

Resolución de sistemas lineales

Contenidos

- Introducción
- Métodos directos
- Métodos iterativos
- La operación \

Introducción

Queremos resolver sistemas de ecuaciones lineales con el mismo número de ecuaciones que de incógnitas con matriz de coeficientes no singular.

Dados los números a_{ij} y b_j para $i, j = 1, 2, \dots, n$ se trata de hallar los números $x_1, x_2, \dots, x_n$ que verifican las n ecuaciones lineales

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\&\vdots \\a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n\end{aligned}$$

Métodos directos

FACTORIZACIÓN LU

Descomponemos la matriz de coeficientes A en una matriz triangular superior U y una matriz triangular inferior L de forma que

$$A = LU$$

Consideramos el sistema de ecuaciones lineales

$$A\mathbf{x} = \mathbf{b}$$

donde la matriz A admite la factorización LU . Entonces

$$A\mathbf{x} = \mathbf{b} \iff LU\mathbf{x} = \mathbf{b}$$

Factorizaremos la matriz A usando la orden de Matlab `lu`

```
A=[2 1 1;1 2 -1;-1 1 2];  
[L U P]=lu(A)
```

$$\begin{aligned}
 L &= \begin{pmatrix} 1.0000 & 0 & 0 \\ 0.5000 & 1.0000 & 0 \\ -0.5000 & 1.0000 & 1.0000 \end{pmatrix} \\
 U &= \begin{pmatrix} 2.0000 & 1.0000 & 1.0000 \\ 0 & 1.5000 & -1.5000 \\ 0 & 0 & 4.0000 \end{pmatrix} \\
 P &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

Si la función `lu` reordena las filas se verifica que $PA = LU$ y estaremos resolviendo $P Ax = P b$ que es equivalente al sistema $Ax = b$.

Para resolver el sistema

- resolvemos $Ly = Pb$ usando *sustitución progresiva* y obtenemos y .
- resolvemos $Ux = y$ usando *sustitución regresiva* y obtenemos x .

ALGORITMO DE SUSTITUCIÓN PROGRESIVA

Si tenemos un sistema

$$Lx = b$$

donde L es una matriz triangular inferior:

$$L = \begin{pmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{pmatrix}$$

La ecuación i -ésima del sistema, para $i = 1, 2, \dots, n$ es:

$$l_{i1}x_1 + l_{i2}x_2 + \dots + l_{ii}x_i = b_i$$

Por lo tanto, para $i = 1, 2, \dots, n$, si l_{ii} no es 0 entonces

$$x_i = \frac{b_i - l_{i1}x_1 - \dots - l_{ii-1}x_{i-1}}{l_{ii}} = \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij}x_j \right)$$

Algoritmo

- Si $l_{11} = 0$, parar (no existe solución única).
- Hacer $x_1 = \frac{b_1}{l_{11}}$

- Para $i = 2, \dots, n$
- -----Si $l_{ii} = 0$, parar (no tiene solución)
- -----Hacer

$$x_i = \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij} x_j \right)$$

Ejercicio 1 Construir una función $x = \text{sust_pro}(L, b)$ donde x es la solución del sistema $Lx = b$ siendo L es una matriz triangular inferior. Resolver el sistema

$$\begin{pmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix}$$

```
L=[2 0 0;1 2 0;1 1 2];
b=[2 -1 0];
x=sust_pro(L,b')
```

$x =$

```
1
-1
0
```

ALGORITMO DE SUSTITUCIÓN REGRESIVA

Si tenemos un sistema

$$Ux = b'$$

donde U es una matriz triangular superior:

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & u_{nn} \end{pmatrix}$$

La ecuación i -ésima del sistema, para $i = 1, 2, \dots, n$ es:

$$u_{ii}x_i + u_{i+1,i}x_{i+1} + \dots + u_{in}x_n = b_i$$

Por lo tanto, para $i = n, n-1, \dots, 1$, si u_{ii} no es 0 entonces

$$x_i = \frac{b_i - u_{ii+1}x_{i+1} - \dots - u_{in}x_n}{u_{ii}} = \frac{1}{u_{ii}} \left(b_i - \sum_{j=i+1}^n u_{ij}x_j \right)$$

Algoritmo

- Si $u_{nn} = 0$, parar (no existe solución única).
- Hacer $x_n = \frac{b_n}{u_{nn}}$
- Para $i = n-1, \dots, 1$
- ---- Si $u_{ii} = 0$, parar (no tiene solución)
- ---- Hacer

$$x_i = \frac{1}{u_{ii}} \left(b_i - \sum_{j=i+1}^n u_{ij}x_j \right)$$

Ejercicio 2 Construir una función $x = \text{sust_re}(U, b)$ donde x es la solución del sistema $Ux = b$ siendo U es una matriz triangular superior. Resolver el sistema

$$\begin{pmatrix} 2 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 9 \\ 4 \\ 4 \end{pmatrix}$$

```
L=[2 1 1;0 2 1;0 0 2];
b=[9 4 4];
x=sust_re(U,b')
```

```
x =
    2.1667
    3.6667
    1.0000
```

Ejercicio 3 Dado el sistema $Ax = b$ con

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 1 & 2 \end{pmatrix} \quad b = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}$$

Resolverlo utilizando la descomposición LU.

