Álgebra Lineal Computacional

2do Cuatrimestre 2023

Laboratorio N° 5: Descomposición LU.

Resolveremos los siguientes ejercicios de la Práctica 3:

Ejercicio 3. Escribir funciones de `Python` que calculen la solución de un sistema:

(a) $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{b}$, siendo $\boldsymbol{L}$ triangular inferior.

(b) $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$, siendo $\boldsymbol{U}$ triangular superior.

Ejercicio 4. Escribir funciones de `Python` que realicen las siguientes tareas:

(a) Calcular la descomposición LU de una matriz dada $\boldsymbol{A}$, asumiendo que no es necesario realizar pivoteos.

(b) Resolver un sistema $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, utilizando la función del ítem anterior y las del ejercicio 3. Aplicar esta función para resolver el ítem c. del ejercicio 2

Consideremos como ejemplo a la siguiente matriz

$$A = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 4 & 3 & 3 & 4 \\ -2 & 2 & -4 & -12 \\ 4 & 1 & 8 & -3 \end{pmatrix}$$

y los siguientes pasos de triangulación, donde $A^{(k)}$ es la matriz que se obtiene a partir de A por el método de eliminación Gaussiana cuando las primeras k columnas ya han sido trianguladas.

<u>1er Paso:</u>

$$A = A^{(0)} = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 4 & 3 & 3 & 4 \\ -2 & 2 & -4 & -12 \\ 4 & 1 & 8 & -3 \end{pmatrix} \rightsquigarrow \begin{array}{l} F_2 \leftarrow F_2 - 2 \cdot F_1 \\ F_3 \leftarrow F_3 - (-1) \cdot F_1 \\ F_4 \leftarrow F_4 - 2 \cdot F_1 \end{array} \rightsquigarrow A^{(1)} = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 0 & 1 & -1 & -2 \\ 0 & 3 & -2 & -9 \\ 0 & -1 & 4 & -9 \end{pmatrix}$$

$$\text{Luego, } A^{(1)} = L_1 A^{(0)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ -2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 2 & 3 \\ 4 & 3 & 3 & 4 \\ -2 & 2 & -4 & -12 \\ 4 & 1 & 8 & -3 \end{pmatrix}$$

2do Paso:

$$A^{(1)} = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 0 & 1 & -1 & -2 \\ 0 & 3 & -2 & -9 \\ 0 & -1 & 4 & -9 \end{pmatrix} \rightsquigarrow \begin{array}{l} F_3 \leftarrow F_3 - 3 \cdot F_2 \\ F_4 \leftarrow F_4 - (-1) \cdot F_2 \end{array} \rightsquigarrow A^{(2)} = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 0 & 1 & -1 & -2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 3 & -11 \end{pmatrix}$$

Luego, $A^{(2)} = L_2 A^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 2 & 3 \\ 0 & 1 & -1 & -2 \\ 0 & 3 & -2 & -9 \\ 0 & -1 & 4 & -9 \end{pmatrix}$

3er Paso:

$$A^{(2)} = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 0 & 1 & -1 & -2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 3 & -11 \end{pmatrix} \rightsquigarrow F_4 \leftarrow F_4 - 3 \cdot F_3 \rightsquigarrow A^{(3)} = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 0 & 1 & -1 & -2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & -2 \end{pmatrix}$$

Luego, $A^{(3)} = L_3 A^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -3 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 2 & 3 \\ 0 & 1 & -1 & -2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 3 & -11 \end{pmatrix}$

Finalmente llegamos a que

$$U = A^{(3)} = L_3 L_2 L_1 A \Longleftrightarrow A = \underbrace{L_1^{-1} L_2^{-1} L_3^{-1}}_{L} U$$

$$A = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 4 & 3 & 3 & 4 \\ -2 & 2 & -4 & -12 \\ 4 & 1 & 8 & -3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -1 & 3 & 1 & 0 \\ 2 & -1 & 3 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 2 & 3 \\ 0 & 1 & -1 & -2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & -2 \end{pmatrix} = LU$$

Ejercicios para el laboratorio

1. Completar la función `elim_gaussiana.py` de `Python` de tal forma que en la iteración k-ésima del algoritmo se calcule la matriz $\widetilde{A}^{(k)}$ según se muestra a continuación.

$$A = \widetilde{A}^{(0)} = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 4 & 3 & 3 & 4 \\ -2 & 2 & -4 & -12 \\ 4 & 1 & 8 & -3 \end{pmatrix} \rightsquigarrow \widetilde{A}^{(1)} = \begin{pmatrix} 2 & 1 & 2 & 3 \\ \color{red}{2} & 1 & -1 & -2 \\ \color{red}{-1} & 3 & -2 & -9 \\ \color{red}{2} & -1 & 4 & -9 \end{pmatrix} \rightsquigarrow$$

$$\rightsquigarrow \widetilde{A}^{(2)} = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 2 & 1 & -1 & -2 \\ -1 & 3 & 1 & -3 \\ 2 & -1 & 3 & -11 \end{pmatrix} \rightsquigarrow \widetilde{A}^{(3)} = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 2 & 1 & -1 & -2 \\ -1 & 3 & 1 & -3 \\ 2 & -1 & 3 & -11 \end{pmatrix}$$

De esta forma logramos optimizar espacio de almacenamiento ya que los coeficientes que formarán parte de la matriz L se almacenan en la parte triangular inferior de $\widetilde{A}^{(k)}$ (se muestran en rojo en el ejemplo). Notar que estos valores en rojo se corresponden a los valores en 0 que fuimos poniendo en cada paso de la triangulación de las $A^{(k)}$. Para $k = n-1$, se obtiene en la parte triangular superior de $\widetilde{A}^{(k)}$ a los coeficientes de U.

Luego,

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -1 & 3 & 1 & 0 \\ 2 & -1 & 3 & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 2 & 1 & 2 & 3 \\ 0 & 1 & -1 & -2 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & -11 \end{pmatrix}$$

2. Además la función de `Python` debe retornar la cantidad de operaciones aritméticas realizadas. Se deberá llevar un conteo de la cantidad de sumas, restas, multiplicaciones y divisiones que se realizan durante la triangulación.

3. Realizar un gráfico que muestre la cantidad de operaciones en función del tamaño de la matriz. Para ello, considerar como ejemplo a la siguiente matriz $\boldsymbol{B}_n = (b_{ij}) \in \mathbb{R}^{n\times n}, n \geq 2$, definida como

$$b_{ij} = \begin{cases} 1 & \text{si } i = j \text{ o } j = n, \\ -1 & \text{si } i > j, \\ 0 & \text{en otro caso.} \end{cases}$$

Ejemplo: $\boldsymbol{B}_2 = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \boldsymbol{B}_3 = \begin{pmatrix} 1 & 0 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 \end{pmatrix}, \boldsymbol{B}_4 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \end{pmatrix}.$

Siendo $\boldsymbol{B}_n = L_n U_n$ la descomposición LU de $\boldsymbol{B}_n$, verificar que $\|U_n\|_\infty = 2^n$.

4. Realizar funciones en `Python` que implementen la sustitución hacia atrás y la sustitución hacia adelante para resolver sistemas triangulares compatibles determinados. Para diferentes vectores aleatorios $\boldsymbol{b} \in \mathbb{R}^n$ resolver el sistema $\boldsymbol{B}_n \boldsymbol{x} = \boldsymbol{b}$ como se indica en el Ejercicio 3.