
TRABAJO DE INFORMÁTICA

Como has podido comprobar, muchos de los cálculos que haces con Máxima para resolver el problema anterior son “*cajas negras*” que aplicas cuando es necesario. Su implementación está hecha y utilizas los comandos correspondientes de Máxima. Por ejemplo, para hallar la inversa de una matriz, utilizas la instrucción de Máxima *inversa*, o para calcular la factorización LU utilizas funciones de paquetes especiales. En el trabajo de informática te vamos a pedir, entre otras cosas, que implementes los algoritmos para hallar la inversa y la solución a sistemas de ecuaciones lineales mediante la factorización LU. Pretendemos que “*destripes*” las cajas negras y seas tú quien programe esos algoritmos.

Implementa en Java la clase *Matriz* con **al menos** los siguientes constructores y métodos.

Constructores

- Constructor para crear una matriz $n \times m$ con todo unos en la diagonal y el resto de elementos igual a cero.
- Constructor de copia: construye una copia de una matriz dada.

Métodos

- Métodos para observar y modificar cualquier elemento (i,j) de la matriz.
- Métodos para hallar la columna (fila) donde se encuentre el máximo elemento en valor absoluto de una fila (columna) a partir de una columna (fila) dada. También métodos para hallar la columna (fila) donde se encuentre el máximo elemento en valor absoluto de una fila (columna) dada.
- Métodos para hallar la fila (columna) con el primer elemento no nulo de una columna (fila) a partir de una fila (columna) dada. Si todos los elementos son nulos, se devuelve -1.
- Métodos que modifiquen la matriz mediante las operaciones elementales de matrices por filas: intercambiar dos filas, cambiar la fila i -ésima por el valor de la fila i -ésima más la fila j -ésima multiplicada por un escalar, multiplicar una fila por un escalar.
- Métodos que modifiquen la matriz mediante las operaciones elementales de columnas.
- Métodos que devuelva una nueva matriz con la suma, resta y multiplicación de la matriz propietaria del método y otra matriz dada (no modifican la matriz).
- Método que retorna la inversa de la matriz propietaria del método mediante el algoritmo de Gauss-Jordan.
- Método que devuelve la matriz L de la descomposición LU (llamar al método *getL()*).
- Método que retorna la matriz U de la descomposición LU (llamar al método *getU()*).

- Método que dada una matriz $nx1$, representando términos independientes, nos devuelve otra matriz $nx1$ con la solución al sistema usando la descomposición LU.
- Método *toString()* para transformar la información de la matriz en un String.

Crea un método main para probar los métodos realizados en donde se resolverá el sistema de ecuaciones propuesto en el trabajo de matemáticas. Se deberá mostrar por pantalla la matriz del sistema de ecuaciones, las matrices L y U de la descomposición y la solución al sistema.

Observaciones importantes.

1. Los métodos que hallan la inversa de una matriz y las matrices L , U solo serán efectivos si la matriz es cuadrada. Además, en el caso de la descomposición LU, tampoco será efectivo si durante el algoritmo para hallar las matrices superior e inferior aparece algún elemento igual a cero en la diagonal. Si la matriz no cumpliera alguna de las condiciones anteriores, se retornará null. Ejemplo: para las matrices

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

se podrá calcular sus inversas, pero no la factorización LU ya que aparecen ceros en la diagonal durante el algoritmo. En el caso de la matriz A , al inicio del algoritmo (suponemos que siempre empezamos con el elemento (0,0)) y en el caso de la matriz B , en el segundo paso del algoritmo aparece un cero en (1,1). En estos casos, retornaríamos las matrices null cuando se invocase los métodos *getL()* y *getU()*. En otras palabras, para hallar la inversa se pueden hacer intercambios de filas, y para la factorización, no. Esta es la razón por la que podemos hallar la inversa de A (se intercambian las dos filas y se continúa) y no su factorización LU.

2. Aunque los métodos *getL()* y *getU()* solo nos devuelven la matriz triangular inferior y superior, respectivamente, de la descomposición LU, al calcular una de ellas, fácilmente obtenemos la otra. Esto es, si se ejecuta

Matriz l = m.getL();

Matriz u = m.getU();

parece razonable que aprovechemos los cálculos hechos al invocar *getL()* para no tener que volver a realizar las mismas operaciones cuando se invoca a *getU()*. Lo mismo ocurriría si llamásemos a los métodos en orden inverso, primero el método *getU()* y más tarde *getL()*. Además, notar que si no se llama a ningún método que modifique la matriz (métodos para intercambiar y multiplicar por reales, filas y columnas, etc), al volver a llamar a uno de los métodos que nos calculan la factorización LU, tampoco sería necesario recalcular las matrices L , U ya que no han cambiado los valores de la matriz.

3. Para poder subir la nota en el trabajo de matemáticas se tienen que tener correctamente implementados los métodos que hallan **la inversa, factorización LU y solución al sistema de ecuaciones.**

Material a presentar en el trabajo de informática.

Un fichero zip con el proyecto BlueJ incluida **la documentación** de todos los métodos implementados. El fichero se enviará mediante la aplicación de presentación de trabajos que aparece en el ADD en los plazos indicados.