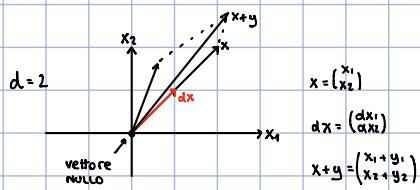


$$\|\cdot\| : \mathbb{R}^d \rightarrow \mathbb{R}_+ \cup \{0\}$$

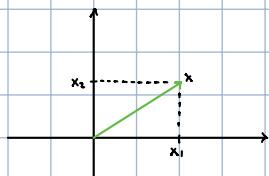
$$\text{PROPRIETÀ: } \|\mathbf{x}\| = 0 \text{ se e solo se } \mathbf{x} = \mathbf{0}$$

$$\|d \cdot \mathbf{x}\| = |d| \cdot \|\mathbf{x}\| \quad \forall d \in \mathbb{R}, \forall \mathbf{x} \in \mathbb{R}^d$$

$$\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$$

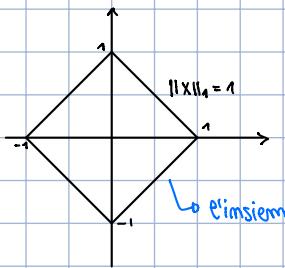


$$\text{NORMA EUCLIDEA: } \|\mathbf{x}\|_2 = \left(\sum_{i=1}^d x_i^2 \right)^{1/2}$$



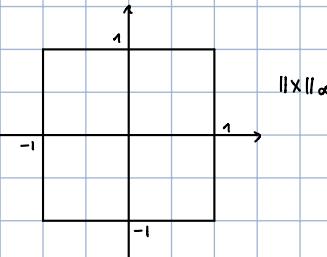
↪ è l'insieme dei punti che hanno norma 2.

$$\|\mathbf{x}\|_1 = \sum_{i=1}^d |x_i| \quad \text{NORMA UNO}$$



↪ è l'insieme dei punti che hanno norma 1

$$\|\mathbf{x}\|_\infty = \max_{i=1 \dots d} |x_i| \quad \text{NORMA INFINITO}$$



$\|\cdot\|'$, $\|\cdot\|''$ sono equivalenti se $C_1 \|\mathbf{x}\|' \leq \|\mathbf{x}\|'' \leq C_2 \|\mathbf{x}\|' \quad \forall \mathbf{x} \in \mathbb{R}^d \rightarrow$ EQUIVALENZA DELLE NORME.

Consideriamo A generica matrice

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{m1} & a_{m2} \end{pmatrix} \quad m \text{ righe, } m \text{ colonne, } m^2 \text{ elementi} \quad A \in \mathbb{R}^{M \times M}$$

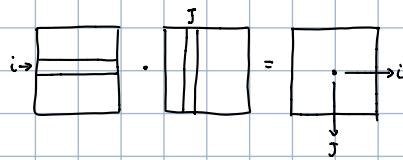
$$A = \begin{pmatrix} a_{ij} \end{pmatrix} \quad m \text{ righe e } m \text{ colonne} \rightarrow A \in \mathbb{R}^{m \times m} \quad A = (a_{ij})_{i=1 \dots m, j=1 \dots m}$$

consideriamo **MATRICI QUADRATI**

$$\alpha \cdot A = (\alpha a_{ij}) \quad i,j=1, \dots, m; \alpha \in \mathbb{R}$$

$$A + B = (a_{ij} + b_{ij}) \quad i,j=1, \dots, m$$

$$A \cdot B = C \quad \text{t.c. } c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}$$



→ il **PRODOTTO MATRICIALE** si definisce quando $A \in \mathbb{R}^{m \times m}$ e $B \in \mathbb{R}^{m \times l}$ allora $C \in \mathbb{R}^{m \times l}$

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj} \quad i=1, \dots, m \quad j=1, \dots, l$$

$$C = A \cdot x = \underbrace{\begin{bmatrix} A \\ \vdots \\ A \end{bmatrix}}_m \cdot \underbrace{\begin{bmatrix} x \\ \vdots \\ x \end{bmatrix}}_m = \underbrace{\begin{bmatrix} ? \\ \vdots \\ ? \end{bmatrix}}_m$$

$$\| \cdot \| : \mathbb{R}^{m \times m} \rightarrow \mathbb{R}_+ \cup \{0\}$$

- $\|A\| = 0$ sse $A = 0 \rightarrow A = \begin{pmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \end{pmatrix}$
- $\|dA\| = |d| \cdot \|A\|, \forall d \in \mathbb{R} \quad \forall A \in \mathbb{R}^{m \times m}$
- $\|A + B\| \leq \|A\| + \|B\|$
- $\|A \cdot B\| \leq \|A\| \cdot \|B\|, \forall A, B \in \mathbb{R}^{m \times m} \rightarrow$ **PROPRIETÀ MOLTIPLICATIVA**
(della matrice matriciale)

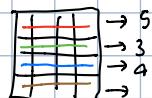
OSS: la definiz. di matrice si estende anche alle matrici rettangolari. (#colonne \neq #righe)

La proprietà 4 vale per $A^{m \times m}$ e $B^{m \times l}$

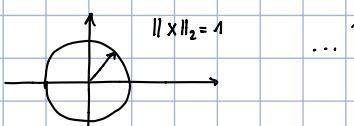
NORMA MATRICIALE INDOTTA

Data una matrice vettoriale $\| \cdot \| : \mathbb{R}^m \rightarrow \mathbb{R}_+ \cup \{0\}$, definisco **NORMA MATRICIALE INDOTTA**

$$\|A\| = \max_{\|x\|=1} \|Ax\|$$



es. $m=2$ matrice euclidea



$$\|A\|_{\infty} = \max_{\|x\|_{\infty}=1} \|Ax\|_{\infty} = \max_{i=1, \dots, m} \sum_{j=1}^m |a_{ij}|$$

Dim: $x \in \mathbb{R}^m$, $\|x\|_\infty = \max_{i=1, \dots, m} |x_i| = 1$

$$(Ax)_i = \sum_{j=1}^m a_{ij} x_j \Rightarrow \|Ax\|_\infty = \max_{i=1, \dots, m} |(Ax)_i|$$

$$|(Ax)_i| = \left| \sum_{j=1}^m a_{ij} x_j \right| \leq \sum_{j=1}^m |a_{ij}| |x_j| \stackrel{\leq 1}{\leq} \sum_{j=1}^m |a_{ij}| \leq \max_{i=1 \dots m} \sum_{j=1}^m |a_{ij}|$$

$$\Rightarrow \|Ax\|_\infty \leq \max_{i=1 \dots m} \sum_{j=1}^m |a_{ij}| = \sum_{j=1}^m |a_{i^*j}|$$

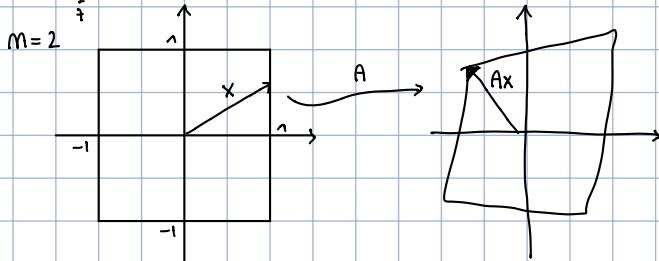
↑ indice del max

$$x^* = \begin{pmatrix} \text{sign}(a_{1^*1}) \\ \vdots \\ \text{sign}(a_{1^*m}) \end{pmatrix} \quad \text{sign}(a) = \begin{cases} 1 & \text{se } a > 0 \\ 0 & \text{se } a = 0 \\ -1 & \text{se } a < 0 \end{cases} \Rightarrow \|x^*\|_\infty = 1$$

$$(Ax^*)_{i^*} = \sum_{j=1}^m a_{i^*j} x_j^* = \sum_{j=1}^m |a_{i^*j}| \Rightarrow \|Ax\|_\infty = \dots \text{c.v.d.}$$

Ej:

$$A = \begin{pmatrix} -1 & 2 & 3 \\ 0 & -4 & -2 \\ 6 & 5 & -1 \end{pmatrix} \rightarrow \begin{array}{l} 1+2+3=6 \\ 4+2=7 \\ 6+5+1=12 \end{array} \quad \begin{array}{l} \text{NORMA MAT. IND.} \\ \Rightarrow \|A\|_\infty = 12 \end{array} \quad \begin{array}{l} \text{NORMA UNO} \\ \Rightarrow \|A\|_1 = 11 \end{array}$$



$$\|A\|_1 = \max_{\|x\|_1=1} \|Ax\|_1 = \max_{\|x\|_1=1} \sum_{j=1}^m |a_{ij}| \rightarrow \text{NORMA UNO}$$

↓ tutti i vettori di norma 1

$$\text{Dim: } x \in \mathbb{R}^m, \|x\|_1 = \sum_{i=1}^m |x_i| = 1$$

$$\|Ax\|_1 = \sum_{i=1}^m |(Ax)_i| = \sum_{i=1}^m \sum_{j=1}^m a_{ij} x_j \leq \sum_{i=1}^m \sum_{j=1}^m |a_{ij}| |x_j| = \sum_{j=1}^m |x_j| \cdot \sum_{i=1}^m |a_{ij}| \leq \max_{j=1 \dots m} \sum_{i=1}^m |a_{ij}| \cdot \sum_{j=1}^m |x_j|$$

$$\Rightarrow \|Ax\|_1 \leq \max_{j=1 \dots m} \sum_{i=1}^m |a_{ij}| = \sum_{i=1}^m |a_{i^*j}|$$

$$x^* = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix} \leftarrow j^* \quad \|x^*\|_1 = 1 \quad Ax^* = \begin{pmatrix} \vdots \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a_{1j^*} \\ \vdots \\ a_{mj^*} \end{pmatrix} \quad \|Ax^*\|_1 = \sum_{i=1}^m |a_{ij^*}| \text{ c.v.d.}$$

Le norme matriciali sono equivalenti: $\| \cdot \|'$, $\| \cdot \|''$ esistono $c_1, c_2 > 0$ t.c. $c_1 \| \cdot \|' \leq \| \cdot \|'' \leq c_2 \| \cdot \|'$, $\forall A$ matrice

$$\| A \|_2 = \sqrt{g(A^T, A)} \quad A^T \text{ è la MATRICE TRASPOSTA (scambio righe con colonne)}$$

$$M = A^T A \quad g(M) \rightarrow \text{Rango di } M = \text{rango spettrale} = \max |\lambda| \quad \lambda \text{ autovalori} \quad Mx = \lambda x, x \neq 0$$

NORMA FROBENIUS $\rightarrow \| A \|_F = \left(\sum_{i,j=1}^m a_{ij}^2 \right)^{1/2}$

$$A = \begin{pmatrix} -1 & 2 & 3 \\ 0 & -4 & -2 \\ 6 & 5 & -1 \end{pmatrix} \quad \| A \|_F = (1+4+9+16+4+36+25+1)^{1/2}$$

diagonali $D = \begin{pmatrix} d_1 & & 0 \\ 0 & \ddots & \vdots \\ \vdots & & 0 \\ 0 & \dots & d_m \end{pmatrix}$

triangolari $T = \begin{pmatrix} \text{SUPERIORE} \\ 0 & 0 & \dots \\ \vdots & 0 & 0 \\ 0 & \dots & 0 \end{pmatrix}$ $T = \begin{pmatrix} 0 & & \dots \\ 0 & 0 & \dots \\ \vdots & 0 & 0 \\ 0 & \dots & 0 \end{pmatrix}$

identità $I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad A \cdot I = A$

matrice inversa $A^{-1} \Rightarrow A^{-1} \cdot A = I = A \cdot A^{-1}$

Sistemi lineari



Rossana Vermiglio

CD Lab  CDLab

Dipartimento di Scienze Matematiche, Informatiche e Fisiche
Università degli Studi di Udine

Gruppo UMI *Modellistica Socio-Epidemiologica* 

Sistema lineare di equazioni

Vogliamo trovare il vettore $x \in \mathbb{R}^n$ che soddisfa al sistema lineare di equazioni

con $m=2$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 & x_1, x_2? \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}$$

$$Ax = b$$

con $A \in \mathbb{R}^{n \times n}$ matrice quadrata e $b \in \mathbb{R}^n$ termine noto.

