Tarea 1. Descomposición LU y Cholesky

Ricardo Chávez Cáliz

6 de septiembre de 2017

Problema 1. Implementar los algoritmos de *Backward y Forward substitution.*

Se implementó el algoritmo de Backward substitution para la obtención del vector solución X del sistema $tS \cdot X = b$, donde tS denota una matriz no singular de $n \times n$ obtenida al tomar entradas aleatorias arriba de la diagonal con distribución U(0,1) y b es un vector aleatorio de tamaño n con entradas de distribución $U(0,1)$. De igual manera se implementó el algoritmo Forward substitution para resolver $tI \cdot X = b$, donde tI denota una matriz no singular obtenida al transponer una matriz tS como se describió antes. Para verificar que los algortimos funcionan correctamente se calcula $tI \cdot X - b$ y $tS \cdot X - b$, los cuales deben aproximarse a $0 \in \mathbb{R}^n$.

Problema 2. Implementar el algoritmo de eliminación gaussiana con pivoteo parcial LUP, 21.1 del Trefethen (p. 160).

Dicho algoritmo sin pivoteo parcial aparece definido en la función EGPP en el código que se adjunta. La matriz L se modificaba y no siempre quedaba triangular inferior. El algoritmo en EDPP se ejecuta en el mismo orden que cuando se implementa el pivoteo parcial y servirá para el análisis.

Problema 3. Dar la descomposición LUP para una matriz aleatoria de entradas $U(0,1)$ de tamaño 5 × 5, y para la matriz

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix}$$

Se Verifica el algoritmo LUP calculando $PLU - A$. En ambos casos obtenemos dos matrices que se aproximan a 0 (entradas de orden pequeño).

Problema 4. Usando la descomposición PLU anterior, resolver el sistema de la forma $Dx = b$ donde D son las matrices del problema 3, para 5 diferentes b aleatorios con entradas $U(0,1)$. Verificando si es o no posible resolver el sistema.

Si $D = PLU$ entonces podemos transformar el sistema de la siguiente manera

$$
\begin{aligned}
Dx &= b \\
PLUx &= b \\
LUx &= P^{-1}b
\end{aligned}
$$

Si $y = Ux$ entonces el $Ly = P^{-1}b$ donde $P^{-1}b$ es fácil de calcular dado que P es de permutación. Usando el algoritmo *Forward substitution* podemos calcular el valor de y dado que L es triangular inferior. Ahora podemos solucionar el sistema $Ux = y$ dado que U es triangular superior, y es conocido y se tiene el algoritmo *Backward substitution*. Se calcula para las matrices solicitadas y se comprueba calculando $D \cdot x - b$. En todos los casos obtenemos un vector en $\mathbb{R}^n$ de norma muy pequeña. Se verifica que el sistema tenga solución calculando $det(D)$ y permitiendo una tolerancia en valor absoluto de 0.001

Problema 5. Implementar el algoritmo de descomposición de Cholesky 23.1 del Trefethen (p. 175).

Es importante señalar que para obtener la forma de Cholesky correcta hay que hacer 0 todos los elementos por debajo de la diagonal. Para verificar que el algortimo funciona correctamente se obtuvo una matriz A de tamaño $n \times n$ con entradas de distribución $U(0,1)$. Para asegurarse de que la matriz fuera simétrica y positiva definida se tomó $B = A^t \cdot A + I$. Así denotando por L a la matriz obtenida de B en el algoritmo de Choleski, $L^t \cdot L - B$ debe aproximarse a $0 \in \mathbb{R}^{n \times n}$.

Problema 6. Comparar la complejidad de su implementación de los algoritmos de factorización de Cholesky y LUP mediante la medición de los tiempos que tardan con respecto a la descomposición de una matriz aleatoria hermitiana definida positiva. Gráficar la comparación.

Para Cholesky se calculó una complejidad computacional de $\frac{m^3}{3}$ mientras que descomposición PLU fue $\frac{2m^3}{3}$. Se calculó el tiempo de ejecución para 200 matrices hermitianas como se describe en el punto 5, para Cholesky y para PLU. Se observa que los tiempo de ejecución se comportan como lo pronosticado teóricamente.

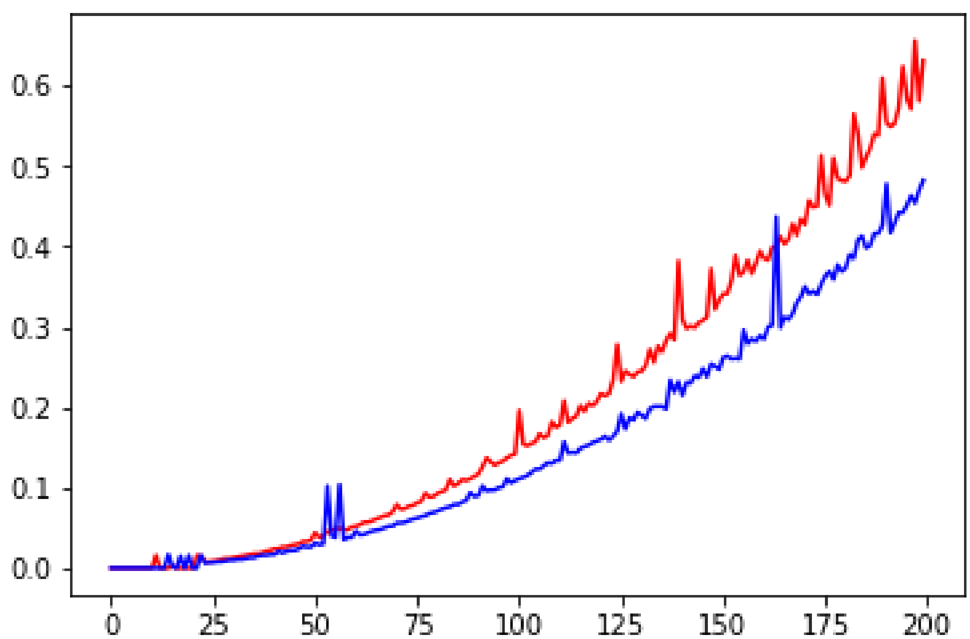

Figura 1: Tiempo de ejecución de Cholesky y descomposición PLU