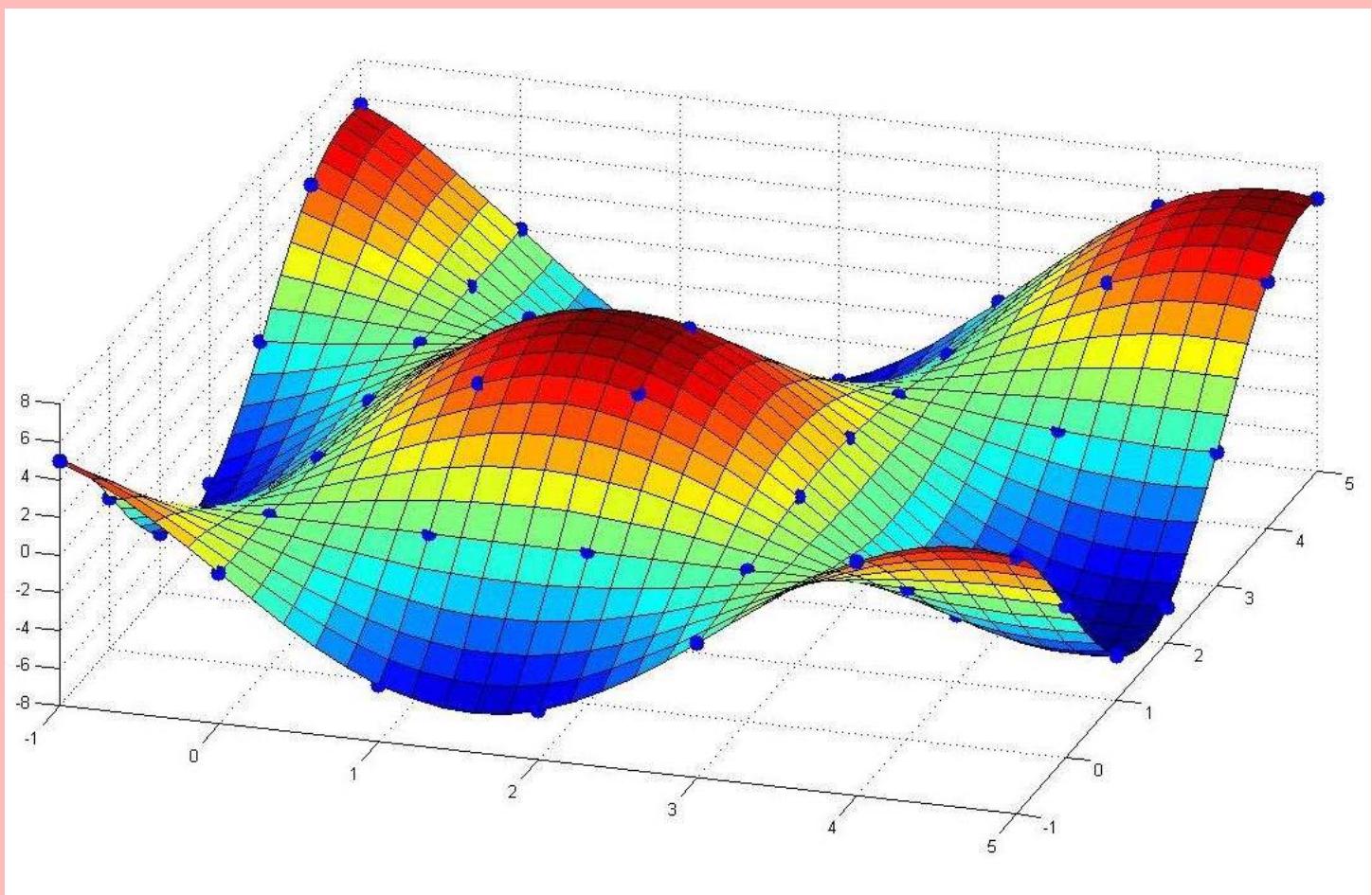


CALCOLO NUMERICO

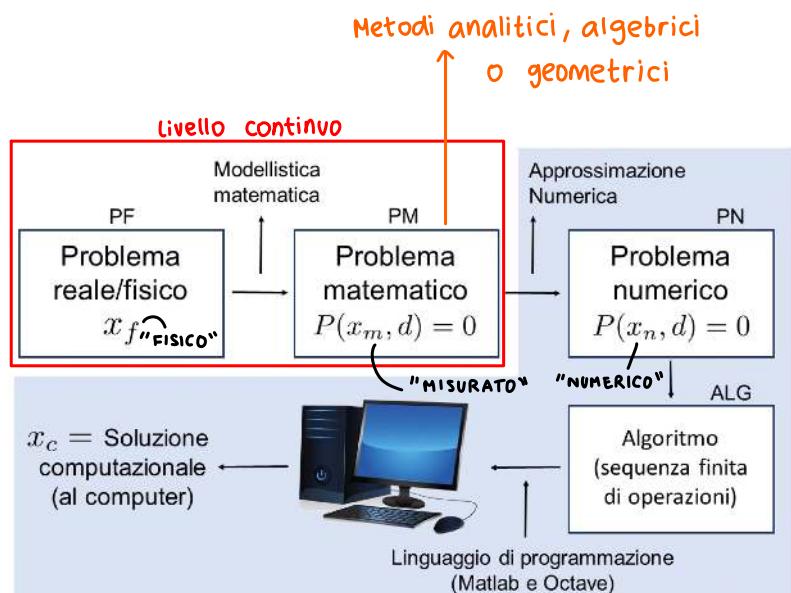


INTRODUZIONE AL CORSO

- E' tutto nelle slide
- Laboratori al calcolatore con Matlab
- Esame scritto (orale facoltativo ± 3) su Form
 - 12 domande a risposta multipla (no penalità) : 6 punti
 - Matlab : 16 punti
 - Una domanda di teoria : 10 punti

CALCOLO NUMERICO

- Informazioni quantitative su un problema fisico (es: vel. sangue in un arteria con stenosi $\rightarrow x_f$)
- Misuro con apparato sperimentale, x_m
- Introduco un modello matematico P , che dipende da x_m e dai dati d . Modello accurato se $x_m \approx x_f$
- Approssimazione rigorosa \rightarrow numerico ("che approssima") con soluzione x_n ; deve essere facilmente risolvibile.
- Errori nel processo di approssimazione
 - Di modello $\rightarrow |x_f - x_m|$
 - Numerico \rightarrow dovuto all'approssimazione numerica ; $|x_m - x_n|$
 - computazionale \rightarrow dovuto all'implementazione al calcolatore ; $|x_n - x_c|$



QUESTIONI FONDAMENTALI

1. Come generare un problema numerico
2. Come trovare algoritmi opportuni
3. Verificare che x_n esiste ed è unica
4. Verificare che sia **stabile** (limitata in funzione dei dati)
5. Verificare che sia **convergente**, ovvero che $|x_n - x_m|$ sia "piccolo"

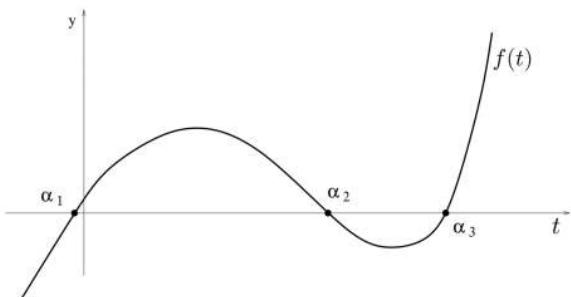
NOTA : Non conosco la x_m , ma so in che range deve trovarsi intuitivamente (in base al contesto)
 Il problema matematico non è risolvibile!
 Da questa necessità si origina tutta la teoria del calcolo numerico → approssimo x_m con una x_n

PROBLEMI MATEMATICI

1 - Radici di equazioni non lineari

- Ricerca degli zeri/radici di una funzione
- La funzione si annulla per valori di α

$x_m = \{\alpha_1, \dots, \alpha_n\} \rightarrow$ radici
 $d = \{a_0, \dots, a_n\} \rightarrow$ coefficiente, vettore



2 - Sistemi lineari

- Possono essere di grandi dimensioni

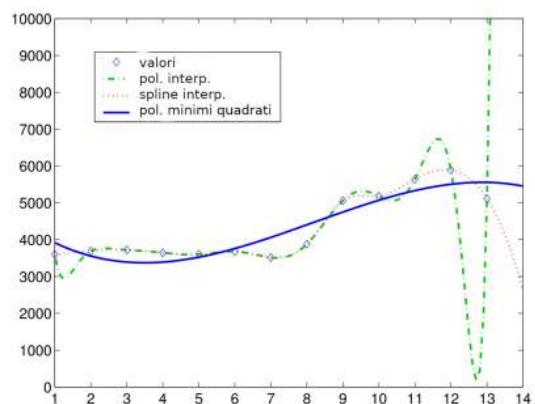
$$A \bar{x} = \bar{b}$$

$$\begin{aligned} x_m &= \bar{x} \\ d &= \{b_i, a_{ij}\} \end{aligned}$$

3 - Interpolazione

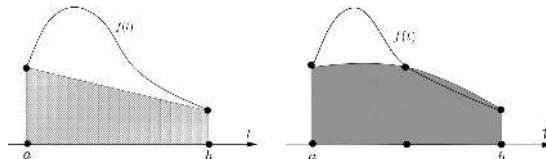
- Approssimo una funzione a partire da dati misurati sperimentalmente

