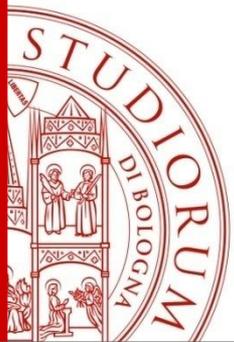


Analisi Numerica

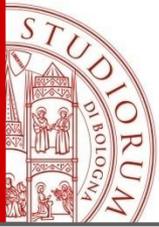
Serena Morigi

- <http://www.dm.unibo.it/~morigi>
- morigi@dm.unibo.it
- Dipartimento di Matematica
Università di Bologna



Condizionamento di un problema matematico

Stabilità dell'**algoritmo** risolutivo



Problemi ed algoritmi

Problema numerico:

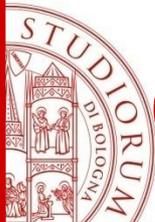
descrizione chiara e non ambigua di una connessione funzionale tra i dati x (input) del problema e i risultati desiderati (output) y .



Algoritmo:

sequenza di di operazioni che devono essere eseguite al fine di ottenere, in un numero finito di passi, da un vettore dati x il corrispondente output $\tilde{\psi}$ non necessariamente uguale ad y .

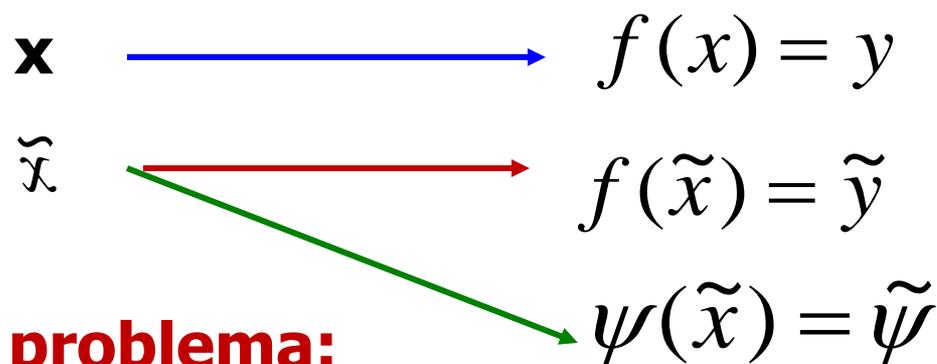
AD OGNI PROBLEMA NUMERICO POSSIAMO ASSOCIARE PIU' ALGORITMI



Condizionamento di un problema

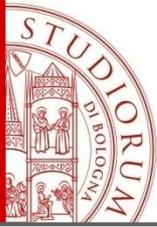
Perturbazioni nei dati: $\tilde{x} = x + \delta x$

Come vengono propagati dal problema (f) gli errori presenti nei dati in assenza di errori di calcolo e indipendentemente dal procedimento di calcolo?



Condizionamento di un problema:

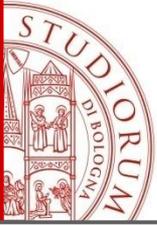
Il confronto tra la risposta analitica $f(x)$ e la risposta $f(\tilde{x}) = \tilde{y}$ ottenuta a partire da dati perturbati.



Problema ben/mal-condizionato

Quando a **piccole perturbazioni** (relative) nei **dati x** corrispondono perturbazioni (relative) sul risultato **$f(x)$** dello stesso ordine di grandezza, il **problema $y=f(x)$** è definito **ben condizionato**, altrimenti è detto **mal condizionato**.

Uno stesso problema può essere mal condizionato per certi dati ma non per altri.

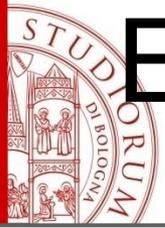


Indice di condizionamento

$$\frac{\|f(x) - f(\tilde{x})\|}{\|f(x)\|} \leq K \frac{\|x - \tilde{x}\|}{\|x\|}$$

K è detto **INDICE DI CONDIZIONAMENTO**

Il condizionamento è legato al problema numerico e non ha alcun legame con gli errori di arrotondamento delle operazioni di macchina nè con il particolare algoritmo utilizzato.



Esempio: problema mal condizionato (Wilkinson 1963)

Calcolare le radici del seguente polinomio:

$$p(x) = (x-1)(x-2)\dots(x-19)(x-20) \\ = x^{20} - 210x^{19} + \dots$$

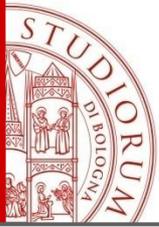
Calcolatore **B=2 t=30**

per il calcolo degli zeri **1, 2, 3, ..., 19, 20**

Per memorizzare i coefficienti del polinomio è necessario effettuare un arrotondamento alla 30 cifra significativa binaria

Perturbiamo ora il coefficiente di x^{19}

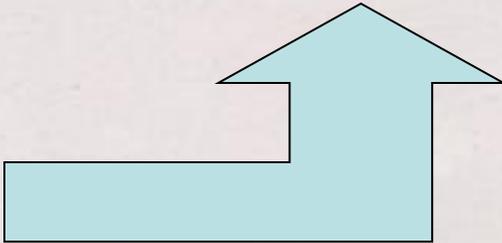
$$-210 \longrightarrow -210 + 2^{-23}$$

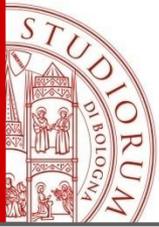


Esempio: problema mal condizionato (Wilkinson 1963)

**Il polinomio diventa: $p(x) + 2^{-23} x^{19} = 0$
come sono variate di conseguenza le radici?**

1.00000 0000	10.09526 6145 ± 0.64350 0904i
2.00000 0000	11.79363 3881 ± 1.65232 9728i
3.00000 0000	13.99235 8137 ± 2.51883 0070i
4.00000 0000	16.73073 7466 ± 2.81262 4894i
4.99999 9928	19.50243 9400 ± 1.94033 0347i
6.00000 6944	
6.99969 7234	
8.00726 7603	
8.91725 0249	
20.84690 8101	





Esempio: problema mal condizionato

Risolvere il sistema lineare:

$$\begin{cases} x + y = 2 \\ 1001x + 1000y = 2001 \end{cases}$$

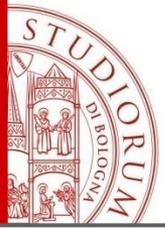
$$x = 1, \quad y = 1$$

Perturbiamo il coefficiente della x dell'1%:

$$\begin{cases} \left(1 + \frac{1}{100}\right)x + y = 2 \\ 1001x + 1000y = 2001 \end{cases}$$

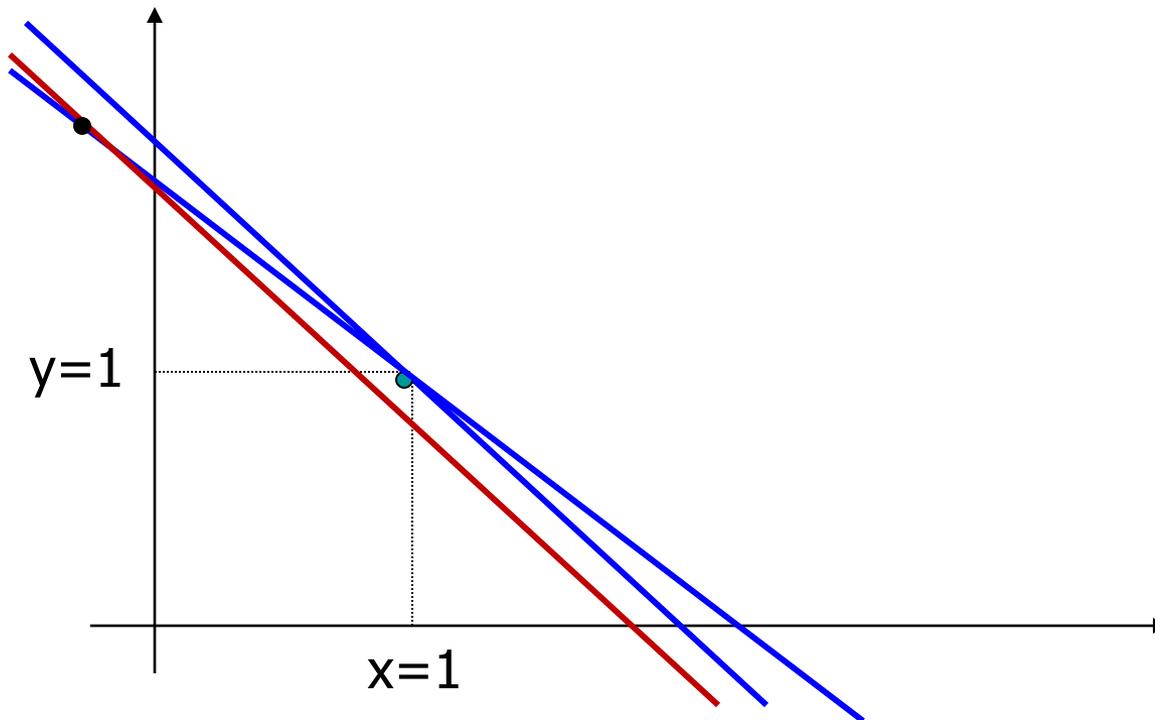
$$x = -1/9 = -0.1111, \quad y = 1901/900 = 2.1122$$

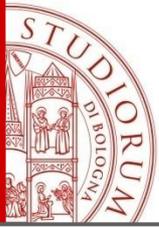
Errore del 110%



Esempio: problema mal condizionato

- Interpretazione geometrica





Algoritmo e stabilità

Stabilità di un algoritmo :

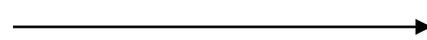
Confronto tra la risposta fornita dall'algoritmo

$$\psi(\tilde{x})$$

con

$$f(\tilde{x}) = y$$

\mathbf{x}



$$\mathbf{f}(\mathbf{x}) = \mathbf{y}$$

\tilde{x}

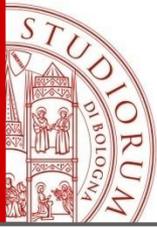


$$f(\tilde{x}) = \tilde{y}$$

$$\psi(\tilde{x}) = \tilde{\psi}$$

E' generato dal calcolo della funzione $\tilde{\psi}$ come composizione di un numero finito di operazioni di macchina

La stabilità di un algoritmo valuta quindi la reazione fornita dall'algoritmo all'introduzione di perturbazioni nei dati iniziali.



