

Práctica 1

Resolución de sistemas de ecuaciones lineales: métodos directos

Índice

1. Operaciones elementales por filas	1
2. Forma escalonada reducida de una matriz	3
3. Resolución de sistemas lineales con el operador \	3
3.1. Descripción del operador \	3
3.2. Solución general de un sistema utilizando \ y kernel	6
4. Redes de flujo	9

1. Operaciones elementales por filas

Sabemos que hay tres tipos de operaciones elementales por filas:

1. Intercambio de filas: una fila de la matriz se puede intercambiar con otra.
2. Multiplicación de una fila: cada elemento de una fila puede ser multiplicado por una constante no nula.
3. Suma de una fila: una fila puede ser sustituida por la suma de esta fila y un múltiplo de otra fila.

Los comandos de Scilab que se pueden utilizar para efectuar estas operaciones elementales en una matriz A son los siguientes:

1. Intercambio de filas aplicado a las filas i y j :

$$A([i, j], :) = A([j, i], :)$$

2. La fila i -ésima se multiplica por p :

$$A(i, :) = p * A(i, :)$$

3. La fila i se cambia por la suma de la fila i y p veces la fila j :

$$A(i, :) = A(i, :) + p * A(j, :)$$

Ejemplo 1. Consideramos la siguiente matriz

$$A = \begin{bmatrix} 0 & -2 & 3 & 9 \\ -4 & 6 & 0 & -4 \\ 2 & -5 & 5 & 17 \end{bmatrix}$$

Efectuaremos, utilizando Scilab, las siguientes operaciones elementales:

1. intercambiamos las filas 1 y 3,
2. multiplicamos por 1/2 la segunda fila,
3. sumamos la primera fila a la segunda.

La secuencia de comandos utilizados y las salidas obtenidas son:

```
-->A=[0 -2 3 9;-4 6 0 -4;2 -5 5 17]
```

```
A =
```

```
0. - 2. 3. 9.
- 4. 6. 0. - 4.
2. - 5. 5. 17.
```

```
-->B=A;
```

```
-->B([1,3],:)=B([3,1],:)
```

```
B =
```

```
2. - 5. 5. 17.
- 4. 6. 0. - 4.
0. - 2. 3. 9.
```

```
-->C=B;
```

```
-->C(2,:)=(1/2)*C(2,:)
```

```
C =
```

```
2. - 5. 5. 17.
```

$$\begin{array}{rrrr} -2. & 3. & 0. & -2. \\ 0. & -2. & 3. & 9. \end{array}$$

```
-->D(2,:)=D(2,:)+D(1,:)
D =
```

$$\begin{array}{rrrr} 2. & -5. & 5. & 17. \\ 0. & -2. & 5. & 15. \\ 0. & -2. & 3. & 9. \end{array}$$

2. Forma escalonada reducida de una matriz

La función **rref** proporciona la forma escalonada reducida de una matriz cualquiera. Veamos un ejemplo.

Ejemplo 2. Consideramos el sistema de ecuaciones lineales

$$\left. \begin{array}{rrcr} -2y & +3z & = & 9 \\ -4x & +6y & & = -4 \\ 2x & -5y & +5z & = 17 \end{array} \right\}$$

cuya matriz ampliada es la del ejemplo 1. Calculamos ahora la forma escalonada reducida de esta matriz:

```
-->A=[0 -2 3 9;-4 6 0 -4;2 -5 5 17];
```

```
-->rref(A)
ans =
```

$$\begin{array}{rrrr} 1. & 0. & 0. & 1. \\ 0. & 1. & 0. & 0. \\ 0. & 0. & 1. & 3. \end{array}$$

Por tanto la solución única del sistema es:

$$x = 1, \quad y = 0, \quad z = 3.$$

3. Resolución de sistemas lineales con el operador \

3.1. Descripción del operador \

Un procedimiento comúnmente utilizado para resolver un sistema de ecuaciones lineales $A\vec{x} = \vec{b}$ con Scilab consiste en introducir la matriz de coeficientes A y el vector de términos independientes \vec{b} y escribir $A \backslash b$. Este operador funciona de la siguiente manera:

1. Si el sistema es compatible determinado entonces $A \setminus b$ proporciona la solución única del sistema.
2. Si el sistema es compatible indeterminado entonces $A \setminus b$ proporciona una de las soluciones (elige una con, a lo sumo, r componentes no nulas, donde r es el rango de A).
3. Si el sistema es incompatible Scilab calcula un vector, llamado **aproximación por mínimos cuadrados** de la solución, es decir, un vector \vec{x}' tal que el valor de la norma $\|A\vec{x}' - \vec{b}\|$ es el mínimo posible de entre todos los posibles valores de \vec{x}' (notamos que este vector no es necesariamente único); Scilab elige uno con, a lo sumo, r componentes no nulas, donde r es el rango de A .

