Scuola Interdipartimentale delle Scienze, dell'Ingegneria e della Salute

DiST Dipartimento di Scienze e Tecnologie

Laurea Magistrale in STN

Applicazioni di Calcolo Scientifico e Laboratorio di ACS

(12 cfu)

prof. Mariarosaria Rizzardi

Centro Direzionale di Napoli – Isola C4
stanza: n. 423 – Lato Nord, 4° piano
tel.: 081 547 6545
email: mariarosaria.rizzardi@uniparthenope.it

Argomenti trattati

- **Calcolo Numerico in MATLAB:**
  - Richiami sull'algebra delle matrici e dei vettori.
  - Richiami sui sistemi lineari e sul metodo di Gauss.
  - Fattorizzazioni di matrice: LU, QR, SVD.
- **Richiami di Algebra Lineare numerica.**

Richiami di Algebra dei vettori e delle matrici

Vettori

$$v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$ vettore colonna (default MATLAB)

$$v = \begin{pmatrix} 1 \\ -2 \\ -1 \end{pmatrix}$$

```
v=[1;-2;-1];
n=numel(v)
n =
     3
[m,n]=size(v)
m =
     3
n =
     1
```

```
n=length(v)
n =
     3
```

```
m=size(v,1)
m =
     3
```

$$w = \begin{pmatrix} w_1 & w_2 & \cdots & w_n \end{pmatrix}$$ vettore riga

$$w = \begin{pmatrix} 1 & -2 & -1 \end{pmatrix}$$

```
w=[1 -2 -1];
n=numel(w)
n =
     3
[m,n]=size(w)
m =
     1
n =
     3
```

```
n=length(w)
n =
     3
```

```
n=size(w,2)
m =
     3
```

Matrici

$$A_{m\times n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n-1} & a_{2,n} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n-1} & a_{m,n} \end{pmatrix}$$

matrice di m righe e n colonne

$$A = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 4 & 1 & 0 & -2 \\ 2 & -2 & -1 & 1 \end{pmatrix}$$

```
A=[2 1 1 0;4 1 0 -2;
   2 -2 -1 1];
[m,n]=size(A)
m =
     3
n =
     4
```

```
n=length(A)
n =
     4
q=numel(A)
q =
    12
```

$$B = \begin{pmatrix} 2 & 4 & 2 \\ 1 & 1 & -2 \\ 1 & 0 & -1 \\ 0 & -2 & 1 \end{pmatrix}$$

trasposta di A

```
B=A.';
[m,n]=size(B)
m =
     4
n =
     4
```

```
m=length(B)
m =
     4
q=numel(B)
q =
    12
```

lunghezza della dimensione massima

Particolari matrici

Matrice trasposta A^{T}

matrice di m righe e n colonne

$$A_{m\times n}=\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n}\\ a_{2,1} & a_{2,2} & \cdots & a_{2,n-1} & a_{2,n}\\ \vdots & \vdots & \cdots & \vdots & \vdots\\ a_{m,1} & a_{m,2} & \cdots & a_{m,n-1} & a_{m,n}\end{pmatrix}$$

matrice di n righe e m colonne

$$B_{n\times m}=A^{\mathsf{T}}_{m\times n}=\begin{pmatrix} a_{1,1} & a_{2,1} & \cdots & a_{m-1,1} & a_{m,1}\\ a_{1,2} & a_{2,2} & \cdots & a_{m-1,2} & a_{m,2}\\ \vdots & \vdots & \cdots & \vdots & \vdots\\ a_{1,n} & a_{2,n} & \cdots & a_{m-1,n} & a_{m,n}\end{pmatrix}$$

$B=A^{\mathsf{T}} \Rightarrow$ si scambiano le righe con le colonne: $b_{ij}=a_{ji}$

Matrice trasposta coniugata A^{H}

Se $a\in\mathbb{C}$: $a=\alpha+i\,\beta$, allora il suo complesso coniugato è dato da: $\bar{a}=\alpha-i\,\beta$

$$A_{m\times n}=\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n}\\ a_{2,1} & a_{2,2} & \cdots & a_{2,n-1} & a_{2,n}\\ \vdots & \vdots & \cdots & \vdots & \vdots\\ a_{m,1} & a_{m,2} & \cdots & a_{m,n-1} & a_{m,n}\end{pmatrix}$$

$$B_{n\times m}=A^{\mathsf{H}}_{m\times n}=\begin{pmatrix} \bar{a}_{1,1} & \bar{a}_{2,1} & \cdots & \bar{a}_{m-1,1} & \bar{a}_{m,1}\\ \bar{a}_{1,2} & \bar{a}_{2,2} & \cdots & \bar{a}_{m-1,2} & \bar{a}_{m,2}\\ \vdots & \vdots & \cdots & \vdots & \vdots\\ \bar{a}_{1,n} & \bar{a}_{2,n} & \cdots & \bar{a}_{m-1,n} & \bar{a}_{m,n}\end{pmatrix}$$

$B=A^{\mathsf{H}} \Rightarrow$ si scambiano le righe con le colonne e si complementa: $b_{ij}=\bar{a}_{ji}$

Se la matrice A è reale, allora $A^{\mathsf{H}}=A^{\mathsf{T}}$.

Richiami di Algebra delle matrici

In **MATLAB** la scrittura **A'** calcola A^{H}, mentre **A.'** calcola A^{T}.

Se la matrice **A** è reale, ovviamente risulta **A'=A.'**.

```
v=sym('v',[3 1])
v =
v1
v2
v3
```

```
u=v'
u =
[conj(v1), conj(v2), conj(v3)]
w=v.'
w =
[v1, v2, v3]
```

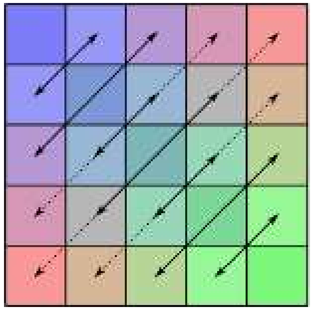
```
v=sym('v',[3 1],'real');
```

```
u=v'
u =
[v1, v2, v3]
w=v.'
w =
[v1, v2, v3]
```

Particolari matrici

Matrice simmetrica

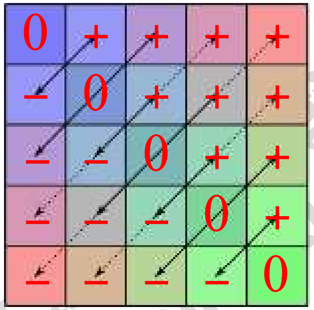
è una matrice quadrata $A(n\times n)$ che coincide con la sua trasposta A^T:



$$A = A^T \quad \overset{\text{def}}{\Longleftrightarrow} \quad a_{i,j} = a_{j,i}, \quad \forall\, i,j$$

Matrice antisimmetrica

è una matrice quadrata $A(n\times n)$ che coincide con l'opposta della sua trasposta A^T:



$$A = -A^T \quad \overset{\text{def}}{\Longleftrightarrow} \quad a_{i,j} = -a_{j,i}, \quad \forall\, i,j$$

Matrice hermitiana

è una matrice quadrata $A(n\times n)$ che coincide con la sua trasposta coniugata A^H:

$$A = A^H \quad \overset{\text{def}}{\Longleftrightarrow} \quad a_{i,j} = \overline{a}_{j,i}, \quad \forall\, i,j$$

Particolari matrici

Matrice inversa* A^{-1}

* esistono anche le inverse generalizzate, le pseudoinverse: ne parleremo in seguito …

Data una matrice quadrata $A(n\times n)$, invertibile (o non singolare), la sua matrice inversa A^{-1} è tale che:

$$AA^{-1} = A^{-1}A = \mathbf{I}$$

Una matrice quadrata A $(n\times n)$ è **invertibile** (o **non singolare**) se:

- rango$(A) = n$ (rango massimo)
- oppure det$(A) \neq 0$

rango: massimo numero di righe o di colonne della matrice linearmente indipendenti (… in seguito)

det: determinante della matrice

Matrice ortogonale

È una matrice quadrata reale $A(n\times n)$ tale che la sua inversa A^{-1} coincide con la trasposta A^{T}:

$$A^{-1} = A^{\mathsf{T}} \overset{\text{def}}{\Longleftrightarrow} A^{\mathsf{T}}A = A\,A^{\mathsf{T}} = \mathbf{I}$$

Matrice unitaria

È una matrice quadrata complessa $A(n\times n)$ tale che la sua inversa A^{-1} coincide con la trasposta coniugata A^{H}:

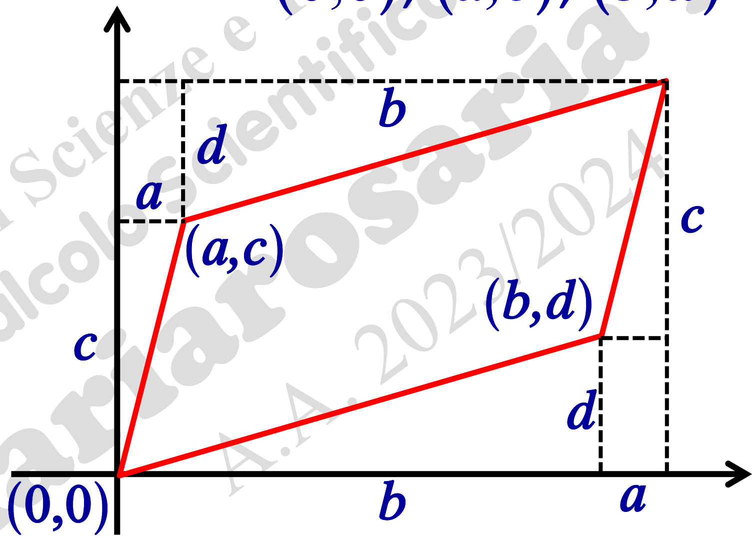
$$A^{-1} = A^{\mathsf{H}} \overset{\text{def}}{\Longleftrightarrow} A^{\mathsf{H}}A = A\,A^{\mathsf{H}} = \mathbf{I}$$

Determinante di una matrice $A(n\times n)$

Per $n=2$, il determinante della matrice $A=\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ è dato da:

$$\det A = ad - bc$$

Il valore assoluto del determinante rappresenta l'area del parallelogramma con vertici in $(0,0)$, (a,c), (b,d).



Esercizio
Verificarlo

Per $n=3$, il valore assoluto del determinante della matrice $A=\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$ rappresenta il volume del parallelepipedo con vertici dati dall'origine e dalle colonne della matrice.

Determinante di una matrice $A(n\times n)$

Teorema di Laplace

- Per $n=1$, cioè per $A=(\alpha)$, il determinante di A è **det** $A=\alpha$.
- Per $n>1$ qualsiasi, il determinante di A è dato da

$$\det A = \sum_{k=1}^{n} a_{h,k} A_{h,k} = \sum_{k=1}^{n} a_{k,h} A_{k,h}, \quad \forall h \in \; 1,2,\ldots,n$$

sviluppo per righe   sviluppo per colonne

cioè il determinante è la somma degli elementi $a_{i,j}$ di una qualsiasi riga (o colonna), ciascuno moltiplicato per il proprio ***complemento algebrico*** $A_{i,j}$.

Il ***complemento algebrico*** $A_{i,j}$ dell'elemento $a_{i,j}$ della matrice A è il numero

$$A_{i,j} = (-1)^{i+j} M_{i,j}$$

dove $M_{i,j}$ (detto ***minore complementare dell'elemento*** $a_{i,j}$) è il determinante della matrice, di dimensione $(n-1)\times(n-1)$, che si ottiene da A cancellando la riga i-sima e la colonna j-sima.

Esempio

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 4 & 4 & 0 \\ 2 & -2 & -1 \end{pmatrix}$$

$a_{3,2}$

$$M_{3,2} = \det\begin{pmatrix} 2 & 1 \\ 4 & 0 \end{pmatrix} = -4$$

$$A_{3,2} = (-1)^{3+2}(-4) = 4$$

Esempio

Calcolare, mediante *Teor. di Laplace*, il determinante della seguente matrice

$$\boldsymbol{A} = \begin{pmatrix} 2 & 1 & 1 \\ 4 & 4 & 0 \\ 2 & -2 & -1 \end{pmatrix}$$

Sviluppando il **det** A lungo la terza colonna, si ha

$$\det A = a_{1,3}A_{1,3} + a_{2,3}A_{2,3} + a_{3,3}A_{3,3} =$$
$$= 1(-1)^{1+3} M_{1,3} + 0(-1)^{2+3} M_{2,3} + (-1)(-1)^{3+3} M_{3,3} = -20$$

essendo

$$M_{1,3} = \det\begin{pmatrix} 4 & 4 \\ 2 & -2 \end{pmatrix} = -16 \qquad M_{2,3} = 0 \qquad M_{3,3} = \det\begin{pmatrix} 2 & 1 \\ 4 & 4 \end{pmatrix} = 4$$

Il calcolo del determinante di una matrice mediante Teor. di Laplace è estremamente inefficiente; in pratica si usa la fattorizzazione $\boldsymbol{A=LU}$

Richiami: principali proprietà del rango e del determinante

$$\boxed{A(m\times n)}$$

$$r(A)=rango(A) \quad \Rightarrow \quad \begin{array}{l} r(A) = r(A^\top) \\ r(AB) \le \min\{r(A), r(B)\} \end{array}$$

$$(AB)^\top = B^\top A^\top \quad \Rightarrow \quad (Au)^\top = u^\top A^\top \qquad \begin{array}{c} Au = z \\ u^\top A^\top = z^\top \end{array}$$

$$\boxed{A(n\times n)}$$

$$\det(AB) = \det(A)\det(B)$$

$$(AB)^{-1} = B^{-1}A^{-1}$$

$$(A^\top)^{-1} = (A^{-1})^\top$$

Matrici strutturate

$$A_{n\times n}=\begin{pmatrix} a_{1,1} & 0 & \cdots & 0 & 0\\ 0 & a_{2,2} & \cdots & 0 & 0\\ \vdots & \vdots & \cdots & \vdots & \vdots\\ 0 & 0 & \cdots & 0 & a_{n,n}\end{pmatrix}$$

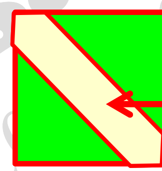
matrice diagonale

È una matrice quadrata avente elementi non nulli solo sulla diagonale principale: $a_{ij}=0,\ \forall i\neq j$

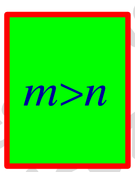
$$A_{n\times n}=\begin{pmatrix} a_{1,1} & a_{1,2} & 0 & \cdots & \cdots & 0\\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & \cdots & 0\\ 0 & a_{3,2} & a_{3,3} & a_{3,4} & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots\\ \vdots & 0 & \cdots & \cdots & a_{n-1,n-1} & a_{n-1,n}\\ 0 & 0 & \cdots & 0 & a_{n,n-1} & a_{n,n}\end{pmatrix}$$

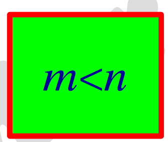
matrice tridiagonale

È una matrice quadrata avente elementi non nulli solo sulla diagonale principale e sulle due diagonali adiacenti: $a_{ij}=0,\ \forall\,|i-j|>1$

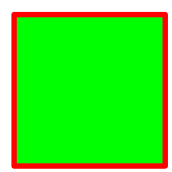
$A =$ 

caso particolare di matrice a banda

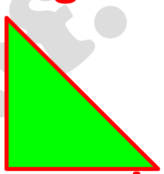
$A =$ 

$A =$ 

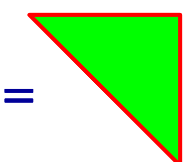
matrici rettangolari: $m\neq n$

$A =$ 

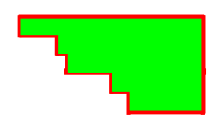
matrice quadrata: $m=n$

$A =$ 

matrice quadrata triangolare inferiore: $a_{ij}=0,\ j>i+1$

$A =$ 

matrice quadrata triangolare superiore: $a_{ij}=0,\ i>j+1$



matrice a scala o **trapezoidale**

Operazioni su matrici e vettori

Uguaglianza Date le matrici $A(m\times n)$ e $B(p\times q)$, allora

$$A = B \quad \overset{\text{def}}{\Longleftrightarrow} \quad \begin{array}{l} m=p, \quad n=q \\ a_{i,j} = b_{i,j} \quad \forall i,j \end{array}$$

uguale size

Addizione Date le matrici A e B di egual size $(m\times n)$, allora

$$C = A+B \quad \overset{\text{def}}{\Longleftrightarrow} \quad c_{i,j} = a_{i,j}+ b_{i,j} \quad \forall i,j$$

