UNIVERSIDAD
DE MURCIA

DEPARTAMENTO DE MATEMÁTICAS

1. Construir métodos

```
iterJacobi(double[][] A, double[] b, double tol, int nmax, double[] x0, boolean imp)
```

y

```
iterGaussSeidel(double[][] A, double[] b, double tol, int nmax, double[] x0, boolean imp)
```

que devuelvan (en un `double[]`) la solución del sistema $Ax = b$ aplicando, respectivamente, los métodos iterativos de Jacobi y de Gauss-Seidel. Aquí `tol` es el error permitido, `nmax` el número máximo de iteraciones permitidas, `x0` el vector de partida y `imp` una variable booleana de manera que cuando tome el valor `true` se imprima también la sucesión de vectores que el método se genera.

2. Estudiar numéricamente la convergencia de los métodos de Jacobi y Gauss-Seidel para las matrices

$$A_1 = \begin{pmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{pmatrix} \qquad A_2 = \begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}$$

3. Considérese el sistema lineal

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 19 \\ 19 \\ -3 \\ -12 \end{pmatrix}$$

(a) Calcular la solución utilizando el método de Gauss con pivote total.

(b) Tomando $(0, 0, 0, 0)$ como vector de partida, encontrar las diez primeras iteradas que resultan de aplicar los métodos de Jacobi y Gauss-Seidel.

4. Constrúyase una variante del método `iterJacobi`, que llamaremos

```
iterJacobiOperaciones,
```

que en lugar de devolver la solución aproximada de un sistema $Ax = b$ devuelva un vector `operaciones` de tres coordenadas enteras (respectivamente, el número de sumas y restas, de multiplicaciones y de divisiones que el método realiza para obtener la solución aproximada). Usarlo para averiguar el número de operaciones que el método de Jacobi necesita para calcular la solución del sistema

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 19 \\ 19 \\ -3 \\ -12 \end{pmatrix}$$

tomando $(0, 0, 0, 0)$ como vector de partida y usando una tolerancia de 10^{-7}.

5. Sea A una matriz regular $n \times n$ y sea X_0 una matriz $n \times n$ arbitraria. Se sabe que, bajo ciertas condiciones para X_0 y A, la sucesión dada por $X_{k+1} = X_k + X_k(\mathrm{Id} - AX_k)$ (con Id la matriz identidad) converge a la inversa de A. Construir un método `iterInversa` que reciba una matriz `A`, una matriz condición inicial `X0`, una tolerancia `tol` y un número máximo de iteraciones `nmax` y devuelva la matriz inversa aproximada `InvAprox` o un mensaje de error si no se ha obtenido la matriz deseada tras `nmax` iteraciones. El criterio que se utilizará para saber si `InvAprox` es la aproximación a A^{-1} deseada es que $\|\mathtt{Id} - \mathtt{A} \cdot \mathtt{InvAprox}\|$ sea menor que `tol` (la norma $\|\cdot\|$ es la que se prefiera). Verificar la eficacia del método con las matrices

$$A = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 1 \\ 0 & 3 & -1 \end{pmatrix}, \quad X_0 = \begin{pmatrix} -3 & 4 & -2 \\ 2 & -1 & 0 \\ 1{,}7 & -1{,}6 & 0{,}6 \end{pmatrix}.$$