Si cerca una risposta alla seguente domanda:

Il vettore b può essere espresso come combinazione lineare delle colonne di A ?

In caso di risposta affermativa, i coefficienti della combinazione lineare saranno le componenti di x .

Ci può essere o meno una soluzione e, se c'è soluzione, può non essere unica.

$$Ax = \sum_{j=1}^m x_j \begin{pmatrix} a_{1j} \\ a_{mj} \end{pmatrix}$$

vettori colonna della matrice A

$\in \text{span} \left\{ \begin{pmatrix} a_{1j} \\ a_{mj} \end{pmatrix} \mid j=1..m \right\}$

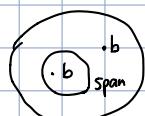
se le colonne di A sono lin. indip.
allora $\text{span} \left\{ \begin{pmatrix} a_{1j} \\ a_{mj} \end{pmatrix} \mid j=1..m \right\} = \mathbb{R}^m$

↳ SPAZIO GENERATO

$\forall b \in \mathbb{R}^m$ esiste $x \in \mathbb{R}^m$ t.c. $Ax = b$

tutto lo spazio che genera

se le colonne di A non sono lin. indip. $\Rightarrow \text{span}\{ \} \neq \mathbb{R}^m$



^{RANGO}
 $\dim \{\text{span}\} = k < m$

Sistema lineare di equazioni, continua

Il sistema ha soluzione unica per ogni vettore b (=problema ben posto) se e soltanto se A è non singolare

Sono equivalenti

- A non singolare
- $\det(A)$ non nullo
- $\text{rango}(A) = n$
 ↳ dim. del sottospazio
- esiste l'inversa A^{-1}

$$\begin{cases} 3x+2y=7 \\ x+3y=7 \\ 2x+y=4 \end{cases} \Rightarrow \begin{array}{l} x=1, y=2 \text{ è qualsiasi} \\ (1,2,z) \text{ soluz.} \end{array}$$

Se A è singolare posso avere infinite soluzioni o nessuna soluzione

Remark

Nel caso $n = 2$ risolvere un sistema lineare corrisponde a determinare il punto di intersezione di due rette.

Introduciamo una perturbazione (sul termine noto b) $\delta b \in \mathbb{R}^n$

$$\delta b = \tilde{b} - b \quad \tilde{b} = b + \delta b \quad \text{e indichiamo con } \epsilon_b \text{ l'ERRORE RELATIVO: } \epsilon_b = \frac{\|\tilde{b} - b\|}{\|b\|} = \frac{\|\delta b\|}{\|b\|}$$

$\hookrightarrow b \neq 0$

Sia $(x + \delta x) \in \mathbb{R}^n$ la soluzione esatta del problema perturbato

$$A(\tilde{x} - x) = A\tilde{x} - Ax = \tilde{b} - b = \delta b$$

$$\hookrightarrow (\tilde{x} - x) = A^{-1} \cdot \delta b \Rightarrow \|\tilde{x} - x\| = \|A^{-1} \cdot \delta b\| \leq \|A^{-1}\| \cdot \|\delta b\|$$

→ soluz. perturbata da calcolare

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

↪ è un vettore che contiene la perturbazione di b

Definendo com ϵ_x l'errore relativo

$$\epsilon_x = \frac{\|\tilde{x} - x\|}{\|x\|} \leq \|A^{-1}\| \cdot \frac{\|\delta b\|}{\|x\|} = \|A^{-1}\| \cdot \frac{\|b\|}{\|x\|} \cdot \frac{\epsilon_b}{\|b\|} \leq \underbrace{\|A^{-1}\| \cdot \|A\|}_{\text{INDICE DI CONDIZIONAMENTO}} \cdot \epsilon_b$$

$\hookrightarrow \text{cond}(A)$

→ anche un piccolo errore nel dato potrebbe darci un grande errore nella risposta (x)

→ il numero di condizionamento lega l'errore nel dato con l'errore nella risposta

Introduciamo anche una perturbazione $\delta A \in \mathbb{R}^{n \times n}$ nella matrice A ed indichiamo com $\epsilon_A = \frac{\|\delta A\|}{\|A\|}$ l'errore relativo

Sia $(x + \delta x) \in \mathbb{R}^n$ la soluzione esatta del problema perturbato

$$\underbrace{(A + \delta A)}_{\text{matrice}}(\tilde{x}) = \tilde{b} \quad \text{com} \quad \epsilon_b = \frac{\|\tilde{b} - b\|}{\|b\|}, \quad \epsilon_A = \frac{\|\tilde{A} - A\|}{\|A\|} \quad \text{dov'è legante com } \epsilon_x$$

$\hookrightarrow A \neq 0$

Allora vale che $\epsilon_x \leq \|A\| \cdot \|A^{-1}\| \cdot (\epsilon_b + \epsilon_A)$

$\frac{1}{1 - \|A\| \cdot \|A^{-1}\| \cdot \epsilon_A}$
cond(A)

Se $\text{cond}(A) \gg 1 \rightarrow \text{PROBLEMA MAL CONDIZIONATO}$

Se $\text{cond}(A)$ "piccolo" $\rightarrow \text{PROBLEMA BEN CONDIZIONATO}$

Condizionamento:

errore nel termine noto b

Introduciamo una perturbazione $\delta b \in \mathbb{R}^n$ del termine noto b e indichiamo con $\epsilon_b = \frac{\|\delta b\|}{\|b\|}$ l'errore. $\delta b = \tilde{b} - b$ $\tilde{b} = b + \delta b$ $\epsilon_b = \frac{\|\tilde{b} - b\|}{\|\tilde{b}\|} = \frac{\|\delta b\|}{\|\tilde{b}\|}$

↳ ERRORE RELATIVO

Sia $x + \delta x \in \mathbb{R}^n$ la soluzione esatta del problema perturbato, i.e.

$$A(x + \delta x) = b + \delta b.$$

↳ soluz. perturbata da calcolare

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

↳ è un vettore che contiene la PERTURBAZIONE di b

Vale $\delta x = A^{-1} \delta b$, da cui si ottiene $\|\delta x\| \leq \|A^{-1}\| \cdot \|\delta b\|$. Definendo l'errore relativo $\epsilon_x = \frac{\|\delta x\|}{\|x\|}$ si ottiene

$$Ax = b \quad A\tilde{x} = \tilde{b} \Rightarrow \epsilon_x = \frac{\|\tilde{x} - x\|}{\|x\|}$$

$$\text{d) } A(\tilde{x} - x) = A\tilde{x} - Ax = \tilde{b} - b = \underline{\delta b}$$

$$\text{L} \rightarrow (\tilde{x} - x) = A^{-1} \cdot \delta b \Rightarrow \|\tilde{x} - x\| = \|A^{-1} \cdot \delta b\| \leq \|A^{-1}\| \cdot \|\delta b\|$$

$$\epsilon_x \leq \text{cond}(A) \epsilon_b$$

dove

$$\text{cond}(A) = \|A\| \|A^{-1}\| \geq 1$$

è il numero di condizionamento.

$$\epsilon_x = \frac{\|\tilde{x} - x\|}{\|x\|} \leq \|A^{-1}\| \cdot \frac{\|\delta b\|}{\|x\|} \cdot \frac{\|b\|}{\|b\|} \leq \underbrace{\|A^{-1}\| \cdot \|A\|}_{\text{INDICE DI CONDIZIONAMENTO}} \cdot \epsilon_b$$

$$\text{d) } (A + \delta A)(\tilde{x}) = \tilde{b} \quad \text{PERTURBAZ. anche in } A$$

$$\epsilon_b = \frac{\|\tilde{b} - b\|}{\|b\|}, \epsilon_A = \frac{\|\delta A\|}{\|A\|}$$

matrice

$$\hookrightarrow A \neq 0$$

$$\tilde{A} \text{ sia non singolare} \quad \epsilon_x \leq \frac{\|A\| \cdot \|A^{-1}\|}{1 - \|A\| \cdot \|A^{-1}\|} (\epsilon_b + \epsilon_A)$$

Condizionamento:

errore nel termine noto b e nella matrice A

Introduciamo anche una perturbazione $\delta A \in \mathbb{R}^{n \times n}$ nella matrice A ed indichiamo con $\epsilon_A = \frac{\|\delta A\|}{\|A\|}$ l'errore.

Sia $x + \delta x \in \mathbb{R}^n$ la soluzione esatta del problema perturbato (che supponiamo esistere)

$$(A + \delta A)(x + \delta x) = b + \delta b.$$

Vale

$$\epsilon_x \leq \frac{\text{cond}(A)}{1 - \text{cond}(A)\epsilon_A} (\epsilon_b + \epsilon_A),$$

- $\text{cond}(A) \gg 1 \rightsquigarrow$ problema **mal condizionato**
- $\text{cond}(A)$ 'piccolo' \rightsquigarrow problema **ben condizionato**

Esempio di matrice mal condizionata

La matrice di Hilbert

MATRICE MAL CONDIZIONATA

$$H_n = \begin{pmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{pmatrix}$$

Vale

$$\|A\|_2 \cdot \|A^{-1}\|_2$$

matrice 2

- $n = 2 \rightsquigarrow \text{cond}_2(H_n) \approx 1.50$ ben condizionato
- $n = 5 \rightsquigarrow \text{cond}_2(H_n) \approx 4.77 \times 10^5$ mal condizionato, perdiamo 5 cifre decimali
- $n = 10 \rightsquigarrow \text{cond}_2(H_n) \approx 1.60 \times 10^{13}$ mal condizionato

Proprietà del numero di condizionamento

Poichè

$$\text{cond}(A) = \left(\max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \right) \cdot \left(\min_{x \neq 0} \frac{\|Ax\|}{\|x\|} \right)^{-1}$$

il numero di condizionamento misura il rapporto tra il massimo "allungamento" e la massima "contrazione" che la matrice A opera ad un vettore non nullo.

Un numero di condizionamento grande significa che la matrice è **quasi singolare**.

- Per ogni matrice A , $\text{cond}(A) \geq 1$
- Per la matrice identità I , $\text{cond}(I) = 1$
- Per ogni matrice A e numero reale $\alpha \neq 0$, $\text{cond}(\alpha A) = \text{cond}(A)$
- Per ogni matrice diagonale $D = \text{diag}(d_i)$, $\text{cond}(D) = \frac{\max |d_i|}{\min |d_i|}$
↳ massimo elemento sulla diagonale
↳ minimo elemento sulla diagonale

Calcolo del numero di condizionamento

→ il CALCOLO dell' INVERSA è da EVITARE !! cerca piuttosto delle riformulazioni

- La definizione del numero di condizionamento coinvolge l'inversa, che in generale è difficile da calcolare.
- Il calcolo del numero di condizionamento può essere più costoso della risoluzione del sistema lineare.
→ si può stimarne $\|A'\|$ facendo $\frac{\|z\|}{\|y\|}$ con y ottimale.
- Dalle proprietà delle norme, se $Az = y$, allora $\|z\| \leq \|A^{-1}\| \cdot \|y\|$ e la limitazione può essere raggiunta scegliendo y in maniera ottimale. Delle stime efficienti si ottengono prendendo in maniera euristica y con $\frac{\|z\|}{\|y\|}$ grande.
- I pacchetti software per la risoluzione di sistemi lineari forniscono delle stime efficienti e affidabili del numero di condizionamento.
- Il numero di condizionamento è utile per stima dell'accuratezza della soluzione approssimata.

Esempio

Condizionamento del calcolo di $y = Ax$

CALCOLARE MATRICE · VETTORE

Data la matrice A invertibile, vogliamo studiare il condizionamento del calcolo di $y = Ax$.