Errore inerente ed algoritmico

Nel calcolo di una f , il valore effettivamente calcolato in corrispondenza dei dati x può essere affetto da errori di due tipi:

- **ERRORE INERENTE:**

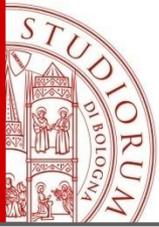
generato dalla rappresentazione dei dati come numeri finiti

$$E_{IN} = \frac{f(\tilde{x}) - f(x)}{f(x)}$$

- **ERRORE ALGORITMICO:**

generato dal calcolo della funzione $\psi(\tilde{x})$ dovuto alle operazioni in aritmetica finita

$$E_{ALG} = \frac{\psi(\tilde{x}) - f(\tilde{x})}{f(\tilde{x})}$$



ERRORE TOTALE

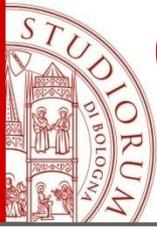
L'accuratezza della soluzione (di quanto si discosta la soluzione calcolata da quella esatta) dipende sia dal condizionamento del problema che dalla stabilità algoritmica.

$$E_{TOT} = \frac{\psi(\tilde{x}) - f(x)}{f(x)}$$

$$E_{TOT} = E_{ALG}(1 + E_{IN}) + E_{IN} \approx E_{ALG} + E_{IN}$$

Infatti:

$$E_{TOT} = \frac{\psi(\bar{x}) - f(x)}{f(x)} = \frac{\psi(\bar{x})}{f(x)} - 1 = \frac{\psi(\bar{x})}{f(\bar{x})} \frac{f(\bar{x})}{f(x)} - 1 = (1 + E_{ALG})(1 + E_{IN}) - 1$$

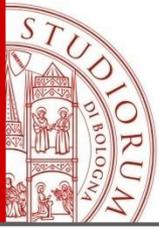


Condizionamento di un problema, algoritmo e stabilità

La stabilità dell'algoritmo non garantisce che il risultato calcolato sia accurato.

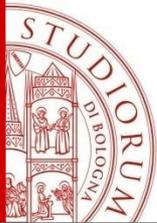
Per un problema mal condizionato la distinzione tra algoritmo stabile e instabile non è molto significativa in quanto l'errore totale risulta dominato dall'errore inerente.

Quindi per un problema mal condizionato è opportuna, in generale, una sua riformulazione.



Condizionamento di un problema, algoritmo e stabilità

La bassa accuratezza dei risultati prodotti da un processo numerico può essere imputabile all'alto condizionamento intrinseco del problema oppure all'instabilità dell'algoritmo utilizzato per produrlo.



Algoritmo Stabile

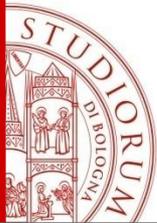
$$\left| E_{ALG} \right| \approx g(n) \cdot eps$$

n = numero di operazioni effettuate

$g(n) = c \cdot n$, $c > 0$, crescita dell'errore lineare

$g(n) = c^n$, $c > 1$ crescita dell'errore esponenziale

Un **algoritmo** numerico è considerato **stabile** se $g(n)$ è **lineare**, cioè l'errore algoritmico è dell'ordine di grandezza della precisione di macchina, **instabile** altrimenti.



Esempio algoritmo instabile

La successione

$$1, \frac{1}{3}, \frac{1}{9}, \dots, \frac{1}{3^n}, \dots$$

può essere generata con le seguenti relazioni ricorrenti:

$$(1) \begin{cases} p_n = \frac{10}{3} p_{n-1} - p_{n-2} \\ p_0 = 1; p_1 = \frac{1}{3} \end{cases}$$

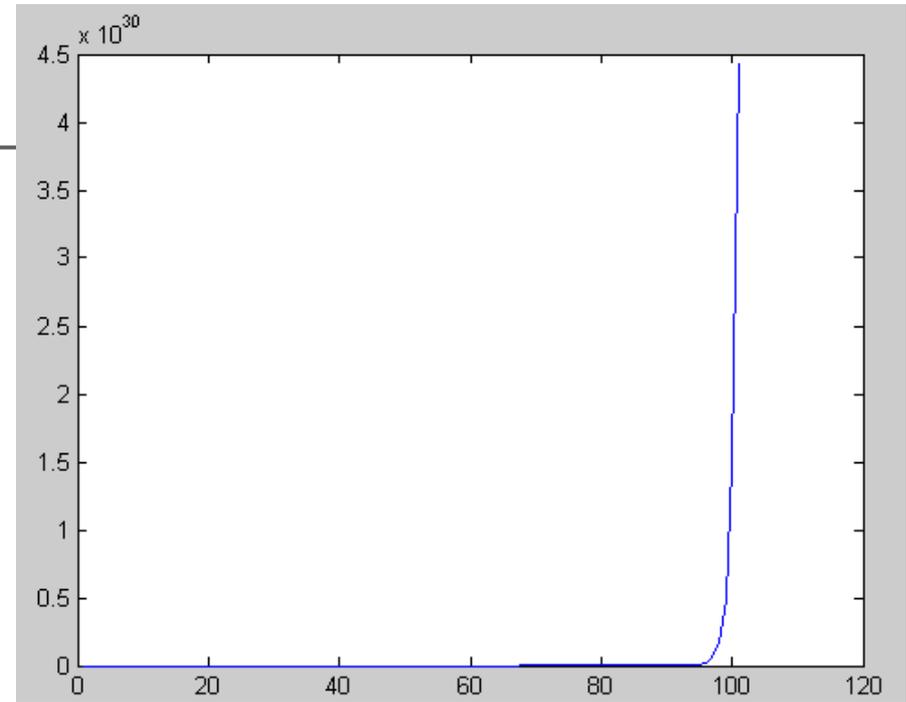
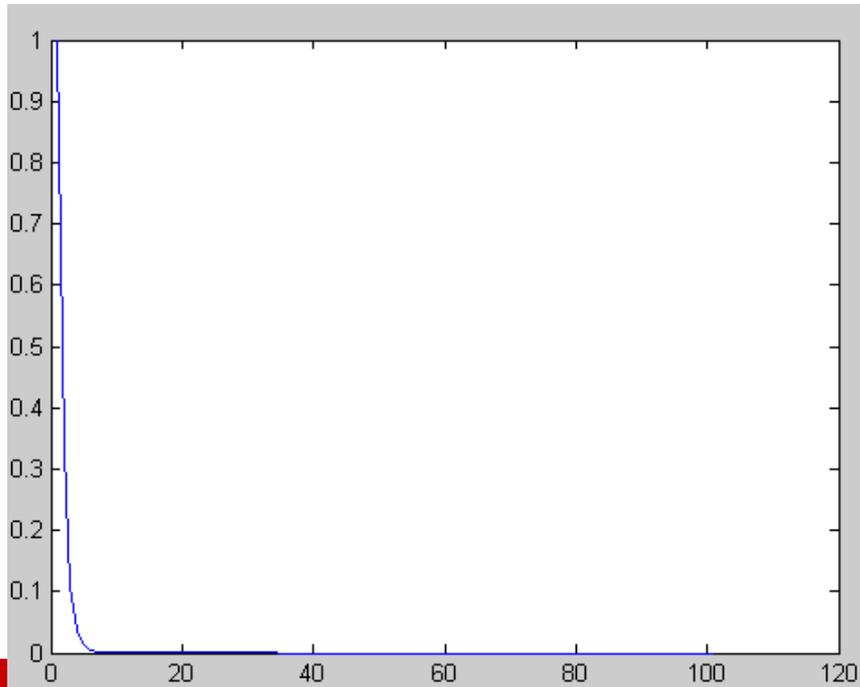
$$(2) \begin{cases} p_n = \frac{1}{3} p_{n-1} \\ p_0 = 1 \end{cases}$$

Generare i primi 100 termini della successione



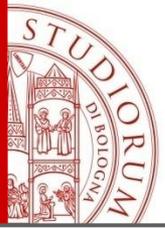
Relazione (1)

```
p1(1)=1;p1(2)=1/3;  
for i=2:100  
    p1(i+1)=10/3*p1(i)-p1(i-1)  
end  
plot(1:101,p1)
```



Relazione (2)

```
p2(1)=1;  
for i=1:100  
    p2(i+1)=1/3*p2(i);  
end  
plot(1:101,p2)
```



Analisi della propagazione dell'errore

$$\tilde{p}_0 = p_0 + \varepsilon, \quad \tilde{p}_1 = p_1 + \varepsilon,$$

$$\tilde{p}_2 = \frac{10}{3} \tilde{p}_1 - \tilde{p}_0 = \frac{10}{3} (p_1 + \varepsilon) - (p_0 + \varepsilon) = p_2 + \frac{7}{3} \varepsilon$$

$$\tilde{p}_3 = \frac{10}{3} \tilde{p}_2 - \tilde{p}_1 = \frac{10}{3} (p_2 + \frac{7}{3} \varepsilon) - (p_1 + \varepsilon) = p_3 + \frac{61}{9} \varepsilon$$

$$\tilde{p}_4 = \frac{10}{3} \tilde{p}_3 - \tilde{p}_2 = \frac{10}{3} (p_3 + \frac{61}{9} \varepsilon) - (p_2 + \varepsilon) = p_4 + \frac{583}{27} \varepsilon$$

.....

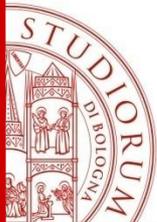
$$\tilde{p}_0 = p_0 + \varepsilon,$$

$$\tilde{p}_1 = \frac{1}{3} \tilde{p}_0 = \frac{1}{3} (p_0 + \varepsilon) = p_1 + \frac{1}{3} \varepsilon$$

$$\tilde{p}_2 = \frac{1}{3} \tilde{p}_1 = \frac{1}{3} (p_1 + \frac{1}{3} \varepsilon) = p_2 + \frac{1}{9} \varepsilon$$

$$\tilde{p}_3 = \frac{1}{3} \tilde{p}_2 = \frac{1}{3} (p_2 + \frac{1}{9} \varepsilon) = p_3 + \frac{1}{27} \varepsilon$$

.....