Moltiplicazione per uno scalare

Data la matrice $A(m\times n)$, ed un numero reale α, allora

$$C = \alpha\cdot A \quad \overset{\text{def}}{\Longleftrightarrow} \quad c_{i,j} = \alpha\cdot a_{i,j} \quad \forall i,j$$

Operazioni su matrici e vettori

Prodotto scalare (standard) di due vettori

Dati due vettori colonna $\boldsymbol{u}$ e $\boldsymbol{v}$ di eguale dimensione n, allora

$$\langle u,v\rangle = \alpha \in \mathbb{R} \quad \overset{\text{def}}{\Longleftrightarrow} \quad \alpha = \sum_{k=1}^{n} u_k v_k = \|\underline{u}\|\,\|\underline{v}\|\cos\theta$$

```
a=u'*v
a=dot(u,v)
```

Prodotto matrice per vettore

Data una matrice $A(m\times n)$ ed un vettore colonna $\boldsymbol{u}$ di dimensione n, allora

```
v=A*u
```

$$v = Au \quad \overset{\text{def}}{\Longleftrightarrow} \quad v_i = \langle A_{i,:},\, u\rangle,\ \forall i, \quad v\in\mathbb{R}^m$$

riga i-sima

Prodotto righe per colonne di due matrici

Date due matrici $A(m\times n)$ e $B(n\times q)$, allora

riga i-sima colonna j-sima

```
C=A*B
```

$$C = AB \quad \overset{\text{def}}{\Longleftrightarrow} \quad c_{i,j} = \langle A_{i,:},\, B_{:,j}\rangle,\ \forall i,j,\ \ C(m\times q)$$

Altre operazioni su matrici e vettori

Prodotto di Hadamard di due vettori o matrici

Date due matrici $\boldsymbol{A}$ e $\boldsymbol{B}$, di eguale size ($m\times n$), allora

$$C = A\odot B \quad \overset{\text{def}}{\Longleftrightarrow} \quad c_{i,j} = a_{i,j}\, b_{i,j},\ \forall i,j,\ \ C(m\times n)$$

È il prodotto "*element-wise*" `.*` di MATLAB `C=A.*B`

Prodotto di Kronecker di due vettori o matrici

Date due matrici $\boldsymbol{A}(m\times n)$ e $\boldsymbol{B}(p\times q)$, allora

$$C = A\otimes B \quad \overset{\text{def}}{\Longleftrightarrow} \quad c_{i,j} = a_{i,j}\, B,\ \ \forall i,j,\ \ C(mp\times nq)$$

È la funzione `kron()` di MATLAB `C=kron(A,B)`

Prodotto esterno di due vettori

Dati due vettori colonna $\boldsymbol{u}$ e $\boldsymbol{v}$ di lunghezza rispettiva m ed n (considerati come particolari matrici), allora il loro ***prodotto esterno*** è dato dal prodotto righe×colonne di $\boldsymbol{u}$ (vettore colonna) per $\boldsymbol{v}^{\mathsf{T}}$ (vettore riga):

$$C = u\circ v = u\, v^{\mathsf{T}} \quad \overset{\text{def}}{\Longleftrightarrow}$$

$\boldsymbol{C}$ è una matrice di size $(m\times n)$ avente rango 1: $c_{ij} = u_i\, v_j$

Operazioni su matrici e vettori in MATLAB

prodotto scalare

```
u=[2 1 1 -2]';
v=[-1 1 2 -2]';
a=u'*v
a =
     5
a=dot(u,v)
a =
     5
a=norm(u)*norm(v)*cos(subspace(u,v))
a =
     5
```

norma-2

angolo

$\|u\|_2\|v\|_2\cos\theta$, θ angolo tra i vettori

prodotto mat×vet

```
A=[ 1  2 -1 0;
   -1 -2  1 1];
u=[2 1 1 -2]';
v=A*u
v =
     3
    -5
```

prodotto mat×mat

```
A=[ 1  2 -1;
   -1 -2  1];
B=[2  1;
   1 -2;
   1 -1];
C=A*B
C =
     3    -2
    -3     2
```

```
u=sym('u',[3 1],'real');
v=sym('v',[3 1],'real');
a=dot(u,v)
a =
u1*v1 + u2*v2 + u3*v3
```

```
A=sym('a',[2 3],'real');
u=sym('u',[3 1],'real');
v=A*u
v =
a1_1*u1 + a1_2*u2 + a1_3*u3
a2_1*u1 + a2_2*u2 + a2_3*u3
```

prodotto di Hadamard ("*element-wise*")

```
u=sym('u',[4 1],'real');
v=sym('v',[4 1],'real');
w=u.*v
w =
u1*v1
u2*v2
u3*v3
u4*v4
```

```
A=sym('a',[2 4],'real');
B=sym('b',[2 4],'real');
C=A.*B
C =
[a1_1*b1_1, a1_2*b1_2, a1_3*b1_3, a1_4*b1_4]
[a2_1*b2_1, a2_2*b2_2, a2_3*b2_3, a2_4*b2_4]
```

Operazioni su matrici e vettori in MATLAB

prodotto di Kronecker di due matrici

```
A=sym('a',[2 3],'real');  B=sym('b',[4 2],'real');
C=kron(A,B)
```

$a_{11}\,B$ $a_{12}\,B$ $a_{13}\,B$

```
C =
[a1_1*b1_1, a1_1*b1_2, a1_2*b1_1, a1_2*b1_2, a1_3*b1_1, a1_3*b1_2]
[a1_1*b2_1, a1_1*b2_2, a1_2*b2_1, a1_2*b2_2, a1_3*b2_1, a1_3*b2_2]
[a1_1*b3_1, a1_1*b3_2, a1_2*b3_1, a1_2*b3_2, a1_3*b3_1, a1_3*b3_2]
[a1_1*b4_1, a1_1*b4_2, a1_2*b4_1, a1_2*b4_2, a1_3*b4_1, a1_3*b4_2]
[a2_1*b1_1, a2_1*b1_2, a2_2*b1_1, a2_2*b1_2, a2_3*b1_1, a2_3*b1_2]
[a2_1*b2_1, a2_1*b2_2, a2_2*b2_1, a2_2*b2_2, a2_3*b2_1, a2_3*b2_2]
[a2_1*b3_1, a2_1*b3_2, a2_2*b3_1, a2_2*b3_2, a2_3*b3_1, a2_3*b3_2]
[a2_1*b4_1, a2_1*b4_2, a2_2*b4_1, a2_2*b4_2, a2_3*b4_1, a2_3*b4_2]
```

$a_{21}\,B$ $a_{22}\,B$ $a_{23}\,B$

Il prodotto di Kronecker non è commutativo in generale

```
C=kron(B,A)
```

$b_{11}\,A$ $b_{12}\,A$

```
C =
[a1_1*b1_1, a1_2*b1_1, a1_3*b1_1, a1_1*b1_2, a1_2*b1_2, a1_3*b1_2]
[a2_1*b1_1, a2_2*b1_1, a2_3*b1_1, a2_1*b1_2, a2_2*b1_2, a2_3*b1_2]
[a1_1*b2_1, a1_2*b2_1, a1_3*b2_1, a1_1*b2_2, a1_2*b2_2, a1_3*b2_2]
[a2_1*b2_1, a2_2*b2_1, a2_3*b2_1, a2_1*b2_2, a2_2*b2_2, a2_3*b2_2]
[a1_1*b3_1, a1_2*b3_1, a1_3*b3_1, a1_1*b3_2, a1_2*b3_2, a1_3*b3_2]
[a2_1*b3_1, a2_2*b3_1, a2_3*b3_1, a2_1*b3_2, a2_2*b3_2, a2_3*b3_2]
[a1_1*b4_1, a1_2*b4_1, a1_3*b4_1, a1_1*b4_2, a1_2*b4_2, a1_3*b4_2]
[a2_1*b4_1, a2_2*b4_1, a2_3*b4_1, a2_1*b4_2, a2_2*b4_2, a2_3*b4_2]
```

$b_{21}\,A$ $b_{22}\,A$

$b_{31}\,A$ $b_{32}\,A$

$b_{41}\,A$ $b_{42}\,A$

Operazioni su matrici e vettori in MATLAB

prodotto di Kronecker di due vettori

```
u=sym('u',[3 1],'real');
v=sym('v',[2 1],'real');
W=kron(u,v)
W =
u1*v1
u1*v2
u2*v1
u2*v2
u3*v1
u3*v2
```

```
W=kron(v,u)
W =
u1*v1
u2*v1
u3*v1
u1*v2
u2*v2
u3*v2
```

non commutativo

```
W=kron(u',v)
W =
[u1*v1, u2*v1, u3*v1]
[u1*v2, u2*v2, u3*v2]
```

```
W=kron(v,u')
W =
[u1*v1, u2*v1, u3*v1]
[u1*v2, u2*v2, u3*v2]
```

commutativo

```
W=kron(u,v')
W =
[u1*v1, u1*v2]
[u2*v1, u2*v2]
[u3*v1, u3*v2]
```

```
W=kron(v',u)
W =
[u1*v1, u1*v2]
[u2*v1, u2*v2]
[u3*v1, u3*v2]
```

prodotto esterno di due vettori

```
u=sym('u',[3 1],'real');
v=sym('v',[2 1],'real');
W=u*v'
W =
[u1*v1, u1*v2]
[u2*v1, u2*v2]
[u3*v1, u3*v2]
```

la matrice che si ottiene dal prodotto esterno di due vettori ha sempre rango 1

```
u=sym('u',[3 1],'real');
v=sym('v',[2 1],'real');
W=v*u'
W =
[u1*v1, u2*v1, u3*v1]
[u1*v2, u2*v2, u3*v2]
```

```
disp(rank(W))
1
```

Sistemi di equazioni lineari

Esempio 1: Si voglia risolvere il sistema

$$\begin{cases} 2x - y = 0 \\ -x + 2y = 3 \end{cases} \iff Au=b$$

forma matriciale del sistema

dove

$$A=\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$$

matrice dei coefficienti

$$b=\begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

vettore dei termini noti

dati del problema

incognite del problema $\rightarrow$ $u=\begin{pmatrix} x \\ y \end{pmatrix}$ $\rightarrow$ **soluzione del problema**

rappresentazione grafica del problema

$$\begin{cases} 2x - y = 0 \\ -x + 2y = 3 \end{cases}$$

Geometria analitica

intersezione di due rette

considera i dati organizzati per righe

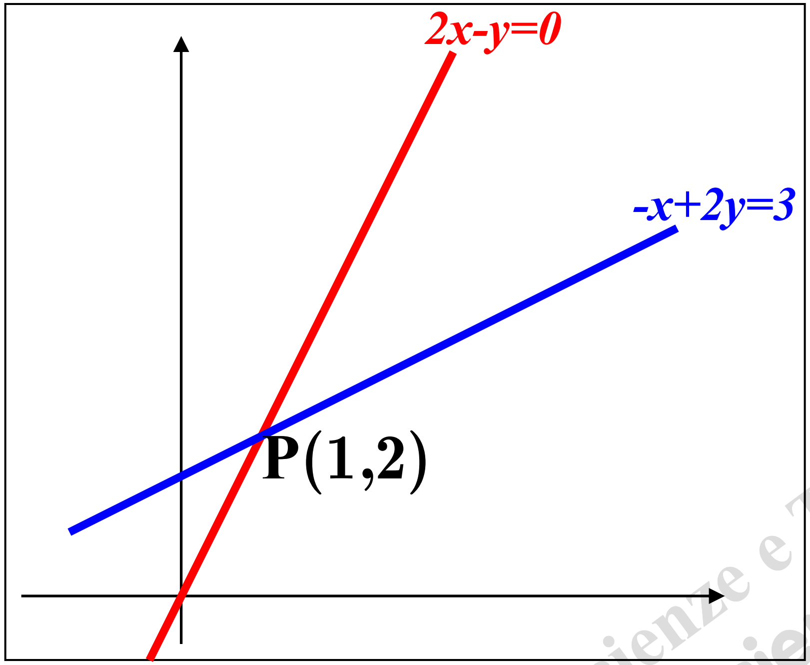
$$\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

Algebra dei vettori

combinazione lineare delle colonne della matrice che restituisce il termine noto

considera i dati organizzati per colonne

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



La Geometria analitica consente di interpretare ciascuna equazione del sistema come una retta del piano e, pertanto, risolvere il sistema significa trovare l'eventuale punto di intersezione delle due rette.

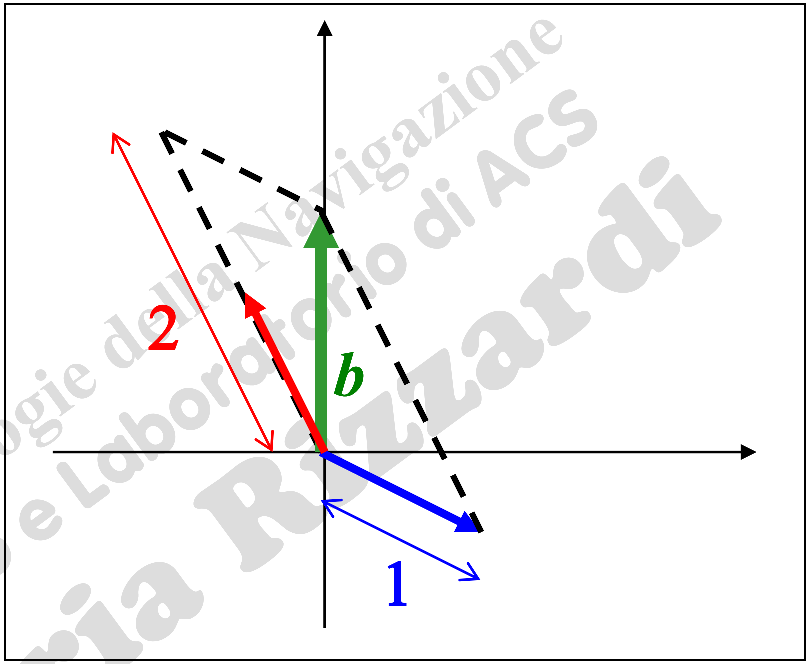
In questo caso le due rette si intersecano nel punto **P** e, pertanto il sistema ammette un'unica soluzione: solo i valori

$$x=1 \quad \text{ed} \quad y=2$$

soddisfano contemporaneamente le due equazioni del sistema.

L’algebra dei vettori (Algebra Lineare) consente di interpretare il sistema come l’eventuale decomposizione del vettore $\boldsymbol{b}$ in una combinazione lineare dei due vettori colonna di $\boldsymbol{A}$:

$$\boldsymbol{b} = \begin{pmatrix} 0 \\ 3 \end{pmatrix} = \boldsymbol{x} \begin{pmatrix} 2 \\ -1 \end{pmatrix} + \boldsymbol{y} \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$



In generale risulta:

$$A \;\; m\times n \; , \; u \;\; n\times 1 \quad \Rightarrow \quad Au = \sum_{k=1}^{n} u_k A_{:,k}$$

combinazione lineare delle colonne di A
con scalari dati dalle componenti di u

Dalla figura si vede che solo sommando (con la regola del parallelogramma)

il vettore $\begin{pmatrix} 2 \\ -1 \end{pmatrix}$ **ed il doppio del vettore** $\begin{pmatrix} -1 \\ 2 \end{pmatrix}$

si ottiene come risultante il vettore $\boldsymbol{b}$.

