Métodos de Jacobi y Gauss-Seidel

Resuelve el sistema Ax = b usando \

```
A = [10, -1, 2, 0 ; -1, 11, -1, 3 ; 2, -1, 10, -1 ; 0, 3, -1, 8]
```

```
A = 4×4
    10    -1     2     0
    -1    11    -1     3
     2    -1    10    -1
     0     3    -1     8
```

```
b = [6 ; 25 ; -11 ; 15]
```

```
b = 4×1
     6
    25
   -11
    15
```

```
x = A \ b
```

```
x = 4×1
    1.0000
    2.0000
   -1.0000
    1.0000
```

Verifica que A es estrictamente diagonal dominante

```
esDiagonalDominante(A)
```

```
ans = logical
   1
```

Verifica que A no es singular

```
det(A)
```

```
ans = 7395
```

```
cond(A)
```

```
ans = 2.3597
```

Resuelve el sistema Ax = b usando el método de Jacobi

```
[x, i] = jacobi1(A, b)
```

```
x = 4×1
    1.0000
    2.0000
   -1.0000
    1.0000
i = 43
```

```
[x, i] = jacobi2(A, b)
```

```
x = 4×1
    1.0000
    2.0000
   -1.0000
    1.0000
i = 44
```

Resuelve el sistema Ax = b usando el método de Gauss-Seidel

```
[x, i] = gaussSeidel1(A, b)
```

```
x = 4×1
    1.0000
    2.0000
   -1.0000
    1.0000
i = 17
```

```
[x, i] = gaussSeidel2(A, b)
```

```
x = 4×1
     1
     2
    -1
     1
i = 17
```

Strict diagonal dominance of A implies that the spectral radius of T is less than one for Jacobi and Gauss-Seidel methods.

```
D = diag(diag(A));
L = tril(A, -1);
U = triu(A, 1);
```

Muestra que el radio espectral de T para Jacobi es menor que uno

```
max(abs(eig(-D \ (L + U))))
```

```
ans = 0.4264
```

Muestra que el radio espectral de T para Gauss-Seidel es menor que uno

```
max(abs(eig(-(D + L) \ U)))
```

```
ans = 0.0898
```

Escribe aquí los métodos de Jacobi y Gauss-Seidel

```
function esDiag = esDiagonalDominante(A)
    [n, m] = size(A);
    if n ~= m, error("No es cuadrada"), end
    esDiag = all(2 * abs(diag(A)) > sum(abs(A), 2));
```

```
end

function [x, i] = jacobi1(A, b)
    if ~esDiagonalDominante(A)
        warning("La matriz no es diagonal dominante estricta.")
    end

    MAX_ITER = 50;
    TOLER = eps;
    D = diag(diag(A));
    L = tril(A, -1);
    U = triu(A, 1);
    T = -D \ (L + U);
    c = D \ b;
    x = zeros(size(b));

    i = 0;
    flag = true;
    while flag
       xp = x;
       x = T * x + c;
       i = i + 1;
       flag = norm((x - xp) / x, inf) > TOLER && i < MAX_ITER;
    end
end

function [x, i] = jacobi2(A, b)
    if ~esDiagonalDominante(A)
        warning("La matriz no es diagonal dominante estricta.")
    end

    MAX_ITER = 50;
    TOLER = eps;
    D = diag(diag(A));
    x = zeros(size(b));

    i = 0;
    flag = true;
    while flag
       xp = x;
       x = x + D \ (b - A * x);
       i = i + 1;
       flag = norm((x - xp) / x, inf) > TOLER && i < MAX_ITER;
    end
end

function [x, i] = gaussSeidel1(A, b)
    if ~esDiagonalDominante(A)
        warning("La matriz no es diagonal dominante estricta.")
    end

    MAX_ITER = 50;
    TOLER = eps;
    D = diag(diag(A));
```

```
    L = tril(A, -1);
    U = triu(A, 1);
    T = -(D + L) \ U;
    c = (D + L) \ b;
    x = zeros(size(b));

    i = 0;
    flag = true;
    while flag
       xp = x;
       x = T * x + c;
       i = i + 1;
       flag = norm((x - xp) / x, inf) > TOLER && i < MAX_ITER;
    end
end

function [x, i] = gaussSeidel2(A, b)
    if ~esDiagonalDominante(A)
        warning("La matriz no es diagonal dominante estricta.")
    end

    MAX_ITER = 50;
    TOLER = eps;
    D = diag(diag(A));
    L = tril(A, -1);
    x = zeros(size(b));

    i = 0;
    flag = true;
    while flag
       xp = x;
       x = x + (D + L) \ (b - A * x);
       i = i + 1;
       flag = norm((x - xp) / x, inf) > TOLER && i < MAX_ITER;
    end
end
```