Costo computazionale: sistemi lineari

Sistemi lineari: $Ax=b$ n numero incognite

Ipotesi:

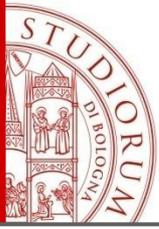
Tempo di esecuzione di 1 moltiplicazione circa 10^{-6} secondi

Regola di Cramer $O((n+1)!)$

(soluzione come quozienti di determinanti di ordine n)

Metodo di Gauss $O(n^3)$

n	Metodo di CRAMER	Metodo di GAUSS
13	24 ore	9.0×10^{-4} sec
14	15 giorni	1.1×10^{-3} sec
20	1.7×10^6 anni	3.1×10^{-3} sec



Esempio

Calcolo matrice per vettore

$$A=[a_{ij}], \quad i=1, \dots, m, \quad j=1, \dots, n$$

$$x=[x_j], \quad j=1, \dots, n$$

$$y = A \cdot x \quad y_i = \sum_{j=1}^n a_{ij} x_j \quad i = 1, \dots, m$$

Algoritmo

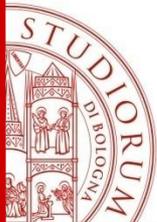
For i=1:m

$$y(i)=A(:,i)*x'; \quad \leftarrow$$

end

Ogni prodotto scalare costa:
n moltiplicazioni
n addizioni

Costo computazionale $2n \cdot m$. Se $n=m$ allora $O(n^2)$



Sistema Lineare

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n = b_3 \\ \dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n = b_m \end{array} \right.$$

Esiste \mathbf{x} che verifichi tutte le equazioni lineari simultaneamente?

Sistemi Lineari

$$\begin{array}{c}
 \begin{matrix}
 \text{matrice} \\
 \text{coefficienti}
 \end{matrix}
 \left[\begin{array}{cccc}
 a_{11} & a_{12} & \cdots & a_{1n} \\
 a_{21} & a_{22} & \cdots & a_{2n} \\
 \vdots & \vdots & \ddots & \vdots \\
 a_{m1} & a_{m2} & \cdots & a_{mn}
 \end{array} \right]
 \begin{matrix}
 \left(\begin{array}{c}
 x_1 \\
 x_2 \\
 \vdots \\
 x_n
 \end{array} \right) \\
 \mathbf{x}
 \end{matrix}
 =
 \begin{matrix}
 \left(\begin{array}{c}
 b_1 \\
 b_2 \\
 \vdots \\
 b_m
 \end{array} \right) \\
 \mathbf{b} \\
 \text{vettore} \\
 \text{termini noti}
 \end{matrix}
 \end{array}$$

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

vettore incognite

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad 1 \leq i \leq m$$

Sistemi Lineari: soluzioni

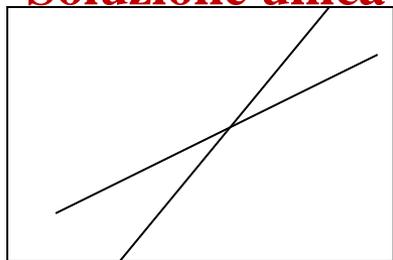
Problema della ricerca del punto (x,y) di intersezione di due rette nel piano

$$\begin{cases} a_{11}x + a_{12}y = b_1 \\ a_{21}x + a_{22}y = b_2 \end{cases}$$

Quante soluzioni ci sono?

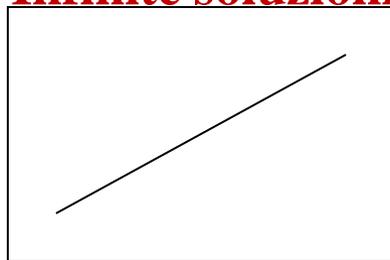
$$\begin{cases} 3x - 4y = 5 \\ 6x - 10y = 2 \end{cases}$$

Soluzione unica



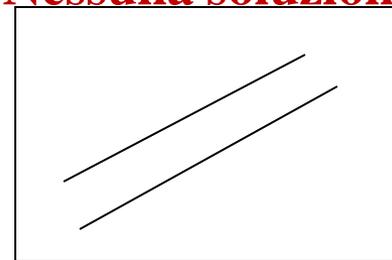
$$\begin{cases} 3x - 4y = 5 \\ 6x - 8y = 10 \end{cases}$$

Infinite soluzioni



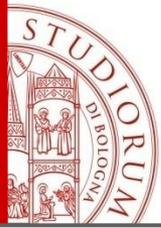
$$\begin{cases} 3x - 4y = 5 \\ 6x - 8y = 3 \end{cases}$$

Nessuna soluzione



Compatibile: il sistema ammette almeno una soluzione

Incompatibile: in caso contrario



Classificazione dei sistemi lineari

m equazioni n incognite

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

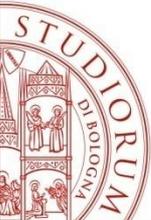
SISTEMA NORMALE (m=n)

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

**SISTEMA
INDETERMINATO**

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

**SISTEMA
SOVRADETERMINATO
(m>n)**



Soluzioni di sistemi lineari

Teorema (Rouche' Capelli)

Il sistema lineare $Ax=b$ ammette soluzione se e solo se la matrice dei coefficienti A e la matrice completa $[A \ b]$ hanno lo stesso rango:

$$\mathbf{Rank}(A)=r=\mathbf{Rank}([A \ b])$$

Se inoltre $r=n$ ► la soluzione è unica

Se invece $r < n$ ► esistono un numero infinito di soluzioni

r variabili incognite possono essere

espresse in termini di $n-r$ variabili con valori arbitrari

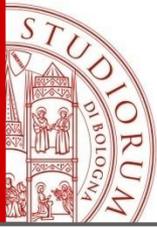
Altrimenti ($\mathbf{Rank}(A) \neq r \neq \mathbf{Rank}([A \ b])$) il sistema non ammette soluzioni

SISTEMA INDETERMINATO

($m < n$) non ci sono sufficienti informazioni per determinare un unico valore per tutte le incognite (r sempre $<$ di n)

SISTEMA SOVRADETERMINATO

($m > n$) ci sono piu' equazioni che incognite (metodo dei minimi quadrati)



Soluzioni di sistemi lineari $m=n$

Teorema

Il sistema lineare normale ($m=n$) $Ax=b$ ammette una ed una sola soluzione se e solo se A è non singolare

Se il sistema è omogeneo ($b=0$), A non singolare allora esiste solo la soluzione nulla $x=0$

A (nxn) è detta NON singolare se soddisfa una delle seguenti condizioni equivalenti:

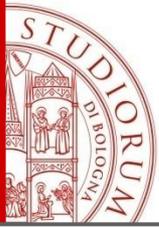
- 1) $\det(A) \neq 0$
- 2) Se esiste la matrice inversa A^{-1} di A
- 3) $\text{rank}(A)=n$

Altrimenti A è singolare.

Se A è non singolare allora esiste A^{-1} inversa di A

$$\underbrace{A^{-1}A}_I x = A^{-1}b$$

$$x = A^{-1}b$$



Esempio

$$7x = 21$$

$$x = \frac{21}{7} = 3.$$

$$x = 7^{-1} \times 21 = 0.142857 \times 21 = 2.99997$$

Il metodo dell'inversa
è meno **efficiente** e meno **accurato!**



Condizionamento di un sistema lineare

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

$\Delta \mathbf{A}$ matrice delle perturbazioni
 $\Delta \mathbf{b}$ }
 $\Delta \mathbf{x}$ } vettori delle perturbazioni

1. $\mathbf{A} (\mathbf{x} + \Delta \mathbf{x}) = \mathbf{b} + \Delta \mathbf{b}$ perturbazione del termine noto
2. $(\mathbf{A} + \Delta \mathbf{A}) (\mathbf{x} + \Delta \mathbf{x}) = \mathbf{b}$ perturbazione matrice coeff.
3. $(\mathbf{A} + \Delta \mathbf{A}) (\mathbf{x} + \Delta \mathbf{x}) = \mathbf{b} + \Delta \mathbf{b}$

Condizionamento di un sistema lineare: caso 1

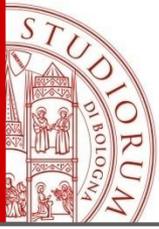
$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

$$\mathbf{A} (\mathbf{x} + \Delta \mathbf{x}) = \mathbf{b} + \Delta \mathbf{b} \quad \text{sottraendo}$$

$$\mathbf{A} \Delta \mathbf{x} = \Delta \mathbf{b} \quad \longrightarrow \quad \Delta \mathbf{x} = \mathbf{A}^{-1} \Delta \mathbf{b}$$

$$\|\Delta \mathbf{x}\| = \|\mathbf{A}^{-1} \Delta \mathbf{b}\| \leq \|\mathbf{A}^{-1}\| \|\Delta \mathbf{b}\| \quad \longrightarrow \quad \|\Delta \mathbf{x}\| \leq \|\mathbf{A}^{-1}\| \|\Delta \mathbf{b}\|$$

$$\|\mathbf{b}\| = \|\mathbf{A} \mathbf{x}\| \leq \|\mathbf{A}\| \|\mathbf{x}\| \quad \longrightarrow \quad \frac{1}{\|\mathbf{x}\|} \leq \|\mathbf{A}\| \frac{1}{\|\mathbf{b}\|}$$



Condizionamento di un sistema lineare: caso 1

$$\frac{\|\Delta x\|}{\|x\|} \leq \underbrace{\|A^{-1}\| \|A\|}_{\text{Indice/numero di condizionamento}} \frac{\|\Delta b\|}{\|b\|}$$