Sistemi di equazioni lineari

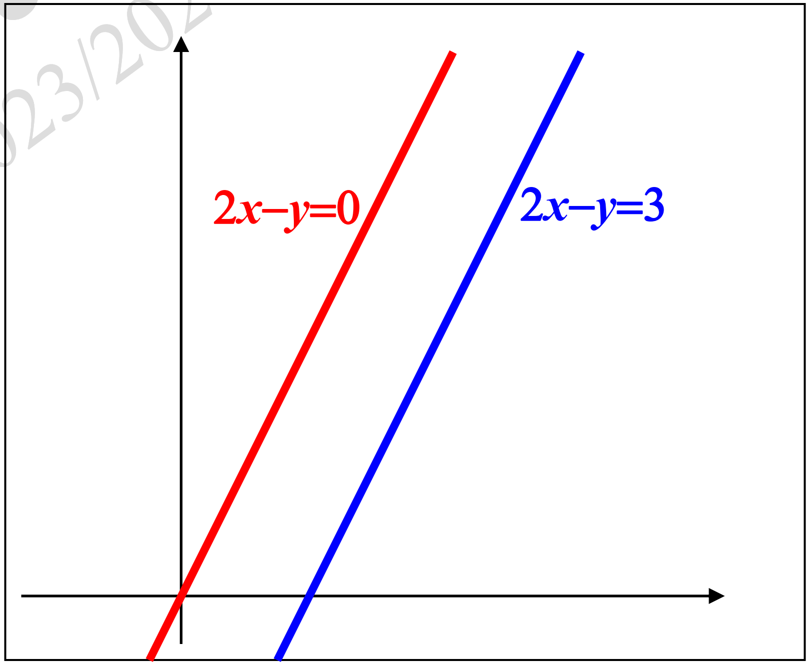
Esempio 2: Si voglia risolvere il sistema

$$\begin{cases} 2x - y = 0 \\ 2x - y = 3 \end{cases} \quad \Longleftrightarrow \quad Au = b$$

dove $A = \begin{pmatrix} 2 & -1 \\ 2 & -1 \end{pmatrix} \quad b = \begin{pmatrix} 0 \\ 3 \end{pmatrix} \quad [A\ b] = \begin{pmatrix} 2 & -1 & 0 \\ 2 & -1 & 3 \end{pmatrix}$

Le righe e colonne di A sono proporzionali; quelle di $[A\ b]$ no!

In questo caso le due rette sono **parallele** e, pertanto il sistema non ammette nessuna soluzione.



Sistemi di equazioni lineari

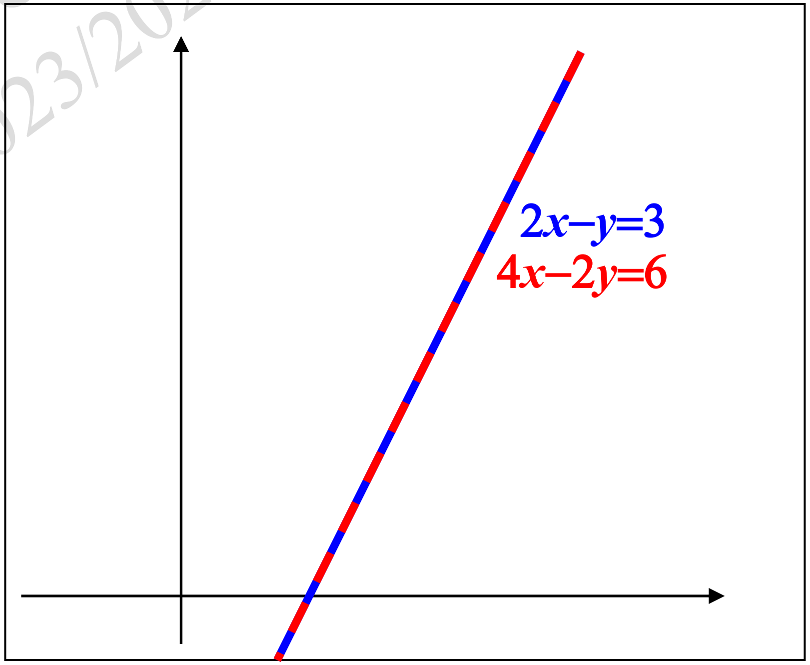
Esempio 3: Si voglia risolvere il sistema

$$\begin{cases} 4x - 2y = 6 \\ 2x - \ \ y = 3 \end{cases} \iff Au=b$$

dove $A = \begin{pmatrix} 4 & -2 \\ 2 & -1 \end{pmatrix}$ $\quad b = \begin{pmatrix} 6 \\ 3 \end{pmatrix}$ $\quad [A\ b] = \begin{pmatrix} 4 & -2 & 6 \\ 2 & -1 & 3 \end{pmatrix}$

In A ed in $[A\ b]$ le righe e le colonne sono proporzionali!

In questo caso le due rette sono **sovrapposte** (*impropriamente parallele*) e, pertanto, il sistema ammette infinite soluzioni.



In generale, un sistema $\boldsymbol{Ax=b}$ di $\boldsymbol{m}$ equazioni in $\boldsymbol{n}$ incognite è:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n = b_3 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots + a_{mn}x_n = b_m \end{cases}$$

dove

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

vettore incognite

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & & \ddots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{pmatrix}$$

matrice dei coefficienti

$$\boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

vettore dei termini noti

Sistema Lineare $Ax=b$

compatibile

(***ammette soluzione***)

incompatibile

(***non ammette soluzione***)

determinato

(***unica soluzione***)

indeterminato

(***infinite soluzioni***)

Sistemi di equazioni lineari

Richiami — in MATLAB...

Un sistema $Ax=b$ può essere

- **compatibile** se $rango(A)=rango([A,b])$ ⟷ `rank(A)==rank([A b])`
- **incompatibile** *altrimenti*.

Teor. di Rouché-Capelli

Il sistema è compatibile se,e solo se, il rango della *matrice completa* $[A\ b]$ è uguale a quello della *matrice incompleta* A.

Un sistema $Ax=b$ **compatibile** può essere

- **determinato** se
  $|A| \neq 0$ (A quadrata)
  o se $r(A)=n$ A rettangolare $(m\times n)$ $m>n$
- **indeterminato**
  $|A| = 0$ (A quadrata)
  oppure A rettangolare

⟷

`det(A) ~= 0` $A(n\times n)$

`rank(A) == n` $A(m\times n)$

$|A|$: determinante di una matrice quadrata

Risoluzione di sistemi lineari determinati
(richiami: il **metodo di Gauss**)

Il **metodo di Gauss** sostituisce al sistema lineare dato un altro che sia:

- **equivalente** (*) a quello di partenza (cioè che abbia la stessa soluzione);
- **più facile da risolvere** (**);

e risolve quest'ultimo.

(*) Un sistema di equazioni lineari si trasforma in uno **equivalente** se:

- si scambiano due equazioni;
- si sostituisce un'equazione con un'altra ottenuta moltiplicandola per un numero non nullo oppure ottenuta come somma algebrica di due equazioni del sistema.

(**) Sistemi lineari **più facili** da risolvere

- ***Sistemi diagonali***
- ***Sistemi triangolari***

Risoluzione di sistemi diagonali e triangolari

Sistema diagonale $Ax = b$

$$\begin{pmatrix} a_{1,1} & 0 & \cdots & 0 & 0 \\ 0 & a_{2,2} & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & a_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Esempio

$$\begin{cases} -2x_1 & = 1 \\ -1x_2 & = -2 \\ -2x_3 & = 4 \\ 3x_4 & = -9 \end{cases} \Rightarrow \begin{aligned} x_1 &= -1/2 \\ x_2 &= 2 \\ x_3 &= -2 \\ x_4 &= -3 \end{aligned}$$

Algoritmo $\Rightarrow$ $\forall i = 1,2,\ldots,n \quad x_i = \dfrac{b_i}{a_{i,i}}$

forma scalare

```
for i=1:n
    x(i)=b(i)/A(i,i);
end
```

forma vettoriale

```
x = b./diag(A);
```

Risoluzione di sistemi diagonali e triangolari

Sistema triangolare inferiore $Ax = b$

Procede "in avanti":
dalla 1ª eq. si ricava x_1, che può essere sostituito nelle equazioni che seguono;
dalla 2ª eq. si ricava x_2, che può essere sostituito nelle equazioni che seguono;
...

$$\begin{pmatrix} a_{1,1} & 0 & \cdots & 0 & 0 \\ a_{2,1} & a_{2,2} & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n-1} & a_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Esempio

$$\begin{cases} -2x_1 & & & & = & 1 \\ 2x_1 & -1x_2 & & & = & -2 \\ & -2x_2 & -2x_3 & & = & 4 \\ 1x_1 & +2x_2 & +1x_3 & -5x_4 & = & -9 \end{cases} \Rightarrow \begin{cases} x_1 = 1/\ -2 \\ x_2 = \ -2-2x_1\ /\ -1 \\ x_3 = \ 4+0x_1+2x_2\ /\ -2 \\ x_4 = \ -9-1x_1-2x_2-1x_3\ /\ -5 \end{cases} \Rightarrow \begin{cases} x_1 = -0.5 \\ x_2 = 1 \\ x_3 = -1 \\ x_4 = 1.9 \end{cases}$$

Algoritmo

Forward Substitution

$$\Rightarrow \begin{cases} x_1 = b_1 / a_{1,1} \\ x_2 = \ b_2 - a_{2,1}x_1\ / a_{2,2} \\ x_3 = \ b_3 - a_{3,1}x_1 - a_{3,2}x_2\ / a_{3,3} \\ \vdots \\ x_n = \left(b_n - \sum_{j=1}^{n-1} a_{n,j}x_j \right) \Big/ a_{n,n} \end{cases}$$

Algoritmo Forward Substitution

$$\begin{cases} x_1 = b_1 / a_{1,1} \\ x_2 = \left(b_2 - a_{2,1} x_1 \right) / a_{2,2} \\ x_3 = \left(b_3 - a_{3,1} x_1 - a_{3,2} x_2 \right) / a_{3,3} \\ \quad \vdots \\ x_n = \left(b_n - \sum_{j=1}^{n-1} a_{n,j} x_j \right) \Big/ a_{n,n} \end{cases}$$

forma scalare

vers. 1

```
function x = forwardSubst_scal1(L,b,n)
    x=zeros(n,1);
    for i=1:n
        x(i)=b(i)/L(i,i);
        for j=1:i-1
            x(i)=x(i) - L(i,j)*x(j)/L(i,i);
        end
    end
end
```

vers. 2

```
function x = forwardSubst_scal2(L,b,n)
    x=zeros(n,1);
    for i=1:n
        x(i)=b(i);
        for j=1:i-1
            x(i)=x(i) - L(i,j)*x(j);
        end
        x(i)=x(i)/L(i,i);
    end
end
```

script chiamante

```
n=…; A=rand(n);
A=tril(A); % triangolare inferiore
b=rand(n,1); x_matlab=A\b;
x_scal1=forwardSubst_scal1(A,b,n);
x_scal2=forwardSubst_scal2(A,b,n);
x_vect=forwardSubst_vect(A,b,n);
disp(max(abs(x_scal1-x_matlab)))
disp(max(abs(x_scal2-x_matlab)))
disp(max(abs(x_vect-x_matlab)))
```

```
n=25;
x_scal1: max(Abs Err) =    2.3283e-10
x_scal2: max(Abs Err) =      0
x_vect : max(Abs Err) =      0
```

meno accurato

```
n=55;
x_scal1: max(Abs Err) =    3.2425e-05
x_scal2: max(Abs Err) =      0
x_vect : max(Abs Err) =      0
```

forma vettoriale

```
function x = forwardSubst_vect(L,b,n)
    x=zeros(n,1);
    x(1)=b(1)/L(1,1);
    for i=2:n
        b(i:n)=b(i:n)-L(i:n,i-1)*x(i-1);
        x(i)=b(i)/L(i,i);
    end
end
```

Risoluzione di sistemi diagonali e triangolari

Sistema triangolare superiore $Ax = b$

Procede "all'indietro":
dalla n^{sima} eq. si ricava x_n, che può essere sostituito nelle equazioni che precedono;
dalla $(n-1)^{\text{sima}}$ eq. si ricava x_{n-1}, che può essere sostituito nelle equazioni che precedono;
…

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n} \\ 0 & a_{2,2} & \cdots & a_{2,n-1} & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & a_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Esempio

$$\begin{cases} 1x_1 & +2x_2 & +1x_3 & -5x_4 & = & 5 \\ & -2x_2 & -2x_3 & 1x_4 & = & -4 \\ & & 2x_3 & -1x_4 & = & -2 \\ & & & -2x_4 & = & 1 \end{cases} \Longrightarrow \begin{cases} x_4 = 1/\ -2 \\ x_3 = \ -2+1x_4\ /\ 2 \\ x_2 = \ -4-1x_4+2x_3\ /\ -2 \\ x_1 = \ 5+5x_4-1x_3-2x_2\ /\ 1 \end{cases} \Longrightarrow \begin{cases} x_4 = -0.5 \\ x_3 = -1.25 \\ x_2 = 3 \\ x_1 = -2.25 \end{cases}$$

Algoritmo

Backward Substitution

$$\begin{cases} x_n = b_n / a_{n,n} \\ x_{n-1} = \ b_{n-1} - a_{n-1,n}x_n\ / a_{n-1,n-1} \\ x_{n-2} = \ b_3 - a_{3,1}x_1 - a_{3,2}x_2\ / a_{3,3} \\ \vdots \\ x_1 = \left(b_1 - \sum_{j=2}^{n} a_{1,j}x_j \right) \Big/ a_{1,1} \end{cases}$$

Algoritmo backward Substitution

$$
\begin{cases}
x_n = b_n / a_{n,n} \\
x_{n-1} = \;\; b_{n-1} - a_{n-1,n} x_n \;\; / a_{n-1,n-1} \\
x_{n-2} = \;\; b_3 - a_{3,1} x_1 - a_{3,2} x_2 \;\; / a_{3,3} \\
\qquad \vdots \\
x_1 = \left(b_1 - \sum_{j=2}^{n} a_{1,j} x_j \right) \Big/ a_{1,1}
\end{cases}
$$

Esercizio

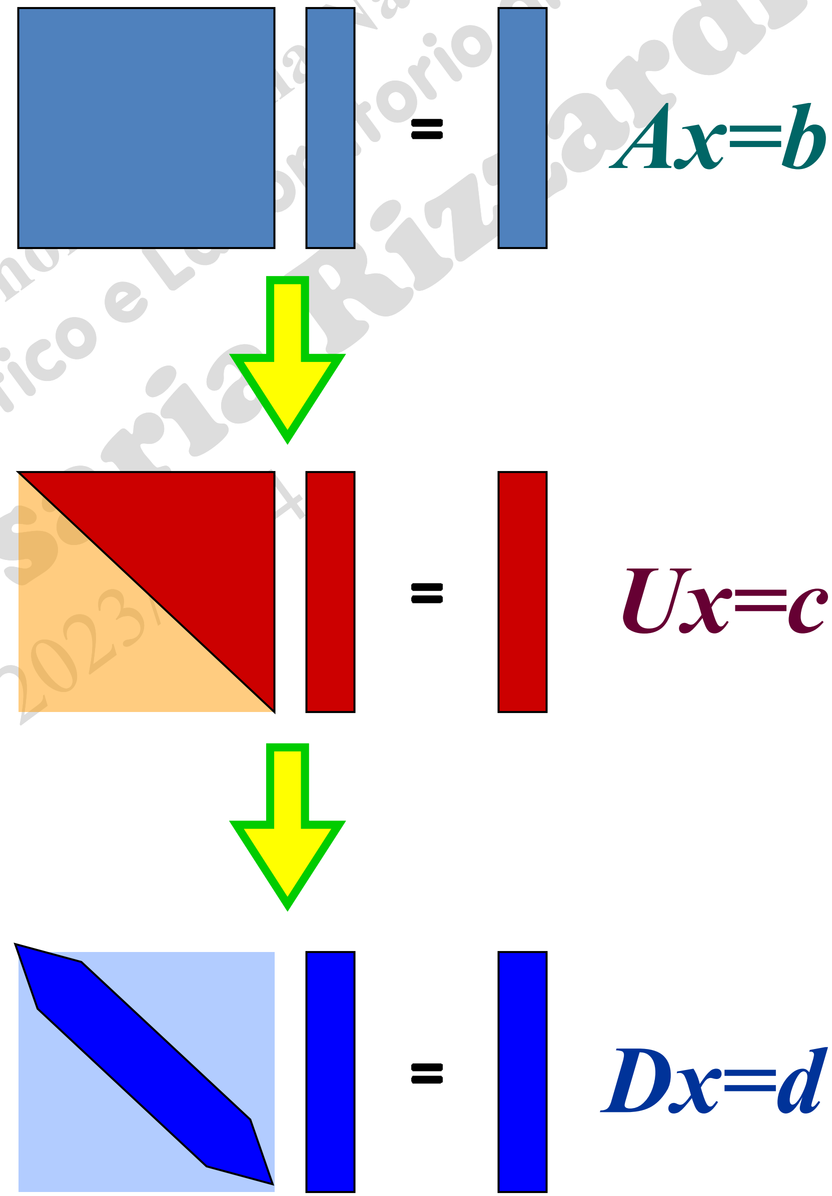
Analogamente al caso dell'algoritmo *Forward Substitution*, scrivere le funzioni MATLAB per implementare le due versioni scalari e quella vettoriale dell'algoritmo *Backward Substitution*