Consideriamo $\tilde{x} \in \mathbb{R}^n$ e indichiamo $\tilde{y} = A\tilde{x} \in \mathbb{R}^n$. Vale

$$\|\tilde{y} - y\| = \|A\tilde{x} - Ax\| = \|A(\tilde{x} - x)\| \leq \|A\| \|\tilde{x} - x\|$$

Poiché A è invertibile, abbiamo che

$$\|x\| = \|A^{-1}y\| \leq \|A^{-1}\| \|y\|$$

da cui

$$\frac{\|\tilde{y} - y\|}{\|y\|} \leq \|A\| \frac{\|x\| \|\tilde{x} - x\|}{\|y\| \|\tilde{x}\|} \leq \underbrace{\text{cond}(A)}_{\|A\| \cdot \|A^{-1}\|} \frac{\|\tilde{x} - x\|}{\|x\|}$$

$y = Ax$ e quindi $\|A^{-1} \cdot Ax\| = \|x\|$

→ moltiplica per $\frac{\|x\|}{\|x\|}$

$\hookrightarrow \|\tilde{y} - y\| \leq \|A\| \cdot \|\tilde{x} - x\|$

$\hookrightarrow \frac{\|x\|}{\|y\|} = \|A^{-1}\|$

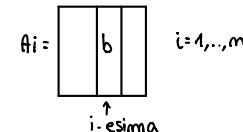
Se $\text{cond}(A) \gg 1$, il problema del calcolo di $y = Ax$ è **mal condizionato**.

La regola di Cramer

Determinare un algoritmo per calcolare la soluz. di un sistema lineare \rightarrow si considerano: **err ALGO, err INERENTE**

La **soluzione di un sistema lineare** può essere teoricamente ottenuta mediante la formula

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad i = 1, \dots, n$$



dove A_i è la matrice ottenuta sostituendo la colonna i -esima di A il termine noto b .

Il **costo** computazionale di tale algoritmo è dell'ordine di $O((n+1)!)$.

In pratica non è utilizzabile: su un calcolatore in grado di eseguire 1 mega-flops (10^6) servirebbero 1.6×10^6 anni per risolvere un sistema di dimensione $n = 20$. Anche con un calcolatore più avanzato con velocità pari a 10^{15} flops, sarebbero necessari 20 anni per risolvere un sistema di dimensione $n = 24$.

Anche aumentando la potenza di calcolo l'algoritmo di Cramer è inadeguato alla risoluzione di un sistema lineare e pertanto tale problema matematico va affrontato sviluppando altri approcci.

$$\det \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = a_{11} \cdot a_{22} - a_{21} \cdot a_{12} \quad (2 \text{ moltiplicaz} + 1 \text{ addizione})$$

$$\det \begin{array}{|c|c|} \hline \bullet & \bullet \\ \hline & \quad \\ \hline \end{array} = C(m) = \underbrace{m C(m-1)}_{\substack{\# \text{ operaz.} \\ \text{per col.}}} + O(m) \quad \begin{array}{l} \text{m determinanti di} \\ \text{mancanti di dim. } m-1 \end{array}$$

Lo tot operaz. al livello corrente

$C(m) > m \cdot C(m-1) > m(m-1) \cdot C(m-2) > \dots > m(m-1) \dots 3 \cdot C(2) > m!$ \rightarrow il calcolo di 1 determinante e io me devo calcolare $m+1 \Rightarrow \Theta(m+1) \cdot O(m!) = O((m+1)!)$

\Rightarrow costo: $(m+1)!$

Idea: Trasformare il sistema lineare in un altro che ha la **stessa soluzione** ma che sia **facilmente risolvibile**.

Quali sono le trasformazioni che non modificano la soluzione di un sistema lineare?

Quali sono i sistemi lineari facilmente risolvibili?

↳ prima capisco i casi **SEMPLE**
poi cerco di ricordarmi a questi.

Trasformazioni possibili

TRASFORMAZIONI che posso applicare al mio sistema lineare e non ne modifichino la soluzione

La soluzione del sistema non si modifica premoltiplicando a sinistra sia la matrice A che il termine noto b per una matrice M invertibile

$$M A x = M b \rightsquigarrow x = (M A)^{-1} M b = A^{-1} M^{-1} M b = A^{-1} b$$

Esempi:

$$\left. \begin{array}{l} \underbrace{M \cdot A x}_{\bar{A} x} = \underbrace{M \cdot b}_{\bar{b}} \\ \bar{A} x = \bar{b} \end{array} \right] \xrightarrow{\text{hanno la stessa soluz.}} x = A^{-1} \cdot b$$

$$\bar{x} = \bar{A}^{-1} \cdot \bar{b} = (M A)^{-1} \cdot (M b) = A^{-1} \cdot \underbrace{M^{-1} \cdot M}_{\text{Id}} \cdot b = A^{-1} \cdot b$$

- matrici di permutazione P

P si ottiene riordinando le righe della matrice identità I .

Premoltiplicando per P la matrice A e il termine noto b , le equazioni vengono riordinate senza modificare la soluzione. Vale $P^{-1} = P^T$

- matrici diagonali D non singolari.

Premoltiplicando per D la matrice A e il termine noto b , ogni equazione viene moltiplicata per il corrispondente termine sulla diagonale (scaling) senza modificare la soluzione.

$$\underbrace{M \cdot A}_{\bar{A}} \underbrace{x}_{\bar{b}} = \underbrace{M \cdot b}_{\bar{b}} \quad \xrightarrow{\text{hanno la stessa soluz.}} \quad x = A^{-1} \cdot b$$

$$\bar{x} = \bar{A}^{-1} \cdot \bar{b} = (MA)^{-1} \cdot (Mb) = A^{-1} \cdot \underbrace{M^{-1} \cdot M}_{\text{Id}} \cdot b = A^{-1} \cdot b$$

es: **MATRICE di PERMUTAZIONE**

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad \text{t.c. } P \cdot x = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_2 \\ x_3 \\ x_1 \end{pmatrix}$$

$$P \cdot A = \begin{pmatrix} a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{11} & a_{12} & a_{13} \end{pmatrix}$$

\Rightarrow principio che se in un sistema di equaz. scambio l'ordine di equaz. la soluz. non cambia
 \hookrightarrow espresso in forma matriciale

es: **MATRICE DIAGONALE**

$$D = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1/2 \end{pmatrix} \quad \text{t.c. } D \cdot x = \begin{pmatrix} -x_1 \\ 2x_2 \\ x_3/2 \end{pmatrix}$$

$$D \cdot A = \begin{pmatrix} -a_{11} & -a_{12} & -a_{13} \\ 2a_{21} & 2a_{22} & 2a_{23} \\ a_{31}/2 & a_{32}/2 & a_{33}/2 \end{pmatrix}$$

\Rightarrow principio che se in un sistema di equaz. moltiplico per un numero tutte le equaz. la soluz. non cambia
 \hookrightarrow espresso in forma matriciale

Casi facili: A diagonale e triangolare

SISTEMI LINEARI SEMPLICI

- $A = \text{diag}(a_{i,i})_{i=1,\dots,n}$, con $a_{i,i} \stackrel{\text{NON SINGOLARE}}{\neq} 0, i = 1, \dots, n$,

$$x_i = \frac{b_i}{a_{i,i}}, \quad i = 1, \dots, n,$$

Faccio m divisioni floating point (flop)

e richiede n operazioni aritmetiche.

$$A = \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & \ddots \end{pmatrix} \quad \begin{array}{l} a_{ii} \neq 0 \\ \hookrightarrow \text{NON SINGOLARE} \end{array}$$

$$Ax = b \iff \{ a_{ii} x_i = b_i \mid i = 1 \dots m \}$$

$$\Rightarrow x_i = \frac{b_i}{a_{ii}} \quad i = 1 \dots m \quad \text{con } m \text{ operaz. flop trovo soluz.}$$

divisioni

OSS: se $a_{ii} = 0$ $0 \cdot x_i = b_i$ $\begin{cases} b_i = 0 & x_i = 1 \rightarrow \text{valore arbitrario} \\ b_i \neq 0 & \text{non riesco a risolvere} \end{cases}$

↪ matrice SINGOLARE

Casi facili: A diagonale e triangolare

$$A = \begin{pmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ \vdots & & \ddots & 0 \\ a_{i1} & a_{i2} & \cdots & a_{ii} \\ & & \cdots & a_{m1} \end{pmatrix}, \quad a_{ii} \neq 0, \quad i=1, \dots, m$$

RIGAi $\{ a_{i1}x_1 + \dots + a_{ii}x_i = b_i, i=1,..,m \}$
(della matrice)

$$\begin{aligned} i=1 &\rightarrow a_{11}x_1 = b_1 \rightarrow x_1 = \frac{b_1}{a_{11}} \\ i=2 &\rightarrow a_{21}x_1 + a_{22}x_2 = b_2 \\ &\hookrightarrow x_2 = b_2 - \frac{a_{21}x_1}{a_{22}} \end{aligned} \Rightarrow X_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j}{a_{ii}}$$

OSS: se $a_{ii} = 0$ format. singolare

$$\left\{ \begin{array}{l} a_{11}x_1 + \dots + a_{1i-1}x_{i-1} = b_1 \\ \vdots \\ a_{ii}x_1 + \dots + a_{ii-1}x_{i-1} = b_i \end{array} \right. \quad \begin{array}{l} \text{se } f \neq \text{ mom } \Rightarrow \text{solo} \\ \text{soluz.} \end{array}$$

- A matrice triangolare inferiore (i.e. $a_{i,j} = 0$, $i = 1, \dots, n-1$, $j = i+1, \dots, n$) con $a_{i,i} \neq 0$, $i = 1, \dots, n$, \rightsquigarrow l'algoritmo di sostituzione in avanti:

$$x_1 = \frac{b_1}{a_{1,1}}$$

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{i,j} x_j}{a_{i,i}}, \quad i = 2, \dots, n.$$

SOCIETATEA DE INVENTI

Il calcolo di x_i richiede $i - 1$ somme, $i - 1$ moltiplicazioni e una divisione, quindi

$$\sum_{i=1}^n (2i-1) = 2 \frac{n(n+1)}{2} - n = n^2$$

operazioni aritmetiche.

Esercizio: matrice triangolare superiore

Exercise

Descrivi ed analizza l'algoritmo di sostituzione all'indietro per la risoluzione di un sistema triangolare superiore, i.e. A è tale che $a_{i,j} = 0$, $i = 2, \dots, n$, $j = 1, \dots, i-1$.

Com'è la MATRICE TRIANGOLARE SUPERIORE \rightarrow SOSTITUZIONE ALL'INDIETRO

$$A = \begin{pmatrix} a_{11} & & a_{1j} \\ \vdots & \ddots & \vdots \\ 0 & \ddots & a_{mm} \\ 0 & \dots & 0 \end{pmatrix}$$

$$a_{mm} x_m = b_m$$

$x_m = \frac{b_m}{a_{mm}}$ e la sostituisco nell'equaz. che calcola $m-1$ \rightarrow Parto dall'ultima equaz. e procede all'indietro \rightarrow A RITROSO

Caso facile: A ortogonale

Caso con **MATRICE ORTOGONALE**

Una **matrice A** si dice **ortogonale** quando

$$AA^T = A^T A = I$$

In questo caso l'inversa è la trasposta di A e pertanto

$$x = A^T b.$$

→ scambio righe con colonne → per trovarla non servono calcoli
→ solo PRODOTTO MATRICIALE

↳ soluz. del sistema LINEARE

Si ha quindi

$$x_i = \sum_{j=1}^n a_{ji} b_j, \quad i = 1, \dots, n.$$

Il calcolo del vettore soluzione richiede n^2 moltiplicazioni e $n(n - 1)$ addizioni, quindi circa $2n^2$ operazioni aritmetiche.

Oss: Il calcolo di un prodotto matrice-vettore e la risoluzione di sistemi triangolari con gli algoritmi di sostituzione sono algoritmi stabili all'indietro.

Caso generale: i metodi diretti

APPROCCIO: METODI DIRETTI

1 Scrivere $A = MN$ con M, N matrici "facili" (=fattorizzare A)

Risolvere $Ax = b$ risolvendo in sequenza due sistemi "facili"

2 $Mz = b \rightsquigarrow z$ $\hookrightarrow \underbrace{MN \cdot x = b}_z$, poi chiamo $z = Nx \Rightarrow Mz = b$ lo so risolvere facilmente (perché matrici facili) e trovo z . Poi so che $z = Nx$ e ricavo x che è la sol.

