Programación: Factorización LU

Objetivos. Programar el método de la factorización LU y la solución de sistemas de ecuaciones elementales usando la factorización LU.

Requisitos. Ciclos `for`, entradas de matrices y vectores, submatrices y subvectores, operaciones elementales por renglones, solución de sistemas de ecuaciones lineales con matrices triangulares.

1. Problema: Factorización LU. Escriba una función que construya la factorización LU de la matriz dada A.

> Entrada: matriz cuadrada A.
>
> Condición que debe cumplir la entrada: se supone que la matriz A es estrictamente invertible, esto es, a la matriz A se puede aplicar la eliminación de Gauss con pivotes diagonales. No tiene que verificar esta condición.
>
> Salida: dos matrices L y U tales que $LU = A$, L es triangular inferior con unos en la diagonal principal, U es triangular superior.

Puede usar el siguiente esbozo del algoritmo:

```
Entrada: matriz cuadrada A.
n := orden de A;
L := matriz identidad de orden n;
U := copia de A;
Para p := 1, ..., n - 1:
    Para i := p + 1, ..., n:
        mu := U[  ,  ] / U[  ,  ];
        L[  ,  ] := mu;
        U[i, p] := 0;
        Para j :=    , ...,    :
            U[  ,  ] -= mu * U[  ,  ];
Salida: L y U.
```

2. Usar operaciones vectoriales (opcional). En lenguajes de programación orientados a matrices se pueden usar operaciones vectoriales para eliminar el ciclo sobre j. Más aún, utilizando la operación de multiplicación de matrices por vectores se puede eliminar el ciclo sobre i.

3. Guardar L y U en una matriz (opcional). Las entradas no triviales de la matriz L se pueden guardar en la parte inferior de la matriz U.

4. Complejidad. Calcular el número de operaciones aritméticas en el algoritmo anterior. La respuesta es un polinomio de n.

5. Pruebas con matrices pequeñas. Componga una matriz triangular inferior L de orden 3 con elementos diagonales iguales a 1 y una matriz triangular superior U de orden 3 con elementos diagonales no nulos. Calcule su producto $A = LU$ y aplique la función del ejercicio anterior a la matriz A.

6. Problema SolveWithLU. Escriba una función que resuelva sistemas de ecuaciones lineales de la forma $Ax = b$ a través de la factorización LU. Use las funciones SolveUT y SolveLT (o SolveLT1) de las clases anteriores.

7. Pruebas con matrices pequeñas. Pruebe la función SolveWithLU con el siguiente sistema de ecuaciones:

$$\begin{bmatrix} 1 & 1 & 2 \\ 2 & -2 & 6 \\ -3 & 1 & -4 \end{bmatrix} \begin{bmatrix} -3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -4 \\ 7 \end{bmatrix}.$$

8. Pruebas con matrices grandes. Haga pruebas de los algoritmos anteriores generando la matriz A y el vector b de manera aleatoria. En estas pruebas se recomienda calcular la norma del vector $Ax - b$ y el tiempo de ejecución de programas. ¿Cómo se cambia el tiempo de ejecución al multiplicar n por 10?