Richiami: metodo di eliminazione di Gauss

caso matrice quadrata
non singolare

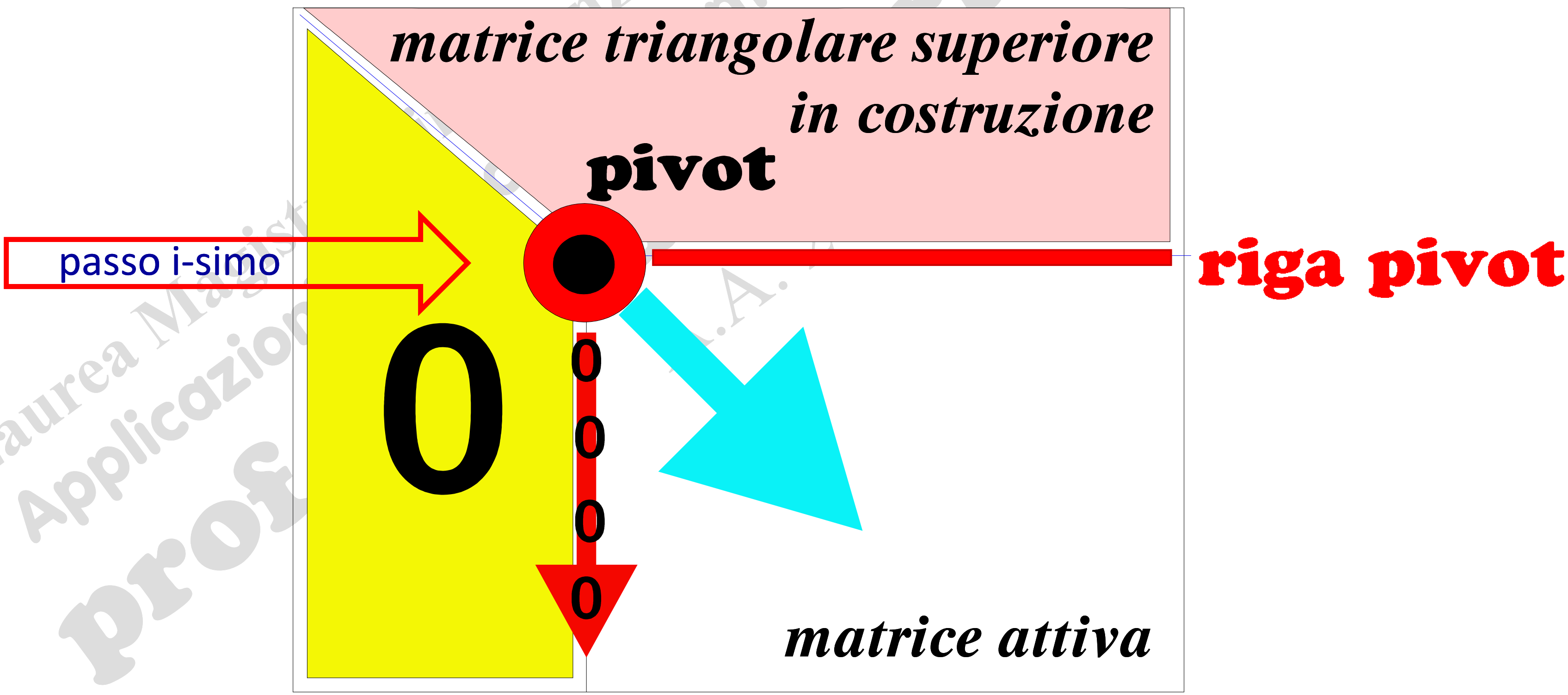
Trasforma il sistema dato in uno **triangolare superiore** equivalente

Risolve il sistema triangolare superiore mediante l'**algoritmo di backward substitution**



Richiami: metodo di eliminazione di Gauss

Il **metodo di eliminazione di Gauss** (detto metodo di Gauss per righe in avanti $G^{\downarrow}$ o "forward sweep") procede per passi modificando le righe al di sotto della **riga pivot** in modo tale da introdurre zeri al di sotto dell'**elemento pivot**.



Esempio: metodo di eliminazione di Gauss

$Ax=b$

$$\begin{cases} 2x_1 + x_2 + x_3 = 1 \\ 4x_1 + x_2 = -2 \\ 2x_1 - 2x_2 - x_3 = -7 \end{cases}$$

$$\begin{array}{cc} A & b \\ \begin{pmatrix} 2 & 1 & 1 \\ 4 & 1 & 0 \\ 2 & -2 & -1 \end{pmatrix} & \begin{pmatrix} 1 \\ -2 \\ -7 \end{pmatrix} \end{array}$$

1° passo: elemento pivot = $a_{1,1}$

- Calcola il moltiplicatore

$$m_{2,1} = 4/2 = 2.$$

- Sottrae dalla seconda riga la prima (***riga pivot***) moltiplicata per $m_{2,1}$.

- Calcola il moltiplicatore

$$m_{3,1} = 2/2 = 1.$$

- Sottrae dalla terza riga la prima (***riga pivot***) moltiplicata per $m_{3,1}$.

$$\begin{cases} 2x_1 + x_2 + x_3 = 1 \\ -x_2 - 2x_3 = -4 \\ -3x_2 - 2x_3 = -8 \end{cases}$$

$$\begin{cases} 2x_1 + x_2 + \ x_3 = \ \ 1 \\ \quad -1x_2 - 2x_3 = -4 \\ \quad -3x_2 - 2x_3 = -8 \end{cases} \qquad \begin{pmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & -3 & -2 \end{pmatrix} \begin{pmatrix} 1 \\ -4 \\ -8 \end{pmatrix}$$

2° passo: elemento pivot = $a_{2,2}$

- Calcola il moltiplicatore

$$m_{3,2} = -3/(-1) = 3.$$

- Sottrae dalla terza riga la seconda (***riga pivot***) moltiplicata per $m_{3,2}$.

$$\begin{cases} 2x_1 + x_2 + \ x_3 = \ \ 1 \\ \quad -1x_2 - 2x_3 = -4 \\ \qquad\qquad 4x_3 = \ \ 4 \end{cases}$$

Sistema triangolare superiore

$$\begin{pmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ -4 \\ 4 \end{pmatrix}$$

Backward substitution

...

Richiami: forma vettoriale del metodo di Gauss

operazione fondamentale: $\text{riga}_{\text{nuova}} = \text{riga}_{\text{vecchia}} - \text{moltiplicatore} \times \text{riga}_{\text{pivot}}$

Metodo di Gauss in avanti (G↓ forward sweep)

$$\begin{cases} 2x_1 + x_2 + x_3 = 1 \\ 4x_1 + x_2 = -2 \\ 2x_1 - 2x_2 - x_3 = -7 \end{cases}$$

```
A=[2 1 1;4 1 0;2 -2 -1];
b=[1 -2 -7]';  M=[A b];
L=eye(3);
```

$$M = \left[\begin{array}{ccc|c} 2 & 1 & 1 & 1 \\ 4 & 1 & 0 & -2 \\ 2 & -2 & -1 & -7 \end{array}\right]$$

1° *passo* $\Rightarrow$ $\text{riga}_{\text{pivot}} = \text{riga}_1$; $\text{elemento}_{\text{pivot}} = a_{11}$

```
rpiv=M(1,:);  piv=M(1,1);
L(2,1)=M(2,1)/piv;  moltiplicatore
M(2,:)=M(2,:) - L(2,1)*rpiv;
L(3,1)=M(3,1)/piv;  moltiplicatore
M(3,:)=M(3,:) - L(3,1)*rpiv;
```

$$L = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ 1 & & 1 \end{bmatrix} \quad M = \left[\begin{array}{ccc|c} 2 & 1 & 1 & 1 \\ 0 & -1 & -2 & -4 \\ 0 & -3 & -2 & -8 \end{array}\right]$$

2° *passo* $\Rightarrow$ $\text{riga}_{\text{pivot}} = \text{riga}_2$; $\text{elemento}_{\text{pivot}} = a_{22}$

```
rpiv=M(2,:);  piv=M(2,2);
L(3,2)=M(3,2)/piv;  moltiplicatore
M(3,:)=M(3,:) - L(3,2)*rpiv;
```

$$L = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ 1 & 3 & 1 \end{bmatrix} \quad M = \left[\begin{array}{ccc|c} 2 & 1 & 1 & 1 \\ 0 & -1 & -2 & -4 \\ 0 & 0 & 4 & 4 \end{array}\right]$$

Richiami: forma vettoriale del metodo di Gauss

Miglioramento MATLAB **G**↓ vettoriale

$1°$ *passo* $\Rightarrow$ $\text{riga}_{\text{pivot}} = \text{riga}_1$; $\text{elemento}_{\text{pivot}} = a_{11}$

```
p=1;
rpiv=M(p,p:end);  piv=M(p,p);
L(2,p)=M(2,p)/piv;
M(2,p:end)=M(2,p:end) - L(2,p)*rpiv;
L(3,p)=M(3,p)/piv;
M(3,p:end)=M(3,p:end) - L(3,p)*rpiv;
```

$$L = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ 1 & & 1 \end{bmatrix} \quad M = \left[\begin{array}{ccc|c} 2 & 1 & 1 & 1 \\ 0 & -1 & -2 & -4 \\ 0 & -3 & -2 & -8 \end{array}\right]$$

$2°$ *passo* $\Rightarrow$ $\text{riga}_{\text{pivot}} = \text{riga}_2$; $\text{elemento}_{\text{pivot}} = a_{22}$

```
p=2;
rpiv=M(p,p:end);  piv=M(p,p);
L(3,p)=M(3,p)/piv;
M(3,p:end)=M(3,p:end) - L(3,p)*rpiv;
```

$$L = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ 1 & 3 & 1 \end{bmatrix} \quad M = \left[\begin{array}{ccc|c} 2 & 1 & 1 & 1 \\ 0 & -1 & -2 & -4 \\ 0 & 0 & 4 & 4 \end{array}\right]$$

(…,p:end): lavora solo sulla matrice attiva evitando di sommare inutilmente elementi nulli

Richiami: forma vettoriale del metodo di Gauss

operazione fondamentale: $\mathbf{riga}_{\mathbf{nuova}} = \mathbf{riga}_{\mathbf{vecchia}} - \mathbf{moltiplicatore} \times \mathbf{riga}_{\mathbf{pivot}}$

Metodo di Gauss all'indietro (G↑ reverse sweep)

$$M = \left[\begin{array}{ccc|c} 2 & 1 & 1 & 1 \\ 0 & -1 & -2 & -4 \\ 0 & 0 & 4 & 4 \end{array}\right]$$

L'algoritmo di backward substitution può essere considerato come *metodo di Gauss all'indietro*, che inserisce zeri al di sopra della diagonale principale, partendo dall'ultima riga del sistema triangolare superiore.
Il sistema triangolare superiore viene così trasformato in un sistema diagonale.

1° *passso* $\Rightarrow$ $\mathbf{riga}_{\mathbf{pivot}} = \mathbf{riga}_3;\ \ \mathbf{elemento}_{\mathbf{pivot}} = a_{33}$

```
p=3;
rpiv=M(p,p:end);  piv=M(p,p);
L(2,p)=M(2,p)/piv;
M(2,p:end)=M(2,p:end) - L(2,p)*rpiv;
L(1,p)=M(1,p)/piv;
M(1,p:end)=M(1,p:end) - L(1,p)*rpiv;
```

$$L = \begin{bmatrix} 1 & & 0.25 \\ 2 & 1 & -0.5 \\ 1 & 3 & 1 \end{bmatrix} \qquad M = \left[\begin{array}{ccc|c} 2 & 1 & 0 & 0 \\ 0 & -1 & 0 & -2 \\ 0 & 0 & 4 & 4 \end{array}\right]$$

2° *passso* $\Rightarrow$ $\mathbf{riga}_{\mathbf{pivot}} = \mathbf{riga}_2;\ \ \mathbf{elemento}_{\mathbf{pivot}} = a_{22}$

```
p=2;
rpiv=M(p,p:end);  piv=M(p,p);
L(1,p)=M(1,p)/piv;
M(1,p:end)=M(1,p:end) - L(1,p)*rpiv;
```

$$L = \begin{bmatrix} 1 & -1 & 0.25 \\ 2 & 1 & -0.5 \\ 1 & 3 & 1 \end{bmatrix} \qquad M = \left[\begin{array}{ccc|c} 2 & 0 & 0 & -2 \\ 0 & -1 & 0 & -2 \\ 0 & 0 & 4 & 4 \end{array}\right]$$

sistema diagonale

Metodo di Gauss per sistemi tridiagonali (Algoritmo di Thomas)

$$Ax = d$$

$$A_{n\times n} = \begin{pmatrix} a_{1,1} & a_{1,2} & 0 & \cdots & \cdots & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & \cdots & 0 \\ 0 & a_{3,2} & a_{3,3} & a_{3,4} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & 0 & \cdots & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & \cdots & 0 & a_{n,n-1} & a_{n,n} \end{pmatrix}$$

L'algoritmo di Gauss può essere semplificato quando applicato a *sistemi tridiagonali*: in tal caso prende il nome di Algoritmo di Thomas.

$$A_{n\times n} = \begin{pmatrix} b_1 & c_1 & 0 & \cdots & \cdots & 0 \\ a_1 & b_2 & c_2 & \cdots & \cdots & 0 \\ 0 & a_2 & b_3 & c_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & 0 & \cdots & \cdots & b_{n-1} & c_{n-1} \\ 0 & 0 & \cdots & 0 & a_{n-1} & b_n \end{pmatrix}$$

La matrice tridiagonale è una matrice sparsa, quindi le tre diagonali possono essere memorizzate in tre vettori diversi, oppure nelle colonne di una matrice (inserendo uno zero nei vettori che non rappresentano la diagonale principale per renderli tutti di eguale lunghezza)

Metodo di Gauss per sistemi tridiagonali
(Algoritmo di Thomas) $O(n)$

$$A_{n\times n} = \begin{pmatrix} b_1 & c_1 & 0 & \cdots & \cdots & 0 \\ a_1 & b_2 & c_2 & \cdots & \cdots & 0 \\ 0 & a_2 & b_3 & c_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & 0 & \cdots & \cdots & b_{n-1} & c_{n-1} \\ 0 & 0 & \cdots & 0 & a_{n-1} & b_n \end{pmatrix}$$

$$Ax = d$$

Versione 1

1) Estrae i tre vettori dalla matrice tridiagonale

```
b=zeros(n,1); a=zeros(n-1,1); c=zeros(n-1,1);
for i=2:n,   a(i-1)=A(i,i-1);   end
for i=1:n,   b(i)=A(i,i);       end
for i=2:n,   c(i-1)=A(i-1,i);   end
```

2) Metodo di Gauss in avanti (Forward elimin.)

```
for i=2:n
    w=a(i-1)/b(i-1);
    b(i)=b(i) – w*c(i-1);
    d(i)=d(i) – w*d(i-1);
end
```

3) Backward substitution

```
x=zeros(n,1); % prealloca il vettore soluzione
x(n)=d(n)/b(n);
for i=n-1:-1:1
    x(i)=(d(i) – c(i)*x(i+1))/b(i);
end
```

Versione 2

1) Prealloca i vettori

```
b=zeros(n,1);
a=zeros(n-1,1);
c=zeros(n-1,1);
```

2) Forward elimination

```
b(1)=A(1,1);
for i=2:n
    a(i-1)=A(i,i-1);
    b(i)=A(i,i);
    c(i-1)=A(i-1,i);
    w=a(i-1)/b(i-1);
    b(i)=b(i) – w*c(i-1);
    d(i)=d(i) – w*d(i-1);
end
```

3) Backward substitution

```
x=zeros(n,1); % prealloca il vettore soluzione
x(n)=d(n)/b(n);
for i=n-1:-1:1
    x(i)=(d(i) – c(i)*x(i+1))/b(i);
end
```

Metodo di Gauss per sistemi tridiagonali (Algoritmo di Thomas)

$$Ax = d$$

$$A_{4\times 4} = \begin{pmatrix} 2 & 2 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

