Risoluzione di sistemi lineari: Il metodo LU

Sia A una matrice $n \times n$, x e b vettori colonna $(1 \times n)$, si vuole risolvere il sistema lineare

$$Ax = b.$$

La soluzione tramite Matlab di questa equazione puó avvenire in 2 modi:

- Usando l'inversa della martice A (x=inv(A)*b)
- Usando l'operatore *backslash* \.

$Ax = b$

```
>> A=[1 2 3; 4 5 6; 7 8 0]
>> b=[12; 33; 36]
>> x=inv(A)*b
x =
    4.0000
    1.0000
    2.0000
```

L'operazione é formalmente corretta ma é numericamente onerosa e lenta. La risoluzione del sistema si ottiene in Matlab usando il simbolo di divisione a sinistra *backslash* \

```
>> x=A\b  % soluzione del sistema Ax=b (x=inv(A)*b)

>> A=[1 2 3; 4 5 6; 7 8 0]
>> b=[12; 33; 36]
>> x=A\b
x =
    4.0000
    1.0000
    2.0000
```

L'operatore *backslash* \ usa algoritmi differenti per trattare diversi tipi di matrici:

- Permutazioni di matrici triangolari.
- Matrici simmetriche e definite positive.
- Sistemi rettangolari sovradeterminati.
- Sistemi rettangolari sottodeterminati.

Sistemi diagonali

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix} \quad b = \begin{pmatrix} -1 \\ 6 \\ -15 \end{pmatrix} \quad Ax = b \quad \text{equivale a} \quad \begin{cases} x_1 & & & = & -1 \\ & 3x_2 & & = & 6 \\ & & 5x_3 & = & -15 \end{cases}$$

La soluzione é

$$x_1 = -1 \qquad x_2 = \frac{6}{3} = 2 \qquad x_3 = \frac{-15}{5} = -3$$

Data: A,b
for $i \leftarrow 1$ **to** n **do**
  $x_i = \frac{b_1}{a_{i,j}}$;
end

Algorithm 1: Algoritmo per la risoluzione di un sistema diagonale

In Matlab si ha:

```
>> A=...   %A matrice diagonale
>> b=...   %b vettore colonna
>> x=b./diag(A)
```

Questo é l'unico caso in cui si usa l'operatore ./ puntuale nella risoluzione di sistemi lineari.

```
>> A=[1 0 0; 0 2 0; 0 0 3]
>> b=[1 ; 1 ; 1]
>> x1=b./diag(A)
x1 =
   1.0000
   0.5000
   0.3333

>> x2=A\b
x2 =
   1.0000
   0.5000
   0.3333
```

Sistemi triangolari:

$$L = \begin{pmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & & 0 \\ \vdots & \cdots & \ddots & \vdots \\ l_{n1} & \cdots & \cdots & l_{nn} \end{pmatrix} \qquad L \quad \text{triangolare inferiore} \quad Ly = b$$

$$U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & & u_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & & \cdots & u_{nn} \end{pmatrix} \qquad U \quad \text{triangolare superiore} \quad Ux = c$$

Sistemi triangolari risolvibili con metodi iterativi di sostituzione in avanti e sostituzione indietro.

$$A = \begin{pmatrix} -2 & 1 & 2 \\ 0 & 3 & -2 \\ 0 & 0 & 4 \end{pmatrix} \quad b = \begin{pmatrix} 9 \\ -1 \\ 8 \end{pmatrix} \quad Ax = b \qquad \text{equivale a} \qquad \begin{cases} -2x_1 & +x_2 & +2x_3 & = & 9 \\ & 3x_2 & -2x_3 & = & -1 \\ & & 4x_3 & = & 8 \end{cases}$$

La soluzione é

$$x_1 = \frac{8}{4} = 2 \qquad x_2 = \frac{1}{3}(-1 + 2x_3) = \frac{3}{3} = 1 \qquad x_3 = \frac{1}{-2}(9 - x_2 - 2x_3) = \frac{4}{-2} = -2.$$