Ejercicio3

```
L =
    1.0000         0         0
    0.5000    1.0000         0
    0.5000    0.3333    1.0000
```

```

U =
    2.0000    1.0000    1.0000
         0    1.5000    1.5000
         0         0    1.0000

P =
    1     0     0
    0     1     0
    0     0     1

b1 =
    2
    4
    6

y =
    2
    3
    4

x =
    0
   -2
    4

```

Métodos iterativos

Los métodos iterativos son, en general, más eficientes que los métodos directos para resolver sistemas de ecuaciones lineales grandes y con matrices de coeficientes con pocos elementos distintos de cero.

Los métodos iterativos se basan en reescribir el problema

$$A\mathbf{x} = \mathbf{b} \iff \mathbf{x} = B\mathbf{x} + \mathbf{c}$$

donde B es una matriz $n \times n$ y $\mathbf{c}$ es un vector columna de dimensión n

Entonces, si $\mathbf{x}^{(0)}$ es una aproximación inicial a la solución, para $k = 1, 2, \dots$

$$\mathbf{x}^{(k)} = B\mathbf{x}^{(k-1)} + \mathbf{c}$$

JACOBI

Algoritmo de Jacobi

- Elegir una aproximación inicial $\mathbf{x}^{(0)}$
- Para $k = 1, 2, \dots, \text{MaxIter}$
- ----Para $i = 1, 2, \dots, n$ calcular

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

- ----Si se cumple el criterio de parada, tomar $\mathbf{x}^{(k)}$ como aproximación a la solución.

Ejercicio 4 Crear una función `x=jacobi(A,b,tol)` que solucione el sistema lineal

$Ax = b$ por el método de Jacobi.

Resolver el sistema

$$\begin{cases} 5x_1 + x_2 - x_3 - x_4 = 1 \\ x_1 + 4x_2 - x_3 + x_4 = 1 \\ x_1 + x_2 - 5x_3 - x_4 = 1 \\ x_1 + x_2 + x_3 - 4x_4 = 1 \end{cases}$$

por el método de Jacobi, con tolerancia relativa 10^{-6} .

```
A=[ 5 1 -1 -1; 1 4 -1 1 ; 1 1 -5 -1; 1 1 1 -4];  
b=[1;1;1;1];  
tol=1e-6;  
x=jacobi(A,b,tol)
```

```
x =  
    0.0937  
    0.2500  
   -0.0937  
   -0.1875
```

GAUSS-SEIDEL

Algoritmo de Gauss-Seidel

- Elegir una aproximación inicial $\mathbf{x}^{(0)}$
- Para $k = 1, 2, \dots, MaxIter$
- ----Para $i = 1, 2, \dots, n$ calcular

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

- ----Si se cumple el criterio de parada, tomar $\mathbf{x}^{(k)}$ como aproximación a la solución.

Ejercicio 5 Crear una función `x=gauss_seidel(A,b,tol)` que solucione el sistema lineal $Ax = b$ por el método de Gauss-Seidel. Resolver el sistema anterior