Se la matrice A è rappresentata come matrice sparsa, è inutile definire i tre vettori per le sue diagonali e l'algoritmo assume una forma più compatta.

```
a=[1 -1 1 0]'; b=[2 2 2 2]';
c=[0 2 1 -1]'; B=[a b c];
A=spdiags(B,[-1 0 1],4,4);
full(A)
ans =
     2     2     0     0
     1     2     1     0
     0    -1     2    -1
     0     0     1     2
```

```
A
A =
   (1,1)     2
   (2,1)     1
   (1,2)     2
   (2,2)     2
   (3,2)    -1
   (2,3)     1
   (3,3)     2
   (4,3)     1
   (3,4)    -1
   (4,4)     2
```

Versione 3

1) Forward elimination

```
for i=2:n
    w=A(i,i-1)/A(i-1,i-1);
    A(i,i)=A(i,i) - w*A(i-1,i);
    d(i)=d(i) - w*d(i-1);
end
```

2) Backward substitution

```
x=zeros(n,1); % prealloca il vettore soluzione
x(n)=d(n)/A(n,n);
for i=n-1:-1:1
    x(i)=(d(i) - A(i,i+1)*x(i+1))/A(i,i);
end
```

```
Algoritmo di Thomas v.1 su matrice piena 50 x 50
Elapsed time is 0.003912 seconds.
Confronto con soluzione xM=A\d: max(abs(xM-x1)) = 5.40012e-13

Algoritmo di Thomas v.2 su matrice piena 50 x 50
Elapsed time is 0.002892 seconds.
Confronto con soluzione xM=A\d: max(abs(xM-x2)) = 5.40012e-13

Algoritmo di Thomas v.3 su matrice piena 50 x 50
Elapsed time is 0.002341 seconds.
Confronto con soluzione xM=A\d: max(abs(xM-x3)) = 5.40012e-13
```

```
Algoritmo di Thomas v.1 su matrice sparsa 50 x 50
Elapsed time is 0.002269 seconds.
Confronto con soluzione xM=A\d: max(abs(xM-x1)) = 0

Algoritmo di Thomas v.2 su matrice sparsa 50 x 50
Elapsed time is 0.002576 seconds.
Confronto con soluzione xM=A\d: max(abs(xM-x2)) = 0

Algoritmo di Thomas v.3 su matrice sparsa 50 x 50
Elapsed time is 0.003728 seconds.
Confronto con soluzione xM=A\d: max(abs(xM-x3)) = 0
```

Fattorizzazioni di una matrice: LU

Se nel metodo di Gauss, applicato ad una matrice quadrata e non singolare A, si considerano le due matrici:

- L: contenente i moltiplicatori (triangolare inferiore);
- U: matrice finale (triangolare superiore) ottenuta da **G↓** (forward sweep).

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & 1 & 0 \\ 2 & -2 & -1 \end{bmatrix} \Rightarrow A^1 = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & -3 & -2 \end{bmatrix} \Rightarrow A^2 = U = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 4 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \dots & 1 & 0 \\ \dots & \dots & 1 \end{bmatrix}$$ dopo il 1° passo $L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ matrice finale dei moltiplicatori $L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix}$

si è ottenuta la cosiddetta fattorizzazione LU della matrice A:

$$A = L\,U$$

* Questa fattorizzazione è utile, tra l'altro, per risolvere il sistema: $Ax = b$.
  Infatti

$$Ax = b \Leftrightarrow LUx = b \Leftrightarrow \begin{array}{l} 1)\ Ly = b \\ 2)\ Ux = y \end{array}$$ due sistemi triangolari da risolvere

* Oppure per calcolare efficientemente il determinante della matrice A.
  Infatti

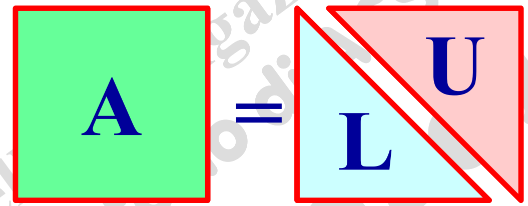
$$\det(A) = \det(U) = \text{prod}(\text{diag}(U))$$

Fattorizzazioni di una matrice: LU in MATLAB…

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

$A=LU$

$PA=LU$



$PP^T=P^TP=I$ matrice ortogonale
$\det(P)=\pm1$ ⟸ P matrice di permutazione ⟹ PA permuta le righe, AP permuta le colonne

Una matrice di permutazione è ottenuta dalla matrice identica applicandole gli scambi di righe e/o di colonne che si vogliono.

```
[L,U,P]=lu(A); disp(P*A)
     3     4     0
    -2     1     1
     1     0     2
```

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

```
P        matrice di permutazione
P  =     = P I
     0        0        1
     0        1        0
     1        0        0
```

$$I=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

Esempio: calcolo del determinante

```
A=[3 4 0;-2 1 1;1 0 2];
detA=det(A)
detA =
    26
[L,U]=lu(A);
disp(prod(diag(U)))
26
```

```
A=[1 0 2;-2 1 1;3 4 0];
detA=det(A)
detA =
   -26
[L,U,P]=lu(A);
disp(prod(diag(U))/det(P))
   -26
```

$\det(P)=\pm1$

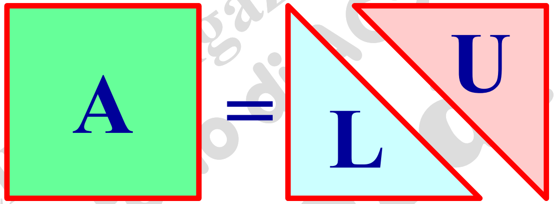
*

Fattorizzazioni di una matrice: LU in MATLAB…

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

A=LU

PA=LU



$PP^T=P^TP=I$ matrice ortogonale
det(P)=±1 ⇦ P matrice di permutazione ⇨ PA permuta le righe, AP permuta le colonne

Esempio: risoluzione di sistemi

```
A=[3 4 0;-2 1 1;1 0 2];
b=[11 3 7]';
[L,U]=lu(A) % Ax=b ⇔ LUx=b
L =
           1           0           0
    -0.66667           1           0
     0.33333    -0.36364           1
U =
           3           4           0
           0      3.6667           1
           0           0      2.3636
y=L\b; % ⇔ Ly=b
x=U\y  % ⇔ Ux=y
x =
           1
           2    stessa soluzione di A\b
           3
```

```
A=[1 0 2;-2 1 1;3 4 0];
b=[7 3 11]';
[L,U,P]=lu(A) % PAx=Pb ⇔ LUx=Pb
L =
           1           0           0
    -0.66667           1           0
     0.33333    -0.36364           1
U =
           3           4           0
           0      3.6667           1
           0           0      2.3636
P =
     0     0     1
     0     1     0
     1     0     0
y=L\(P*b); % ⇔ Ly=b
x=U\y      % ⇔ Ux=y
x =
           1
           2
           3
```

Fattorizzazioni di una matrice: LU

in MATLAB…

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

$$A=LU$$

$$PA=LU$$

↑ P matrice di permutazione

Esempi: A rettangolare

sistema indeterminato

$$A=\begin{bmatrix}-2 & 1 & 1\\ 1 & 0 & 2\end{bmatrix}$$

```
A=[-2 1 1;1 0 2];
b=[3 7]';
[L,U]=lu(A) % Ax=b ⇔ LUx=b
L =
       1     0
    -0.5     1
U =
    -2       1        1
     0     0.5      2.5
y=L\b; % ⇔ Ly=b
x=U\y  % ⇔ Ux=y
x =
           0.2
             0
           3.4
```

soluzione particolare

A = L U

$m<n$

sistema indeterminato

$$A=\begin{bmatrix}1 & 0 & 2\\ -2 & 1 & 1\end{bmatrix}$$

```
A=[1 0 2;-2 1 1];
b=[7 3]';
[L,U,P]=lu(A) % PAx=Pb ⇔ LUx=Pb
L =
       1     0
    -0.5     1
U =
    -2       1        1
     0     0.5      2.5
P =
       0     1
       1     0
y=L\(P*b); % ⇔ Ly=Pb
x=U\y      % ⇔ Ux=y
x =
          -2.2
             0
           2.6
```

soluzione particolare

P A = L U

$m>n$

$$A=\begin{bmatrix}-2 & 1\\ 1 & 0\\ 1 & 2\end{bmatrix}$$

```
A=[-2 1 1;1 0 2]';
[L,U,P]=lu(A)
L =
       1         0
    -0.5         1
    -0.5       0.2
U =
      -2         1
       0       2.5
P =
       1       0       0
       0       0       1
       0       1       0
```

L
U
P

Fattorizzazioni di una matrice: LU

Nel caso di matrice A simmetrica e definita positiva la fattorizzazione $A = LU$ diventa:

$$A = LL^{\mathsf{T}} = U^{\mathsf{T}}U \quad \text{(Algoritmo di Cholesky)}$$

la cui complessità computazionale $O(n^3/6)$ è la metà di quella di Gauss $O(n^3/3)$.

in MATLAB

```
A=rand(3);
A=tril(A);
A=A+A';
```

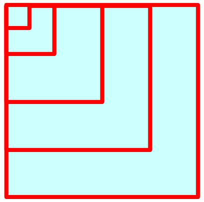
crea una matrice simmetrica

```
U=chol(A); max(max(U'*U-A))
ans = 2.2204e-16
```

$A(n\times n)$ **definita positiva**: sono equivalenti

- La forma quadratica $\boxed{x^{\mathsf{T}}Ax}$ è positiva: $x^{\mathsf{T}}Ax > 0,\ \forall x \neq \underline{0}$
- Tutti gli autovalori λ di A sono positivi: $\forall\lambda : Ax = \lambda x,\ \ \lambda > 0$
- Tutti i minori principali* sono positivi.

* I minori principali sono i determinanti delle sottomatrici centrate sulla diagonale principale



forma quadratica

```
N=2; A=sym('a',N,'real');
x=sym('x',[N 1],'real');
F=expand(x'*A*x); pretty(F)
         2           2
a1_1 x1  + a2_2 x2  + a1_2 x1 x2 + a2_1 x1 x2
```

Fattorizzazioni di una matrice: QR in MATLAB…

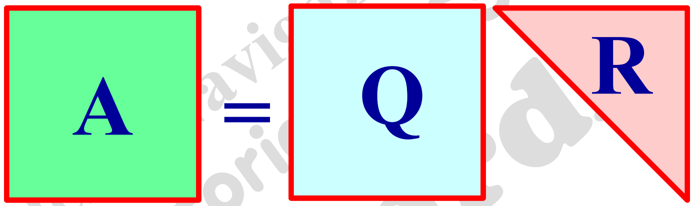
```
[Q,R]=qr(A)
[Q,R,P]=qr(A)
[Q,R]=qr(A,0)
```

$A=QR$

$AP=QR$

P matrice di permutazione

m=n



Q matrice ortogonale: $Q^TQ = QQ^T = I$

R matrice triangolare superiore

$AP=QR \Rightarrow A=QRP^{-1}=QRP^T$

Esempio: risoluzione di un sistema

```
A=[3 4 0;-2 1 1;1 0 2];
b=[11 3 7]';
[Q,R]=qr(A) % Ax=b ⇔ QRx=b
Q =
     -0.80178     -0.59152    -0.085126
      0.53452     -0.77353       0.3405
     -0.26726      0.22751      0.93638
R =
      -3.7417      -2.6726            0
            0      -3.1396     -0.31851
            0            0       2.2133
x=R\(Q'*b)  % ⇔ Rx=Q^Tb
x =
            1
            2
            3   stessa soluzione di A\b
```

```
A=[3 4 0;-2 1 1;1 0 2];
b=[11 3 7]';
[Q,R,P]=qr(A) % Ax=b ⇔ QRx=b
Q =
     -0.97014      0.22711    -0.085126
     -0.24254     -0.90842       0.3405
            0      0.35098      0.93638
R =
      -4.1231      -2.4254     -0.24254
            0       2.8491     -0.20646
            0            0       2.2133
P =
     0     1     0
     1     0     0
     0     0     1
% Ax=QRP^Tx=b ⇔ QRy=b,  dove y=P^Tx
y=R\(Q'*b);   x=P*y
x =
            1
            2
            3
```

Fattorizzazioni di una matrice: QR in MATLAB…

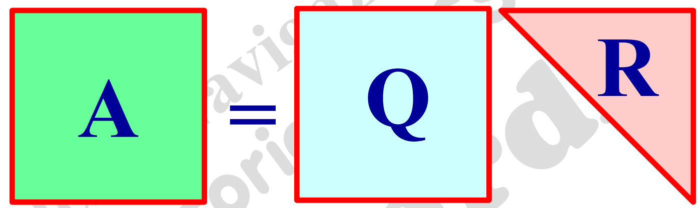
```
[Q,R]=qr(A)
[Q,R,P]=qr(A)
[Q,R]=qr(A,0)
```

$A=QR$

$AP=QR$

P matrice di permutazione

m=n



Q matrice ortogonale: $Q^TQ = QQ^T = I$

R matrice triangolare superiore

$AP=QR \Rightarrow A=QRP^{-1}=QRP^T$

Esempio: determinante della matrice

```
A=[3 4 0;-2 1 1;1 0 2];
[Q,R]=qr(A)
Q =
     -0.80178      -0.59152     -0.085126
      0.53452      -0.77353        0.3405
     -0.26726       0.22751       0.93638
R =
      -3.7417       -2.6726             0
            0       -3.1396      -0.31851
            0             0        2.2133
disp(det(A))
     26
% det(A)=det(QR)=det(Q)×det(R)=±det(R)
[prod(diag(R)) det(Q)]
ans =
           26             1
```

```
A=[3 4 0;-2 1 1;1 0 2];
[Q,R,P]=qr(A)
Q =
     -0.97014       0.22711     -0.085126
     -0.24254      -0.90842        0.3405
            0       0.35098       0.93638
R =
      -4.1231       -2.4254      -0.24254
            0        2.8491      -0.20646
            0             0        2.2133
P =
     0     1     0
     1     0     0
     0     0     1
% det(A)=det(QRPᵀ)=det(Q)×det(R)×det(Pᵀ)
[prod(diag(R)) det(Q) det(P)]
ans =
          -26             1            -1
```

Fattorizzazioni di una matrice: QR in MATLAB…

```
[Q,R]=qr(A)
[Q,R,P]=qr(A)
[Q,R]=qr(A,0)
```

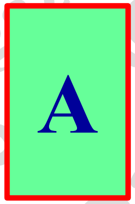
economy size

$A=QR$

$AP=QR$

P matrice di permutazione

$m>n$



Esempio: A rettangolare

```
A=[3 0;-2 1;1 0];
[Q,R]=qr(A)
Q =
   -0.80178     -0.50709     -0.31623
    0.53452     -0.84515   1.3878e-17
   -0.26726     -0.16903      0.94868
R =
    -3.7417      0.53452
          0     -0.84515
          0            0
```

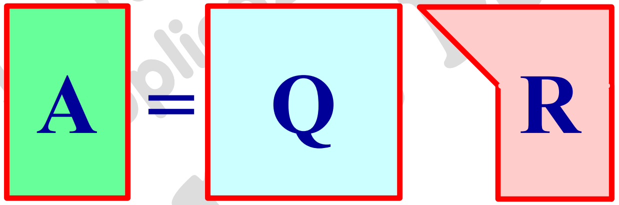
```
A=[3 0;-2 1;1 0];
[Q,R]=qr(A,0)
Q =
   -0.80178     -0.50709
    0.53452     -0.84515
   -0.26726     -0.16903
R =
    -3.7417      0.53452
          0     -0.84515
```

economy size

```
Q'*Q
ans =
      1.0000     -0.0000
      0.0000      1.0000
```

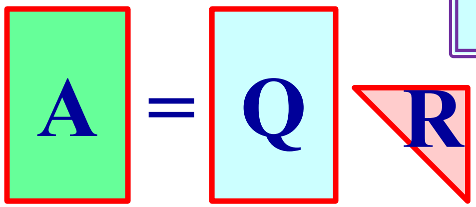
```
Q*Q'
ans =
      0.9000     -0.0000      0.3000
     -0.0000      1.0000     -0.0000
      0.3000     -0.0000      0.1000
```

