Laboratorio di Calcolo Numerico

M.R. Russo

Università degli Studi di Padova
Dipartimento di Matematica Pura ed Applicata

A.A. 2009/2010

INDICE

- Sistemi lineari
- Casi particolari
- Eliminazione di Gauss
- Fattorizzazione LU
- Fattorizzazione Cholesky

Sistemi Lineari

La risoluzione di sistemi lineari può essere affrontata in MatLab in modo estremamente efficiente, originariamente MatLab è stato progettato proprio per svolgere questo compito.

Sistemi Lineari

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

$$A\,x = b$$

La soluzione tramite MatLab di questa equazione può avvenire in due modi.

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

Sistemi lineari

$$A\,x = b \longrightarrow x = invA\ b$$

```
>> 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 è formalmente corretta ma è numericamente onerosa e lenta.

Sistemi lineari

La risoluzione del sistema si ottiene in MatLab usando il simbolo di divisione a sx *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
```

Sistemi lineari

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 lineari: casi particolari

Sistemi triangolari:

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & & 0 \\ \vdots & & \ddots & \vdots \\ l_{n1} & & \cdots & l_{nn} \end{bmatrix}$$

L triangolare inferiore

L y= b

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & & u_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & & \cdots & u_{nn} \end{bmatrix}$$

U triangolare superiore

U x= c

Sistemi triangolari risolvibili con metodi di *sostituzione in avanti* e *sostituzione indietro*

Sistemi triangolari

$$A = \begin{bmatrix} -2 & 1 & 2 \\ 0 & 3 & -2 \\ 0 & 0 & 4 \end{bmatrix} \qquad b = \begin{bmatrix} 9 \\ -1 \\ 8 \end{bmatrix}$$

A x= b

equivale a

$$\begin{array}{rcrcrcr} -2x_1 & + & x_2 & + & 2x_3 & = & 9 \\ & & 3x_2 & + & -2x_3 & = & -1 \\ & & & & 4x_3 & = & 8 \end{array}$$

Risolvendo con sostituzioni all' indietro si ha:

$$x_3 = \frac{8}{4} = 2$$

$$x_2 = \frac{1}{3}(-1 + 2x_3) = \frac{3}{3} = 1$$

$$x_1 = \frac{1}{-2}(9 - x_2 - 2x_3) = \frac{4}{-2} = -2$$

Sistemi triangolari

Algoritmo di sostituzione all'indietro:

$$x_n = \frac{b_n}{u_{nn}}$$
$$x_i = \frac{1}{u_{ii}}\left(b_i - \sum_{j=i+1}^{n} u_{ij}x_j\right) \quad \text{per } i = n-1, \ldots, 1.$$

Algoritmo di sostituzione in avanti:

$$x_1 = \frac{b_1}{\ell_{11}}$$
$$x_i = \frac{1}{\ell_{ii}}\left(b_i - \sum_{j=1}^{i-1} \ell_{ij}x_j\right) \quad \text{per } i = 2, \ldots, n.$$

Eliminazione di Gauss

Nel metodo di eliminazione di Gauss il sistema lineare di partenza viene trasformato in uno equivalente ma di più facile soluzione in quanto la matrice del nuovo sistema ha forma triangolare.

$$A = [a_{ij}] = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \longrightarrow \begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & 0 & a_{33}^{(2)} & \dots & a_{3n}^{(2)} \\ \vdots & \vdots & \dots & \dots & \vdots \\ 0 & 0 & \dots & 0 & a_{nn}^{(n-1)} \end{pmatrix}$$

Eliminazione di Gauss

Al primo passo vogliamo eliminare gli elementi della prima colonna al di sotto della diagonale principale:

- sottraiamo la prima equazione moltiplicata per $\frac{a_{21}}{a_{11}}$ dalla seconda equazione:

$$
\begin{array}{l}
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2 \\
\dfrac{a_{21}}{a_{11}}(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n) = \dfrac{a_{21}}{a_{11}}b_1 \\
\hline
(a_{22} - \dfrac{a_{21}}{a_{11}}a_{12})x_2 + (a_{23} - \dfrac{a_{21}}{a_{11}}a_{13})x_3 + \ldots + (a_{2n} - \dfrac{a_{21}}{a_{11}}a_{1n})x_n = b_2 - \dfrac{a_{21}}{a_{11}}b_1
\end{array}
$$

Eliminazione di Gauss

- sottraiamo la prima equazione moltiplicata per $\frac{a_{31}}{a_{11}}$ dalla terza equazione.
- ...
- sottraiamo la prima equazione moltiplicata per $\frac{a_{n1}}{a_{11}}$ dalla n-sima equazione.