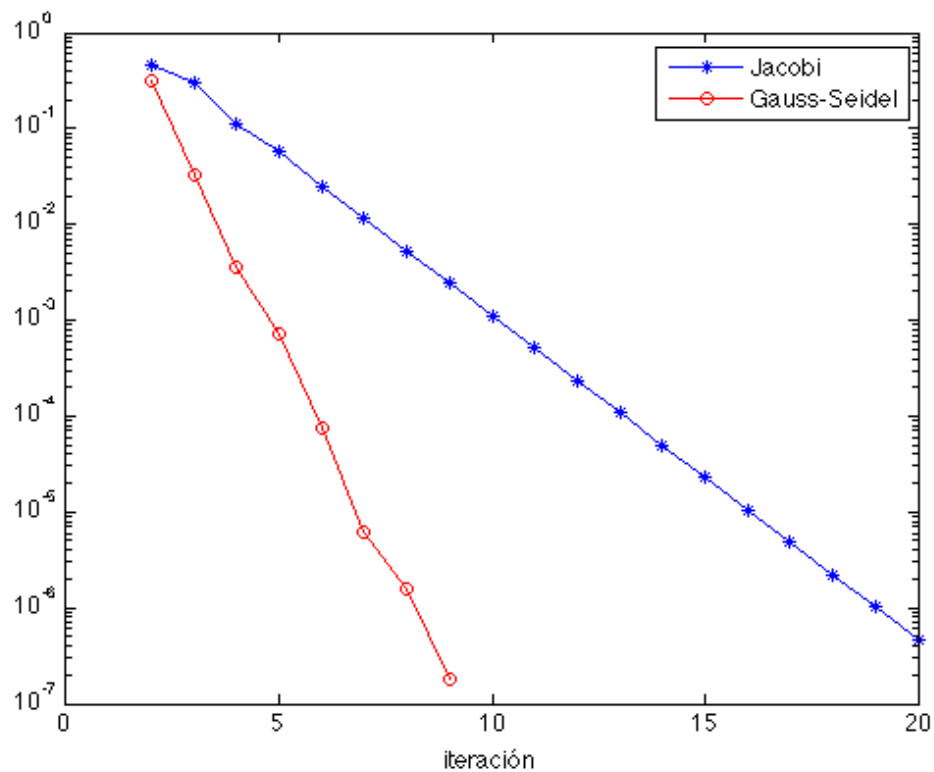
```
x=gauss_seidel(A,b,tol)
```

```
x =  
    0.0937  
    0.2500  
   -0.0938  
   -0.1875
```

Ejercicio 6 Modificar las funciones anteriores y llamarlas `jacobi2` y `gauss_seidel2`, de forma que devuelvan la sucesión de soluciones y la sucesión de errores relativos. Utilizarlas para:

- Dibujar la sucesión de errores relativos en escala logarítmica al resolver el sistema anterior por Jacobi y Gauss-Seidel.

Ejercicio6

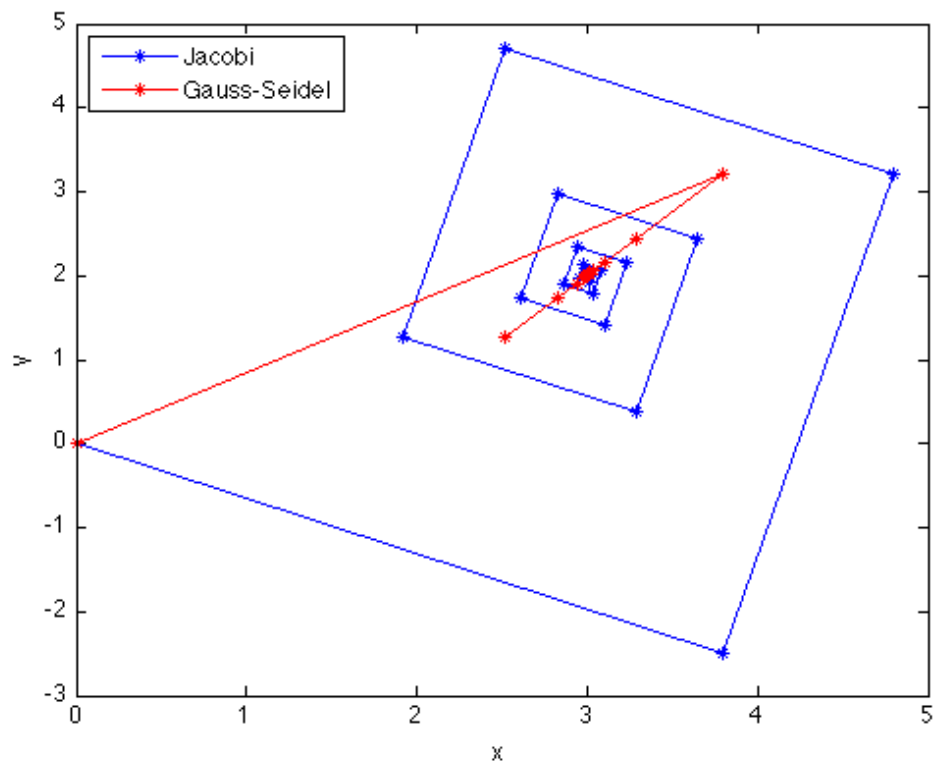


- Resolver por Jacobi y Gauss-Seidel el sistema

$$\begin{cases} 5x + 2y = 19 \\ 3x - 2y = 5 \end{cases}$$

con tolerancia relativa 10^{-2} y dibujar las sucesivas soluciones en el plano XY.

Ejercicio7



La operación \

Ejercicio

Resolver el sistema del ejercicio 4 utilizando el operador \.

```
A=[ 5 1 -1 -1; 1 4 -1 1 ; 1 1 -5 -1; 1 1 1 -4];
b=[1;1;1;1];
x=A\b
```

```
x =
    0.0937
    0.2500
   -0.0938
   -0.1875
```