$\neq \mathbf{I}$



Q matrice ortogonale $Q^TQ = QQ^T = I$

R matrice trapezoidale superiore



Q con colonne ortogonali $Q^TQ = I$

R matrice triangolare superiore

Fattorizzazioni di una matrice: QR in MATLAB…

```
[Q,R]=qr(A)
[Q,R,P]=qr(A)
[Q,R]=qr(A,0)
```

economy size

$A=QR$

$AP=QR$

P matrice di permutazione

$m<n$ **A**

Esempio: A rettangolare

```
A=[3 -2 1;0 1 0];
[Q,R]=qr(A)
Q =
     1     0
     0     1
R =
     3    -2     1
     0     1     0
```

```
A=[3 -2 1;0 1 0];
[Q,R]=qr(A,0)
Q =
     1     0
     0     1
R =
     3    -2     1
     0     1     0
```

economy size

A = **Q** **R**

Q matrice ortogonale $Q^TQ = QQ^T = I$

R matrice trapezoidale superiore

Fattorizzazioni di una matrice : SVD

in MATLAB…

Singular Value Decomposition

```
[U,S,V]=svd(A)
[U,S,V]=svd(A,0)
[U,S,V]=svd(A,"econ")
```

$$A=USV^T$$

$$m=n$$

$$\boxed{A} = \boxed{U}\,\boxed{S}\,\boxed{V^T}$$

U,V matrici ortogonali
S matrice diagonale contenente i valori singolari σ_k

Esempio: A quadrata

```
A=[1 2 1; 2 2 2;1 2 1];
[U,S,V]=svd(A) % equiv. a svd(A,0)
U =
   -0.5000    0.5000   -0.7071
   -0.7071   -0.7071   -0.0000
   -0.5000    0.5000    0.7071
S =
    4.8284         0         0
         0    0.8284         0
         0         0    0.0000
V =
   -0.5000   -0.5000    0.7071
   -0.7071    0.7071    0.0000
   -0.5000   -0.5000   -0.7071
```

```
A=[-2 1 1;1 0 2];
[U,S,V]=svd(A,"econ")
U =
   -0.5000    0.5000   -0.7071
   -0.7071   -0.7071   -0.0000
   -0.5000    0.5000    0.7071
S =
    4.8284         0         0
         0    0.8284         0
         0         0    0.0000
V =
   -0.5000   -0.5000    0.7071
   -0.7071    0.7071    0.0000
   -0.5000   -0.5000   -0.7071
```

```
U'*U
ans =
    1.0000   -0.0000    0.0000
   -0.0000    1.0000         0
    0.0000         0    1.0000
```

```
U*U'
ans =
    1.0000   -0.0000    0.0000
   -0.0000    1.0000         0
    0.0000         0    1.0000
```

```
V'*V
ans =
    1.0000   -0.0000    0.0000
   -0.0000    1.0000         0
    0.0000         0    1.0000
```

```
V*V'
ans =
    1.0000   -0.0000    0.0000
   -0.0000    1.0000         0
    0.0000         0    1.0000
```

Fattorizzazioni di una matrice : SVD

in MATLAB…

Singular Value Decomposition

```
[U,S,V]=svd(A)
[U,S,V]=svd(A,0)
[U,S,V]=svd(A,"econ")
```

$A = USV^T$

$m = n$

$$A = U \; S \; V^T$$

U,V matrici ortogonali

S matrice diagonale contenente i valori singolari σ_k

Esempi

Risoluzione di un sistema

```
A=[1 2 1; 2 2 2;1 2 2];
b=[11 3 7]';
x=A\b
x =
          -4
         9.5
          -4
[U,S,V]=svd(A);
x1=V*diag(1./diag(S))*U'*b
x1 =
          -4
         9.5
          -4
```

Calcolo del determinante

```
A=[1 2 1; 2 2 2;1 2 2];
det(A)
ans =
    -2
[U,S,V]=svd(A);
[prod(diag(S)) det(U) det(V)]
ans =
            2            1           -1
```

Fattorizzazioni di una matrice : SVD in MATLAB…

Singular Value Decomposition

```
[U,S,V]=svd(A)
[U,S,V]=svd(A,0)
[U,S,V]=svd(A,"econ")
```

$A = USV^T$

$m > n$

A

Esempio: A rettangolare

```
A=[3 0;-2 1;1 0];
[U,S,V]=svd(A)
U =
   -0.78449     -0.53346     -0.31623
    0.56231     -0.82692   1.3878e-17
    -0.2615     -0.17782      0.94868
S =
     3.7816            0
          0      0.83622
          0            0
V =
   -0.98888      -0.1487
     0.1487     -0.98888
```

```
A=[3 0;-2 1;1 0];
[U,S,V]=svd(A,0) % equiv. a svd(A, "econ")
U =
   -0.78449     -0.53346
    0.56231     -0.82692
    -0.2615     -0.17782
S =
     3.7816            0
          0      0.83622
V =
   -0.98888      -0.1487
     0.1487     -0.98888
```

```
U'*U
ans =
     1.0000    -0.0000
    -0.0000     1.0000
```

```
U*U'     ≠ I
ans =
     0.9000    -0.0000     0.3000
    -0.0000     1.0000    -0.0000
     0.3000    -0.0000     0.1000
```

$$A = U\,S\,V^T$$

U,V matrici ortogonali
S matrice diagonale rettangolare contenente i valori singolari σ_k

$$A = U\,S\,V^T$$

U matrice con colonne ortogonali
S matrice diagonale contenente i valori singolari σ_k

Fattorizzazioni di una matrice : SVD in MATLAB…

Singular Value Decomposition

```
[U,S,V]=svd(A)
[U,S,V]=svd(A,0)
[U,S,V]=svd(A,"econ")
```

$A=USV^T$

$m<n$

A

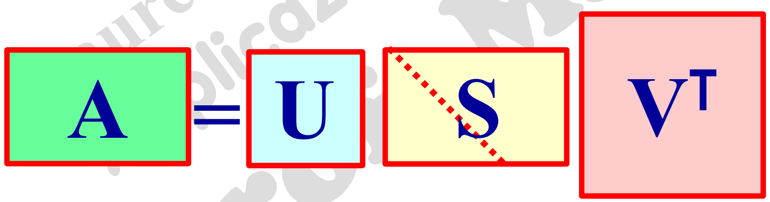
Esempio: A rettangolare

```
A=[-2 1 1;1 0 2];
[U,S,V]=svd(A) % equiv. a svd(A,0)
U =
     1     0
     0    -1
S =
    2.4495            0            0
         0       2.2361            0
V =
   -0.8165     -0.44721     -0.36515
   0.40825  -2.7756e-17     -0.91287
   0.40825     -0.89443      0.18257
```

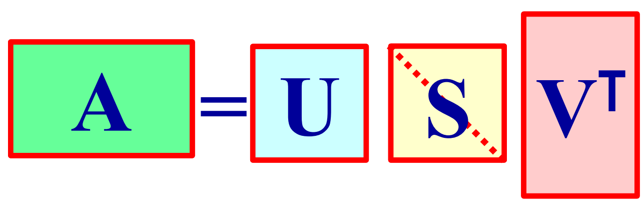
```
A=[-2 1 1;1 0 2];
[U,S,V]=svd(A,"econ")
U =
     1     0
     0    -1
S =
    2.4495            0
         0       2.2361
V =
   -0.8165     -0.44721
   0.40825  -2.7756e-17
   0.40825     -0.89443
```

```
V'*V
ans =
    1.0000   -0.0000
   -0.0000    1.0000
```

```
V*V'      ≠ I
ans =
    0.8667   -0.3333    0.0667
   -0.3333    0.1667    0.1667
    0.0667    0.1667    0.9667
```



U,V matrici ortogonali
S matrice diagonale rettangolare contenente i valori singolari σ_k



V matrice con colonne ortogonali
S matrice diagonale contenente i valori singolari σ_k

Richiami: generalizzazione del metodo di Gauss

Sistemi lineari indeterminati

Risoluzione di un sistema

$$Ax = b$$

dove A è una matrice rettangolare o quadrata (rango non massimo).

1) Caso omogeneo

$$Ax = 0$$

2) Caso non omogeneo

$$Ax = b \neq 0$$

Esempio

Si voglia risolvere il sistema $\boldsymbol{Ax=0}$ dove

$$A = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 6 & 9 \\ 3 & 9 & 8 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 6 & 9 \\ 3 & 9 & 8 \end{pmatrix} \xrightarrow{1^\circ \text{ p. } G^{\downarrow}} \begin{pmatrix} 1 & 3 & 2 \\ 0 & \boxed{0} & 5 \\ 0 & \boxed{0} & 2 \end{pmatrix} \xrightarrow{2^\circ \text{ p. } G^{\downarrow}} \textbf{?}$$

pivot???

... cosa fare ?

pivot!!!

Si passa ...
...alla colonna successiva
... e si continua ...

$$\begin{pmatrix} 1 & 3 & 2 \\ 0 & 0 & 5 \\ 0 & 0 & 2 \end{pmatrix} \xrightarrow{2^\circ \text{ p. } G^{\downarrow}} \begin{pmatrix} 1 & 3 & 2 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{pmatrix}$$

Si è ottenuto: $A = LS$

con S ***matrice a scala*** (o ***trapezoidale superiore***), e

$$Ax=0 \quad \Leftrightarrow \quad Sx=0$$

dove

$$L = \begin{pmatrix} 1 & & \\ 2 & 1 & \\ 3 & 2/5 & 1 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & 3 & 2 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{pmatrix}$$

pivot

Un sistema omogeneo è sempre compatibile ($x=0$ è sempre soluzione); in questo caso però ci sono anche altre soluzioni oltre quella banale.

$\Downarrow$

sistema omogeneo indeterminato = infinite soluzioni

Quanti sono i pivot ($\neq 0$) ? 2

$$S = \begin{pmatrix} 1 & 3 & 2 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{pmatrix}$$

Definizione operativa: rango di matrice

$$r = rango(A) = rango(S)$$

$$=$$

numero di pivot

Nell'esempio

$$r = rango(A) = rango(S) = 2 < 3 = n$$

dove $A(3\times3)$

Riprendendo l'esempio, a $\boldsymbol{Sx=0}$ corrisponde il sistema

$$\begin{cases} x_1 + 3x_2 + 2x_3 = 0 \\ 5x_3 = 0 \\ 0 = 0 \end{cases} \Leftrightarrow \begin{cases} x_1 + 3x_2 + 2x_3 = 0 \\ 5x_3 = 0 \end{cases} \Leftrightarrow$$

2 equazioni, 3 incognite
(matrice rettangolare larga)

$$\begin{cases} x_1 = -3x_2 - 2x_3 \\ x_3 = 0 \end{cases} \Leftrightarrow \begin{cases} x_1 = -3\lambda \\ x_2 = \lambda \\ x_3 = 0 \end{cases} \forall \lambda \in \mathbb{R} \Leftrightarrow \boxed{x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \lambda \begin{pmatrix} -3 \\ 1 \\ 0 \end{pmatrix}, \ \forall \lambda \in \mathbb{R}}$$

soluzione generale del sistema omogeneo indeterminato

Ancora $\begin{cases} x_1 + 3x_2 + 2x_3 = 0 \\ 5x_3 = 0 \end{cases} \Leftrightarrow \begin{cases} x_1 + 2x_3 = -3x_2 \\ 5x_3 = 0 \end{cases}$

$$S = \begin{array}{c} \begin{array}{ccc} x_1 & & x_3 \end{array} \\ \begin{pmatrix} 1 & 3 & 2 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \qquad \Updownarrow \quad \boldsymbol{Sx = 0}$$

Cosa sono le variabili $\boldsymbol{x_1}$ e $\boldsymbol{x_3}$?
… e la variabile $\boldsymbol{x_2}$?

Definizione: Le r variabili (dove r=*rango*(A)) corrispondenti alle colonne dei pivot (≠0) si dicono **variabili fondamentali**; le rimanenti $n-r$ si dicono **variabili libere**.

Il sistema triang. sup. $\begin{cases} x_1 + 2x_3 = -3x_2 \\ 5x_3 = 0 \end{cases}$ esprime le ***variabili fondamentali*** in funzione dell'unica variabile libera.

Dopo aver applicato $\mathbf{G}^{\downarrow}$

$$\boldsymbol{Ax = 0} \quad \Leftrightarrow \quad \boldsymbol{Sx = 0}$$

come applicare $\mathbf{G}^{\uparrow}$ per ottenere la soluzione generale **?**

Nell'esempio, si assegna il valore **1** all'unica **variabile libera** x_2 e poi si risolve il seguente sistema triangolare

$$\begin{cases} x_1 + 2x_3 = -3 \\ \quad\quad 5x_3 = 0 \end{cases}$$

che si ottiene da $\boldsymbol{S}$:

- eliminando le righe nulle;
- considerando solo le colonne delle variabili fondamentali nella matrice dei coefficienti;
- prendendo come termine noto l'opposto del vettore colonna corrispondente alla variabile libera.

	x_1 (V.F.)	x_2 (V.L.)	x_3 (V.F.)
x	?	1	?

$$\begin{cases} x_1 + 2x_3 = -3 \\ \qquad 5x_3 = 0 \end{cases} \Leftrightarrow \begin{pmatrix} 1 & 2 \\ & 5 \end{pmatrix}\begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = -\begin{pmatrix} \mathbf{3} \\ \mathbf{0} \end{pmatrix} \qquad S = \begin{pmatrix} 1 & \boxed{\mathbf{3}} & 2 \\ 0 & \boxed{\mathbf{0}} & 5 \\ \cancel{0} & \cancel{0} & \cancel{0} \end{pmatrix}$$

$$-$$

$$\begin{pmatrix} 1 & 2 & \mathbf{-3} \\ & 5 & \mathbf{0} \end{pmatrix} \xrightarrow{G^{\uparrow}} \begin{pmatrix} 1 & 0 & \mathbf{-3} \\ & 5 & \mathbf{0} \end{pmatrix} = \begin{pmatrix} 1 & \\ & 5 \end{pmatrix}\begin{pmatrix} \boldsymbol{I} & \begin{matrix} -3 \\ 0 \end{matrix} \end{pmatrix}$$

x	
x_1 (V.F.)	**−3**
x_2 (V.L.)	**1**
x_3 (V.F.)	**0**

soluzione $x = \lambda \begin{pmatrix} -3 \\ 1 \\ 0 \end{pmatrix}$

Esempio 2

Si voglia risolvere il sistema $\boldsymbol{Ax} = \mathbf{0}$ dove

$$A = \begin{pmatrix} 1 & 3 & 2 & 1 \\ 2 & 6 & 4 & 2 \\ 3 & 9 & 6 & 4 \end{pmatrix}$$

pivot!

$$A = \begin{pmatrix} 1 & 3 & 2 & 1 \\ 2 & 6 & 4 & 2 \\ 3 & 9 & 6 & 4 \end{pmatrix} \xrightarrow{1^\circ \text{ p. } G^{\downarrow}} \begin{pmatrix} 1 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 3 & 2 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} = S$$

$r = rango(A) = 2$

$r = 2$ variabili fondam. x_1 e x_4

$n\text{-}r = 2$ variabili libere x_2 e x_3

Che valori assegnare alle 2 variabili libere?

Quale sistema triangolare risolvere?

Si assegna a turno il valore **1** a ciascuna variabile libera e zero alle altre…

	$x^{(1)}$	$x^{(2)}$
x_1 (v.f.)	**?**	**?**
x_2 (v.l.)	**1**	**0**
x_3 (v.l.)	**0**	**1**
x_4 (v.f.)	**?**	**?**

num. di var. libere=**2** ⇒ doppia infinità di soluzioni

…e si risolve un opportuno **sist. triang. multiplo !**

Un sistema lineare **multiplo** $AX=B$ consiste di più sistemi lineari (tanti quante sono le colonne di B) aventi la stessa matrice dei coefficienti.