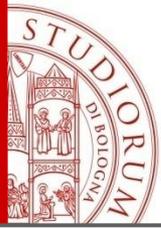
Errore relativo nella soluzione

cond(A)
 $\mu(A)$
 $K(A)$

Errore relativo nei dati

Fattore di amplificazione delle perturbazioni (relative) introdotte in b

Condizionamento di un sistema lineare: caso 2



$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b}$$

\mathbf{A} non singolare

$\mathbf{A} + \Delta\mathbf{A}$ non singolare

$$\mathbf{Ax} = \mathbf{b} = (\mathbf{A} + \Delta\mathbf{A})(\mathbf{x} + \Delta\mathbf{x})$$

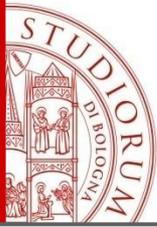
$$0 = \mathbf{A}\Delta\mathbf{x} + \Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})$$

$$\Rightarrow \Delta\mathbf{x} = -\mathbf{A}^{-1}\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})$$

$$\Rightarrow \|\Delta\mathbf{x}\| = \|\mathbf{A}^{-1}\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})\| \leq \|\mathbf{A}^{-1}\| \|\Delta\mathbf{A}\| \|\mathbf{x} + \Delta\mathbf{x}\|$$



$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x} + \Delta\mathbf{x}\|} \leq \|\mathbf{A}^{-1}\| \|\mathbf{A}\| \frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}$$

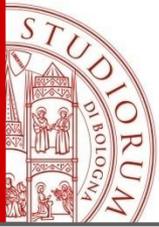


Numero di condizionamento

- **$K(A)$ piccolo** (ordine n^p $p=0,1,2,3$):
il problema/matrice è ben condizionato
- **$K(A)$ grande** (ordine 10^n):
Il problema/matrice è mal condizionato
(es. Matrice di Hilbert)

E' una misura di quanto una matrice è vicina ad essere singolare.

$K(A)$ grande $\rightarrow A$ è quasi singolare



Indice di condizionamento PROPRIETA'

$$K(A) = \|A\| \|A^{-1}\|$$

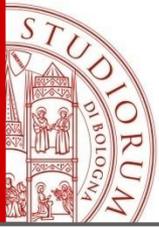
$$1. \quad K(A) \geq 1$$

$$\text{Infatti : } K(A) = \|A\| \|A^{-1}\| \geq \|AA^{-1}\| = \|I\| = 1$$

$$2. \quad K_2(A) = 1 \Leftrightarrow A = \alpha Q,$$

Q unitaria/ortogonale ($Q^T = Q^{-1}$)

$$\text{Infatti : } \|Q\|_2 = \sqrt{\rho(Q^T Q)} = \sqrt{\rho(I)} = 1$$



Indice di condizionamento PROPRIETA'

$$3. \quad \text{Se } A \text{ simmetrica, } K_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$$

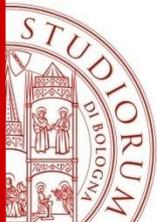
$$\|A\|_2 = \sqrt{\rho(A^T A)} = \sqrt{\rho(A^2)} = \sqrt{\rho^2(A)} = \rho(A) = |\lambda_{\max}|$$

$$\|A^{-1}\|_2 = \frac{1}{|\lambda_{\min}|}$$

Se inoltre A è hermetiana/simmetrica e definita positiva

$$\|A\|_2 = \lambda_{\max} \quad \|A^{-1}\|_2 = \frac{1}{\lambda_{\min}}$$

La matrice A è tanto meglio condizionata quanto più vicini sono tra loro i suoi autovalori.



Matrice di Hilbert

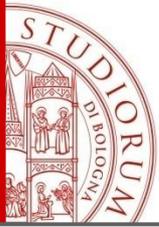
$$a_{ij}^{(n)} = \frac{1}{i+j-1}$$

$$i, j = 1, 2, \dots, n$$

$$n=5$$

$$A^{(5)} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}$$

$$[A^{(5)}]^{-1} = \begin{bmatrix} 25 & -300 & 1050 & -1400 & 630 \\ -300 & 4800 & -18900 & 26880 & -12600 \\ 1050 & -18900 & 79380 & -117600 & 56700 \\ -1400 & 26880 & -117600 & 179200 & -88200 \\ 630 & -12600 & 56700 & -88200 & 44100 \end{bmatrix}$$



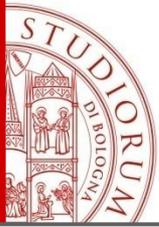
Matrice di Hilbert

$$\mu(A) = \|A\| \|A^{-1}\| \approx e^{3.5n}$$

n	$\mu_2(A^{(n)})$	$\mu_\infty(A^{(n)})$
2	1.505	27
3	$5.241 \cdot 10^2$	$7.480 \cdot 10^2$
4	$1.551 \cdot 10^4$	$2.837 \cdot 10^4$
5	$4.766 \cdot 10^5$	$9.436 \cdot 10^5$
6	$1.495 \cdot 10^7$	$2.907 \cdot 10^7$
7	$4.754 \cdot 10^8$	$9.852 \cdot 10^8$
8	$1.526 \cdot 10^{10}$	$3.387 \cdot 10^{10}$
9	$4.932 \cdot 10^{11}$	$1.099 \cdot 10^{12}$
10	$1.603 \cdot 10^{13}$	$3.535 \cdot 10^{13}$

Norma 2

Norma infinito



Sistemi mal condizionati: esempio

esatto
$$\begin{cases} 1.000x + 2.000y = 3.000 \\ 0.499x + 1.001y = 1.5 \end{cases} \quad x = 1; y = 1$$

Perturbiamo sulla terza e quarta cifra decimale i coefficienti:

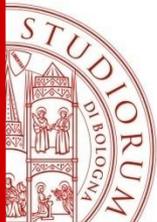
perturbato
$$\begin{cases} 1.000x + 2.000y = 3.000 \\ 0.500x + 1.002y = 1.5 \end{cases} \quad x = 3; y = 0$$

$$A = \begin{bmatrix} 1.000 & 2.000 \\ 0.499 & 1.001 \end{bmatrix} \quad x = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad K(A) = 2083$$

$$AA = \begin{bmatrix} 1.000 & 2.000 \\ 0.500 & 1.002 \end{bmatrix} \quad xx = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \quad K(AA) = 3127$$

$$\frac{\|AA - A\|}{\|A\|} \approx 5.7e - 4 \quad \frac{\|xx - x\|}{\|x\|} \approx 1.58$$

Perturbando poco i dati del problema abbiamo modificato completamente il risultato



Sistemi ben condizionati: esempio

esatto
$$\begin{cases} 2.000x - 1.000y = -1.000 \\ -1.000x + 2.000y = 5.000 \end{cases} \quad x = 1; y = 3$$

Perturbiamo sulla terza e quarta cifra decimale i coefficienti:

perturbato
$$\begin{cases} 2.000x - 1.000y = -1.000 & x = 0.9993; \\ -1.001x + 2.001y = 5.000 & y = 2.9987 \end{cases}$$

$$A = \begin{bmatrix} 2.000 & -1.000 \\ -1.000 & 2.000 \end{bmatrix} \quad x = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad K(A) = 3$$

$$AA = \begin{bmatrix} 2.000 & -1.000 \\ -1.001 & 2.001 \end{bmatrix} \quad xx = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad K(AA) = 3.001$$

$$\frac{\|AA - A\|}{\|A\|} \approx 4.7e-4 \quad \frac{\|xx - x\|}{\|x\|} \approx 4.7e-4$$

In questo caso la perturbazione sui dati è accettabile



Esempio

DATI forniti da:

US Air Force Phillips Laboratory

Immagine originale: 256x256pixel

Vettore \underline{x} $65536 = 256^2$

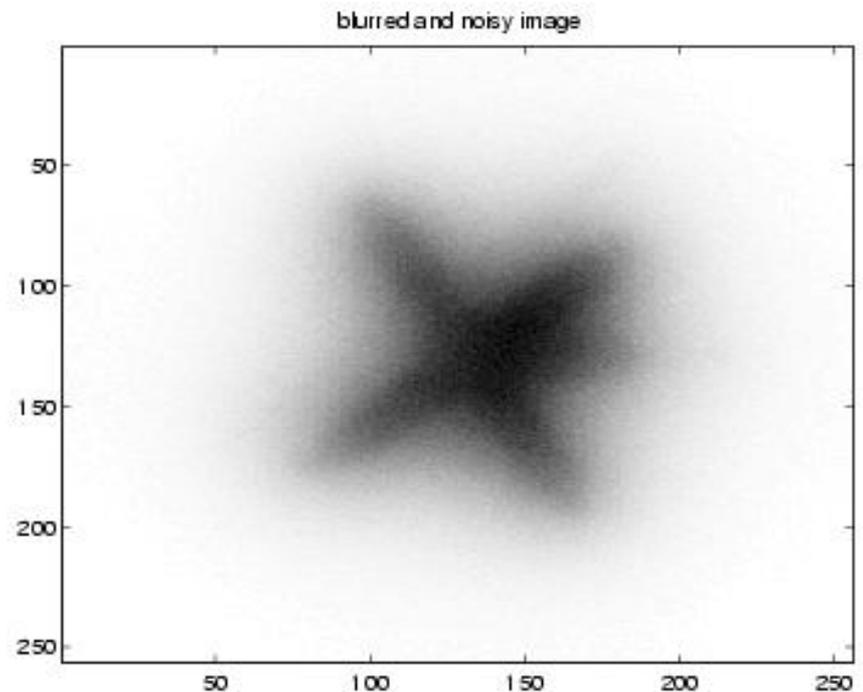
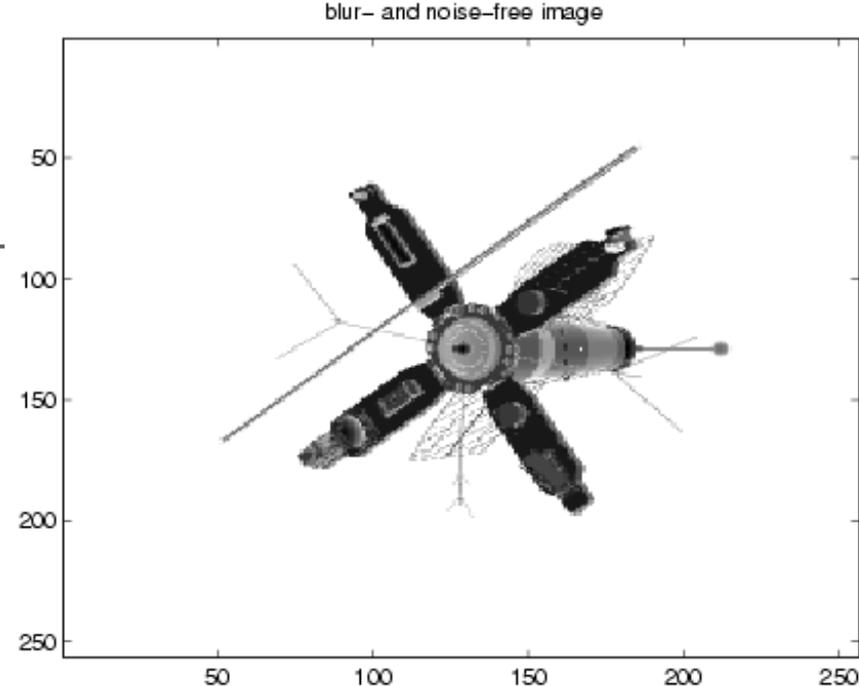
A operatore di sfuocamento
 65536×65536 ;