3 $Nx = z \rightsquigarrow x$

Come trovo la giusta FATTORIZZAZIONE di A ? (ovvero scriverla come PRODOTTO)

Esempi notevoli:

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

- Fattorizzazione $A = LU$ con L unitriangolare inferiore e U triangolare superiore
 - ↳ TRIANGOLARE inferiore, ma sulla diagonale ha solo 1
- Fattorizzazione di Cholesky $A = LL^T$ con L triangolare inferiore
- Fattorizzazione $A = QR$ con Q ortogonale e R triangolare superiore

La fattorizzazione Lu

Consideriamo un sistema lineare generale. Possiamo ricondurci alla soluzione di sistemi triangolari?

Se

↗ prova a scrivere A come il prodotto di L e U

$$A = Lu \quad \text{fattorizzazione } Lu$$

dove

- L è una matrice triangolare inferiore (Lower) con $l_{i,i} = 1$,
 $i = 1, \dots, n$
- U è una matrice triangolare superiore (Upper)

allora

$$Ax = Lu x = b \rightarrow \begin{cases} Lz = b & \rightsquigarrow z \\ Ux = z & \rightsquigarrow x \end{cases}$$

SOSTITUZ. IN AVANTI $O(m^2)$
SOSTITUZ. ALL'INDIETRO $O(m^2)$

- esiste sempre la fattorizzazione LU ? \rightarrow No
- come costruire un algoritmo per il calcolo di $L = (l_{i,j})$, $U = (u_{i,j})$?

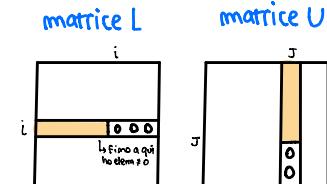
→ Considero gli elementi di A come prodotto matriciale

Un **approccio diretto (tecnica compatta)** consiste nel considerare le relazioni

$$a_{i,j} = \sum_{k=1}^{\min\{i,j\}} l_{i,k} u_{k,j}, i, j = 1, \dots, n$$

e risolvere rispetto $l_{i,j}$ e $u_{i,j}$

⇒ con questo approccio non si riesce a controllare le **PROPRIETÀ DI STABILITÀ**
⇒ APPROCCIO **NON USATO**



Example

Consideriamo il caso $n = 2$. Dalla relazione segue

$$\left\{ \begin{array}{l} a_{1,1} = l_{1,1} u_{1,1} = u_{1,1} \rightarrow u_{1,1} \\ a_{1,2} = l_{1,1} u_{1,2} = u_{1,2} \rightarrow u_{1,2} \\ a_{2,1} = l_{2,1} u_{1,1} \rightarrow l_{2,1} \\ a_{2,2} = l_{2,1} u_{1,2} + l_{2,2} u_{2,2} \end{array} \right. \begin{array}{l} \text{perché DIAG.} \\ \text{.} \end{array}$$

es. $A = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$ non posso applicare questo approccio, nonostante sia **NON SINGOLARE**

Risolvendo le quattro equazioni rispetto le quattro incognite $l_{2,1}$, $u_{1,1}$, $u_{1,2}$, $u_{2,2}$ si ottiene $u_{1,1} = a_{1,1}$, $u_{1,2} = a_{1,2}$, $l_{2,1} = a_{2,1}/a_{1,1}$, $u_{2,2} = a_{2,2} - a_{2,1} * a_{1,2}/a_{1,1}$. \star deve essere $\neq 0$

L'algoritmo di eliminazione di Gauss

L'idea base dell'**algoritmo di eliminazione di Gauss** consiste nel ridurre la matrice A alla forma triangolare superiore attraverso un numero **finito di trasformazioni**, descritte da matrici triangolari inferiori facilmente invertibili (matrici elementari di Gauss).

Tali matrici di trasformazione devono annullare (=eliminare) gli elementi diversi da zero nella parte triangolare inferiore di A

Matrici elementari di Gauss

Dato $v \in \mathbb{R}^n$ con $v_k \neq 0$, possiamo annullare (=eliminare) tutte le sue componenti $v_i, i = k+1, \dots, n$ mediante la trasformazione

$$\left(\begin{array}{cccccc} 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & -\frac{v_{k+1}}{v_k} & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -\frac{v_n}{v_k} & 0 & \dots & 1 \end{array} \right) \left(\begin{array}{c} v_1 \\ \vdots \\ v_k \\ v_{k+1} \\ \vdots \\ v_n \end{array} \right) = \left(\begin{array}{c} v_1 \\ \vdots \\ v_k \\ 0 \\ \vdots \\ 0 \end{array} \right)$$

\uparrow
 G_k

matrice elementare di Gauss

Il divisore $v_k \neq 0$ si chiama **pivot**

Gli elementi $m_{i,k} = \frac{v_i}{v_k}, i = k+1, \dots, n$ sono i **moltiplicatori**

Matrici elementari di Gauss, continua

La trasformazione $t = G_k w$ di un generico vettore $w \in \mathbb{R}^n$ è descritta dalle relazioni

elementi
di $A^{(i,n)}$

$$\begin{cases} t_i = w_i, \quad i = 1, \dots, k, & \rightarrow \text{PARTE che non viene toccata} \\ t_i = w_i - \frac{v_i * w_k}{v_k} = w_i - m_{i,k} * w_k, \quad i = k + 1, \dots, n \end{cases}$$

G_k è triangolare inferiore, facilmente invertibile. La sua inversa è

$$G_k^{-1} = \begin{pmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & \frac{v_{k+1}}{v_k} & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \frac{v_n}{v_k} & 0 & \dots & 1 \end{pmatrix}$$

combiò solo
il segno agli
elementi

sceglio 1 come PNOT

$$\begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{1} & 1 & 0 \\ -\frac{3}{1} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ -2+2=0 \\ -3+3=0 \end{pmatrix}$$

sceglio invece 2 come PNOT

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

INVERSA

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

Matrici elementari di Gauss, esempio

Dato

$$v = \begin{pmatrix} 2 \\ -3 \\ 4 \\ -5 \end{pmatrix}$$

si ottiene

$$G_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{3}{2} & 1 & 0 & 0 \\ -\frac{4}{2} & 0 & 1 & 0 \\ \frac{5}{2} & 0 & 0 & 1 \end{pmatrix} \quad G_1 v = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

PIVOT = 2

$$G_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{4}{3} & 1 & 0 \\ 0 & -\frac{5}{3} & 0 & 1 \end{pmatrix}, \quad G_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{5}{4} & 1 \end{pmatrix}$$

PIVOT = -3

PIVOT = 4

$$G_2 v = \begin{pmatrix} 2 \\ -3 \\ 0 \\ 0 \end{pmatrix}$$

L'algoritmo di Gauss di base

L'**algoritmo di eliminazione di Gauss** lavora sulle colonne della matrice A eliminando passo dopo passo tutti gli elementi nella parte triangolare inferiore.

Posta $A^{(0)} = A$, la trasformazione

$$A^{(k)} = G_k A^{(k-1)}$$

al k -esimo passo, $k = 1, 2, \dots, n-1$, è descritta dalla matrice elementare

$$G_k = \begin{pmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & -l_{k+1,k} & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -l_{n,k} & 0 & \dots & 1 \end{pmatrix},$$

elementi della matrice elem. GAUSSIANA

$$\text{dove } l_{i,k} = \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}}, i = k+1, \dots, n.$$

G_k risulta definita se l'elemento pivotale $a_{k,k}^{(k-1)} \neq 0$, $k = 1, \dots, n-1$.



$$\left(\begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & & & \\ \vdots & & & \\ a_{m1} & & & \end{array} \right) \quad \text{e voglio ridurla in forma triangolare superiore (inizio dalla prima colonna)}$$

voglio eliminare

$A^{(0)}$

Se $a_{11}^{(0)} \neq 0$

$$\left(\begin{array}{cccc} 1 & & & \\ -a_{21}^{(0)} & 1 & & \\ \vdots & & \ddots & \\ -a_{m1}^{(0)} & & & 1 \end{array} \right) \cdot \left(\begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & & & \\ \vdots & & & \\ a_{m1} & & & \end{array} \right) = \left(\begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1m} \\ 0 & & & \\ \vdots & & & \\ 0 & & & a_{m1}^{(1)} \end{array} \right)$$

questa parte è l'unica che si modificherà

$A^{(0)}$

$A^{(1)}$

Lo si sposta lungo la diagonale

MATTR. EL. DI GAUSS

$$\left(\begin{array}{cccc} 1 & & & \\ 0 & 1 & & \\ 0 & -\frac{a_{22}^{(0)}}{a_{22}} & \dots & \\ 0 & -\frac{a_{m2}^{(0)}}{a_{22}} & & 1 \end{array} \right) \cdot \left(\begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1m} \\ 0 & a_{22}^{(1)} & & \\ \vdots & a_{32}^{(1)} & & a_{33}^{(1)} \\ 0 & a_{m2}^{(1)} & & \end{array} \right) \Rightarrow \dots \text{dopo } n-1 \text{ passi} \dots \Rightarrow \left(\begin{array}{cccc} & & & \\ 0 & & & \\ 0 & & & \\ \vdots & & & \\ 0 & & & \end{array} \right) = U$$

com i PIVOT

$A^{(1)}$

$A^{(n-1)}$

mi sposto lungo la diagonale

Se $a_{22}^{(1)} \neq 0$

Le voglio eliminare

\Rightarrow P¹ IMPLEMENTAZ. effettiva Non usai prodotti matriciali

L'algoritmo di Gauss, continua

In forma matriciale si ottiene

$$\textcolor{red}{U} = A^{(n-1)} = (G_{n-1} \cdots G_2 G_1) A = GA$$

G è il prodotto di tutte le matrici elementari

dove $G = (G_{n-1} \cdots G_2 G_1)$ risulta essere una matrice triangolare inferiore con uno sulla diagonale principale.

$G^{-1} = (G_1^{-1} G_2^{-1} \cdots G_{n-1}^{-1})$ è una matrice triangolare inferiore con uno sulla diagonale ed è facilmente calcolabile ed osservando che il prodotto di tali matrici è essenzialmente la loro "unione".

Ponendo $\textcolor{blue}{L} = G^{-1}$ otteniamo la fattorizzazione

$$\textcolor{blue}{L} \textcolor{red}{U} = A.$$

es:

$$G^{-1} = \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 4 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \right]^{-1} = \text{faccio l'inversa di ciascuna} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & -4 & 1 \end{pmatrix}$$

G_1^{-1} G_2^{-1}

L'algoritmo di Gauss, continua

In realtà non viene calcolato come prodotto matriciale

L'uso delle matrici elementari di Gauss è proposto per fornire una descrizione formale del processo di fattorizzazione ma **non** sono implementate esplicitamente. **L'algoritmo di eliminazione di Gauss è descritto da**

for $k = 1, \dots, n-1$, \rightarrow individua il passo (considera solo gli elementi della diagonale)

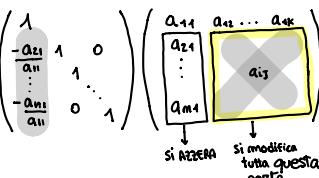
for $i = k+1, \dots, n$

elemento
della matrice
el. di Gauss L

$$l_{i,k} = \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}} \rightarrow 1 \text{ op. aritmetica}$$

$\Rightarrow 2(m-k)^2$ op. aritmetiche per tutti gli a_{ij}
 $m-k$ op. per calcolare tutti gli l_{ik}

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



elemento
della nuova
matrice

$$a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - l_{i,k} a_{k,j}^{(k-1)} \rightarrow 2 \text{ op. aritmetiche}$$

end

end

end

eseguite in arit.
esatta

\rightarrow se tutte le op. sono fatte con operaz. esatte, allora

c'è solo l'ERRORE INERENTE e la soluz. è esatta!

Ma purtroppo così non è. \rightarrow c'è anche l'errore

\hookrightarrow ci sono ERALGO ed ERRINERENTE

Gli elementi $l_{i,j}$, $i = 2, \dots, n$, $j = 1, \dots, i-1$ definiscono la matrice L mentre $U = (a_{i,j}^{(n-1)})$, $i = 1, \dots, n$, $j = i, \dots, n$.

L'algoritmo di eliminazione di Gauss e la fattorizzazione LU sono semplicemente due modi di esprimere lo stesso procedimento

L'algoritmo di Gauss: esistenza ed unicità $A = LU$

Per poter applicare l'algoritmo gli elementi pivotali $a_{k,k}^{(k-1)}$ devono essere non nulli ad ogni passo.