Analogamente a prima, il sistema triangolare multiplo si ottiene da $\boldsymbol{S}$

- eliminando le righe nulle;
- considerando solo le colonne delle variabili fondamentali nella matrice dei coefficienti;
- prendendo come termini noti gli opposti delle colonne corrispondenti alle variabili libere.

variabili fondamentali

$$\begin{pmatrix} 1 & 1 \\ & 1 \end{pmatrix} \begin{pmatrix} x_1^1 & x_1^2 \\ x_4^1 & x_4^2 \end{pmatrix} = \begin{pmatrix} -3 & -2 \\ 0 & 0 \end{pmatrix} \qquad \boldsymbol{S} = \begin{pmatrix} 1 & 3 & 2 & 1 \\ 0 & 0 & 0 & 1 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \end{pmatrix}$$

$-$

$$\begin{pmatrix} 1 & 1 \\ & 1 \end{pmatrix}\begin{pmatrix} x_1^{\ 1} & x_1^{\ 2} \\ x_4^{\ 1} & x_4^{\ 2} \end{pmatrix} = \begin{pmatrix} -3 & -2 \\ 0 & 0 \end{pmatrix}$$

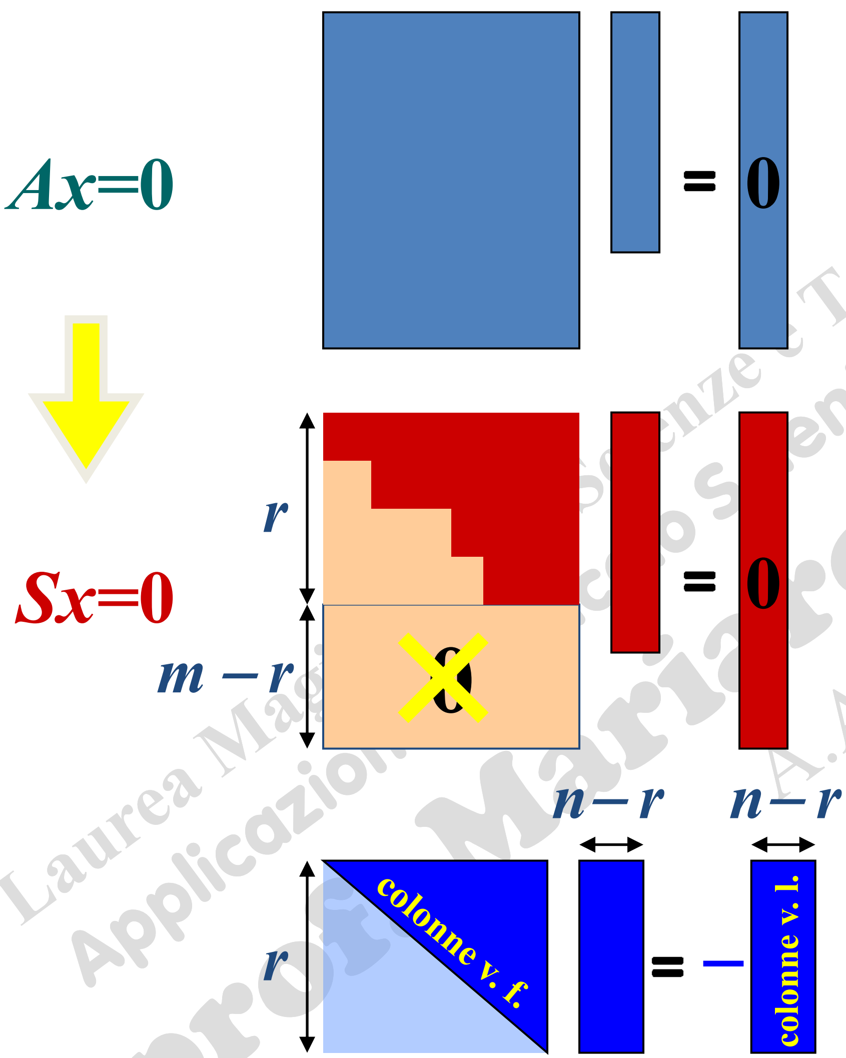
$$\begin{pmatrix} 1 & 1 & -3 & -2 \\ 0 & 1 & 0 & 0 \end{pmatrix} \xrightarrow{G^{\uparrow}} \begin{pmatrix} 1 & 0 & -3 & -2 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{I} & \begin{matrix} -3 & -2 \\ 0 & 0 \end{matrix} \end{pmatrix}$$

	$x^{(1)}$	$x^{(2)}$
x_1 (v.f.)	-3	-2
x_2 (v.l.)	1	0
x_3 (v.l.)	0	1
x_4 (v.f.)	0	0

Pertanto
la soluzione generale
del sistema è

$$\boldsymbol{x} = \lambda\begin{pmatrix} -3 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \boldsymbol{\mu}\begin{pmatrix} -2 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \qquad \forall \lambda, \boldsymbol{\mu} \in \mathbb{R}$$

In generale $Ax=0 \iff Sx=0$ dove $A(m \times n)$ e $r < m$.



Si assegna a turno il valore **1** a ciascuna variabile libera e **0** alle altre…

sistema triangolare multiplo

Esempio 3 Si voglia risolvere il sistema $\boldsymbol{Ax}=\underline{\mathbf{0}}$ dove

$$\boldsymbol{A}=\begin{pmatrix}1&3&2&1&1\\2&6&4&2&2\\3&9&6&4&2\end{pmatrix}$$

$$\boldsymbol{A}=\begin{pmatrix}1&3&2&1&1\\2&6&4&2&2\\3&9&6&4&2\end{pmatrix}\xrightarrow{1^\circ\text{ p. }G\downarrow}\begin{pmatrix}1&3&2&1&1\\0&0&0&0&0\\0&0&0&1&-1\end{pmatrix}$$

pivot!

$$\begin{pmatrix}1&3&2&1&1\\0&0&0&1&-1\\\cancel{0}&\cancel{0}&\cancel{0}&\cancel{0}&\cancel{0}\end{pmatrix}=\boldsymbol{S}$$

cambiare segno

$$\begin{array}{cc|ccc}\text{V.F.}&&&\text{V.L.}&\\\hline 1&1&-3&-2&-1\\0&1&0&0&1\end{array}$$

	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$
x_1 (v.f.)	?	?	?
x_2 (v.l.)	1	0	0
x_3 (v.l.)	0	1	0
x_4 (v.f.)	?	?	?
x_5 (v.l.)	0	0	1

$$\begin{pmatrix} 1 & 1 & -3 & -2 & -1 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \xrightarrow{\;G^{\uparrow}\;} \begin{pmatrix} 1 & 0 & -3 & -2 & -2 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$
x_1 $^{(v.f.)}$	-3	-2	-2
x_2 $^{(v.l.)}$	1	0	0
x_3 $^{(v.l.)}$	0	1	0
x_4 $^{(v.f.)}$	0	0	1
x_5 $^{(v.l.)}$	0	0	1

x soluzione generale del sistema

$$\boldsymbol{x} = \alpha \begin{pmatrix} -3 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} -2 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \gamma \begin{pmatrix} -2 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \qquad \forall \alpha, \beta, \gamma \in \mathbb{R}$$

Richiami: generalizzazione del metodo di Gauss

Sistemi lineari indeterminati

Risoluzione di un sistema

$$Ax = b$$

dove A è una matrice rettangolare o quadrata (rango non massimo).

1) Caso omogeneo

$$Ax = 0$$

2) Caso non omogeneo

$$Ax = b \neq 0$$

Esempio: sistema incompatibile

Si voglia risolvere il sistema $Ax=b$ dove

$$A=\begin{pmatrix}1 & 3 & 2\\ 2 & 6 & 9\\ 3 & 9 & 8\end{pmatrix};\qquad b=\begin{pmatrix}1\\ -3\\ 2\end{pmatrix}$$

$$[A,b]=\left(\begin{array}{ccc|c}1 & 3 & 2 & 1\\ 2 & 6 & 9 & -3\\ 3 & 9 & 8 & 2\end{array}\right)\xrightarrow{1^\circ\ \text{p. } G\downarrow}\left(\begin{array}{ccc|c}1 & 3 & 2 & 1\\ 0 & 0 & 5 & -5\\ 0 & 0 & 2 & -1\end{array}\right)$$

$$\left(\begin{array}{ccc|c}1 & 3 & 2 & 1\\ 0 & 0 & 5 & -5\\ 0 & 0 & 2 & -1\end{array}\right)\xrightarrow{2^\circ\ \text{p. } G\downarrow}\left(\begin{array}{ccc|c}1 & 3 & 2 & 1\\ 0 & 0 & 5 & -5\\ \boxed{0 \quad 0 \quad 0} & & & ?\ \boxed{1}\end{array}\right)=[S,c]$$

Il **sistema è incompatibile** perché alla riga nulla di S corrisponde nel termine noto una componente non nulla.

Esempio: sistema compatibile

Si voglia risolvere il sistema $\boldsymbol{Ax=b}$ dove

$$\boldsymbol{A}=\begin{pmatrix}1&3&2\\2&6&9\\3&9&8\end{pmatrix};\qquad \boldsymbol{b}=\boxed{\begin{pmatrix}1\\-3\\1\end{pmatrix}}$$

$$[\boldsymbol{A},\boldsymbol{b}]=\left(\begin{array}{ccc|c}1&3&2&1\\2&6&9&-3\\3&9&8&1\end{array}\right)\xrightarrow{1^\circ\text{ p. }G^{\downarrow}}\left(\begin{array}{ccc|c}1&3&2&1\\0&0&5&-5\\0&0&2&-2\end{array}\right)$$

pivot

$$\left(\begin{array}{ccc|c}1&3&2&1\\0&0&5&-5\\0&0&2&-2\end{array}\right)\xrightarrow{2^\circ\text{ p. }G^{\downarrow}}\left(\begin{array}{ccc|c}1&3&2&1\\0&0&5&-5\\\hline 0&0&0&!\ 0\end{array}\right)$$

Il **sistema è compatibile** perché alla riga nulla di $\boldsymbol{S}$ corrisponde nel termine noto una componente nulla.

Si è ottenuto:

$$A = LS$$

con S ***matrice a scala***, e

$$Ax=b \quad \Leftrightarrow \quad Sx=c$$

dove

$$L = \begin{pmatrix} 1 & & \\ 2 & 1 & \\ 3 & 2/5 & 1 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & 3 & 2 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ -5 \\ 0 \end{pmatrix}$$

Il **sistema è compatibile** perché alla riga nulla di S corrisponde una componente nulla anche nel termine noto c!

sistema compatibile indeterminato

=

infinite soluzioni

variabili fondamentali: x_1 e x_3;
variabile libera: x_2.

$$r = rango(A) = 2$$

$$S = \begin{pmatrix} 1 & 3 & 2 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{pmatrix}$$

$$n - r = 1$$

semplice infinità di soluzioni

A $\boldsymbol{Sx=c}$ corrisponde il sistema

$$\begin{cases} x_1 + 3x_2 + 2x_3 = 1 \\ 5x_3 = -5 \\ 0 = 0 \end{cases} \Leftrightarrow \begin{cases} x_1 = -3x_2 + 3 \\ x_3 = -1 \end{cases}$$

$$\begin{cases} x_1 = -3\lambda + 3 \\ x_2 = \lambda \\ x_3 = -1 \end{cases} \quad \forall\lambda \in \mathbb{R} \Leftrightarrow \boxed{x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ -1 \end{pmatrix} + \lambda \begin{pmatrix} -3 \\ 1 \\ 0 \end{pmatrix}, \quad \forall\lambda \in \mathbb{R}}$$

Cosa sono ???

soluzione generale del sistema non omogeneo indeterminato

Ponendo $\lambda=0$ in $$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ -1 \end{pmatrix} + \lambda \begin{pmatrix} -3 \\ 1 \\ 0 \end{pmatrix}, \quad \forall \lambda \in \mathbb{R}$$

risulta $$Ax^p = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 6 & 9 \\ 3 & 9 & 8 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ -3 \\ 1 \end{pmatrix} = b$$

cioè si è ottenuta una **soluzione particolare $x^{(p)}$** di $Ax = b$.

Invece $$Ax^0 = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 6 & 9 \\ 3 & 9 & 8 \end{pmatrix} \begin{pmatrix} -3 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

cioè $\lambda x^{(0)}$ è la **soluzione generale** di $Ax = 0$ (sistema omogeneo associato).

Teorema

La soluzione generale di un sistema non omogeneo compatibile indeterminato

$$Ax = b$$

si ottiene sommando alla soluzione generale del sistema omogeneo associato

$$Ax = 0$$

una soluzione particolare di

$$Ax = b$$

Sistemi lineari indeterminati

Se si definisce $\mathcal{N}(A)$, lo Spazio Nullo della matrice $A(m\times n)$, come l'insieme delle soluzioni del sistema omogeneo $Ax=\mathbf{0}$, cioè

$$\mathcal{N}\;\;A \;=\; u\in\mathbb{R}^m : Au=\vec{0}$$

allora la **soluzione generale** di un sistema indeterminato è data da:

- un qualsiasi vettore dello Spazio Nullo $\mathcal{N}(A)$ se il sistema è omogeneo;
- la somma di una soluzione particolare del sistema più un qualsiasi vettore dello Spazio Nullo $\mathcal{N}(A)$ se il sistema è non omogeneo.

Lo Spazio Nullo Sinistro $\mathcal{N}(A^T)$ non è altro che lo Spazio Nullo della matrice trasposta di A.

Esempio 1 MATLAB simbolico

Si voglia risolvere il sistema ***Ax=b*** dove

$$\boldsymbol{A}=\begin{pmatrix}1 & 3 & 2 & -1\\ 2 & 6 & 9 & -2\\ 3 & 9 & 8 & -3\end{pmatrix};\qquad \boldsymbol{b}=\begin{pmatrix}1\\ -3\\ 1\end{pmatrix}$$

```
A=sym([1 3 2 -1;2 6 9 -2;3 9 8 -3])
A =
[1, 3, 2, -1]
[2, 6, 9, -2]
[3, 9, 8, -3]
b=sym([1 -3 1]')
b =
 1
-3
 1
```

```
[m,n]=size(A); m, n
m = 3          num. equazioni
n = 4          num. incognite
rank(A), rank(A) == rank([A b])
ans =
   2
ans =
  logical
   1
```

```
Xp = A\b
Warning: Solution is not unique because
the system is rank-deficient.
Xp =           soluzione particolare di Ax=b
 3
 0
-1
 0
N = null(A)        base per Spazio Nullo
N =
[-3, 1]
[ 1, 0]
[ 0, 0]
[ 0, 1]
```

```
syms mu [size(N,2) 1] real
Xg=Xp + N*mu          soluzione generale di Ax=b
Xg =
mu2 - 3*mu1 + 3
            mu1
             -1
            mu2
```

```
simplify(A*Xg == b)
ans =
symtrue
symtrue            verifica
symtrue
```

Esempio 2 MATLAB simbolico

Si voglia risolvere il sistema $Ax=b$ dove $A=\begin{pmatrix}3 & -6\\ 1 & -2\end{pmatrix}$; $b=\begin{pmatrix}15\\ 5\end{pmatrix}$

```
A=sym([3 -6;1 -2])
[3, -6]
[1, -2]
b=sym([15 5]')
b =
 15
  5
```

```
[m,n]=size(A); m, n
m = 2
n = 2
rank(A), rank(A) == rank([A b])
ans =
    1
ans =
  logical
  1
```

m = 2 ← num. equazioni

n = 2 ← num. incognite

```
Xp = A\b
Warning: Solution is not unique because
the system is rank-deficient.
Xp =
 5
 0
N = null(A)
N =
2
1
```

Xp ← soluzione particolare di $Ax=b$

N = null(A) ← base per Spazio Nullo

```
syms mu [size(N,2) 1] real
Xg=Xp + N*mu
Xg =
2*mu1 + 5
      mu1
```