Immagine perturbata \underline{b}

$$\underline{Ax} = \underline{b} \quad \underline{b} = \underline{b} + \text{err}$$

Determinare un'approssimazione
dell'immagine \underline{x} , risolvendo:

$$\underline{Ax} = \underline{b}$$



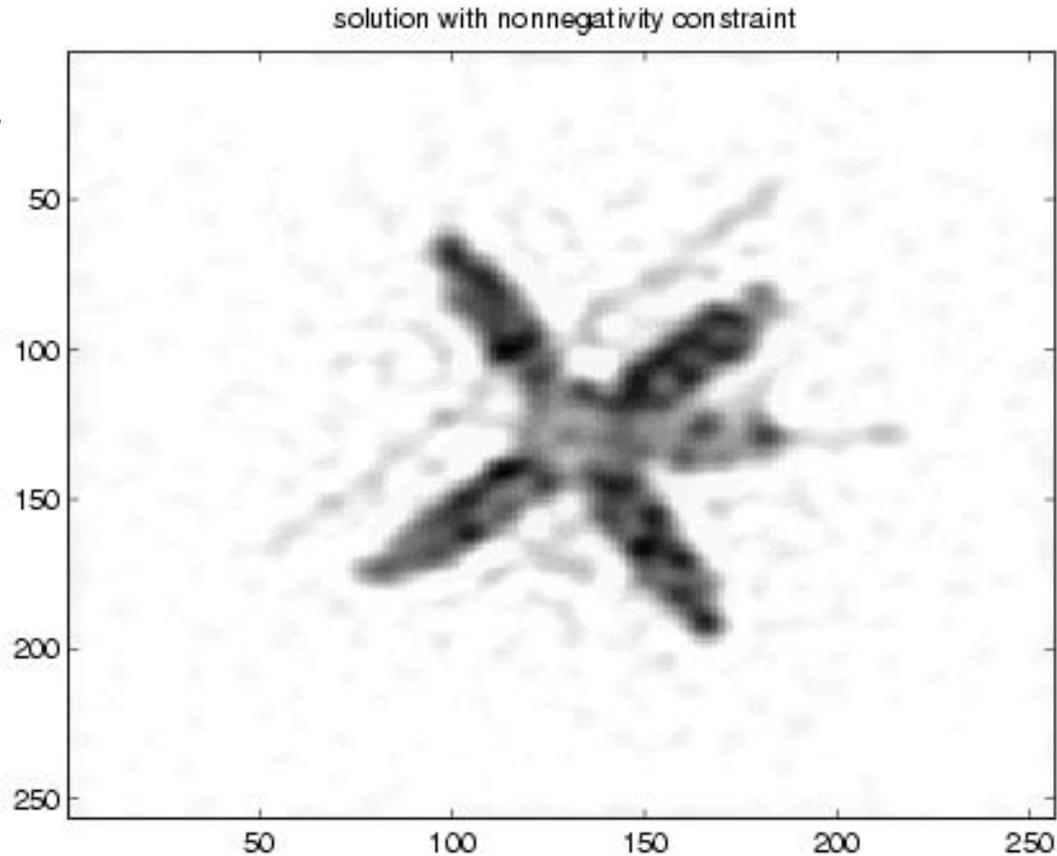
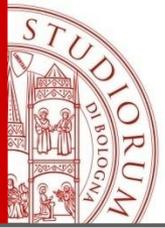


Immagine ricostruita come soluzione del sistema

$$\mathbf{Ax}=\mathbf{b}$$

$$\text{Vettore } x \text{ } 65536 = 256^2$$



Metodi Numerici per risolvere sistemi lineari $m=n$

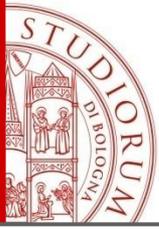
METODI DIRETTI

L'esatta soluzione viene costruita, in assenza di errori di arrotondamento nei dati e nei calcoli, in un numero finito di passi.

METODI ITERATIVI

La soluzione e' ottenuta come limite di una successione di soluzioni di problemi lineari più semplici.

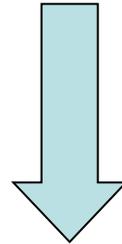
Diversamente dal caso precedente la matrice A non viene modificata durante il calcolo. Anche in assenza di errori di arrotondamento si deve comunque operare un troncamento del procedimento risolutivo commettendo un errore



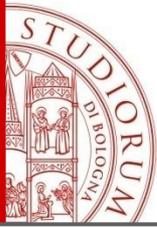
Metodo di eliminazione di Gauss

$$Ax=b$$

Se fosse possibile decomporre $A=LU$, A non singolare
 L triangolare inferiore, U triangolare superiore



$$LUx = b \quad \Leftrightarrow \begin{cases} Ly = b \\ Ux = y \end{cases}$$



Sistema triangolare inferiore

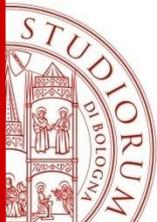
$$\begin{cases} a_{11}x_1 & = b_1 \\ a_{21}x_1 + a_{22}x_2 & = b_2 \\ \dots & \\ a_{i1}x_1 + a_{i2}x_2 \dots + a_{ii}x_i & = b_3 \\ \dots & \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n & = b_n \end{cases}$$

Metodo di sostituzione in avanti (forward substitution)

Costo computazionale

$$\begin{cases} x_1 = \frac{b_1}{a_{11}} \\ x_k = (b_k - \sum_{j=1}^{k-1} a_{kj}x_j) / a_{kk} \quad k = 2, \dots, n \end{cases} \quad a_{ii} \neq 0, i = 1, \dots, n$$

$$\approx \frac{n^2}{2}$$



Esempio 1:

data A trovare L e U tale che $A=LU$

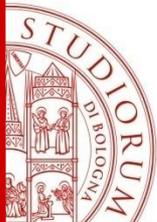
$$A_1 = A = \begin{bmatrix} 2 & 1 & 0 \\ 4 & 5 & 2 \\ 6 & 15 & 12 \end{bmatrix}$$

STEP 1

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix} \quad A_2 = L_1 A_1 = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 3 & 2 \\ 0 & 12 & 12 \end{bmatrix}$$

STEP 2

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{bmatrix} \quad A_3 = L_2 A_2 = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 3 & 2 \\ 0 & 0 & 4 \end{bmatrix} = U \quad L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{bmatrix}$$



Esempio 2:

data A trovare L e U tale che $A=LU$

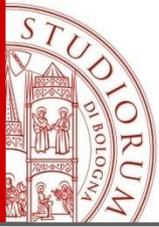
$$A_1 = A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 4 & 2 & 1 \end{bmatrix}$$

STEP 1

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \quad A_2 = L_1 A_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -2 & 0 \\ 0 & -2 & -3 \end{bmatrix}$$

STEP 2

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \quad A_3 = L_2 A_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix} = U \quad L = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 4 & 1 & 1 \end{bmatrix}$$



Fattorizzazione LU

$$A_3 = L_2 A_2$$

$$A_3 = L_2 L_1 A_1$$

$$A_3 = L_2 L_1 A$$

1. A_3 triangolare superiore con elementi diag. non zero
2. L_i triangolare inferiore è non sing. \leftrightarrow Elem. Diag non zero
3. L_i^{-1} è triangolare inferiore

$$A = [L_1]^{-1} [L_2]^{-1} A_3$$

$$A = \underbrace{[L_2 L_1]^{-1}}_L A_3 = LU$$

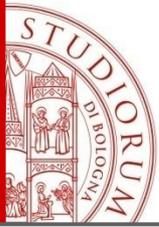
$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{bmatrix} \quad L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{bmatrix}$$

Fattorizzazione LU

$$A=LU$$

$$L = \begin{bmatrix} 1 & 0 & \cdot & \cdot & 0 \\ m_{21} & 1 & 0 & \cdot & \cdot \\ m_{31} & m_{32} & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 0 \\ m_{n1} & m_{n2} & m_{n3} & \cdot & 1 \end{bmatrix} \quad U = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdot & a_{1n} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdot & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdot & a_{3n}^{(3)} \\ 0 & 0 & 0 & \cdot & \cdot \\ 0 & 0 & 0 & 0 & a_{nn}^{(n)} \end{bmatrix}$$

Gli elementi di A_n sulla diagonale principale si chiamano **PERNI** e devono essere diversi da zero affinché si abbia fattorizzazione LU



Esempio 1

$$b_1 = b = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$$

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix}$$

$$L_1 b_1 = \begin{bmatrix} 2 \\ -1 \\ -2 \end{bmatrix} = b_2$$

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{bmatrix}$$

$$L_2 b_2 = \begin{bmatrix} 2 \\ -1 \\ 2 \end{bmatrix} = b_3 = y$$

Esempio 2

$$b_1 = b = \begin{bmatrix} 4 \\ 6 \\ 9 \end{bmatrix}$$

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \quad L_1 b_1 = \begin{bmatrix} 4 \\ 2 \\ -7 \end{bmatrix} = b_2$$

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \quad L_2 b_2 = \begin{bmatrix} 4 \\ 2 \\ -9 \end{bmatrix} = b_3 = y$$

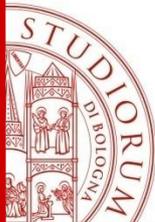
$$b_3 = L_2 b_2$$

$$b_3 = L_2 L_1 b_1$$

$$b_3 = L_2 L_1 b$$

$$b = [L_1]^{-1} [L_2]^{-1} b_3$$

$$b = \underbrace{[L_2 L_1]^{-1}}_L b_3 = Ly$$



Metodo di eliminazione di Gauss

STEP 1 $L_1[A \mid b]$

y si ottiene applicando successivamente le trasformazioni elementari di Gauss L_i al vettore b , così come U si ottiene applicando successivamente tali trasformazioni alla matrice A .

