

Appunti di Calcolo Numerico

Tommaso Renzoni

Questi appunti **NON** sono pensati per studiare da zero.

Sono stati scritti velocemente a lezione e potrebbero contenere errori o frasi poco chiare.

Consiglio di usarli solo come supporto insieme alle lezioni del prof. Massei e al materiale ufficiale.

I. Appunti presi a lezione

Analisi dell'errore per funzioni scalari	2
Richiami di algebra lineare	13
Sistemi lineari	32
Interpolazione di funzioni	51
Integrazione numerica	64
Calcolo di autovalori e autovettori	73

II. Schemi pratici per lo scritto 76

I. Appunti presi a lezione

Analisi dell'errore per funzioni scalari

1. Numeri in virgola mobile

Fissato $\beta > 1$, $\beta \in \mathbb{N}$, si può trovare:

- $e \in \mathbb{Z}$ esponente
- $\{\alpha_j\}$ successione

Tale che $x = \text{sgn}(x)\beta^e \sum_{j=1}^{\infty} \alpha_j \beta^{-j}$

Esempio (pi greco):

$$3.14 = 1 \cdot 0.314 \cdot 10^1 = 1 \cdot 0.0314 \cdot 10^3$$

Sorgono un paio di problemi:

- la rappresentazione non è unica
- $0.\overline{9} = 1$ e dunque hanno la stessa rappresentazione

Teorema (di rappresentazione):

Dati $\beta > 1$, $\beta \in \mathbb{N}$, $x \in \mathbb{R}$, $x \neq 0$

Esiste ed è unica la rappresentazione in virgola mobile normalizzata $x = \text{sgn}(x)\beta^3 \sum_{j=1}^{\infty} \alpha_j \beta^{-j}$
tale che:
mantissa

- $\alpha_j \neq 0$
- $\nexists k \in \mathbb{N} : \alpha_j = \beta - 1 \quad \forall j > k$

Proposizione:

Si ha che $\frac{1}{\beta} \leq \sum_{j=1}^{\infty} \alpha_j \beta^{-j} < 1$

Dimostrazione:

- $\sum_{j=1}^{\infty} \alpha_j \beta^{-j} \geq \alpha_1 \beta^{-1} \geq \beta^{-1} = \frac{1}{\beta}$ visto che $\alpha_1 \neq 0$
- $\sum_{j=1}^{\infty} \alpha_j \beta^{-j} < (\beta - 1) \sum_{j=1}^{\infty} \beta^{-j} = (\beta - 1) \left(\frac{1}{1 - \beta^{-1}} - 1 \right) = (\beta - 1) \frac{\beta^{-1}}{1 - \beta^{-1}} =$

$$= (\beta - 1) \frac{1}{\beta - 1} = 1 \quad \square$$

1.1. Numeri di macchina

Definizione (numeri di macchina):

Dati $\beta \in \mathbb{N}$, $m \in \mathbb{N}$, $L, U \in \mathbb{Z}$: $L \leq U$

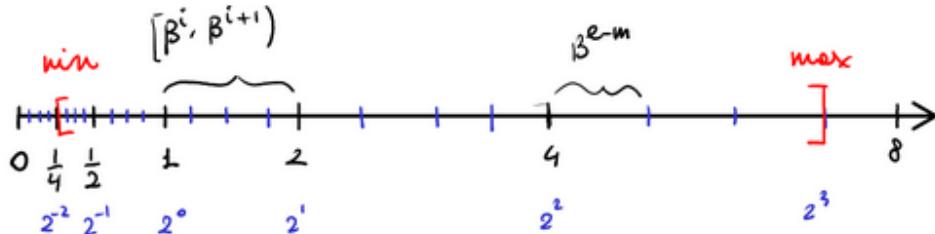
L'insieme dei numeri di macchina $F(\beta, m, L, U)$ è l'insieme degli $x \in \mathbb{R}$ tali che:

- $x = \text{sgn}(x)\beta^e \sum_{j=1}^m \alpha_j \beta^{-j}$,
- $L \leq e \leq U$,
- $\alpha_j \in \{0, \dots, \beta - 1\}$, $\alpha_1 \neq 0$

unito all'insieme $\{0\}$.

I numeri di macchina sono un sottoinsieme dei numeri reali esattamente rappresentabili da un software.

Esempio $\beta = 2$, $m = 3$, $L = -1$, $U = 3$



Tra ogni intervallo ci sono 5 numeri di macchina equispaziati in scala logaritmica. In ogni intervallo $[2^i, 2^{i+1})$ ci sono numeri macchina con $c = i + 1$ in cui varia solo la mantissa.

La distanza tra i punti è $\beta^c \cdot \beta^{-m} = \beta^{c-m}$

Numero minimo e massimo rappresentabile:

- $\beta^L \cdot 1 \cdot \beta^{-1} = \beta^{L-1}$ //minimo rappresentabile
- $\beta^U \cdot (\beta - 1) \sum_{j=1}^m \beta^{-j} = \beta^U (\beta - 1) \left(\frac{1 - \beta^{-m-1}}{1 - \beta^{-1}} - 1 \right)$ (somma parziale)
 $= \beta^U (\beta - 1) \frac{1 - \beta^{-m}}{\beta - 1} = \beta^U (1 - \beta^{-m})$ //massimo rappresentabile

Quanti sono i numeri macchina:

$$2 \cdot (U - L + 1) (\beta^m - \beta^{m-1}) + 1$$

segno	esponente	mantissa	$\{0\}$
-------	-----------	----------	---------

1.1.1. Arrotondamento

$x \in \mathbb{R}, x \neq F(\beta, m, L, U)$

- $|x| > \beta^U(1 - \beta^{-m})$ overflow
- $|x| < \beta^{L-1}$ underflow
- $\beta^{L-1} \leq |x| \leq \beta^U(1 - \beta^{-m})$ arrotondamento

Definizione (troncamento):

$RD : R \rightarrow F(\beta, m, L, U)$

$RD(x)$ è uno dei due numeri macchina adiacenti

- troncamento round-down: $TR(x) = \lfloor x \rfloor$
- troncamento round-up: $UP(x) = \lceil x \rceil$
- troncamento round-to-nearest: $RN(x) = \begin{cases} \lceil x \rceil, & \alpha_{m+1} \geq \beta/2 \\ \lfloor x \rfloor, & \alpha_{m+1} < \beta/2 \end{cases}$

Dove $\lfloor x \rfloor := \text{sgn}(x)\beta^e \sum_{j=1}^m \alpha_j \beta^{-j}$, $\lceil x \rceil := \text{sgn}(x)\beta^e \left(\sum_{j=1}^m \alpha_j \beta^{-j} + \beta^{-m} \right)$

Valgono le seguenti diseguaglianze:

- $|TR(x) - x| \leq \beta^{e-m}$
- $|RN(x) - x| \leq \beta^{e-m} - 2$ ed inoltre $|x - RN(x)| = \min_{z \in F(\beta, m, L, U)} |z - x|$

Quindi, d'ora in poi useremo sempre $RN(x)$.

Definizione (errore assoluto e relativo):

L'errore assoluto dell'arrotondamento $\delta_x := |x - RN(x)| \leq \frac{1}{2}\beta^{e-m} = u \cdot |x|$

L'errore relativo dell'arrotondamento $\epsilon_x := \frac{|x - RN(x)|}{|x|} \leq \frac{1}{2}\beta^{1-m} = u$

Si ricava che $|\delta_x| \leq \frac{1}{2}\beta^{e-m}$, $|\epsilon_x| \leq \frac{\beta^{e-m}}{2\beta^{e-1}} \Rightarrow |\epsilon_x| \leq \frac{1}{2}\beta^{1-m}$ che non dipende da e
minorante del modulo di x

I numeri in virgola mobile garantiscono un errore relativo limitato in modo uniforme, se non si ha over/under flow. Il numero $u = \frac{1}{2}\beta^{1-m}$ si chiama **precisione di macchina**.

Standard IEEE 754:

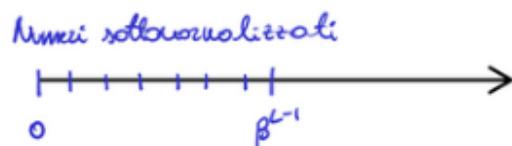
Rappresentazione	memoria	m (mantissa)	esponente	segno	L	U
Half precision	16 bit	11 bit	5 bit	1 bit	-14	15
Single precision	32 bit	24 bit	8 bit	1 bit	-126	127
Double precision	64 bit	53 bit	11 bit	1 bit	-1022	1023
Quadruple precision	128 bit	113 bit	15 bit	1 bit	-16382	16383

1.1.2. Numeri sottonormalizzati

Rappresentazione di numeri $x \in [0, \beta^{L-1}]$ per rendere l'underflow più graduale.

Si usa un valore speciale dell'esponente che implicitamente significa $e = L$, $\alpha_1 = 0$ e si usano le restanti cifre della mantissa. Ad esempio $2^L \cdot 0.001011\dots$

Ottengo valori equispaziati tra $[0, \beta^{L-1}]$ con distanza β^{L-m}



Sui numeri sottonormalizzati l'errore relativo non è limitato da $\frac{\beta^{1-m}}{2}$ e aumenta con l'avvicinarsi allo zero.

2. Errore nel calcolo di funzione:

Le quattro operazioni di macchina sono \oplus , \ominus , \otimes , \oslash dove $x \oplus y = RN(x + y)$ ecc...
In generale non valgono le proprietà associativa e distributiva.

Esempio (proprietà associativa):

$$F(10, 3, -99, 99), \quad x = 0.135 \cdot 10^{-4}, \quad y = 0.258 \cdot 10^{-2}, \quad z = -0.251 \cdot 10^{-2}$$

$$\begin{aligned} x \oplus (y \oplus z) &= 0.135 \cdot 10^{-4} \oplus (0.258 \cdot 10^{-2} \oplus -0.251 \cdot 10^{-2}) \\ &= 0.135 \cdot 10^{-4} \oplus 0.700 \cdot 10^{-4} \\ &= 0.835 \cdot 10^{-4} \end{aligned}$$

$$\begin{aligned} (x \oplus y) \oplus z &= (0.135 \cdot 10^{-4} \oplus 0.258 \cdot 10^{-2}) \oplus -0.251 \cdot 10^{-2} \\ &= 0.259 \cdot 10^{-2} \oplus -0.251 \cdot 10^{-2} \\ &= 0.800 \cdot 10^{-4} \end{aligned}$$

Sia $f: \mathbb{R}^m \rightarrow \mathbb{R}$

Si vuole calcolare $f(p_0)$, $p_0 = (x_1^{(0)}, \dots, x_m^{(0)}) \in \mathbb{R}^m$

Fonti dell'errore:

- f viene approssimata con una funzione \tilde{f} che coinvolge solo operazioni aritmetiche di base.
Ad esempio si approssimano esponenziali con sviluppo di Taylor.

2. \tilde{f} viene "tradotta" in un algoritmo \tilde{f}_a , ovvero una formula che coinvolge $\oplus, \otimes, \ominus, \oslash$

3. p_0 potrebbe non essere un numero di macchina e viene approssimato con $RN(p_0) =: p_1$

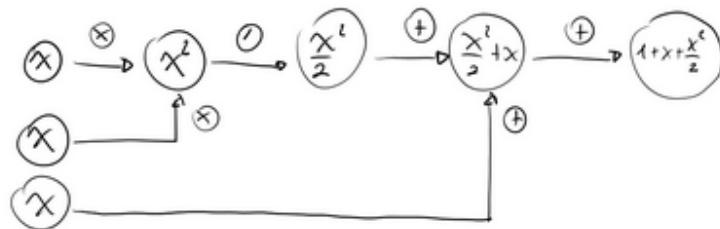
Vogliamo $f(p_0)$, ma in realtà calcoliamo $\tilde{f}_a(p_1)$

Esempio (calcolo di e^π):

- approssimiamo $e^x \approx 1 + x + \frac{x^2}{2} =: \tilde{f}(x)$

- $RN(\pi) = 3.1415 =: p_1$

- calcoliamo $1 \oplus (p_1 \oplus ((p_1 \otimes p_1) \oslash 2)) = \tilde{f}_a(p_1)$



Definizione (errore totale):

Data $f: \mathbb{R}^m \rightarrow \mathbb{R}$, $p_0 \in \mathbb{R}^m$ ed un algoritmo,

l'errore totale è dato da $\delta_f = \tilde{f}_a(p_1) - f(p_0)$ dove $p_1 := RN(p_0)$

2.1. Funzioni razionali:

Per semplificarcici la vita assumiamo che f sia una funzione razionale, ovvero definibile con un numero di operazioni in $\{+, -, \times, /\}$. Quindi $\tilde{f}(x) = f(x)$.

Errore assoluto: $\delta_f = f_a(p_1) - f(p_0) = f_a(p_1) - f(p_1) + f(p_1) - f(p_0) = \delta_a + \delta_d$

$\delta_f = \delta_a + \delta_d$ δ_a errore algoritmico δ_d errore inerente

Errore relativo: $\epsilon_f = \frac{\delta_f}{f(p_0)} = \frac{f_a(p_1) - f(p_0)}{f(p_0)} = \frac{\delta_a}{f(p_0)} + \frac{\delta_d}{f(p_0)} = \frac{f_a(p_1) - f(p_1)}{f(p_0)} + \epsilon_d$

$\epsilon_f = \epsilon_a + \epsilon_d$ ϵ_d errore inerente

$$= \frac{f_a(p_1) - f(p_1)}{f(p_1)} \cdot \frac{f(p_1)}{f(p_0)} + \epsilon_d = \frac{f_a(p_1) - f(p_1)}{f(p_1)} \left(1 + \frac{f(p_1) - f(p_0)}{f(p_0)}\right) + \epsilon_d$$

ϵ_a errore algoritmico

$= \epsilon_a(1 + \epsilon_d) + \epsilon_d = \epsilon_a + \epsilon_a \epsilon_d + \epsilon_d \approx \epsilon_a + \epsilon_d$ (si trascura $\epsilon_a \epsilon_d$ perché di

ordine superiore

In generale per limitare $|\delta_f|$ si cercano disuguaglianze $|\delta_a| < \tau_1$, $|\delta_d| < \tau_2 \Rightarrow |\delta_f| < \tau_1 + \tau_2$

2.1.1. Problemi ben/mal condizionati (come stimare ϵ_d):

Errore assoluto inerente:

$$\delta_d := f(p_1) - f(p_0)$$

Sotto l'ipotesi che $f \in C^1(D)$, $D \subseteq \mathbb{R}^m : p_0 \in D$ si può sfruttare lo sviluppo di Taylor di f centrato in p_0 troncato al prim'ordine.

$$\begin{aligned} f(p_1) - f(p_0) &= f(p_0) + \nabla f(\tilde{p})^T (p_1 - p_0) - f(p_0) \text{ dove } \tilde{p} \text{ vive sul segmento } \overline{p_1 p_0} \\ &\quad (\text{formula di Taylor con resto di Lagrange}) \\ &= \nabla f(\tilde{p})^T (p_1 - p_0) = \sum_{j=1}^m \frac{\partial f}{\partial x_j}(\tilde{p}) \cdot (x_j^{(1)} - x_j^{(0)}) \\ &\approx \sum_{j=1}^m \frac{\partial f}{\partial x_j}(p_0) \cdot (x_j^{(1)} - x_j^{(0)}) \end{aligned}$$

coeffienti di amplificazione δ_j errore sulla componente j di p_0

$$\delta_d \approx \sum_{j=1}^m \frac{\partial f}{\partial x_j}(p_0) \cdot \delta_j$$

Se ho una derivata molto grande in modulo, fare un piccolo errore di arrotondamento nella componente j di p_0 può portare ad un errore molto grande.

Errore relativo inerente:

$$\epsilon_d := \frac{f(p_1) - f(p_0)}{f(p_0)}$$

$$\epsilon_d = \frac{\sum_{j=1}^m \frac{\partial f}{\partial x_j}(p_0) \cdot \delta_j}{f(p_0)} = \sum_{j=1}^m \frac{x_j^{(0)}}{f(p_0)} \frac{\partial f}{\partial x_j}(p_0) \cdot \frac{x_j^{(1)} - x_j^{(0)}}{x_j^{(0)}}$$

coeffienti di amplificazione ϵ_j errore sulla componente j di p_0

$$\epsilon_d \approx \sum_{j=1}^m \epsilon_j \rho_j \text{ dove } \rho_j := \frac{x_j^{(0)}}{f(p_0)} \frac{\partial f}{\partial x_j}(p_0)$$

Problemi in cui si deve calcolare f in cui i coeffienti ρ_j sono grandi in modulo (o ce n'è almeno uno grande) si dicono **mal condizionati**. Viceversa, i coeffienti ρ_j sono vicini a 1, ovvero tali che ϵ_d risulta dello stesso ordine degli errori ϵ_i , $i = 1, \dots, m$, allora il problema del calcolo della funzione si dice **ben condizionato**.

2.1.2. Errori inerenti delle operazioni aritmetiche di base:

$$m = 2, \quad f(x, y) = x \text{ op } y$$

Operatore	errore inerente δ_d	errore inerente relativo ϵ_d
$x \oplus y$	$\delta_x + \delta_y$	$\frac{x}{x+y}\epsilon_x + \frac{y}{x+y}\epsilon_y$ (A)
$x \ominus y$	$\delta_x - \delta_y$	$\frac{x}{x-y}\epsilon_x - \frac{y}{x-y}\epsilon_y$ (B)
$x \otimes y$	$y\delta_x + x\delta_y$	$\epsilon_x + \epsilon_y$ (C)
$x \oslash y$	$\frac{1}{y}\delta_x - \frac{x}{y^2}\delta_y$	$\epsilon_x - \epsilon_y$ (D)

Addizione e sottrazione:

- Errore assoluto: non ci sono problemi perché è la somma (o differenza) degli errori assoluti
- Errore relativo: l'errore relativo può diventare grande se i due termini x, y sono molto vicini in modulo. Questo effetto si chiama **cancellazione numerica**.

Ad esempio: $1.23456789 - 1.23456700 = 0.000000089$ e gli errori hanno molto peso.

Moltiplicazione:

- Errore assoluto: dipende dai valori x, y , ma non ci sono amplificazione drastiche,
- Errore relativo: non ci sono problemi perché è la somma (o differenza) degli errori relativi

2.1.3. Algoritmi stabili/instabili (come stimare ϵ_a):

Abbiamo fissato un algoritmo $f_a(x)$ e vogliamo stimare:

Errore assoluto algoritmico:

$$\delta_a := f_a(p_1) - f(p_1)$$

Errore relativo algoritmico:

$$\epsilon_a := \frac{f_a(p_1) - f(p_1)}{f(p_1)}$$

Assumiamo che sui dati iniziali non ci siano errori di arrotondamento i.e.

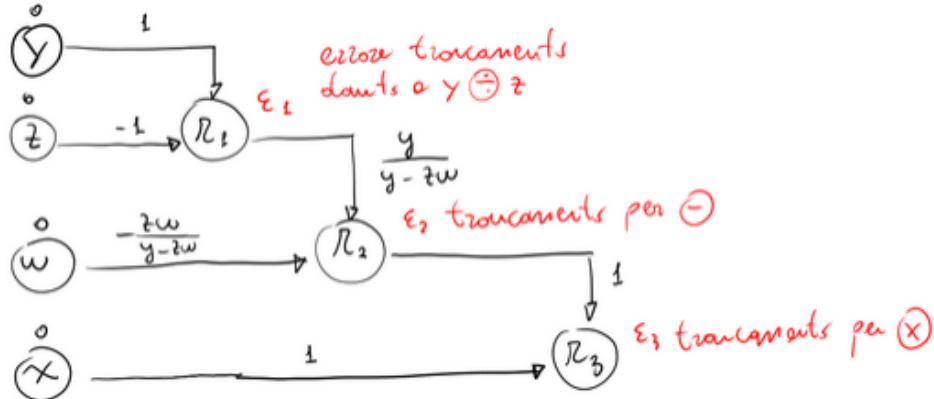
$p_1 = RN(p_0) \in F(\beta, m, L, U)$. Si segue l'errore generato dall'algoritmo sul grafo che lo rappresenta sfruttando le relazioni per l'errore inerente nelle quattro operazioni aritmetiche.

Esempio $f(x, y, z, w) = x \left(\frac{y}{z} - w \right)$, guardiamo l'errore relativo

$$r_1 = \frac{y}{z},$$

$$r_2 = r_1 - w,$$

$$r_3 = r_2 \cdot x$$



$$\begin{aligned}\epsilon_{r_3} &= \epsilon_3 + 1\epsilon_x + 1\epsilon_{r_2} = \epsilon_3 + \epsilon_{r_2} = \epsilon_3 + \epsilon_2 + \left(-\frac{zw}{y-zw} \right) \epsilon_w + \frac{y}{y-zw} \epsilon_{r_1} = \\ &= \epsilon_3 + \epsilon_2 + \frac{y}{y-xw} \epsilon_{r_1} = \epsilon_3 + \epsilon_2 + \frac{y}{y-zw} (\epsilon_1 + 1\epsilon_y - 1\epsilon_0) = \\ &= \epsilon_3 + \epsilon_2 + \frac{y}{y-zw} \epsilon_1 = \epsilon_a\end{aligned}$$

Vale che $|\epsilon_1|, |\epsilon_2|, |\epsilon_3| \leq u$,
nel caso di errori assoluti: $|\delta_1|, |\delta_2|, |\delta_3| \leq u \cdot \max\{|x|, |y|, |z|\}$

Esempio $f(x, y) = x^2 - y^2$

Algoritmo 1:

$$r_1 = x \cdot x$$

$$r_2 = y \cdot y$$

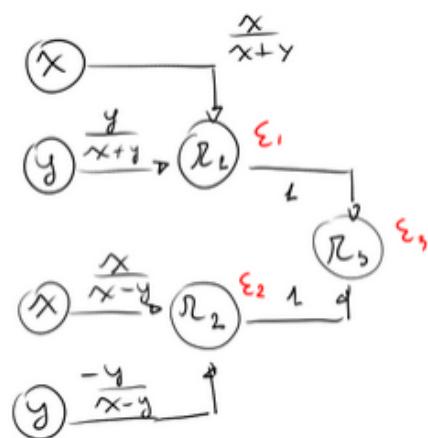
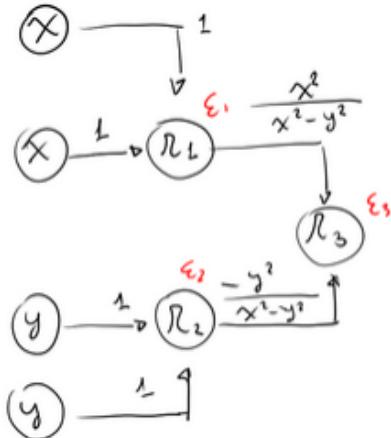
$$r_3 = r_1 - r_2$$

Algoritmo 2:

$$r_1 = x + y$$

$$r_2 = x - y$$

$$r_3 = r_1 \cdot r_2$$



$$\epsilon_a = \epsilon_3 + \frac{x^2}{x^2 - y^2} \epsilon_1 - \frac{y^2}{x^2 - y^2} \epsilon_2$$

$$\epsilon_a = \epsilon_1 + \epsilon_2 + \epsilon_3$$

Qual'è migliore? l'algoritmo 2 perché siamo sempre sicuri che $|\epsilon_a| < 3u$ perché ogni addendo è limitato da u . Mentre per l'algoritmo 1 posso solo dire che $|\epsilon_a| < \left(1 + \frac{|x^2 + y^2|}{|x^2 - y^2|}\right)u$ e non ho

controllo sull'errore. Se per esempio $|x| \approx |y|$ il denominatore tende a zero ed è un casino. Dunque l'algoritmo 1 è instabile.

Se l'algoritmo è tale da produrre errori accettabilmente limitati nella sua applicazione, si dice **stabile**, mentre è detto **instabile** nel caso contrario.

2.1.4. Problema diretto

Abbiamo visto due tecniche per stimare ϵ_a e ϵ_d .

Le possiamo usare per stimare:

$$|\epsilon_f| \leq |\epsilon_d| + |\epsilon_a|, \\ |\delta_f| \leq |\delta_d| + |\delta_a|$$

Problema diretto: Date f , una stima degli errori δ_{x_i} e un algoritmo:
Stimare δ_f per $p_0 \in D \subseteq R^m$

Esempio $f(x_1, x_2) = \frac{x_1}{x_2}, f_a(x_1, x_2) = x_1 \oslash x_2$

$$D = [1, 3] \times [4, 5], p_0 \in D$$

Errore inerente:

Supponendo che $|\delta_{x_i}| < \frac{1}{2} \cdot 10^{-2}, i = 1, 2$ vogliamo dare una maggiorazione a δ_f .

$$\frac{\partial f}{\partial x_1} = \frac{1}{x_2} \Rightarrow A_1 = \sup_{x \in D} \left| \frac{\partial f}{\partial x_1}(x) \right| = \sup_{(x_1, x_2) \in D} \left| \frac{1}{x_2} \right| = \frac{1}{4},$$

$$\frac{\partial f}{\partial x_2} = -\frac{x_1}{x_2^2} \Rightarrow A_2 = \sup_{x \in D} \left| \frac{\partial f}{\partial x_2}(x) \right| = \sup_{(x_1, x_2) \in D} \left| -\frac{x_1}{x_2^2} \right| = \frac{3}{16}$$

$$\text{Dunque } |\delta_f| \leq |\delta_a| + |\delta_d| \leq |\delta_a| + \frac{1}{4} \delta_{x_1} + \frac{3}{16} \delta_{x_2} = |\delta_a| + \frac{1}{4} \cdot \frac{1}{2} \cdot 10^{-2} + \frac{3}{16} \cdot \frac{1}{2} \cdot 10^{-2}$$

Errore algoritmico:

$$\delta_a = \sum_{j=1}^m \frac{\partial f}{\partial x_j}(p_0) \cdot \delta_j \Rightarrow |\delta_a| \leq \sum_{j=1}^m \left| \frac{\partial f}{\partial x_j}(p_0) \right| \cdot |\delta_j| \leq \sum_{j=1}^m \sup_{p \in D} \left| \frac{\partial f}{\partial x_j}(p) \right| \cdot |\delta_j| \\ |\delta_a| = \left| \frac{x_1}{x_2} \right| \cdot \frac{1}{2} \cdot 10^{-2} \text{ dato solo dall'errore di troncamento}$$

Dunque l'errore assoluto:

$$|\delta_f| \leq \left| \frac{x_1}{x_2} \right| \cdot \frac{1}{2} \cdot 10^{-2} + \frac{1}{4} \cdot \frac{1}{2} \cdot 10^{-2} + \frac{3}{16} \cdot \frac{1}{2} \cdot 10^{-2} = \frac{23}{32} \cdot 10^{-2} \\ \leq 1$$

2.1.5. Problema inverso

Problema inverso: dato $\tau > 0$, $f, p_0 \in \mathbb{R}^m$ determinare un algoritmo e un valore di u per avere $|\delta_f| < \tau$

Esempio $f(x_1, x_2) = \frac{x_1}{x_2}$, $D = [1, 2] \times [-2, -1]$

si vuole trovare un algoritmo e u per cui $|\delta_f| < 10^{-2}$

Scegliamo $x_1 \otimes x_2$ algoritmo e stimiamo $|\delta_{x_1}|, |\delta_{x_2}|$

$|\delta_{x_1}| \leq u \cdot |x_i| \leq 2u, i = 1, 2$

$$A_1 = \sup_D \left| \frac{\partial f}{\partial x_1} \right| = 1$$

$$A_2 = \sup_D \left| \frac{\partial f}{\partial x_2} \right| = 2$$

$$|\delta_{x_1}| \quad |\delta_{x_2}|$$

$$|\delta_f| \leq |\delta_a| + 2u + 2u \leq \max_D \left| \frac{x_1}{x_2} \right| \cdot u + 4u = 2u + 4u = 6u$$

$$|\delta_f| \leq 6u \leq 10^{-2} \Rightarrow u = \frac{1}{6} \cdot 10^{-2}$$

2.2. Funzioni non razionali

Ad esempio, $f = e^{\cos(x+y)}$.

Utilizziamo un'approssimazione razionale di $\tilde{f} \approx f$. $(f \rightarrow \tilde{f} \rightarrow \tilde{f}_a \rightarrow \tilde{f}_a(p_1))$

$$\delta_f = \tilde{f}_a(p_1) - f(p_0) = \tilde{f}_a(p_1) - \tilde{f}(p_1) + \tilde{f}(p_1) - f(p_1) + f(p_1) - f(p_0) = \delta_a + \delta_{AN} + \delta_d$$

errore algoritmico δ_a errore analitico δ_{AN} errore inerente δ_d

L'errore analitico dipende da cosa usiamo come approssimazione razionale. Noi useremo Taylor.

Teorema:

$f: \mathbb{R} \rightarrow \mathbb{R}$, $f \in C^{k+1}(\mathbb{R})$ allora dato $x_0 \in \mathbb{R}$ si ha che

$$f(x) = \sum_{j=0}^k f^{(j)}(x_0) \frac{(x-x_0)^j}{j!} + \frac{f^{(k+1)}(\eta)}{(k+1)!} (x-x_0)^{k+1}, \quad \eta \in [x_0, x] \text{ oppure } \eta \in [x, x_0]$$

Se prendo $T_\eta(x) = \sum_{j=0}^k f^{(j)}(x_0) \frac{(x-x_0)^j}{j!}$ questa è una approssimazione razionale di $f(x)$ e,

sapendo qualcosa in più su x, x_0 e su f , posso stimare la grandezza dell'errore analitico:

$T_\eta(x) - f(x)$ per x in un certo insieme.

Esempio:

Approssima $e^x \approx 1 + x + \frac{x^2}{2}$ (Taylod di grado 2 centrato in $x_0 = 0$)

$$R(x) = e^x - \left(1 + x + \frac{x^2}{2}\right) = \frac{e^\eta x^3}{6}$$

Supponiamo $x \in [-1, 1]$, quando è grande l'errore analitico relativo?

$$|\epsilon_{AN}| = \frac{|R(x)|}{|f(x)|} \leq \frac{\max_{x,\eta \in [-1,1]} |e^\eta \cdot x^3|}{\min_{x \in [-1,1]} |e^x|} \cdot \frac{1}{6} = \frac{e \cdot 1^3}{6e^{-1}} = \frac{e^2}{6} \approx 1.23$$

Per un k generico

$$|\epsilon_{AN}| \leq \frac{\max_{x,\eta \in [-1,1]} |e^\eta \cdot x^{k+1}|}{\min_{x \in [-1,1]} |e^x|} \cdot \frac{1}{(k+1)!} \leq \frac{e}{e^{-1}(k+1)!} = \frac{e^2}{(k+1)!}$$

Al passo k ci sono $\mathcal{O}(k)$ operazioni per $k = 0, \dots, n$.

Il costo totale $\mathcal{O}(n^2)$

Richiami di Algebra Lineare

1. Matrici

Sia $f: \mathbb{C}^n \rightarrow \mathbb{C}^m$ una funzione che soddisfa le proprietà di:

- **Additività:** $f(x+y) = f(x) + f(y) \quad \forall x, y \in \mathbb{C}^n$
- **Omogeneità:** $f(\lambda x) = \lambda f(x) \quad \forall x \in \mathbb{C}^n, \lambda \in \mathbb{C}$

Allora f è una trasformazione lineare ed esiste un'unica rappresentazione matriciale
 $\exists! A \in \mathbb{C}^{m \times n}: f(x) = A \cdot x \quad \forall x \in \mathbb{C}^n$

Definizione (matrice trasposta e trasposta coniugata):

Data $A \in \mathbb{C}^{m \times n}$, la matrice $B \in \mathbb{C}^{n \times m}$ i cui elementi sono $b_{ij} = a_{ji}$ si dice matrice trasposta di A e si indica con A^T , la matrice $B \in \mathbb{C}^{n \times m}$ i cui elementi sono $b_{ij} = \bar{a}_{ji}$ si dice matrice trasposta coniugata di A e si indica con A^H

1.1. Operazioni tra matrici

1.1.1. Somma

$A, B \in \mathbb{C}^{m \times n}, C = A + B: c_{ij} = a_{ij} + b_{ij}$, vale la proprietà associativa e commutativa.

