SISTEMA TRIANGULAR INFERIOR

$$
\begin{bmatrix}
l_{11} & 0 & 0 & \dots & 0 \\
l_{21} & l_{22} & 0 & \dots & 0 \\
l_{31} & l_{32} & l_{33} & \dots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn}
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}
=
\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}
$$

Método de eliminación Gaussiana

El método de eliminación Gaussiana consiste en reducir el sistema $Ax = b$ a un sistema equivalente (es decir, que tenga la misma solución), de la forma

$$Ux = \tilde{b}, \tag{1}$$

donde U es una matriz triangular superior y $\tilde{b}$ un vector actualizado. Así el sistema resultante se resuelve por sustitución regresiva. Denotemos el sistema original por $A^{(1)}x = b^{(1)}$. El proceso empleado consiste en reemplazar las ecuaciones por combinaciones no triviales de las otras.

Así, consideremos la matriz no singular $A \in I\!R^{n\times n}$ y suponga que el elemento $a_{11}^{(1)}$ es no nulo. Consideremos los multiplicadores

$$m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}},\ i = 2, 3, \dots, n$$

donde, en general $a_{ij}^{(1)}$ donota el elemento que esta en la fila i y columna j de $A^{(1)}$. Es posible eliminar la incognita x_1 de las ecuaciones dos en adelante, por simple sustracción a la fila i, $i = 2, 3, \dots, n$ de la primera fila previamente multiplicada por m_{i1} y haciendo lo mismo para el vector b.

Si ahora definimos

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)},\ i, j = 2, \dots, n$$

$$b_i^{(2)} = b_i^{(1)} - m_{i1}b_1^{(1)},\ i = 2, \dots, n$$

donde $b_i^{(1)}$ denota la componente i-ésima del vector $b^{(1)}$, entonces se obtiene un

sistema equivalente al anterior

$$
\begin{pmatrix}
a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \dots & a_{1n}^{(1)} \\
0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2n}^{(2)} \\
0 & a_{32}^{(2)} & a_{33}^{(2)} & \dots & a_{3n}^{(2)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \dots & a_{nn}^{(2)}
\end{pmatrix}
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix}
=
\begin{pmatrix} b_1^{(1)} \\ b_2^{(2)} \\ b_3^{(2)} \\ \vdots \\ b_n^{(2)} \end{pmatrix}
$$

que lo denotaremos por $A^{(2)}x = b^{(2)}$.

Análogamente, podemos eliminar la incognita x_2 de las ecuaciones $3, \dots, n$. Siguiendo con este proceso un número finito de veces se obtiene el sistema

$$A^{(k)}x = b^{(k)},\ 1 \le k \le n \tag{2}$$

donde, para $k \geq 2$, la matriz $A^{(k)}$ toma la siguiente forma:

$$A^{(k)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \dots & \dots & \dots & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & \dots & \dots & \dots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \dots & \dots & \dots & \dots & a_{3n}^{(3)} \\ \vdots & \vdots & \ddots & \ddots & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \dots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 0 & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 0 & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{pmatrix},$$

para realizar este proceso, hemos asumido que $a_{ii}^{(i)} \neq 0$, para $i = 1, \dots, k-1$.

Notemos que para $k = n$ obtenemos el sistema triangular superior

$$\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \dots & \dots & \dots & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & \dots & \dots & \dots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \dots & \dots & \dots & \dots & a_{3n}^{(3)} \\ \vdots & \vdots & \ddots & \ddots & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \dots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 0 & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & a_{nn}^{(n)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_k \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^{(1)} \\ b_2^{(2)} \\ b_3^{(3)} \\ \vdots \\ \vdots \\ b_k^{(k)} \\ \vdots \\ b_n^{(n)} \end{pmatrix} \tag{3}$$

Los valores $a_{kk}^{(k)}$ son llamados pivotes y deben ser valores no nulos para $k = 1, \dots, n-1$.

Ejemplo

$$\begin{pmatrix} 1 & -3 & 2 \\ -2 & 8 & -1 \\ 4 & -6 & 5 \end{pmatrix} \overset{m_{21}=-2}{\sim} \begin{pmatrix} 1 & -3 & 2 \\ 0 & 2 & 3 \\ 4 & -6 & 5 \end{pmatrix}$$

$$\overset{m_{31}=4}{\sim} \begin{pmatrix} 1 & -3 & 2 \\ 0 & 2 & 3 \\ 0 & 6 & -3 \end{pmatrix}$$

$$\overset{m_{32}=3}{\sim} \begin{pmatrix} 1 & -3 & 2 \\ 0 & 2 & 3 \\ 0 & 0 & -12 \end{pmatrix}$$

$$U = \begin{pmatrix} 1 & -3 & 2 \\ 0 & 2 & 3 \\ 0 & 0 & -12 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & 3 & 1 \end{pmatrix}.$$

Notemos que

$$LU = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & 3 & 1 \end{pmatrix} \begin{pmatrix} 1 & -3 & 2 \\ 0 & 2 & 3 \\ 0 & 0 & -12 \end{pmatrix} = \begin{pmatrix} 1 & -3 & 2 \\ -2 & 8 & -1 \\ 4 & -6 & 5 \end{pmatrix} = A$$

ALGO DE LU Y MATRICES TRIDIAGONALES

Si $A = LU$, entonces resolver $Ax = b$ es equivalente a:

1. Resolver $Ly = b$, y luego
2. Resolver $Ux = y$.

Como estos sistemas son triangulares (inferior y superior) ellos son fácilmente resueltos.