$d = \{(x_0, y_0), \dots, (x_n, y_n)\} \rightarrow$ misure
 $x_n = \{a_0, \dots, a_n\} \rightarrow$ coefficienti, vettore
 x_m è la funzione reale che non conosco



4 - Integrazione

$$\begin{aligned} x_m &= \int_a^b f(t) dt \\ d &= \{a, b, f\} \end{aligned}$$



Formule dei trapezi (a sinistra) e di Simpson (a destra)

5 - Equazioni Differenziali Ordinarie (EDO)

$$\begin{aligned} d &= \{t_0, y_0, f\} \\ x_m &= y \rightarrow \text{funzione} \end{aligned} \quad \left\{ \begin{array}{l} y'(t) = f(t, y(t)) \quad t_0 \geq t \\ y(t_0) = y_0 \end{array} \right.$$

ESEMPIO CONCRETO

PF: determinare la numerosità di una popolazione di batteri $\rightarrow x_f$ = numero di batteri

$$\text{PM : } d = \begin{cases} y_0 = \text{numero di batteri all'istante iniziale } t = t_0 = 0 \\ B = \text{numero max di batteri } (B \geq y_0) \\ k = \text{fattore di crescita} \end{cases}$$

$x_m = y(t) = \text{numero di batteri all'istante } t > t_0$



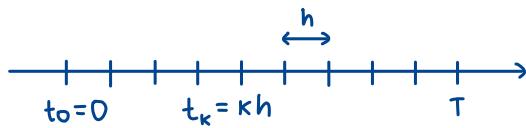
$$\text{EDO (PM)} \quad \left\{ \begin{array}{l} y'(t) = k y(t) (1 - y(t)/B), \quad \forall t \geq t_0 \\ y(t_0) = y_0 \end{array} \right. \quad f(t, y)$$

MODELLO MATEMATICO
DI CRESCITA

Non sono in grado di risolvere analiticamente PM \rightarrow Introduco approssimazione numerica

Approssimazione. Vogliamo approssimare la derivata $y'(t) = f(t, y(t))$

A tal fine, introduco una partizione sul dominio \rightarrow discretizzazione



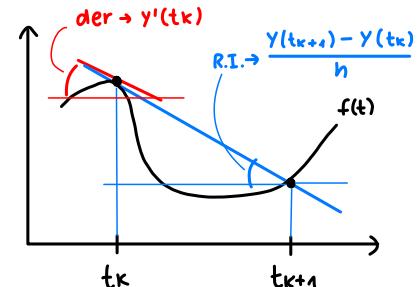
$\downarrow h \Rightarrow \uparrow$ precisione, MA problema più difficile

Idea per scrivere PN: scrivo PM per $t = t_k$

$$y'(t_k) = f(t_k, y(t_k))$$

PASSO dalla derivata al rapporto incrementale

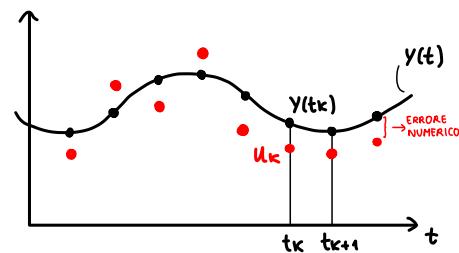
$$y'(t_k) \rightarrow \frac{y(t_{k+1}) - y(t_k)}{t_{k+1} - t_k} = \frac{y(t_{k+1}) - y(t_k)}{h}$$



Introduco soluzione numerica $x_n = u_k, k=1, \dots$

PM	PN
$y'(t_k) = f(t_k, y(t_k)) \quad \forall k$	$\frac{u_{k+1} - u_k}{h} = f(t_k, u_k) \quad \forall k$

dove u_k è un'approssimazione di $y(t_k)$



NB: $x_m = y(t)$ è una funzione di t

$x_n = u_k$ è un vettore (u_1, \dots, u_k, \dots)

Quesiti : STABILITÀ : x_n non deve esplodere $\rightarrow |u_k| \leq C y_0$

CONVERGENZA : dovrò verificare che $|u_k - y(t_k)| \rightarrow 0 \quad \forall k$, per $h \rightarrow 0$

1. RICERCA DI RADICI DI FUNZIONI NON LINEARI

Il primo dei cinque quesiti che analizziamo è la ricerca degli zeri di una funzione. Vediamo due esempi. Si tratta di una ricerca numerica, quindi basata su approssimazioni.

Esempio 1 : Piano di investimenti

$$M = v \sum_{k=1}^n (1+r)^k = v \frac{1+r}{r} [(1+r)^n - 1]$$

ogni anno moltiplico per $(1+r)$

Ogni anno investo v euro, dopo n anni si è accumulato M . Qual è il tasso di rendimento (interesse) annuo?

$$\Rightarrow f(r) = M - v \frac{1+r}{r} [(1+r)^n - 1] = 0$$

Quali sono i valori di r per cui $f(r) = 0$?
L'equazione è non lineare, non la so risolvere a mano.

Esempio 2

Gas a temperatura T soggetto a pressione p . Qual è il volume occupato?

Equazione di stato :

$$\left[p + a \left(\frac{N}{V} \right)^2 \right] (V - N_b) = kNT$$

con

- a, b coefficienti tipici del gas in esame
- N = numero di molecole di gas contenute in V
- k = costante di Boltzman

DATI NOTI

per gas perfetti :
 $pV = nRT$

Legge di van der Waals
(per gas reali)

Dati p, T , a quale volume V si trova il mio gas?
L'equazione è non lineare!

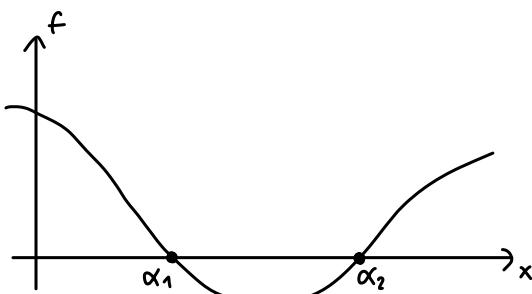
$$\Rightarrow f(V) = \left[p + a \left(\frac{N}{V} \right)^2 \right] (V - N_b) - kNT = 0$$

Problema matematico : Ricerca numerica degli zeri di funzione

PM : cerco $\alpha \in [a, b] : f(\alpha) = 0$, con $f : [a, b] \rightarrow \mathbb{R}$, continua

Dati : $d = [a, b, f] ; x_m = \alpha$

Per la risoluzione numerica, introduco dei metodi iterativi locali



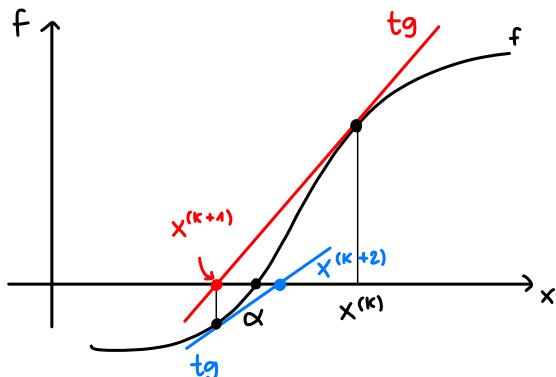
Metodi iterativi locali

Regole per costruire un metodo iterativo :

1. Scelgo $x^{(0)} \in [a, b]$ come punto di partenza
→ La scelta deve essere sensata e appropriata al contesto ($r=0,02, V=1\text{m}^3$)
2. Genero una legge di aggiornamento : $X^{(k)} \rightarrow X^{(k+1)}$ ($X^{(0)} \rightarrow X^{(1)} \rightarrow X^{(2)} \rightarrow \dots \rightarrow X^{(k)} \rightarrow \dots$)
3. Obiettivo matematico $\lim_{k \rightarrow +\infty} X^{(k)} = \alpha$ convergenza

METODO DI NEWTON (O METODO DELLE TANGENTI)

IDEA: Dato $X^{(k)}$, genero $X^{(k+1)}$ come lo zero della tangente ad $f(x)$ in $X^{(k)}$



Genero degli $X^{(k+1)}, X^{(k+2)}, \dots$ sempre più vicini ad α

Legge di Aggiornamento

$$tg : y(x) = f(X^{(k)}) + f'(X^{(k)}) (x - X^{(k)})$$

DALLA SERIE DI TAYLOR
 $y_2 = y_1 + y_1' \cdot (x_2 - x_1)$

scelgo $X^{(k+1)}$ tale che sia lo zero di $y(x)$

$$\downarrow \\ y(X^{(k+1)}) = 0 \rightarrow 0 = f(X^{(k)}) + f'(X^{(k)}) (X^{(k+1)} - X^{(k)})$$

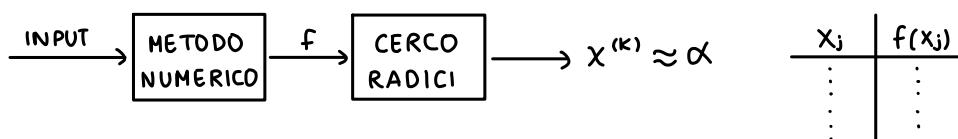
$$X^{(k+1)} = X^{(k)} - \frac{f(X^{(k)})}{f'(X^{(k)})} \quad k \geq 0$$

Nell' ipotesi che $f'(X^{(k)}) \neq 0 \quad \forall k$

- E' convergente questo metodo? (vedremo poi)

PROBLEMATICA

(i) Spesso la f non è conosciuta analiticamente, ma per punti.