↳ non posso dividere per 0

Vale

$$a_{k,k}^{(k-1)} \neq 0, k = 1, \dots, n-1$$

se e solo se

↳ Sottomatrice di testa di dim 1

↳ Sottomatrice di testa di dim 2

↳ $\det = 1 \cdot 5 - 2 \cdot 4 = 0$

le sottomatrici principali di testa di A di ordine k , A_k , sono non singolari $k = 1, 2, \dots, n-1$

Non è sufficiente la non singolarità della matrice. \Rightarrow com'è PNOT PARZIALE questa limitazione non c'è

Theorem

Sia $A \in \mathbb{R}^{n \times n}$ una matrice non singolare e tale che tutte le sottomatrici A_k , $k = 1, \dots, n-1$ siano non singolari. Allora esiste unica la fattorizzazione LU di A . Inoltre risulta che U è non singolare. Viceversa data una fattorizzazione LU con U non singolare, allora la matrice A e le sottomatrici A_k , $k = 1, \dots, n-1$ sono non singolari.

matrici in cui il valore (assoluto) di ogni elemento sulla diagonale è maggiore (in val. ass.) alla somma di tutti gli elementi sulla stessa riga.

Le matrici simmetriche definite positive e le matrici a dominanza diagonale stretta verificano le ipotesi di tale teorema.

L'algoritmo di Gauss, continua

Applicando le trasformazioni elementari G_k anche al termine noto b , i.e.

$$z = b^{(n-1)} = (G_{n-1} \cdots G_2 G_1) b \stackrel{\text{po applico } n-i \text{ volte}}{=} \underbrace{L^{-1}}_G b,$$

si ottiene la soluzione del sistema triangolare $Lz = b$.

La soluzione x si trova risolvendo $Ux = A^{(n-1)}x = z$.

Il sistema iniziale $Ax = b$ è stato trasformato in un sistema equivalente (i.e. con la stessa soluzione) $Ux = z$ con matrice triangolare superiore senza il calcolo "esplicito" della fattorizzazione.

L'algoritmo di base di Gauss nella forma "implicita" è descritto da

```
for k = 1, ..., n - 1, for i = k + 1, ..., n
     $l_{i,k} = \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}}$ ,  $b_i^{(k)} = b_i^{(k-1)} - l_{i,k} b_k^{(k-1)}$ 
    for j = k + 1, ..., n
         $a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - l_{i,k} a_{k,j}^{(k-1)}$ 
    end
    end
end
```

\hookrightarrow applico la trasformazione anche al vettore B

L'algoritmo di Gauss: complessità computazionale

Ad ogni passo si eseguono $2(n - k)^2$ addizioni e moltiplicazioni per il calcolo degli elementi $a_{i,j}^{(k)}$ e $(n - k)$ divisioni per il calcolo degli elementi $l_{i,k}$.

In totale

$$\begin{aligned} 2 \sum_{k=1}^{n-1} (n - k)^2 + \sum_{k=1}^{n-1} (n - k) &= 2 \sum_{\ell=1}^{n-1} \ell^2 + \sum_{\ell=1}^{n-1} \ell \\ &= 2 \frac{(2n-1)n(n-1)}{6} + \frac{n(n+1)}{2} \end{aligned}$$

La complessità computazionale dell'algoritmo di Gauss è circa $\frac{2n^3}{3}$ operazioni aritmetiche (flop).

L'algoritmo di Gauss con pivot parziale

L'algoritmo di Gauss non è applicabile se, ad un certo passo k , $a_{k,k}^{(k-1)} = 0$. Questo limite può essere superato **scambiando la riga k -esima con una delle righe successive della matrice che ha in colonna k -esima un elemento non nullo.**

Oss Se $a_{i,k}^{(k-1)} = 0$, $i = k, \dots, n$, la matrice A risulterebbe singolare, contro la nostra ipotesi.

In linea di principio qualsiasi elemento $a_{i,k}^{(k-1)}$, $i = k+1, \dots, n$, non nullo potrebbe essere portato in posizione pivotale. In pratica per migliorare la stabilità all'indietro dell'algoritmo, le righe vengono scambiate in modo da portare in posizione pivotale l'elemento $a_{i_k,k}^{(k-1)}$ tale che

$$|a_{i_k,k}^{(k-1)}| = \max_{i=k, \dots, n} |a_{i,k}^{(k-1)}| \text{ pivot parziale}$$

↳ scelgo come pivot l'elemento più grande (in val. assoluto) in posiz. pivotale (guardando tutta la colonna)

Risulta

- $|l_{i,k}| \leq 1$, $i = k, \dots, n$, $k = 1, \dots, n-1$
- $\max_{i,j} |u_{i,j}| = \max_{i,j} |a_{i,j}^{(n-1)}| \leq 2^{n-1} \max_{i,j} |a_{i,j}|$

Com la matrice $\begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$ non posso fare niente

se scambio $\begin{pmatrix} 2 & 3 \\ 0 & 1 \end{pmatrix}$ così posso procedere \rightarrow PERMUTAZ. DELLE RIGHE

\rightarrow cominciamo partendo in posiz. pivotate quello che im val. assoluto è maggiore, es. $\begin{pmatrix} 0 & 1 & 0 \\ 2 & 3 & 2 \\ -3 & 0 & 5 \end{pmatrix} \Rightarrow \begin{pmatrix} -3 & 0 & 5 \\ 2 & 3 & 2 \\ 0 & 1 & 0 \end{pmatrix}$

PIVOT PARZIALE

e la tecnica usata sopra posso anche usarla se pivot $\neq 0$

$$\begin{pmatrix} 1 & 1 & 0 \\ 2 & 3 & 2 \\ -3 & 0 & 5 \end{pmatrix} \Rightarrow \begin{pmatrix} -3 & 0 & 5 \\ 2 & 3 & 2 \\ 1 & 1 & 0 \end{pmatrix} \rightarrow \text{mi permette di controllare la stabilità}$$

$|-3| > |2| > |1|$

PROPRIETÀ *

$$|a_{ij}^{(k)}| = |a_{ij}^{(k-1)} - e_{ik} \cdot a_{kj}^{(k-1)}| \leq |a_{ij}^{(k-1)}| + |e_{ik}| \cdot |a_{kj}^{(k-1)}| \stackrel{k \in \mathbb{N}}{\leq} 2 \max_{i,j} |a_{ij}^{(k-1)}|$$

$$\max_{i,j} |a_{ij}^{(k)}| \leq 2 \max_{i,j} |a_{ij}^{(k-1)}| \text{ e vale } \forall k$$

$$\text{a moi interessa } k = m-1 \rightarrow \max_{i,j} |v_{ij}| \leq 2 \max_{i,j} |a_{ij}| \stackrel{m-2}{\leq} \dots \leq 2^{m-1} \cdot \max_{i,j} |a_{ij}|$$

L'algoritmo di Gauss: pivot parziale e stabilità all'indietro

come studio LA STABILITÀ dell' ALGORITMO?

→ portare tutti errori commessi per effetto delle operaz. di macchina sui dati

La stabilità viene studiata con l'analisi dell'errore all'indietro: la soluzione \tilde{x} calcolata con l'algoritmo di eliminazione di Gauss in aritmetica di macchina con dati A, b i cui elementi sono numeri di macchina, risulta essere la soluzione esatta di un problema perturbato

contiene i dati
errori sull'equazione A

$$(A + E)\tilde{x} = b,$$

$$\left\| \begin{pmatrix} -1 & 3 \\ 0 & -4 \end{pmatrix} \right\| = \begin{pmatrix} 1 & 3 \\ 0 & 4 \end{pmatrix}$$

dove per la matrice E vale la limitazione

$$|A| \leq |B| \text{ sse } |a_{ij}| \leq |b_{ij}| \quad \text{VAL. ASS.} \quad |E| \leq 4\bar{\mu}(|A| + |\tilde{L}||\tilde{U}|) + O(\bar{\mu}^2)$$

→ stabilità che cresce con m

$$\left\| \begin{pmatrix} -1 & 3 \\ 0 & 4 \end{pmatrix} \right\|_{\infty} = \max_{\infty} \{4, 3\} = 4$$

Somma
di ogni riga

e \tilde{L}, \tilde{U} sono le matrici della fattorizzazione calcolate in aritmetica di macchina ($|M| = (|m_{i,j}|)$) e la disequazione va letta elemento per elemento).

La strategia del pivot parziale migliora le proprietà di stabilità all'indietro dell'algoritmo perchè permette di controllare il modulo degli elementi delle matrici L ed U ed è pertanto essenziale per un'implementazione numericamente stabile dell'algoritmo di Gauss, anche se non è garanzia di risultati accurati. La stabilità in avanti dipende dal condizionamento della matrice.

L'algoritmo di Gauss PP, continua

In forma matriciale vale

$$\textcolor{red}{U} = A^{(n-1)} = (G_{n-1}P_{n-1} \cdots G_2P_2G_1P_1)A = GA$$

delle RIGHE
matrice di **PERMUTAZIONE** che mi porta le pivot \max (della colonna 1) in prima posiz.

dove P_k sono opportune matrici di permutazione che scambiano la riga k -esima con la riga i_k .

La matrice $G = G_{n-1}P_{n-1} \cdots G_2P_2G_1P_1$ in questo caso **non risulta triangolare**, ma vale

$$PA = LU$$

con $P = P_{n-1} \cdots P_2P_1$ e $L = PG^{-1}$

Il **metodo di eliminazione di Gauss** con la variante del **pivot parziale (PP)** fornisce una fattorizzazione **LU** di un'opportuna permutazione della matrice **A**, con la sola ipotesi di **A** non singolare.

↳ e quindi **NON** di A ORIGINALE

$$U = G_2 P_2 G_1 P_1 A \quad P = P_2 P_1$$

$$= \underbrace{G_2 P_2 G_1 P_2^{-1}}_{L^{-1}} P_2 P_1 A$$

$$P_2 = P_2^{-1} \quad \text{perché } P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$G_1 = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & 0 & 1 \end{pmatrix} \quad P_2 G_1 = \begin{pmatrix} 1 & 0 & 0 \\ b & 0 & 1 \\ a & 1 & 0 \end{pmatrix} \quad \text{e } G_1 \text{ non è più TRIANGOLARE}$$

$$P_2 G_1 P_2 = \begin{pmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ a & 0 & 1 \end{pmatrix} \quad \text{così è TRIANGOLARE}$$

$$G_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & c & 1 \end{pmatrix} \quad \Rightarrow \quad L^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ a & c & 1 \end{pmatrix} \quad \text{e } L = \begin{pmatrix} 1 & 0 & 0 \\ -b & 1 & 0 \\ -a & -c & 1 \end{pmatrix}$$

L'algoritmo di Gauss PP, continua

La **risoluzione del sistema lineare $Ax = b$** può essere ora **ricondotta** alla risoluzione di due sistemi triangolari

$$\left\{ \begin{array}{l} \textcolor{blue}{L}z = \textcolor{blue}{P}b \\ \textcolor{red}{U}x = z \end{array} \right. \begin{array}{l} \xrightarrow{\text{trovo } z} \\ \xleftarrow{\text{trovo } x} \end{array}$$

$\textcolor{blue}{P}Ax = \textcolor{blue}{P}b$
 $\textcolor{blue}{L} \cdot \textcolor{blue}{U}$

Applicando le trasformazioni elementari e le permutazioni di righe non solo alla matrice A ma anche al termine noto b , i.e.

$$z = b^{n-1} = (G_{n-1}P_{n-1} \cdots G_2P_2G_1P_1)b = Gb,$$

si ottiene la soluzione del sistema triangolare $Lz = Pb$. L'ultimo passo per calcolare la soluzione x del sistema originario consiste nel risolvere il sistema $Ux = z$ (algoritmo di Gauss con PP nella forma "implicita")

L'algoritmo di Gauss PP: esempio

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 2 & 2 & 3 \\ 1 & 2 & 1 \end{pmatrix} \xrightarrow{\text{ho il PIVOT=0} \rightarrow \text{permuto}}$$

$$P_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightsquigarrow P_1 A = \begin{pmatrix} 2 & 2 & 3 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{pmatrix}$$

$$G_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/2 & 0 & 1 \end{pmatrix} \rightsquigarrow A^{(1)} = G_1 P_1 A \begin{pmatrix} 2 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 1 & -1/2 \end{pmatrix}$$

L'algoritmo di Gauss PP: esempio

$$A^{(1)} = \begin{pmatrix} 2 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 1 & -1/2 \end{pmatrix}$$

Id, manifatturante

↑

P₂

↑

Id, manifatturante

↑

P₂

il max lo cerco solo
sotto al PIVOT

↑

P₂A⁽¹⁾

↑

Id, manifatturante

↑

P₂A⁽¹⁾

$$G_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \rightsquigarrow A^{(2)} = G_2 P_2 A^{(1)} = \underbrace{\begin{pmatrix} 2 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & -3/2 \end{pmatrix}}_{u}$$

Esempio, continua

$$G = G_2 P_2 G_1 P_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ -1 & -1/2 & 1 \end{pmatrix} \rightarrow \text{non è TRIANG INF.}$$

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow \text{per portare } G \text{ in forma TRIANGOLARE inferiore}$$

$$L = PG^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/2 & 1 & 1 \end{pmatrix}$$

L'algoritmo di Gauss: determinante, termini noti multipli

Il **metodo di eliminazione di Gauss con PP** può essere usato per calcolare

- **determinante di una matrice:**

$$\det(PA) = \det(LU)$$

$$\det(PA) = \det(P) \cdot \det(A) = \underbrace{\det(L)}_{=1} \underbrace{\det(U)}_{\substack{\text{I UNICI} \\ \text{KEL' U' K' E' DIAGONALI}}}$$

$$\det(P) \det(A) = \det(L) \det(U) = \prod_{k=1}^n u_{k,k}$$

quindi, poichè $\det(P) = (-1)^r$, il determinante di A è dato dal prodotto degli elementi principali di U a meno del segno che è determinato dal numero r degli scambi di riga effettuati.

