Universidad de los Andes
Departamento de Ingeniería Eléctrica y Electrónica
IELE 2009L – Laboratorio de computación científica para IEE
Gerardo Andrés Riaño Briceño (201112388)

Informe laboratorio 4

Descomposición LU

Para entender el algoritmo de descomposición LU, vale la pena destacar que éste caracteriza de manera sencilla el proceso de eliminación Gaussiana, con el cual se puede obtener una matriz triangular superior a partir de una matriz dispersa. La matriz U es el resultado de la eliminación Gaussiana y la matriz L es el registro de las operaciones para conseguir dicho resultado.

Sea $A \in \mathrm{R}^{\mathrm{n}\times\mathrm{n}}$, una matriz dispersa y no singular. Se requieren al menos n operaciones para conseguir una matriz triangular superior a partir del proceso de eliminación Gaussiana con la matriz A. Si las operaciones entre filas en dicho proceso se representan como un producto de matrices se tiene que,

$$E_n \cdots E_3 E_2 E_1 A = U$$

Por ende,

$$L = E_1^{-1} E_2^{-1} E_3^{-1} \cdots E_n^{-1}$$

$$A = LU$$

Con eso en mente, se procede a caracterizar el algoritmo de descomposición en Matlab, que permite determinar L y U sin necesidad de alojar memoria para las matrices E_i $(i = 1,2,\dots,n)$ ni tampoco utilizar capacidad computacional para hallar sus inversas.

1. Se inicializan las matrices L y U. L como una matriz triangular inferior de unos, para que después del proceso de descomposición la diagonal tenga unos. U como una matriz de ceros con dimensión $\mathrm{n} \times \mathrm{n}$.
2. En cada iteración se fijan los valores de la diagonal de U, y para la primera iteración $U_{11} = A_{11}$ ya que la primera fila de A no se modifica durante el proceso de eliminación.

```
U(j,j) = v(j);
```

3. Para la iteración j-ésima se determinan los factores por los que debe multiplicarse la fila j para eliminar las filas desde la j+1 hasta la n y son esos factores los que se guardan como componentes de la matriz L, y son equivalentes a las componentes de las matrices E.

```
L(j+1:n,j) = v(j+1:n)/v(j);
```

4. Los componentes de U se determinan a partir de los coeficientes de eliminación gaussiana, asociados a la matriz L. Como se sabe, por la descripción que se hizo anteriormente, la descomposición Gaussiana puede representarse como un producto de matrices, de la forma $E_j A^{(j)} = A^{(j+1)}$. Si el producto se descompone por filas, conociendo las componentes de L, y recorriendo las columnas de A, se pueden determinar las componentes de la triangular superior U, resolviendo el sistema,

```
% El vector "a" hace referencia a la columna j-ésima de A
```

```
z = L(1:j-1,1:j-1)\a(1:j-1);
U(1:j-1,j) = z; % Se determinan las componentes superiores
% Se lleva a cabo el proceso de eliminación
v(j:n) = a(j:n) - L(j:n , 1:j-1)*z;
U(j,j) = v(j); % Se actualiza el valor de la diagonal de U
```

Descomposición de Cholesky

De manera similar al caso de la descomposición LU, el algoritmo de Cholesky es un caso particular de la eliminación Gaussiana, y puede pensarse como un caso de factorización LU en el que $U = L^T$. Sin embargo, para garantizar que la descomposición es única, es necesario que la matriz $A \in \mathrm{R}^{\mathrm{n}\times\mathrm{n}}$ sea positiva definida. El algoritmo que se lleva a cabo consiste en recorrer cada columna de la matriz A, y se hace una eliminación fila a fila, restando a cada columna el producto interno de filas y columnas de A,

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} u_{ik}u_{jk}}{u_{jj}}$$

```
A(j:n, j) = A(j:n,j) - A(j:n, 1:j-1)*A(j,1:j-1)';
```

Finalmente, se calculan los valores de la diagonal principal dividiendo cada cada componente por la raíz cuadrada de la diagonal j-ésima de A. Teniendo en cuenta que,

$$u_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} u_{ik}^2}$$

Con el siguiente comando se hacen dos pasos con una sola línea de código, se normalizan las componentes u_{ij} por los valores de la diagonal principal y se fijan los valores de la diagonal principal.

```
A(j:n,j) = A(j:n,j)/sqrt(A(j,j));
```

Como se trabaja sobre la matriz A y solo se modifican los valores de su diagonal y de su sección inferior, G es igual a la matriz triangular inferior que se deriva de A, i.e., la matriz u.

Prueba de desempeño

Para caracterizar el desempeño de los algoritmos, éstos se ejecutan para resolver sistemas de ecuaciones con distintas dimensiones y se mide el tiempo de ejecución en cada caso, como se muestra en l**a** Figura 1. Como es de esperarse, en ambos casos el tiempo de ejecución es mayor en la medida que el tamaño del sistema de ecuaciones lineales aumenta, al parecer con una tendencia lineal.