In questa situazione non posso calcolare f'
(non conosco f)

(ii) Potrebbe essere $f'(X^{(k)}) = 0 \rightarrow$ il metodo si arresta nei punti di massimo o minimo.
In entrambi i casi non posso utilizzare Newton.

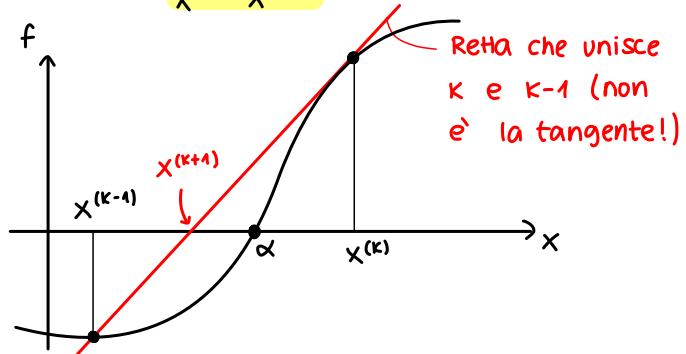
METODO DELLE SECANTI → Newton modificato, gestisce le due precedenti problematiche

IDEA: sostituisco $f'(X^{(k)})$ con il rapporto incrementale $\frac{f(X^{(k)}) - f(X^{(k-1)})}{X^{(k)} - X^{(k-1)}}$

Legge di aggiornamento :

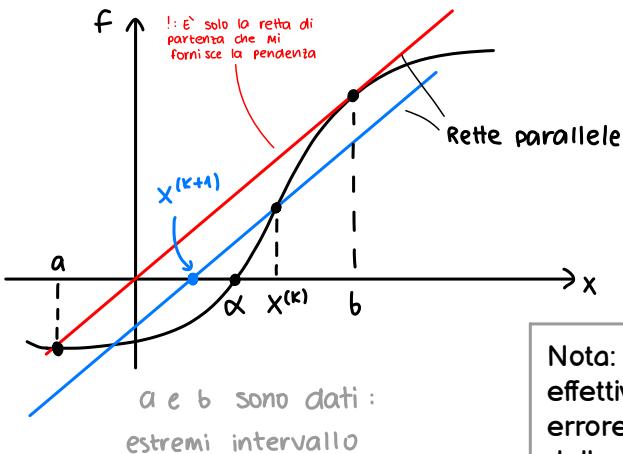
$$X^{(k+1)} = X^{(k)} - f(X^{(k)}) \frac{X^{(k)} - X^{(k-1)}}{f(X^{(k)}) - f(X^{(k-1)})}$$

- E' convergente?
- Ho bisogno di due input : $k, k-1$



METODO DELLE CORDE

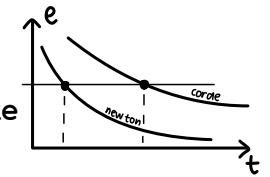
Approssimo $f'(x^{(k)}) \rightarrow \frac{f(b) - f(a)}{b - a}$



Metodo Sempificativo di Newton e Secanti.
Non vario la pendenza delle curve di cui si valuta lo zero. Lo mantengo costante.
 \uparrow semplicità $\Rightarrow \downarrow$ velocità, \downarrow precisione

$$x^{(k+1)} = x^{(k)} - f(x^{(k)}) \frac{b-a}{f(b)-f(a)}$$

Nota: la velocità di calcolo per la singola iterazione è effettivamente minore, ma se considero una soglia di errore minima da raggiungere, utilizzando il metodo delle corde, che è meno preciso, devo compiere molte più iterazioni rispetto al metodo di Newton o delle Secanti. Per cui, il metodo delle corde risulta essere complessivamente più lento. $t_{\text{COMPLESSIVO}} = t_{\text{ITERAZIONE}} \cdot n_{\text{ITERAZIONI}}$

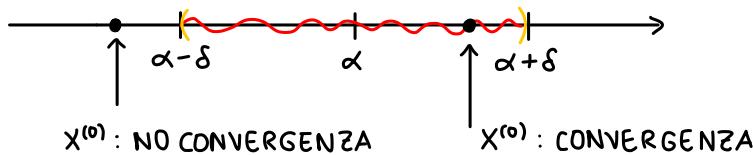


Cosa vuol dire che i metodi sono locali?

METODO LOCALE : se $\exists \delta > 0$: se $x^{(0)} \in (\alpha - \delta ; \alpha + \delta)$

Allora $\lim_{k \rightarrow +\infty} |x^{(k)} - \alpha| = 0$

Ho convergenza per metodo locale solo se parto da un $x^{(0)}$ sufficientemente vicino ad α .



Tutti e 3 metodi introdotti SONO locali.

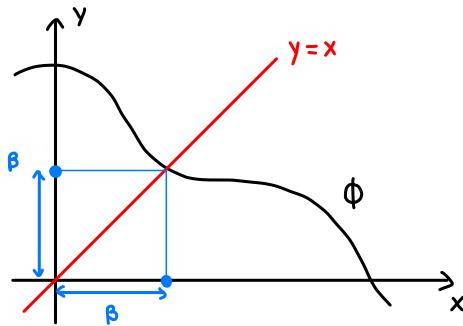
Io non conosco α , ma
devo fare una stima
ingegneristica/fisica su dove
potrà cadere il mio valore.
Conosco il problema in analisi
e scelgo $x^{(0)}$ su basi
ingegneristiche/fisiche del problema

Definizione di punto fisso

Vogliamo definire il problema di punto fisso :

Data $\Phi[a, b] \rightarrow \mathbb{R}$, cerco PUNTO FISSO $\beta \in [a, b]$, cioè $\beta = \Phi(\beta)$

→ intersezione di Φ con la bisettrice



RELAZIONE TRA PUNTI FISSI E ZERI

(i) Se α è zero di $f(x)$, allora α è punto fisso di $\Phi(x) = x - f(x)$.

Infatti, se $f(\alpha) = 0 \Rightarrow \Phi(\alpha) = \alpha - f(\alpha) = \alpha$

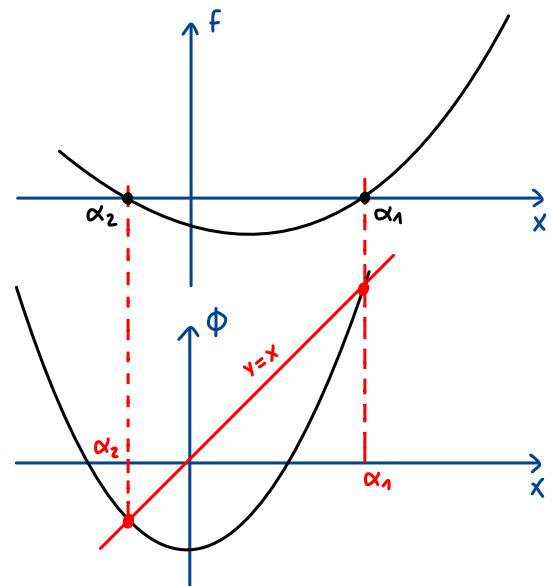
(ii) Viceversa, se β è punto fisso di Φ , allora β è radice di $f(x) = x - \Phi(x)$.

