Programación: Sistemas unitriangulares inferiores

Objetivos. Programar en el lenguaje de MATLAB el método de la *sustitución hacia adelante* para resolver sistemas de ecuaciones lineales con matrices unitriangulares inferiores. Analizar la complejidad computacional del problema.

Requisitos. Matrices triangulares, notación breve para las sumas, experiencia de programación en el lenguaje de MATLAB: funciones, ciclos, vectores, matrices, notación : (dos puntos), subvectores, submatrices.

1. Definición (matrices unitriangulares inferiores). Una matriz cuadrada $L \in \mathcal{M}_n(\mathbb{R})$ se llama *unitriangular inferior* si es triangular inferior y todas sus entradas diagonales son 1. Un sistema de ecuaciones lineales de la forma $Lx = b$ se llama *unitriangular inferior* si la matriz L es unitriangular inferior.

2. Ejemplo. Resolver el siguiente sistema unitriangular inferior de ecuaciones lineales:

$$\left\{\begin{array}{rcrcrcrcr} x_1 & & & & & & & = & -2; \\ 3x_1 & + & x_2 & & & & & = & -5; \\ -2x_1 & + & 4x_2 & + & x_3 & & & = & 5; \\ -3x_1 & + & x_2 & & & + & x_4 & = & 3. \end{array}\right.$$

Solución. La primera ecuación nos da el valor de x_1. De la segunda ecuación despejamos x_2 y sustituimos el valor de x_1, etc.:

$$x_1 = \underbrace{\qquad}_{?};$$

$$x_2 = -5 - 3x_1 = -5 - (-6) = \underbrace{\qquad}_{?};$$

$$x_3 = 5 - \bigl(-2x_1 + 4x_2\bigr) = 5 - \bigl(\qquad\qquad\qquad\bigr) = \underbrace{\qquad}_{?};$$

$$x_4 = \underbrace{\qquad}_{?} - \bigl(\underbrace{\qquad\qquad\qquad}_{?} + 0x_3\bigr) = \underbrace{\qquad}_{?} - \underbrace{\qquad}_{?} = \underbrace{\qquad}_{?}.$$

Comprobación:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ -2 & 4 & 1 & 0 \\ -3 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -2 \\ \underline{\qquad} \\ \underline{\qquad} \\ \underline{\qquad} \end{bmatrix} = \begin{bmatrix} -2+0+0+0 \\ \underline{\qquad\qquad} \\ \underline{\qquad\qquad} \\ \underline{\qquad\qquad} \end{bmatrix} = \begin{bmatrix} -2 \\ \underline{\qquad} \\ \underline{\qquad} \\ \underline{\qquad} \end{bmatrix}. \qquad \checkmark \quad \square$$

Notación breve para sumas

3. Ejemplos.

$$\sum_{k=4}^{6} b_k = \underset{\text{con } k=4}{b_k} + \underset{\text{con } k=5}{b_k} + \underset{\text{con } k=6}{b_k} = b_4 + b_5 + b_6.$$

$$\sum_{r=2}^{5} a_r = a_2 + a_3 + a_4 + a_5, \qquad \sum_{j=3}^{6} c_j = \underbrace{\qquad\qquad\qquad}_{?},$$

$$\sum_{k=2}^{3} c_k d_k = c_2 d_2 + c_3 d_3, \qquad \sum_{p=1}^{4} b_p y_p = \underbrace{\qquad\qquad\qquad\qquad}_{?}.$$

4. Sumas que dependen de parámetros.
La primera suma depende de k (k aparece en la respuesta); la segunda depende de i:

$$\sum_{j=1}^{4} A_{k,j} = A_{k,1} + A_{k,2} + A_{k,3} + A_{k,4}; \qquad \sum_{j=2}^{3} A_{i,j} x_j = \underbrace{\qquad\qquad}_{?}.$$

5. Ejemplos: escribir con $\sum$.

$$A_{5,1} y_1 + A_{5,2} y_2 = \sum_{j=1}^{2} A_{5,j} y_j. \qquad b_2 + b_3 + b_4 + b_5 + b_6 = \sum_{j=} \underbrace{\qquad}_{?}.$$

$$c_3 a_3 + c_4 a_4 + c_5 a_5 = \sum \underbrace{\qquad\quad}_{?}. \qquad a_2 b_3 + a_3 b_4 + a_4 b_5 = \sum \underbrace{\qquad\quad}_{?}.$$

6. Convenio: la suma del conjunto vacío de sumandos es cero.

Ejemplo: $$\sum_{j=3}^{2} a_j = \sum_{\substack{j\in\mathbb{Z}:\\ 3\le j\le 2}} a_j = \sum_{j\in\emptyset} a_j = 0.$$

Fórmulas de la sustitución hacia adelante en el caso de un sistema unitriangular inferior (se recomienda deducir las fórmulas antes de la clase práctica)

7. Fórmulas para $n = 4$. Consideremos un sistema de ecuaciones de la forma $Lx = b$, donde L es una matriz real unitriangular inferior de orden 4.