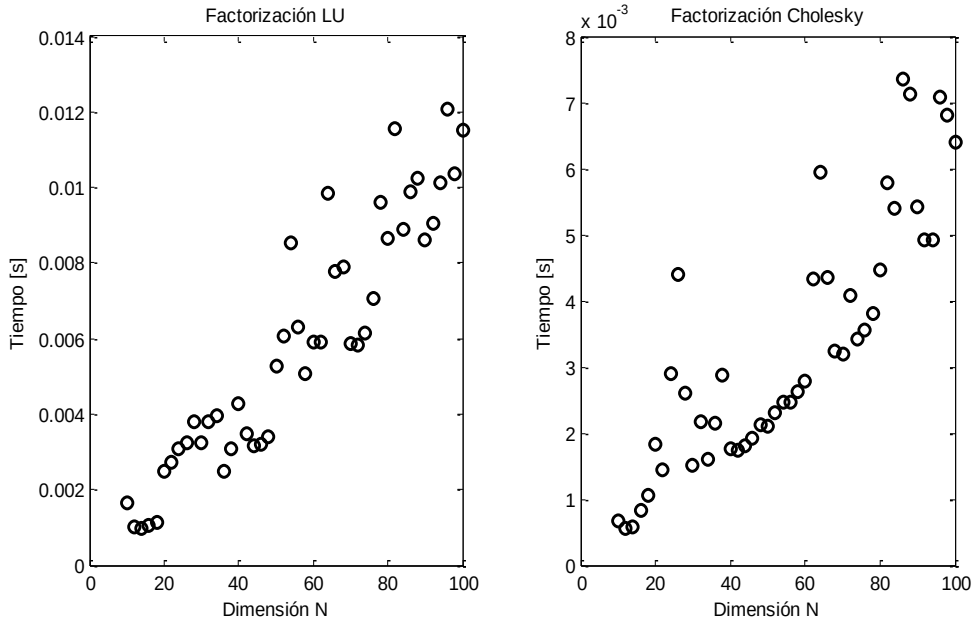

Figura 1. Tiempos de ejecución de algoritmos de descomposición, LU y Cholesky.

Sin embargo, si se estudia el número de veces que se interviene la matriz en cada caso se puede definir con mayor precisión la complejidad temporal de cada algoritmo. En el caso de la factorización LU, para un sistema de n ecuaciones, se requiere multiplicar una fila y luego restar el resultado a otra, lo que requiere un orden de n operaciones. Como hay n filas, para la eliminación de las componentes de la primera columna se requerirán n^2 operaciones, para la segunda, por ser más corta $(n-1)^2$ y así sucesivamente hasta la última columna. Por lo cual, el algoritmo de descomposición LU tiene complejidad cúbica, como se muestra a continuación,

$$n^2 + (n-1)^2 + \cdots + 2^2 + 1^2 = \sum i^2 = \frac{n(n+1)(2n+1)}{6} \rightarrow LU \in O\left(\frac{n(n+1)(2n+1)}{6}\right)$$

$$LU \in O(n^3)$$

Por otro lado, el algoritmo de Cholesky, requiere $(n-1)$ divisiones, $\frac{1}{3}n(n^2-1)$ multiplicaciones y $\frac{1}{3}n(n^2-1)$ restas. Por lo cual la complejidad del algoritmo de Cholesky también es $O(n^3)$. Sin embargo, revisando las expresiones en detalle $\frac{2n^3}{3}+\frac{n}{3}-1 < \frac{n^3}{3}+\frac{n^2}{2}+\frac{n}{6}$, y así la complejidad temporal de ambos algoritmos pueda describirse como $O(n^3)$, el algoritmo de Cholesky resulta más eficiente que el de descomposición LU. A pesar de que no es apropiado caracterizar algoritmos con mediciones de tiempo computacional, en la Figura 2 se hace una caracterización empírica y se muestra que el algoritmo de Cholesky es ligeramente más eficiente que el de descomposición LU, como se demostró anteriormente. Cabe anotar, que las variaciones en los tiempos computacionales se dan porque la capacidad de procesamiento que el computador le asigna a Matlab varía en el tiempo. Además, la caracterización para un intervalo $n \in [10,100]$ es

limitado y por eso, no es posible apreciar la tendencia cúbica del tiempo de ejecución de los algoritmos.

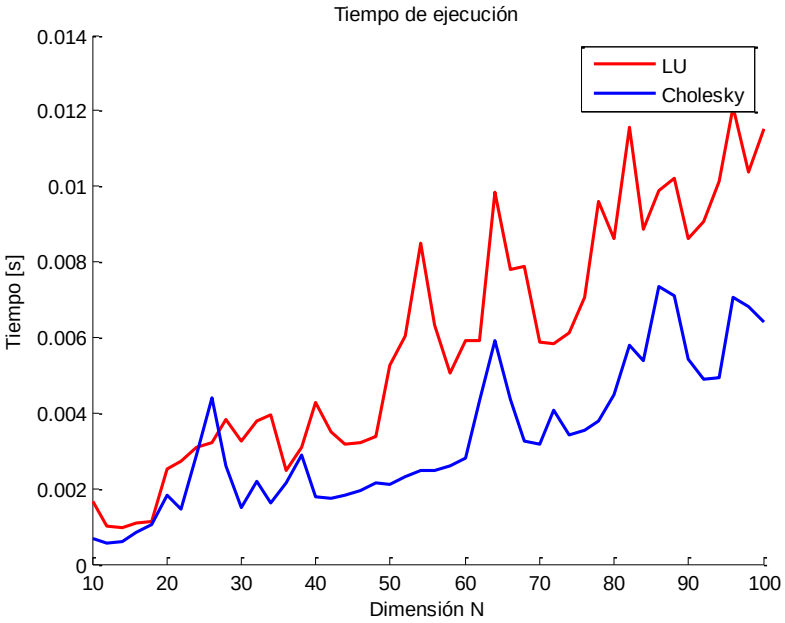

Figura 2. Comparación del tiempo de ejecución de los algoritmos.