Infatti, se $\beta = \Phi(\beta) \Rightarrow f(\beta) = \beta - \Phi(\beta) = 0$

Esempio

$$f(x) = -x^2 + x + 1 \rightarrow \alpha_{1,2} = \frac{-1 \pm \sqrt{1+4}}{-2} = \frac{1 \pm \sqrt{5}}{2}$$

Allora α_1, α_2 sono i punti fissi di $\Phi(x) = x - f(x) = x^2 - 1$



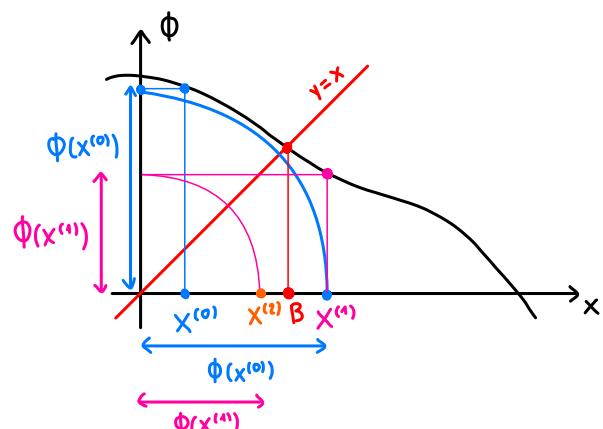
Posso riscrivere il problema

$$\left\{ \begin{array}{l} \text{Trovare } \alpha \text{ tale che } f(\alpha) = 0 \\ \text{e } \Phi(\alpha) = \alpha \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} \text{cercare } \beta \text{ tale che} \\ \beta = \Phi(\beta) \text{ con } \Phi(x) = x - f(x) \end{array} \right.$$

Iterazioni di punto fisso

Per trovare numericamente i punti fissi introduco il metodo di iterazioni di punto fisso :

Dato $x^{(0)} \in [a, b]$, $x^{(k+1)} = \Phi(x^{(k)})$ $k \geq 0$



Esempio 1

$$\Phi(x) = \cos(x) \quad \beta \text{ esiste, MIPF converge}$$

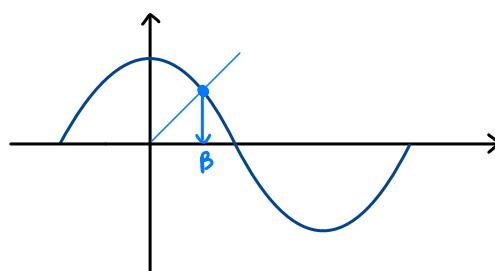
$$x^{(0)} = 1$$

$$x^{(1)} = \cos(x^{(0)}) = \cos(1) = 0,540302\dots$$

$$x^{(2)} = \cos(x^{(1)}) = 0,8575\dots$$

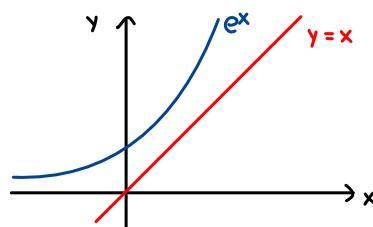
⋮

$$x^{(20)} = \cos(x^{(19)}) = 0,7392\dots \xrightarrow{k \rightarrow \infty} \beta = 0,7390\dots$$



Esempio 2

$\Phi(x) = e^x$ NON ESISTONO PUNTI FISSI!



Esempio 3

Punto fisso β esiste, ma MIPF non converge

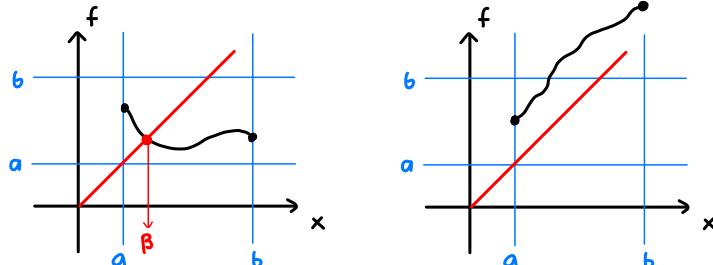
$$\Phi(x) = x^2 - 1 \rightarrow \beta_{1,2} = \frac{1 \pm \sqrt{5}}{2}$$

$$x^{(0)} = 1 \rightarrow x^{(1)} = (x^{(0)})^2 - 1 = 0 \\ \rightarrow x^{(2)} = (x^{(1)})^2 - 1 = -1 \rightarrow x^{(3)} = 0 \dots \text{Non converge}$$

TEOREMA : $\exists!$ PUNTO FISSO

Sia $\Phi: [a, b] \rightarrow \mathbb{R}$ continua e sia $x^{(0)}$ in $[a, b]$ assegnato. Allora :

(i) se $\Phi: [a, b] \rightarrow [a, b]$, allora esiste un punto fisso $\beta \in [a, b]$



(ii) Se $\exists L < 1 : |\Phi(x_1) - \Phi(x_2)| \leq L |x_1 - x_2| \quad \forall x_1, x_2 \in [a, b]$

allora • β è unico

• $\lim_{k \rightarrow \infty} x^{(k)} (= \Phi(x^{(k-1)})) = \beta$ MIPF converge

Sia $\Phi: [a, b] \rightarrow \mathbb{R}$, continua e derivabile, allora:

(i) se la $|\Phi'(\beta)| < 1$, allora $\exists \delta > 0$:

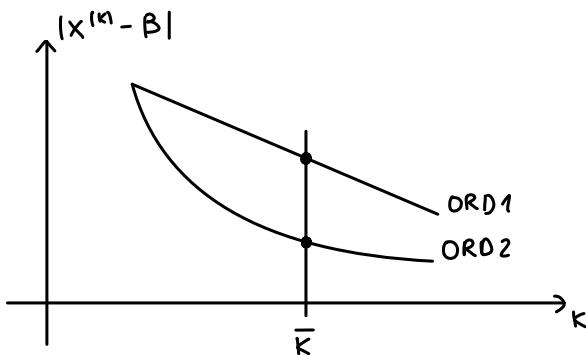
$\forall x^{(0)} \in (\beta - \delta, \beta + \delta)$, la $x^{(k)}$ converge a β ($\lim_{k \rightarrow \infty} x^{(k)} = \beta$)

INOLTRE: $\lim_{k \rightarrow \infty} \frac{|x^{(k+1)} - \beta|}{|x^{(k)} - \beta|} = \Phi'(\beta)$ " Errore nuovo < errore vecchio "

ORDINE DI CONVERGENZA 1
(errore scala linearmente)

(ii) se inoltre $\Phi \in C^2$ e $\Phi'(\beta) = 0$ e $\Phi''(\beta) \neq 0$, allora:

$\lim_{k \rightarrow \infty} \frac{|x^{(k+1)} - \beta|}{|x^{(k)} - \beta|^2} = \frac{\Phi''(\beta)}{2}$ ORDINE DI CONVERGENZA 2 → ha velocità maggiore
(scala quadraticamente)



NOTA: Non conosco β , come faccio a dire se $\Phi'(\beta) < 1$ o $= 0$?
Vedremo che gli esercizi sono casi particolari

Operativamente il precedente risultato può essere approssimato per k (sufficientemente grande) finito, come segue:

$$\text{se } x^{(0)} \in (\beta - \delta, \beta + \delta) \Rightarrow (i) \quad \frac{|x^{(k+1)} - \beta|}{\text{errore al passo } k+1} \simeq \frac{|\Phi'(\beta)|}{\text{fattore di riduzione}} \cdot \frac{|x^{(k)} - \beta|}{\text{errore al passo } k}$$

↓

$$\simeq |\Phi'(x^{(k)})| \text{ che è calcolabile!}$$

(ii) Se $\Phi'(\beta) = 0$ e $\Phi''(\beta) \neq 0 \Rightarrow$

$$|x^{(k+1)} - \beta| \simeq \left| \frac{\Phi''(\beta)}{2} \right| |x^{(k)} - \beta|^2$$

Esempio: $\Phi(x) = x^2 - 1$ avevamo trovato che MIPF non converge per $x^{(0)} = 1$

$$\beta_{1,2} = \frac{1 \pm \sqrt{5}}{2} \quad \Phi'(x) = 2x \Rightarrow \Phi'(\beta_{1,2}) = 1 \pm \sqrt{5} \quad |\Phi'(\beta_{1,2})| > 1$$

MIPF non poteva convergere

Ricordiamo: se α è radice di $f(x)$ ($f(\alpha) = 0$), allora α è punto fisso di $\Phi(x) = x - f(x)$ ($\alpha = \Phi(\alpha)$). Inoltre noto che α è anche punto fisso di $\Phi_q(x) = x - f(x)/q$, $\forall q \neq 0$ (*)

$$\text{Infatti: } \Phi_q(\alpha) = \alpha - \frac{f(\alpha)}{q} = \alpha$$

Interpreto i metodi iterativi per le radici come MIPF:

- **CORDE** $x^{(k+1)} = x^{(k)} - f(x^{(k)}) \frac{b-a}{f(b)-f(a)} \Rightarrow \Phi_c(x) = x - f(x) \frac{b-a}{f(b)-f(a)} \rightarrow \frac{1}{a}$

Infatti se applico MIPF a Φ_c , ottengo il METODO DELLE CORDE:

$$x^{(k+1)} = \Phi_c(x^{(k)}) = x^{(k)} - f(x^{(k)}) \frac{b-a}{f(b)-f(a)}$$

- **SECANTI** $x^{(k+1)} = x^{(k)} - f(x^{(k)}) \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})}$
 \Rightarrow SECANTI può essere scritto come (*) con $q = \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$

Il metodo delle secanti non può essere visto come MIPF (perché q dipende da k)

- **NEWTON** $x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})} \rightarrow q = f'(x^{(k)})$

Introduco $\Phi_N(x) = x - \frac{f(x)}{f'(x)}$, allora MIPF applicato a Φ_N mi dà NEWTON.