Come risultato si avrà la nuova matrice con elementi nulli sulla prima colonna, dal secondo in poi e il sistema diventa

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & \vdots & \dots & \vdots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ \vdots \\ b_n^{(1)} \end{pmatrix}$$

Eliminazione di Gauss

Al secondo passo, consideriamo il sistema ridotto che si ha ignorando la prima equazione del sistema e la prima colonna della nuova matrice che abbiamo ottenuta (che ha tutti 0 al di sotto dell'elemento diagonale).

$$
\begin{matrix}
a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\
\vdots & \dots & \vdots \\
a_{n2}^{(1)} & \dots & a_{nn}^{(1)},
\end{matrix}
$$

A questa sottomatrice applichiamo lo stesso procedimento di prima, sottraendo, quindi, la prima equazione della sottomatrice moltiplicata per $\frac{a_{32}^{(1)}}{a_{22}^{(1)}}$ dalla seconda equazione della sottomatrice, e così via.

Eliminazione di Gauss

Dopo questo passo il sistema sarà equivalente a

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

Eliminazione di Gauss

Dopo aver applicato questo procedimento n-1 volte si ottiene un sistema triangolare superiore semplice da risolvere tramite sostituzione all'indietro.

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

Eliminazione di Gauss

$$A = \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix} \qquad b = \begin{pmatrix} 2 \\ 3.5 \end{pmatrix}$$

Per applicare il metodo di Gauss, dobbiamo moltiplicare la prima equazione per $\frac{3}{2}$ *e sottrarla dalla seconda*

$$\begin{array}{lc} 3x_1 + 2x_2 = 3.5 & - \\ \frac{3}{2}(2x_1 + 1x_2) = \frac{3}{2}2 & = \\ \hline 0x_1 + 0.5x_2 = 0.5 & \end{array}$$

Il sistema equivalente diventa

$$\begin{pmatrix} 2 & 1 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 0.5 \end{pmatrix}$$

Eliminazione di Gauss

Si vuole risolvere il sistema seguente

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

trasformandolo in un sistema triangolare facilmente risolvibile

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

Eliminazione di Gauss

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 trasformazioni su A e b in modo da avere

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

Eliminazione di Gauss

Infine si trasforma 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 *U x=b* con sostituzioni all'indietro si ottiene la soluzione

$$x = (1, 2, -1)'.$$

Eliminazione di Gauss

Algoritmo di eliminazione di Gauss

for $k = 1, \ldots, n-1$
    for $i = k+1, \ldots, n$
        $m_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$
        for $j = k+1, \ldots, n$
            $a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}$
        end
        $b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)}$
    end
end

Fattorizzazione LU

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*

$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ m_{n1} & \cdots & m_{nn-1} & 1 \end{pmatrix} \quad U = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & a_{nn}^{(n)} \end{pmatrix}$$

$$Ax = b \longrightarrow LUx = b. \longrightarrow \begin{array}{l} Ly = b \\ Ux = y. \end{array}$$

Fattorizzazione LU

Il generico passo k-esimo del metodo può essere ottenuto premoltiplicando la matrice A per la matrice M che raccoglie i moltiplicatori

$$
\begin{aligned}
&A_1 = M_1 \cdot A \\
&A_2 = M_2 \cdot A_1 = M_2 \cdot M_1 \cdot A \\
&A_3 = M_3 \cdot A_2 = M_3 \cdot M_2 \cdot M_1 \cdot A \\
&\vdots \\
&A_{n-1} = M_{n-1} \cdot A_{n-2} = M_{n-1} \cdot M_{n-2} \cdots M_2 \cdot M_1 \cdot A \\
&U = M_{n-1} \cdot M_{n-2} \cdots M_2 \cdot M_1 \cdot A
\end{aligned}
$$

$$
A = LU \quad \Rightarrow \quad L = (M_{n-1} \cdot M_{n-2} \cdots M_2 \cdot M_1)^{-1}
$$