1.1.2. Moltiplicazione

$$A \in \mathbb{C}^{m \times k}, B \in \mathbb{C}^{k \times n}, C = AB: C \in \mathbb{C}^{m \times n}, c_{ij} = \sum_{l=1}^k a_{il} b_{lj} \quad \forall i, j$$

Il costo computazionale della moltiplicazione è $\mathcal{O}(m \times n \times k)$, se $m = n = p$ ha costo $\mathcal{O}(n^3)$

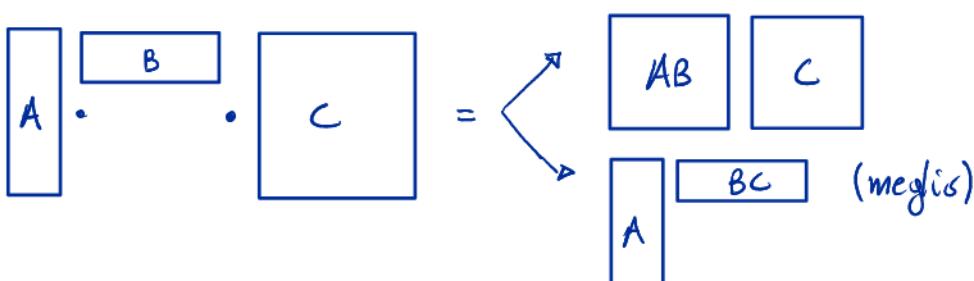
- Non vale la proprietà commutativa $AB \neq BA$
- Vale la proprietà associativa $(AB)C = A(BC)$
- Vale la proprietà distributiva $(A+B)C = AC + BC$
- $AB = 0 \Rightarrow A = 0 \vee B = 0$ ovvero l'anello delle matrici non è un dominio di integrità.

Le proprietà non valgono per i costi computazionali. Se ho matrici con poche righe o poche colonne, mi conviene mantenere questa proprietà più a lungo possibile nei risultati intermedi.

Esempio confronta il costo computazionale di $(AB)C = A(BC)$:

- $AB = R \rightarrow \mathcal{O}(m \cdot n \cdot p), BC = S \rightarrow \mathcal{O}(m \cdot p \cdot q) \Rightarrow (AB)C \rightarrow \mathcal{O}(m \cdot p \cdot (n+q))$
- $BC = R \rightarrow \mathcal{O}(n \cdot p \cdot q), AR = S \rightarrow \mathcal{O}(m \cdot n \cdot q) \Rightarrow A(BC) \rightarrow \mathcal{O}(m \cdot q \cdot (m+q))$

Se n è piccolo $(AB)C \rightarrow \mathcal{O}(n^3), A(BC) \rightarrow \mathcal{O}(n^2)$



Osservazione: in AB si può vedere il risultato riga per riga o colonna per colonna in questi modi:

$$A = \begin{bmatrix} A^1 \\ \dots \\ A^m \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & \dots & B_p \end{bmatrix}$$

$$C = AB = \begin{bmatrix} AB_1 & \dots & AB_p \end{bmatrix} = \begin{bmatrix} A^1B \\ \dots \\ A^mB \end{bmatrix}$$

Le colonne di $C = AB$ sono combinazioni lineari delle colonne di A , le righe di $C = AB$ sono combinazioni lineari delle righe di B .

Esempio moltiplicare per matrici diagonali è equivalente a scalare le righe o le colonne per le entrate diagonali:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & & \\ & 3 & \\ & & 4 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 4 \\ 2 & 3 & 4 \\ 2 & 3 & 4 \end{bmatrix}, \quad \begin{bmatrix} 2 & & \\ & 3 & \\ & & 4 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

Esercizio: trovare i costi computazionali asintotici per il calcolo di $(A + B)C = AC + BC$ dove $B, A \in \mathbb{C}^{m \times n}$, $C \in \mathbb{C}^{n \times p}$

1. $(A + B) = R \rightarrow \mathcal{O}(mn)$ (devo eseguire mn addizioni)
 $RC \rightarrow \mathcal{O}(mnp)$ (mnp moltiplicazioni e $m(n-1)p$ addizioni)
2. $AC \rightarrow \mathcal{O}(mnp)$
 $BC \rightarrow \mathcal{O}(mnp)$

I costi asintotici sono $\mathcal{O}(mn) + \mathcal{O}(mnp)$ e $\mathcal{O}(mnp)$, rispettivamente. Se n è piccolo e m, p grandi, allora $(A + B)C$ ha costo $\mathcal{O}(mn)$ e $AC + BC$ ha costo $\mathcal{O}(mnp)$.

1.2. Determinante

Definizione (determinante con sviluppo di Laplace):

Sia $A^{n \times n}$, si definisce $\det(A) : \mathbb{C}^{n \times n} \rightarrow \mathbb{C}$,

$$\det(A) := \begin{cases} a_{11}, & n = 1 \\ a_{11}a_{22} - a_{12}a_{21}, & n = 2 \\ \sum_{j=1}^n (-1)^{i+j} \det(A_{ij}), & n > 2 \end{cases}$$

dove A_{ij} è la sottomatrice ottenuta eliminando la riga i e la colonna j

Il calcolo del determinante mediante la valutazione dello sviluppo di Laplace ha costo $\mathcal{O}(n!)$

Proprietà del determinante:

- A è invertibile se e solo se $\det A \neq 0$ ($\det A = 0 \implies A$ singolare),
- $\det A = \det A^T$, $\det A = \det A^H$,

- Se A diagonale o triangolare allora $\det A = \prod_{i=1}^n a_{ii}$

Teorema (Binet-Cauchy):

Sia $C = AB$: $A \in \mathbb{C}^{n \times n}$, $B \in \mathbb{C}^{n \times n}$, allora $\det C = \det A \cdot \det B$

In generale, dati $A \in \mathbb{C}^{n \times p}$, $B \in \mathbb{C}^{p \times n}$

$$\det C = \begin{cases} 0, & p < n \\ \sum_j A_{[j]} B_{[j]}, & p \geq n \end{cases}$$

dove $A_{[j]}$, $B_{[j]}$ sono i minori di ordine n relativi alla stessa scelta di indici in A e in B

1.2.1. Sottomatrici

Definizione (sottomatrice):

Una sottomatrice di A è una matrice ottenuta prendendo la restrizione di A a un sottoinsieme di righe e di colonne di A .

$A \in \mathbb{C}^{n \times n}$, $I, J \subseteq \{1, \dots, n\}$: $|I| = n_1$, $|J| = n_2$

$\mathbb{C}^{n_1 \times n_2} \ni A(I, J)$ tale che le entrate sono ottenute "incrociando" le righe corrispondenti a I con le colonne con indice in J .

Una sottomatrice tale che gli insiemi I, J usati per estrarla verificano $I = J$ si dice **sottomatrice principale**. Inoltre, una sottomatrice di ordine k tale che $I = J = \{1, \dots, k\}$ si dice **sottomatrice principale di testa**.

Esempio: $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, $A_{[2][3]} = \begin{bmatrix} 1 & 3 \\ 7 & 9 \end{bmatrix}$, $A_{[2][3]} = \begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$ s.m. principali

$A_1 = \{1\}$, $A_2 = \{1, 2\}$, $A_3 = A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ s.m. principali de testa

Definizione (minore):

Il minore di ordine k , $k \in \{1, \dots, n\}$ è il determinante di una sottomatrice $n \times n$

Definizione (rango):

Il rango di A è il massimo numero di righe o di colonne linearmente indipendenti (ovvero che formano una base di \mathbb{C}^n) i.e. l'ordine massimo dei minori di A diversi da 0

Proprietà del rango:

- $\det A \neq 0$ sse $\text{rk}(A) = n$, $\text{rk}(A) < n$ sse $\det A = 0$
- $\text{rk}(A) = \dim(\mathcal{J}(A))$, $A \in \mathbb{C}^{m \times n}$ dove $\mathcal{J}(A) := \{y \in \mathbb{C}^m : y = Ax, x \in \mathbb{C}^n\}$

1.3. Matrici inverse

Definizione (matrice inversa):

Sia $A \in \mathbb{C}^{n \times n}$: $\det A \neq 0$

$A^{-1} \in \mathbb{C}^{n \times n}$: $AA^{-1} = A^{-1}A = I_n$

$$A^{-1} := B: b_{ij} = (-1)^{i+j} \frac{\det(A_{ii})}{\det(A)}$$

Proprietà delle matrici inverse:

- $\exists A^{-1}$ sse $\det A \neq 0$ i.e. $rk(A) = n$ i.e. A ha n righe o colonne linearmente indipendenti
- Usiamo il teorema di B-C per trovare $\det(A^{-1})$

$$1 = \det(I) = \det(AA^{-1}) = \det A \det A^{-1} \implies \det(A^{-1}) = \frac{1}{\det A}$$

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

Dim: $(B^{-1}A^{-1})AB = B^{-1}(A^{-1}A)B = B^{-1}B = I$ \square

- $(A^T)^{-1} = (A^{-1})^T =: A^{-T}$, $(A^H)^{-1} = (A^{-1})^H =: A^{-H}$

1.4. Matrici particolari

Definizione (matrici simmetriche ed hermitiane):

Sia $A \in \mathbb{R}^{n \times n}$, A si dice

- Simmetrica se $A = A^T$,
- Antisimmetrica se $A = -A^T$,
- Ortogonale se $A^T A = A A^T = I$ i.e. $A^{-1} = A^T$,

Sia $A \in \mathbb{C}^{n \times n}$, A si dice

- Hermitiana se $A = A^H$,
- Antihermitiana se $A = -A^H$,
- Unitaria se $A^H A = A A^H = I$ i.e. $A^{-1} = A^H$,
- Normale se $A^H A = A A^H$.

Oss: $\{A simmetriche\} \subseteq \{A hermitiane\} \subseteq \{A normale\}$
 $\{A ortogonali\} \subseteq \{A unitarie\}$ e insieme hanno inversa "facile"

Definizione (matrici di permutazione):

$A \in \mathbb{R}^{n \times n}$ si dice matrice di permutazione sse si ottiene da I permutando le righe o le colonne

Esempio: consideriamo la permutazione che sposta la prima riga alla seconda posizione, la seconda riga alla terza posizione, la terza riga alla quarta posizione. $\Rightarrow P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

Proprietà delle matrici di permutazione:

- A di permutazione è ortogonale i.e. $P^T P = P P^T = I$
- Dim:** P^T è ottenuto con la permutazione inversa \square
- Il prodotto di A con una matrice di permutazione produce una permutazione delle righe di A (se P è a sx) o delle colonne di A (se P è a dx).

2. Sistemi lineari

Abbiamo un sistema lineare di m equazioni in n incognite

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1m}x_m = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2m}x_m = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nm}x_m = b_n \end{cases} \equiv Ax = b : A \in \mathbb{C}^{m \times n}, x \in \mathbb{C}^m, b \in \mathbb{C}^n$$

Teorema (Rouche-Capelli):

$Ax = b$ ammette almeno una soluzione se $\text{rk}(A) = \text{rk}(A|b)$

Inoltre, se il sistema è sovradeterminato ($m \geq n$):

- Se $\text{rk}(A) = n \Rightarrow \exists!$ soluzione unica
- Se $\text{rk}(A) < n \Rightarrow \exists$ infinite soluzioni e l'insieme delle soluzioni forma un sottospazio vettoriale affine di dimensione $n - \text{rk}(A)$

Esempio

$$\begin{cases} x - 2y - 2z = 0 \\ -2x - y + 4z = 3 \\ x - 2y - 2z = 0 \end{cases} \Rightarrow \begin{cases} x = 2y + 2z \\ -5y = 3 \end{cases} \Rightarrow \begin{cases} x = \frac{-6}{5} + 27 \\ y = \frac{-5}{3} \\ z \end{cases}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ vettore delle incognite, } A = \begin{bmatrix} 1 & -2 & -2 \\ -2 & -1 & 4 \\ 1 & -2 & -2 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$$

$$\text{Le soluzioni sono } \left\{ \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\frac{6}{5} + 2z \\ -\frac{3}{5} \\ z \end{bmatrix}, z \in \mathbb{R} \right\} = \left\{ \begin{bmatrix} -6/5 \\ -3/5 \\ 0 \end{bmatrix} + z \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}, z \in \mathbb{R} \right\} \text{ s.v. affine.}$$

$$= \begin{bmatrix} -6/5 \\ -3/5 \\ 0 \end{bmatrix} + \text{span} \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} \text{ s.v. di dimensione } n - \text{rk}(A) = 3 - 2 = 1$$

Definizione (sistema omogeneo):

Se $b = \begin{bmatrix} 0 \\ \dots \\ 0 \end{bmatrix}$, il sistema si dice omogeneo ed esiste sempre almeno una soluzione $\left(x = \begin{bmatrix} 0 \\ \dots \\ 0 \end{bmatrix} \right)$

Definizione (kernel):

L'insieme delle soluzioni di $Ax = 0$ si chiama kernel. $\ker(A) = \{x \in \mathbb{R}^m : Ax = 0\}$

Si osserva che $\dim(\ker(A)) = m - \text{rk}(A)$

2.1. Metodi di risoluzione

Metodo diretto:

Se $m = n$, $\det A \neq 0 \Rightarrow$ la soluzione di $Ax = b$ è unica, $x = A^{-1}b$ (inefficiente)

Metodo di Cramer:

Un altro modo per esprimere la soluzione è con la regola di Cramer: il vettore soluzione si può scrivere componente per componente come $x_j = \frac{\det(A_j)}{\det A}$, dove A_j matrice ottenuta da A sostituendo la colonna j con b .

Inefficiente poichè ci sono $\mathcal{O}(n)$ determinanti da calcolare.

3. Autovalori e autovettori

Definizione (autovalore e autovettori dx/sx)

Sia $A \in \mathbb{C}^{n \times n}$, $\lambda \in \mathbb{C}$ si dice **autovalore** di A se $\exists v \in \mathbb{C}^n$, $v \neq 0$ tale che $Av = \lambda v$

- v si dice **autovettore destro** per A
- $w \in \mathbb{C}^n$ si dice **autovettore sinistro** per A se rispetta $w^H A = \lambda w^H$

Osservazione: w è un autovettore sinistro per A con autovalore λ se e solo se è autovettore destro per A^H con autovalore $\bar{\lambda}$. In altre parole $w^H A = \lambda w^H \Leftrightarrow A^H w = \bar{\lambda} w$

Inoltre, se $A = A^H$ allora autovettori sinistri e destri coincidono, e gli autovalori sono reali.

3.1. Calcolo di autovalori

```
% Matlab  
[v,d]=eig(A)  
% v matrice con gli autovettori, d matrice diagonale con autovalori
```

$$Av = \lambda v \Leftrightarrow Av - \lambda v = 0 \Leftrightarrow (A - \lambda I)v = 0$$

Per il teorema di Rouche Capelli, esiste soluzione non banale ($v \neq 0$) sse $\det(A - \lambda I) = 0$
 $p_A(\lambda) := \det(A - \lambda I)$ si dice polinomio caratteristico di A .

Teorema (fondamentale dell'algebra):

Un polinomio a coefficienti complessi di grado n ammette esattamente n radici, ciascuna contata con molteplicità.

Da questo deriva che $\det(A - \lambda I) = c_0 + c_1 \lambda + \dots + c_n \lambda^n$ ammette $\lambda_1, \dots, \lambda_n \in \mathbb{C}$ radici

$$\begin{aligned} \text{Osservazione: } p_A(\lambda) &= c_n(\lambda - \lambda_1) \cdots (\lambda - \lambda_n) \\ &= (-1)^n (\lambda - \lambda_1) \cdots (\lambda - \lambda_n) \end{aligned}$$

Sviluppando:

$$= (-1)^n \lambda^n + (-1)^{n-1} \sigma_1 \lambda^{n-1} + \dots + \sigma_{n-1} \lambda + \sigma_n$$

Dove σ_s è la somma dei minori principali di ordine s i.e. le somme dei prodotti degli autovalori presi s alla volta.

Si osserva anche che

- $\sigma_1 = \sum_{i=1}^n \lambda_i =: \text{Tr}(A)$
- $\sigma_n = \det(A) = \prod_{i=1}^n \lambda_i$, quindi $\det(A) = 0 \iff \exists \lambda_i = 0$

Esempio: $A = \begin{bmatrix} 1 & 3 \\ 0 & 4 \end{bmatrix} \rightarrow A - \lambda I = \begin{bmatrix} (1-\lambda) & 3 \\ 0 & (4-\lambda) \end{bmatrix}$

$$P_\lambda(A) = \det(A - \lambda I) = (1-\lambda)(4-\lambda) = \lambda^2 - 5\lambda + 4 \underset{\text{Tr}(A)}{\circledcirc} \det A$$

Dunque ogni matrice $A \in \mathbb{C}^{n \times n}$ ha esattamente n autovalori contati con molteplicità algebrica

3.1.1. Molteplicità algebrica e geometrica

Definizione (molteplicità algebrica):

$\tilde{\lambda} \in \mathbb{C}$ autovalore di $A \in \mathbb{C}^{n \times n}$ ha molteplicità algebrica $\alpha(\tilde{\lambda}) = k$ se ha molteplicità k come radice del polinomio caratteristico.

Si ha che $\sum_{\tilde{\lambda}} \alpha(\tilde{\lambda}) = n$

Secondo la teoria di Galois, non esistono formule generali (basate su operazioni aritmetiche e radicali) per risolvere equazioni polinomiali di grado ≥ 5 . Quindi, per il calcolo degli autovalori si usano metodi numerici iterativi (es. algoritmo QR), poiché non esiste una formula chiusa in generale.

Definizione (molteplicità geometrica):

Sia $\tilde{\lambda} \in \mathbb{C}$ autovalore di $A \in \mathbb{C}^{n \times n}$, si dice molteplicità geometrica la dimensione dell'autospazio associato a λ .

$$\gamma(\lambda) := \dim(\ker(A - \lambda I))$$

Proprietà delle molteplicità:

- per Rouche Capelli, $\gamma(\lambda) = n - \text{rk}(A - \lambda I)$
- $1 \leq \gamma(\lambda) \leq \alpha(\lambda) \leq n$
si osserva che se $\lambda_i \neq \lambda_j \forall i \neq j$ allora $\gamma(\lambda_i) = \alpha(\lambda_i) = 1$ (grazie al cazzo)

3.1.2. Matrici diagonalizzabili

Definizione (matrici simili):

$A, B \in \mathbb{C}^{n \times n}$ si dicono simili se $\exists S \in \mathbb{C}^{n \times n}$ invertibile tale che $B = S^{-1}AS$

Matrici simili hanno gli stessi autovalori con stesse molteplicità algebriche e geometriche. Inoltre, se v è autovettore per A associato a λ allora $S^{-1}v$ è autovettore per B associato a λ

Proposizione. Se $A \sim B$, allora A e B hanno gli stessi autovalori con le stesse molteplicità algebriche e geometriche.

Inoltre, se v è autovettore per A associato a λ , allora $S^{-1}v$ è autovettore per B associato a λ .

Dimostrazione. Supponiamo che $Av = \lambda v$. Allora:

$$B(S^{-1}v) = S^{-1}AS(S^{-1}v) = S^{-1}Av = S^{-1}(\lambda v) = \lambda S^{-1}v$$

quindi $S^{-1}v$ è un autovettore di B associato allo stesso autovalore λ .

Ora, calcoliamo il polinomio caratteristico di B :

$$\begin{aligned}\det(B - \lambda I) &= \det(S^{-1}AS - \lambda I) \\ &= \det(S^{-1}(A - \lambda I)S) \\ &= \det(S^{-1}) \cdot \det(A - \lambda I) \cdot \det(S) \\ &= \frac{1}{\det(S)} \cdot \det(S) \cdot \det(A - \lambda I) \\ &= \det(A - \lambda I)\end{aligned}$$

Quindi A e B hanno lo stesso polinomio caratteristico e dunque gli stessi autovalori (con stesse molteplicità algebriche).

Per la molteplicità geometrica, notiamo che:

$$(B - \lambda I)x = 0 \Leftrightarrow S^{-1}(A - \lambda I)Sx = 0 \Leftrightarrow (A - \lambda I)(Sx) = 0$$

quindi la dimensione del nucleo di $B - \lambda I$ è uguale a quella di $A - \lambda I$, cioè le molteplicità geometriche coincidono. □

Definizione. Sia $A \in \mathbb{C}^{n \times n}$. Diciamo che A è **diagonalizzabile** se esiste S invertibile tale che

$$S^{-1}AS = \text{diag}(\lambda_1, \dots, \lambda_n).$$

Osservazione. Se A è diagonalizzabile, allora $S^{-1}AS$ ha gli autovalori sulla diagonale:

$$S^{-1}AS = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \Rightarrow AS = S \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

$$\Rightarrow [As_1 \ \dots \ As_n] = [\lambda_1 s_1 \ \dots \ \lambda_n s_n] \Rightarrow As_j = \lambda_j s_j$$

Quindi le colonne di S sono autovettori di A .

Allora $AS = S \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$, quindi A è diagonalizzabile se e solo se posso scegliere n autovettori linearmente indipendenti, e posso trovare una base di \mathbb{C}^n composta da autovettori di A .

Teorema. Sia $A \in \mathbb{C}^{n \times n}$. Allora A è diagonalizzabile se e solo se $\alpha(\lambda) = \gamma(\lambda)$ per ogni autovalore λ di A .

Esempio: $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $\det(\lambda - \lambda I) = (1-\lambda)^3 \Rightarrow 1 \text{ è uno autovalore di } A$
con $\delta(1) = 3$

$$A - I = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \ \text{rk}(A - I) = 3, \ \delta(1) = n - \text{rk}(A - I) = 3 - 3 = 0$$

Questa matrice non è diagonalizzabile con una trasformazione per similitudine

Oss: se $\lambda_i = \lambda_j \forall i \neq j, i, j \in 1, \dots, n$ allora diagonalizzabile ($d(\lambda_i) = d(\lambda_j) = 1$)

Oss: A diagonalizzabile, $x \in \mathbb{C}^n \Rightarrow \exists c_1, \dots, c_n \in \mathbb{C} : x = \sum_{i=1}^n c_i v_i$, v_i autovalori
anche posso trovare una base di \mathbb{C}^n con autovalori di A.

In un caso particolare le basi di autovalori può essere scelte unitarie/ortogonale

Th spettrale: $A = A^H$ ($A \in \mathbb{R}^{n \times n}$, $A = A^T$) allora

1. A diagonalizzabile
2. gli autovalori sono reali
3. si può scegliere una base di autovalori unitaria (ortogonale)
i.e. posso scegliere autovalori $v_i : \langle v_i, v_i \rangle = \sum_{j=1}^n v_{ij}^2 = 1$
i.e. posso trovare $V \in \mathbb{C}^{n \times n} : V^H A V = [x_1 \dots x_n]$, $V^H V = W^H = I$

Def: una matrice Hermitiana si dice DEFINITA POSITIVA se $\lambda_i > 0 \forall i$

3.2. Proprietà di autovalori e autovettori

1. (λ, v) autocoppia per $A \in \mathbb{C}^{n \times n}$, $\delta \in \mathbb{C}$

allora $A + \delta I$ ha come autocoppie $(\lambda + \delta, v)$

dim: $(A + \delta I)v = Av + \delta v = \lambda v + \delta v = (\lambda + \delta)v$

questo è il
caso $g(x) = x + \delta$

2. En gen. supponiamo $g(x) = \sum_{\delta=0}^d g_\delta x^\delta$, posso definire $g(A) := \sum_{\delta=0}^d g_\delta A^\delta \in \mathbb{C}^{n \times n}$

(λ, v) autocoppie per $A \Leftrightarrow (g(\lambda), v)$ autocoppie per $g(A)$

3. (λ, v) autocoppie per A, A invertibile $\Leftrightarrow (\lambda^{-1}, v)$ autocoppie per A^{-1}

dim: $Av = \lambda v \Rightarrow A^{-1}Av = \lambda A^{-1}v \Rightarrow v = \lambda A^{-1}v \Rightarrow \frac{1}{\lambda}v = A^{-1}v$

4. Se $A \in \mathbb{R}^{n \times n}$ allora se λ è autovalore di A
i.e. gli autovalori complessi di A e entrambi reali sono complessi coniugati

$$\begin{aligned} \text{dim: } \overline{\lambda} &= \overline{P_A(\lambda)} = \overline{(-1)^n \lambda^n + (-1)^{n-1} \operatorname{tr} A \lambda^{n-1} + \dots + \operatorname{tr} A \lambda + \operatorname{tr} A} \\ &= (-1)^n \overline{\lambda}^n + (-1)^{n-1} \overline{\operatorname{tr} A} \lambda^{n-1} + \dots + \overline{\operatorname{tr} A} \lambda + \overline{\operatorname{tr} A} \end{aligned}$$

Def: $\rho(A) := \max_{i=1, \dots, n} |\lambda_i|$ RAGGIO SPEGTRALE di A

Th. $\lim_{n \rightarrow \infty} A^n = 0 \Leftrightarrow \rho(A) < 1$

4. NORME

Si vuole definire le grandezze di un vettore. Formalmente si considerano funzioni con proprietà specifiche, chiamate norme.

Def: $\|\cdot\|: \mathbb{C}^n \rightarrow \mathbb{R}^+$ è detta **NORMA VETTORIALE** se verifica:

$$(i) \|x\| = 0 \Leftrightarrow x = \vec{0}$$

$$(ii) \|\alpha x\| = \|\alpha\| \cdot \|x\| \quad \forall \alpha \in \mathbb{C}, x \in \mathbb{C}^n$$

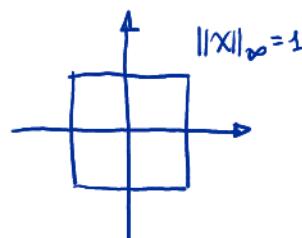
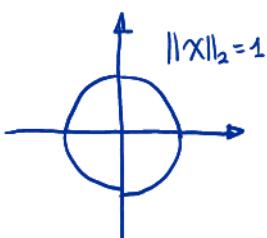
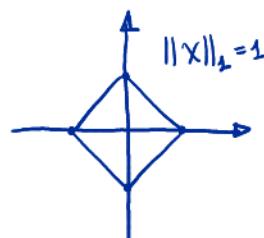
$$(iii) \|x+y\| \leq \|x\| + \|y\| \quad \forall x, y \in \mathbb{C}^n \text{ disegualanza triangolare}$$

Esempi: $\|x\|_1 := \sum_{i=1}^n |x_i|$ (se $n=1$, $\|x\|_1 = |x|$)

$$\|x\|_2 := \sqrt{\sum_{i=1}^n |x_i|^2}, \quad \|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, \quad \|x\|_\infty = \max_{i=1, \dots, n} |x_i|$$

Se def non "diverse" perché l'inviano di vettori di norme \neq , cambia geometria del vettore delle norme che scelgo

Esempio \mathbb{R}^2 :



Th (NORME EQUIVALENTE): date $\|\cdot\|_{n_1}, \|\cdot\|_{n_2}$

$$\exists \alpha, \beta \in \mathbb{R}^+: \alpha \|x\|_{n_1} \leq \|x\|_{n_2} \leq \beta \|x\|_{n_1} \quad \forall x \in \mathbb{C}^n \quad (*)$$

Oss: se vale (*) allora $\beta^{-1} \|x\|_{n_2} \leq \|x\|_{n_1} \leq \alpha^{-1} \|x\|_{n_2}$

Se si considerano spazi vettoriali a dimensione infinita cl th. non vale più

Per le norme più comuni ($\|x\|_1$, $\|x\|_2$, $\|x\|_\infty$) valgono:

- $\|x\|_2 \leq \|x\|_1 \leq \sqrt{n} \|x\|_2$ *in dim con Cauchy-Schwarz: $\langle x_1, x_2 \rangle \leq \sqrt{\|x_1\|_2 \|x_2\|_2}$*
- $\|x\|_\infty \leq \|x\|_1 \leq n \|x\|_\infty$
- $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n} \|x\|_\infty$ *in dim con def*

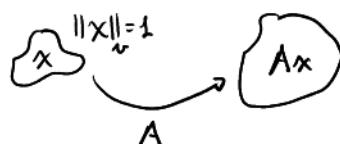
Def: $\|\cdot\|: \mathbb{C}^{m \times n} \rightarrow \mathbb{R}^+$ si dice **NORMA MATRICIALE** se verifica:

- (i) $\|A\| = 0 \Leftrightarrow A = [0]$
- (ii) $\|\alpha A\| = |\alpha| \cdot \|A\| \quad \forall \alpha \in \mathbb{C}, A \in \mathbb{C}^{m \times n}$
- (iii) $\|A + B\| \leq \|A\| + \|B\| \quad \forall A, B \in \mathbb{C}^{m \times n}$ *subadditività rispetto alle norme*
- (iv) se $m = n$: $\|A \cdot B\| \leq \|A\| \cdot \|B\| \quad \forall A, B \in \mathbb{C}^{n \times n}$ *submoltiplicatività*

Dfn: una norma matriciale $\|\cdot\|_M$ si dice **INDOTTA** (o naturale) se

$$\exists \|\cdot\|_v: \|A\|_M = \sup_{\|x\|_v \neq 0} \frac{\|Ax\|_v}{\|x\|_v} = \sup_{\|x\|_v=1} \|Ax\|_v$$

Una norma è tanto grande quanto "allarga" i vettori di norma 1:



D'ORA IN POCO CONSIDERIAMO SOLO MATRICI QUADRATE.