$$\left\{\begin{array}{ccccccccc} x_1 & & & & & & & = & b_1; \\ L_{2,1}x_1 & + & x_2 & & & & & = & b_2; \\ L_{3,1}x_1 & + & L_{3,2}x_2 & + & x_3 & & & = & b_3; \\ L_{4,1}x_1 & + & L_{4,2}x_2 & + & L_{4,3}x_3 & + & x_4 & = & b_4. \end{array}\right.$$

La primera ecuación nos da el valor de la incógnita x_1. De la segunda ecuación despejamos x_2 (expresamos x_2 a través de x_1). De la tercera ecuación despejamos la incógnita x_3 (la expresamos a través de x_1 y x_2). De la cuarta ecuación despejamos x_4.

$$x_1 =$$

$$x_2 =$$

$$x_3 = \underbrace{\qquad}_{?} - \Big(\underbrace{\qquad\qquad}_{?} + \underbrace{\qquad\qquad}_{?}\Big) = \qquad - \sum \qquad\qquad .$$

$$x_4 =$$

8. Fórmulas de la sustitución hacia adelante en el caso de un sistema unitriangular inferior. Generalice las fórmulas del ejercicio anterior. Sea A una matriz unitriangular inferior $n \times n$ y sea $b \in \mathbb{R}^n$. Escriba una fórmula para calcular x_i:

$$x_i = \qquad\qquad\qquad\qquad . \tag{1}$$

9. Suma sobre el conjunto vacío. Por definición, cualquier suma sobre un conjunto vacío es cero. Por ejemplo,

$$\sum_{i=4}^{3} a_i = \sum_{i\in\mathbb{Z}\colon\, 4\le i\le 3} a_i = \sum_{i\in\emptyset} a_i = 0.$$

Escriba la fórmula (1) para $i = 1$ y determine si esta es correcta.

Programación en el lenguaje de MATLAB
de la sustitución hacia adelante
en el caso de sistemas unitriangulares inferiores

10. Comandos para sacar una parte del renglón o columna.

```
A = [11 12 13 14; 21 22 23 24; 31 32 33 34; 41 42 43 44]

A(3, 1:3)

w = [71; 72; 73; 74]

w(1:3)
```

$$A = \begin{bmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{bmatrix}, \qquad w = \begin{bmatrix} 71 \\ 72 \\ 73 \\ 74 \end{bmatrix}.$$

11. Comando para calcular el producto interno. Sean $u \in \mathcal{M}_{1\times n}(\mathbb{R})$, $v \in \mathcal{M}_{n\times 1}(\mathbb{R})$. Entonces la suma

$$\sum_{j=1}^{n} u_j v_j = u_1 v_1 + \cdots + u_n v_n$$

es simplemente el producto matricial uv.

```
u = [3 4 5]

v = [6 7 8]

u * v
```

12. Utilizando la notación del Ejercicio 10 exprese la siguiente suma como un producto interno de ciertas submatrices de A y v:

$$\sum_{k=1}^{3} A_{3,k} w_k.$$

Escriba un comando en el lenguaje de MATLAB que calcule esta suma.

13. Problema SolveUniLT. Escriba una función SolveLT que resuelva el sistema de ecuaciones lineales $Lx = b$.

> Entrada: una matriz L y un vector b. Se supone que L es cuadrada y **unitriangular inferior**, y que la longitud de b coincide con el orden de L.
>
> Salida: un vector x tal que $Lx = b$.

14. Comprobación 1. Compruebe la función `SolveUniLT` usando los siguientes datos:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 5 & 1 & 0 \\ -4 & -2 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}.$$

```
L = ...

b = ...

x = SolveUniLT[L, b]
```

15. Comprobación 2. Haga otra comprobación con su propio ejemplo, para $n = 4$.

Análisis de la complejidad computacional

16. Fórmula para la suma de una progresión aritmética (repaso). Recuerde la fórmula para la siguiente suma:

$$\sum_{p=1}^{q} p = \underbrace{\qquad\qquad\qquad}_{?}.$$

17. Primera solución (formal). Calcule el número de las operaciones de multiplicación que se necesitan para calcular la expresión

$$\sum_{k=1}^{j-1} L_{j,k} x_k.$$

Calcule el número de las operaciones de multiplicación que se necesitan realizar el algoritmo programado en las páginas anteriores.

18. Número de las entradas de cada tipo en una matriz cuadrada.
Consideremos una matriz cuadrada:

$$\begin{matrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} \\ A_{4,1} & A_{4,2} & A_{4,3} & A_{4,4} \end{matrix}$$

Calcule el número de las entradas (para una matriz cuadrada $n \times n$):

- el número total de las entradas;
- en la diagonal principal;
- fuera de la diagonal principal;
- por debajo de la diagonal principal.

19. Segunda solución (conceptual). Analice la correspondencia entre las entradas de la matriz L (ubicadas por abajo de la diagonal principal) y las operaciones de multiplicación que se usan en algoritmo que hemos programado.

20. La complejidad del algoritmo es la óptima. Explique por qué el problema no se pueden resolver aplicando menos operaciones que $\dfrac{n(n-1)}{2}$.