- **termini noti multipli:** $AX = B$ con $B \in \mathbb{R}^{n \times m}$, $m \leq n$

Applicando l'algoritmo con PP anche alla matrice B, ci si riconduce alla **soluzione di m sistemi triangolari**

$$Ux^{(i)} = L^{-1}Pb^{(i)} = Gb^{(i)} \quad i = 1, \dots, m$$

dove $B = (b^{(1)}, \dots, b^{(m)})$, $X = (x^{(1)}, \dots, x^{(m)})$ e $b^{(i)}, x^{(i)}$ sono vettori colonna.

Il costo computazionale totale è $O\left(\frac{2n^3}{3} + 2mn^2\right)$ flop.

L'algoritmo di Gauss: A singolare

Il metodo di Gauss con il pivot parziale si può ancora applicare anche se la matrice A del sistema è singolare.

Se al k -esimo passo risulta

$$a_{i,k}^{(k-1)} = 0, \quad i = k, \dots, n$$

↑ tutti gli elementi della colonna k -esima: NON posso usare PIVOT PARZIALE!
└ mom faccio niente e prosegua

non si esegue nessuna trasformazione, i.e.

$$G_k = I, \quad P_k = I \quad \left(\begin{matrix} U_{11} & \dots & \overset{k}{\downarrow} \\ \dots & \dots & 0 \\ U_{m1} & \dots & U_{mm} \end{matrix} \right)$$

e si prosegue con la colonna $k+1$ -esima.

⇒ riesco comunque a trovare una soluz.
Se il sistema è consistente

La matrice $U = A^{(n-1)}$ avrà l'elemento $u_{k,k}$ nullo e risulta singolare.

Ma nelle ipotesi di consistenza del sistema, si può calcolare una soluzione con l'algoritmo di sostituzione all'indietro.

Definisco il RANGO come # colonne ≠ 0

L'algoritmo di Gauss con pivot totale

Il metodo di eliminazione di Gauss può essere implementato anche nella variante del **pivot totale** (PT):

ad ogni passo viene portato in posizione pivotale l'elemento $a_{i_k, j_k}^{(k-1)}$ tale che

$$|a_{i_k, j_k}^{(k-1)}| = \max_{i, j=k, \dots, n} |a_{i, j}^{(k-1)}|.$$

Tale strategia richiede non solo uno scambio di righe ma anche di colonne e porta alla seguente fattorizzazione

↳ mom cambia
il sistema

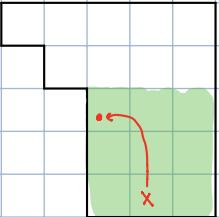
↳ cambia il sistema perché
sto rimanimando le variabili

$$PAQ = LU,$$

con P e Q sono opportune matrici di permutazione.

La risoluzione del sistema lineare $Ax = b$ può essere ora ricondotta alla risoluzione di due sistemi triangolari e ad riordinamento delle varie componenti descritto dalla matrice Q :

$$\left\{ \begin{array}{l} Lz = Pb \rightarrow \text{trovo } z \\ Uy = z \leftarrow \text{trovo } y \\ x = Qy \leftarrow \end{array} \right.$$



→ cerca il pivot massimo qua dentro
Scambio di righe e colonne

$$\left(\begin{array}{ccccc} 1 & & & & \\ & 1 & & & \\ & & 0 & 0 & 1 \\ & 0 & 1 & 0 & \\ & 1 & 0 & 0 & \end{array} \right) \tilde{A} \left(\begin{array}{ccccc} 1 & & & & \\ & 1 & & & \\ & & 0 & 0 & 1 \\ & 0 & 1 & 0 & \\ & 1 & 0 & 0 & \end{array} \right)$$

↓
scambia terza
con quinta riga

↓
scambia le colonne

L'algoritmo di Gauss PT: esempio

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 2 & 2 & 3 \\ 1 & 2 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 5 \\ 15 \\ 8 \end{pmatrix}$$

0 in posiz. pivotale
com
com
PIVOT PARZIALE cerca nella stessa colonna
PIVOT TOTALE cerca in tutta la matrice

$$P_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad Q_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \rightsquigarrow P_1 A Q_1 = \begin{pmatrix} 3 & 2 & 2 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Scambio righe
Scambio colonne

$$G_1 = \begin{pmatrix} 1 & 0 & 0 \\ -1/3 & 1 & 0 \\ -1/3 & 0 & 1 \end{pmatrix} \rightsquigarrow A^{(1)} = G_1 P_1 A Q_1 = \begin{pmatrix} 3 & 2 & 2 \\ 0 & 1/3 & -2/3 \\ 0 & 4/3 & 1/3 \end{pmatrix}$$

L'algoritmo di Gauss PT: esempio

$$A^{(1)} = \begin{pmatrix} 3 & 2 & 2 \\ 0 & 1/3 & -2/3 \\ 0 & 4/3 & 1/3 \end{pmatrix}, \quad b^{(1)} = G_1 P_1 b = \begin{pmatrix} 15 \\ 0 \\ 3 \end{pmatrix}$$

$$P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad Q_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightsquigarrow P_2 A^{(1)} Q_2 = \begin{pmatrix} 3 & 2 & 2 \\ 0 & 4/3 & 1/3 \\ 0 & 1/3 & -2/3 \end{pmatrix}$$

$$G_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1/4 & 1 \end{pmatrix} \rightsquigarrow A^{(2)} = G_2 P_2 A^{(1)} Q_2 = \begin{pmatrix} 3 & 2 & 2 \\ 0 & 4/3 & 1/3 \\ 0 & 0 & -3/4 \end{pmatrix}$$

L'algoritmo di Gauss PT: esempio

$$U = A^{(2)} = \begin{pmatrix} 3 & 2 & 2 \\ 0 & 4/3 & 1/3 \\ 0 & 0 & -3/4 \end{pmatrix}, \quad b^{(2)} = G_2 P_2 b^{(1)} = \begin{pmatrix} 15 \\ 3 \\ -3/4 \end{pmatrix}$$

$$y = U^{-1} b^{(2)} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} \rightsquigarrow x = Q_1 Q_2 y = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

scambia
1 con 3 Id

L'algoritmo di Gauss PT, continua

La stabilità numerica dell'algoritmo con il pivot totale è teoricamente migliore di quella del pivot parziale.

Ma la sua implementazione richiede $(n - k + 1)^2$, $k = 1, 2, \dots, n - 1$, confronti contro gli $n - k$, $k = 1, 2, \dots, n - 1$, del pivot parziale ed è quindi più costosa.

In pratica la proprietà di stabilità numerica dell'algoritmo con il pivot parziale è più che adeguata per la risoluzione di un sistema lineare con l'algoritmo di eliminazione di Gauss.

L'algoritmo di Gauss: aspetti implementativi

Qualora non sia necessario mantenere le informazioni sulla matrice A , nell'implementare il metodo di eliminazione di Gauss, l'area di memoria riservata per la matrice A può essere utilizzata per memorizzare la matrice L nella parte strettamente triangolare inferiore e la matrice U nella parte triangolare superiore.

Nel caso della variante del pivot parziale, non è necessario eseguire effettivamente gli scambi di riga. La posizione della i -esima riga può essere individuata utilizzando un vettore $riga$ che inizialmente ha componenti $riga_i = i$, $i = 1, \dots, n$. Quando due righe k e ℓ vengono scambiate, esse rimangono nella locazione originaria ed il vettore $riga$ tiene conto dell'ordine, scambiando fra loro $riga_\ell$ e $riga_k$.

Discorso analogo si applica per il pivot totale dove si utilizza un vettore col per tener conto delle permutazioni di colonne.

for $k=1, \dots, u-1$

for $i = k+1, \dots, u$

$$a_{ik} = \frac{a_{ik}}{a_{kk}}$$

for $j = k+1, \dots, m$

$$a_{ij} = a_{ij} - a_{ik} \cdot a_{kj}$$

end

end

end

Esempio

$$A = \begin{pmatrix} 6 & -5 & 1 & 2 \\ -2 & -3 & 4 & 7 \\ -3 & 4 & 5 & 8 \\ 2 & -3 & 4 & 6 \end{pmatrix}$$

Esempio

$$A = \begin{pmatrix} 6 & -5 & 1 & 2 \\ \frac{-1}{3} & \frac{-14}{3} & \frac{13}{3} & \frac{23}{3} \\ \frac{2}{3} & \frac{9}{6} & \frac{33}{6} & \frac{27}{3} \\ \frac{1}{3} & \frac{-4}{3} & \frac{11}{3} & \frac{16}{3} \end{pmatrix}$$

dopo il 1° passo

Esempio

$$A = \begin{pmatrix} \frac{6}{-1} & \frac{-5}{-14} & \frac{1}{13} & \frac{2}{23} \\ \frac{3}{-1} & \frac{3}{-9} & \frac{3}{193} & \frac{3}{321} \\ \frac{2}{1} & \frac{28}{2} & \frac{28}{17} & \frac{28}{22} \\ \frac{1}{3} & \frac{7}{7} & \frac{7}{7} & \frac{7}{7} \end{pmatrix}$$

Esempio

$$A = \left(\begin{array}{cccc} 6 & -5 & 1 & 2 \\ \frac{-1}{3} & \frac{-14}{3} & \frac{13}{3} & \frac{23}{3} \\ \frac{-1}{2} & \frac{-9}{28} & \frac{193}{28} & \frac{321}{28} \\ \frac{1}{3} & \frac{2}{7} & \frac{68}{193} & \frac{-173}{193} \end{array} \right)$$

Diagram illustrating the structure of matrix A. The matrix is divided into four colored regions: a green top-right corner, an orange bottom-left corner, a green bottom-right corner, and a green bottom-center column. The green regions are labeled with 'u' at the top and 'L' at the bottom. The orange region is labeled with 'L' at the bottom.

Algoritmo di Gauss-Jordan

L'algoritmo di Gauss-Jordan trasforma la matrice A in una matrice diagonale in n passi. Il passo k -esimo, $A^{(k)} = J_k A^{(k-1)}$, è descritto da

$$J_k = \begin{pmatrix} 1 & \dots & -l_{1,k} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -l_{k-1,k} & 0 & \dots & 0 \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & -l_{k+1,k} & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -l_{n,k} & 0 & \dots & 1 \end{pmatrix},$$

al pivot

eliminiamo anche la parte superiore
della diagonale, oltre che alla
parte inferiore (al pivot)

↳ è un'estensione di
quella due fasi di Gauss
che eliminava solo sotto
i.e. pivot

dove $l_{i,k} = \frac{a_{i,k}^{k-1}}{a_{k,k}^{k-1}}$, $i = 1, \dots, n$. In forma matriciale

$$D = A^{(n)} = (J_n \dots J_2 J_1) A$$

con D matrice diagonale. Applicando le stesse trasformazioni al termine noto b , i.e.

$$d = b^{(n)} = (J_n \dots J_2 J_1) b,$$

si ottiene un sistema diagonale $Dx = d$ equivalente a quello di partenza.