Fattorizzazione LU

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

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

e si può procedere alla risoluzione del sistema

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

Fattorizzazione 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
```

Soluzione sistema con fattorizzazione 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
```

Fattorizzazione LU

Essendo L ed U di forma triangolare è possibile memorizzare la fattorizzazione LU direttamente nella stessa area di memoria di occupata da A, U occupa la parte triangolare superiore di A, inclusa la diagonale principale, L la triangolare inferiore con elementi unitari sulla diagonale.

Si parla di fattorizzazioni LU sul posto della matrice A.

Ci sono differenti versioni di tale fattorizzazione che dipendono dall'ordine secondo cui vengono eseguiti i cicli.

Fattorizzazione LU sul posto

```
function [A] = fatt_LU_kji(A)
n=size(A,1);
for k=1:n-1
  A(k+1:n,k)=A(k+1:n,k)/A(k,k);
  for j=k+1:n
    for i=k+1:n
      A(i,j)=A(i,j)-A(i,k)*A(k,j);
    end,
  end
end
```

Versione kji: l'operazione fondamentale di tale algoritmo consiste nel moltiplicare uno scalare per un vettore, aggiungervi un altro vettore e memorizzare il risultato.

```
function [A] = fatt_LU_ijk(A)
n=size(A,1);
for i=1:n
  for j=2:i
    A(i,j-1) = A(i,j-1)/A(j-1,j-1);
    for k=1:j-1
      A(i,j)=A(i,j)-A(i,k)*A(k,j) ;
    end
  end
  for j=i+1:n
    for k=1:i-1
      A(i,j)=A(i,j)-A(i,k)*A(k,j) ;
    end
  end
end
```

Versione ijk: l'operazione fondamentale di tale algoritmo consiste nel prodotto scalare. E' detto anche schema compatto di Doolittle.

Le formule in cui il ciclo su i precede il ciclo su j sono orientate per righe, le altre sono orientate per colonne.

Tecnica 'pivoting'

Per evitare possibili divisioni per 0 e per rendere l'algoritmo di eliminazione o di fattorizzazione LU stabili rispetto alla propagazione degli errori di arrotondamento si usa la strategia di pivoting che consiste nello scambio di righe o colonne opportune.

Il risultato della fattorizzazione LU è

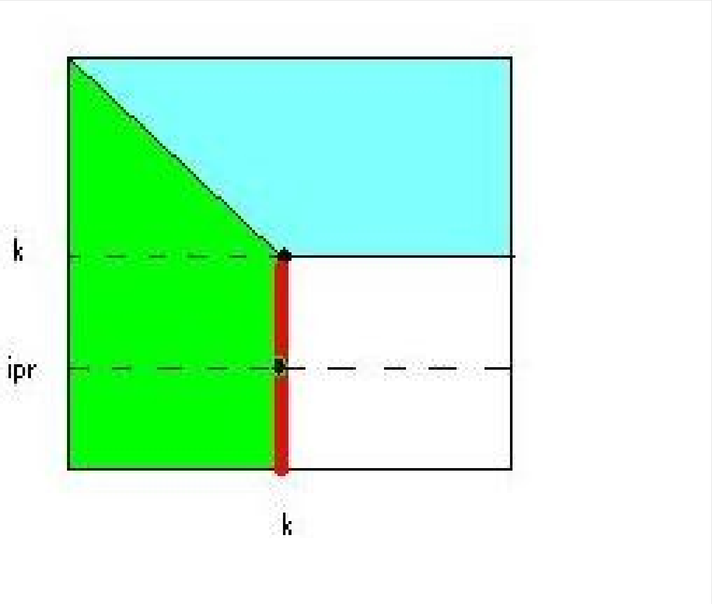
$$PA = LU$$

essendo P una matrice di permutazione che tiene conto degli scambi avvenuti.