Notamos que, si el sistema es compatible, Scilab proporciona una de las soluciones. Si el sistema no es compatible, Scilab nos presenta un vector que **no es una solución**. Esto quiere decir que debemos ser muy cuidadosos con el operador \setminus .

Ejemplo 3. Consideramos el siguiente sistema de ecuaciones lineales:

$$\left. \begin{array}{rrc} -2y & +3z & = 9 \\ -4x & +6y & = -4 \\ 2x & -5y & +5z = 17 \end{array} \right\}$$

Utilizaremos el operador \setminus :

```
-->A=[0 -2 3; -4 6 0; 2 -5 5]; b=[9; -4; 17];
```

```
-->x=A\b
```

```
x =
```

```
1.
```

```
0.
```

```
3.
```

Ésta es la única solución del sistema lineal porque A es una matriz cuadrada invertible (tiene rango máximo):

```
-->rank(A)
```

```
ans =
```

```
3.
```

Ejemplo 4. Consideramos el siguiente sistema de ecuaciones lineales:

$$\left. \begin{array}{rrc} x & +y & +z = 1 \\ x & +y & +z = 2 \\ 2x & +2y & +2z = 3 \end{array} \right\}$$

que es, evidentemente, incompatible

```

-->A=[1 1 1; 1 1 1; 2 2 2]; b=[1; 2; 3];

-->x=A\b
warning :
matrix is close to singular or badly scaled. rcond =    0.0000D+00

x  =

    1.5
    0.
    0.

```

Scilab da un resultado que no es una solución del sistema.

Ejemplo 5. Consideramos el sistema

$$\left. \begin{array}{rrcr} x & +y & +z & = 1 \\ x & +y & +z & = 1 \\ 2x & +2y & +2z & = 2 \end{array} \right\}$$

que tiene, evidentemente, una cantidad infinita de soluciones.

```

-->A=[1 1 1; 1 1 1; 2 2 2]; b=[1; 1; 2];

-->x=A\b
warning :
matrix close to singular or badly scaled. rcond =    0.0000D+00

x  =

    1.
    0.
    0.

```

En este caso hemos obtenido una solución particular del sistema. De hecho,

```

-->A*x
ans  =

    1.
    1.
    2.

```

3.2. Solución general de un sistema utilizando \ y kernel

Si un sistema tiene una cantidad infinita de soluciones, hemos visto que el operador \ proporciona sólo una de las soluciones del sistema. No obstante, es posible obtener **todas** las soluciones de una manera fácil calculando el **núcleo** de la matriz de coeficientes. Primero debemos aclarar este concepto:

El **núcleo** de una matriz A es el conjunto de soluciones del sistema homogéneo cuya matriz de coeficientes es A, es decir, $A\vec{x} = \vec{0}$. Por ejemplo, el núcleo de la matriz

$$A = \begin{bmatrix} 1 & 2 & -3 \\ 2 & 4 & -6 \\ 5 & 10 & -15 \end{bmatrix}$$

es el conjunto de soluciones del sistema

$$\begin{bmatrix} 1 & 2 & -3 \\ 2 & 4 & -6 \\ 5 & 10 & -15 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

El núcleo de una matriz se puede calcular fácilmente con Scilab utilizando la función **kernel**:

```
-->A=[1 2 -3; 2 4 -6; 5 10 -15];
```

```
-->kernel(A)
```

```
ans =
```

```
- 0.1195229    0.9561829
```

```
0.8440132    - 0.0439019
```

```
0.5228345    0.2894597
```

El núcleo de la matriz es el conjunto de combinaciones lineales de los vectores columna de la matriz obtenida, es decir:

$$\text{Ker } A = \left\{ \alpha \begin{bmatrix} -0,1195229 \\ 0,8440132 \\ 0,5228345 \end{bmatrix} + \beta \begin{bmatrix} 0,9561829 \\ -0,0439019 \\ 0,2894597 \end{bmatrix} \text{ con } \alpha, \beta \in \mathbb{R} \right\}.$$

(Esto también puede expresarse diciendo que los vectores columna de la matriz forman un **sistema de generadores** del núcleo).

