UNIVERSIDAD
DE MURCIA

DEPARTAMENTO DE MATEMÁTICAS

En esta práctica completamos el estudio de los métodos directos de resolución de sistemas de ecuaciones lineales. Hemos añadido a `Matrices` los métodos `double[] solucionFactorQR(double[][] A, double[] b)`, donde se implementa el método de Householder de resolución de sistemas, y `double[][][] factLUDoolittle(double[][] A)`, que devuelve la factorización LU de una matriz cuadrada (nótese que el resultado se devuelve en una matriz $2 \times n \times n$, donde los coeficientes $[0][i][j]$ corresponden a los coeficientes ij de la matriz L, y los coeficientes $[1][i][j]$ a los coeficientes ij de la matriz U).

1. Construir un método

`double[] resolverSistemaLUDoolittle(double[][] A, double[] b)`

que devuelva la solución del sistema $Ax = b$ aplicando la factorización LU de Doolittle. Diseñar una aplicación para resolver, usando el método anterior, el sistema

$$\begin{cases} 4x_0 - x_1 - x_2 = 1 \\ -x_{k-1} + 4x_k - x_{k+1} - x_{k+2} = 0 \quad (k = 1, 2, \ldots, n-3) \\ -x_{n-3} + 4x_{n-2} - x_{n-1} = 0 \\ -x_{n-2} + 4x_{n-1} = 0 \end{cases}$$

para $n = 20$.

2. Construir un método

`double[][][] factLUCrout(double[][] A)`

que devuelva la factorización LU de Crout de la matriz A en una matriz tridimensional `LU` donde `LU[0][][]` se corresponderá con la matriz L de la descomposición y `LU[1][][]` con la matriz U.

Aplicarlo a la matriz

$$A = \begin{pmatrix} 6 & 2 & 1 & -1 \\ 2 & 4 & 1 & 0 \\ 1 & 1 & 4 & -1 \\ -1 & 0 & -1 & 3 \end{pmatrix}$$

y verificar la bondad de la descomposición multiplicando ambas matrices y restando el resultado a A.

3. Recuérdese que si A es una matriz tridiagonal entonces su factorización LU de Doolittle viene dada por ciertas matrices

$$L = \begin{pmatrix} 1 & & & & & \\ l_0 & 1 & & & & \\ & l_1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & l_{n-3} & 1 & \\ & & & & l_{n-2} & 1 \end{pmatrix}, \quad U = \begin{pmatrix} p_0 & u_0 & & & & \\ & p_1 & u_1 & & & \\ & & p_2 & u_2 & & \\ & & & \ddots & \ddots & \\ & & & & p_{n-2} & u_{n-2} \\ & & & & & p_{n-1} \end{pmatrix}.$$

Construir un método

`double[][] factTridiagonal(double[][] F)`

que reciba como dato una matriz tridiagonal y devuelva su factorización LU de Doolittle. Se introducirán al método los datos a través de una matriz F formada por tres filas de longitudes $n-1$, n y $n-1$ (donde n es la dimensión de la matriz de partida), que corresponden respectivamente a la diagonales inferior, principal y superior de la matriz. Los resultados se devolverán en una matriz de similares características donde las tres filas serán, respectivamente, la diagonal inferior de L, la diagonal principal de U y la diagonal superior de U.

Construir un método

```
double[] resolverSistemaTridiagonal(double[][] F, double[] b)
```

que reciba como datos una matriz tridiagonal A (a través de una matriz F con el formato anterior) y un vector b y devuelva la solución x del sistema $Ax = b$. Usar el método para resolver el sistema

$$\begin{aligned} 3x - 8y &= 2 \\ 2x + 6z &= -3 \\ 7y + z - 9u &= 0 \\ 8z - 5u &= 1 \end{aligned}$$

4. La matriz de Hilbert H_n es una matriz $n \times n$ cuyos coeficientes vienen dados por la fórmula

$$(H_n)_{ij} = \frac{1}{i+j+1}, \quad 0 \leq i, j < n.$$

Resolver el sistema $H_n x = u$ para $n = 2, 3, \ldots, 12$, donde u es el vector n-dimensional dado por $u_i = 1$, $0 \leq i < n$, utilizando los métodos `solucionFactorQR` y `gaussParcial`. Comparar las soluciones obtenidas con la que se puede calcular directamente usando la matriz inversa de H_n, que puede probarse viene dada por la fórmula

$$(H_n^{-1})_{ij} = (-1)^{i+j}(i+j+1)\binom{n+i}{n-1-j}\binom{n+j}{n-1-i}\binom{i+j}{i}^2.$$