Ejercicio de matrices - Análisis Numérico

Juan Sebastian Santamaria Palomino
juansantamaria@javeriana.edu.co

Natalia Andrea Navas Calderón
natalianavas@javeriana.edu.co

Jorge Rodrigo Salgado Tello
salgadojorge@javeriana.edu.co

13 de agosto de 2019

1. Problema

Dado el siguiente sistema:

$$\begin{aligned} 3.5x + 4.2y &= 1.3 \\ 0.5x + \tfrac{1}{3}y &= \tfrac{7}{4} \end{aligned} \tag{1}$$

Se desea averiguar si hay error en las entradas o coeficientes de la matriz sin necesidad de resolverla de nuevo. Además de ello, se requiere observar el comportamiento del algoritmo de Gauss cuando es aplicado en matrices de gran tamaño, por último, se desea encontrar la solución de la matriz a partir de métodos diferentes a reducción, tales como descomposición y factorización.

2. Análisis de de sensibilidad

Cuando una matriz presenta un error en las entradas o coeficientes, se dice que la matriz está perturbada, adicionalmente, si la matriz sufre un cambio considerable en el resultado al realizar pequeñas perturbaciones en las entradas, es posible decir que la matriz está mal condicionada. Para realizar el análisis de sensibilidad, es necesario usar el número de condición que se define como: $cond(A) = ||A||||A^{-1}||$. El inverso de este número permite conocer si la matriz está mal condicionada, si el resultado es muy cercano a 0 la matriz está mal condicionada, y si el resultado es muy cercano a 1 la matriz está bien condicionada. Para este ejercicio, cambiando la posición $A_{2,1}$ por un 4.5, se obtuvo como resultado 0.2878788, y dado que dicho valor es cercano a 0 es posible afirmar que la matriz está mal condicionada.

3. Aplicación de Gauss Jordan en matrices grandes

Una de las preguntas que surgió cuando se usó el algoritmo de Gauss para resolver sistemas de ecuaciones es: ¿ Qué pasa cuando el algoritmo se enfrenta a matrices grandes?. Para resolver esta duda se realizaron tres pruebas, cada una con matrices de diferentes tamaños. Para el primer caso se probó con una matriz de 10×10, con la cual el algoritmo hizo su trabajo de manera correcta. En el segundo caso se probó con una matriz de 100×100, con la cual el algoritmo también hizo su trabajo de manera correcta. En el último caso se probó con una matriz de 1000×1000, con la cual el algoritmo no terminó de ejecutarse y debía matarse para poder finalizar, por ello, es posible concluir que el algoritmo falla con matrices de un gran tamaño.

4. Descomposición de matrices

Para solucionar un sistema de ecuaciones por descomposición, se utilizó el método de LU. La ganancia obtenida con este método es que si tenemos la descomposición LU de A, entonces el sistema $AX = b$ lo podemos resolver para distintas matrices b sin tener que estar aplicando el método de eliminación gaussiana cada vez desde el principio. Para aplicar este método se requiere de la matriz L y U. Siendo L la triangular superior (con 1's en la diagonal) y U es triangular inferior, entonces el sistema $AX = b$ se puede resolver usando L y U de la siguiente manera.

- Resolvemos $LY = b$ y obtenemos la solución Y*.
- Resolvemos $UX = Y^*$ y obtenemos la solución X* del sistema.

X* es la solución del sistema original pues como $UX^*= Y^*$ entonces $LY^*= LUX^*= AX^*= b$. Dada la matriz de la forma $AX = b$

$$A = \left[\begin{array}{cc:c} 3.5 & 4.2 & 1.3 \\ 0.5 & \frac{1}{3} & \frac{7}{4} \end{array}\right] \tag{2}$$

Su descomposición se da mediante:

$$L = \begin{bmatrix} 1.0 & 0.0 \\ 0.5 & 1.0 \end{bmatrix} \tag{3}$$

$$U = \begin{bmatrix} 3.5 & 4.2 \\ 0.0 & \frac{1}{3} \end{bmatrix} \tag{4}$$

Al realizar la descomposición LU de A se obtiene como resultado:

$$\begin{bmatrix} 3.5 & 4.2 \\ 0.143 & -0.267 \end{bmatrix} \tag{5}$$

Finalmente, se resuelve el sistema $Ax = b$, dando como resultado:

$$[7.410714 \quad -5.866071] \tag{6}$$

5. Factorización de matrices

La factorización de matrices es particularmente muy útil para elevar las matrices a diferentes potencias. La factorización de una matriz $A = (P)(D)(P^{-1})$, dónde P son los vectores propios de A, y D es la diagonal de los vectores propios de A. Esto lleva a que $A^k = (P)(D^k)(P^{-1})$. La ventaja de este método es que elevar D a una potencia k es mucho menos costoso en número de operaciones y consumo de recursos de procesamiento que hacer esta misma operación en A, ya que D es una matriz diagonal.

La matriz A, definida por:

$$A = \begin{bmatrix} 3.5 & 4.2 \\ 0.5 & \frac{1}{3} \end{bmatrix} \tag{7}$$

Tiene la siguiente diagonal D de valores propios:

$$A = \begin{bmatrix} 4.06 & 0.0 \\ 0.0 & -0.230 \end{bmatrix} \tag{8}$$

Una matriz de vectores propios P:

$$A = \begin{bmatrix} 0.991 & -0.75 \\ 0.133 & 0.664 \end{bmatrix} \tag{9}$$

Conociendo lo anterior se calcula el inverso de P, (P^{-1}):

$$A = \begin{bmatrix} 0.877 & 0.987 \\ -0.17 & 1.31 \end{bmatrix} \tag{10}$$

Luego se realiza la multiplicación PDP^{-1}, dando como resultado la matriz A, tal como se observa a continuación:

$$A = \begin{bmatrix} 3.5 & 4.2 \\ 0.5 & \frac{1}{3} \end{bmatrix} \tag{11}$$

Con la anterior equivalencia se puede realizar el cálculo de potencias sin necesidad de multiplicar la matriz por si misma, para este ejercicio se realiza D^k, donde $k = 5$:

$$A = \begin{bmatrix} 9.26 \times 10^{38} & 0.0 \\ 0.0 & 1.31 \times 10^{-41} \end{bmatrix} \tag{12}$$

Por último se demuestra que $PD^kP^{-1} = A^k$ con el siguiente resultado:

$$A = \begin{bmatrix} 8.04 \times 10^{38} & 9.06 \times 10^{38} \\ 1.08 \times 10^{38} & 1.21 \times 10^{38} \end{bmatrix} \tag{13}$$