	INGENIERIA INFORMATICA *13042 Laboratorio de Matemáticas para la computación* *Curso 2004-2005*

Práctica 4: Métodos de resolución de sistemas de ecuaciones lineales

Objetivos: Aplicar dos técnicas de resolución de sistemas de ecuaciones lineales:
- Un método finito basado en la descomposición *LU* de la matriz de coeficientes *A*.
- El método iterativo de Gauss-Seidel.

1. Suponga un sistema de ecuaciones lineales de fórmula general Ax=b. Si la matriz de coeficientes A se puede factorizar mediante el producto de una matriz triangular inferior , con una matriz triangular superior *U* (A=LU), podemos resolver el sistema de ecuaciones mediante dos etapas:

   a) Obtener z a partir de la ecuación Lz = b (en donde L y b son conocidos).

   b) Obtener x a partir de la ecuación Ux = z (en donde U y z son ya conocidos).

   MATLAB posee el operador **lu** para calcular la factorización *LU*:

   - si se invoca `[L,U,P]=lu(A)`, se obtiene la factorización *LU* de *A* con pivote: `P` es la matriz de permutación, `L` es la triangular inferior y `U` es la triangular superior: *PA=LU*.
   - si se invoca `[L,U]=lu(A)`, se obtiene la factorización *LU* de *A*, donde *L* aparece con filas permutadas, `L`=$P^T L$: $A=(P^T L)U$.

   La instrucción x=A\b es equivalente a [L,U]=lu(A), x=U\(L\b) o también a [L,U,P]=lu(A), x=U\(L\(Pb)). Estas últimas formas pueden ser más convenientes si se quiere resolver varios sistemas con una misma matriz *A*.
   El sistema de ecuaciones a) es fácil de resolver mediante el procedimiento de sustitución progresiva:

   $$z_i = (b_i - \sum_{j=1}^{i-1} l_{ij} z_j) / l_{ii} \qquad \text{(i=1,2,...,n)}$$

   Y el sistema b) es fácil de resolver mediante en procedimiento de sustitución regresiva:

   $$x_i = (z_i - \sum_{j=i+1}^{n} u_{ij} x_j) / u_{ii} \qquad \text{(i=n,n-1,...,1)}$$

   Implemente el método descrito sobre MatLab mediante una macro que realice lo siguiente:

   1. Construya la matriz de Toeplitz usando la instrucción: `A=toeplitz(1:5)`
   2. Calcule la factorización *LU* de *A* con pivote, empleando la función intrínseca de MatLab `[l,u,p]=lu(A).`
   3. Usando el resultado del apartado anterior, resuelva el sistema `Ax=b` para *b* igual a cada uno de los siguientes vectores mediante los dos pasos siguientes:

   $$\begin{pmatrix} 0 \\ 2 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 10^{-3} \\ -0.33 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 9 \\ 4 \\ 2 \\ 0 \\ -1 \end{pmatrix}$$

   4. Calcule el vector solución z empleando el algoritmo de sustitución progresiva mediante una la función `z = suspro(l,b)` que ha programarse, la cual admite como parámetros la matriz triangular inferior y el vector de términos independientes.
   5. Calcule el vector solución x empleando el algoritmo de sustitución regresiva mediante una función `x = susreg(u,z)`, similar a la descrita en el punto anterior y que admita como parámetros la matriz triangular superior y el vector de términos independientes.

2. El método de Jacobi resuelve un sistema de ecuaciones lineales Ax=b generando una serie de vectores $x^{(k)}$ empleando la siguiente fórmula de recurrencia:

$$x_i^{(k)} = (b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{(k-1)}) / a_{ii}$$

Implemente el método de Jacobi mediante la función MatLab:

```
[x, delta, m] = jacobi(A, b, Delta, M)
```

que admita como parámetros la matriz de coeficientes (A), el vector de términos independientes (b), el criterio de convergencia (Delta) y el número máximo de iteraciones (M) y que devuelva el vector de soluciones (x), la convergencia alcanzada (delta) y el número de iteraciones realizadas (m).
Aplique esta función a la resolución del siguiente sistema de ecuaciones:

$$A = \begin{pmatrix} 62 & 24 & 1 & 8 & 15 \\ 23 & 50 & 7 & 14 & 16 \\ 4 & 6 & 58 & 20 & 22 \\ 10 & 12 & 19 & 66 & 3 \\ 11 & 18 & 25 & 2 & 54 \end{pmatrix}, b = \begin{pmatrix} 110 \\ 110 \\ 110 \\ 110 \\ 110 \end{pmatrix}$$