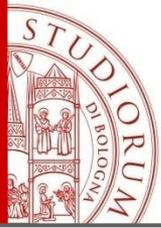
STEP 2 $L_2L_1[A \mid b]$

..... Dopo $n-1$ passi

$$L_{n-1}L_{n-2}\cdots L_2L_1[A \mid b] = [A_n \mid b_n] = [U \mid y]$$

Il metodo di Gauss consiste nella fattorizzazione LU della matrice $[A|b]$ (che risolve anche $Ly=b$) e nella risoluzione con sostituzione all'indietro del sistema triangolare:

$$Ux = y$$



Metodo di eliminazione di Gauss

$$L_{n-1}L_{n-2}\dots L_2L_1[A \mid b] = [A_n \mid b_n]$$

$$[A \mid b] = [L_1]^{-1}[L_2]^{-1}[L_3]^{-1}\dots[L_{n-1}]^{-1}[A_n \mid b_n]$$

$$[A \mid b] = \underbrace{[L_{n-1}L_{n-2}L_{n-3}\dots L_1]^{-1}}_L \underbrace{[A_n \mid b_n]}_{U \mid y} = L[U \mid y]$$

→ $A = LU$

→ $Ly = b$

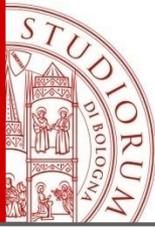
$$Ux = y$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ -9 \end{bmatrix}$$

$$x_3 = 3$$

$$x_2 = -1$$

$$x_1 = 2$$



Metodo di Eliminazione di Gauss

Data una matrice A tale che $\det(A_k) \neq 0, \forall k = 1, \dots, n$ e quindi in particolare non singolare, ed un vettore $a_{i,n+1}, i=1, \dots, n$, (colonna dei termini noti), si ha:

For $k=1, \dots, n-1$

For $i=k+1, \dots, n$

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$$

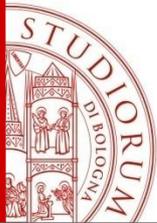
For $j=k+1, \dots, n+1$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}$$

end j

end i

end k



Costo computazionale

Fattorizzazione

Divisioni per ottenere L: $(n-1) + (n-2) + \dots + 1 = \frac{n(n-1)}{2}$

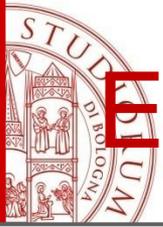
Moltiplicazioni per ottenere U :
 $(n-1)^2 + (n-2)^2 + \dots + (n-(n-1))^2 = \frac{(n-1)n(2n-1)}{6}$

Totale:
 $\frac{n(n-1)}{2} + \frac{(n-1)n(2n-1)}{6} \approx \frac{1}{3}n^3$

Risoluzione del sistema lineare triangolare $\frac{1}{2}n^2$

Perciò in totale la risoluzione del sistema lineare costa

$$O\left(\frac{1}{3}n^3 + \frac{1}{2}n^2\right)$$



ESISTENZA della Fattorizzazione LU

SE $\det(A) \neq 0$ **ESISTE SEMPRE UNA FATTORIZZAZIONE $A=LU$???**

ESEMPIO:

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad A^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Ma non esistono matrici L, U tali che $A=LU$

Condizione per avere una fattorizzazione LU di A :

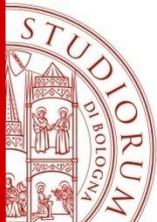
I PERNI $a_{kk}^{(k)}$, $k=1, \dots, n$, sono tutti diversi da zero



I minori principali di ordine k , $k=1, 2, \dots, n$, sono diversi da zero



Allora esiste una ed una sola fattorizzazione LU di A



Strategia del perno nullo

- Un sistema lineare normale $Ax=b$ con A non singolare ammette sempre soluzione
- Come garantire che Gauss non fallisca per una A generica con la sola condizione di essere non singolare?

Esempio

$$\begin{cases} 3y = 5 \\ x + 2y = 0 \end{cases} \quad A = \begin{bmatrix} 0 & 3 \\ 1 & 2 \end{bmatrix}, \quad \det(A) = -3$$

equivalente a

$$\begin{cases} x + 2y = 0 \\ 3y = 5 \end{cases} \quad A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}, \quad a_{11} \neq 0 \quad b = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$$

Pivoting

Soluzione

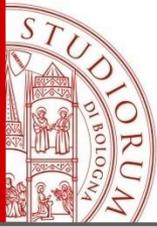
Scambiare di posto 2 equazioni

$$\begin{bmatrix} a_{kk}^{(k)} \\ a_{k+1,k}^{(k)} \\ a_{k+2,k}^{(k)} \\ \dots \\ a_{nk}^{(k)} \end{bmatrix}$$

$$\leftarrow = 0$$

Se non esistesse un $a_{ik}^{(k)} \neq 0 \quad i = k+1, k+2, \dots, n$

allora A sarebbe singolare



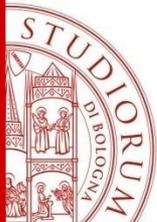
Pivoting – Strategia per avere un algoritmo stabile

Pivoting parziale (scambio righe)

Scegliere r uguale al più piccolo intero $\geq k$ tale che

$$\left| a_{rk}^{(k)} \right| = \max_{k \leq i \leq n} \left| a_{ik}^{(k)} \right|$$

e, se $r \neq k$, scambiare l'equazione k -esima con la r -esima



Matrici di Permutazione

- Differiscono dalla matrice Identità per lo scambio di righe
- Premoltiplicazione \Rightarrow scambio di righe,
Postmoltiplicazione \Rightarrow scambio di colonne

Esempio: P_1 scambia riga 1 e 3

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} a_{31} & a_{32} & a_{33} & a_{34} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{11} & a_{12} & a_{13} & a_{14} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

- Permutazioni multiple P

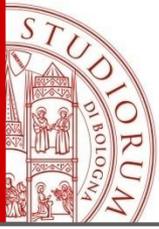
$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} a_{31} & a_{32} & a_{33} & a_{34} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ a_{11} & a_{12} & a_{13} & a_{14} \end{bmatrix}$$

Teorema

Sia $A(n \times n)$ non singolare, allora esiste una matrice di permutazione $P(n \times n)$ non singolare per cui $PA=LU$

Infatti:

Se A è non singolare, allora esiste per ogni ordine almeno una sottomatrice non singolare, quindi applicando su A opportuni scambi di righe e/o colonne si può fare in modo che A abbia le sottomatrici principali non singolari.



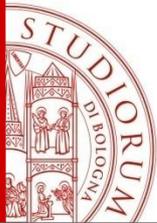
Metodo di Gauss con Pivoting

$$Ax = b$$

$$PAx = Pb \quad (PA = LU)$$

$$LUx = Pb$$

$$\begin{cases} Ly = Pb \\ Ux = y \end{cases} \Rightarrow \begin{cases} y = L^{-1}Pb \\ U = L^{-1}PA \end{cases}$$



Calcolo matrice inversa

Il problema di determinare l'inversa A^{-1} di una matrice A quadrata $n \times n$ non singolare si riconduce al problema di risolvere n sistemi normali.

$$A \cdot A^{-1} = I$$

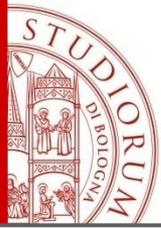
$$A^{-1} = (x_1, x_2, \dots, x_n), I = (e_1, e_2, \dots, e_n)$$

$$e_j = [0 \dots 0 \underset{j}{1} 0 \dots 0]^T$$

Risolvere: $Ax_j = e_j, \quad j = 1, \dots, n$

La matrice A è la medesima: fattorizzare $A=LU$ una sola volta

Costo n^3



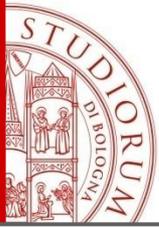
Fattorizzazione di Cholesky per sistemi lineari $Ax=b$

Teorema: Se A è una matrice **simmetrica definita positiva** allora esiste ed è unica una matrice R triangolare inferiore, con elementi positivi sulla diagonale principale tale che:

$$A = RR^T$$

Calcolo della soluzione del sistema lineare $Ax=b$:

$$\underbrace{RR^T}_y x = b \quad \Leftrightarrow \begin{cases} Ry = b \\ R^T x = y \end{cases}$$



Fattorizzazioni di matrici

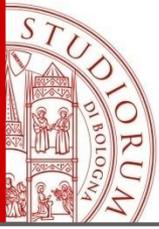
1. Fattorizzazione **LU** (algoritmo di **Gauss**)
2. Fattorizzazione **RR^T** (algoritmo di **Cholesky**)
3. Fattorizzazione **QR** (algoritmo **Householder**)

L = matrice triangolare inferiore

U = matrice triangolare superiore

R = matrice triangolare inferiore

Q = matrice ortogonale (**QQ^T=I**)