Infatti: $x^{(k+1)} = \Phi_N(x^{(k)}) = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$

anche qui ho dipendenza da k , però a differenza del metodo delle secanti in questo caso posso calcolare f' generica e poi valutarla per ogni k , mentre nelle secanti dovrai calcolare q per ogni k diverso.

Studiamo la convergenza:

- **CORDE** $\Phi_c(x) = x - f(x) \frac{b-a}{f(b)-f(a)}$

NOTA: In generale, $\Phi'_c(\alpha) = 1 - f'(\alpha) \frac{b-a}{f(b)-f(a)} \neq 0 \rightarrow$ ORDINE 1

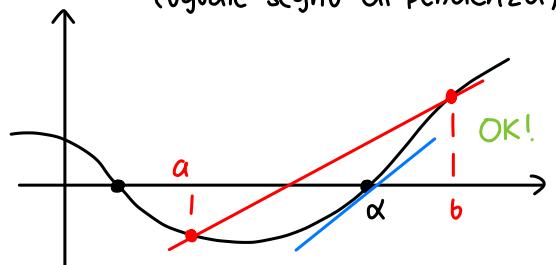
Convergenza per $|\Phi'_c(\alpha)| < 1 \rightarrow -1 < 1 - f'(\alpha) \frac{b-a}{f(b)-f(a)} < 1$

$$-2 < -f'(\alpha) \frac{b-a}{f(b)-f(a)} \quad \leftarrow \quad \rightarrow -f'(\alpha) \frac{b-a}{f(b)-f(a)} < 0$$

$$b-a < \frac{2}{f'(\alpha)} (f(b)-f(a))$$

b-a sufficientemente piccolo
 \Rightarrow METODO LOCALE

$\begin{cases} f'(\alpha) \text{ e} \\ \frac{f(b)-f(a)}{b-a} \end{cases}$ devono avere lo stesso segno (uguale segno di pendenza)



- **NEWTON** $\Phi_N(x) = x - \frac{f(x)}{f'(x)}$

$$\Phi'_N(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2}$$

\Rightarrow se $f'(\alpha) \neq 0$, allora $\Phi_N(\alpha) = 0$ (perché $f(\alpha) = 0$) \rightarrow ORDINE 2

DEF: α Radice di $f(x)$ ha molteplicità $m \geq 1$ se $f^{(j-1)}(\alpha) = 0, \forall j \leq m$.

Quindi se $m=1 \rightarrow$ NEWTON ordine 2

se invece $m > 1 \rightarrow$ NEWTON ordine 1, perché $\Phi'(\alpha) = 1 - 1/m$

• **METODO DI NEWTON MODIFICATO** : $\Phi_N^m(x) = x - m \frac{f(x^{(k)})}{f'(x^{(k)})}$.

Si ha : $\Phi_N^m(\alpha) = 0 \rightarrow$ ORDINE 2

Tuttavia, non
conosco m
ci sono metodi
numerici per
stimare m

• **SECANTI** Altri argomenti teorici portano a dimostrare che ordine $\approx 1,6$

$$\Rightarrow |x^{(k+1)} - \alpha| \approx C |x^{(k)} - \alpha|^{1.6}$$

Criteri di arresto

Convergenza : $\lim_{k \rightarrow \infty} |x^{(k)} - \alpha| = 0$

Potrei arrestare le iterazioni quando, scelta una opportuna tolleranza $\epsilon > 0$,

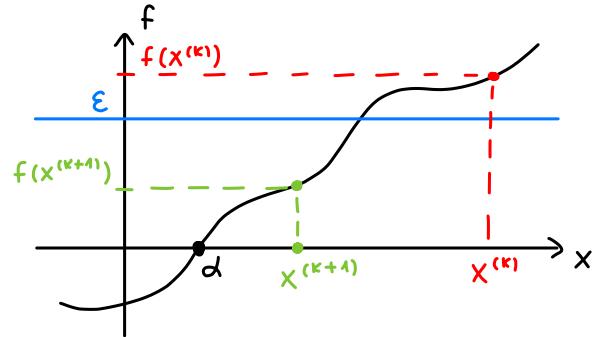
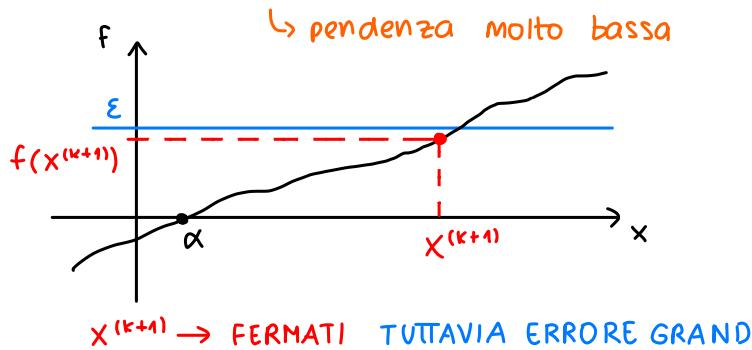
$$|x^{(k)} - \alpha| < \epsilon \rightarrow \text{NON IMPLEMENTABILE}$$

Introduco 2 criteri implementabili :

(1) CRITERIO SU RESIDUO

Quando $|f(x^{(k)})| < \epsilon \rightarrow$ IMPLEMENTABILE

Se però $|f'(\alpha)| << 1$ allora il CRITERIO sul RESIDUO non è affidabile :



- $X^{(k)} \rightarrow \text{NON FERMATI}$
- $X^{(k+1)} \rightarrow \text{FERMATI}$

(2) CRITERIO SU INCREMENTO

: Mi fermo quando $|x^{(k+1)} - x^{(k)}| < \epsilon$

si può dimostrare che $\frac{|\alpha - x^{(k)}|}{\text{errore}} \approx \left| \frac{1}{1 - \Phi'(\alpha)} \right| \cdot \frac{|x^{(k+1)} - x^{(k)}|}{\text{incremento}}$

Questo criterio è affidabile per $\Phi'(\alpha) \approx 0$ (ad esempio Newton)

Nota: spesso vengono implementati i CRITERI RELATIVI (%) : $\left| \frac{f(x^{(k)})}{f(x^{(0)})} \right| < \epsilon ; \frac{|x^{(k+1)} - x^{(k)}|}{|x^{(k+1)}|} < \epsilon$

2A. METODI DIRETTI PER SISTEMI LINEARI

Consideriamo il seguente sistema lineare

$$Ax = b$$

ove $A \in \mathbb{R}^{n \times n}$ di componenti a_{ij} e $b \in \mathbb{R}^n$ sono assegnati e $x \in \mathbb{R}^n$ è il vettore delle incognite

Queste sono n equazioni lineari nelle incognite x_j

La i-esima equazione è

$$\sum_{j=1}^n a_{ij}x_j = b_i \rightarrow a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i \quad \forall i = 1, \dots, n$$

PRODOTTO RIGA per COLONNA

- $A \rightarrow$ una matrice invertibile e quadrata di dimensioni n e componenti a_{ij}
- $b \rightarrow$ termine noto
- $x \rightarrow$ vettore incognite

Sistema di n equazioni in n incognite

Esempio

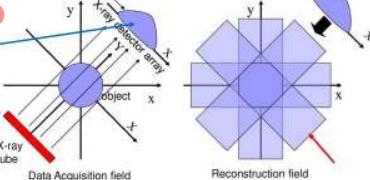
Tomografia Assiale Computerizzata (TAC)

Raggi X emessi vengono in parte assorbiti dal corpo



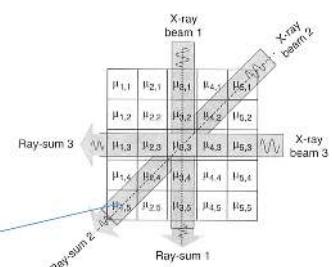
Rotazione emettitore

Si effettuano numerose misure di assorbimento come differenza fra segnale emesso e ricevuto



Le misure di assorbimento sono proporzionali alla densità del tessuto

Si numerano i pixel come componenti di un vettore e si riordinano in accordo con gli assorbimenti $\mu_{ij} \rightarrow x_k$



Ogni misura b_i è data dalla somma degli assorbimenti x_k lungo la linea di emissione i

$$\rightarrow Ax = b$$

con b_i la misura lungo la linea i
 x_j la densità del pixel j
 $a_{ij} = 1$ se pixel j appartiene a linea i
dimensione = numero pixel/voxel
→ tiene conto dell'assorbimento

il sistema lineare ha come numero di incognite il numero di pixel.

SOLUZIONE ESATTA

La soluzione del sistema esiste unica se $\det(A) \neq 0$

La sua espressione è data da

$$x_j = \frac{\det(A_j)}{\det(A)}$$

con $A_j = [a_1 \dots a_{j-1} \ b \ a_{j+1} \dots a_n]$ Formula di Cramer
e a_i le colonne di A

Tuttavia essa è inutilizzabile!

