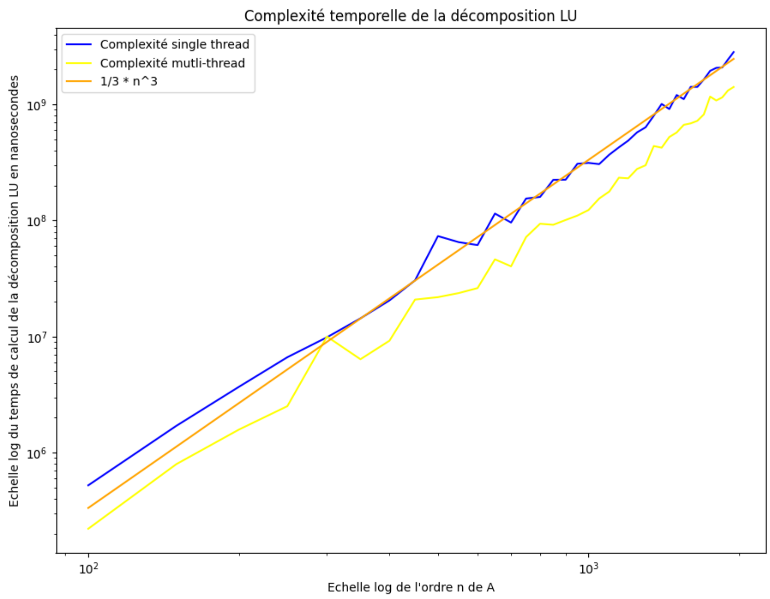

FIGURE 2 – Graphe de la complexité temporelle de la décomposition LU.

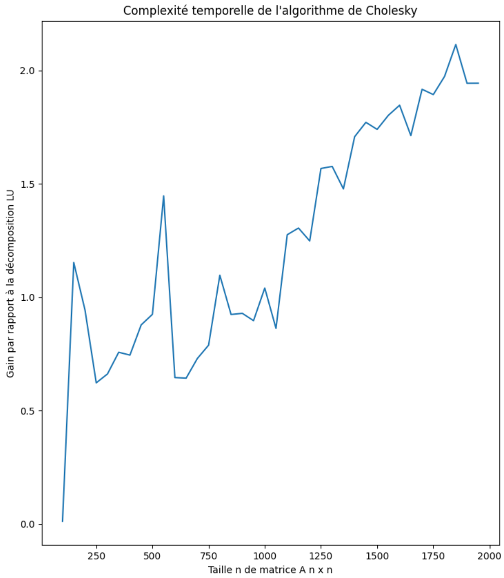

FIGURE 3 – Graphe du gain de la complexité de l'algorithme de Cholesky par rapport