Ahora enunciaremos un teorema que muestra como obtener la solución general de un sistema de ecuaciones lineales a partir de

- una solución particular, y
- el núcleo de la matriz de coeficientes.

Teorema. Sea $A\vec{x} = \vec{b}$ un sistema de ecuaciones lineales compatible y sea \vec{x}_0 una solución particular. Entonces la solución general del sistema es:

$$\vec{x} = \vec{x}_0 + \lambda_1 \vec{u}_1 + \lambda_2 \vec{u}_2 + \dots + \lambda_n \vec{u}_n, \quad \lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{R},$$

donde $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_n\}$ es un sistema de generadores del núcleo de A.¹

Ejemplo 6. Consideramos el sistema de ecuaciones lineales

$$\begin{bmatrix} 1 & 0 & 2 & 3 \\ 7 & 1 & 1 & 1 \\ 8 & 1 & 3 & 4 \\ 9 & 1 & 5 & 7 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} = \begin{bmatrix} 6 \\ 10 \\ 16 \\ 22 \end{bmatrix}.$$

Aplicamos, con Scilab, el operador `\` para estudiar la compatibilidad del sistema:

```
-->A=[1 0 2 3; 7 1 1 1 ; 8 1 3 4; 9 1 5 7]; b=[6; 10; 16; 22];

-->x=A\b
warning :
matrix is close to singular or badly scaled. rcond =    2.2204D-18

x  =

    1.2
    0.
    0.
    1.6

-->clean(A*x-b)
ans  =

    0.
    0.
    0.
    0.
```

De los resultados anteriores podemos ver que el sistema es compatible y que el vector $\vec{x}_0 = (1, 2, 0, 1, 6)$ es una solución particular. Ahora calculamos el núcleo de la matriz de coeficientes:

```
-->kernel(A)
ans  =
```

¹La demostración es muy fácil: \vec{x} es una solución del sistema $\Leftrightarrow A\vec{x} = \vec{b} \Leftrightarrow A\vec{x} - A\vec{x}_0 = \vec{b} - A\vec{x}_0 \Leftrightarrow A(\vec{x} - \vec{x}_0) = \vec{0}$ (porque $A\vec{x}_0 = \vec{b}$) $\Leftrightarrow \vec{x} - \vec{x}_0$ pertenece al núcleo de A.

$$\begin{array}{rcl} - & 0.1490641 & - 0.0418627 \\ & 0.8434185 & 0.5144634 \\ & 0.4510273 & - 0.7061368 \\ - & 0.2509968 & 0.4847121 \end{array}$$

Esto quiere decir que el sistema tiene una cantidad infinita de soluciones y que su solución general es:

$$\underbrace{\begin{bmatrix} 1,2 \\ 0 \\ 0 \\ 1,6 \end{bmatrix}}_{\vec{x}_0} + \lambda_1 \begin{bmatrix} -0,1490641 \\ 0,8434185 \\ 0,4510273 \\ -0,2509968 \end{bmatrix} + \lambda_2 \begin{bmatrix} -0,0418627 \\ 0,5144634 \\ -0,7061368 \\ 0,4847121 \end{bmatrix}, \quad \lambda_1, \lambda_2 \in \mathbb{R}.$$

Ejemplo 7. Consideramos el sistema de ecuaciones lineales (con la misma matriz de coeficientes que antes)

$$\begin{bmatrix} 1 & 0 & 2 & 3 \\ 7 & 1 & 1 & 1 \\ 8 & 1 & 3 & 4 \\ 9 & 1 & 5 & 7 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 5 \end{bmatrix}.$$

Como antes, utilizando `\` para estudiar la compatibilidad del sistema:

```
-->b=[1; 2; 3; 5];
```

```
-->clean(A*x-b)
```

```
ans =
```

```
0.
0.
0.
- 1.
```

De estos resultados vemos que el operador `\` nos da un vector que **no es una solución**. Esto implica que **el sistema es incompatible**.