Exercise

Scrivi una pseudocodifica per l'algoritmo di Gauss-Jordan.

La sua **stabilità** può essere **migliorata** applicando la tecnica del **pivot parziale** nella parte inferiore della matrice. In questo caso risulta che l'algoritmo di Gauss-Jordan con pivot parziale è stabile quanto quello di Gauss con pivot parziale.

L'algoritmo di **Gauss-Jordan** richiede **circa n^3 operazioni aritmetiche**.

Può essere applicato per il calcolo dell'inversa. È normalmente possibile riformulare i problemi che richiedono il calcolo della matrice inversa nella risoluzione di sistemi lineari, evitando il calcolo esplicito dell'inversa.

es. calcola $A^{-1}b = x \Rightarrow Ax = b$ e così evito il calcolo dell'inversa

risolvendo un sist. lineare

Un esempio introduttivo

Supponiamo che

- A sia non singolare
- $A = LU$
- $A^T = A$ (simmetrica)

$$U = \begin{pmatrix} U_{11} & & & \\ & U_{22} & & \\ & & \ddots & \\ & & & U_{nn} \end{pmatrix} = \begin{pmatrix} U_{11} & & & \\ & U_{22} & & \\ & & \ddots & \\ & & & U_{nn} \end{pmatrix} \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}$$

Allora scrivendo

$$U = D\tilde{U}$$

con $D = \text{diag}(u_{1,1}, \dots, u_{n,n})$ risulta $\tilde{u}_{i,i} = 1, i = 1, \dots, n$, dalla simmetria della matrice si ha

$$A^T = U^T L^T = \underbrace{\tilde{U}^T}_{U = D\tilde{U} \text{ e } D^T = D} D L^T = L D \tilde{U} = A$$

Di conseguenza per l'unicità della fattorizzazione LU si ottiene

$$L = \tilde{U}^T$$

e

$$A = LDL^T$$

Matrici speciali

- $A \in \mathbb{R}^{n \times n}$ è simmetrica, $A^T = A$, e definita positiva,
 $x^T A x > 0$ per ogni $x \in \mathbb{R}^n \setminus \{0\}$
- $A \in \mathbb{R}^{n \times n}$ è diagonale dominante per colonne

$$|a_{j,j}| > \sum_{i=1, i \neq j}^n |a_{i,j}|, \quad j = 1, \dots, n$$

- $A \in \mathbb{R}^{n \times n}$ è tridiagonale

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

Algoritmo di Cholesky

Sia A è simmetrica e definita positiva, allora

$$\downarrow$$
$$A = A^T \Rightarrow a_{ij} = a_{ji}$$

$$A = HH^T$$



$$x^T A x = \boxed{x^T} \cdot \boxed{A x} \in \mathbb{R} \text{ e se } x^T A x > 0 \forall x \in \mathbb{R}$$

dove H è triangolare inferiore con $h_{i,i} > 0, i = 1, \dots, n$.

$$\text{il generico } a_{ij} = \sum_{k=1}^{\min(i,j)} h_{ik} \cdot h_{jk} \quad i=1 \dots n \quad j=1 \dots i \quad a_{ii} = \sum_{k=1}^{i-1} h_{ik}^2 + h_{ii}^2 \quad i=1 \dots n \quad a_{ij} = \sum_{k=1}^{j-1} h_{ik} h_{jk} + h_{ij} h_{jj} \Rightarrow h_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} h_{ik}^2} \quad h_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} h_{ik} h_{jk} \right) \cdot \frac{1}{h_{jj}}$$

Gli elementi di H possono essere calcolati con l'algoritmo di Cholesky

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

$$h_{j,j} = \sqrt{a_{j,j} - \sum_{k=1}^{j-1} h_{j,k}^2}$$

$$h_{i,j} = \frac{1}{h_{j,j}} \left(a_{i,j} - \sum_{k=1}^{j-1} h_{i,k} h_{j,k} \right), \quad i = j+1, \dots, n, \quad j \neq n,$$

procedendo per colonne.

Algoritmo di Cholesky: stabilità e complessità computazionale

L'algoritmo di Cholesky è sempre numericamente stabile all'indietro. Vale in fatti un risultato analogo a quello visto per l'algoritmo di eliminazione di Gauss e la stabilità all'indietro è legata al modulo degli elementi della matrice H . Da $a_{jj} = \sum_{k=1}^j h_{j,k}^2$ si ottiene

$$|h_{j,k}| \leq \sqrt{a_{j,j}}, \quad k = 1, \dots, j, j = 1, \dots, n$$

maggiorati

e quindi il valore assoluto di tutti gli elementi di H sono limitati e non sono necessari scambi di riga.

Per il calcolo degli elementi $h_{i,j}$ sono richieste j operazioni moltiplicative e j operazioni additive, e quindi per il calcolo della i -esima riga si effettuano $2 \sum_{j=1}^{i-1} j \approx i^2$ operazioni aritmetiche, mentre il calcolo di $h_{j,j}$ è di ordine inferiore ma richiede anche l'estrazione di radice. La complessità computazionale dell'algoritmo di Cholesky è $O(\frac{n^3}{3})$ con n estrazioni di radice.

Matrici a dominanza diagonale

L' implementazione dell'algoritmo di Gauss con PP **non** richiede lo scambio di righe: l'elemento pivotale è al posto giusto ad ogni passo.

L'algoritmo di Gauss di base è sempre **stabile**.

Matrici tridiagonali

La **fattorizzazione LU** in questo caso può essere **calcolata facilmente**.
Osservando che

$$L = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_2 & 1 & 0 & \cdots & 0 \\ 0 & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & l_{n-1} & 1 & 0 \\ 0 & \cdots & 0 & l_n & 1 \end{pmatrix}, \quad U = \begin{pmatrix} u_1 & v_1 & 0 & \cdots & 0 \\ 0 & u_2 & v_2 & \cdots & 0 \\ 0 & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{n-1} & v_{n-1} \\ 0 & \cdots & 0 & 0 & u_n \end{pmatrix}$$

si ha

$$\left\{ \begin{array}{l} v_i = c_i, \quad i = 1, 2, \dots, n, \\ \end{array} \right. \quad \text{e} \quad \left\{ \begin{array}{l} u_1 = b_1 \\ \text{for } i = 2, \dots, n \\ \quad l_i = a_i / u_{i-1} \\ \quad u_i = b_i - l_i c_{i-1} \\ \text{end} \end{array} \right.$$

Tale algoritmo richiede $3(n - 1)$ operazioni aritmetiche e la soluzione del sistema originale si riduce alla soluzione di due sistemi bidiagonali.
Esercizio: determina il costo computazionale della risoluzione di tali sistemi.

Errore e residuo

Il **residuo** di una soluzione approssimata \tilde{x} è definito da

$$r = b - A\tilde{x}.$$

Se **A è una matrice non singolare**, vale

$$\|x - \tilde{x}\| = 0 \quad \text{se e soltando se} \quad \|r\| = 0$$

$$\begin{aligned} r &= b - A\tilde{x} = Ax - A\tilde{x} \\ &= A(x - \tilde{x}) \\ \Rightarrow (x - \tilde{x}) &= A^{-1}r \\ \|x - \tilde{x}\| &= \|A^{-1}r\| \leq \|A^{-1}\| \cdot \|r\| \\ \frac{\|x - \tilde{x}\|}{\|x\|} &\leq \|A^{-1}\| \frac{\|r\|}{\|x\|} \leq \|A^{-1}\| \cdot \|A\| \cdot \frac{\|r\|}{\|b\|} = \text{cond}(A) \frac{\|r\|}{\|b\|} \\ b = Ax &\quad \|b\| \leq \|A\| \cdot \|x\| \\ \Rightarrow \frac{1}{\|x\|} \leq \frac{\|b\|}{\|b\|} &\quad \text{red.} \end{aligned}$$

Si può stimare l'errore relativo in \tilde{x} valutando la norma del **residuo relativo** $\frac{\|r\|}{\|b\|}$.

In pratica un “piccolo” residuo relativo mi assicura un “piccolo” errore relativo solo se la matrice è ben condizionata:

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \text{cond}(A) \frac{\|r\|}{\|b\|}.$$

Errore e residuo, continua

Per l'errore relativo rispetto alla soluzione approssimata \tilde{x} si ottiene

$$\frac{\|x - \tilde{x}\|}{\|\tilde{x}\|} \leq \text{cond}(A) \frac{\|r\|}{\|A\| \cdot \|\tilde{x}\|},$$

che conferma il ruolo del condizionamento della matrice $\text{cond}(A)$ nella stima dell'errore attraverso la valutazione del residuo e che un "piccolo" residuo non mi assicura una soluzione accurata.

Oss: \tilde{x} è soluzione esatta di un problema perturbato e

$$\frac{\|r\|}{\|A\| \cdot \|\tilde{x}\|} \leq \frac{\|E\|}{\|A\|}.$$

$$\begin{aligned} (A + E)\tilde{x} &= b = Ax \\ A\tilde{x} + E\tilde{x} &= Ax \\ r = b - A\tilde{x} &\Rightarrow A\tilde{x} = E\tilde{x} \\ \|r\| &\leq \|E\| \|\tilde{x}\| \\ \frac{\|r\|}{\|\tilde{x}\|} &\leq \|E\| \quad \frac{\|r\|}{\|\tilde{x}\| \|A\|} \leq \frac{\|E\|}{\|A\|} \end{aligned}$$

La sua analisi ci permette di concludere che

- un **residuo "grande"** implica un **"grande"** errore (all'indietro) nella matrice A che mi dice che l'algoritmo usato per calcolare \tilde{x} è instabile (all'indietro)
- un **algoritmo stabile** (all'indietro) **fornisce un "piccolo" errore nel residuo**, che garantisce soluzione accurata solo se la matrice è ben condizionata.

Gli algoritmi visti per

- il prodotto matrice-vettore
- la soluzione di sistemi lineari triangolari

sono **stabili all'indietro** e quindi ci assicurano che l'errore nel residuo è un piccolo multiplo della precisione di macchina.

Consideriamo la risoluzione in sequenza dei due sistemi triangolari, prevista dalla procedura. Risolvendo il sistema $Lz = b$ troviamo quindi una soluzione \tilde{z} per cui vale

$$\frac{\|z - \tilde{z}\|}{\|\tilde{z}\|} \leq \text{cond}(L) \frac{\|r\|}{\|b\|}.$$

Segue quindi che per la soluzione \tilde{x} del sistema $Ux = \tilde{z}$ si ha

$$\frac{\|x - \tilde{x}\|}{\|\tilde{x}\|} \leq \text{cond}(U) \frac{\|z - \tilde{z}\|}{\|\tilde{z}\|} \leq \text{cond}(L) \text{cond}(U) \frac{\|r\|}{\|b\|}.$$

Poichè $\text{cond}(A) \leq \text{cond}(L) \cdot \text{cond}(U)$ il risultato \tilde{x} può essere affetto da un errore in avanti significativo anche se i due sistemi triangolari sono stati risolti con algoritmi stabili all'indietro.

Errore e residuo, esempio

Consideriamo il sistema lineare $Ax = b$ dove

$$A = \begin{pmatrix} 100 & 99 \\ 99 & 98 \end{pmatrix}, b = \begin{pmatrix} 199 \\ 197 \end{pmatrix}.$$

Supponiamo che due diversi algoritmi forniscano le seguenti soluzioni approssimate

$$\tilde{x}_1 = \begin{pmatrix} 1.01 \\ 1.01 \end{pmatrix}, \tilde{x}_2 = \begin{pmatrix} 2.97 \\ -0.99 \end{pmatrix}.$$

Basandoci sui residui

$$r_1 = \begin{pmatrix} -1.99 \\ -1.97 \end{pmatrix}, r_2 = \begin{pmatrix} 0.01 \\ -0.01 \end{pmatrix}$$

risulta da preferire \tilde{x}_2 , mentre è \tilde{x}_1 la più accurata. La soluzione esatta è infatti

↳ le informaz. date dai RESIDUO
possono non essere sempre esatte

$$x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Sia \tilde{x} una soluzione approssimata della soluzione x in un sistema lineare calcolata, per esempio, con l'algoritmo di eliminazione di Gauss.