Oss: le norme matriciali indotte da $\|x\|_1$, $\|x\|_2$, $\|x\|_\infty$ sono:

- $\|A\|_1 = \max \sum_{i=1}^n |a_{i1}|$ *max norme delle colonne di A*
- $\|A\|_\infty = \max \sum_{i=1}^n |a_{i1}|$ *max norme delle righe di A*

$$\bullet \|A\|_2 = \sqrt{\rho(A^H A)} \leftarrow \text{DIFFICILE DA CALCOLARE}$$

raggio spettrale

Esempio (NORMA DI FROBENIUS): $\|A\|_F = \sqrt{\sum_{i,s=1}^n |a_{is}|^2}$ norma non indotta

ci si accorge che non è indotta perché $\|I\|_F = \left\| \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \right\| = \sqrt{n}$ e per ogni norma indotta:

$$\|I\| = \sup_{\|x\| \neq 0} \frac{\|Ix\|}{\|x\|} = \sup_{\|x\| \neq 0} \frac{\|x\|}{\|x\|} = 1 \neq \sqrt{n} \quad \forall n > 1$$

Oss: $\|I\|_M \neq 1 \Rightarrow \|\cdot\|_M$ non è indotta
 $\|I\|_M = 1 \Rightarrow \|\cdot\|_M$ indotta, non è detta

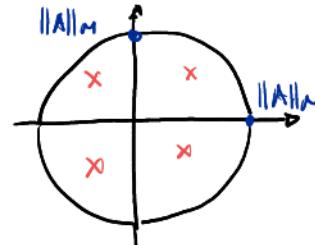
Def: $\|\cdot\|_M$ si dice **COMPATIBILE** con $\|\cdot\|_v$ se vale $\|Ax\|_v \leq \|A\|_M \cdot \|x\|_v \quad \forall A, x$

Oss: le norme matriciali indotte sono compatibili con le norme vettoriali che le inducono: $\frac{\|Ax\|_v}{\|x\|_v} \leq \|A\|_M \Rightarrow \|Ax\|_v \leq \|A\|_M \cdot \|x\|_v$ per costruzione

Oss: le dimostrazioni precedenti e le proprietà delle norme ci vede che gli autovalori non in modulo più piccoli di $\|A\|_M$ se $\|\cdot\|_M$ è compatibile con una norma vettoriale. Infatti, se (x, λ) autocoppie per A i.e. $Ax = \lambda x, x \neq 0$

$$\Rightarrow \underbrace{\|\lambda x\|_v}_{\|\lambda\| \cdot \|x\|_v} = \|Ax\|_v \leq \|A\|_M \cdot \|x\|_v \Rightarrow |\lambda| \leq \|A\|_M$$

tutti gli autovalori di A
appartengono a $\{z \in \mathbb{C} : |z| \leq \|A\|_M\}$



Th (HIRSCH): se $A \in \mathbb{C}^{n \times n}$ allora $\forall \|\cdot\|$ norma matriciale vale $\delta(A) \leq \|A\|$

¹
simplificazione del risultato precedente (non c'entra la compatibilità)

Dimm: (λ, x) autocoppie per A , si considera $B = [x \mid \overset{\circ}{0} \mid \cdots \mid \overset{\circ}{0}]$ per cui vale

$$A \cdot B = \begin{bmatrix} Ax & A_0 & \dots & A_0 \end{bmatrix} = \begin{bmatrix} \lambda x & \overset{\circ}{0} & \cdots & \overset{\circ}{0} \end{bmatrix} = \lambda \begin{bmatrix} x & \overset{\circ}{0} & \cdots & \overset{\circ}{0} \end{bmatrix} = \lambda B$$

$$\frac{|\lambda| \cdot \|B\|}{\|B\|} = \frac{\| \lambda B \|}{\|B\|} = \frac{\|A \cdot B\|}{\|B\|} \leq \frac{\|A\| \cdot \|B\|}{\|B\|} \Rightarrow |\lambda| \leq \|A\| \text{ e in particolare } \delta(A) \leq \|A\| \quad \square$$

Esempio: la norma di A potrebbe essere arbitrariamente più grande del modulo degli autovalori. Un esempio

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \text{ i.e. } \begin{cases} a_{ij} = 1, & i < j \\ a_{ij} = 0, & i \geq j \end{cases} \quad \forall x_i \in \sigma(A), x_i = 0, \quad \|A\|_1 = \|A\|_\infty = n-1$$

5. TEOREMI DI GERSHGORIN

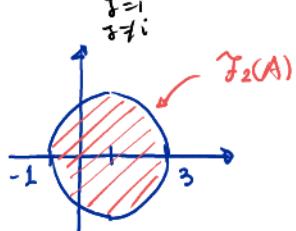
(RAFFINAZIONE DEL TH. HIRSCH)

Def: dato $A \in \mathbb{C}^{n \times n}$ definiamo i CERCHI DI GERSHGORIN associati a λ_i :

$$\mathcal{F}_i(A) := \{z \in \mathbb{C} : |z - \alpha_{ii}| \leq s_i\}, \quad s_i = \sum_{j=1, j \neq i}^n |\alpha_{ij}| \geq 0$$

Esempio: $A = \begin{bmatrix} * & * & * \\ -1 & 1 & i \\ * & * & * \end{bmatrix}$ $\mathcal{F}_2(A) = ?$

$$\alpha_{22} = 1, \quad s_2 = |-1| + |i| = 2$$



5.1 Primo teorema di Gershgorin

Th. (I° GERSHGORIN): $\lambda \in \mathbb{C}$ autovалore $\Rightarrow \lambda \in \bigcup_{i=1}^n \mathcal{F}_i(A)$

Dim: (λ, x) autocoppie per A , indichiamo con K l'indice dello componente di modulo massimo in x

$x \neq 0$ perche' autovettore, $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad |x_k| \geq |x_j| \quad \forall j = 1, \dots, n, \quad |x_k| > 0$

Suggeriamo che $Ax = \lambda x$, considerare le righe K (o l'equazione K) in questa uguaglianza: $a_{K1}x_1 + a_{K2}x_2 + \dots + a_{Kn}x_n = \lambda x_K$ I.e. $\sum_{j=1}^n a_{Kj}x_j = \lambda x_K$

tutti K -esimi elementi

$$\Leftrightarrow \sum_{\substack{j=1 \\ j \neq K}}^n a_{Kj}x_j = \lambda x_K - a_{KK}x_K = (\lambda - a_{KK})x_K$$

$$\Rightarrow |\lambda - a_{KK}| \cdot |x_K| = \left| \sum_{\substack{j=1 \\ j \neq K}}^n a_{Kj}x_j \right| \leq \sum_{\substack{j=1 \\ j \neq K}}^n |a_{Kj}| \cdot |x_j|$$

Dato che $|x_K| > 0$ posso dividere per $|x_K|$ e λx e s_K

$$\Rightarrow |\lambda - a_{KK}| \leq \sum_{\substack{j=1 \\ j \neq K}}^n |a_{Kj}| \frac{|x_j|}{|x_K|} \leq 1 \leq \sum_{\substack{j=1 \\ j \neq K}}^n |a_{Kj}| =: s_K$$

perche' non nullo

$|\lambda - a_{KK}| \leq s_K \Rightarrow \lambda \in \mathcal{F}_K(A) \Rightarrow \lambda \in \bigcup_{i=1}^n \mathcal{F}_i(A)$ dato che l'argomento utilizzato vale per un qualsiasi autovалore \square

Oss: durante la dimostrazione si nota che $\lambda \in \mathcal{F}_K(A)$: K indica dell'elemento di modulo max dell'autovалore; tuttavia, si noti non corrisponde a λ , se x quindi non corrisponde a K .

Def: una matrice si dice a **PREDOMINANZA DIAGONALE FORTE** se le gli elementi più grandi sulla diagonale v.c. $|a_{ii}| > \sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}| = g_i \quad \forall i = 1, \dots, n$

Corollario del I° th. G.: se A è a predominanza diagonale forte, allora A è non singolare ($\det A \neq 0$, $\exists A^{-1}$ etc ...)

Dim: tutti i cerchi di G. di A non comprendono l'origine v.e. $0 \notin \bigcup_{i=1}^n \Xi_i(A)$
 $\Rightarrow 0$ non è autovalore di A . \square

Matlab: calcolo dei raggi g_i :

```
for i=1:n
    for j=1:n
        if i ~= j
            R(i) = R(i) + abs(A(i,j));
        end
    end
end
```

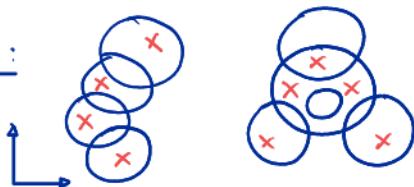
```
n=size(A,1)
c=diag(A)
r=abs(A-diag(c)) * ones(n,1)
figure
hold on
for j=1:n
    circle(c(j),r(j), 'b');
end
```

più efficiente:

5.2 Secondo teorema di Gershgorin

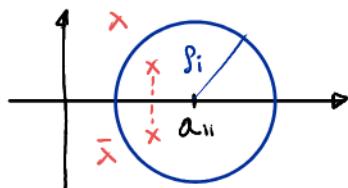
Th (II° GERSHGORIN): se M_1 unione di s cerchi di G., M_2 unione di $s-n$ cerchi e se vale $M_1 \cap M_2 = \emptyset$ allora M_1 contiene esattamente s autovalori e M_2 contiene esattamente gli altri $n-s$

Esempio:



Oss: il I° th. G. non ci dice che c'è sicuramente un autovalore per ogni $\Xi_i(A)$, cioè potrebbe essere che $\Xi_i(A)$ non contiene nessun autovalore.
Il II° th. G. ci dice che se ho un cerchio isolato allora contiene per forza un autovalore. Di conseguenza, se A ha n cerchi disgiunti allora A ha n autovalori distinti ovvero A è diagonalizzabile

Oss: se $A \in \mathbb{R}^{n \times n}$ allora se λ autovalore di $A \Rightarrow \bar{\lambda}$ autovalore di A .
Se ho un cerchio di G. che contiene $\lambda = \alpha + i\omega$, allora contiene $\bar{\lambda} = \alpha - i\omega$



\Rightarrow se ho un cerchio isolato allora contiene necessariamente un autovalore reale

Oss (GERSH. PER A^T): Notiamo $\sigma(A) = \sigma(A^T)$ si può prendere come regione dove stanno gli autovalori l'intersezione $\left\{ \bigcup_{i=1}^n \mathcal{E}_i(A) \right\} \cap \left\{ \bigcup_{i=1}^n \mathcal{E}_i(A^T) \right\}$

N.B. $\left\{ \bigcup_{i=1}^n \mathcal{E}_i(A) \right\} \cap \left\{ \bigcup_{i=1}^n \mathcal{E}_i(A^T) \right\} \neq \bigcup_{i=1}^n (\mathcal{E}_i(A) \cap \mathcal{E}_i(A^T)) \leftarrow$ più piccolo

$\mathcal{E}_i(A^T) = \left\{ z \in \mathbb{C} : |z - \alpha_{ii}| < \sum_{\substack{j=1 \\ j \neq i}}^n |\alpha_{ij}| \right\}$ è un cerchio con stesso centro di $\mathcal{E}_i(A)$, ma raggio calcolato nelle colonne

5.3 Matrici irriducibili e terzo teorema di Gershgorin

Def: $A \in \mathbb{C}^{n \times n}$, si dice **GRAFO ASSOCIAZO** di A . $G(A) = (V, E)$, $V = \{1, \dots, n\}$
 E è l'insieme definito da $(i, j) \in E \Leftrightarrow \alpha_{ij} \neq 0$

Def: $G(A) = (V, E)$ **FORTEMENTE CONNESSO** se $\forall i, s \in \{1, \dots, n\}$ esiste un cammino orientato da i a s .

Def: $A \in \mathbb{C}^{n \times n}$ **IRRIDUCIBILE** se $G(A)$ è fortemente连通的

Th (III° GERSHGORIN): $A \in \mathbb{C}^{n \times n}$ irriducibile allora

se ho un autovalore $\lambda \in \partial \left(\bigcup_{i=1}^n \mathcal{E}_i(A) \right) \Rightarrow \lambda \in \bigcap_{i=1}^n \partial(\mathcal{E}_i(A))$

$$\text{d.e. } \exists K \in \{1, \dots, n\} : |\lambda - \alpha_{KK}| = \sum_{\substack{j=1 \\ j \neq K}}^n |\alpha_{Kj}|$$

$$\text{d.e. } \forall K \text{ vale } |\lambda - \alpha_{KK}| = \sum_{\substack{j=1 \\ j \neq K}}^n |\alpha_{Kj}|$$

In pochi pareri se λ sta sul bordo di un cerchio deve stare sul bordo di tutti i cerchi.

Esempio:

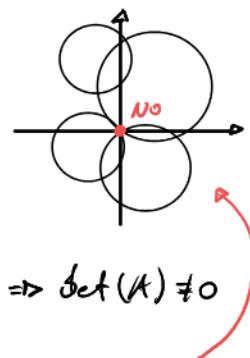


$n=3$ gestire è impossibile



può succedere

Def: $A \in \mathbb{C}^{n \times n}$ e PREDOMINANZA DIAGONALE DEBOLE se
 $|a_{ii}| \geq \sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}|$ e $\exists k \in \{1, \dots, n\} : |a_{kk}| > \sum_{\substack{i=1 \\ i \neq k}}^n |a_{ki}|$



Corollario III° G: A e predominanza diag debole e irriducibile $\Rightarrow \det(A) \neq 0$

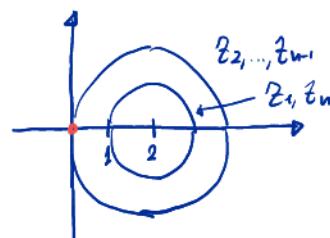
Dim: il vettore θ appartenente ai cerchi per cui vale $|a_{ii}| = \sum_{i \neq j} |a_{ij}|$, ma esiste almeno un cerchio che non contiene θ (quello t.c. $|a_{kk}| > \sum_{i \neq k} |a_{ki}|$)
 $\Rightarrow \theta$ non può essere autovalore per III° Geršgorin \square

Esempio:

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad A : \begin{cases} a_{ii} \neq 0 \\ a_{ii}, i \neq 0 \\ 0 \text{ altrimenti} \end{cases}$$



$G(A)$ fortemente connesso $\Rightarrow A$ irriducibile. I cerchi hanno centro 2 ed hanno raggi 2 ($\text{per } z_2(X), \dots, z_{n-1}(X)$) oppure raggi 1 ($\text{per } z_1(X), z_n(X)$)



θ non può essere autovalore per a. III

6. MATRICI RIDUCIBILI

6.1. Matrici a blocchi

Si può scrivere una matrice con i suoi sottoblocki purché questi abbiano dimensioni compatibili.

$$A = \begin{bmatrix} A_{11} & \dots & A_{1t} \\ \vdots & \ddots & \vdots \\ A_{t1} & \dots & A_{tt} \end{bmatrix} \text{ dove } A_{11} \in \mathbb{C}^{m_1 \times m_1}, \sum_{i=1}^t m_i = m, \sum_{j=1}^t n_j = n$$

Def: $A \in \mathbb{C}^{n \times n}$ TRIANGOLARE A BLOCCHI se $A = \begin{bmatrix} A_{11} & \dots & 0 \\ A_{21} & \dots & A_{2n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \dots & A_{nn} \end{bmatrix}$ opp. $A = \begin{bmatrix} A_{11} & \dots & A_{1n} \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & A_{nn} \end{bmatrix}$

$A \in \mathbb{C}^{n \times n}$ DIAGONALE A BLOCCHI se $A = \begin{bmatrix} A_{11} & 0 \\ 0 & \dots & A_{nn} \end{bmatrix}$

6.2. Sistemi lineari con matrici riducibili

Def: $A \in \mathbb{C}^{n \times n}$, $n \geq 2$ si dice RIDUCIBILE se $\exists \Pi$ matrice di permutazione f.c.

$$\Pi A \Pi^T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad A_{11} \in \mathbb{C}^{K \times K}, \quad A_{22} \in \mathbb{C}^{(n-K) \times (n-K)}, \quad K \in \{1, \dots, n-1\}$$

Oss: possono esserci più Π che portano A in forma triangolare e blocchi a diversi valori di K .

Oss: se $\exists \Pi_1$: $\Pi_1 A \Pi_1^T$ triangolare e blocchi superiore allora $\exists \Pi_2$:
 $\Pi_2 A \Pi_2^T$ triangolare e blocchi inferiore

SISTEMI LINEARI

$Ax = b$, $A \in \mathbb{C}^{n \times n}$ riducibile, considera Π : $\Pi A \Pi^T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$ allora comincia procedere come segue:

$$Ax = b \Leftrightarrow \underbrace{\Pi A}_{\substack{1 \\ \text{deve far sì che}} \text{ } \Pi^T \cdot \Pi = I} x = \underbrace{\Pi b}_{\substack{B \\ \text{le cambia l'ordine delle variabili}}} \Leftrightarrow \underbrace{B y = c}_{\substack{2 \\ \text{le scalista le equazioni}}} \Leftrightarrow \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

$c_1, y_1 \in \mathbb{C}^K$
 $c_2, y_2 \in \mathbb{C}^{(n-K)}$

Una volta risolto $By = c$, trova x mediante $\Pi x = y \Rightarrow x = \Pi^T y$

$$\begin{cases} A_{11}y_1 + A_{12}y_2 = c_1 \\ A_{22}y_2 = c_2 \end{cases} \leftarrow \text{trova } y_2, \text{ sistema } (m-K) \times (n-K)$$

Trovato y_2 lo sostituisce sopra e vedi a sinistra $A_{11}y_1 = c_1 - A_{12}y_2$ \Rightarrow $\text{caso } K \times K$

Ma perché? Eliminazione Gaussiana $\rightarrow \Theta(n^3)$

$$\text{One} \rightarrow \Theta(K^3 + (m-K)^3 + K(m-K)^3) = \Theta(K^3 + (m-K)^3)$$

Se K (i.e. il blocco di zeri) è piccolo il guadagno non è tanto. Se $K = \frac{m}{2}$ il costo $\Theta((\frac{m}{2})^3 + (m - \frac{m}{2})^3) = \Theta(2(\frac{m}{2})^3) = \Theta(\frac{m^3}{4})$ ovvero guadagno un fattore 4 sul tempo di esecuzione.

Oss: se i blocchi A_{11}, A_{12} non sono a loro volta riducibili, il procedimento si può chiamare ricorsivamente quando si risolvono sistemi con A_{11} e A_{22}

6.3 Teorema di riducibilità di una matrice

Th: $A \in \mathbb{C}^{n \times n}$ riducibile $\Leftrightarrow G(A)$ non è fortemente connesso.

Dim: $\begin{matrix} 1 & \xrightarrow{n} & i_1 \\ \vdots & \vdots & \vdots \\ n & \xrightarrow{n} & i_n \end{matrix} \quad \Pi \in \mathbb{R}^{n \times n}$ permutante Π per zifre.

Se considero $\Pi A \Pi^T = \begin{bmatrix} & & \\ & \ddots & \\ & & i_m \end{bmatrix}$, il grafo associato a $\Pi A \Pi^T$ è ugualmente quello associato ad A , ma solo cambiato nome ai vertici.

$G(A)$ fortemente connesso $\Leftrightarrow G(\Pi A \Pi^T)$ fortemente connesso

$\Rightarrow) A$ riducibile $\Rightarrow \exists \Pi: B = \Pi A \Pi^T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, A_{11} \in \mathbb{C}^{k \times k}, A_{22} \in \mathbb{C}^{(n-k) \times (n-k)}$

$G(B)$ non è fortemente connesso perché da $(k+1, \dots, n)$ non posso raggiungere dei vertici $\{1, \dots, k\}$

$\Leftarrow) G(A)$ non è fortemente connesso i.e. $\exists (i, j)$: partendo da i non posso raggiungere j . Partitiono $\{1, \dots, n\}$ in $P = \{\text{vertici raggiungibili da } i\}, Q = P^c$.

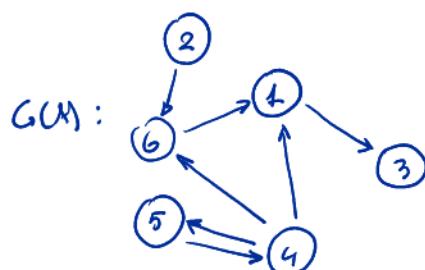
Non può esserci un arco (i, j) : $i \in P, j \in Q$, altrimenti avrei violato la def che i vertici in Q non sono raggiungibili da i .

Considero Π permutazione che mette in testa tutti gli indici in Q e in coda gli indici di P e ottengo: $\Pi A \Pi^T = \begin{array}{c|c} Q & \begin{bmatrix} * & * \\ 0 & * \end{bmatrix} \\ \hline P & \end{array}$ i.e. A riducibile \square

Oss: $k = |Q|, n - k = P$. La dimostrazione si aggiunge come trovare Π .

Esempio (come trovare Π):

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 3 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 2 & -1 & -2 \\ 0 & 0 & 0 & -1 & 2 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{6 \times 6}$$



Vertici di partenza	Vertici raggiungibili	vertici non raggiungibili
1	1 3	2 4 5 6
2	1 3 6 2	4 5
3	3	1 2 4 5 6
4	1 3 4 5 6	2
5	1 3 4 5 6	2
6	1 3 6	2 4 5

$$P_6 = \{1, 3, 6\} \quad U_6 = \{2, 4, 5\}$$

$$\Pi = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} 2 \\ 4 \\ 5 \\ 1 \\ 3 \\ 6 \end{matrix}$$

$$\Pi A \Pi^T = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 3 \\ 0 & 2 & -1 & 3 & 0 & -2 \\ 0 & -1 & 2 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 2 \\ 4 \\ 5 \\ 1 \\ 3 \\ 6 \end{matrix}$$

Se vogliamo risolvere $Ax = b \rightarrow \Pi A x = \Pi b$
 $\rightsquigarrow \Pi A \Pi^T \underbrace{\Pi x}_y = c \quad \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$

$$A_{22} y_2 = c_2 \Rightarrow \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_6 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_3 \\ b_6 \end{bmatrix} \rightsquigarrow (x_1, x_3, x_6)$$

$$A_{11} y_1 = c_1 - A_{12} y_2 \Rightarrow \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} b_2 \\ b_4 \\ b_5 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 3 \\ 3 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_6 \end{bmatrix}$$

Ora: se A_{ii} è invertibile, $i=1, 2$, posso cercare Π_1, Π_2 : $\Pi_i A_{ii} \Pi_i^T = \begin{bmatrix} A_{ii}^{(i)} & A_{i2}^{(i)} \\ 0 & A_{22}^{(i)} \end{bmatrix}$

$$\begin{bmatrix} \Pi_1 & \\ \hline & \Pi_2 \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} \Pi_1^T & \\ \hline & \Pi_2^T \end{bmatrix} = \begin{bmatrix} \Pi_1 A_{11} \Pi_1^T & \Pi_1 A_{12} \Pi_2^T \\ 0 & \Pi_2 A_{22} \Pi_2^T \end{bmatrix} =$$

$$= \begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & * \\ 0 & A_{22}^{(1)} & A_{11}^{(2)} \\ 0 & A_{11}^{(2)} & 0 & A_{22}^{(2)} \end{bmatrix}, \text{ applicando ricorsivamente finché posibili otengo una matrice triangolare e ben definita.}$$

1 sistema lineare grossso \rightsquigarrow tanti sottoproblemi più semplici.

SISTEMI LINEARI

METODI DIRETTI: se eseguiti su sistema erette convergono in un n. finito di operazioni (e.g. eliminazione di Gauss).

METODI ITERATIVI: $\{x_k\}_{k \in \mathbb{N}}$: $\lim_{k \rightarrow \infty} x_k = x$, in generale non si trova x con un n. finito di operazioni.

METODI DIRETTI

Esempio (METODO DI GAUSS): Cramer è un metodo diretto robusto.

Se mi ha un sistema $Ux=c$ con U matrice triangolare superiore: $\begin{cases} U_{11}x_1 + \dots + U_{1n}x_n = c_1 \\ \vdots \\ U_{nn}x_n = c_n \end{cases}$

SOSTITUZIONE ALL'INDIETRO: parto da $x_n = \frac{c_n}{U_{nn}}$ e ritalgo.

$$\begin{cases} x_n = c_n/U_{nn} \\ x_j = (c_j - \sum_{i=j+1}^n U_{ji}x_i) / U_{jj} \quad \forall j = n-1, n-2, \dots, 1 \end{cases}$$

al passo j : $(n-j)$ moltiplicaz.
 $(n-j-1)$ somme, 2 divisioni

$$\Rightarrow \sum_{j=1}^n (2(n-j)+1) = n+2 \sum_{j=1}^{n-1} j = n + \frac{2(n-1)n}{2} = n + n(n-1) \approx \Theta(n^2)$$

Oss: il metodo con $Lx=c$, L triangolare inferiore si chiama SOSTITUZIONE IN AVANTI $\rightarrow \Theta(n^2)$

Il metodo di Gauss trasforma $Ax=b$ in $Ux=b$ equivalente e applica sostituzione all'indietro. Si sostituiscono le equazioni con cancellazioni lineari scelte in modo da annullare alcune variabili.

Alla riga j rompe la regola e moltiplica per $-\frac{a_{jj}}{a_{jj}}$ e procede iterativamente per $j=2, \dots, n$ e ottengo il sistema triangolare. $a_{ij} := \frac{a_{ij} - a_{ij}^{(j-1)}}{a_{jj}^{(j-1)}}$ si chiamano MOLTIPLICATORI. Per poter evitare l'algoritmo nero $a_{jj}^{(j-1)} \neq 0 \quad \forall j=1, \dots, n-1$. Inoltre $a_{jj}^{(j-1)} \approx 0$ può causare instabilità numerica.

Il costo delle riduzione $Ux=c$ è $\Theta(\frac{2}{3}n^3)$, quindi il costo di risolvere $Ax=b$ è dominato dalla riduzione.

FATTORIZZAZIONE LU

Il primo passo dell'algoritmo di Gauß equivale a moltiplicare $Ax=b$ a sx per una matrice

$$H_2 = \begin{bmatrix} 1 & & & \\ -l_{21} & \ddots & 0 & \\ \vdots & & \ddots & \\ -l_{n1} & & & 1 \end{bmatrix}, \quad l_{21} = \frac{\alpha_{21}}{\alpha_{11}}, \quad H_2 A = \begin{bmatrix} 1^{\text{ro rig}} \\ 2^{\text{ro rig}} + l_{21} \cdot 1^{\text{ro rig}} \\ \vdots \\ \dots \end{bmatrix}, \quad H_2 A x = H_2 b$$

$$\text{Il } k\text{-esimo passo è equivalente a premoltiplicare } H_k = \begin{bmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{k+1,k} & \ddots & \\ & & \vdots & & 1 \end{bmatrix}, \quad l_{kk} = \frac{\alpha_{kk}}{\alpha_{(k-1)(k-1)}}$$

la prima parte del metodo di Gauß equivale a fare $\underbrace{H_{n-1} H_{n-2} \dots H_1}_U A x = \underbrace{b}_{b}$
 (sarebbero invertibili)

$$H_{n-1} \dots H_1 A = U \Rightarrow A = H_1^{-1} \dots H_{n-1}^{-1} U = LU$$

Proprietà di H_i :

i) inversa facile: $H_i^{-1} = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & l_{ni} & 1 \end{bmatrix}$

ii) moltiplicazione facile:

$$H_i \cdot H_j = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & -l_{ni} & -l_{nj} \\ & & l_{mi} & l_{mj} \end{bmatrix}$$

$$H_i^{-1} \cdot H_j^{-1} = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & l_{ni} & l_{nj} \\ & & l_{mi} & l_{mj} \end{bmatrix}$$

Dalle proprietà viste che $L = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & l_{21} & \ddots \\ & & \vdots & & 1 \\ & & l_{n1} & \dots & l_{n(n-1)} \end{bmatrix}$. Una volta eseguita la prima parte di Gauß allora già trovata la fattorizzazione LU

Oss: se conosciamo già LU la risoluzione di $Ax=b$ costa $\mathcal{O}(n^2)$

$$Ax=b \Rightarrow \underbrace{Lx}_y = \underbrace{b}_y \rightarrow \begin{cases} \text{risolvere prima } Ly=b \\ \text{e dopo } Ux=y \end{cases} \leftarrow \text{sono 2 sistemi triangolari}$$

VARIANTI DEL MEG.

Si vuole risolvere $Ax = b$, $A \in \mathbb{C}^{n \times n}$, $B \in \mathbb{C}^{n \times s}$, sto risolvendo s sistemi lineari con la stessa matrice dei coefficienti. $Ax_1 = b_1, \dots, Ax_s = b_s$.

Possiamo eseguire le mosse di Gauss sulla matrice aumentata $[A|B] \rightarrow \left[\begin{array}{ccc|c} * & * & \dots & B^{(1)} \\ 0 & * & \dots & \\ 0 & 0 & \dots & \end{array} \right]$
 $\rightarrow \dots \rightarrow \left[\begin{array}{ccc|c} I & I & \dots & B^{(n-1)} \\ 0 & 0 & \dots & \\ 0 & 0 & \dots & \end{array} \right]$, alla fine risolverò i sistemi $Ux_1 = b_1^{(n-1)}, \dots, Ux_s = b_s^{(n-1)}$

Esempio:

$$X = \begin{bmatrix} X_1^{(1)} & X_2^{(1)} \\ X_1^{(2)} & X_2^{(2)} \\ X_3^{(1)} & X_3^{(2)} \end{bmatrix} \quad \begin{bmatrix} -2 & 1 & 1 & -2 & 0 \\ -1 & 3 & 1 & 3 & 0 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} -2 & 1 & 1 & -2 & 0 \\ 0 & 5/2 & 1/2 & -2 & 0 \\ 0 & 0 & 1/5 & 1/5 & 1 \end{bmatrix} \Rightarrow X = \begin{bmatrix} 1 & 2/11 \\ -1 & -1/11 \\ 1 & 5/11 \end{bmatrix}$$

Algoritmo Gaus-Jordan: caso particolare per calcolare l'inverso di A , con $n=5$
 $AX = I$, se mi fa l'algoritmo di Gauss con $B = I$, i vettori x_1, \dots, x_n sono le colonne di A^{-1} , infatti $AA^{-1} \equiv I$

IMPLEMENTAZIONE DELL'ALGORITMO DI GAUSS SU MATLAB

```
function X = gauss_solve(A, B)
    n = size(A, 1);
    s = size(B, 2);
    AB = [A, B]; % Costruisce la matrice aumentata

    % Eliminazione di Gauss
    for j = 1:n-1
        for i = j+1:n
            AB(i, j:n+s) = AB(i, j:n+s) - AB(j, j:n+s) * (AB(i, j) / AB(j, j));
        end
    end

    % Soluzione del sistema triangolare superiore
    X = zeros(n, s);
    for j = 1:s
        X(:, j) = supsolve(AB(:, 1:j), AB(:, n+j));
    end
end
```

SOSTITUZIONE
ALL'INDIETRO

```
function x=sup_solve(U,b)
s=size(U);
n=s(1);
x=zeros(n,1);
for i=n:-1:1
    p=b(i); % accumulatore
    for j=i+1:n
        p=p-U(i,j)*x(j);
    end
    x(i)=p/U(i,i);
end
```

PIVOTING

Cosa fare se tra i numeri $a_{ij} = 0$ (≈ 0)? Si può modificare l'algoritmo introducendo matrici di permutazione ovvero al passo si scambiano le righe del sistema lineare in questo modo:

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1} & a_{s2} & \cdots & a_{sn} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array} \right]$$

Cerchiamo l'elemento di **max modulis** i.e. un indice h :
 $|a_{h,j}| = \max_{i=1,\dots,n} |a_{i,j}|$ e poi scambiare la riga j con la h .

\Rightarrow Dopo lo scambio posso continuare l'algoritmo di Gauss perché le due i moltiplicazioni sono in modulo tutti ≤ 1 . Inoltre, se $\det A \neq 0$ allora necessariamente $a_{jj}^{(s-1)} \neq 0$.

Def: l'algoritmo di Gauss con queste modifiche si chiama **eliminazione con PIVOTING PARZIALE** (o pivoting per colonne)

Oss: se $a_{jj}^{(s-1)} = 0$ dopo lo scambio significa
 $= \det \left(\begin{array}{cc|c} * & * & * \\ * & * & * \\ \hline 0 & 0 & 0 \end{array} \right) \det \left(\begin{array}{cc|c} * & * & * \\ * & * & * \\ \hline 0 & 0 & 0 \end{array} \right) = 0$

$$\left[\begin{array}{cc|c} * & * & * \\ * & * & * \\ \hline 0 & 0 & 0 \end{array} \right] =: A^{(s)} \Rightarrow \det(A^{(s)}) = 0$$

esercizio: implementare
algo Gauss con pivoting
parziale in matlab

Esempio: $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 2 \\ 7 & 8 & 0 & 3 \end{array} \right] \xrightarrow{1 \leftrightarrow 3} \left[\begin{array}{ccc|c} 7 & 8 & 0 & 3 \\ 4 & 5 & 6 & 2 \\ 1 & 2 & 3 & 1 \end{array} \right] \xrightarrow{\text{mon}} \left[\begin{array}{ccc|c} 7 & 8 & 0 & 3 \\ 0 & 3/7 & 6 & 2/7 \\ 0 & 6/7 & 3 & 4/7 \end{array} \right] \xrightarrow{2 \leftrightarrow 3} \left[\begin{array}{ccc|c} 7 & 8 & 0 & 3 \\ 0 & 6/7 & 6 & 4/7 \\ 0 & 3/7 & 6 & 2/7 \end{array} \right] \xrightarrow{\text{mon}} \left[\begin{array}{ccc|c} 7 & 8 & 0 & 3 \\ 0 & 6/7 & 3 & 4/7 \\ 0 & 0 & 9/2 & 0 \end{array} \right]$$

Oss: ogni scambio di righe equivale a premoltiplicare A per una matrice Π_j

$$A \rightarrow \Pi_1 A \rightarrow H_1 \Pi_1 A \rightarrow \dots \rightarrow H_{n-1} \Pi_{n-1} \dots H_1 \Pi_1 A = U \boxed{1}$$

Per quanto riguarda la fattorizzazione LU: si ottiene la fattorizzazione del tipo:

$$\boxed{LU} = \boxed{\Pi} A \rightarrow \text{composizione di tutte le permutazioni}$$

\downarrow matrice triangolare alla fine

Attenzione: dev applicare sempre le regole: se si passa a applicare Π_3 , dev applicare lo stesso scambio alle prime 3-1 colonne di L

Esempio: $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix} \xrightarrow{\text{P}_1 A} \begin{bmatrix} 7 & 8 & 0 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \end{bmatrix} \xrightarrow{H_1, P_1, A} \begin{bmatrix} 7 & 8 & 0 \\ 0 & 3 & 2 & 0 \\ 0 & 4 & 7 & 3 \end{bmatrix} \xrightarrow{P_2 H_1, P_1, A} \begin{bmatrix} 7 & 8 & 0 \\ 0 & 4 & 7 & 3 \\ 0 & 3 & 1 & 0 \end{bmatrix} \xrightarrow{H_2, P_2, H_1, A} \begin{bmatrix} 7 & 8 & 0 \\ 0 & 4 & 7 & 3 \\ 0 & 0 & 3 & 1/2 \end{bmatrix} = U$

$P_1 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 0 \end{bmatrix}$ $L = \begin{bmatrix} 1 & 0 & 0 \\ 4/7 & 1 & 0 \\ 1/7 & 1 & 1 \end{bmatrix}$ $P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3/2 \end{bmatrix}$

$P_1: \begin{array}{l} 1 \rightarrow 3 \rightarrow 2 \\ 2 \rightarrow 2 \rightarrow 1 \\ 3 \rightarrow 1 \rightarrow 2 \end{array}$ $P_2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

Oss: per Bdc si ha $\det(PA) = \det P \cdot \det A$

$$\det(LU) = \det L \cdot \det U$$

$$\Rightarrow \det(A) = \underbrace{\det(P)}_{(-1)^n \text{ da tempo } 1}^{-1} \cdot \underbrace{\det(L)}_{\text{dove } n \text{ #rcalci}} \cdot \underbrace{\det(U)}_{\prod_{j=1}^n u_{jj}^{(j-1)}} = (-1)^n \cdot \prod_{j=1}^n u_{jj}^{(j-1)}$$



Quindi si può usare l'algoritmo di Gauss per il calcolo del determinante $\Theta(\frac{2}{3}n^3)$
MOLTO meglio della via per Laplace $\Theta(n!)$

Esercizio: calcola LU e LU con pivoting di $A = \begin{bmatrix} 4 & 1 & 1 & 1 \\ -4 & 2 & 1 & 1 \\ 4 & -2 & 1 & 2 \\ 0 & 3 & 0 & 0 \end{bmatrix}$, $A = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 2 & 0 \\ 4 & 4 & 3 \end{bmatrix}$

controllare $\text{NORM}(A - L \cdot U) \approx 0$

$[L, U] = LU(A)$, $[L, U, P] = LU(A)$ (con pivoting parziale)

CONDIZIONAMENTO DI UN SISTEMA LINEARE

Se $A \in \mathbb{C}^{n \times n}$, $b \in \mathbb{C}^n$, si vuole risolvere $x: Ax=b$, ma ci sono dei errori nei dati e/o arrotondamenti troviamo $x + \delta x \in \mathbb{C}^n$ che risolve $(A + \delta A)(x + \delta x) = b + \delta b$, con δ VETTORE DI PERTURBAZIONE

Se $\delta A, \delta b$ piccole, NON possiamo concludere che δx è piccolo rispetto a x .