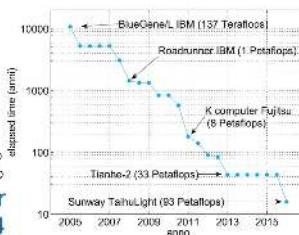
Infatti, il calcolo di un

determinante $\sim n!$ operazioni

1 Teraflops = 10^{12} operazioni al secondo

1 Petaflops = 10^{15} operazioni al secondo

Tempo di calcolo sul miglior supercomputer disponibile per un sistema lineare con $n=24$



per una matrice $n \times n$, calcolare il determinante richiede $n!$ operazioni

(caso in cui la soluzione esatta esiste ma non è applicabile)

Metodi numerici

Data $A \in \mathbb{R}^{n \times n}$, $\det(A) \neq 0$ (INVERTIBILE) e dato $b \in \mathbb{R}^n$, cerco $x \in \mathbb{R}^n$ tale che

$$A x = b$$

- n equazioni in n incognite x_j
- i-esima equazione $\sum_{j=1}^n a_{ij} x_j = b_i$

Partiamo analizzando due casi particolari e poi generalizziamo

1) Considero il caso particolare di **MATRICI TRIANGOLARI INFERIORI**

$$A = L, \quad l_{ij} = 0 \text{ per } i < j$$

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

Voglio risolvere $Lx = b$

$$1^{\text{a}} \text{ RIGA: } \sum_{j=1}^n l_{1j} x_j = b_1 \rightarrow l_{11} x_1 = b_1 \rightarrow x_1 = \frac{b_1}{l_{11}} \quad (\text{riesco a risolvere la prima riga})$$

$$2^{\text{a}} \text{ RIGA: } \sum_{j=1}^n l_{2j} x_j = b_2 \rightarrow l_{21} x_1 + l_{22} x_2 = b_2 \rightarrow x_2 = \frac{b_2 - l_{21} x_1}{l_{22}} \quad \text{con } x_1: \text{noto}$$

Generalizzando, ho il **METODO DELLE SOSTITUZIONI IN AVANTI**:

$$\boxed{x_1 = b_1 / l_{11}}$$

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} x_j}{l_{ii}}, \quad i = 2, \dots, n \quad \xrightarrow{\text{noto}}$$

Esempio

$$L = \begin{pmatrix} 2 & 0 & 0 \\ -1 & 4 & 0 \\ 7 & 3 & -5 \end{pmatrix} \quad \underline{b} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad : \quad Lx = \underline{b}$$

$$x_1 = 1/2$$

$$x_2 = \frac{2 - (-1 \cdot 1/2)}{4} = 5/8$$

$$x_3 = \frac{3 - (7 \cdot 1/2 + 3 \cdot 5/8)}{-5} = 19/40$$

Quanto costa (in termini di operazioni) il MSA?

$$\# \text{OPERAZIONI} = \sum_{i=1}^n (\downarrow \quad + \quad \downarrow \quad + \quad \downarrow \quad + \quad \downarrow) = \sum_{i=1}^n (2i - n) = n^2 \text{ FLOPS}$$

SOTTRAZIONE DIVISIONE MOLTIPLICAZIONI SOMME

es: per TAC, $n = 10^4 \Rightarrow \# \text{OPERAZIONI MSA} = 10^8 \text{ FLOPS}$ (flops = operazioni)

→ **CPU TIME << 1s** (accettabile)

2) Lo stesso procedimento si applica per **MATRICI TRIANGOLARI SUPERIORI**

$$U: u_{ij} = 0 \quad i > j$$

$$U = \begin{pmatrix} u_{11} & & u_{1j} \\ & u_{22} & \\ 0 & u_{3j} & \dots & u_{nn} \end{pmatrix}$$

Ho il **METODO DELLE SOSTITUZIONI ALL'INDIETRO**

$$\boxed{x_n = b_n / u_{nn}}$$

$$x_i = \frac{b_i - \sum_{j=i+1}^n u_{ij} x_j}{u_{ii}} \quad \xrightarrow{\text{noto}}$$

CPU TIME = n^2 FLOPS

3) Matrice generale: introduco le sottomatrici principali di A : $A_{p,i} \in \mathbb{R}^{i \times i}$, $i=1, \dots, n$

TEOREMA: Se $\det(A_{p,i}) \neq 0 \quad \forall i=1, \dots, n$, allora esistono una matrice triangolare inferiore L (con $l_{ii}=1$) e una triangolare superiore U , tali che

$$A = LU$$

Fattorizzazione LU

Fissiamo una coppia di matrici, ma ne esistono tante diverse. è una scelta della letteratura per avere un vincolo

$$A = \begin{pmatrix} a_{11} & \dots & a_{1i} & a_{1i+1} & \dots & a_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{i1} & \dots & a_{ii} & a_{i+1,1} & \dots & a_{i+1,n} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{ni} & a_{n+1,1} & \dots & a_{n+1,n} \end{pmatrix} \rightarrow A_{p,i}$$

Considero ora $A \underline{x} = \underline{b}$. Supponiamo che A soddisfi il teorema ($\det(A_{p,i}) \neq 0, \forall i$)

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

FATT LU

$$LU \underline{x} = \underline{b}$$

$$\begin{cases} L \underline{y} = \underline{b} \\ U \underline{x} = \underline{y} \end{cases}$$

1 SISTEMA GENERICO

$\underline{y} \in \mathbb{R}^n$ variabile ausiliaria

2 SISTEMI TRIANGOLARI EQUIVALENTI

La risoluzione di $A \underline{x} = \underline{b}$ con la FATT. LU può essere fatta con i METODI DELLE SOSTITUZIONI APPLICATI A $L \underline{y} = \underline{b} \rightarrow U \underline{x} = \underline{y} \rightarrow \underline{x}$. Tuttavia:

1) Come trovo L e U ? → Metodo dell'eliminazione Gaussiana (MEG)

2) Quanto costerà trovarli? → Numero operazioni $\frac{2}{3} n^3$

METODO DELL'ELIMINAZIONE GAUSSIANA (MEG)

Si introducono matrici $A^{(k)}$ e termini noti $b^{(k)}$ ad ogni passo

Siano $A^{(1)} = A$, $a_{ij}^{(k)} = (A^{(k)})_{ij}$

$A^{(1)} = A \rightarrow A^{(2)} \rightarrow \dots \rightarrow A^{(k)} \rightarrow A^{(k+1)} \rightarrow \dots \rightarrow A^{(n)} = U$

$b^{(1)} = b \rightarrow b^{(2)} \rightarrow \dots \rightarrow b^{(k)} \rightarrow b^{(k+1)} \rightarrow \dots \rightarrow b^{(n)} = y$

L'idea è quella di annullare gli elementi $a_{ij}^{(k)}$ con

$i \geq k$ e $j < k$ oppure

$2 \leq i \leq k-1$ e $j < i$

combinò linearmente le ultime righe della matrice

$$A^{(k)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & \dots & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & \dots & \dots & a_{2n}^{(2)} \\ 0 & 0 & \dots & \dots & \dots & a_{3n}^{(3)} \\ \vdots & \vdots & \ddots & \dots & \dots & \vdots \\ 0 & 0 & \dots & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & a_{kk}^{(k)} & \dots & a_{nn}^{(k)} \end{pmatrix}$$

mi permette di ricavare y , quindi risolve metà problema

Per fare ciò si sottrae un multiplo della riga k alle successive in modo da annullare gli elementi desiderati

$$\begin{aligned} \text{per } k=1, \dots, n-1 \\ \text{per } i=k+1, \dots, n \\ l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \\ \text{per } j=k+1, \dots, n \\ a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)} \\ b_i^{(k+1)} = b_i^{(k)} - l_{ik} b_k^{(k)} \end{aligned}$$

Al termine del MEG abbiamo:

$$A^{(n)} = U \quad l_{ij} \rightarrow L \quad b^{(n)} = y$$

sono 3 cicli for dove viene aggiornato l'elemento $a_{ij}^{(k)}$ per ottenere 0 nel trapezio verde, ↳ la parte gialla

Per una generica matrice $A \in \mathbb{R}^{2 \times 2}$:

$$\begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

I 6 elementi ignoti di L e di U dovranno allora soddisfare le seguenti equazioni (non lineari)

$$\begin{aligned} (e_1) \quad l_{11}a_{11} = a_{11}, & \quad (e_2) \quad l_{11}a_{12} = a_{12}, \\ (e_3) \quad l_{21}a_{11} = a_{21}, & \quad (e_4) \quad l_{21}a_{12} + l_{22}a_{22} = a_{22}. \end{aligned}$$

Il sistema è **indeterminato**, presentando più incognite che equazioni. Per eliminare l'indeterminazione fissiamo arbitrariamente pari a 1 gli elementi diagonali di L , aggiungendo perciò le equazioni $l_{11} = 1$ e $l_{22} = 1$.