Pivoting parziale

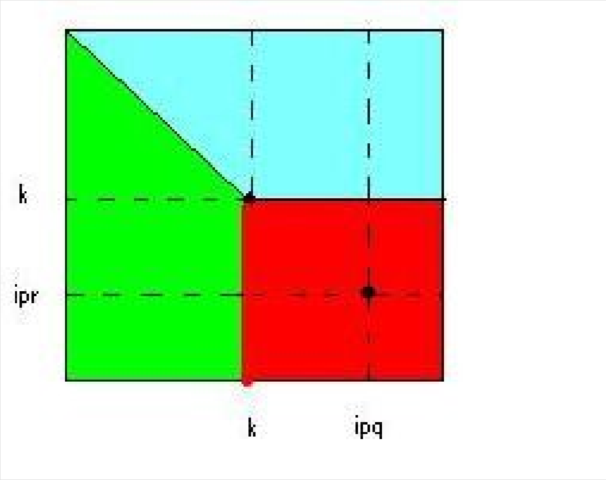
Pivoting parziale: si cerca il coefficiente a_{pk} di modulo massimo con p>k, si scambia la riga p con la riga k

for $k = 1, \ldots, n-1$
 cerco più piccolo p tale che $|a_{pk}^{(k)}| = \max_{k \le i \le n} |a_{ik}^{(k)}|$
 scambio la riga k con la riga p
 for $i = k+1, \ldots, n$
  $m_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$
  for $j = k+1, \ldots, n$
   $a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}$
  end
  $b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)}$
 end
end



Pivoting totale

Pivoting totale: si cerca il coefficiente a_{pp} di modulo massimo su tutte le righe i e le colonne j, con i e j>k, e si scambia la riga i con la riga k e la colonna j con la colonna k. Tale tecnica ha costi computazionali maggiori quindi si preferisce sempre il pivoting parziale.



Matrici di permutazione

Si consideri la matrice di permutazione P di dimensione 3

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Qualunque sia la matrice A(3,3), moltiplicandola a sinistra per P si ottiene lo scambio della seconda e terza riga di A, mentre moltiplicandola a destra per P si ottiene lo scambio della seconda e terza colonna di A:

$$\mathrm{PA} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{31} & a_{32} & a_{33} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

$$\mathrm{AP} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{13} & a_{12} \\ a_{21} & a_{23} & a_{22} \\ a_{31} & a_{33} & a_{32} \end{pmatrix}$$

LU con pivoting

$$PA = LU \Rightarrow PAx = LUx$$

$$Ax = b \Rightarrow LUx = Pb$$

Si risolvono i sistemi triangolari

$$\begin{aligned} Ly &= Pb \\ Ux &= y \end{aligned}$$

Fattorizzazione Cholesky

Se A è una matrice simmetrica definita positiva, esiste un'unica matrice L triangolare inferiore per cui si ha la fattorizzazione

$$A = LL^T$$

Calcolata L si procede alla risoluzione dei due sistemi triangolari:

$$Ly = b$$
$$L^T x = y$$

Fattorizzazione Cholesky

In MatLab la fattorizzazione di Cholesky si ottiene con il comando chol

```
>> A=[1 -1 -1;-1 2 0;-1 0 3];
>> R=chol(A)
R =

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

*R'*R=A, R corrisponde a L'*

Fattorizzazione Cholesky

Permette di sapere se A è definita positiva attraverso un flag p che risulta essere 0:

```
>> [R,p]=chol(A)
R =

    1    -1    -1
    0     1    -1
    0     0     1

p =

    0
```

Fattorizzazione Cholesky

Se A non è definita positiva e la funzione chol viene richiamata con un solo argomento in uscita, verrà visualizzato un messaggio di errore:

```
>>A=[1 2 3;2 5 4;3 4 8]
>> R=chol(A)
??? Error using ==> chol
Matrix must be positive definite.
```

Usando due argomenti in uscita, non segnala errore ma risulta p=3.

```
>> [R,p]=chol(A)
>>p=3
```