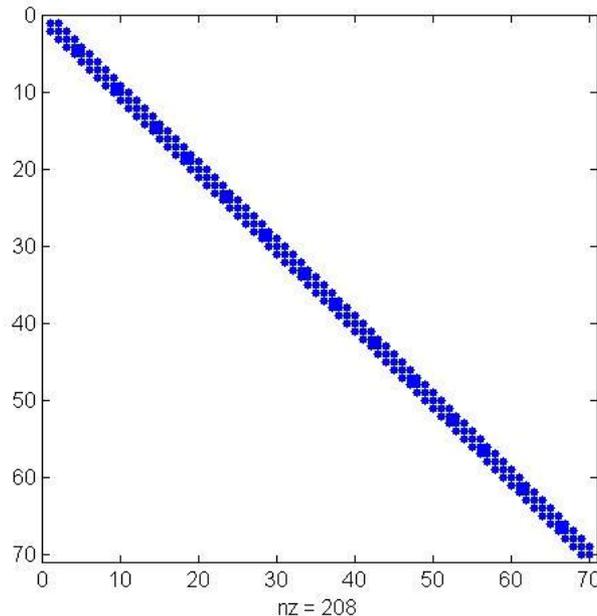


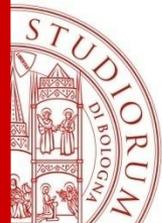
Matrici sparse

Una matrice è ritenuta **sparsa** quando il numero degli elementi che possono essere diversi dallo zero è di ordine $O(n)$

Esempio:

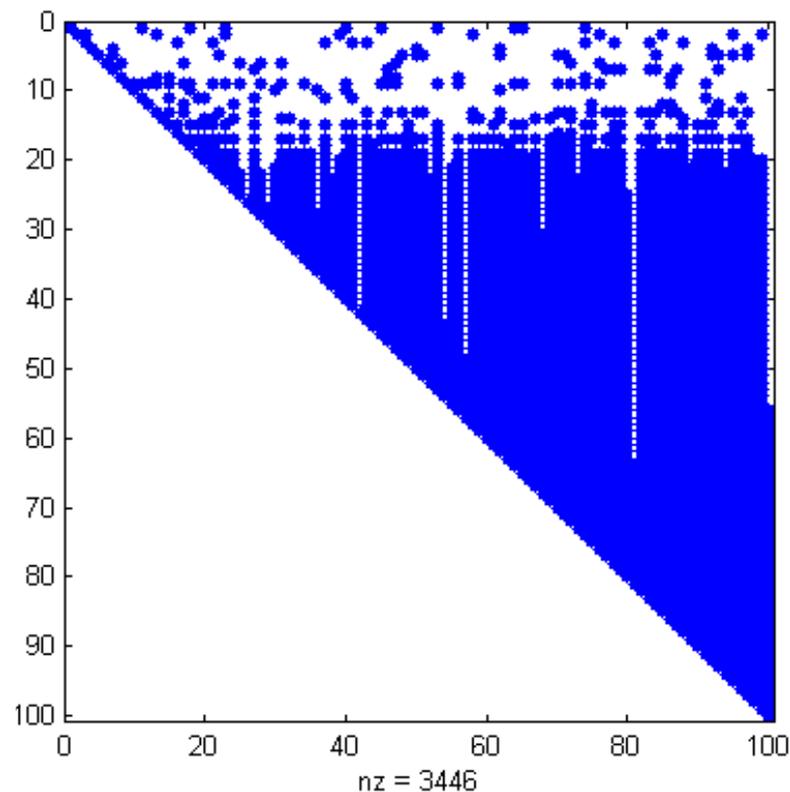
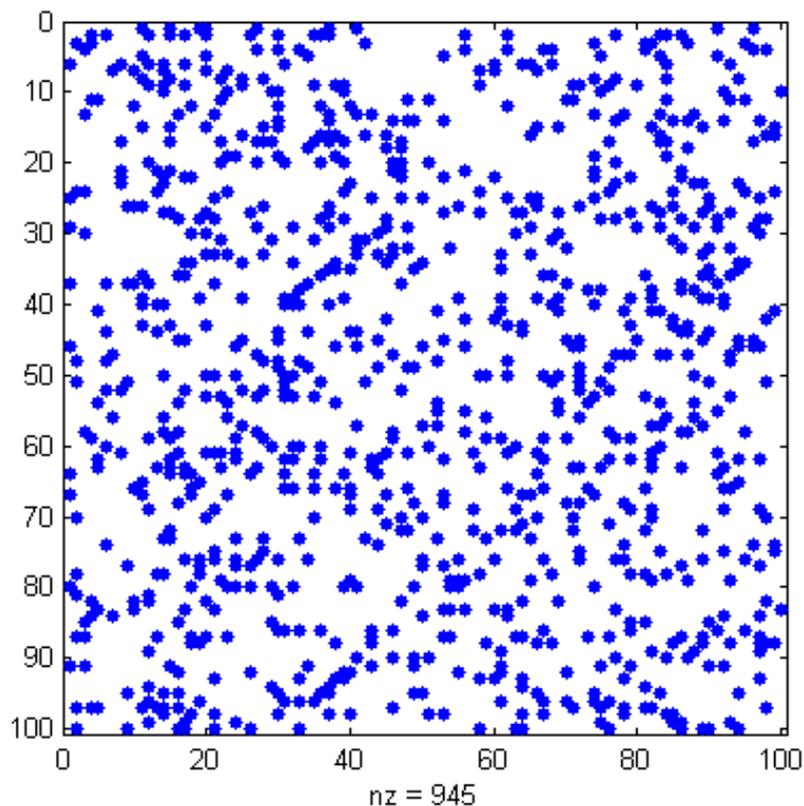
matrice tridiagonale `>>A=diag([1:70])+diag([1:69],-1)+diag([1:69],1);`
`>>spy(A)`

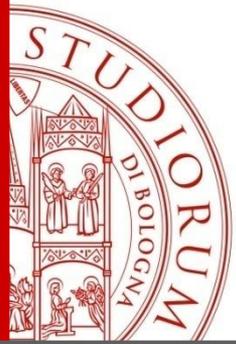




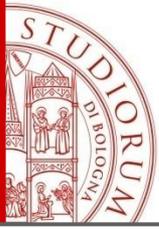
Matrici sparse, effetto fill-in

```
>> A=sprand(100,100,0.1)
>> [L,U,P]=lu(A);
>> spy(A);
>> spy(U);
```



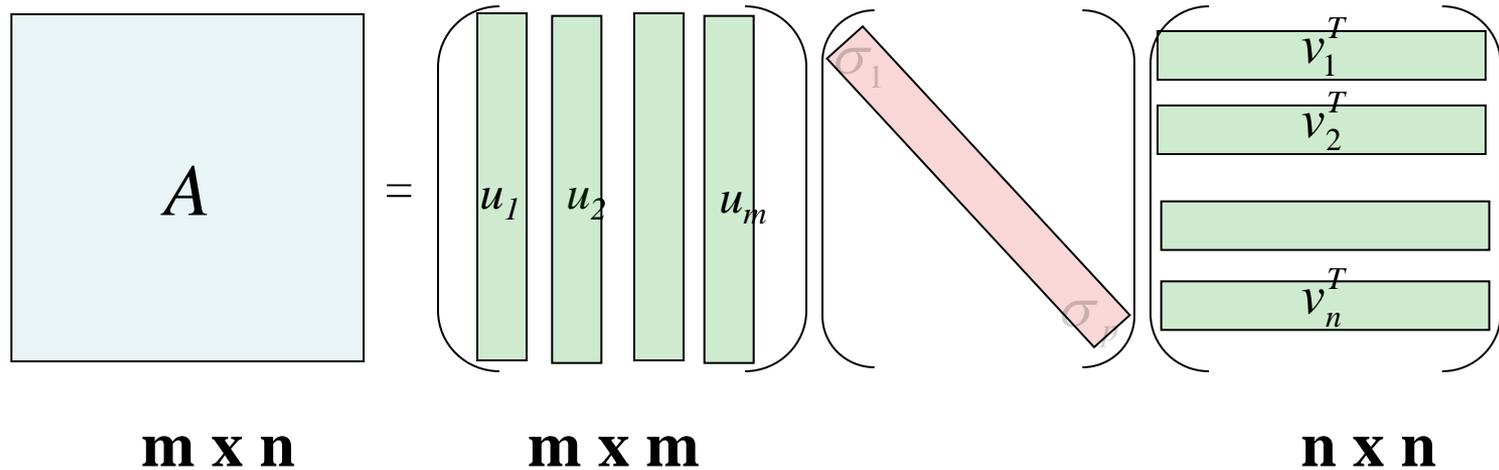


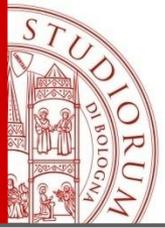
Decomposizione in valori singolari



Decomposizione in valori singolari

$$A = U \Sigma V^T$$





Decomposizione in valori singolari (SVD)

Data una matrice $A \in R^{m \times n}$ esistono due matrici ortogonali

$$U \in R^{m \times m} \quad V \in R^{n \times n}$$

tali che $A = U \Sigma V^T$, (quindi $\Sigma = U^T A V$)

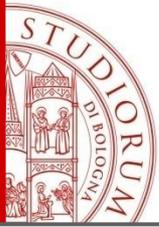
dove la matrice

$$\Sigma \in R^{m \times n} \quad \Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

è diagonale con

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0, \quad p = \min(m, n)$$

chiamati **VALORI SINGOLARI** di A



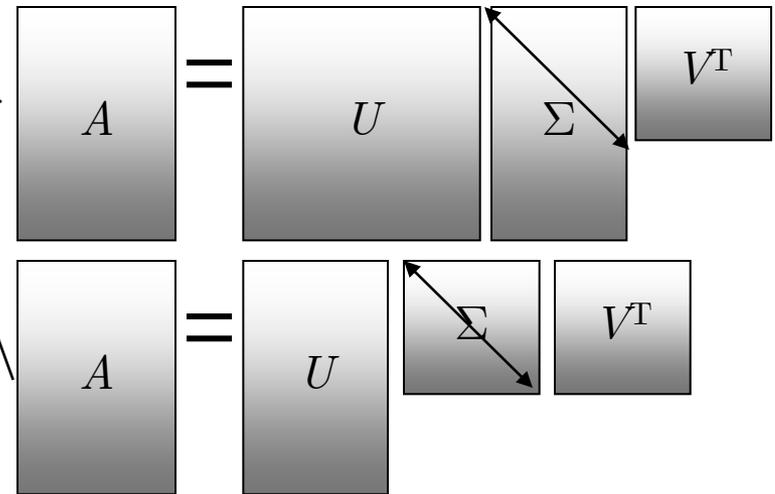
Decomposizione in valori singolari (SVD)

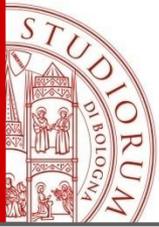
- σ_i^2 sono gli autovalori di $A^T A$
- Le colonne di U (di V) rappresentano gli **AUTOVETTORI** di AA^T (di $A^T A$)
- I vettori u_i, v_i sono definiti rispettivamente **VETTORI SINGOLARI SINISTRI E DESTRI**
- Se A è **hermetiana def. pos.** I valori singolari sono anche gli autovalori di A e gli autovettori sono le colonne di V .