Esempio: $\delta A = 0$ (autometrice erottica)

$$\begin{bmatrix} 1 & -1 \\ 1 & -1,00001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 \\ 1 & -0,00001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0,99999 \\ 1 \end{bmatrix} \Rightarrow x = \begin{bmatrix} -10^{-6} \\ -1 \end{bmatrix}$$

$$\text{Se } A \neq 0 : \begin{bmatrix} 3 & 4 \\ 3 & 9,00001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 7,00001 \end{bmatrix} \Rightarrow x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 4 \\ 3 & 3,99999 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 7,00001 \end{bmatrix} \Rightarrow x = \begin{bmatrix} 2,2/3 \\ -4 \end{bmatrix}$$

Perturbazioni dell'ante b
 10^{-6} in A, b , gessare perturbazione
dell'antice dell'ante in x

Da cosa dipende l'errore relativo $\frac{\|\delta x\|}{\|x\|}$?

Facciamo delle stime "dall'alto": assumiamo $\delta A = 0$, $\text{det } A \neq 0$

$$A(x + \delta x) = b + \delta b \Leftrightarrow A\delta x = \delta b \rightsquigarrow \delta x = A^{-1}\delta b \Rightarrow \|\delta x\| \leq \|A^{-1}\| \cdot \|\delta b\| \quad (1)$$

$$Ax = b \Rightarrow \|Ax\| = \|b\| \Rightarrow \|A\| \cdot \|x\| \geq \|b\| \rightsquigarrow \|x\| \geq \frac{\|b\|}{\|A\|} \quad (2)$$

$$(1+2) \frac{\|\delta x\|}{\|x\|} \leq \frac{\|A^{-1}\| \cdot \|\delta b\| \cdot \|A\|}{\|b\|} = \frac{\|\delta b\|}{\|b\|} \cdot \underbrace{\|A\| \cdot \|A^{-1}\|}_{\text{errore relativo in } b, \text{ controllabile}}$$

NUMERO DI CONDIZIONAMENTO
delle notizie segreti alle norme $\|\cdot\|$
 $\mu(A) := \|A\| \cdot \|A^{-1}\|$

Oss: se $\mu(A) \gg 1$ allora l'errore relativo POTREBBE essere molto più grande
dell'errore relativo nei dati e il problema si dice MAL CONDIZIONATO
(Worst-case)

Parlavamo dell'analisi dell'errore: supponiamo che abbiano perturbazioni in A e b (dovuti a errori di misurazione), per cui risolviamo $(A + \delta A)(x + \delta x) = b + \delta b$

Quanto più è grande l'errore relativo? $\frac{\|x + \delta x - x\|}{\|x\|} = \frac{\|\delta x\|}{\|x\|}$

$$\Rightarrow \text{se } \delta A = 0 \Rightarrow \frac{\|\delta x\|}{\|x\|} \leq \mu(A) \cdot \frac{\|\delta b\|}{\|b\|}, \quad \mu(A) = \|A\| \cdot \|A^{-1}\| \in \mathbb{R}^+$$

NUMERO DI CONDIZIONAMENTO

Oss: se $\mu(A) \approx 10^k$, k è il numero di cifre significative che si perdono nella soluzione calcolata ($x + \delta x$). Il caso migliore è quando $\mu(A) = 1 = 10^0$

Th: se $\|\delta A\| \cdot \|A^{-1}\| < 1 \Rightarrow \frac{\|\delta x\|}{\|x\|} \leq \frac{\mu(A)}{1 - \mu(A) \cdot \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right)$ caso peggior

Oss: $\delta A = 0$ si ottiene il risultato visto prima

Calcolare $\mu(A)$ è costoso perché serve calcolare A^{-1} . Allora si può rifinare; ad esempio se $A = A^H$ e considera $\|\cdot\|_2$ allora possiamo usare metodi per gli autovalori (e fissa caso):

$$\|A\|_2 \cdot \|A^{-1}\|_2 = g(A) \cdot g(A^{-1}) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \rightarrow \begin{array}{l} \text{(più grandi è} \\ \text{più piccole in} \\ \text{modulo)} \end{array}$$

Stime a posteriori: supponiamo di aver calcolato $\tilde{x} = x + \delta x \in \mathbb{C}^n$ con un qualche metodo. Si può guardare al vettore $r = b - A\tilde{x}$ **RESIDUO**.

Se $\|r\|$ piccolo $\Rightarrow \|x - \tilde{x}\|$ piccolo? Se sottraiamo $Ax = b$ da $r = b - A\tilde{x}$:

$$A(x - \tilde{x}) = r \Rightarrow x - \tilde{x} = A^{-1}r \Rightarrow \|x - \tilde{x}\| \leq \|A^{-1}\| \cdot \|r\|,$$

mentre $\|x\| \geq \frac{\|b\|}{\|A\|}$ si ottiene che $\frac{\|x - \tilde{x}\|}{\|x\|} \leq \frac{\|A\| \cdot \|A^{-1}\| \cdot \|r\|}{\|b\|} = \mu(A) \cdot \frac{\|r\|}{\|b\|}$ non dipende dal metodo.

Se $\mu(A) \approx 1$ allora $\|r\|$ e $\frac{\|x - \tilde{x}\|}{\|x\|}$ sono comparabili, se $\mu(A) \gg 1$ possono differire parecchio.

NON SI INVERTONO LE MATRICI!

$Ax = b \Rightarrow x = A^{-1}b$ fa schifo!

- caso $n=1$, $ax=b$, ci sono 2 algoritmi
 - $x = b/a$ 1 operazione 1 arrotondamento
 - $a^{-1}, x = a^{-1} \cdot b$ 2 operazioni 2 arrotondamenti
- caso generale, $Ax=b$
 - $A \setminus b$ (MATLAB) $\rightarrow O(\frac{2}{3}n^3)$
 - $\text{inv}(A) \cdot b \rightarrow O(\frac{8}{3}n^3)$, meno efficiente se $\mu(A)$ è alto la differenza è notevole

METODI ITERATIVI

Successione $\{x^{(k)}\}_{k \in \mathbb{N}}$: $\lim_{k \rightarrow \infty} x^{(k)} = x$ con x soluzione di $Ax=b$

Motivo: $O(n^3)$ conta troppo altri per matrice non disponibile, ma conoscere $v \rightarrow b$
 Idee per un "buon" metodo iterativo:

- ogni iterazione deve costare meno di $O(n^3)$ (es. prodotti matrice-vettore, sistemi lineari triangolari, diagonali...)
- date une certe accettate (es. norme del residuo) non mi servono troppe iterazioni per raggiungerle. (se l'iterazione conta n^2 , ma mi servono n iterazioni, allora non ha senso)

METODI DI PUNTO FISSO

MATRICE DI ITERAZIONE

$Ax-b=0$ si risolve come un'equazione di vettore $X = Hx+c$, $H \in \mathbb{C}^{n \times n}$, $c \in \mathbb{C}^n$

Modo 1) scelgo $G \in \mathbb{C}^{n \times n}$ invertibile e considero $x = x - G(Ax-b) = (I-GA)x - Gb$

Modo 2) $A = A_1 + A_2$, $Ax=b \Leftrightarrow (A_1 + A_2)x = b \Leftrightarrow A_1x = -A_2x + b \Leftrightarrow x = \underbrace{-A_1^{-1}A_2x}_{H} + \underbrace{A_1^{-1}b}_{c}$

Una volta trovata l'eq. $x = Hx+c$ considero il seguente metodo iterativo:

$$\begin{cases} x^{(0)} \text{ dato} \\ x^{(k+1)} = Hx^{(k)} + c, \quad k=1,2,\dots \end{cases}$$

Th della razza: il metodo $\begin{cases} x^{(0)} \text{ (anche scatto)} \\ x^{(k+1)} = Ax^{(k)} + c \end{cases}$ converge $\forall x^{(0)} \in \mathbb{C}^n$ se $\|f(H)\| < 1$

Dim: $x = Hx+c$ con x sol. esatta.

$$\underbrace{x^{(k+1)}}_{\dots} - x = Hx^{(k)} + c - (Hx+c) = H(\underbrace{x^{(k)} - x}_{\dots}) \Rightarrow e^{(k+1)} = He^{(k)} = H^K e^{(0)}$$

per cui $\lim_{K \rightarrow \infty} e^{(k+1)} = \lim_{K \rightarrow \infty} H^K e^{(0)} = 0$ poiché $g(H) < 0 \Rightarrow \lim_{K \rightarrow \infty} H^K = 0$ □

Corollario: $\|H\| < 1 \Rightarrow g(H) < 1$ quindi il metodo converge

$\|H\| > 1 \Rightarrow g(H) > 1$ NON POSSO DIRE NULLA

$|\det(H)| \geq 1 \Rightarrow g(H) > 1$ quindi il metodo non converge

$|\det(H)| < 1 \Rightarrow g(H) < 1$ NON POSSO DIRE NULLA

Velocità asintotica di convergenza: dalla dimostrazione, $\frac{\|e^{(k)}\|}{\|e^{(0)}\|} \leq \|H^K\|$

dell'algebra lineare, quando K è abbastanza grande $\|H^K\| \approx g(H)^K$

i.e. $\lim_{K \rightarrow \infty} \sqrt[K]{\|H^K\|} = g(H)$ allora:

Se ho 2 metodi da p.f. t.c. $g(H_1) < g(H_2) < 1 \Rightarrow$ il 1° converge più velocemente,
inoltre posso passare a stimare il numero di iterazioni K che mi servono per
rendere $\frac{\|e^{(k)}\|}{\|e^{(0)}\|} = \frac{\|x^{(k)} - x\|}{\|e^{(0)}\|} \leq \delta \in \mathbb{R}^+$; mi basta impostare $g(H)^K \leq \delta \Rightarrow K \geq \frac{\log(\delta)}{\log(g(H))}$

Criteri d'arresto: calcolo K è ottico poiché non $K = \lceil \log(\delta) / \log(g(H)) \rceil$
approssima $g(H)$; i più pratici controllano una di queste 2 condizioni

$$1) \frac{\|R^{(k)}\|}{\|b\|} < \delta \text{ TOLLERANZA}$$

$$2) \|x^{(k+1)} - x^{(k)}\| < \delta \text{ differenza tra le ultime iterazioni}$$

METODI DI JACOBI E GAUSS - SEIDEL

$$A = D - E - F = \begin{bmatrix} \cancel{0} \\ \cancel{0} \end{bmatrix} + \begin{bmatrix} \cancel{0} \\ \cancel{\Delta} \end{bmatrix} + \begin{bmatrix} \cancel{0} \\ \cancel{\nabla} \end{bmatrix} \Rightarrow Ax = b \Leftrightarrow (D - E - F)x = b$$

D $-E$ $-F$

Jacobi: $Dx = (E+F)x = b \Rightarrow x = \underbrace{D^{-1}(E+F)}_H x + \underbrace{D^{-1}b}_C$

$$\begin{cases} x^{(0)} \text{ dato} \\ x^{(k+1)} = D^{-1}(E+F)x^{(k)} + D^{-1}b = D^{-1}((E+F)x^{(k)} + b) \end{cases}$$

Oss: ad ogni iterazione calcola un prodotto matrice-vettore e risolve un sistema diagonale $\rightarrow O(n)$

$$H_s = D^{-1}(E+F) = -\begin{bmatrix} \frac{1}{a_{11}} & & \\ & \ddots & \\ & & \frac{1}{a_{nn}} \end{bmatrix} \begin{bmatrix} 0 & a_{12} - a_{11} & & \\ a_{21} & 0 & \ddots & \\ & \ddots & \ddots & a_{n-1,n} \\ a_{n1} & a_{n2} & \ddots & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & & \\ -\frac{a_{21}}{a_{11}} & 0 & \ddots & \\ & \ddots & \ddots & -\frac{a_{n-1,n}}{a_{11}} \\ -\frac{a_{n1}}{a_{11}} & -\frac{a_{n2}}{a_{11}} & \ddots & 0 \end{bmatrix}$$

Le equazioni che descrivono il passo di Jacobi sono $x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(k)} \right)$

Oss: per calcolare $x_i^{(k+1)}$ serve tutte le componenti di $x^{(k)}$ \Rightarrow non si può sovraccaricare $x^{(k)}$ prima di aver calcolato tutto $x^{(k+1)}$

metodo di Jacobi per risolvere $Ax=b$ in matlab

```
function x = jacobi(A, b, k)
    n = size(A, 1);
    x = zeros(n, 1); % Vettore iniziale
    D = diag(A); % Estrazione della diagonale di A
    R = A - diag(D); % Resto della matrice A senza la diagonale

    for j = 1:k
        x = (b - R * x) ./ D; % Calcolo della nuova iterazione
        res = norm(b - A * x); % Calcolo della norma del residuo
        fprintf('Iterazione %d, Residuo = %.6f\n', j, res);
    end
end
```

Gauss-Seidel: $(D-E)x = Fx + b \Rightarrow x = \underbrace{(D-E)^{-1}F}_H x + \underbrace{(D-E)^{-1}b}_C$

$$\begin{cases} x^{(0)} \text{ data} \\ x^{(k+1)} = (D-E)^{-1}F x^{(k)} + (D-E)^{-1}b \quad H_{GS} = (D-E)^{-1}F \end{cases}$$

Oss: non si fornisce H_{GS} , ma in ogni iterazione si calcola

$$x^{(k)} \rightsquigarrow Fx^{(k)} \quad O(n^2)$$

$$\text{si risolve } (D-E)y = Fx^{(k)} \quad (y = (D-E)^{-1}Fx^{(k)}) \quad O(n^2)$$

$(D-E)^{-1}b$ si calcola una volta risolvendo $(D-E)y = b$

Oss: H_{GS} ha come prime colonne $\begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ perché

$$\underbrace{\begin{bmatrix} \Delta \\ \vdots \\ 0 \end{bmatrix}}_{F = \begin{bmatrix} 0 & \lambda_2 & \dots & \lambda_n \end{bmatrix}}^{-1} \begin{bmatrix} 0 \\ \Delta \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} (D-E)^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ (D-E)^{-1} \lambda_2 \\ \vdots \\ (D-E)^{-1} \lambda_n \end{bmatrix} = \begin{bmatrix} 0 & * & \dots & * \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

posso risolvere $(D-E)x = d_s$

Oss: $x^{(k+1)} = (D-E)^{-1} F x^{(k)} + D^{-1} b$ si moltiplicano per $D^{-1}(D-E)$

$$x^{(k+1)} = \boxed{D^{-1} E x^{(k+1)}} + D^{-1} F x^{(k)} + D^{-1} b$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{h=i+1}^n a_{ih} x_h^{(k)} \right), \quad i=1, \dots, n$$

In questo modo ho 2 vantaggi:

- non risolvono sistemi triangolari
- per calcolare $x_i^{(k+1)}$ non ho bisogno di $x_h^{(k)}$, $h < i$; posso sfruttare le entrate di $x^{(k)}$

Oss: se solo due Gauß-Hidel hanno l'origine $a_{ii} \neq 0$ b_i , se queste condizioni non c'è posso fare trasformazioni simili sul sistema. Ad esempio applicare una matrice di permutazione.

Abbiamo visto il metodo del punto fisso:

31/03/2025

$$\begin{cases} x^{(0)} \text{ dato} \\ x^{(k+1)} = Hx^{(k)} + c, \quad \lim_{k \rightarrow \infty} x^{(k)} = x \text{ se } \|H\| < 1 \end{cases}$$

Abbiamo visto i metodi di Jacobi e Gauss-Seidel.

-) Jacobi: $H = D^{-1}(E+F)$, $A = \underbrace{\begin{bmatrix} a_{11} & & \\ & \ddots & \\ & & a_{nn} \end{bmatrix}}_D - \underbrace{\begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}}_E - \underbrace{\begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}}_F$
-) Gauss-Seidel: $H = (D-E)^{-1}F$

Esercizio d'esame: $A := \begin{bmatrix} 1 & 2/3 & 4/9 \\ 2/3 & 1 & 2/3 \\ 4/9 & 2/3 & 1 \end{bmatrix}$

- i. calcola H_G e determina se è convergente
ii. calcola H_S e determina se è convergente

i. $(D-E) = \begin{bmatrix} 1 & 0 & 0 \\ 2/3 & 1 & 0 \\ 4/9 & 2/3 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} 0 & -\frac{2}{3} & -\frac{4}{9} \\ 0 & 0 & -\frac{2}{3} \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \lambda_1 & \lambda_2 \\ 0 & 0 & 0 \end{bmatrix}$

$$H_{GS} = (D-E)^{-1}F = \left[(D-E)^{-1} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right] \left[(D-E)^{-1} \lambda_1 \right] \left[(D-E)^{-1} \lambda_2 \right] = \begin{bmatrix} 0 & x_1 & y_1 \\ 0 & x_2 & y_2 \\ 0 & x_3 & y_3 \end{bmatrix}$$

$$\begin{cases} x_1 = -2/3 \\ 2/3x_1 + x_2 = 0 \\ 4/9x_1 + 2/3x_2 + x_3 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = -2/3 \\ x_2 = -2/3x_1 = 4/9 \\ x_3 = -4/9x_1 - 2/3x_2 = 8/27 - 8/27 = 0 \end{cases}$$

$$\begin{cases} y_1 = -4/9 \\ 2/3y_1 + y_2 = -2/3 \\ 4/9y_1 + 2/3y_2 + y_3 = 0 \end{cases} \Rightarrow \begin{cases} y_1 = -4/9 \\ y_2 = -2/3 - 2/3y_1 = -10/27 \\ y_3 = -4/9y_1 - 2/3y_2 = \frac{36}{81} = \frac{4}{9} \end{cases}$$

$$H_{GS} = \begin{bmatrix} 0 & -2/3 & -4/9 \\ 0 & 4/9 & -10/27 \\ 0 & 0 & 4/9 \end{bmatrix}$$

Gli autovettori sono i tanti sulle diagonale $\Rightarrow g(H) = \frac{4}{9} < 1 \Rightarrow H_G$ convergente

ii. $D = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}, \quad H_S = D^{-1}(E+F) = E+F = \begin{bmatrix} 0 & -2/3 & -4/9 \\ -2/3 & 0 & -2/3 \\ -4/9 & -2/3 & 0 \end{bmatrix}$ bisogna trovare gli autovettori.

$$\det(HI-H_S) = \det \begin{bmatrix} 1 & 2/3 & 4/9 \\ 2/3 & 1 & 2/3 \\ 4/9 & 2/3 & 1 \end{bmatrix} = \lambda(x^2 - 4/9) - \frac{2}{3}\left(\frac{2}{3}x - \frac{8}{27}\right) + \frac{4}{9}\left(\frac{4}{3} - \frac{4}{3}x\right) =$$

$$= \lambda(x^2 - \frac{4}{9}) - \frac{4}{3}(x - \frac{4}{3}) + \frac{4}{9}\left(\frac{4}{3} - \frac{4}{3}x\right) = x^3 - \frac{4}{3}x - \frac{4}{3}(x - \frac{4}{3} + \frac{4}{3}x) = x^3 - \frac{32}{81}x + \frac{32}{81}$$

$p(\lambda)$ è divisibile per $(\lambda - \frac{4}{3})$ visto che $\frac{4}{3} \in \sigma(H)$, $p(\lambda) = (\lambda - \frac{4}{3})(\text{grado 2})$

$$\begin{array}{r|l} x^3 - \frac{80}{81}x + \frac{32}{81} & \lambda - \frac{4}{3} \\ -3 + \frac{4}{3}x^2 & x^2 + \frac{4}{3}x - \frac{32}{81} \\ \hline 1 & \end{array} \Rightarrow p(\lambda) = (\lambda - \frac{4}{3})(x^2 + \frac{4}{3}x - \frac{32}{81})$$

$$\frac{-4}{3} \pm \sqrt{\frac{16}{81} + \frac{32}{3}} = \sigma(H_S), \quad \left| \frac{-4}{3} - \sqrt{\frac{16+32}{81}} \right| > 1 \Rightarrow H_S \text{ diverge}$$

Teorema: se $A \in \mathbb{C}^{n \times n}$ soddisfa una delle seguenti

i. A ha predominanza diagonale forte

ii. A irriducibile e ha predominanza diagonale debole

\Rightarrow il metodo GS e di scambi dom. iterazioni convergenti applicati a $Ax=b$

Dim: si osserva che $a_{ii} \neq 0 \forall i=1, \dots, n$

i. diagonale forte \Rightarrow omio

ii. diagonale debole \Rightarrow se avessi $a_{ii}=0 \Rightarrow a_{jj}=0 \forall j=1, \dots, n$ NECESSARIAMENTE

ma la matrice è non singolare e sarebbe assurdo per Gerh.

$$A = \begin{bmatrix} * & * & \dots & * \\ 0 & * & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & * \end{bmatrix}$$

(Jacobi)

$$H_J = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & & \\ -\frac{a_{12}}{a_{11}} & 0 & \dots & \\ & \ddots & \ddots & -\frac{a_{nn}}{a_{nn}} \\ & & \ddots & 0 \end{bmatrix}, \quad \|H_J\|_\infty = \max_{i=1, \dots, n} \sum_{j=1}^n \left| \frac{a_{ij}}{a_{ii}} \right|$$

i. se A è a diagonale forte $|a_{ii}| > \sum_{\substack{j \neq i \\ j=1}} |a_{ij}| \Rightarrow \|H_J\|_\infty < 1 \Rightarrow g(H_J) \leq \|H_J\| < 1$

ii. se A è diagonale debole gli autovalori λ di H_J sono t.c. $|\lambda| \leq 1$, ma dato che la matrice è irriducibile, non vi possono essere $|\lambda|=1$ (tutti i cerchi di G. hanno centro in 0 e raggio ≤ 1 e ce ne sono almeno uno con raggio < 1).

(Gershgorin)

Supponiamo che esiste $\lambda \in \sigma(H_A)$: $|\lambda| \geq 1$, allora vale $0 = \det(\lambda I - H_A)$

$$= \det(\lambda I) - (\lambda I - H_A) = \det(\lambda I - (\lambda I - H_A)) = \det((\lambda I - H_A)^{-1} \cdot [\lambda(\lambda I - H_A)])$$

$$= \det(\lambda I - H_A)^{-1} \cdot \det(\lambda(\lambda I - H_A)) \text{ per Binet-Cauchy}$$

$\overset{\text{X}}{0} \quad \overset{\text{X}}{\lambda} \text{ per } \lambda \neq 0$

$$\Rightarrow 0 = \det(\lambda(\lambda I - H_A)) = \underset{0}{\lambda^n} \det(\lambda I - H_A) \Rightarrow 0 = \det(\lambda I - H_A)$$

$$= \det \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

$\overset{\text{X}}{B}$

gli elementi sopra la diagonale hanno modulo $\leq |a_{ii}|$
corrispondenti alla matrice A

Quindi A DDF \Rightarrow B DDF, A_{000} è irriducibile $\Rightarrow B_{000}$ è irriducibile.

Per Gerhgorin B è non singolare e questo va in contraddizione con $\det B = 0$ \square

SISTEMI RETTANGOLARI

$Ax = b$, $A \in \mathbb{C}^{m \times n}$ m equazioni in n incognite, $x \in \mathbb{C}^n$, $b \in \mathbb{C}^m$

Supponiamo che A sia di rango nono, i.e. $\text{rk}(A) := \min\{m, n\}$. Anche qui si può fare lo svolgimento Gaussiano con o senza pivoting.

($m < n$)

$$\left[\begin{array}{c|c} * & x \\ \vdots & \vdots \\ * & x \\ \hline 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{array} \right]$$

rendo il vettore quadrato aggiungendo $(n-m)$ eq. $0=0$

$$\left[\begin{array}{c|c} * & x \\ \vdots & \vdots \\ * & x \\ \hline 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{array} \right] \rightarrow \left[\begin{array}{c|c|c} * & x & * \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{array} \right]$$

$n-m$ parametri liberi SISTEMI SOTTODETERMINATI

($m > n$)

$$\left[\begin{array}{c|c} * & x \\ \vdots & \vdots \\ * & x \\ \hline 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{array} \right] \xrightarrow{\text{ricond.}} \left[\begin{array}{c|c|c} * & x & * \\ \vdots & \vdots & \vdots \\ 0 & 0 & ? \\ \hline 0 & 0 & ? \\ \vdots & \vdots & \vdots \\ 0 & 0 & ? \end{array} \right]$$

le ultime $m-n$ equazioni sono delle forme
 $0=?$ SISTEMI SOVRADETERMINATI.

Se tutti $i?$ sono 0 allora ha soluzione,
 Altrimenti non ha soluzione: $\text{rk}(Ab) \neq \text{rk}(A)$

Se il vettore non ha soluzione si considera il problema di minimizzare il
 rendere $b - Ax = R(x)$, ovvero si cerca di risolvere $\min_{x \in \mathbb{C}^n} \|R(x)\|_2$ PROBLEMI DEI
 MINIMI QUADRATICI

Oss: è equivalente a risolvere $\min_{x \in \mathbb{C}^n} \|R(x)\|_2^2$, liberandosi dalla radice delle
 norme che rompe i collegamenti.

Supponiamo di risolvere $\min_{x \in \mathbb{C}^n} \|R(x)\|_2^2$. Consideriamo il funzionale $\Psi(x) : \mathbb{R}^n \rightarrow \mathbb{R}^+$

$\Psi(x) := \underbrace{(b - Ax)^T (b - Ax)}_{\|b - Ax\|_2^2}$ è convessa (ns sim) \Rightarrow i p.t. di minimo sono quelli che
 verificano $\nabla \Psi(x) = 0$

$$\|R(x)\|_2^2 = \sum_{i=1}^m R_i(x)^2, \text{ dove } R_i(x)^2 = (b_i - \sum_{j=1}^n a_{ij} x_j)^2$$

$$\frac{\partial}{\partial x_j} R_i(x)^2 = 2(b_i - \sum_{j=1}^n a_{ij} x_j)(-a_{ij}), \quad i=1, \dots, m$$

$$\frac{\partial}{\partial x_j} \Psi(x) = \sum_{i=1}^m \frac{\partial}{\partial x_j} (R_i(x)^2) = 2 \sum_{i=1}^m \left[\sum_{j=1}^n a_{ij} a_{ij} x_j - a_{ij} b_i \right] = 2 \sum_{i=1}^m \sum_{j=1}^n a_{ij} a_{ij} x_j - \sum_{i=1}^m a_{ij} b_i$$

$$= 2 \left(\sum_{i=1}^m a_{ij} \sum_{j=1}^n a_{ij} x_j - \sum_{i=1}^m a_{ij} b_i \right) = 2 (A^T A x - A^T b)_j$$

$$\Rightarrow \nabla \Psi(x) = 0 \iff A^T A x - A^T b = 0 \iff A^T A x = A^T b \quad \text{SISTEMA DELLE EQ. NORMALI}$$

spoiler: esistono sempre
Le soluzioni del problema (se esistono) sono le soluzioni di $A^T A x = A^T b$ (sistema quad.)

Abbiamo visto il problema di minimi quadrati: $Ax = b$, $A \in \mathbb{C}^{m \times n}$, $m > n$

Se $\text{rk}(A|b) > \text{rk}(A)$ si risolve $\min_{x \in \mathbb{C}^n} \|Ax - b\|_2 \equiv \min_{x \in \mathbb{C}^n} \|b - Ax\|_2$ (P_1)

Il p.t. di minimo di (P_1) è lo stesso del problema $\min_{x \in \mathbb{C}^n} \|Ax - b\|^2$, $\Psi(x): \mathbb{C}^n \rightarrow \mathbb{R}^+$
 $\Psi(x) := \|b - Ax\|^2 = (b - Ax)^H(b - Ax) = \sum_{i=1}^m |r_i|^2$ convessa.

Per semplicità si considera $\Psi(x): \mathbb{R}^n \rightarrow \mathbb{R}^+$, con le soluzioni $\nabla \Psi(x) = 0$. Dato che $\Psi(x)$ è quadratica ci sono m sistemi lineari $A^T A x = A^T b$

Th. $A^T A x = A^T b$ ammette soluzione $\forall A \in \mathbb{R}^{m \times n}$ e $\forall b \in \mathbb{C}^m$

Inoltre, la soluzione è unica se $\text{rk}(A) = n$ ($m > n$), quindi A è rk max

Dim: Per l'esistenza dobbiamo verificare $A^T b \in \text{Im}(A^T A)$ i.e. $\text{rk}(A^T A | A^T b) = \text{rk}(A^T A)$. Questo perché $A^T b \in \text{Im}(A^T A) \Rightarrow A^T b = \sum_{j=1}^m c_j A_j$ dove c colonne di $A^T A$.

$$\text{Im}(A^T A) = \{y \in \mathbb{C}^n : y = A^T A z, z \in \mathbb{C}^n\}$$

$$A^T b \in \text{Im}(A) = \{y \in \mathbb{C}^n : A z, z \in \mathbb{C}^n\} \text{ allora ho che } A^T b = A^T A z \in \text{Im}(A).$$

In generale, possiamo scrivere $\mathbb{C}^m = \text{Im}(A) \oplus \text{Im}(A)^\perp$, dove $\text{Im}(A)^\perp := \{y : y^H z = 0 \forall z \in \text{Im}(A)\}$. $b \in \mathbb{C}^m \Rightarrow b = b_1 + b_2$, $b_1 \in \text{Im}(A)$, $b_2 \in \text{Im}(A)^\perp$

$$A^H b = A^H b_1 + A^H b_2, \text{ dove } A^H b_1 \in \text{Im}(A^T A), A^H b_2 = \begin{bmatrix} a_1^H b_2 \\ a_2^H b_2 \\ \vdots \\ a_m^H b_2 \end{bmatrix} = 0 \text{ perché } b_2 \in \text{Im}(A)^\perp$$

e le colonne di A stanno in $\text{Im}(A)$

Per l'unicità, bisogna verificare che $\det(A^T A) \neq 0$, $\det(A^T A) = \sum_{\substack{\uparrow \\ [k]}} \det(A_{[k]}) \det(A_{[k]})$

Bisogna dimostrare che $\exists A_{[k]} : \det(A_{[k]}) \neq 0$.