El teorema (visto) nos dice que, salvo un cambio de filas (ecuaciones), todo sistema lineal del tipo $Ax = b$ (con A no singular) puede ser resuelto aplicando la descomposición LU.

ALGORITMO PARA RESOLVER $Ax = b$ USANDO LA DESCOMPOSICION LU

Para $\quad i = 1, 2, \ldots, n$

$$y_i = b_i - \sum_{k=1}^{i-1} L_{ik} y_k$$

Fin i

Para $\quad i = n, n-1, \ldots, 1$

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

Fin i

TEOREMA

Sea $A \in \mathbb{R}^{n\times n}$ no singular. Existen matrices L triangular inferior y U triangular superior tales que

$$A = LU$$

o bien,

$$PA = LU,$$

donde P es una matriz de permutación de filas.

ALGORITMO DE DESCOMPOSICION LU

Para $\quad k = 1, 2, \ldots, n$

$L_{kk} = 1$

Para $\quad j = k, k+1, \ldots, n$

$$U_{kj} = A_{kj} - \sum_{p=1}^{k-1} L_{kp} U_{pj}$$

Para $\quad i = k+1, k+2, \ldots, n \ (k \neq n)$

$$L_{ik} = \frac{1}{U_{kk}} \left(A_{ik} - \sum_{p=1}^{k-1} L_{ip} U_{pk} \right)$$

Fin k

ALGORITMO DE DESCOMPOSICION LU

$U_{11} = b_1$

$L_{11} = 1$

Para $\quad i = 2, 3, \ldots, n$

$\quad L_{ii} = 1$

$\quad U_{i-1\,i} = c_{i-1}$

$\quad L_{i\,i-1} = \dfrac{a_i}{U_{i-1\,i-1}}$

$\quad U_{ii} = b_i - L_{i\,i-1} c_{i-1}$

Fin i

Los valores que no aparecen en la descomposición valen cero.

Lo interesante de este método es que es $O(8n)$. Lo que es un avance con respecto al $O\left(\frac{2n^3}{3}\right)$ de Gauss o LU.

Si $n = 100$, entonces $\begin{cases} \left(\frac{2n^3}{3}\right) \sim 666{,}666 & \text{(MEG o LU)} \\ 8n \sim 800 & \text{(Thomas)} \end{cases}$

MATRICES TRIDIAGONALES

En muchas aplicaciones nos encontraremos con matrices tridiagonales invertibles, del tipo:

$$A = \begin{pmatrix} b_1 & c_1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ a_2 & b_2 & c_2 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & a_3 & b_3 & c_3 & \cdot & \cdot & \cdot & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & 0 & 0 & \ldots & 0 & a_n & b_n \end{pmatrix}$$

En este caso los algoritmos anteriores quedan:

Usando LU podemos llegar a un algoritmo *barato* para resolver $Ax = d$.

$w_1 = d_1$

Para $\quad i = 2, 3, \ldots, n$

$\quad w_i = d_i - L_{i\,i-1} w_{i-1}$

Fin i

$x_n = \dfrac{w_n}{U_{nn}}$

Para $\quad i = n-1, n-2, \ldots, 1$

$\quad x_i = \dfrac{w_i - U_{i\,i+1} x_{i+1}}{U_{ii}}$

Fin i

Este algoritmo es conocido como el algoritmo de *Thomas*.

Uso del comando LU

El comando LU es usado para encontrar la descomposición *LU* de una matriz no singular A.

Este comando acepta básicamente dos sintaxis; la primera es:

```
>> [L,U]=lu(A);
```

La salida es una matriz triangular superior *U* y una matriz ”psicológicamente triangular inferior” (es decir, un producto de una triangular inferior por una matriz de permutación) *L*, tal que *A*=*L*∗*U*.

La otra forma de usar este comando es:


```
>> [L,U,P]=lu(A);
```

La salida es una matriz triangular superior *U*, una matriz triangular inferior *L* y una matriz de permutación *P* tales que, *P*∗*A*=*L*∗*U*.

EJEMPLO

Consideremos la matriz

$$A = \begin{bmatrix} 0 & 1 & 3 \\ 4 & 2 & 5 \\ 7 & 2 & 8 \end{bmatrix}$$


```
>> A=[0 1 3;4 2 5;7 2 8];

>> [L,U]=lu(A)

L =
         0    1.0000         0
    0.5714    0.8571    1.0000
    1.0000         0         0
U =
    7.0000    2.0000    8.0000
         0    1.0000    3.0000
         0         0   -2.1429
```



```
>> [L,U,P]=lu(A)
L =
    1.0000         0         0
         0    1.0000         0
    0.5714    0.8571    1.0000
U =
    7.0000    2.0000    8.0000
         0    1.0000    3.0000
         0         0   -2.1429
P =
     0     0     1
     1     0     0
     0     1     0
```