SVD-economy

$$A_{m,n} = U_{m,m} \Sigma_{m,n} V_{n,n}^T$$

$$A_{m,n} = U_{m,n} \Sigma_{n,n} V_{n,n}^T$$





1) Approssimazione di A lower-rank

Approssimare una matrice con somma di matrici più semplici

$$A = U\Sigma V^T \quad A = E_1 + E_2 + \dots + E_p \quad p = \min(m, n)$$

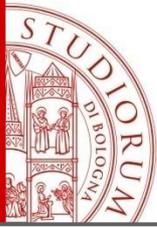
$$E_i = \sigma_i u_i v_i^T \quad \text{matrice a rango 1} \quad \|E_i\| = \sigma_i$$

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_p = 0, \quad r < p$$

allora **r** è il **rango** di A e

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T$$

dove u_i e v_i sono le colonne delle due matrici U e V



Approssimazione di A lower-rank

TEOREMA

Sia data la SVD di una matrice A, supponiamo che il rango di A sia r.
Se, fissato un intero positivo k, $k < r$, definiamo

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T$$

A_k è un'approssimazione di A di rango k.

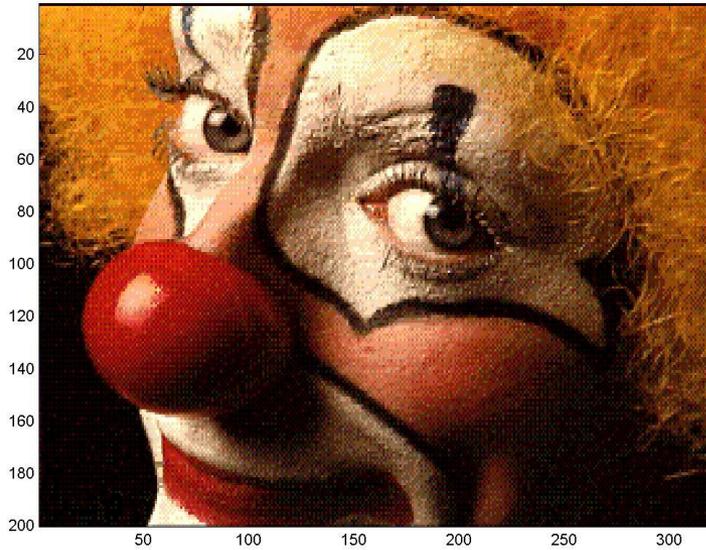
Errore che si commette nel considerare A_k anziché A:

$$\min_{B \in \beta} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$$
$$\beta = \{B \in R^{m \times n} : \text{rango}(B) = k\}$$

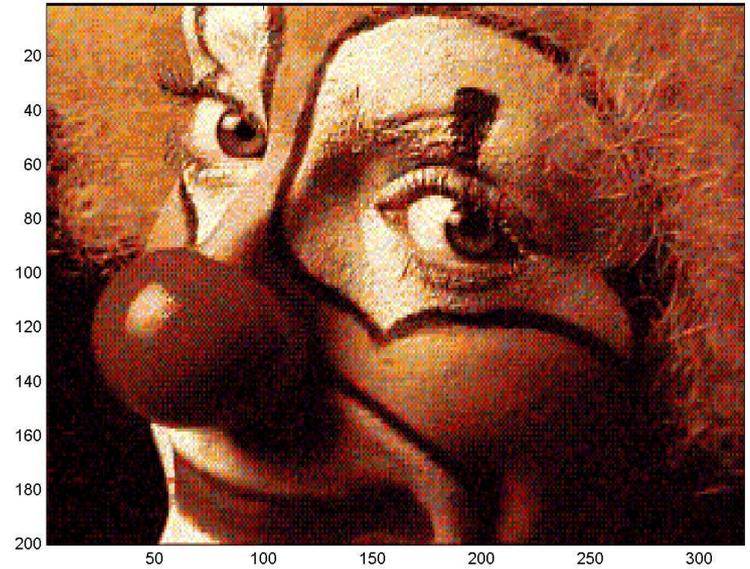
Il più piccolo valore singolare non nullo di A è la distanza di A dall'insieme di tutte le matrici a rango deficiente.

σ_i in ordine decrescente -> approx. migliora per k che aumenta

SVD: esempio compressione



(a)



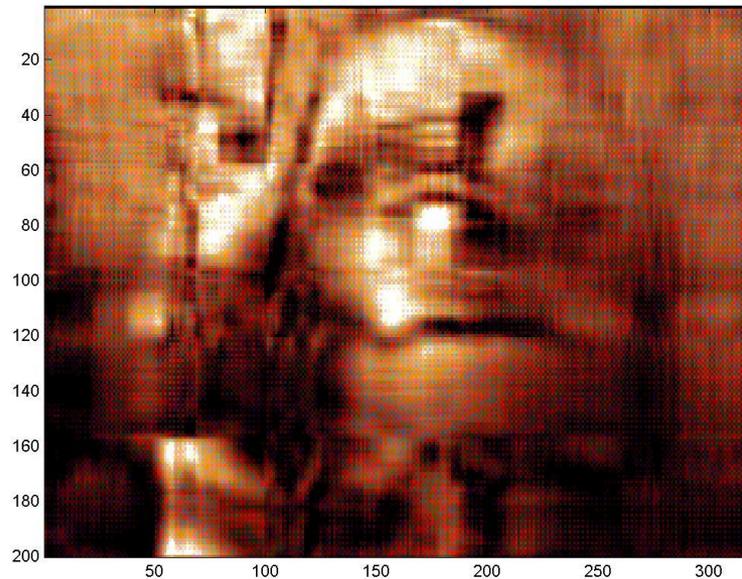
(b)

Numero diadi

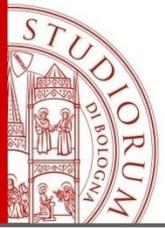
(a) 200

(b) 100

(c) 10



(c)



2) Risoluzione Sistemi Lineari malcondizionati

- Risolvere $Ax=y$, A mal condizionata

$$A = U\Sigma V^T$$

$$x = A^{-1}y = V \Sigma^{-1} U^T y$$

$$x = \sum_{i=1}^k \frac{\langle u_i^T, y \rangle}{\sigma_i} v_i$$

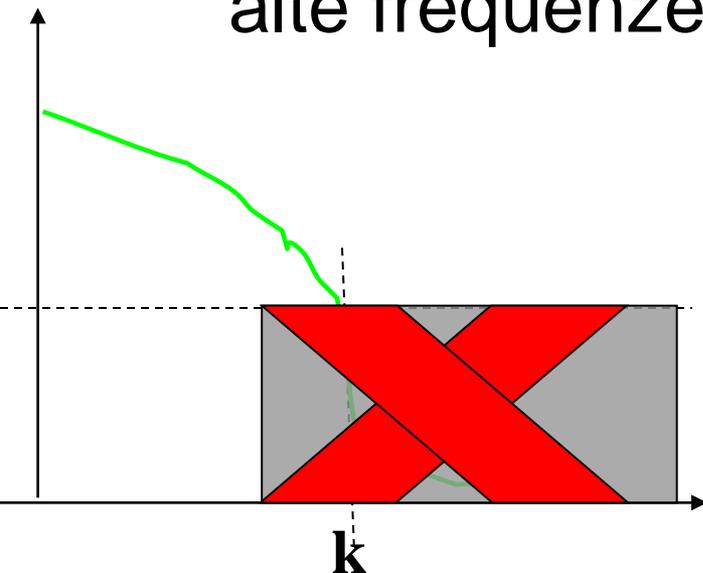
- Rumore nei dati (vettore termine noto):

$$y = \hat{y} + \delta y$$

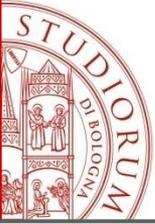
Regolarizzazione: Truncated SVD

$$x = A^{-1}(\hat{y} + \delta y) = \sum_{i=1}^r \frac{\langle u_i^T, \hat{y} \rangle}{\sigma_i} v_i + \sum_{i=1}^r \frac{\langle u_i^T, \delta y \rangle}{\sigma_i} v_i$$

- Eliminare l'effetto dovuto a piccoli valori singolari
- Un filtro elimina le alte frequenze del rumore e le alte frequenze di informazioni utili



Livello Troncamento
 $k < r$ r rango max

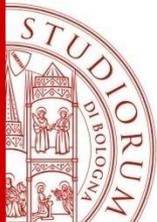


Risoluzione sistemi lineari malcondizionati con il metodo Truncated SVD

- Calcolo decomposizione SVD di A, $A = U\Sigma V^T$
- Elimina valori singolari minori di toll.
- Costruisci soluzione x:

$$x = A^{-1}(\hat{y} + \delta y)$$

$$x \approx V_k \Sigma_k^{-1} U_k^T y = \sum_{i=1}^k \frac{\langle u_i^T, y \rangle}{\sigma_i} v_i$$



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Serena Morigi

Dipartimento di Matematica

morigi@dm.unibo.it

<http://www.dm.unibo.it/~morigi>