Questa esiste sempre perché $\text{rk}(A) = n$ max.

Binet-Cauchy generalizzato. Se sono tutte
vane in tutte le possibili scelte
di n indici in $\{1, \dots, m\}$

□

Esempio: calcola la soluzione nel senso dei minimi quadrati di $\begin{cases} ax_1 - x_2 = 1 \\ -x_1 + x_2 = 0 \\ x_1 + 2x_2 = 1 \end{cases}$

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 1 \\ 1 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, A^T A = \begin{bmatrix} 2 & -1 & 1 \\ -1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 6 & -1 \\ -1 & 6 \end{bmatrix}$$

$$A^T b = \begin{bmatrix} 2 & -1 & 1 \\ -1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$A^T A x = A^T b \Leftrightarrow \begin{cases} 6x_1 - x_2 = 3 \\ -x_1 + 6x_2 = 1 \end{cases} \Rightarrow \begin{cases} x_1 = 19/35 \\ x_2 = 9/35 \end{cases}$$

$\|x\|_2 = \|b - Ax\|_2$

$$\|x\|_2 = \sqrt{\frac{19}{35}^2 + \frac{9}{35}^2} = \sqrt{\frac{362}{1225}} = \sqrt{\frac{362}{1225}}$$

Esercizio: $\begin{bmatrix} 1 & 1 & 1 \\ 1 & \alpha & 1 \\ 0 & 0 & \alpha \\ \beta & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix},$ $\alpha, \beta \in \mathbb{R}$ parametri. Per quali α, β la soluzione del problema minimi quadrati NON è unica? (i.e. $\text{rk}(A) < n = 3$)

Oss: A los rx 3 nos da solamente 1 d.m. 3×3 invertible.

de possibili s.m 3×3 nero 4. Supponiamo che i minori siano tutti nulli.

$$A_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix}, \det(A_1) = \lambda(\lambda - 1) \quad A_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \lambda & 1 \\ \beta & 0 & 0 \end{bmatrix}, \det(A_2) = \beta(1 - \lambda),$$

$$A_3 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & \alpha \\ \beta & 0 & 0 \end{bmatrix}, \det(A_3) = \alpha \beta, \quad A_4 = \begin{bmatrix} 1 & \alpha & 1 & 1 \\ 0 & 0 & \alpha & 1 \\ \beta & 0 & 0 & 1 \end{bmatrix}, \det(A_4) = \alpha^2 \beta$$

$$\begin{cases} \alpha(\alpha-1) = 0 \\ \beta(1-\alpha) = 0 \\ \alpha\beta = 0 \\ \alpha^2\beta^2 = 0 \end{cases} \Rightarrow \alpha = 0, \beta = 0 \text{ or } \alpha = 1, \beta = 0$$

$$A \in \mathbb{C}^{n \times m} \quad (n \times m) \quad (m \times n)$$

Oss: Risolvere le equazioni razionali coste come calcolare $A^H A \rightsquigarrow \mathcal{O}(mn^2)$ + calcolare $A^H b \rightsquigarrow \mathcal{O}(mn)$ + risolvere $A^H A x = A^H b \rightsquigarrow \mathcal{O}(n^3)$. Quindi $\mathcal{O}(mn^2 + n^3)$.

El problema i que $A^T A$ ha un numer di condicionaments alto, ad exemplis se $m = n$ si ha $\mu(A^T A) = \mu(A)^2$. Se $\mu(A^T A)$ non i molt controlat ($< 10^4$, tè piccolo) si applica un mètode alternatiu.

METODO QR PER PROBLEMI AI MINIMI QUADRATI

Def: At $\mathbb{C}^{m \times n}$, $m \geq n$, we have **FACTORIZZAZIONE QR** di A in uscita (Q, R) : $A = QR$, at $\mathbb{C}^{m \times m}$ unitarie ($Q^H Q = Q Q^H = I$), $R \in \mathbb{C}^{m \times n}$ triangolare superiore in sensi rettangolare i.e. $R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$, $R_1 \in \mathbb{C}^{n \times n}$ triangolare superiore

Oss: $A = QR$, potrei prendere $D = \begin{bmatrix} d_1 & \\ & \ddots & d_m \end{bmatrix}$ con $|d_j| \geq 1$, $j=1, \dots, m$ e nello che $A = Q_2 R_2$ dove $Q_2 = QD$ mantiene perciò diagonalizzabile, $R_2 = D^T R$ triangolare sup.

Prop. le norme 2 e la norma di Frobenius sono equivalenti per moltiplicazioni con matrici unitarie, i.e. $\|A\|_2 = \|QAT\|_2 = \|AQ\|_2 = \|Q, A, Q\|_2$, $\|A\|_F = \|QAT\|_F = \|AQ\|_F = \|Q, A, Q\|_F$, con Q_1, Q_2, Q unitarie

$$\text{Dim: } \|A\|_2 := \sqrt{\rho(A^H A)} \Rightarrow \|AQ\|_2 = \sqrt{\rho((QA)^H A Q)} = \sqrt{\rho(Q^H A^H A Q)} = \sqrt{\rho(A^H A)}$$

matrice simile ad $A^H A$ poiché $Q^H = Q^{-1} \Rightarrow$ ha gli stessi autovalori di $A^H A$

$$\|A\|_F := \sqrt{\text{tr}(A^H A)} \Rightarrow \|AQ\|_F = \sqrt{\text{tr}(Q^H A^H A Q)} = \sqrt{\text{tr}(A^H A)}$$

sono degli autovalori di $Q^H A^H A Q$, uguali per similitudine a quelli di $A^H A$ □

Nel problema $\min_{x \in \mathbb{C}^n} \|b - Ax\|_2 = \min_{x \in \mathbb{C}^n} \|b - QRx\|_2 = \min_{x \in \mathbb{C}^n} \|Q^H b - Rx\|_2$ per le propriez. []

Oss: $c := Q^H b = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$ n ragionevoli, osserva che $\|c - Rx\|_2^2 = \left\| \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} - \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x \right\|_2^2 = \left\| \begin{bmatrix} c_1 - R_1 x \\ c_2 \end{bmatrix} \right\|_2^2 =$
 $= \|c_1 - R_1 x\|^2 + \|c_2\|^2$ le ragioni più piccole possibile.
↓ non possiamo farci nulla perché non dipende da x
Selezio $x: R_1 x = c_1$ (possso farlo se R_1 invertibile ovvero se A è di rk max)
 $x = R_1^{-1} c_1$

Algoritmo (METODO QR):

- 1. calcolo fattorizzazione $A = QR \rightsquigarrow \mathcal{O}(m \cdot n^2)$
- 2. calcolo $Q^H b = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \rightsquigarrow \mathcal{O}(m \cdot n)$ dipende da come viene calcolata Q
- 3. Risolve $R_1 x = c_1$ sistema lineare triangolare $n \times n \rightsquigarrow \mathcal{O}(n^2)$

1. Come il metodo di Gauss, si premoltiplica A per matrici unitarie che "intensificano" gli elementi sotto le diagonale $A = \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{bmatrix}$ 😊

Prop: dato $v \in \mathbb{C}^m$ \exists H_v unitaria t.c. $H_v \cdot v = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix} = c \cdot e_1$, $c \in \mathbb{C}$
 Dato che moltiplicare per matrici unitarie non cambia $\|\cdot\|_2$ allora $|c| = \|v\|_2$

Tuttavia, H_0 non è utile e può essere scritta come: $\tilde{H}_0 = I - \frac{2\tilde{v}\tilde{v}^H}{\|\tilde{v}\|_2^2}$ dove
 $\tilde{v} = v \pm \|v\|_2 \cdot e_1 = \begin{bmatrix} v_1 \pm \|v\|_2 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$

MATRICE DI HOUSEHOLDER ASSOCIASTA AL VETTORE v

Smiley: secolo la matrice di Householder associata ad a_1 : $H_1 [a_1 | a_2 | \dots | a_n] = \begin{bmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$.

Per il passo 2 secolo la matrice di Householder associata a a_2 : H_2

$$\tilde{H}_2 \text{ } (m-1) \times (n-1) \text{ è premoltiplicatore } \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \tilde{H}_2 \\ \vdots & & & \\ 0 & & & \end{bmatrix} = H_2$$

Dopo n passi $\rightarrow H_n H_{n-1} \cdots H_1 A = \begin{bmatrix} R \\ 0 \end{bmatrix} \Rightarrow A = H_1^H \cdot H_2^H \cdots H_n^H R$
 Q unitarie poiché prodotto di matrici unitarie

Oss: per ogni costo $\mathcal{O}(mn^2)$ (e anche costi $\mathcal{O}(mn)$ per $Q^H b$) si usa il fatto che
 $H_2 := \text{Identità} + \text{range } 1 = I + ab^T = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & 1 \end{bmatrix} + \begin{bmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & 1 \end{bmatrix}$

$H_1 \cdot B = B - \underbrace{\left(\begin{array}{c|c} 0 & \cdot \\ \hline \cdot & B \end{array} \right)}_{\text{COSTO } \mathcal{O}(mn)}$, nel caso $H_2(b) \approx \mathcal{O}(m) \Rightarrow H_n H_{n-1} \cdots H_1 b \approx \mathcal{O}(mn)$

Oss: si può usare la fattorizzazione QR per risolvere sistemi lineari (nel caso $m=n$ viene $\mathcal{O}(\frac{4}{3}n^3)$ che è asintoticamente equivalente ma con il doppio rispetto a Gau).

$$Ax = b \Leftrightarrow QRx = b \approx Rx = Q^H b$$

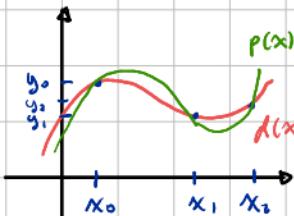
Oss: se $A \in \mathbb{R}^{m \times n}$ si può restare nei reali ed avere $Q \in \mathbb{R}^{m \times m}$, $Q^T Q = I$, $R \in \mathbb{R}^{m \times n}$

```
function H = householder_mat(v)
    H = eye(length(v));
    vt = v;
    if vt(1) < 0
        vt(1) = vt(1) - norm(v);
    else
        vt(1) = vt(1) + norm(v);
    end
    H = H - 2 * vt * vt' / norm(vt)^2;
end
```

MATLAB: calcola la matrice Householder associata a v .

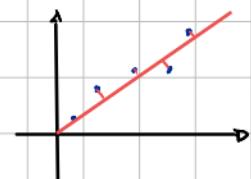
PRO TIP: $B = blkdiag(A_1, \dots, A_n)$ continua una matrice diagonale a blocchi.

INTERPOLAZIONE DI FUNZIONI



- 1) Cerchiamo un POLINOMIO che passa per i punti $\{x_0, \dots, x_K\}$.
Se una funzione è regolare, un polinomio è una buona appross. (stileggi di Taylor).

- 2) Oppure una funzione che passa "vicino" a quei punti:
oss: minimizza la distanza al quadrato (RETTA DI REGRESSIONE)



1. Trovare $p(x)$: $p(x_i) = y_i \quad \forall i=0, \dots, K \rightarrow K+1$ equazioni

$$p(x) = a_0 + a_1 x + \dots + a_d x^d \rightarrow d+1 \text{ equazioni}$$

Oss: se $d:=K$ ha $K+1$ equazioni con $K+1$ incognite. $\begin{cases} a_0 + a_1 x_0 + \dots + a_K x_0^K = y_0 \\ \vdots \\ a_0 + a_1 x_K + \dots + a_K x_K^K = y_K \end{cases}$ (sist. lineare)

Oss: le basi dei monomi $\{1, x, x^2, \dots, x^K\}$ non sono i l'una nello (*)

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^K \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_K & x_K^2 & \dots & x_K^K \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_K \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_K \end{bmatrix}$$

VANDERMONDE

Oss: se $i \neq s$, $x_i = x_s$ ci sono due righe identiche e ricorronte i singolari.



Th: $\det(V) = \prod_{0 \leq i < j \leq K} (x_j - x_i)$ (infatti se $x_i = x_s \forall i \neq s \Rightarrow V$ è invertibile)

Per trovare $p(x)$ risolviamo $V \cdot a = y$, e $p(x) = \sum_{i=0}^K a_i x^i$

Oss: $\deg(p(x)) \leq K$ in generale, ($\forall a^K \neq 0 \Rightarrow \deg(p(x)) < K$)

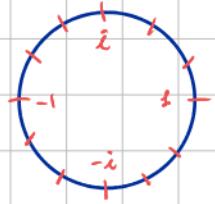


$$p(x) = a_0 + a_1 x + \dots + a_K x^K$$

Così si determina $p(x) \sim O(K^3)$

Ricordiamo il numero di condizionamento $\mu(V) = \|V\| \cdot \|V'\|$ controlla l'errore in $\begin{bmatrix} a_0 \\ \vdots \\ a_K \end{bmatrix}$.
Si può dimostrare che $\mu(V)$ cresce esponenzialmente in K per $\{x_0, \dots, x_K\}$ equispaziati. $\mu \leq \frac{1}{\sqrt{J}} \approx 10^{10}$ l'errore nei coefficienti è grande, ma i residui $|p(x_i) - y_i|$ sono "OK"

Esempio: un bel caso che viene "bene" sono p.ti equispaziati sulla circonferenza nel piano complesso \Rightarrow trasformata veloce di Fourier



(*) carica base e prendi $\{\varphi_0(x), \varphi_1(x), \dots, \varphi_K(x)\}$ con φ_i polinomi.

Se i una base, allora $p(x) = a_0 \varphi_0(x) + \dots + a_K \varphi_K(x)$ t.c. $p(x_i) = a_0 \varphi_0(x_i) + \dots + a_K \varphi_K(x_i)$

$$\begin{bmatrix} \varphi_0(x_0), \dots, \varphi_K(x_0) \\ \vdots \\ \varphi_0(x_K), \dots, \varphi_K(x_K) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_K \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_K \end{bmatrix} \quad \text{l'entrata } V_{ij} = \varphi_i(x_j) \quad \forall i, j = 0, \dots, K$$

Allora visto che una possibile scelta delle basi sono i MONOMI. Ci sono infinite scelte di base, ma alcune scelte sono le basi di LAGRANGE, di NEWTON. Queste basi ci daranno un numero di condizionamento migliore perché $V = M\tilde{V} \Rightarrow \mu(V) \leq \mu(M) \cdot \mu(\tilde{V})$

Oss: dato $Ax=b$, $x \approx \tilde{x}$, si ha che $\|x - \tilde{x}\| \leq \mu(A) \cdot \|A\tilde{x} - b\|$

errore n. colonna

n. righe

Oggi esempio: $A = \begin{bmatrix} 1 & \varepsilon \\ 1 & 1/\varepsilon \end{bmatrix}$, $\varepsilon \geq 0$, $\mu_2(A) = \sqrt{\lambda_1 \lambda_2} = \sqrt{\|A\|_2 \cdot \|A^T\|_2} = \frac{1}{\varepsilon}$

$$x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad b = \begin{bmatrix} \varepsilon \\ 1 \end{bmatrix}, \quad \tilde{x} = \begin{bmatrix} 1 \\ 1+\delta \end{bmatrix}$$

$$\|x - \tilde{x}\| = |\delta|, \quad \|A\tilde{x} - b\|_2 = \left\| \begin{bmatrix} 0 \\ \delta\varepsilon \end{bmatrix} \right\|_2 = |\delta| \cdot \varepsilon \rightsquigarrow |p(x_i) - y_i|$$

\Rightarrow errore sui coefficienti

può essere che i coefficienti
siano fuori controllo, ma
il residuo è ok

BASE DI LAGRANGE

$\{x_0, \dots, x_K\}$ guarda i nodi di interpolazione. Cosa $\{l_0(x), \dots, l_K(x)\}$ che non annullano in tutti i nodi a parte uno, $l_i(x) = (x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_K)$.

Calcolarla è scomodo, quindi si normalizza: $l_i = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$.

↳ molti res. ($x - x_i$)

Sai che $\deg(l_i) = K$.

$$p(x) = a_0 l_0(x) + \dots + a_K l_K(x) \quad \text{t.c.}$$

$$p(x_i) = \underbrace{a_0 l_0(x_i)}_0 + \underbrace{a_1 l_1(x_i)}_0 + \dots + \underbrace{a_{i-1} l_{i-1}(x_i)}_0 + \underbrace{a_i l_i(x_i)}_1 + \dots + \underbrace{a_K l_K(x_i)}_0 = y_i \Leftrightarrow a_i = y_i$$

$$l_i(x_j) = 0 \quad \forall j \neq i$$

per la normalizzazione $l_i(x_i) = 1$

Il polinomio interpolante è una combinazione lineare dei $p(x_i)$. $p(x) = \sum_{i=0}^K y_i l_i(x)$

$$V = \begin{bmatrix} l_0(x_0) & l_1(x_0) & \dots & l_K(x_0) \\ l_0(x_1) & l_1(x_1) & \dots & l_K(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ l_0(x_K) & l_1(x_K) & \dots & l_K(x_K) \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \neq 0 \end{bmatrix} = I \Rightarrow I\alpha = y \Rightarrow \alpha = y \text{ (grazie al caros)}$$

Esempio:

x	0	1	2	-1
y	1	1	3	3

$$x_0=0 \Rightarrow l_0(x) = \frac{(x-1)(x-2)(x+1)}{(-1)(-2)(1)}$$

$$x_1=1 \Rightarrow l_1(x) = \frac{(x-0)(x-2)(x+1)}{(1)(-1)(2)}$$

$$x_2=2 \Rightarrow l_2(x) = \frac{(x-0)(x-1)(x+1)}{(2)(1)(3)}$$

$$x_3=-1 \Rightarrow l_3(x) = \frac{(x-0)(x-1)(x-2)}{(-1)(-2)(-3)}$$

$$p(x) = 1l_0(x) + 1l_1(x) + 3l_2(x) + 3l_3(x) = \dots$$

Problema: se aggiungo un nodo di interpolazione x_{K+1} cambia completamente il polinomio interpolante. Questo problema lo risolve la base di Newton

BASE DI NEWTON

$$\{x_0, \dots, x_K\}, \quad n_0(x) = 1, \quad n_1(x) = x - x_0, \quad n_2(x) = (x - x_0)(x - x_1), \quad \dots, \quad n_K = (x - x_0) \dots (x - x_{K-1})$$

Oss: $n_i(x)$ dipende solo da $\{x_0, \dots, x_{i-1}\}$. Se aggiungi il nodo x_{K+1} allora $n_{K+1}(x) = (x - x_0) \dots (x - x_K)$ e tutti gli n_i non cambiano $\forall i = 0, \dots, K$

$$V = \begin{bmatrix} n_0(x_0) & n_1(x_0) & \dots & n_K(x_0) \\ \vdots & & & \\ n_0(x_K) & \dots & n_K(x_K) \end{bmatrix}, \quad p(x) = \sum_{i=0}^K a_i n_i(x), \quad p(x_c) = y_i \Leftrightarrow V\alpha = y$$

Si ottiene che V è una matrice triangolare superiore $V = \begin{bmatrix} 1 & * & & \\ & \ddots & & \\ & & 1 & \times \\ & & & \end{bmatrix}$ volendo potrei normalizzarla

Oss: $\mathcal{O}(K^2)$ per calcolare a_i , notare che dipende solo da $\{y_0, \dots, y_i\}$

$$\text{Def: } d[x_0, \dots, x_n, x] := \begin{cases} d(x), & n=-1 \quad d[x]=f(x) \\ \frac{d(x) - d(x_0)}{x - x_0}, & n=0 \quad d[x_0, x] = \frac{d(x) - d(x_0)}{x - x_0} \\ d[x_0, \dots, x_{n-2}, x] - d[x_0, \dots, x_{n-1}], & n>0 \quad \text{def ricorsiva} \end{cases}$$

Th: $a_i = d[x_0, \dots, x_i]$ DIFFERENZE DIVISE

Ancora sull'interpolazione di Newton:

11/01/2025

$(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k)$ con $x_i \neq x_j \forall i \neq j$ p.t. che riguarda interpolare con un polinomio di grado al più k , $\exists! p_k(x)$

$p_k(x)$ si può rappresentare nella base dei monomi $\{1, \dots, x^k\}$, ma anche nella base di Lagrange, come $p_k(x) = \sum_{j=0}^k y_j l_j(x)$ con $l_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^{k-1} \frac{x - x_i}{x_j - x_i} \quad \forall j$

C'è anche la base di Newton, $p_k(x) = \sum_{j=0}^k d[x_0, \dots, x_j] M_j(x)$

dove $M_0(x) = 1$, $M_j(x) = (x - x_0) \dots (x - x_{j-1}) \quad \forall j \geq 1$.

I coefficienti $d[x_0, \dots, x_j]$ si definiscono per ricorrenza.

Def: Se $f: \mathbb{R} \rightarrow \mathbb{R}$ e numeri $x_0, \dots, x_{k-1} \in \mathbb{R}$, $x_i \neq x_j \forall i \neq j$

la funzione DIFFERENZA DIVISI DI ORDINE k è data da:

$$\begin{cases} d[x] = d(x) & , k=0 \\ d[x_0, x] = \frac{d(x) - d(x_0)}{x - x_0} & , k=1 \\ d[x_0, \dots, x_{k-1}, x] = \frac{d[x_0, \dots, x_{k-2}, x] - d[x_0, \dots, x_{k-1}]}{x - x_{k-1}} & , k \geq 2 \end{cases}$$

approx derivata di ordine k

Oss: per $k=1$, $d[x_0, x]$ è un rapporto incrementale. Per $k \geq 2$, $d[x_0, \dots, x_{k-1}, x]$ è un'approssimazione migliore della derivata di ordine k .

Proprietà:

(i) **simmetria**: per ogni permutazione (i_0, \dots, i_k) di $\{0, \dots, k\}$ si ha che

$$d[x_0, x_1, \dots, x_k] = d[x_{i_0}, x_{i_1}, \dots, x_{i_k}]$$

(ii) $d[x_0, \dots, x_{k-1}, x]$ non è definita se x_0, \dots, x_{k-1} perché il denominatore si annulla
Tuttavia $d \in C^k(\mathbb{I})$ con $\{x_0, \dots, x_k\} \subset \mathbb{I} \Rightarrow d$ **prolungabile** per continuità su tutto \mathbb{I}

(iii) Se $f \in C^k(\mathbb{I}) \Rightarrow \exists \varepsilon \in (\min x_i, \max x_i): d[x_0, \dots, x_k] = \frac{f^{(k)}(\varepsilon)}{k!}$

*conseguenza
del teorema
di Lagrange*

Come calcolare le diff. divise:

Sono $k+1$ p.t. di interpolazione. Si possono calcolare le differenze finite fino all'ordine k secondo la relazione (es $k=3$):

x	$f(x)$	DD1	DD2	DD3
x_0	$f(x_0)$			
x_1	$f(x_1)$	$d[x_0, x_1]$		
x_2	$f(x_2)$	$d[x_0, x_1]$	$d[x_0, x_1, x_2]$	
x_3	$f(x_3)$	$d[x_0, x_1]$	$d[x_0, x_1, x_2]$	$d[x_0, x_1, x_2, x_3]$

Per il polinomio di Newton risulta
la differenze diverse nella diagonale.

(ogni)
↓

$$A = \begin{bmatrix} f(x_0) & & & & \\ & f(x_1) & d[x_0, x_1] & & \\ & & d[x_0, x_1] & d[x_0, x_1, x_2] & \\ & & & d[x_0, x_1, x_2] & d[x_0, x_1, x_2, x_3] \\ & & & & d[x_0, x_1, x_2, x_3] \end{bmatrix}$$

$A = (a_{ij})$ triangolare superiore:

$$a_{ij} = \frac{a_{i, j-1} - a_{i-1, j-1}}{x_i - x_{j-1}} \quad \forall i, j$$

Esempio:

x	$f(x)$	DD1	DD2	DD3
0	3			
1	8	5		
2	15	6	1	
4	17	11	2	1/2

$$d[x_0, x_1, x_2] = \frac{d[x_0, x_1] - d[x_0, x_1]}{x_2 - x_1} = 1$$

$$d[x_0, x_1, x_2] = 1 = 2$$

$$d[x_0, x_1, x_2, x_3] = 1 = \frac{1}{2}$$

Il polinomio di Newton ha grado al più 3:

$$P_3(x) = 3 + 5(x-0) + 1(x-0)(x-1) + \frac{1}{2}(x-0)(x-1)(x-2)$$

\uparrow
 $x_0 \quad x_1 \quad x_2$
BASE DI NEWTON

$$= 3 + 5x + x^2 - x - \frac{1}{2}(x-1)(x-2)$$

$$= \frac{x^3}{2} - \frac{x^2}{2} + 5x + 3 \quad \leftarrow \begin{array}{l} \text{BASE DI} \\ \text{MONOMI} \end{array}$$

Oss: se una colonna del quadro delle differenze diverse è tutta uguale, gli elementi delle colonne successive sono tutti nulli e in particolare, il grado del polinomio di interpolazione è uguale all'ordine dell'ultima differenza diversa nelle diagonale $\neq 0$

Esempio:

x	$f(x)$	DD1	DD2	DD3	DD4
0	5				
-1	3	2			
2	3	-1	-1		
-2	-3	7	-7	1	
3	11	2	0	1	0

il polinomio interpolante ha grado 3

$$P_4(x) = 5 + 2(x-0) - 1(x-0)(x+1) + 1(x-0)(x+1)(x-2)$$

$$= x^3 - 2x^2 - x + 5 \quad \text{base monomiale}$$

Oss: date una tabella di valori $(x_0, y_0), \dots, (x_k, y_k)$ da interpolare, poniamo zordinarli come voci su nel quadro delle differenze finite.

Esempio:

x	$d(x) = y$
0	2
1	1
2	4
-1	$3d+1$
3	11

Se $\lambda \in \mathbb{R} \setminus \{0, 1, -1, 3\}$, determinare di che modo che il polinomio interpolante abbia grado minimo. Ricavare i nodi che non dipendono da λ per primi:

x	$d(x)$	DD1	DD2
0	2		
1	1	-1	
3	11	3	
-1	$3d+1$	$1-3d$	$\frac{3d-2}{2}$
λ	4	$2/d$	$\frac{2+d}{d(d-1)}$

Imponiamo $\begin{cases} \frac{3d-2}{2} = 2 \Rightarrow d=2 \\ \frac{2+d}{d(d-1)} = 2 \Rightarrow d=2 \end{cases}$ che calcola.

Quindi $d=2 \Rightarrow$ le entrate della col. DD2 sono uguali
 \Rightarrow polinomio di grado 2 (minimo)
 $P_4(x) = 2-x+2x(x-1) = 2x^2-3x+2$

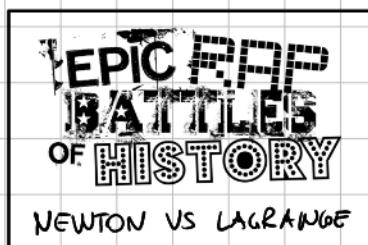
Errore nell'interpolazione polinomiale

Th: Siano $x_0, \dots, x_k \in \mathbb{R}$ con $x_i \neq x_j \forall i \neq j$. Sia $f: \mathbb{R} \rightarrow \mathbb{R}$: $f \in C^{k+1}(I)$ con $I \subseteq \mathbb{R}$, $\{x_0, \dots, x_k\} \subseteq I$. Se $P_k(x)$ è il polinomio di interpolazione di grado al più k per i punti $(x_0, f(x_0)), \dots, (x_k, f(x_k))$ allora

$$f(x) - P_k(x) = \pi(x) \frac{f^{(k+1)}(\xi)}{(k+1)!}$$

per un certo $\xi \in [\min\{x_0, \dots, x_k\}, \max\{x_0, \dots, x_k\}]$
con $\pi(x) = \prod_{s=0}^k (x - x_s)$

Oss: date una stima di $|f^{(k+1)}|$ e di $|\pi(x)|$ se può stimare l'errore del polinomio di interpolazione quando si vuole $P_k(x)$ fuori dai nodi $\{x_s\}_{s=0}^k$



	NEWTON	LAGRANGE
Base	$h_3(x) = (x - x_0)_{\text{III}}(x - x_{3-1})$	$l_3 = \prod_{i=0}^3 \frac{x - x_i}{x_3 - x_i}$
Coeff.	$\delta[x_0, \dots, x_3]$	$\delta(x_3) = y_3$
	Se valuto $P_k(x)$ e voglio $P_{k+1}(x)$ bisogna calcolare y_{k+1} e $\delta[x_0, \dots, x_k]$	Se voglio $P_{k+1}(x)$ devo ricalcolare tutti gli l_j
	Passo vedere facilmente il grado del polinomio	bisogna espanderlo nelle forme di monomi.



il meme del Marci

INTERPOLAZIONE OSCULATORIA DI HERMITE

Si considera anche la derivata di f nei nodi di interpolazione. Si può costruire un polinomio le cui valutazioni e quelle delle sue derivate equivalenti ai valori di f ed f' .

Esempio: conosciamo posizione e velocità di un oggetto in certi istanti di tempo e vogliamo approssimare la traiettoria.

Def: dati $x_0, \dots, x_r \in I \subseteq \mathbb{R}$ e $2k+2$ valori $\{f(x_0), \dots, f(x_r)\}$ e $\{f'(x_0), \dots, f'(x_r)\}$ di una funzione $f \in C^1(I)$. Si dice **POLINOMIO DI INTERPOLAZIONE DI HERMITE** per f ai nodi $\{x_0, \dots, x_r\}$ il polinomio $H_{2k+1}(x)$ di $\deg(H) \leq 2k+1$: $H_{2k+1}(x_i) = f(x_i)$, $H'_{2k+1}(x_i) = f'(x_i)$

$$(*) \quad H'_{2k+1}(x_i) = f'(x_i)$$

Th: se $x_i \neq x_j \forall i \neq j$ allora $\exists!$ il polinomio di Hermite H_{2k+1} per $f \in C^1(I)$

Dim: vedere le dispense (ho dim anche per la notizia di Vandermonde)

Come si costruisce H_{2k+1} ? Si potrebbe formular come un'intera lineare, no non conviene per gli stessi problemi di condizionamento del corso di ladermonde.

Si procede in maniera analoghe all'interpolazione di Lagrange, ma generalizzata per estendere le condizioni sulle derivate.

$$\text{Più precisamente cerchiamo un polinomio } H_{2k+1}(x) = \sum_{j=0}^K h_{0,j}(x) d(x_j) + \sum_{j=0}^K h_{1,j}(x) d'(x_j)$$

dove $h_{0,j}(x)$ e $h_{1,j}(x)$ polinomi di grado $2k+1$:

$$h_{0,j}(x_i) = \begin{cases} 1, & i=j \\ 0, & i \neq j \end{cases}, \quad h_{0,j}'(x_i) = 0; \quad h_{1,j}(x_i) = 0, \quad h_{1,j}'(x_i) = \begin{cases} 1, & i=j \\ 0, & i \neq j \end{cases} \quad \forall 0 \leq j \leq K \quad \text{☺}$$

Inoltre grante proprietà abbiamo che

$$H_{2k+1}(x_i) = \sum_{j=0}^K h_{0,j}(x_i) d(x_j) + \sum_{j=0}^K h_{1,j}(x_i) d'(x_j) = f(x_i) + 0 \quad \forall 0 \leq i \leq K$$

$$H_{2k+1}'(x_i) = \sum_{j=0}^K h_{0,j}'(x_i) d(x_j) + \sum_{j=0}^K h_{1,j}'(x_i) d'(x_j) = 0 + d'(x_i) \quad \forall 0 \leq i \leq K$$

Vogliamo rispettare tutte le condizioni di interpolazione (x)

Ora: come nell'interpolazione classica, stiamo riuscendo a cercare una rappresentazione del H_{2k+1} nelle basi dei monomi $\{1, x, \dots, x^{2k+1}\}$ e cerchiamo un'expressione rispetto alle basi $\{h_{0,0}(x), \dots, h_{0,K}(x), h_{1,0}(x), \dots, h_{1,K}(x)\}$ in cui i coefficienti delle combinazioni lineari sono $d(x_0), \dots, d(x_K), d'(x_0), \dots, d'(x_K)$.