Il vettore errore $e = x - \tilde{x}$ è soluzione (esatta) del sistema $Ae = r$, che risolto in aritmetica di macchina fornirà una stima dell'errore con cui "raffinare" la soluzione iniziale.

Sia $x_0 = \tilde{x}$ l'algoritmo di *raffinamento iterativo* si propone di migliorare la stima iniziale potenzialmente fino alla precisione di macchina u nel seguente modo:

```
x(0)=\tilde{x}, d=x(0)
for k=1,2,...,kmax
    r(k-1)=b-Ax(k-1)
    risolvi A(e(k-1))=r(k-1)      in "doppia" precisione
    x(k)=x(k-1)+e(k-1)
    if ||e(k-1)||*||x(k-1)|| >= ||d||*||x(k) ||
        stop "non converge" end
    if ||e(k-1)|| <= u ||x(k)|| then stop "stima ottenuta" end
    d=e(k-1)
end
```

L' algoritmo richiede la memorizzazione sia della matrice iniziale A per calcolare ad ogni passo iterativo il residuo r_{k-1} , sia delle matrici L , U , P della fattorizzazione per risolvere in maniera efficiente ad ogni passo il sistema lineare $Ae_{k-1} = r_{k-1}$ per calcolare le correzioni e_{k-1} .

Inoltre, poichè il calcolo dei residui comporta la cancellazione, risulta indispensabile che siano calcolati con precisione più elevata per ottenere un effettivo miglioramento.

Si può dimostrare che se la matrice A non è troppo malcondizionata il test di arresto viene verificato. In pratica se in poche iterazioni non è soddisfatto è inutile proseguire: la matrice è molto malcondizionata e può capitare che le norme delle correzioni $\|e_{k-1}\|$ vadano aumentando.

Un po' di storia

Joseph F. Grcar *"How ordinary elimination became Gaussian elimination"*
Historia Mathematica, Volume 38, Issue 2 (2011), pp. 163–218

Un algoritmo equivalente a quello di eliminazione di Gauss appare nei testi matematici **Jiuzhang Suanshu (Nove Capitoli dell'Arte Matematica)** dell'antica Cina. Il metodo per la risoluzione delle equazioni lineari risale quindi al **terzo secolo AC!**

Nel Rinascimento in Europa si trovano dei riferimenti al metodo in alcuni libri di testo. Tale conoscenza nasce da contatti con l'Oriente.

Newton in uno scritto del 1670 e Rolle nel 1690 forniscono separatamente le regole per risolvere le equazioni eliminando le variabili, ma **non** usano la parola **eliminazione**, che Lacroix introduce nel 1804 nei libri di testo scolastici.

Gauss nel 1810 propone un metodo risolutivo speciale per risolvere problemi ai minimi quadrati che possono essere formulati con le equazioni normali.

Il suo metodo è adottato e fortemente raccomandato per risolvere problemi di astronomia e geodesia (*). Viene migliorato e poi sostituito da altre tecniche, per utilizzare al meglio le opportunità offerte dal uso di strumenti di calcolo, quali le calcolatrici professionali (Doolittle, Cholesky).

Nella prima metà del XIX secolo il metodo è presentato nei libri di algebra come "**metodo di eliminazione**", ma dalla seconda metà del XIX secolo nei libri di astronomia e geodesia l'approccio per problemi ai minimi quadrati inizia ad essere citato come "**metodo di Gauss**".

(*) La Geodesia si occupa della misura e della rappresentazione della Terra, del suo campo gravitazionale, delle maree e dei movimenti della crosta.

L'invenzione dei calcolatori elettronici permette di utilizzarlo in campi scientifici diversi e inizia a essere citato come "**metodo di eliminazione di Gauss**" dopo la seconda Guerra Mondiale. Von Neumann [1947] è l'ultimo matematico a definire la tecnica di eliminazione secondo Lacroix [1804] e Gauss [1809].

Il riferimento alla tecnica risolutiva come "**metodo di eliminazione di Gauss**" ha quindi un valore onorifico, ma al suo sviluppo hanno contribuito diversi studiosi.

Il 27 settembre 1998: Larry Page e Sergey Brin, due studenti di Stanford, fondano Google per **classificare le pagine web in base alla loro importanza (Page Rank)** assumendo che

- l'importanza di una pagina dipende dai link e non dal contenuto
- l'importanza di una pagina viene distribuita in parti uguali alle pagine a cui punta
- l'importanza di una pagina è la somma delle frazioni di importanza delle pagine che puntano ad essa

Page Rank, continua

Indichiamo con x_i il page rank della pagina i , allora

$$x_i = p_{i1}x_1 + p_{i2}x_2 + p_{i3}x_3 + \cdots + p_{in}x_n$$

dove p_{ij} è la probabilità di passare dalla pagina j alla pagina i che vale

$$p_{ij} = \begin{cases} \frac{1}{\ell_j} & \text{se esiste il link } j \rightarrow i \\ 0 & \text{altrimenti} \end{cases}$$

con ℓ_j numero totale di link uscenti dalla pagina j . Otteniamo il sistema lineare

$$x = Px$$

di dimensione n pari al numero totale delle pagine web che sono circa 21 miliardi!!

Per risolvere il problema non basta utilizzare calcolatori più potenti, come Cray XT5 che esegue 2.3×10^{15} flops, c'è bisogno di costruire algoritmi efficienti che tengano conto della struttura e della sparsità della matrice.

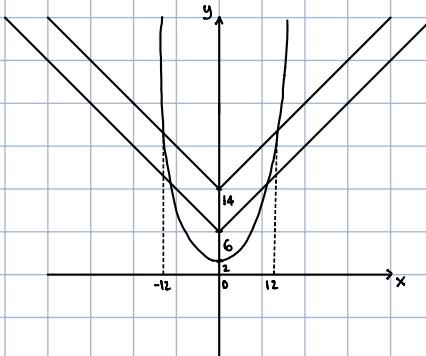
esame 25.2.2020

$$A = \begin{pmatrix} 2 & \alpha & -4 \\ \alpha & 2 & \alpha \\ -4 & -\alpha & 10 \end{pmatrix}$$

- disegna il grafico di $\alpha \rightarrow \|A\|_\infty$

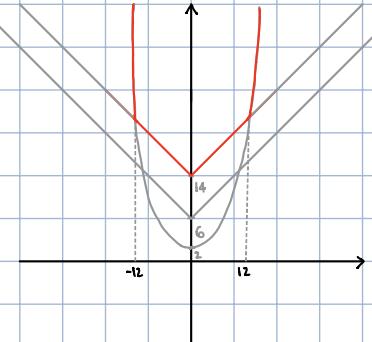
$$\|A\|_\infty = \max_{i=1,2,3} \sum_{j=1}^3 |a_{ij}| \\ = \max \{|\alpha|+6, 2|\alpha|+2, |\alpha|+14\}$$

$$A = \begin{pmatrix} 2 & \alpha & -4 \\ \alpha & 2 & \alpha \\ -4 & -\alpha & 10 \end{pmatrix} \rightarrow |\alpha|+6 \\ \rightarrow 2|\alpha|+2 \\ \rightarrow |\alpha|+14$$



$$\alpha > 0 \quad 2\alpha + 2 \stackrel{?}{=} \alpha + 14 \\ \Rightarrow \alpha = 12 \text{ (INTERSEZ.)}$$

$$\Rightarrow \text{LA NORMA INFINTO di } A \text{ e} \quad \|A\|_\infty = \begin{cases} |\alpha| + 14 & |\alpha| \leq 12 \\ 2|\alpha| + 2 & |\alpha| > 12 \end{cases}$$

- Calcola fattorizz. LU di A, determinando per quali α si può ottenere $LU = A$

$$\begin{array}{c} G_1 \\ \downarrow \\ \left(\begin{array}{ccc} 1 & 0 & 0 \\ -\frac{\alpha}{2} & 1 & 0 \\ \frac{\alpha}{2} & 0 & 1 \end{array} \right) \left(\begin{array}{ccc} 2 & \alpha & -4 \\ \alpha & 2 & \alpha \\ -4 & -\alpha & 10 \end{array} \right) = \left(\begin{array}{ccc} 2 & \alpha & -4 \\ 0 & \frac{4-\alpha^2}{2} & 3\alpha \\ 0 & \alpha & 2 \end{array} \right) \end{array} \quad \begin{array}{l} -\frac{\alpha^2}{2} + 2 = \frac{4-\alpha^2}{2} \\ 2\alpha - \alpha = \alpha \\ \frac{4}{2}\alpha + \alpha = 3\alpha \\ -8 + \alpha + 10 = 2 \end{array}$$

$$\begin{array}{c} G_2 \\ \downarrow \\ \left(\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{-2\alpha}{4-\alpha^2} & 1 \end{array} \right) \left(\begin{array}{ccc} 2 & \alpha & -4 \\ 0 & \frac{4-\alpha^2}{2} & 3\alpha \\ 0 & \alpha & 2 \end{array} \right) = \left(\begin{array}{ccc} 2 & \alpha & -4 \\ 0 & \frac{4-\alpha^2}{2} & 3\alpha \\ 0 & 0 & \frac{8(1-\alpha^2)}{(4-\alpha^2)^2} \end{array} \right) \end{array} \quad \begin{array}{l} \frac{4-\alpha^2}{2} \text{ per essere pivot } \neq 0 \Rightarrow \text{quando } \alpha = 0? \text{ se } \alpha = \pm 2 \Rightarrow \alpha \neq \pm 2 \end{array}$$

$$\begin{array}{c} \Gamma \rightarrow G_1^{-1} \cdot G_2^{-1} \\ L = \left(\begin{array}{ccc} 1 & 0 & 0 \\ \frac{\alpha}{2} & 1 & 0 \\ -2 & \frac{2\alpha}{4-\alpha^2} & 1 \end{array} \right) \text{ Se } \alpha \neq \pm 2 \\ \det(A) = \det(L) \cdot \det(U) = \frac{2}{2} \cdot 8(1-\alpha^2) = 0 \text{ per } \alpha \neq 1 \end{array}$$

- se $|a| < 4$ ie Pivot Parziale al primo passo scambia la prima con la terza riga

- se $d = 3$, calcola $PA = LU$

$$A = \begin{pmatrix} 2 & 3 & -4 \\ 3 & 2 & 3 \\ -4 & -3 & 10 \end{pmatrix} \quad P_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \text{ e così scambio 1 con 3 riga} \quad P_1 A = \begin{pmatrix} -4 & -3 & 10 \\ 3 & 2 & 3 \\ 2 & 3 & -4 \end{pmatrix}$$

$$\begin{matrix} G_1 \\ \downarrow \end{matrix} \quad \begin{pmatrix} 1 & 0 & 0 \\ \frac{3}{4} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} -4 & -3 & 10 \\ 3 & 2 & 3 \\ 2 & 3 & -4 \end{pmatrix} = \begin{pmatrix} -4 & -3 & 10 \\ 0 & -\frac{1}{4} & \frac{21}{2} \\ 0 & \frac{3}{2} & 1 \end{pmatrix}$$

dato che $|\frac{1}{4}| < |\frac{3}{2}|$ allora P_2 deve scambiare 2 e 3 riga

$$P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & \frac{1}{4} \end{pmatrix} \cdot \begin{pmatrix} -4 & -3 & 10 \\ 0 & \frac{3}{2} & 1 \\ 0 & -\frac{1}{4} & \frac{21}{2} \end{pmatrix} = \begin{pmatrix} -4 & -3 & 10 \\ 0 & \frac{3}{2} & 1 \\ 0 & 0 & \frac{1}{6} + \frac{21}{2} \end{pmatrix} \leftarrow U$$

$$\begin{matrix} \uparrow \\ G_2 \end{matrix}$$

$$U = G_2 P_2 G_1 P_1 A = G_2 P_2 G_1 P_2 \overset{P}{\overbrace{P_2 P_1}} A$$