A questo punto, il sistema può essere risolto procedendo nel modo seguente: dalle (e_1) ed (e_2) ricaviamo gli elementi a_{11} ed a_{12} della prima riga di U . Se a_{11} è non nullo, da (e_3) si trova allora l_{21} (cioè la prima colonna di L , essendo l_{11} già fissato pari a 1) e, quindi, da (e_4) , l'unico elemento non nullo a_{22} della seconda riga di U .

$$\begin{aligned} \text{per } k=1, \dots, n-1 \\ \text{per } i=k+1, \dots, n \\ l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \\ \text{per } j=k+1, \dots, n \\ a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)} \\ l_{ik}^{(k+1)} = l_{ik}^{(k)} - l_{ik} b_k^{(k)} \end{aligned}$$

MEG: Costo computazionale

Per ogni $k=1, \dots, n$, abbiamo:

$n - k$ operazioni

$2(n - k)^2$ operazioni

$2(n - k)$ operazioni

$$\begin{aligned} &\text{per } k = 1, \dots, n-1 \\ &\text{per } i = k+1, \dots, n \\ &l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \\ &\text{per } j = k+1, \dots, n \\ &a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)} \\ &b_i^{(k+1)} = b_i^{(k)} - l_{ik} b_k^{(k)} \end{aligned}$$

$$\text{In totale: } \sum_{k=1}^{n-1} (2(n-k)^2 + 3(n-k)) = \sum_{p=1}^{n-1} (2p^2 + 3p) =$$

Ricordando che:

$$\sum_{k=1}^n k = \frac{n(n+1)}{2} \quad \text{e} \quad \sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\begin{aligned} \text{otteniamo: } \sum_{k=1}^{n-1} (2(n-k)^2 + 3(n-k)) &= \sum_{p=1}^{n-1} (2p^2 + 3p) \\ &= 2 \frac{n(n-1)(2n-1)}{6} + 3 \frac{n(n-1)}{2} \end{aligned}$$

Costo complessivo Fattorizzazione LU
(MEG + sostituzioni indietro): $\sim 2/3n^3$

trascuro n^2

Es (TAC): $n = 10^4 \rightarrow \# \text{flops } 10^{12} \rightarrow \text{CPU TIME} \simeq 1 \text{s (accettabile)}$

Es : $A = \begin{pmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{pmatrix} ; \quad \underline{b} = \begin{pmatrix} 11/6 \\ 13/12 \\ 47/60 \end{pmatrix}$

(verifica a mano che le 3 sottomatrici principali hanno $\det \neq 0$)

Posso applicare FATT LU \rightarrow MEG

- $A^{(1)} = A$
- Eseguo primo step del MEG ($k=1$) $\rightarrow A^{(2)}$

$k=1 |$

$$i=2 \quad | \quad l_{21} = \frac{a_{21}^{(1)}}{a_{11}^{(1)}} = 1/2$$

$$j=2 \quad | \quad a_{22}^{(2)} = a_{22}^{(1)} - l_{21} a_{12}^{(1)} = 1/3 - 1/2 \cdot 1/2 = 1/12$$

$$j=3 \quad | \quad a_{23}^{(2)} = a_{23}^{(1)} - l_{21} a_{13}^{(1)} = 1/4 - 1/2 \cdot 1/3 = 1/12$$

$$i=3 \quad | \quad l_{31} = \frac{a_{31}^{(1)}}{a_{11}^{(1)}} = 1/3$$

$$j=2 \quad | \quad a_{32}^{(2)} = a_{32}^{(1)} - l_{31} a_{12}^{(1)} = 1/4 - 1/3 \cdot 1/2 = 1/12$$

$$j=3 \quad | \quad a_{33}^{(2)} = a_{33}^{(1)} - l_{31} a_{13}^{(1)} = 1/5 - 1/3 \cdot 1/3 = 4/45$$

$$\Rightarrow A^{(2)} = \begin{pmatrix} 1 & 1/2 & 1/3 \\ 0 & 1/12 & 1/12 \\ 0 & 1/12 & 4/45 \end{pmatrix}$$

parte che cambia

- Eseguo secondo step MEG ($k=2$) $\rightarrow A^{(3)} = U$

$k=2 |$

$$i=3 \quad | \quad l_{32} = \frac{a_{32}^{(2)}}{a_{22}^{(2)}} = \frac{1/12}{1/12} = 1$$

$$j=3 \quad | \quad a_{33}^{(3)} = a_{33}^{(2)} - l_{32} a_{23}^{(2)} = \frac{4}{45} - 1 \cdot \frac{1}{12} = \frac{16-15}{180} = \frac{1}{180}$$

$$\Rightarrow A^{(3)} = U = \begin{pmatrix} 1 & 1/2 & 1/3 \\ 0 & 1/12 & 1/12 \\ 0 & 0 & 1/180 \end{pmatrix}$$

- Applicando i 2 step anche a b trovo:

$$\underline{b}^{(3)} = \underline{y} = (1/6 \quad 1/6 \quad 1/180)^T$$

Infine: $U\underline{x} = \underline{y} \rightarrow \underline{x}$ determinata con MSI:

$$x_3 = (1/180) / (1/180) = 1$$

$$x_2 = (1/6 - 1/12 \cdot 1) / (1/12) = 1$$

$$x_1 = (1/6 - 1/3 \cdot 1 - 1/2 \cdot 1) / 1 = 1$$

NOTA: In pratica non posso verificare la veridicità dell'ipotesi del teorema, perché calcolo determinante ti costa troppo.

Allora introduco condizioni sufficienti di esistenza L e U

condizioni di applicabilità del MEG

1) Matrici a dominanza diagonale stretta:

$$1a) \text{ per righe: } |a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad \forall i$$

$$1b) \text{ per colonne: } |a_{ii}| > \sum_{j \neq i} |a_{ji}| \quad \forall i$$

2) Matrici simmetriche definite positive:

$$A = A^T \quad \text{e} \quad \forall \underline{z} \in \mathbb{R}^n, \quad \underline{z}^T A \underline{z} > 0 \quad \text{se } \underline{z} \neq 0$$

Se queste due condizioni non sono verificate, a priori non possiamo stabilire se il MEG è applicabile

FATTORIZZAZIONE DI CHOLESKI

E' un caso di adattamento del MEG a un caso particolare di matrici simmetriche definite positive.

Consiste nel porre $r_{11} = \sqrt{a_{11}}$ e per $j = 2, \dots, N$:

$$\begin{cases} r_{ij} = \frac{1}{r_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj} \right), & i = 1, \dots, j-1 \\ r_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} r_{kj}^2} \end{cases}$$

↓
TUTTI gli autovalori
sono positivi

Si ha $A = R^T R$, con R triangolare superiore
costo computazionale $\sim n^3/3$ (metà del MEG)

Esempio (2):

$$A = \begin{pmatrix} 1 & 1 & 3 \\ 2 & 2 & 2 \\ 3 & 6 & 4 \end{pmatrix}; \det(A) \neq 0$$

Applico primo step del MEG [...]:

$$A^{(1)} = \begin{pmatrix} 1 & 1 & 3 \\ 0 & \boxed{-4} & 0 \\ 0 & 3 & -5 \end{pmatrix} \quad \text{problema!} \quad a_{22}^{(2)} = 0 \Rightarrow \text{Nonostante il sistema lineare sia risolvibile, il MEG si arresta, perché non posso dividere per 0.}$$

In generale, se l'elemento $a_{kk}^{(k)} = 0$, allora il MEG si arresta. Questo non vuol dire che il sistema non è risolvibile.

Perché? Notiamo che nell'esempio (2) $\det(A_{p,2}) = 0$.

Quindi, in generale: se $\det(A_{p,i}) = 0$, $i < n$, allora $a_{ii}^{(i)}$ è l'elemento pivotale $\neq 0 \rightarrow$ MEG si interrompe.

PIVOTING

Serve per "curare" il MEG se $\det(A_{p,i}) = 0$ per qualche $i < n$.

Idea: Scambio riga i con riga k in $A^{(i)}$ purché $k > i$ e $a_{ki}^{(i)} \neq 0$ (scambio con riga sotto)

In questo modo il nuovo (dopo lo scambio) $a_{ii}^{(i)} \neq 0$ e quindi il MEG procede.