Cerchiamo $h_{0,j}(x)$, $h_{1,j}(x)$ $\forall 0 \leq j \leq K$ che rispettano le proprietà ☺:

$$\text{Idee: } h_{0,j}(x) = (Ax+B) l_j^2(x),$$

$$h_{1,j}(x) = (Cx+D) l_j^2(x)$$

$$\text{dove } A, B, C, D \in \mathbb{R} \text{ sono costanti da determinare e } l_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^{K} \frac{x-x_i}{x_j-x_i}$$

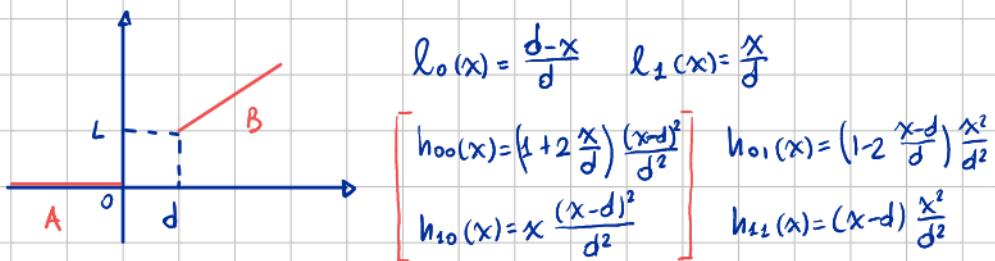
Imponendo ☺ e facendo i calcoli (si vedano le dispense) si ottiene che

$$h_{0,j}(x) = [1 - 2l_j'(x_j)(x-x_j)] l_j^2(x), \quad h_{1,j}(x) = (x-x_j) l_j^2(x)$$

$$\text{Quindi } H_{2k+1} = \sum_{j=0}^K d(x_j) [1 - 2l_j^2(x_j)(x-x_j)] l_j^2(x) + \sum_{j=0}^K d'(x_j) (x-x_j) l_j^2(x).$$

Esempio: supponiamo di avere due basi di funzioni e di volerli unire in un'unica C^1

$$\text{e.e. vogliamo } H_3(x) : H_3(0)=0, H_3(1)=L, \quad H_3'(0)=0, \quad H_3'(1)=S \quad (\text{coeff. angolare})$$



$$H_3(x) = 0 \cdot h_{00}(x) + L h_{01}(x) + 0 \cdot h_{10}(x) + S h_{11}(x) = L \left[1 - 2\frac{x-d}{d} \right] \frac{x^2}{d^2} + S (x-d) \frac{x^2}{d^2}$$

Si ne l'errore?

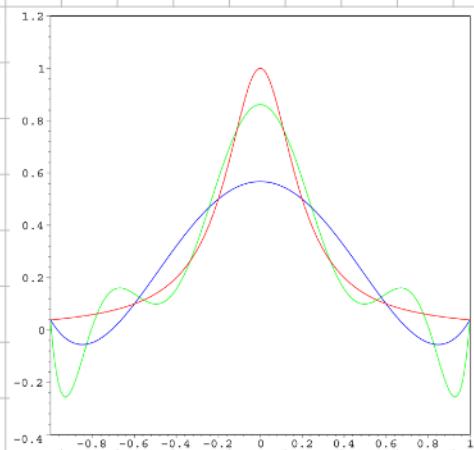
Th: sia $f \in C^{2k+2}(\mathbb{I})$, allora $f(x) - H_{2k+1}(x) = \pi^2(x) \frac{f^{(2k+2)}(s)}{(2k+2)!}$ per un certo
 $s \in [\min\{x_0, \dots, x_k, x\}, \max\{x_0, \dots, x_k, x\}]$ e
 con $\pi^2(x) = \prod_{s=0}^k (x - x_s)^2$

Nel contesto dell'interpolazione polinomiale, per diminuire l'errore $|f(x) - P_k(x)|$ bisogna aumentare il grado del polinomio interpolante. MA:

1. richiede di valutare f in più punti (non è detto neanche possibile)

2. talvolta, se ne fa fare così tanti non ne ha ottenuta un polinomio migliore

Esempio di Runge:



$$\blacksquare \quad d(x) = \frac{1}{1+x^2}$$

■ polinomio interpolante di grado k

■ polinomio interpolante di grado $k+1$

Nodi equispaziati $\Rightarrow \lim_{k \rightarrow +\infty} \max_{x \in [a,b]} |f(x) - P_k(x)| \neq 0$

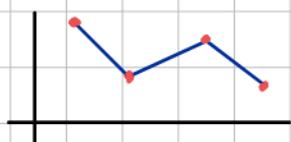
Si usano scelti particolari di nodi (ad esempio nodi di Chebyshev)

INTERPOLAZIONE POLINOMIALE A TRATTI

Si divide $[a, b]$ in sottointervalli delle forme $[x_{i-1}, x_i]$ (k sottointervalli). In ogni si considera il polinomio interpolante di f in x_{i-1}, x_i di grado $n < k$

$$p_i : [x_{i-1}, x_i] \rightarrow \mathbb{R} \quad p_i(x_{i-1}) = y_{i-1}, \quad p_i(x_i) = y_i$$

Interpolazione lineare a tratti:



$$\begin{cases} p_1(x) = m_1 x + n_1 \\ p_2(x) = m_2 x + n_2 \\ p_3(x) = m_3 x + n_3 \end{cases}$$

$$F_3(x) = \begin{cases} p_1(x), & x \in [x_0, x_1] \\ p_2(x), & x \in [x_1, x_2] \\ p_3(x), & x \in [x_2, x_3] \end{cases}$$

Funzioni spline:

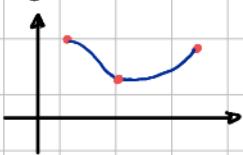
Per ottenere approssimazioni più regolari si usano polinomi di grado 3:

Def: dati $(x_i, y_i) \in \mathbb{R}^2$, $i = 0, \dots, K$, si definisce $S_3(x)$ a tratti come segue

i. $S_3(x)$ è un polinomio di grado 3 inizialmente a $[x_{i-1}, x_i]$

ii. $S_3(x_i) = y_i$, $i = 0, \dots, K$

iii. $S_3(x) \in C^2([a, b])$



per essere C^2 nelle $p_{i-1}'(x_i) = p_i'(x_i)$

$p_{i-1}''(x) = p_i''(x)$

Oss: $S_3(x)$ è definita da K polinomi $p_i(x)$, $i = 1, \dots, K$ t.c. $p_i : [x_{i-1}, x_i] \rightarrow \mathbb{R}$

$$p_i(x_i) = y_i, \quad i = 1, \dots, K$$

$$p_i(x_{i-1}) = y_{i-1}, \quad i = 1 \dots K$$

$$p_i'(x_i) = p_{i+1}'(x_i), \quad i = 1 \dots K-1$$

$$p_i''(x_i) = p_{i+1}''(x_i), \quad i = 1 \dots K-1$$

$\left. \begin{array}{l} 4K-2 \text{ equazioni, ogni } p_i(x) \text{ ha 4 coeff.} \\ \Rightarrow S_3(x) \text{ è determinata da } 4K \text{ parametri} \end{array} \right\}$

\Rightarrow abbiamo ancora 2 gradi di libertà. Per determinare le ultime 2 condizioni si può optare per:

• $p_i''(x_0) = p_K''(x_K) = 0$ SPLINE NATURALE

• $p_1'(x_0) = p_K'(x_K)$, $p_i''(x_0) = p_K''(x_K)$ SPLINE PERIODICA

• $p_i'(x_0) = f'(x_0)$, $p_K'(x_K) = f'(x_K)$ SPLINE VINCOLATA
se le corrono

Th: sia x_0, \dots, x_K nodi equipuntati su $[a, b]$ ($x_i = a + i h$, $h = \frac{b-a}{K}$, $i = 0, \dots, K$), $y_0, \dots, y_K \in \mathbb{R}$:

i) \exists ! spline naturale $S_3(x)$ per i pti (x_i, y_i) , $i = 0, \dots, K$

$$\text{ii) } p_i(x) = \left[y_{i-1} + \left(m_{i-1} + \frac{2y_i - y_{i-1}}{h} \right) (x - x_{i-1}) \right] \left(\frac{x - x_i}{h} \right)^2 + \left[y_i + \frac{m_i - 2y_i}{h} (x - x_i) \right] \left(\frac{x - x_{i-1}}{h} \right)^2$$

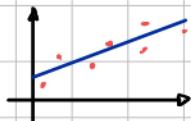
dove m_i sono le soluzioni di

$$\begin{bmatrix} 2 & 1 & & & \\ 1 & 4 & 1 & & \\ & 1 & 4 & 1 & \\ & & 1 & 2 & \\ & & & 1 & \end{bmatrix} \begin{bmatrix} m_0 \\ 1 \\ m_1 \\ \vdots \\ m_K \end{bmatrix} = \frac{3}{h} \begin{bmatrix} y_1 - y_0 \\ y_2 - y_0 \\ \vdots \\ y_{K-1} - y_{K-2} \\ y_K - y_{K-2} \end{bmatrix}$$

Dim (idea): su $[x_{i-1}, x_i]$ si considera il polinomio interpolante di 4 punti dove m_i rappresenta $f'(x_i)$ (che non si conosce). Si determinano gli m_i imponendo le condizioni sulle derivate 1° e 2° dei polinomi negli estremi □

Oss: il risultato si generalizza a nodi non equipuntati (ma distanti) e splines vincolate e periodiche

APPROXIMAZIONE AI MINIMI QUADRATI



Se i pti sono discollegati e insensibili tra loro polinomi interpo-

REGRESSIONE LINEARE: cerca le rette che passa più vicino dai punti.

Generalizziamo l'interpolazione polinomiale in 2 direzioni:

1. assumiamo che ci siano più p.t. che funzioni modelli (nel caso dell'interpolazione polinomiale t, x, x^2, \dots)

2. assumiamo l'utilizzo di funzioni modelli non polinomiali

$(x_0, y_0), \dots, (x_k, y_k)$ p.t. (con $y_i = f(x_i)$), $m+1$ funzioni modelli $g_0(x), \dots, g_m(x)$ con $m \leq k$.

Scriviamo $\Phi(x) \approx f(x)$: $\Phi(x) = \sum_{i=0}^m c_i g_i(x)$, $c_i \in \mathbb{R}$
da determinare

Esempio: $g_0(x) = 1$, $g_1(x) = x \Rightarrow \Phi(x) = c_0 + c_1 x$ regressione lineare

$$g_0(x) = e^{2x}, g_1(x) = \sin\left(\frac{x}{2}\right), g_2(x) = \frac{1}{3x^2+4} \Rightarrow \Phi(x) = c_0 e^{2x} + c_1 \sin\left(\frac{x}{2}\right) + \frac{c_2}{3x^2+4}$$

Come determinare c_i ? Dobbiamo definire cosa vuol dire $\Phi(x)$ passa "vicino ai p.t."

Una candidata funzione dell'errore può essere $\Psi(c_0, \dots, c_m) := \sum_{i=0}^k (\Phi(x_i) - y_i)^2$, ma non va bene perché nonostante + rottura + salto fare Φ ; $\Psi(c) \geq 0$ se punti molto distanti $\cancel{\cancel{}}$!

Vogliamo ricercare $\Psi(c) = \sum_{i=0}^k |\Phi(x_i) - y_i|$, peccato che non sia differentiabile. Come calcolare le minime?

Def: norma degli scarti quadratici $\Psi(c) := \sum_{i=0}^k (\Phi(x_i) - y_i)^2$

Il metodo dei minimi quadrati consiste nello scegliere $c = \begin{bmatrix} c_0 \\ \vdots \\ c_m \end{bmatrix} \in \mathbb{R}^{m+1}$ come il p.t. di minima di $\min_{c \in \mathbb{R}^{m+1}} \Psi(c)$.

$$\Psi(c) = \sum_{i=0}^k (\Phi(x_i) - y_i)^2 = \sum_{i=0}^k \left(\sum_{j=0}^m c_j g_j(x_i) - y_i \right)^2 = \sum_{i=0}^k ((Ac)_i - b_i)^2$$

$$A = \begin{bmatrix} g_0(x_0) & g_1(x_0) & \cdots & g_m(x_0) \\ \vdots & \vdots & \ddots & \vdots \\ g_0(x_k) & g_1(x_k) & \cdots & g_m(x_k) \end{bmatrix}, \quad c = \begin{bmatrix} c_0 \\ \vdots \\ c_m \end{bmatrix}, \quad b = \begin{bmatrix} y_0 \\ \vdots \\ y_k \end{bmatrix}$$

$$A \in \mathbb{R}^{(m+1) \times (m+1)}, \quad c \in \mathbb{R}^{m+1}, \quad b \in \mathbb{R}^{k+1}$$

\therefore minima le $\| \cdot \|_2$ del residuo di $Ac = b$, che è un sistema nonadeterminato (+ eq che incappa). Quindi ricercare soluzioni $\min_{c \in \mathbb{R}^{m+1}} \|Ac - b\|_2$ (metodo eq. normale o metodi QR)

Esempio:

x	y
1	1.4
2	3.1
3	4.8
4	6.8
5	8.5

cerchiamo la retta e la parabola che approssimano i p.ti nel senso
dei minimi quadrati.

Retta

$$E_1(x) = c_0 + c_1 x$$

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix} \quad b = \begin{bmatrix} 1.4 \\ 3.1 \\ 4.8 \\ 6.8 \\ 8.5 \end{bmatrix}$$

$$\min_c \|A_c - b\|_2 \rightarrow A^T A c = A^T b$$

$$A^T A = \begin{bmatrix} 5 & 15 \\ 15 & 35 \end{bmatrix}, \quad A^T b = \begin{bmatrix} 24.6 \\ 91.7 \end{bmatrix}, \quad \text{si risolve il sistema } 2x2 \quad \text{e si ottiene } c = \begin{bmatrix} 0.45 \\ 1.79 \end{bmatrix}$$
$$\Rightarrow \Phi_1(x) = 1.79x + 0.45$$

Parabola

$$G_0(x) = 1, \quad G_1(x) = x, \quad G_2(x) = x^2 \Rightarrow \Phi_2(x) = 0.0021x^2 + 1.66x + 0.45$$

INTEGRAZIONE NUMERICA

28/04/2025

Oggi $I(g \cdot d) = \int_a^b g(x) d(x) dx$, $d: C([a,b]) \rightarrow \mathbb{R}$

$g(x)$ si chiama **FUNZIONE PESO**. $g(x): [a,b] \rightarrow \mathbb{R}^+$ t.c. $M_n := \int_a^b x^k g(x) dx < +\infty$ $k=0,1,\dots$

MOMENTI DELLA FUNZIONE PESO

Perché approssimare $\int_a^b d(x) dx$ ($g=1$)?

i. spesso non si può trovare primitiva di d (e.g. funzioni ellittiche, più variabili)

ii. anche esistente, può essere costoso calcolarla e valutarla.

iii. peso corrisponde solo in dei punti. L'idea è cercare una formula

$$\int_a^b d(x) g(x) dx \approx \sum_{i=0}^n d(x_i) \cdot a_i \quad \text{PESI FORMULA DI QUADRATURA}$$

NODI

FORMULE DI QUADRATURA

Def: dati $[a,b]$, $g(x)$ definiamo $J_n(\cdot)$ formula di quadratura su $(n+1)$ nodi x_0, \dots, x_n con pesi a_0, \dots, a_n la funzione $J_n: C([a,b]) \rightarrow \mathbb{R}$, $d \mapsto \sum_{i=0}^n a_i d(x_i)$

Def: l'**errore** di una formula di quadratura $E_n(d) := \int_a^b d(x) g(x) dx - J_n(d) = \int_a^b d(x) g(x) dx - \sum_{i=0}^n a_i d(x_i)$. Anche $E_n: C([a,b]) \rightarrow \mathbb{R}$, $d \mapsto E_n(d)$

Oss: J_n, E_n sono funzioni lineari e.c. $J_n(c_1 d_1 + c_2 d_2) = \sum a_i (c_1 d_1 + c_2 d_2)(x_i) = \sum a_i (c_1 d_1(x_i) + c_2 d_2(x_i)) = \sum a_i c_1 d_1(x_i) + \sum a_i c_2 d_2(x_i) = c_1 \sum a_i d_1(x_i) + c_2 \sum a_i d_2(x_i)$

Quando una formula di quadratura è accurata?

Def: date J_n , si indica con $m \in \mathbb{N}$ il **GRADO DI PRECISIONE** come l'intero t.e. $E_n(1) = E_n(x) = \dots = E_n(x^m) = 0, E_n(x^{m+1}) \neq 0$

Oss: siccome E_n e J_n sono lineari, J_n ha grado di precisione m se J_n esatta per tutti i polinomi di grado $\leq m$

Esempio: $p=1$, $[a,b] = [-1,1]$, $n=1$, $x_0 = -1$, $x_1 = 1$

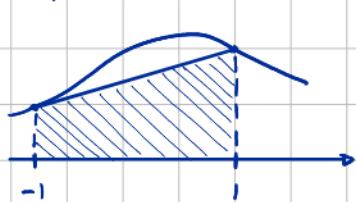
$J_1(d) = a_0 d(-1) + a_1 d(1) \approx \int_{-1}^1 d(x) dx$, qual'è la funzione reale per a_0, a_1 ? Scegliendoli in modo da massimizzare il grado di precisione di J_1 :

$$\begin{cases} E_n(1) = 0 \\ E_n(x) = 0 \end{cases} \Rightarrow \begin{cases} \int_{-1}^1 dx - (\alpha_0 + \alpha_1) = 0 \\ \int_{-1}^1 x dx - (-\alpha_0 + \alpha_1) = 0 \end{cases} \Rightarrow \begin{cases} 2 - \alpha_0 - \alpha_1 = 0 \\ 0 + \alpha_0 - \alpha_1 = 0 \end{cases} \Rightarrow \begin{cases} \alpha_0 = 1 \\ \alpha_0 = \alpha_1 \end{cases} \Rightarrow \alpha_0 = \alpha_1 = 1$$

$\underline{\underline{I(x)=0}}$

$\Rightarrow J_1(f) = f(-1) + f(1)$ il grado di precisione è ALMENO 1, vediamo se è anche di grado 2: $E_n(x^2) = \int_{-1}^1 x^2 dx - (1+1) = \frac{x^3}{3} \Big|_{-1}^1 - 2 = \frac{2}{3} - 2 \neq 0 \Rightarrow m=1$

Graphicamente:



$$(\text{base maggiore} + \text{minore}) \cdot \frac{\text{altezza}}{2} = [f(1) + f(-1)] \cdot \frac{2}{2} = f(1) + f(-1)$$

↑
FORMULA DEL TRAPEZIO

Oss: in un generico intervallo $[a, b]$ si può generalizzare le formule dei trapezi come $[f(a) + f(b)] \frac{b-a}{2}$. Ma possiamo aggiungere altri punti per raffinarle;

Esempio: aggiungiamo un altro nodo, ad esempio il p.t. medio dell'intervallo.

$x_0 = -1, x_1 = 0, x_2 = 1, J_2(f) = \alpha_0 f(-1) + \alpha_1 f(0) + \alpha_2 f(1)$, notiamo il grado:

$$\begin{cases} E_n(1) = 0 \\ E_n(x) = 0 \\ E_n(x^2) = 0 \end{cases} \Rightarrow \begin{cases} 2 = \alpha_0 + \alpha_1 + \alpha_2 \\ 0 = -\alpha_0 + \alpha_2 \\ \frac{2}{3} = \alpha_0 + \alpha_2 \end{cases} \Rightarrow \begin{cases} \alpha_0 = \alpha_2 = \frac{1}{3} \\ \alpha_1 = \frac{4}{3} \end{cases}$$

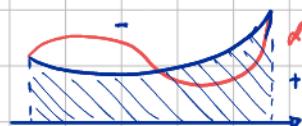
$$\Rightarrow J_2(f) = \frac{1}{3} (f(-1) + 4f(0) + f(1)) \approx \int_a^b f(x) dx \quad \text{FORMULA DI SIMPSON IN } [-1, 1]$$

Il grado è ALMENO 2, vediamo...

$$E_2(x^3) = \int_{-1}^1 x^3 dx - \frac{1}{3} (-1 + 4 \cdot 0 + 1) = 0 \Rightarrow m \geq 3$$

$$E_2(x^4) = \int_{-1}^1 x^4 dx - \frac{1}{3} (1 + 4 \cdot 0 + 1) = \frac{2}{5} - \frac{2}{3} \neq 0 \Rightarrow m=3$$

L'integrale è sotto, non l'interpolazione!



Oss: per un generico intervallo $[a, b]$ le formule di Simpson dritte

$$\frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right).$$

Quisamente si può generalizzare verso polinomi interpolanti di grado più alto (x si fa con più nodi)

Torniamo al problema di minimizzazione del grado di precisione dati x_0, \dots, x_n :

$$\begin{cases} E_n(1) = 0 \\ E_n(x) = 0 \\ \dots \\ E_n(x^n) = 0 \end{cases} \Rightarrow \begin{cases} M_0 = a_0 + \dots + a_n \\ M_1 = a_0 x_0 + \dots + a_n x_n \\ \dots \\ M_n = a_0 x_0^n + \dots + a_n x_n^n \end{cases} \Rightarrow \begin{bmatrix} 1 & \dots & 1 \\ x_0 & \dots & x_n \\ \vdots & & \vdots \\ x_0^n & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} M_0 \\ \vdots \\ M_n \end{bmatrix}$$

(n+1) incognite a_0, \dots, a_n in (n+1) eq. V

Sisteme lineari con matrice di Vandermonde $\Rightarrow \det V \neq 0$ se $x_i \neq x_j \forall i \neq j$

E.c. $\exists!$ formula di quadratura su x_0, \dots, x_n che ha grado di precisione $\geq n$.

Problema più difficile: cercare wie i nodi che i per un modo di minimizzare il grado di precisione. Le incognite sono $(2n+2)$: $x_0, \dots, x_n, a_0, \dots, a_n$

$$\begin{cases} E_n(1) = 0 \\ E_n(x) = 0 \\ \dots \\ E_n(x^{2n+1}) = 0 \end{cases} \Rightarrow \begin{cases} a_0 + a_1 + \dots + a_n = M_0 \\ a_0 x_0 + a_1 x_1 + \dots + a_n x_n = M_1 \\ \dots \\ a_0 x_0^{2n+1} + a_1 x_1^{2n+1} + \dots + a_n x_n^{2n+1} = M_{2n+1} \end{cases} \quad \text{SISTEMA DI EQ. NON LINEARI}$$

Th: Se $x_i \neq x_j \Rightarrow \exists!$ soluzione di $\text{SISTEMA DI EQ. NON LINEARI}$ per ogni scelta di $[a, b]$, n , $f(x)$.

In particolare, la soluzione fornisce una formula di quadratura con grado di precisione $\geq 2n+1$

Def: La formula di quadratura che minimizza il grado di precisione ottenibile con n+1 nodi si chiama **FORMULA GAUSSIANA** in $[a, b]$ con n+1 nodi.

Esempio: ($f=1$, $n=1$, $[a, b] = [-1, 1]$)

$$\int_a^b f(x) dx \approx a_0 f(x_0) + a_1 f(x_1), \quad a \leq x_0 < x_1 \leq b$$

$$\begin{cases} E_1(1) = 0 \\ E_1(x) = 0 \\ E_1(x^2) = 0 \\ E_1(x^3) = 0 \end{cases} \Rightarrow \begin{cases} a_0 + a_1 = 2 \\ a_0 x_0 + a_1 x_1 = 0 \\ a_0 x_0^2 + a_1 x_1^2 = 2/3 \\ a_0 x_0^3 + a_1 x_1^3 = 0 \end{cases} \Rightarrow \begin{aligned} a_0 &= a_1 = 1, \\ x_0 &= -\frac{\sqrt{3}}{3}, \quad x_1 = +\frac{\sqrt{3}}{3} \end{aligned}$$

$$S_1(f) = f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right) \quad \text{formula gaussiana per } n=1, f=1, [-1, 1] \\ E_1(x^4) = \frac{2}{5} - \left(\frac{9}{3^4} + \frac{9}{3^4}\right) = \frac{2}{5} - \frac{2}{3} \neq 0 \Rightarrow m=3$$

Ricopr.: $\int_a^b f(x) g(x) dx$ con $g(x)$ funzione pers., caso tipico $g(x) = 1$

Si approssima con $\sum_{i=0}^n a_i f(x_i)$, se possiamo scegliere i nodi le realte ottiene sono le domande per cui. Altrimenti scegli gli a_i in modo da minimizzare il grado di precisione.

ERRORE FORMULE DI QUADRATURA

Quello che ci chiediamo adesso è $E_n(f) = I(gf) - J_n(f) = ?$

Esercizio: $I(f) = \int_{-1}^1 |x| f(x) dx \approx a f(x_0) + b f(x_1)$.

Determina a, b, x_0, x_1 per maximizzare il grado di precisione di $J_1(\cdot)$ e determina.

Per un th. esiste sempre la soluzione, con $a, b \neq 0$, $x_1 \neq x_0$, $x_1, x_0 \in [a, b]$

$$\begin{aligned} \int_{-1}^1 |x| dx &= \int_{-1}^0 -x dx + \int_0^1 x dx = 1 = a+b \\ \int_{-1}^1 x |x| dx &= \int_{-1}^0 -x^2 dx + \int_0^1 x^2 dx = 0 = ax_0 + bx_1 \Rightarrow \begin{cases} a+b=1 \\ ax_0+bx_1=0 \\ ax_0^2+bx_1^2=1/2 \\ ax_0^3+bx_1^3=0 \end{cases} \\ \int_{-1}^1 x^2 |x| dx &= \int_{-1}^0 -x^3 dx + \int_0^1 x^3 dx = \frac{1}{2} = ax_0^2 + bx_1^2 \\ \int_{-1}^1 x^3 |x| dx &= \int_{-1}^0 -x^4 dx + \int_0^1 x^4 dx = 0 = ax_0^3 + bx_1^3 \end{aligned}$$

Moltiplico la 2° per $(-x_0^2)$ e la somma alle 4°: $b x_1^3 - b x_1 x_0^2 = 0$

$$\Rightarrow b x_1 (x_1^2 - x_0^2) = 0 \Rightarrow \begin{cases} b \neq 0 \\ x_1 \neq 0 \end{cases} \Rightarrow b x_1 (x_1 - x_0)(x_1 + x_0) = 0 \Rightarrow x_1 = 0 \text{ oppure } x_1 = -x_0$$

Sì può verificare che $x_1 = 0$ porta a un errore quindi $x_1 = -x_0$, e dalla 2° ottengo $a = b$. In particolare, dalla 1° ottengo $2a = 1 \Rightarrow a, b = \frac{1}{2}$. In fine, dalla 3° ho che $x_0^2 = \frac{1}{2} \Rightarrow x_0 = \pm \frac{1}{\sqrt{2}} = -\frac{1}{\sqrt{2}} \Rightarrow x_1 = -x_0 = \frac{1}{\sqrt{2}}$

$$J_1(f) = \frac{1}{2} f(-\frac{1}{\sqrt{2}}) + \frac{1}{2} f(\frac{1}{\sqrt{2}})$$

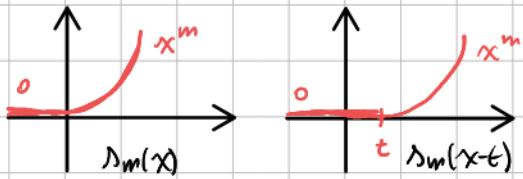
$$E_1(f) = \int_{-1}^1 x^4 |x| dx - J_1(f) = \int_{-1}^0 -x^5 dx + \int_0^1 x^5 dx - J_1(f) = \frac{1}{3} - \frac{1}{2} (\frac{1}{4} + \frac{1}{4}) = \frac{1}{3} - \frac{1}{4} \neq 0$$

\therefore il grado di precisione è $m=3$

Th. Peano: $f \in C^{m+1}([a, b])$ con $m = \deg(J_n(\cdot))$

$$\Rightarrow E_n(f) = \frac{1}{m!} \int_a^b f(t) G(t) dt, \text{ con } G(t) = E_n(\lambda_m(x-t)) \text{ nucleo di Peano.}$$

$$G(t) = \int_a^b \lambda_m(x-t) f(x) dx - \sum_{i=0}^n a_i \lambda_m(x-t), \text{ con } \lambda_m(x) = \begin{cases} x^m, & x \geq 0 \\ 0, & x < 0 \end{cases}$$



Pertanto $E_n(f) = \frac{1}{m!} \int_a^b d^{(m+1)}(t) G(t) dt$
richiede di calcolare l'integrale della
derivata (che dipende dalla funzione).

Però, se $G(t)$ non cambia segno in $[a, b]$, per il th. medie integrale

$$E_n(f) = \frac{1}{m!} d^{(m+1)}(\varepsilon) \int_a^b G(x) dx \quad \text{dove } \varepsilon \in [a, b] \quad (\text{meglio, ma c'è ancora } G(t))$$

$\int_a^b G(x) dx$ non dipende da d , né su valori ε per $d(x) = x^{m+1}$ ottengo
 $E_n(x^{m+1}) = \frac{1}{m!} (m+1)! \int_a^b G(t) g(x) dt \Rightarrow \int_a^b G(t) g(x) dt = \frac{E_n(x^{m+1})}{m!}$

Corollario: $d \in C^{m+1}([a, b])$, $G(t)$ non cambia segno in $[a, b]$

$$\Rightarrow E_n(d) = \frac{E_n(x^{m+1})}{m+1} \cdot d^{(m+1)}(\varepsilon), \quad \varepsilon \in [a, b]$$



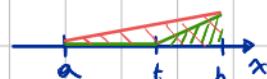
mette del 02/05/20

Ora: se voglio trovare una diseguaglianza del tipo $|E_n(f)| \leq M$, posso prendere
 $M = \max_{[a, b]} |d^{(m+1)}| \cdot \frac{E_n(x^{m+1})}{m+1}$ → le posso calcolare esplicitamente

le posso utilizzare coi metodi di studio della funzione o con maggiorazione esplicito

Ma l'ipotesi $G(t)$ non cambia segn è ragionevole?

Esempio: per la formula dei trapezi ($n=m=1$) la $\lambda_1(x-t)$



$$G(t) = \int_a^b \lambda_1(x-t) dx = \lambda_1(b-x),$$

$\lambda_1(b-x)$ non varia tempo $\int_a^b \lambda_1(x-t) dx \Rightarrow G(t)$ non cambia segn

$$G(t) = \int_{-1}^1 \lambda_1(x-t) dx - [\lambda_1(-1-t) + \lambda_1(1-t)], \quad t \in [-1, 1]$$

$$= \int_{-t}^1 \lambda_1(x-t) dx - [0 + 1-t] =$$

$$= \frac{(x-t)^2}{2} \Big|_{-t}^1 - (1-t) = \frac{(1-t)^2}{2} - 1+t$$

$$= \frac{1+t^2-2t-2+t^2}{2} = \frac{t^2-1}{2} \leq 0 \quad \forall t \in [-1, 1]$$

In generale, per la formula dei trapezi: $E_1(f) = \frac{f''(\varepsilon)}{2} E_1(x^2)$

$$E_1(x^2) = \int_a^b x^2 dx - \frac{b-a}{2} (a^2 + b^2) = \frac{b^3 - a^3}{3} \cdot \frac{b-a}{2} (a^2 + b^2) = \dots = -\frac{(b-a)^3}{6}$$

$$\therefore E_1(f) = -\frac{(b-a)^3}{12} f''(\xi), \quad \xi \in [a, b].$$

Ad es., nel caso $a=-1, b=1 \Rightarrow E_1(f) = -\frac{2}{3} f''(\xi)$

Esempio: anche per la formula di Simpson $G(\xi)$ non calcola bene in $[-1, 1]$

$$E_2(f) = \left(d^{(m)}(\xi) / 4! \right) \cdot E_2(x^4) = \frac{\lambda^{(m)}(\xi)}{24} \left(-\frac{4}{15} \right) = -\frac{1}{30} d^{(m)}(\xi), \quad \xi \in [a, b]$$

Si può fare lo stesso ragionamento per $[a, b]$

FORMULE DI QUADRATURA INTERPOLATORIE

$\int_a^b P_n(x) f(x) dx$ con $P_n(x)$ polinomio di grado $\leq n$ interpolante.

$$d(x) = P_n(x) + R_n(x), \text{ dove } R_n(x) = \frac{\pi(x)}{(m+1)!} d^{(m+1)}(\xi), \quad \xi \in [a, b]$$

$$\Rightarrow \int_a^b d(x) f(x) dx = \int_a^b P_n(x) f(x) dx + \int_a^b R_n(x) f(x) dx =$$

$$= \int_a^b \sum_{i=0}^n d(x_i) l_i(x) + \int_a^b R_n(x) f(x) dx = \sum_{i=0}^n d(x_i) \underbrace{\int_a^b l_i(x) f(x) dx}_{\text{node}} + E_n(f)$$

$P_n(x)$ nelle loro di
logranje, $l_i = \prod_{j \neq i} \frac{(x-x_j)}{(x_i-x_j)}$

i nodi di $I_n(f)$ sono esattamente x_i ($i=0, \dots, n$)
e i pesi sono $\int_a^b l_i(x) f(x) dx$

Oss: per costruzione, le formule di interpolazione hanno deg $\geq n$ (se i i
polinomi di deg $\leq n$, $P_n = d$, e quindi $\int_a^b P_n = \int_a^b d$)

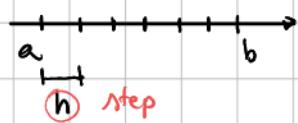


V.V., dati $n+1$ nodi c'è una sola formula di quadratura con deg $\geq n$ mentre
quei nodi e quelli deve essere le formule di quadratura interpolatorie
interpolazione \leftrightarrow integrazione

FORMULE NEWTON-COTES

($\xi=1$), si prendono nodi equispaziati in $[a, b]$. Quindi gli estremi (a, b) sono
inclusi in parola di formule di N.C. dunque, le consideriamo adess:

$$x_0 = a, x_1 = a + h, x_2 = a + 2h, \dots, x_r = a + rh, \dots, x_n = b$$



$$h = \frac{b-a}{n} \text{ passo della formula}$$

$$\text{perciò } \alpha_i = \int_a^b l_i(x) dx$$

Proprietà: per ogni n volte di n volte che $G(\epsilon)$ non cambia segno in $[a, b]$

$$\text{per cui vale per ogni formula n.c. } E_n(\lambda) = \frac{\lambda^{m+1}}{(m+1)!} \cdot E_n(x^{m+1})$$

Proprietà: per $n > 7$ (se has più di 8 nodi) le formule di n.c. hanno più di un segno alternato (per $n \leq 7$ sono sempre positivi) \rightarrow CANCELLAZIONE NUMERICA, volutamente le formule dicono instabile.