Ejemplo 8. Consideramos el sistema de ecuaciones lineales

$$\begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 1 \\ 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

Como antes, aplicamos primero el operador `\`:

```
-->A=[1 2 3; -1 0 1; 0 3 1]; b=[1; 2; 3];
```



```

-->x=A\b
x =

- 1.7
  0.9
  0.3

-->clean(A*x-b)
ans =

0.
0.
0.

```

Los resultados muestran que el sistema es compatible y que el vector $\vec{x}_0 = (-1,7, 0,9, 0,3)$ es una solución. Ahora calculamos el núcleo de la matriz de coeficientes:

```

-->kernel(A)
ans =

[]

```

Esto quiere decir que el núcleo es trivial, es decir, $\text{Ker } A = \{\vec{0}\}$. Por consiguiente la única solución del sistema es \vec{x}_0 , la obtenida utilizando \.

4. Redes de flujo

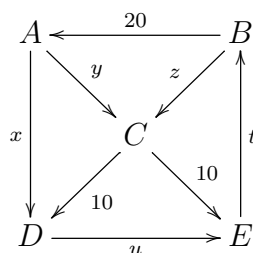
Cuando investigamos el flujo de una cantidad a través de una red nos aparecen sistemas de ecuaciones lineales. Estas redes las podemos encontrar en diversos campos de la ciencia, economía, estadística o ingeniería. Dos ejemplos de este tipo son los patrones de flujo de tráfico a través de una ciudad y la distribución de productos de los fabricantes a los consumidores por medio de una red de distribuidores y vendedores.

Una red consta de un conjunto de puntos, llamados nodos (o vértices), y arcos dirigidos que conectan todos o parte de los nodos. El flujo está indicado por un número o una variable. Un flujo de redes tiene que cumplir las siguientes condiciones:

- El flujo total que entra a un nodo es igual al flujo total que sale del nodo.
- El flujo total que entra dentro de la red es igual al flujo total que sale de la red.

Ejemplo 9. Consideremos una pequeña red cerrada de tubos a través de los cuales fluye un

líquido como se describe en el siguiente grafo:



Los arcos representan los tubos, y las intersecciones entre los tubos corresponden a los nodos de la red. El peso de cada arco indica la cantidad de litros de líquido que fluye por hora y las flechas indican las direcciones de los flujos.

A partir de las condiciones básicas de todo flujo de redes y de la descripción de la red concreta, podemos obtener el siguiente sistema de ecuaciones: cada nodo da lugar a una ecuación lineal.

$$\left. \begin{array}{lcl} \text{Nodo } A: & x + y & = 20 \\ \text{Nodo } B: & z + 20 & = t \\ \text{Nodo } C: & y + z & = 20 \\ \text{Nodo } D: & x + 10 & = u \\ \text{Nodo } E: & u + 10 & = t \end{array} \right\},$$

Para resolver el sistema, escribimos su matriz ampliada

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 20 \\ 0 & 0 & 1 & -1 & 0 & -20 \\ 0 & 1 & 1 & 0 & 0 & 20 \\ 1 & 0 & 0 & 0 & -1 & -10 \\ 0 & 0 & 0 & -1 & 1 & -10 \end{bmatrix}.$$

y calculamos, con Scilab, la forma escalonada reducida de M:

```
-->M=[1 1 0 0 0 20;0 0 1 -1 0 -20;0 1 1 0 0 20;
-->      1 0 0 0 -1 -10;0 0 0 -1 1 -10]); rref(M)
ans =
```

```
1.    0.    0.    0.   -1.   -10.
0.    1.    0.    0.    1.    30.
0.    0.    1.    0.   -1.   -10.
0.    0.    0.    1.   -1.    10.
0.    0.    0.    0.    0.    0.
```

Como se puede ver, el sistema es compatible indeterminado y la solución paramétrica del sistema es: $x = -10 + \lambda$, $y = 30 - \lambda$, $z = -10 + \lambda$, $t = 10 + \lambda$, $u = \lambda$, con $\lambda \in \mathbb{R}$. Como los valores de las incógnitas son litros de líquido, han de cumplir $x, y, z, t, u \geq 0$ y, por lo tanto, $10 \leq \lambda \leq 30$. Concluimos, entonces, que hay una cantidad infinita de posibilidades para la distribución de flujos (una para cada valor de λ en el intervalo $[10, 30]$).