Xg=Xp + N*mu ← soluzione generale di $Ax=b$

```
simplify(A*Xg == b)
ans =
symtrue
symtrue
```

verifica

Esempio 1 MATLAB numerico

Si voglia risolvere il sistema $\boldsymbol{Ax=b}$ dove

$$\boldsymbol{A} = \begin{pmatrix} 1 & 3 & 2 & -1 \\ 2 & 6 & 9 & -2 \\ 3 & 9 & 8 & -3 \end{pmatrix}; \quad \boldsymbol{b} = \begin{pmatrix} 1 \\ -3 \\ 1 \end{pmatrix}$$

```
A=[1 3 2 -1;2 6 9 -2;3 9 8 -3];
b=[1 -3 1]';
[m,n]=size(A); m, n
m = 3
n = 4
rank(A), rank(A) == rank([A b])
ans =
   2
ans =
  logical
   1
```

num. equazioni

num. incognite

```
Xp = A\b
Warning: Rank deficient, rank = 2, tol
=  1.084160e-14.
Xp =
          0
          1
         -1
          0
N=null(A,"rational")
N =
  -3   1
   1   0
   0   0
   0   1
```

soluzione particolare di $Ax=b$

base per Spazio Nullo

```
syms mu [size(N,2) 1] real
Xg=Xp + N*mu
Xg =
mu2 - 3*mu1
     mu1 + 1
          -1
         mu2
```

soluzione generale di $Ax=b$

```
simplify(A*Xg == b)
ans =
symtrue
symtrue
symtrue
```

verifica

Esempio 2 MATLAB numerico

Si voglia risolvere il sistema $Ax=b$ dove $A=\begin{pmatrix}3 & -6\\ 1 & -2\end{pmatrix}$; $b=\begin{pmatrix}15\\ 5\end{pmatrix}$

```
A=[3 -6;1 -2];  b=[15 5]';
[m,n]=size(A); m, n
m = 2
n = 2
r=rank(A), rank(A) == rank([A b])
r =
   1
ans =
 logical
  1
Xp = A\b
Warning: Matrix is singular to
working precision.
Xp =
 NaN
 NaN
```

m = 2 ← num. equazioni

n = 2 ← num. incognite

???

```
[L,U,P]=lu([A b])
L =
             1             0
       0.33333             1
U =
     3    -6    15
     0     0     0
P =
     1     0
     0     1
Xp = U(1:r,1:end-1)\U(1:r,end)
Xp =
             0
          -2.5
N = null(A,"rational")
N =
2
1
```

Xp ← soluzione particolare di $Ax=b$

N ← base per Spazio Nullo

```
syms mu [size(N,2) 1] real
Xg=Xp + N*mu
Xg =
2*mu1
Mu1 - 5/2
```

Xg ← soluzione generale di $Ax=b$

```
simplify(A*Xg == b)
ans =
symtrue
symtrue
```

verifica

Potenza dell'operatore di divisione di MATLAB per risolvere sistemi lineari

Sia dato il sistema $\boldsymbol{Ax} = \boldsymbol{b}$ dove $\boldsymbol{A}(m \times n)$, $\boldsymbol{x}(n \times 1)$, $\boldsymbol{b}(m \times 1)$ sono array reali, allora l'istruzione

```
x=A\b
```

restituisce …

1 la soluzione del sistema, se questo è compatibile determinato

Esempio

```
A=[1 0 2;-2 1 1;3 4 0]; b=[7 3 11]';
```

$\boldsymbol{Ax} = \boldsymbol{b} \Longleftrightarrow \boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$

```
x=A\b
x =
     1
     2
     3
```

```
x=inv(A)*b
x =
     1
     2
     3
```

$(\boldsymbol{Ax})^\top = \boldsymbol{b}^\top \Longleftrightarrow \boldsymbol{x}^\top \boldsymbol{A}^\top = \boldsymbol{b}^\top \Longleftrightarrow \boldsymbol{x}^\top = \boldsymbol{b}^\top (\boldsymbol{A}^\top)^{-1}$

```
xt=b'/A'
xt =
     1     2     3
```

```
xt=b'*inv(A')
xt =
     1     2     3
```

2

la soluzione "ai minimi quadrati" (LS - *Least Squares solution*), se il sistema è incompatibile, dove

$$\boldsymbol{x}\ LS\ sol \overset{\text{def}}{\Longleftrightarrow} \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2 = \min_y \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{y}\|_2 \le \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{y}\|_2 \quad \forall \boldsymbol{y} \in \mathbb{R}^n$$

Se ***rango***$(A)=n$ la soluzione ai minimi quadrati è unica.

Se ***rango***$(A)<n$ la soluzione non è unica e ***MATLAB*** ne restituisce una con il minor numero di componenti nulle.

Esempio 1

```
A=[1 2;1 2;3 -4]; b=[7 3 11]';
disp([rank(A) rank([A b])])
   2     3
x=A\b
x =
    4.2000
    0.4000
norm(b-A*x)
ans =
    2.8284
y=rand(2,1); norm(b-A*y)
…
```

Esempio 2

```
A=[1 1;1 1;1 1]; b=[1 0 0]';
disp([rank(A) rank([A b])])
   1     2
x=A\b
Warning: Rank deficient, rank = 1
tol =   1.1538e-015.
x =
    0.3333
         0
```

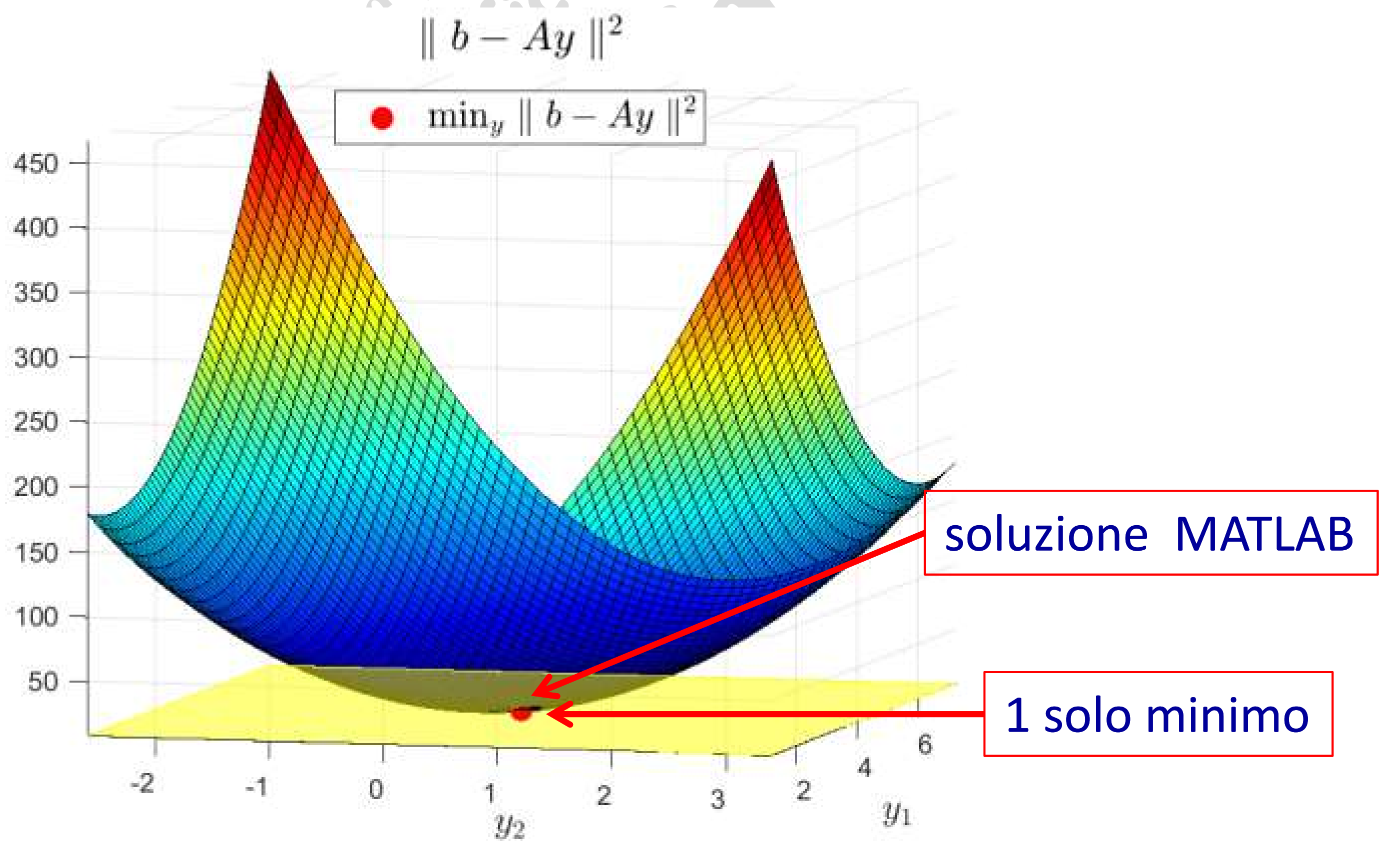
MATLAB restituisce come soluzione un punto col numero di componenti non nulle pari al rango di A

2

la soluzione "ai minimi quadrati" (LS - *Least Squares solution*), se il sistema è incompatibile, dove

$$\boldsymbol{x}\ LS\ sol \overset{\text{def}}{\Longleftrightarrow} \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2 = \min_y \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{y}\|_2 \le \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{y}\|_2 \quad \forall \boldsymbol{y}\in\mathbb{R}^n$$

Esempio 1: ***rango(A)=n***

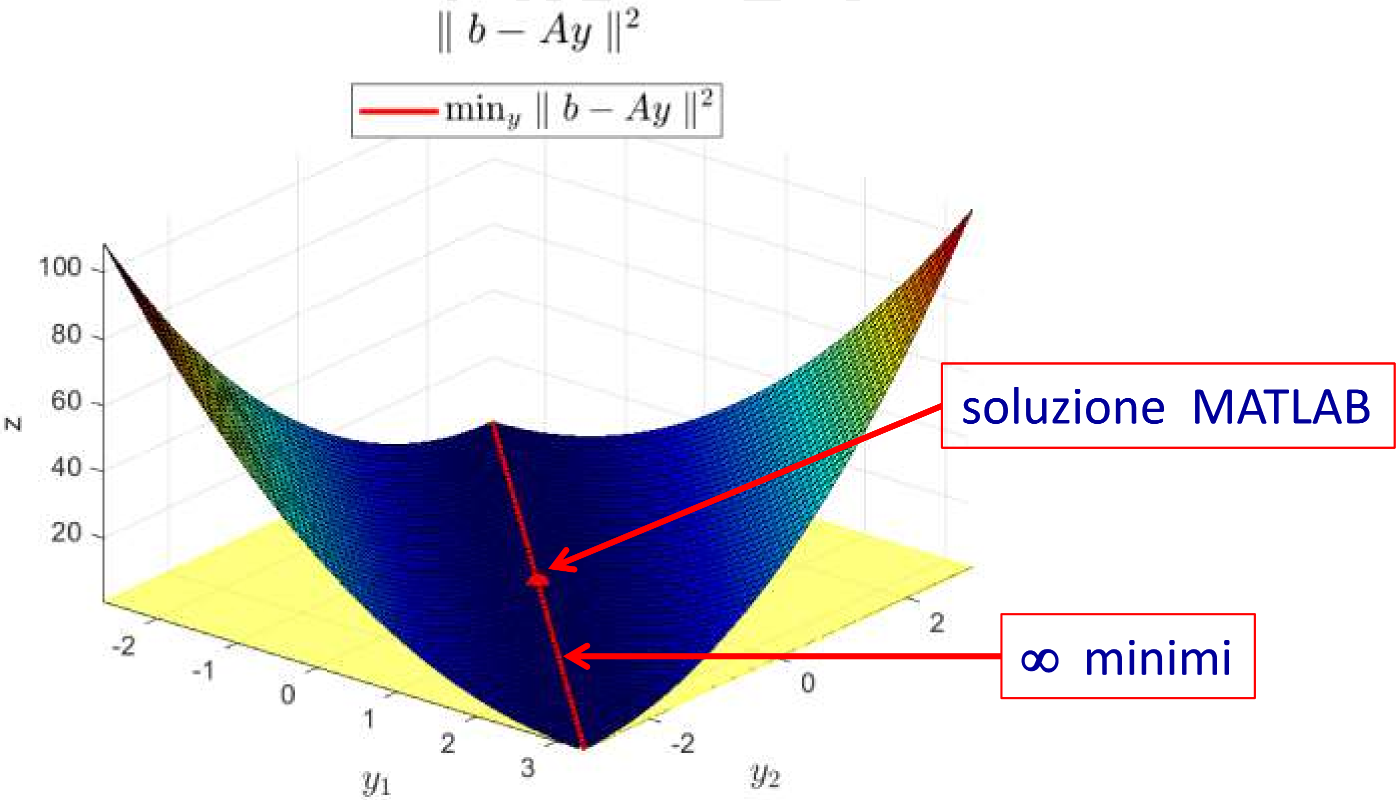


2

la soluzione "ai minimi quadrati" (LS - *Least Squares solution*), se il sistema è incompatibile, dove

$$\boldsymbol{x}\ LS\ sol \overset{\text{def}}{\Longleftrightarrow} \|\boldsymbol{b}-A\boldsymbol{x}\|_2 = \min_y \|\boldsymbol{b}-A\boldsymbol{y}\|_2 \le \|\boldsymbol{b}-A\boldsymbol{y}\|_2 \quad \forall \boldsymbol{y} \in \mathbb{R}^n$$

Esempio 2: ***rango(A)<n***



3 una **soluzione particolare** (con al più r componenti non nulle dove r=*rango*(A)), se il **sistema è compatibile indeterminato.**

Richiami: La **soluzione generale** x di un sistema indeterminato

$Ax=b$ si scrive $$x = x^{(p)} + X^{(0)}\Lambda, \quad \forall \Lambda$$

dove $X^{(0)}\Lambda$ è la soluzione generale di $Ax = 0$

$x^{(p)}$ è una soluzione particolare di $Ax = b$

Esempio 3

```
A=[1 0 -1 2; -2 1 0 1];
b=[7 3]'; xp=A\b
xp =
       0.2
         0
         0
       3.4
X0=null(A,'rational')
X0 =
     1    -2
     2    -5
     1     0
     0     1
```

soluzione particolare di $Ax=b$

$\Lambda=(\lambda_1, \lambda_2)^T$

$$\lambda_1 \begin{pmatrix} 1 \\ 2 \\ 1 \\ 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} -2 \\ -5 \\ 0 \\ 1 \end{pmatrix}$$

soluzione generale di $Ax=0$

```
X0=null(A)
X0 =
      0.27101     -0.30532
       0.2367     -0.88165
      0.88165       0.2367
      0.30532      0.27101
```

```
dot(X0(:,1),X0(:,2))
ans =    -1.1102e-16
[norm(X0(:,1)) norm(X0(:,2))]
ans =
          1            1
```

colonne ortonormali: ortogonali fra loro e normalizzate a 1

Esempio: condizionamento di una matrice

Anche usando un algoritmo stabile ***(m. di Gauss con pivoting)***, un **problema malcondizionato** conduce ad una soluzione inaccurata!

```
n=9; A1=rand(n); A2=hilb(n);
x=ones(n,1); b1=A1*x; b2=A2*x;
x1=A1\b1; x2=A2\b2;                     soluzioni di riferimento
format short e; disp([norm(x-x1) norm(x-x2)])
 4.1108e-15   1.6531e-05                 errore assoluto
disp([cond(A1) cond(A2)])
 4.8875e+01   4.9315e+11
n=12; A1=rand(n); A2=hilb(n);
x=ones(n,1); b1=A1*x; b2=A2*x;
x1=A1\b1; x2=A2\b2;
Warning: Matrix is close to singular or badly scaled. Results
may be inaccurate. RCOND =  2.592620e-17.
format short e; disp([norm(x-x1) norm(x-x2)])
 1.0495e-14   6.5475e-01                 errore assoluto
disp([cond(A1) cond(A2)])
 8.1593e+01   1.6804e+16
```

matrice di Hilbert: esempio di matrice malcondizionata

indice di condizionamento

indica la sensibilità della soluzione del sistema all'errore nei dati del problema

Autovalori e autovettori

Richiami

Data una matrice $A(n\times n)$, si definisce

λ ***autovalore*** $\overset{def}{\Longleftrightarrow}$

$$\exists\, x \neq \underline{0} \;:\; Ax = \lambda x$$

x ***autovettore*** relativo a λ $\overset{def}{\Longleftrightarrow}$

$$Ax = \lambda x$$

in MATLAB...

```
A=[0   -6   -1
    6    2  -16
   -5   20  -10];
[V,D]=eig(A)
V =
-0.8326    -0.1203 + 0.2123i  -0.1203 - 0.2123i
-0.3553     0.4691 + 0.4901i   0.4691 - 0.4901i
-0.4248     0.6249 - 0.2997i   0.6249 + 0.2997i

D =
-3.0710                    0                  0
      0     -2.4645 +17.6008i                  0
      0                    0   2.4645 -17.6008i
```

La matrice **D** contiene sulla diagonale gli ***autovalori*** di **A**, mentre le colonne della matrice **V** sono gli ***autovettori*** corrispondenti.

```
[V,D]=eig(A,'vector'); D
D =
       -3.071 +         0i
      -2.4645 +    17.601i
      -2.4645 -    17.601i
```

D è il vettore degli ***autovalori*** di **A**