Per esprimere l'errore delle formule di n.c. ci serve trovare $E_n(x^{m+1})$. In questo assume un'espressione del tipo

$$E_n(\lambda) = \begin{cases} C_n \cdot h^{n+2} \cdot \lambda^{(n+1)}, & n \text{ pari} \\ C_n \cdot h^{n+3} \cdot \lambda^{(n+2)}, & n \text{ dispari} \end{cases}$$

$$\underline{\text{Esempio (trapezi)}}: n=1, h=b-a \Rightarrow -\frac{(b-a)^3}{12} \lambda''(\epsilon) = -\frac{h^3}{12} \lambda''(\epsilon)$$

$$\underline{\text{Esempio (Simpson)}}: n=2, h=\frac{b-a}{2} \Rightarrow -\frac{(b-a)^5}{2880} \cdot \lambda^{(IV)}(\epsilon) = -\frac{1}{90} h^5 \lambda^{(IV)}(\epsilon)$$

Oss: il grado di precisione è l'ordine delle potenze di h in lo stesso per una formula con grado pari n , e non quello con grado dispari. In genere, a parte i trapezi, si considerano più spesso le formule n.c. con n pari.

FORMULE GAUSSIANE

Proprietà: il nucleo di base delle formule Gaussiane non cambia segno, quindi formule dell'errore base dal corollario

Proprietà: i pesi α_i delle formule Gaussiane sono sempre positivi quindi non ci sono problemi di instabilità anche se n grande.

le formule Gaussiane sono cuiuscenti determinate rispetto $[a, b]$ e $f(x)$.

I nodi delle formule Gaussiane si possono scrivere come zeri di particolari polinomi, detti **polinomi ortogonali**.

Più precisamente, sono gli zeri di $\{q_n\}_{n \in \mathbb{N}}$, $\deg(q_n) = n$
 $\langle q_n, p \rangle = 0 \quad \forall p: \deg(p) < n$

$\langle q_n, p \rangle := \int_a^b q_n(x) p(x) f(x) dx$ prodotto scalare fra polinomi.

Esempi:

- $[a, b]$, $f(x) := 1 \rightarrow$ polinomi di LEGENDRE
- $[a, b]$, $f(x) = \frac{1}{\sqrt{1-x^2}} \rightarrow$ polinomi di CHEBYSHEV DI 1^o SPECIE

Ci sono metodi ad-hoc per trovare gli zeri dei polinomi ortogonali su $[-1, 1]$.

Poi puoi ottenerne quelli su $[a, b]$ con un cambio di variabili.

$$[-1, 1] \rightarrow [a, b]$$

$$x \rightarrow \frac{b-a}{2} + \frac{b+a}{2}$$

Esercizio esame: $\int_{-1}^1 d(x) dx$, $\delta_1 = a_0 d(-\frac{1}{2}) + a_1 d(\frac{2}{3})$,

determina a_0, a_1 , per uscire dall'intervallo $[a, b]$ e calcola $G(t)$ associato a $\delta_1(\cdot)$

$$E_1(1) = 0 \Rightarrow \int_{-1}^1 1 dx = 2 = a_0 + a_1 \Rightarrow \left\{ a_1 = \frac{6}{7} \right.$$

$$E_1(x) = 0 \Rightarrow \int_{-1}^1 x dx = 0 = -\frac{a_0}{2} + \frac{2}{3} a_1 \quad \left\{ a_0 = \frac{8}{7} \right.$$

$$\Rightarrow \delta_1(t) = \frac{8}{7} d(-\frac{1}{2}) + \frac{6}{7} d(\frac{2}{3})$$

$$\delta_1(t) = \frac{8}{7} d(-\frac{1}{2}) + \frac{6}{7} d(\frac{2}{3})$$

$$E_1(x^2) = \int_{-1}^1 x^2 dx - \left(\frac{8}{7} \frac{1}{4} + \frac{6}{7} \frac{4}{9} \right) = \frac{x^3}{3} \Big|_{-1}^1 - \left(\frac{1}{28} - \frac{8}{21} \right) = \frac{2}{3} - \frac{6+16}{21} = \frac{2}{3} - \frac{2}{3} = 0 \quad \text{ugue.}$$

$$E_1(x^3) = \int_{-1}^1 x^3 dx - \left(-\frac{8}{3} \frac{1}{8} + \frac{6}{7} \frac{8}{27} \right) = 0 - \left(-\frac{1}{3} + \frac{16}{63} \right) \neq 0 \Rightarrow \text{(m=2)}$$

$$G(t) = E_1(\lambda_2(x-t)) = \int_{-1}^1 \lambda_2(x-t) dx - \left(\frac{8}{7} \lambda_2(-\frac{1}{2}-t) + \frac{6}{7} (\lambda_2(\frac{2}{3}-t)) \right) = (*)$$

$$\text{con } \lambda_2(x-t) = \begin{cases} (x-t)^2, & x > t \\ 0, & x \leq t \end{cases} \quad (...) \quad \text{f}$$

$$(*) = \int_t^1 (x-t)^2 - (...) = \frac{(x-t)^3}{3} \Big|_t^1 - (...) = \frac{(1-t)^3}{3} - (...)$$

$$\frac{3}{7} \lambda_2 (-\frac{1}{2} - t) + \frac{6}{7} (\lambda_2 (\frac{2}{3} - t)) = \begin{cases} 0, & t \in [2/3, 1] \\ (2^*) , & t \in [-\frac{1}{2}, \frac{2}{3}] \\ 0, & t \in [\frac{2}{3}, 1] \end{cases}$$

-1 $-\frac{1}{2}$ $\frac{2}{3}$ 1

non c'è (1*)

non c'è (2*)

$$\Rightarrow G(t) = \begin{cases} \frac{(1-t)^3}{3} - (-\frac{1}{2} - t)^2 \frac{8}{3} - \frac{6}{7} (\frac{2}{3} - t)^2, & t \in [-1, -\frac{1}{2}] \\ \frac{(1-t)^3}{3} - \frac{6}{7} (\frac{2}{3} - t^2), & t \in [-\frac{1}{2}, \frac{2}{3}] \\ \frac{(1-t)^3}{2}, & t \in [\frac{2}{3}, 1] \end{cases}$$

CALCOLO DI AUTOVALORI E AUTOVETTORI

$A \in \mathbb{C}^{n \times n}$, vogliamo trovare le copie (λ, v) : $A v = \lambda v$

i) calcolare una (λ, v) , ad esempio $\rho(A)$ oppure $\sqrt{\rho(A^H A)}$

ii) calcola tutte le copie i.e. una coppia per ogni λ distinto

(i) Metodi delle potenze

Sia A diagonalizzabile, $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$

Dato che A è diagonalizzabile, allora \exists una base di vettori

$$\text{t.c. } A v^{(s)} = \lambda_s v^{(s)} \quad \forall s \in \{1, \dots, n\}, \quad c_1, \dots, c_n : z = \sum_{s=1}^n c_s v^{(s)}$$

$$\text{Idee: } z \in \mathbb{C}^n \Rightarrow A \cdot z = A \left(\sum c_s v^{(s)} \right) = \sum c_s A v^{(s)} = \sum c_s \lambda_s v^{(s)}$$

$$\Rightarrow A^2 \cdot z = A(Az) = \sum c_s \lambda_s A v^{(s)} = \sum c_s \lambda_s^2 v^{(s)}$$

$\Rightarrow A^k \cdot z = \sum c_s \lambda_s^k v^{(s)}$ dove $c_s \lambda_s^k v^{(s)}$ diventa dominante

Quindi sarebbe una buona idea $\begin{cases} z^{(0)} = z \\ z^{(n+1)} = A z^{(n)} = A^{n+1} z^{(0)} = A^{n+1} z \end{cases}$

Teorema: A Hermitiana con $|\lambda_1| > \dots \geq |\lambda_n| > 0$, $h \in \{1, \dots, n\}$ un indice t.c. $v_h^{(1)} \neq 0$. Sia $z^{(0)} : [v^{(1)}]^H z^{(0)} \neq 0$ i.e. non è ortogonale all'autovettore dominante.

Allora $\begin{cases} z^{(0)} \\ A z^{(k+1)} = A z^{(k)} \end{cases}$ è t.c. $\lim_{n \rightarrow \infty} \frac{z^{(k)}}{z_h^{(k)}} = \tilde{v}^{(1)}$ che è un

multiplo scalare di $v^{(1)}$, $\tilde{v}^{(1)} = \frac{v^{(1)}}{v_h^{(1)}}$

$$\lim_{n \rightarrow \infty} \frac{[z^{(k)}]^H A z^{(k)}}{[z^{(k)}]^H z^{(k)}} = \lambda_1$$

Oss: $R(x) = \frac{x^H A x}{x^H x}$ quoziente di Rayleigh. Se $x : Ax = \lambda x$

$$\text{allora } R(x) = \frac{x^H A x}{x^H x} = x \frac{x^H x}{x^H x} = \lambda$$

Recall th. spettrale: $A = A^H \Rightarrow A$ diagonalizzabile, \exists un base ortonormale di autovettori $v^{(1)}, \dots, v^{(n)} \in \mathbb{C}^n$ t.c. $A v^{(j)} = \lambda_j v^{(j)}$ e $[v^{(i)}]^H v^{(j)} = \langle v^{(i)}, v^{(j)} \rangle$

$$= \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

$$A = V D V^H, \quad D = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}, \quad V = \begin{bmatrix} v^{(1)} & \cdots & v^{(n)} \end{bmatrix}, \quad V^H V = I$$

Algorithm nell'implementazione $z^{(k)}$ viene salvato per le no norme per evitare underflow / overflow

$$\begin{aligned} z^{(0)} &\text{ dato in input} \\ k &= 1, \dots \\ y^{(k)} &= A z^{(k-1)} \\ z^{(k)} &= y^{(k)} / \|y^{(k)}\|_2 \end{aligned}$$

$$\begin{aligned} z^{(0)} &\text{ dato in input} \\ k &= 1, \dots \\ y^{(k)} &= A^{-1} z^{(k-1)} \quad \text{Inizializza } A \cdot y^{(k)} = z^{(k-1)} \\ z^{(k)} &= y^{(k)} / \|y^{(k)}\|_2 \quad \text{con Gauss} \\ \tilde{x}_n &= \left[\frac{[z^{(n)}]^H A z^{(n)}}{[z^{(n)}]^H z^{(n)}} \right]^{-1}, \quad \tilde{V}^{(n)} = z^{(n)} \quad \begin{array}{l} \text{METODO} \\ \text{POTENZE} \\ \text{INVERSE} \end{array} \end{aligned}$$

Ogni iterazione:

- matrice \times vettore: $\mathcal{O}(n^2)$
- $\| \cdot \|_2$ vettore: $\mathcal{O}(n)$

Se calcola $\mathbf{A}^{-1} \mathbf{b}$ prima del \mathbf{A}
 $\mathcal{O}(m^3) + \mathcal{O}(\# it \cdot n^2)$

Criterio di arresto: $\|R(z^{(k)}) - R(z^{(k-1)})\| < \text{TOL}$

Attenzione: $\|z^{(k)} - z^{(k+1)}\|_2 \leq \text{TOL}$ non è una buona idea perché l'autovettore non è univocamente definito ($z^{(k)}$ e $z^{(k+1)}$ sono a multipli diversi di $0^{(1)}$)

Oss:

- non è necessario che $A = A^H$, basta diagonalizzabile
- $[V^{(1)}]^H z^{(0)} \neq 0$ è vero con probabilità 1 se parts da un vettore casuale
- $|\lambda_1| > |\lambda_2|$ è importante, la successione non si scilla

Oss: per il metodo sarebbe utile voler calcolare il prodotto matrice · vettore $A \cdot z$.
Più essere utile se la matrice è definita come somma di generi.

Esempio: calcola $\|A\|_2 = \sqrt{\rho(A^H A)}$.

Applica metodi delle potenze in $A^H A$ che non mi chiede di calcolarli esplicitamente perché avere $(A^H A)z = A^H(Az) \Rightarrow$ 2 prodotti matrice · vettore

(i) Varianti del metodo

Vogli si autosalvo di modulo minimo $(\lambda_n, v^{(n)})$ e si l'unico $|\lambda_n| < \dots \leq |\lambda_1|$

Applica il metodo ad A^{-1} . Infatti, A^{-1} ha autosalvi λ_j^{-1} e quindi

$$\left| \frac{1}{\lambda_n} \right| > \dots \Rightarrow \left| \frac{1}{\lambda_1} \right| \text{ e le gli stessi } v^{(1)}, \dots, v^{(n)}$$

Le successioni in A^{-1} convergono a $(\frac{1}{\lambda_n}, v^{(n)})$

In generale, il metodo si può modificare per calcolare l'autosalvo con λ più vicino a $\sigma + i\mathbb{C}$. Basta applicare il metodo a $A = [A - \sigma \cdot I]^{-1}$, che ha gli stessi autovettori di A e come autosalvi $\frac{1}{\lambda_j - \sigma}$.

Quindi l'autosalvo di B con modulo più grande è quello che corrisponde a

$$\min_{j=1, \dots, n} |\lambda_j - \sigma|$$

Algorithm (potenze con shift):

?⁽⁰⁾ dato

$$A = \sigma \cdot I + \tilde{A}$$

$$\forall k = 1 \dots$$

$$\text{risolvere } (A - \sigma \cdot I) y^{(k)} = z^{(k-1)} \Rightarrow$$

$$z^{(k)} = y^{(k)} / \|y^{(k)}\|_2$$

$$\tilde{\lambda} = \frac{1}{[z^{(k)}]^H A z^{(k)}} + \sigma$$

$$\tilde{v} = z^{(k)}$$

II. Schemi per lo scritto

Richiami di Algebra Lineare

1. Norme

Si vuole definire la grandezza di un vettore. Formalmente si considerano funzioni con proprietà specifiche, chiamate norme.

Definizione (norma vettoriale):

$\|\cdot\|$ è una norma vettoriale se verifica:

- $\|x\| = 0 \Leftrightarrow x = \vec{0}$
- $\|\alpha x\| = \|\alpha\| \cdot \|x\| \quad \forall \alpha \in \mathbb{C}, x \in \mathbb{C}^n$
- $\|x + y\| \leq \|x\| + \|y\| \quad \forall x, y \in \mathbb{C}^n$ diseguaglianza triangolare

Esempio:

$$\|x\|_1 = \sum_{i=1}^n |x_i|, \quad \|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}, \quad \|x\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^2}$$

Teorema (norme equivalenti):

Date due norme $\|x\|_{n_1}, \|x\|_{n_2}$

$$\exists a, b \in \mathbb{R}^+ : a\|x\|_{n_1} \leq \|x\|_{n_2} \leq b\|x\|_{n_1}$$

Definizione (norma matriciale):

$\|\cdot\|$ è una norma matriciale se verifica:

- $\|A\| = 0 \Leftrightarrow A = [0]$
- $\|\alpha A\| = \|\alpha\| \cdot \|A\| \quad \forall \alpha \in \mathbb{C}, x \in \mathbb{C}^{m \times n}$
- $\|A + B\| \leq \|A\| + \|B\| \quad \forall A, B \in \mathbb{C}^{m \times n}$ diseguaglianza triangolare
- $m = n \Rightarrow \|A \cdot B\| \leq \|A\| \cdot \|B\| \quad \forall A, B \in \mathbb{C}^{n \times n}$ submoltiplicatività

Definizione (norma matriciale indotta):

Si parte da $\|\cdot\|_v$ norma vettoriale.

$$\text{Si dice che } \|\cdot\|_M \text{ è indotta se } \|A\|_M = \sup_{\|x\|_v \neq 0} \frac{\|Ax\|_v}{\|x\|_v} \equiv \sup_{\|x\|_v=1} \|Ax\|_v$$

Ovvero prendo tutti i vettori diversi da 0, guardo quanto A li allunga in norma. Normalizzo per la lunghezza del vettore così da considerare solo la proporzione di allungamento. Il massimo di questa proporzione è la norma matriciale.

Inoltre, se $\|A\|_M \geq \frac{\|Ax\|_v}{\|x\|_v}$, la norma matriciale si dice compatibile con quella vettoriale. Ad

esempio, le norme matriciali indotte sono compatibili con le norme vettoriali che le inducono.

Alcune norme matriciali indotte:

- $\|A\|_1 = \max \sum_{i=1}^n |a_{ij}|$ // massima somma delle colonne
- $\|A\|_\infty = \max \sum_{j=1}^n |a_{ij}|$ // massima somma delle righe
- $\|A\|_2 = \sqrt{\rho(A^H A)}$

Teorema (di Hirsch):

$$A \in \mathbb{C}^{n \times n} \Rightarrow \rho(A) \leq \|A\| \quad \forall \|.\|$$

dim

2. Teoremi di Gershgorin

Un disco di Gershgorin $\mathcal{F}_i(A) := \left\{ z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}| \right\}$

Teorema (I Gershgorin):

$$\lambda \in \bigcup_{i=1}^n \mathcal{F}_i(A) \quad \forall \lambda \in \sigma(A)$$

Corollario:

Se A è a predominanza diagonale forte, allora $\det(A) \neq 0$

dim

Teorema (II Gershgorin):

$$M_1 = \bigcup_{i=1}^s \mathcal{F}_i(A), \quad M_2 = \bigcup_{i=s+1}^{n-s} \mathcal{F}_i(A), \quad M_1 \cap M_2 = \emptyset$$

$$\Rightarrow \#\{M_1 \cap \sigma(A)\} = s, \quad \#\{M_2 \cap \sigma(A)\} = n - 2$$

Osservazioni:

- Se un autovalore complesso è in un cerchio, dev'esserci anche il suo coniugato
- Un cerchio isolato contiene necessariamente un autovalore reale
- Siccome gli autovalori di A sono gli stessi di A^T , allora posso prendere come regione l'intersezione $\{\bigcup \mathcal{F}_i(A)\} \cap \left\{ \bigcup \mathcal{F}_i(A^T) \right\}$, dove i cerchi calcolati sulla matrice trasposta hanno lo stesso centro, ma raggio calcolato sulle colonne (vabbe sticazzi)

dim

Teorema (III Gershgorin):

Se $A \in \mathbb{C}^{n \times n}$ irriducibile, allora se un autovalore sta sul bordo di un cerchio deve stare sul bordo di tutti i cerchi.

Corollario:

Se A è a predominanza diagonale debole e irriducibile, allora $\det(A) \neq 0$

dim

3. Matrici riducibili

Definizione (matrice riducibile):

$A \in \mathbb{C}^{n \times n}$ riducibile se $\exists \Pi$ matrice di permutazione tale che $\Pi A \Pi^T = \begin{bmatrix} A_1 & A_2 \\ & A_3 \end{bmatrix}$

Teorema (caratterizzazione di matrici riducibili):

A riducibile sse il suo grafo associato non è fortemente connesso

dim

Per trovare la matrice Π partiziono $N = P \cup Q$ dove P_i sono i vertici raggiungibili dal nodo i , Q_i quelli non raggiungibili. Prendo P_i , Q_i tali che $\forall j \neq i$, in P_j non ci sono i nodi contenuti in Q_i

Poi costruisco la partizione che parte (dall'alto o da sinistra) con gli indici in Q , poi con quelli in P

Sistemi lineari

1. Algoritmo di Gauss e simili

Algoritmo (Gauss classico):

$\forall k = 1, \dots, n-1$ //smaltimento gaussiano

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

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

$\forall j = k, \dots, n$

$$a_{ij} \leftarrow a_{ij} - l_{ik}a_{kj}$$

$$b_i \leftarrow b_i - l_{ik}b_k$$

$$x_n = \frac{b_n}{a_{nn}} \quad //\text{sostituzione all'indietro}$$

$\forall i = n-1, \dots, 1$

$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}}$$

Algoritmo (Gauss con pivoting parziale):

$\forall k = 1, \dots, n-1$ //smaltimento gaussiano

$$|a_{pk}| = \max_{i=k, \dots, n} |a_{ik}|$$

Scambio la riga k con la riga p sia in A che in b

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

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

$\forall j = k, \dots, n$

$$a_{ij} \leftarrow a_{ij} - l_{ik}a_{kj}$$

$$b_i \leftarrow b_i - l_{ik}b_k$$

$$x_n = \frac{b_n}{a_{nn}} \quad //\text{sostituzione all'indietro}$$

$\forall i = n-1, \dots, 1$

$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}}$$

Osservazioni:

- Se $a_{pk} = 0 \Leftrightarrow \det(A) = 0$
- Nella fattorizzazione LU devo scambiare anche le colonne di L

Algoritmo (Fattorizzazione LU):

$$L := I, U := A$$

$\forall k = 1, \dots, n-1$ //smaltimento gaussiano

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

$$l_{ik} = \frac{u_{ik}}{u_{kk}}$$

$$\forall j = k, \dots, n$$

$$u_{ij} \leftarrow u_{ij} - l_{ik}u_{kj}$$

$$b_i \leftarrow b_i - l_{ik}b_k$$

2. Condizionamento di sistemi lineari

Sia $\bar{x} + \delta\bar{x}$ la soluzione di $[A + \delta A][x + \delta x] = b + \delta b$, δ vettore di perturbazione

Teorema (limitatezza della perturbazione):

$$\text{Se } \|\delta A\| \cdot \|A^{-1}\| < 1 \Rightarrow \frac{\|\delta x\|}{\|x\|} \leq \frac{\mu(A)}{1 - \mu(A) \frac{\|\delta A\|}{\|A\|}} \left[\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right]$$

Dove $\mu(A) := \|A\| \cdot \|A^{-1}\|$ numero di condizionamento

Osservazione:

- $\|A\|_2 \cdot \|A^{-1}\|_2 = \rho(A) \cdot \rho(A^{-1}) = \frac{\lambda_{max}}{\lambda_{min}}$

Proposizione: se $\mu(A) \approx 1$ allora $\|A\bar{x} - b\| \approx \frac{\|x - \bar{x}\|}{\|x\|}$

3. Metodi iterativi

$$\lim_{x \rightarrow \infty} x_k = \bar{x} \text{ tale che } A\bar{x} = b$$

Metodi di punto fisso:

$$x = Hx + c$$

Teorema (convergenza dei metodi di punto fisso):

$$x_k = \begin{cases} x_0, & k = 0 \\ Hx_{k-1} + c, & k > 0 \end{cases} \text{ converge sse } \rho(H) < 1$$

dim

Corollario:

- $\|H\| < 1 \Rightarrow \rho(H) < 1$, ma $\|H\| > 1 \not\Rightarrow \rho(H) > 1$
- $|\det(H)| \geq 1 \Rightarrow \rho(H) > 1$, ma $|\det(H)| < 1 \not\Rightarrow \rho(H) < 1$

3.1. Metodo di Jacobi e Gauss-Seidel

$A = D - E - F$ dove D diagonale, E triangolare inferiore con zeri sulla diagonale, F diagonale superiore con zeri sulla diagonale.

$$H_J = D^{-1}(E + F), \quad c_J = D^{-1}b$$

$$H_{GS} = (D - E)^{-1}F, \quad c = (D - E)^{-1}b$$

Teorema (convergenza dei metodi di Jacobi e Gauss-Seidel):

Se A è a predominanza diagonale forte, oppure A è a predominanza diagonale debole e irriducibile, allora i metodi di Jacobi e Gauss-Seidel convergono per $Ax = b$

dim

il metodo di Gauss-Seidel è più efficiente e converge più rapidamente (o almeno non peggio) del metodo di Jacobi. Questo vantaggio si manifesta in due modi principali:

- Gauss Seidel ha un raggio spettrale più piccolo o al più uguale rispetto a quello di Jacobi. Questo perchè usa le componenti del vettore appena calcolate nello stesso passo iterativo. Questo significa che le informazioni più recenti (e presumibilmente più accurate) vengono propagate immediatamente nel calcolo delle successive componenti, avvicinandosi più rapidamente alla soluzione

4. Problema dei minimi quadrati

Se un sistema è sovradeterminato e $\text{rk}(A|b) \neq \text{rk}(A)$, esso non ha soluzione, e si considera il problema di minimizzare il residuo $\min \|Ax - b\|_2^2$

Teorema (sistema delle equazioni normali):

Si dimostra che la soluzione è data da $A^T Ax = A^T b$, e che essa esiste sempre. Inoltre, essa è unica se A ha rango massimo.

dim

4.1. Fattorizzazione QR

$A = QR$ dove Q unitaria, $R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$ con R_1 triangolare superiore.

Esistono infinite fattorizzazioni QR di A .

In questo modo:

$$\begin{aligned} \min \|Ax - b\|_2 &= \min \|QRx - b\|_2 = \min \|Rx - Q^H b\|_2 = \min \left\| \begin{bmatrix} R_1 x_1 \\ 0 x_2 \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \right\|_2 \\ &= \min \|R_1 x_1 - c_1\| + \|c_2\| = \min \|R_1 x_1 - c_1\| \end{aligned}$$

Siccome R_1 è quadrata e invertibile, allora il minimo si trova risolvendo il sistema triangolare

$$R_1x = c_1$$

Complettia computazionale:

1. Calcolo la fattorizzazione $A = QR$, $\mathcal{O}(m \cdot n^2)$
2. Calcolo $Q^H b = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$, $\mathcal{O}(m \cdot n)$
3. Risolvo $R_1 x = c_1$ sistema lineare triangolare, $\mathcal{O}(n^2)$

Costruire QR tramite riflessioni di Householder:

L'idea è azzerare, colonna per colonna, gli elementi sotto la diagonale premoltiplicando A per matrici unitarie che sono riflesse di Householder.

Dato $v \in \mathbb{C}^k$, esiste una matrice unitaria H unitaria con $Hv = ce_1$, $|c| = \|v\|_2$, dove

$$e_1 = [1 \ 0 \ \dots \ 0]^T$$

$$\text{Una scelta comune è } H = I - \frac{2\tilde{v}\tilde{v}^H}{\|\tilde{v}\|_2^2} \text{ dove } \tilde{v} = v \pm \|v\|_2 \cdot e_1 = \begin{bmatrix} v_1 \pm \|v\|_2 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

1. Per la prima colonna a_1 di A , costruisci H_1 tale che $H_1 a_1 = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ e applica H_1 a tutta

$$\text{la matrice: } H_1 \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix} = \begin{bmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & * & \dots & * \end{bmatrix}$$

2. Per la 2 colonna ignora la prima riga, costruisci H_2 e fai $H_2 H_1 A = \begin{bmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ 0 & 0 & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & * \end{bmatrix}$

$$3. \text{ Dopo } n \text{ passi: } H_n \cdots H_2 H_1 A = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = R$$

$$\text{Quindi } A = H_1^H H_2^H \cdots H_n^H R = QR$$

Interpolazione

1. Interpolazione polinomiale

Dato un insieme di punti e di valori $(x_i, f(x_i))$, costruisco un polinomio interpolante $p(x)$ tale che $p(x_i) = f(x_i) \forall i = 0, \dots, k$.

Proposizione: se ho $k + 1$ punti distinti, $\exists!$ polinomio interpolante di grado al più k .

1.1. Base dei monomi

$$Va = y \Leftrightarrow \begin{bmatrix} 1 & \dots & x_0^k \\ \vdots & \ddots & \vdots \\ 1 & \dots & x_k^k \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ | \\ a_k \end{bmatrix} = \begin{bmatrix} y_0 \\ | \\ y_k \end{bmatrix}$$

Risovo per a così che $p(x) = \sum a_i x^i$, costo computazionale $\mathcal{O}(n^3)$

Problema: $\mu(V) := \|V\| \cdot \|V^{-1}\|$ cresce esponenzialmente per x_i equispaziati, quindi l'errore sui coefficienti è grande, anche se il residuo è limitato.

1.2. Base di Lagrange

$$l_i(x_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

$$l_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} \implies V = \begin{bmatrix} l_0(x_0) & \dots & \overset{0}{l_k(\overbrace{x_0})} \\ \vdots & \ddots & \vdots \\ \overset{0}{l_0(\overbrace{x_k})} & \dots & l_k(x_k) \end{bmatrix} = I \implies Ia = y \implies p(x) = \sum y_i l_i(x)$$

Problema: se aggiungo un nodo x_{k+1} cambia tutto il polinomio interpolante.

1.3. Base di Newton

$$h_i(x) = \prod_{j=0}^{i-1} (x - x_j) \text{ che dipende solo dai primi } x_0, \dots, x_{i-1}$$

$$\text{In questo modo } V = \begin{bmatrix} * & & & \\ * & * & & \\ * & * & * & \end{bmatrix} \implies p(x) = \sum_{i=0}^k f[x_0, \dots, x_i] h_i(x)$$

Definizione (differenze divise):

$$\begin{cases} f[x_i] = y_i \\ f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i} \end{cases}$$

Algoritmo: come trovare le differenze divise:

Costruisci una matrice $A : a_{ij} = \frac{a_{i,j-1} - a_{j-1,j-1}}{x_i - x_{j-1}}$ $\forall i \geq j$

x	$f(x)$	DD_1	DD_2	DD_3
x_0	$f(x_0)$			
x_1	$f(x_1)$	$f[x_0, x_1]$		
x_2	$f(x_2)$	$f[x_0, x_2]$	$f[x_0, x_1, x_2]$	
x_3	$f(x_3)$	$f[x_0, x_3]$	$f[x_0, x_1, x_3]$	$f[x_0, x_1, x_2, x_3]$

Esempio:

$$x = (0, 1, 2, 4), f(x) = (3, 8, 15, 17)$$

x	$f(x)$	DD_1	DD_2	DD_3
0	3			
1	8	$\frac{8-3}{1-0} = 5$		
2	15	$\frac{15-3}{2-0} = 6$	$\frac{6-5}{2-1} = 1$	
4	17	$\frac{17-3}{4-0} = 4$	$\frac{3.5-5}{4-1} = -1$	$\frac{-0.5-1}{4-2} = -3$

Se una colonna del quadro delle differenze divise è tutta uguale, gli elementi delle colonne successive sono tutti nulli e, in particolare, il grado del polinomio di interpolazione è uguale all'ordine dell'ultima differenza divisa sulla diagonale diversa da zero.