$x_n = \frac{b_n}{u_{nn}}$;
for $i \leftarrow n-1$ **to** 1 **do**
| $x_i = \frac{1}{u_{ii}}\left(b_i - \sum_{j=1+1}^{n} u_{ij}x_j\right)$;
end

Algorithm 2: Algoritmo di sostituzione all'indietro

$x_1 = \frac{b_1}{l_{11}}$;
for $i \leftarrow 2$ **to** n **do**
| $x_i = \frac{1}{l_{ii}}\left(b_i - \sum_{j=1}^{i-1} u_{ij}x_j\right)$;
end

Algorithm 3: Algoritmo di sostituzione in avanti

Si vuole risolvere il sistema seguente

$$\begin{pmatrix} 2 & 6 & 4 \\ 1 & 0 & 2 \\ 2 & 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 10 \\ -1 \\ 5 \end{pmatrix}$$

Trasformando in un sistema triangolare facilmente risolvibile

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Posto

$$A^{(0)} = A \qquad A^{(0)} \begin{pmatrix} 2 & 6 & 4 \\ 1 & 0 & 2 \\ 2 & 2 & 1 \end{pmatrix}$$

$$b^{(0)} = b \qquad b^{(0)} \begin{pmatrix} 10 \\ -1 \\ 5 \end{pmatrix}$$

Si effettuano le trasformazioni su $A^{(0)}$ e $b^{(0)}$ in modo da avere

$$A^{(1)} = \begin{pmatrix} 2 & 6 & 4 \\ 0 & \times & \times \\ 0 & \times & \times \end{pmatrix} \quad b^{(1)} = \begin{pmatrix} 10 \\ \times \\ \times \end{pmatrix} \quad \rightarrow \quad A^{(1)} = \begin{pmatrix} 2 & 6 & 4 \\ 0 & -3 & 0 \\ 0 & -4 & -3 \end{pmatrix} \quad b^{(1)} = \begin{pmatrix} 10 \\ -6 \\ -5 \end{pmatrix}$$

Infine trasformando la matrice in una triangolare superiore tramite combinazione lineare tra le righe

$$U = A^{2)} = \begin{pmatrix} 2 & 6 & 4 \\ 0 & -3 & 0 \\ 0 & 0 & -3 \end{pmatrix} \quad y = b^{(2)} = \begin{pmatrix} 10 \\ -6 \\ 3 \end{pmatrix}$$

Risolvendo il sistema $Ux = b$ con sostituzioni all'indietro si ottiene la soluzione

$$x = (1, 2, -1)^T$$

for $k \leftarrow 1$ **to** $n-1$ **do**
  for $i \leftarrow k+1$ **to** n **do**
    $m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$;
    for $j \leftarrow k+1$ **to** n **do**
      $a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}$
    end
    $b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)}$
  end
end

Algorithm 4: Algoritmo di eliminazione di Gauss

Raccogliendo i moltiplicatori in una matrice triangolare inferiore L con diagonale unitaria e considerando la matrice triangolare superiore U ottenuta al passo $n-1$ si ottiene la fattorizzazione $A = LU$

$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ m_{12} & 1 & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ m_{n1} & \cdots & m_{n,n-1} & 1 \end{pmatrix} \quad U = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{nn}^{(n)} \end{pmatrix}$$

$$Ax = b \quad \rightarrow \quad LUx = b \quad \rightarrow \quad \begin{matrix} Ly = b \\ Ux = y \end{matrix}$$

Matlab calcola la fattorizzazioen LU di una matrice con il comando:

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

e si puó procedere alla risoluzione del sistema

```
>> y=L\b;
>> x=U\y
```

Di seguito un esempio di decomposizione LU

```
>> A=[1 0 1; 1 3 2; 1 -3 -8];
>> [L,U] = lu(A)
L =

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

```
U =

     1     0     1
     0     3     1
     0     0    -8
```

Esempio di risoluzione di un sistema lineare attraverso la decomposizione LU

```
>> A=[1 0 1; 1 3 2; 1 -3 -8];
>> b=[1; 2; 3]
>> [L,U] = lu(A)
>> y=L\b;
>> x=U\y


x =
    -1.3750
     0.4583
    -0.3750
```