Esempio:

$$x = (0, 1, \alpha, -1, 3), f(x) = (2, 1, 4, 3\alpha + 1, 11), \alpha \in \mathbb{R} \setminus \{0, 1, -1, 3\}$$

Determina α in modo da minimizzare il grado del polinomio interpolante.

Riarrango i nodi per mettere in fondo quelli che non dipendono da α .

x	$f(x)$	DD_1	DD_2	DD_3
0	2			
1	1	-1		
3	11	$\frac{11-2}{3} = 3$	$\frac{3+1}{3-1} = 2$	
-1	$3\alpha + 1$	$\frac{3\alpha + 1 - 2}{-1} = 1 - 3\alpha$	$\frac{1 - 3\alpha + 1}{-1 - 1} = \frac{3\alpha - 2}{2}$...
α	4	$\frac{2}{\alpha}$	$\frac{2/\alpha + 1}{\alpha - 1} = \frac{2 + \alpha}{\alpha(\alpha - 1)}$...

$$\text{Impongo} \begin{cases} \frac{3\alpha - 2}{2} = 2 \Rightarrow \alpha = 2 \\ \frac{2 + \alpha}{\alpha(\alpha - 1)} = 2 \Rightarrow \alpha = 2 \end{cases}$$

Quindi se $\alpha = 2$ le entrate della colonna DD_2 sono tutte uguali quindi il polinomio interpolante di minimo grado è di grado 2.

1.4. Interpolazione osculatoria di Hermite

Dati x_0, \dots, x_n e dati $f(x_i), f'(x_i) \forall i = 0, \dots, n$, cerchiamo un polinomio $H(x)$: $H(x_i) = f(x_i)$, $H'(x_i) = f'(x_i) \forall i = 0, \dots, n$.

Si dimostra che se $x_i \neq x_j \forall i \neq j$ allora esiste ed è unico il polinomio di interpolazione di Hermite. Inoltre, si ha che $\deg(H) \leq 2n + 1$.

$$\text{Cerchiamo } H_1(x) = H(x) + H'(x) = \sum_{j=0}^k f(x_j)h_{0j}(x) + \sum_{j=0}^k f'(x_j)h_{1j}(x)$$

Teorema:

$$\begin{cases} h_{0j}(x_i) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \\ h'_{0j}(x_i) = 0 \\ h_{1j}(x_i) = 0 \\ h'_{1j}(x_i) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \end{cases} \Rightarrow \begin{cases} H(x_i) = f(x_i) + 0 \\ H'(x_i) = 0 + f'(x_i) \end{cases} \Rightarrow H_1(x) = H(x_i) + H'(x_i)$$

Corollario:

Posso costruire h_{0j} e h_{1j} in questo modo:

- $h_{0j}(x) = (1 - 2l'_j(x_j)(x - x_j))l_j^2(x)$
- $h_{1j}(x) = (x - x_j)l_j^2(x)$

2. Interpolazione polinomiale a tratti

2.1. Lineare

$$p_i(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i) \Rightarrow p_i(x_i) = y_i, p_i(x_{i+1}) = y_{i+1}$$

2.2. Spline cubica

Definizione (spline cubica):

$S_3(x)$ è definita a tratti come segue:

- $S_3(x)$ è un polinomio di grado 3 se ristretta a $[x_{i-1}, x_i]$
- $S_3(x_i) = y_i \quad \forall i$
- $S_3(x) \in C^2([a, b])$

Per trovare la funzione spline interpolante si risolve il seguente sistema:

$$\begin{cases} p_i(x_i) = y_i, & i = 1, \dots, k \\ p_i(x_{i-1}) = y_{i-1}, & i = 1, \dots, k \\ p'_i(x_i) = p'_{i+1}(x_i), & i = 1, \dots, k-1 \\ p''_i(x_i) = p''_{i+1}(x_i), & i = 1, \dots, k-1 \end{cases}$$

Siccome ci sono ancora 2 gradi di libertà si aggiungono due condizioni tra le seguenti:

- $p''_1(x_0) = p''_k(x_k) = 0$ //spline naturale
- $p'_1(x_0) = p'_k(x_k), p''_1(x_0) = p''_k(x_k)$ //spline periodica
- $p'_1(x_0) = f'(x_0), p'_k(x_k) = f'(x_k)$ //spline vincolata, se conosco $f'(x_0), f'(x_k)$

3. Approssimazione ai minimi quadrati

Ammettiamo che si abbiano più punti che funzioni modello, e ammettiamo l'utilizzo di funzioni modello non polinomiali.

$$\Phi(x) = \sum_{i=0}^m c_i G_i(x) \text{ dove } G_i \text{ sono le funzioni modello e } c_i \text{ le costanti da determinare}$$

Si vuole trovare le costanti che minimizzano $\Psi(c) := \sum_{i=1}^k (\Phi(x_i) - y_i)^2$.

Si tratta di risolvere un problema ai minimi quadrati con il metodo delle equazioni normali o con il metodo QR.

$$\min_{c \in \mathbb{R}^{m+1}} \|Ac - b\|_2 \text{ dove } A = \begin{bmatrix} G_0(x_0) & \dots & G_m(x_0) \\ \vdots & \ddots & \vdots \\ G_0(x_k) & \dots & G_m(x_k) \end{bmatrix}, \quad c = \begin{bmatrix} c_0 \\ \vdots \\ c_m \end{bmatrix}, \quad b = \begin{bmatrix} y_0 \\ \vdots \\ y_k \end{bmatrix}$$

Integrazione

Approssima $I(\rho \cdot f) = \int_a^b f(x)\rho(x)dx$, $f \in C([a, b])$

$\rho(x)$ si chiama funzione peso, che fa pesare più o meno la funzione $f(x)$ nell'integrale.

$\rho : [a, b] \rightarrow \mathbb{R}^+$, $m_n := \int_a^b x^k \rho(x) dx < \infty \forall k = 0, 1, \dots$ momenti della funzione peso.

1. Formule di quadratura

Definizione (formula di quadratura ed errore):

$$\int_a^b f(x)\rho(x)dx \approx \sum_{i=0}^n f(x_i) \cdot a_i =: J_n(f)$$

L'errore è $E_n(f) := I(\rho f) - J_n(f)$.

Definizione (grado di precisione):

Il grado di precisione è l'intero d tale che $E_n(1) = \dots = E_n(x^d) = 0$, $E_n(x^{d+1}) \neq 0$.

In alternativa, $\sum_{i=0}^n a_i x_i^k = m_k \forall k = 0, 1, \dots, d$.

Si nota che la formula di quadratura e l'errore godono di linearità. Quindi, J_n ha grado di precisione d se e solo se integra esattamente tutti i polinomi di grado al più d .

- Formula dei trapezi: $J_1(f) = [f(a) + f(b)] \frac{b-a}{2}$ grado di precisione almeno 1.
 - Formula dei trapezi generalizzata: $J_n(f) = \frac{h}{2} \left(f(a) + 2 \sum_{k=1}^{n-1} f(a + kh) + f(b) \right)$,
 - L'errore $E_n(f) = -\frac{h^2(b-a)}{12} f''(\epsilon)$, dove $h = \frac{b-a}{n}$
- Formula di Simpson: $J_2(f) = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$ grado di prec. almeno 2.
 - Formula di Simpson generalizzata: $E_n(f) = -\frac{h^4(b-a)}{180} f^{(4)}(\epsilon)$

Massimizza il grado di precisione dati x_0, \dots, x_n . Quindi devo trovare i pesi a_0, \dots, a_n .

Imponiamo l'esattezza per $k = 0, \dots, n$:

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_0 & x_1 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_0^n & x_1^n & \dots & x_n^n \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_n \end{bmatrix} \Leftrightarrow Va = m$$

$\det(V) \neq 0$ sse tutti i nodi sono distinti. Ovvero esiste ed è unica formula di quadratura su

x_0, \dots, x_n che ha grado di precisione al più n .

Formula gaussiana:

formula di quadratura che massimizza il grado di precisione ottimizzando sia sugli x_i che sugli a_i , $(2n + 2)$ incognite.

Teorema:

Se i nodi sono distinti esiste ed è unica la soluzione per ogni scelta di $[a, b], n, \rho(x)$. In particolare, la soluzione fornisce una formula di quadratura con grado di precisione al più $2n + 1$

1.1. Errore formule di quadratura:

Teorema (Peano):

Se $J_n(f)$ è la formula di quadratura con grado di precisione m , allora l'errore si scrive come

$$E_n(f) := I(f) - J_n(f) = \frac{1}{m!} \int_a^b f(t)^{(m+1)} G(t) dt$$

Dove $G(t)$ è il nucleo di peano, costruito a partire da come la formula approssima i monomi.

$$G(t) = E_n(s_m(x-t)) = \int_a^b s_m(x-t) \rho(x) dx - \sum_{i=0}^n a_i s_m(x_i - t) \text{ con}$$

$$s_m(x) = \begin{cases} x^m, & x \geq 0 \\ 0, & x < 0 \end{cases}$$



Purtroppo il teorema ci richiede di calcolare l'integrale della derivata, che dipende dalla funzione.

Però, se $G(t)$ non cambia segno in $[a, b]$, per il teorema della media integrale si ha che:

$$E_n(f) = \frac{f^{(m+1)}(\epsilon)}{m!} \int_a^b G(t) dt \text{ dove } \epsilon \in [a, b].$$

Come calcolare $\int_a^b G(t) dt$?

Basta applicare la formula all'unico polinomio che sbaglia di poco, cioè $f(x) = x^{m+1}$

$$E_n(x^{m+1}) = \frac{1}{m!} (m+1)! \int_a^b G(t) dt \Rightarrow \int_a^b G(t) dt = \frac{E_n(x^{m+1})}{m+1}$$

Corollario del teorema di Peano:

Se f derivabile $m+1$ volte, $G(t)$ non cambia segno su $[a, b]$, allora

$$E_n(f) = \frac{E_n(x^{m+1})}{m+1} \cdot f^{(m+1)}(\epsilon), \quad \epsilon \in [a, b]$$

Se voglio trovare una disegualanza tipo $|E_n(f)| \leq M$ posso prendere

$$|E_n(f)| \leq \frac{|E_n(x^{m+1})|}{m+1} \cdot \max_{x \in [a,b]} |f(x)^{(m+1)}|$$

Il primo fattore dipende solo dalla formula di quadratura, calcolabile una volta per tutte. Il secondo fattore dipende dalla funzione f

Esempio: per trapezi $G(x)$ non cambia segno, posso calcolare l'errore:

$$E_n(x^{m+1}) = I(x^{m+1}) - J_n(x^{m+1}).$$

Per la formula dei trapezi il grado di precisione è $m = 1$, quindi calcolo $E_n(x^2)$

$$I(x^2) = \int_a^b x^2 dx = \frac{x^3}{3} \Big|_a^b = \frac{b^3 - a^3}{3}$$

$$J_1(f) = \frac{b-a}{2}(f(a) + f(b)) \Rightarrow J_1(x^2) = \frac{b-a}{2}(a^2 + b^2)$$

$$E_1(x^2) = \frac{b^3 - a^3}{3} - \frac{b-a}{2}(a^2 + b^2) = \dots = -\frac{(b-a)^3}{6}$$

$$\text{Dal corollario si ha che } E_1(f) = \frac{E_1(x^2)}{2} f''(\epsilon) = -\frac{(b-a)^3}{12} f''(\epsilon), \quad \epsilon \in [a, b]$$

1.2. Formule di quadratura interpolatorie

$\int_a^b P_n(x) \rho(x) dx$ con $P_n(x)$ polinomio di grado al più n interpolante.

$$f(x) = P_n(x) + R_n(x) \text{ dove } R_n(x) = \frac{\pi(x)}{(m+1)!} f^{(m+1)}(\epsilon), \quad \epsilon \in [a, b]$$

$$\begin{aligned} \Rightarrow \int_a^b f(x) \rho(x) dx &= \int_a^b P_n(x) \rho(x) dx + \int_a^b R_n(x) \rho(x) dx = \\ &= \int_a^b \sum_{i=0}^n f(x_i) l_i(x) dx + \int_a^b R_n(x) \rho(x) dx = \sum_{i=0}^n f(x_i) \int_a^b l_i(x) \rho(x) dx + E_n(f) \\ &\quad \text{nodi} \quad \text{pesi} \end{aligned}$$

Questo è proprio il prototipo di una formula di quadratura : i nodi sono esattamente x_i e i pesi sono $\int_a^b l_i(x) \rho(x) dx$.

Se f è un polinomio di grado al più n , allora $f = P_n$ e quindi l'integrale del polinomio interpolatorio e della funzione coincidono. Dunque, la formula interpolatoria è esatta per tutti i polinomi di grado al più n . Per polinomi di grado maggiore di n , l'errore $E_n(f)$ non è nullo in generale.

Dato un insieme di nodi x_0, \dots, x_n , esiste una sola formula di quadratura che usa esattamente quei nodi e ha grado di precisione almeno n . Questa è proprio la formula costruita tramite interpolazione. Infatti, il sistema dei momenti con quei nodi ammette soluzione unica, e i pesi che ottengo sono proprio $\int l_i(x) \rho(x) dx$

Interpolazione \leftrightarrow integrazione.

- L'interpolazione costruisce un polinomio che approssima f ai nodi
- L'integrazione di quel polinomio dà una formula di quadratura con stessi nodi.

1.3. Formule di Newton Cotes

$(n+1)$ nodi equispaziati su $[a, b]$.

$$x_i = a + ih, \quad h = \frac{b-a}{n}, \quad \forall i = 0, \dots, n$$

I pesi sono $a_i = \int_a^b l_i(x) dx$

Per ogni scelta di n vale che $G(t)$ non cambia sgn in $[a, b]$, quindi $E_n(f) = \frac{E_n(x^{m+1})}{(m+1)!} f^{(m+1)}(\epsilon)$.

Inoltre, se ho più di 8 nodi ($n > 7$) le formule hanno più a_i con segno alternato (per $n \leq 7$ sono sempre positivi) quindi è possibile avere cancellazione numerica, per cui valutare le formule diventa instabile.

Torniamo all'errore: $E_n(f) = \begin{cases} c_n \cdot h^{n+2} \cdot f^{(n+1)}(\epsilon), & n \text{ pari} \\ c_n \cdot h^{n+3} \cdot f^{(n+2)}(\epsilon), & n \text{ dispari} \end{cases}$

Esempio:

- Trapezi: $n = 1, h = b - a \Rightarrow E_1(f) = -\frac{(b-a)^3}{12} f^{(2)}(\epsilon)$
- Simpson: $n = 2, h = \frac{b-a}{2} \Rightarrow E_2(f) = -\frac{1}{90} \left(\frac{b-a}{2}\right)^5 f^{(4)}(\epsilon)$

Il grado di precisione e l'ordine della potenza di h è lo stesso per una formula con grado pari a n , e mai quella con grado dispari. In genere, a parte i trapezi, si considerano più spesso le formule con n pari.

1.4. Formule Gaussiane

Il nucleo di Peano delle formule gaussiane non cambia segno, quindi la formula dell'errore è data dal corollario. I pesi α_i delle formule gaussiane sono sempre positivi quindi non ci sono problemi di instabilità numerica.

Le formule gaussiane sono univocamente determinate fissato $[a, b]$ e $\rho(x)$.

Sia $\{q_n\}$ la famiglia dei polinomi ortogonali rispetto al prodotto scalare

$$\langle q_n, p \rangle := \int_a^b q_n(x) p(x) \rho(x) dx,$$

Il nodo x_i è esattamente uno degli zeri del polinomio $q_{i+1}(x)$. I nodi sono tutti distinti e reali.

Esempi:

- $[a, b], \rho(x) = 1$ polinomi di Legendre
- $[a, b], \rho(x) = \frac{1}{\sqrt{1-x^2}}$ polinomi di Chebyshev di prima specie.

Ci sono metodi ad-hoc per trovare gli zeri di polinomi ortogonali su $[-1, 1]$. Poi si possono ottenere quelli su $[a, b]$ con un cambio variabili: $[1, 1] \rightarrow [a, b] \implies x \rightarrow \frac{b-a}{2} + \frac{b+a}{2}$

Equazioni non lineari

1. Caso scalare

$f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = 0$.

Cerchiamo $\{x_n\}_{n \in \mathbb{N}}$ tale che $\lim_{n \rightarrow \infty} x_n = \alpha$ radice di f .

I metodi sono della forma $x_{n+1} = \phi_n(x_n, \dots, x_{n-k+1})$

Se la funzione di iterazione è costante $\phi_n = \phi \forall n$ si dice che il metodo è stazionario.

Definizione (ordine di convergenza di un metodo scalare):

Un metodo converge ad α se $\lim_{n \rightarrow \infty} |x^n - \alpha| = 0$

Un metodo ha convergenza di ordine $p \geq 1$ se $\exists c \neq 0$, $c < \infty$ tale che $\lim_{n \rightarrow \infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|^p} = c$

Se $p = 1$ (convergenza lineare, l'errore si riduce di un fattore c ogni passo), serve anche $0 < c < 1$

Definizione (convergenza locale):

Un metodo converge localmente se $\exists S \subseteq R : \alpha \in S$ tale che per ogni scelta x_0, \dots, x_{1-k} in S si ha che x_n converge ad α .

In parole povere, se parto vicino che le cose funzionano. Per sistemi lineari si fa sempre convergenza globale, adesso è un casino.

Come localizzare radici di una funzione:

- se $f(x)$ è continua e $f(a)f(b) < 0 \implies \exists \alpha \in [a, b] : f(\alpha) = 0$
- studio della derivata
- separazione grafica

1.1. Metodi stazionari a due punti

1.1.1. Ricerca binaria

Questo metodo fa cagare, supponiamo che ci sia una sola radice in $[a, b]$, ad ogni passo al caso pessimo l'errore si dimezza (asintoticamente).

$|x_{n+1} - \alpha| \leq \frac{b-a}{2^n} \implies \lim_{n \rightarrow \infty} |x_n - \alpha| = 0$ quindi converge.

Ordine di convergenza $\lim_{n \rightarrow \infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|} = \frac{1}{2}$ fa schifo.

$$\text{Tolleranza } |x_n - \alpha| < \beta \implies \frac{b-a}{2^n} \leq \beta \implies n \geq \log_2 \left(\frac{b-a}{\beta} \right) \implies n = \left\lceil \log_2 \left(\frac{b-a}{\beta} \right) \right\rceil$$

1.1.2. Metodo delle secanti

Cerco l'intersezione con l'asse delle x della retta che passa per $(x_n, f(x_n))$ e $(x_{n-1}, f(x_{n-1}))$

$$\therefore x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

Teorema: se $f \in C^2([a, b])$ allora il metodo converge localmente con $p = \frac{1 + \sqrt{5}}{2} > 1$

1.2. Metodi stazionari a un punto (metodi di punto fisso)

$\begin{cases} x_0 \text{ dato} \\ x_{n+1} = \phi(x_n) \end{cases}$ tale che ϕ è una funzione che ha un punto fisso per ogni radice di f
ovvero $\phi(\alpha) = \alpha \quad \forall \alpha : f(\alpha) = 0$

Il modo canonico per costruirla è $\phi(x) = x - g(x)f(x)$ dove $g(x) \neq 0$ vicino ad α .

Così, per costruzione $\phi(\alpha) = \alpha - g(\alpha)f(\alpha) = \alpha$.
 $\underset{=0}{=}$

La condizione $g(x) \neq 0$ serve a garantire l'unicità quindi la condizione necessaria e sufficiente:
 $\phi(\alpha) = \alpha \Leftrightarrow f(\alpha) = 0$.

Teorema (convergenza locale):

Sia ϕ derivabile, α punto fisso di $\phi(x)$.

$\exists \rho > 0, k \in (0, 1) : |\phi'(x)| \leq k \quad \forall x \in \mathcal{B}_{\alpha, \rho} \implies$

- $x_n \in \mathcal{B}_{\alpha, \rho} \quad \forall x_0 \in \mathcal{B}_{\alpha, \rho}$
- $\lim_{n \rightarrow \infty} x_n = \alpha \quad \forall x_0 \in \mathcal{B}_{\alpha, \rho}$
- α è l'unico punto fisso di $\phi(x)$ in $\mathcal{B}_{\alpha, \rho}$

dim

Corollario:

Se $|\phi'(\alpha)| < 1$ il metodo converge localmente.

$|\phi'(x)| \leq k$ significa che ϕ è contrattiva, ovvero è meno pendente di $f(x) = x$.

Convergenza monotona e alternata:

- Se $\phi'(x) > 0$ convergenza monotona poiché $\operatorname{sgn}(x_{n+1} - \alpha) = \operatorname{sgn}(x_n - \alpha)$
- Se $\phi'(x) < 0$ convergenza alternata poiché $\operatorname{sgn}(x_{n+1} - \alpha) = -\operatorname{sgn}(x_n - \alpha)$

Teorema (ordine di convergenza):

Sia ϕ derivabile p volte, α punto fisso di $\phi(x)$.

$$\phi'(\alpha) = \dots = \phi^{(p-1)}(\alpha) = 0, \quad \phi^{(p)}(\alpha) \neq 0, \quad p \in \mathbb{Z}$$

\implies la successione $\phi(x_n)$ converge con ordine p ad α

dim

1.2.1. Metodo di Newton (metodo delle tangenti)

Funzione di iterazione: $\phi(x) = x - \frac{f(x)}{f'(x)}$

$$\begin{aligned} \text{Metodo: } & \begin{cases} x_0 \text{ dato} \\ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \end{cases} \end{aligned}$$

Interpretazione geometrica:

Considero la tangente al grafico di $f(x)$ nel punto $(x_0, f(x_0))$. L'intersezione con l'asse x fornisce il punto successivo x_1 . Iterando il procedimento, si ottiene una successione $\{x_n\}$ convergente sotto opportune ipotesi a una radice di $f(x)$.

Il metodo di Newton è lento per $f(x) \approx x^n$, $n \gg 1$.

Definizione (ordine di una radice):

Una radice \tilde{x} è di ordine $r > 0$ se $\lim_{x \rightarrow \infty} \frac{f(x)}{(x - \tilde{x})^r} = c \neq 0$, $c < \infty$

Equivalentemente, se $f(\tilde{x}) = f'(\tilde{x}) = \dots = f^{(r-1)}(\tilde{x}) = 0$, $f^{(r)}(\tilde{x}) \neq 0$.

Teorema (convergenza del metodo di Newton per radici semplici):

Il metodo di Newton converge localmente con ordine $p \geq 2$ se α è una radice semplice. In

particolare, si ha che $\lim_{n \rightarrow \infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|^2} = \frac{1}{2} \frac{f''(\alpha)}{f'(\alpha)}$.

- Se $f''(\alpha) = 0$ converge con ordine $p > 2$,
- Se $f''(\alpha) \neq 0$ converge con ordine $p = 2$.

dim

Il metodo di Newton converge localmente con ordine $p = 1$ se α è una radice multipla.

Questo vuol dire che si può guardare direttamente f e le sue derivate in α per studiare la convergenza del metodo di Newton.

Teorema (convergenza globale):

Un metodo converge globalmente se $|\phi'(x)| < 1$ e $\phi(x) \in [a, b] \forall x \in [a, b]$

Corollario per Newton:

Se $\alpha \in [a, b]$ radice semplice di f , f' e f'' hanno segno costante su $[a, b]$ allora Newton converge con ordine $p \geq 2$ per ogni $x_0 : f(x_0) \cdot f''(x_0) > 0$

- $f' > 0, f'' > 0$ convessa crescente, converge per $f(x_0) > 0$
- $f' < 0, f'' > 0$ convessa decrescente, converge per $f(x_0) > 0$
- $f' > 0, f'' < 0$ concava crescente, converge per $f(x_0) < 0$
- $f' < 0, f'' < 0$ concava decrescente, converge per $f(x_0) < 0$

2. Caso vettoriale

$f(x) = 0, f: \mathbb{R}^m \rightarrow \mathbb{R}^m$, troviamo $\alpha \in \mathbb{R}^m : f(\alpha) = \vec{0}$

2.1. Metodi stazionari a un punto (metodi di punto fisso)

$$\begin{cases} x^{(0)} \text{ dato} \\ x^{(n+1)} = \phi(x_n) \end{cases}$$

Dove $\phi(x_n) = x - G(x_n)f(x_n)$ tale che $\det(G(\alpha)) \neq 0$

Definizione (ordine di convergenza di un metodo vettoriale):

Un metodo converge ad α se $\lim_{n \rightarrow \infty} \|x^{(n)} - \alpha\|_2 = 0$

Converge ad α con ordine $p \geq 1$ se $\lim_{n \rightarrow \infty} \frac{\|x_{n+1} - \alpha\|_2}{\|x_n - \alpha\|_2^p} = c$

Se $p = 1$ serve anche $0 < c < 1$

Definisco al volo il Jacobiano di una funzione $\phi(x)$:

$$J_\phi(x) := \begin{bmatrix} \frac{\partial \phi_1}{\partial x_1} & \cdots & \frac{\partial \phi_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial \phi_m}{\partial x_1} & \cdots & \frac{\partial \phi_m}{\partial x_m} \end{bmatrix} = [\nabla \phi(x_1) \mid \dots \mid \nabla \phi(x_m)]$$

Teorema (convergenza locale):

Sia $\phi \in C^1(\Omega)$, $\Omega \subseteq \mathbb{R}^m$, $\alpha \in \Omega$ punto fisso di $\phi(x)$.

Se $\rho(J_\phi(\alpha)) < 1$ il metodo converge localmente

Inoltre, se per qualsiasi norma vale $\|J_\phi(\alpha)\| < 1$ allora il metodo converge localmente.

Se $\exists \|J_\phi(\alpha)\| \geq 1$ non si può dire nulla.

2.1.1. Metodo di Newton-Rhapson

Funzione di iterazione: $\phi(x) = x - J_f^{-1}(x)f(x)$

Metodo: $\begin{cases} x_0 \text{ dato} \\ x_{n+1} = x_n - J_f^{-1}(x_n)f(x_n) \end{cases}$

Osservazione sulla complessità del metodo:

- Ad ogni passo valuto $J_f(x_n)$, quindi n^2 funzioni nonlineari
- Devo risolvere $J_f(x_n) = f(x_n)$, quindi $\mathcal{O}(m^3)$

Teorema (convergenza del metodo di Newton-Rhapson):

Se $J_f(\alpha)$ è nonsingolare allora $\exists S \subseteq \Omega$ intorno di α tale che $\forall x_0 \in S$ il metodo converge in maniera superlineare ($p \geq 2$).

Questo vuol dire che si può guardare direttamente f e il suo Jacobiano α per studiare la convergenza del metodo di Newton-Rhapson

Calcolo di autovalori e autovettori

$A \in \mathbb{C}^{n \times n}$, voglio trovare $(\lambda, v) : Av = \lambda v$

1. Metodo delle potenze

Calcolo una sola (λ, v) , ad esempio $\rho(A)$, oppure $\|A\|_2 = \sqrt{\rho(A^H A)}$.

Teorema:

Sia A diagonalizzabile, siano $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$, e sia $h \in \{1, \dots, n\}$ un indice tale per cui l'autovettore associato a λ_1 ha componente h nonzero, i.e. $v_h^{(1)} \neq 0$. Sia inoltre $z \in \mathbb{C}$ che non sia ortogonale a λ_1 i.e. $[v^{(1)}]^H z^{(0)} \neq 0$.

Allora $\begin{cases} z^{(0)} = z \\ z^{(k+1)} = Az^{(k)} \end{cases}$ è tale che $\lim_{n \rightarrow \infty} \frac{z^{(k)}}{z_h^{(k)}} = \tilde{v}^{(1)} = \frac{v^{(1)}}{v_h^{(1)}}$

Ed inoltre $\lim_{n \rightarrow \infty} \frac{[z^{(k)}]^H Az^{(k)}}{[z^{(k)}]^H z^{(k)}} = \lambda_1$

Dove $R(x) := \frac{x^H Ax}{x^H x}$ quoquente di Rayleigh

Condizioni per la convergenza:

- $|\lambda_1| > |\lambda_2|$ per convergenza lineare verso $v^{(1)}$. Più piccolo è il rapporto $\left| \frac{\lambda_2}{\lambda_1} \right|$ più veloce è la convergenza
- Il vettore iniziale $z^{(0)}$ non deve avere comp. nulla in direzione di $v^{(1)}$ i.e. $[v^{(1)}]^H z^{(0)} \neq 0$

Nella pratica $z^{(k)}$ viene diviso per la sua norma per evitare underflow o overflow.

Algoritmo (metodo delle potenze):

$z^0 \leftarrow$ input (non ortogonale a $v^{(1)}$)

$\forall k = 1, \dots$

$$y^{(k)} = Az^{(k-1)} \quad // \mathcal{O}(n^2)$$

$$z^{(k)} = \frac{y^{(k)}}{\|y^{(k)}\|_2} \quad // \mathcal{O}(n)$$

Stima dell'autovalore dominante: $\lambda_k = R(z^{(k)})$

Se $|\lambda_k - \lambda_{k-1}| < \text{TOL} \Rightarrow \text{STOP}$

$$\tilde{\lambda}_1 = R(z^{(n)}) = \frac{[z^{(n)}]^H A z^{(n)}}{[z^{(n)}]^H z^{(n)}}$$

$$\tilde{v}^{(n)} = z^{(n)}$$

Complessità: $\mathcal{O}(\#it \cdot n^2)$

Si osserva che per il metodo serve saper valutare solo il prodotto matrice-vettore $A \cdot z$.

Può essere utile se la matrice è definita come risultato di operazioni, ad esempio $\|A\|_2 = \sqrt{\rho(A^H A)}$

Applico il metodo su $A^H A$ che non mi chiede di calcolarlo esplicitamente.

Posso fare $(A^H A)z = A^H(Az)$ che mi richiede 2 prodotti matrice-vettore.

1.1. Metodo delle potenze inverse

Se voglio $(\lambda_n, v^{(n)}) : |\lambda_n| < |\lambda_{n-1}| \leq \dots \leq |\lambda_1|$, basta applicare il metodo ad A^{-1} .

Infatti A ha autovalori λ_j^{-1} e si ha che $\left| \frac{1}{\lambda_n} \right| > \dots \geq \left| \frac{1}{\lambda_1} \right|$ con stessi autovettori.

La successione convergerà all'autocoppia $\left(\frac{1}{\lambda_n}, v^{(n)} \right)$.

Algoritmo (metodo delle potenze inverse):

$z^0 \leftarrow \text{input}$

$A = \Pi \cdot LU$ //fattorizzo una sola volta $\mathcal{O}(n^3)$

$\forall k = 1, \dots$

$$y^{(k)} = A^{-1} z^{(k-1)} \quad // \mathcal{O}(n^2), \text{ risolvo } LUy^{(k)} = z^{(k-1)}$$

$$z^{(k)} = \frac{y^{(k)}}{\|y^{(k)}\|_2} \quad // \mathcal{O}(n)$$

Se $|R(z^{(k)}) - R(z^{(k-1)})| < \text{TOL} \Rightarrow \text{STOP}$

$$\tilde{\lambda}_n = \left[\frac{[z^{(n)}]^H A z^{(n)}}{[z^{(n)}]^H z^{(n)}} \right]^{-1}$$

$$\tilde{v}^{(n)} = z^{(n)}$$

Complessità: $\mathcal{O}(n^3)$

1.2. Metodo delle potenze con shift

Voglio calcolare l'autocoppia con λ più vicino a $\sigma \in \mathbb{C}$. Basta applicare il metodo a $[A - \sigma \cdot I]^{-1}$

Che ha come autovalori $\frac{1}{\lambda_j - \sigma}$, e come autovettori gli stessi di A .

Algoritmo (metodo delle potenze inverse):

$z^0 \leftarrow \text{input}$

$A - \sigma I = \Pi LU$

$\forall k = 1, \dots$

Risolvo $(A - \sigma I)y^{(k)} = z^{(k-1)}$

$$z^{(k)} = \frac{y^{(k)}}{\|y^{(k)}\|_2} \quad // \mathcal{O}(n)$$

Se $|R(z^{(k)}) - R(z^{(k-1)})| < \text{TOL} \Rightarrow \text{STOP}$

$$\tilde{\lambda} = \frac{1}{[z^{(n)}]^H A z^{(n)}} + \sigma$$

$$\tilde{v} = z^{(n)}$$

Complessità: $\mathcal{O}(\#\text{it} \cdot n^2)$