Di fatto il pivoting corrisponde a pre-moltiplicare A e \underline{b} per una matrice di permutazione \underline{P}

Esempio:

$$A = \begin{pmatrix} 1 & 1 & 3 \\ 2 & 2 & 2 \\ 3 & \boxed{6} & 4 \end{pmatrix} \quad \begin{matrix} \text{scambio} \\ \text{riga 2} \\ \text{con riga 3} \end{matrix} \Rightarrow A = \begin{pmatrix} 1 & 1 & 3 \\ 3 & 6 & 4 \\ 2 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} A$$

In generale, se scambio riga i con riga $k > i$, P è l'identità ove la riga i è scambiata con la riga k . Quindi si ha: $A \underline{x} = \underline{b}$

$$\downarrow \quad \text{FATT LU (MEG) è su } PA: \\ PA \underline{x} = P \underline{b}$$

$$\downarrow \quad \begin{cases} L \underline{y} = P \underline{b} \\ U \underline{x} = \underline{y} \end{cases} \quad \begin{matrix} 2 \text{ sistemi} \\ \text{triangolari} \end{matrix}$$

Come implemento il pivoting? Non posso calcolare tutti i determinanti, ma inserisco un ciclo IF che verifichi se $a_{ki} \neq 0$ (e poi scambia).

PROPAGAZIONE ERRORI DI ARROTONDAMENTO

Il calcolatore non memorizza i numeri in quanto tali, bensì una loro approssimazione, perché dispone di un numero finito di celle di memoria.

numero \underline{z} → memorizzato come $\hat{\underline{z}} = \underline{z} + \delta \underline{z}$
errore di arrotondamento

Il calcolatore usa **aritmetica finita**, affetta da errori di arrotondamento. Invece, l'**aritmetica esatta** esiste "nella mia testa", senza errori di arrotondamento.

(es: somma $\underline{x} = \underline{z} + \underline{y} \rightarrow \hat{\underline{x}} = \hat{\underline{z}} + \hat{\underline{y}} = \underline{z} + \delta \underline{z} + \underline{y} + \delta \underline{y}$)

Svolgendo un numero elevato ($\sim n^3$) di operazioni, gli E di A δz possono propagare, diventando rilevanti.

Tornando al MEG : $A\underline{x} = \underline{b} \rightarrow (A + \delta A)\hat{\underline{x}} = (\underline{b} + \delta \underline{b})$
soluzione in
aritmetica esatta soluzione in aritmetica
finita, al calcolatore

Domanda: \underline{x} e $\hat{\underline{x}}$ sono simili?

Introduco discrepanza $\delta \underline{x} = \hat{\underline{x}} - \underline{x}$

Al fine di quantificare $\delta \underline{x}$, introduco notazione :

- **NORMA EUCLIDEA** $\underline{v} \in \mathbb{R}^n : \|\underline{v}\|_2 = \sqrt{\sum_{j=1}^n v_j^2}$
- **NORMA MATRICE** $\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$
autovalore max
- **NUMERO DI CONDIZIONAMENTO** : $K_2(A) = \|A\|_2 \|A^{-1}\|_2$

TEOREMA : Vale la seguente stima per la risoluzione di $A\underline{x} = \underline{b}$

$$\frac{\|\delta \underline{x}\|_2}{\|\underline{x}\|} \leq \frac{K_2(A)}{1 - K_2(A)} \frac{\|\delta A\|_2}{\|A\|_2} \left(\frac{\|\delta \underline{b}\|_2}{\|A\|_2} + \frac{\|\delta \underline{b}\|_2}{\|\underline{b}\|} \right)$$

Il fattore $\frac{K_2(A)}{1 - K_2(A)} \frac{\|\delta A\|_2}{\|A\|_2}$ amplifica gli E di A. È una funzione crescente di $K_2(A)$

$$(y(x) = \frac{x}{1 - \alpha x})$$

Se $K_2(A) \gg 1 \Rightarrow \|\delta \underline{x}\|_2$ è molto grande e quindi la soluzione del MEG non è affidabile

per stimare $K_2(A)$ (ricorda che non conosco A^{-1} !) Si utilizzano algoritmi specifici.
 Se però A è simmetrica definita positiva, allora $K_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$

Riferendoci allo schema iniziale (del corso) noto che il MEG è un metodo esatto in aritmetica esatta, cioè non commette alcun errore numerico → in A.E. mi da' \underline{x} . Tuttavia, in aritmetica finita, mi da' $\hat{\underline{x}}$ perché ho un errore computazionale.
(Propagazione Edi A.)

Non si tratta di un errore dovuto alla discretizzazione (che infatti non avviene in questo caso) ma è dovuto al fatto che il calcolatore approssima i numeri male.

STABILITÀ'

$$A \underline{x} = \underline{b} \quad \longrightarrow \quad (A + \delta A) \hat{\underline{x}} = \underline{b} + \delta \underline{b}$$

Introduco il residuo : $\underline{r} = \underline{b} - A \hat{\underline{x}}$ ($\hat{\underline{x}}$ non soddisfa il sistema lineare)

Noto che $\underline{0} = \underline{b} - A \underline{x}$ (la soluzione esatta non lascia residuo)

sottraggo a membro : $A(\underline{x} - \hat{\underline{x}}) = \underline{r}$

↓

$$\delta \underline{x} = A^{-1} \underline{r}$$

↓

$$(1) \quad \|\delta \underline{x}\|_2 = \|A^{-1}\| \|\underline{r}\|_2$$

Risultato generale

$$B \in \mathbb{R}^{n \times n}, \underline{v} \in \mathbb{R}^n$$

$$\|B \underline{v}\|_2 \leq \|B\|_2 \|\underline{v}\|_2$$

$$\underline{b} = A \underline{x} \rightarrow \|\underline{b}\|_2 = \|A \underline{x}\|_2 \leq \|A\|_2 \|\underline{x}\|_2$$

↓

$$(2) \quad \frac{1}{\|\underline{b}\|_2} \leq \frac{\|A\|_2}{\|\underline{b}\|_2}$$

$$(1) + (2) : \frac{1}{\|\underline{x}\|_2} \|\delta \underline{x}\|_2 \leq \frac{\|A^{-1}\| \|A\|_2}{K_2(A)} \cdot \frac{\|\underline{r}\|_2}{\|\underline{b}\|_2}$$

$$\frac{\|\delta \underline{x}\|_2}{\|\underline{x}\|_2} \leq K_2(A) \frac{\|\underline{r}\|_2}{\|\underline{b}\|_2}$$

Quindi, il residuo mi dà una concreta stima di $\delta \underline{x}$ a patto che $K_2(A)$ sia piccolo.
↳calcolabile

Esempio : matrice di Hilbert (vedi slide)

PIVOTING (ripresa)

Osservazione : fra tutte le operazioni, quella che propaga maggiormente l'EdiA. è la sottrazione (poiché può generare i "falsi zero").

Nel MEG ha una sottrazione : $a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)}$.

Vorrei quindi l_{ik} piccolo, ma $l_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$

Vorrei quindi $a_{kk}^{(k)}$ grande.

Quindi, anche se non ce ne fosse bisogno ($a_{kk}^{(k)} \neq 0$), opero lo scambio di righe.

Scambio riga k con riga $r > k$ tale che

$$a_{rr}^{(k)} = \max_{j \geq k} a_{jk}^{(k)}$$

Questo limita la propagazione degli errori di arrotondamento.

PIVOTING TOTALE

Idea: estendo la ricerca del massimo valore da usare come nuovo $a_{kk}^{(k)}$ anche alle colonne a destra:
 scambio riga k con riga r e colonna k con colonna q .
 tali che :
$$a_{rq}^{(k)} = \max_{i,j \geq k} a_{ij}^{(k)}$$

Ciò equivale a : $P A Q = L U$ FATT. $L U$ è fatta su matrice permutata
 ↑
 permutazione colonne

$$\begin{aligned}
 A\underline{x} = \underline{b} &\longrightarrow PA\underline{x} = P\underline{b} \\
 &\downarrow \\
 \boxed{PAQ} \quad \boxed{Q^{-1}\underline{x}} &= P\underline{b} \\
 LU &\qquad\qquad\qquad x^* \\
 PAQ \quad \underline{x}^* &= P\underline{b} \\
 &\downarrow \\
 LU \underline{x}^* &= P\underline{b} \quad \longrightarrow \quad \begin{cases} L\underline{y} = P\underline{b} \\ U\underline{x}^* = \underline{y} \\ x = Q\underline{x}^* \end{cases}
 \end{aligned}$$

Nota: il Pivoting totale limita al massimo gli effetti di propagazione degli errori di arrotondamento dovuti a $K_2(A) \gg 1$. Tuttavia, se $K_2(A) \gg 1$ rimangono problemi.

Il **mal condizionamento** di una matrice può essere limitato ma non rimosso.

FILL-IN

Introduciamo il concetto di pattern di una matrice B : rappresentazione grafica per evidenziare elementi non nulli, per cui si mette un segno in posizione ij se $B_{ij} \neq 0$

(x)

DEF: Una matrice B è **sparsa** se il numero degli elementi nulli è molto maggiore del numero degli elementi non-nulli, cioè se:

$$\lim_{n \rightarrow +\infty} \frac{\#\text{ELEM} \neq 0}{n^2} = 0 \quad \text{cioè se : } \#\text{ELEM} \neq 0 = \mathcal{O}(n) \text{ (circa } n)$$

Un $\Theta(k)$ ("O-grande") è qualcosa che cresce almeno tanto velocemente quanto k

Esempio : B TRIDIAGONALE

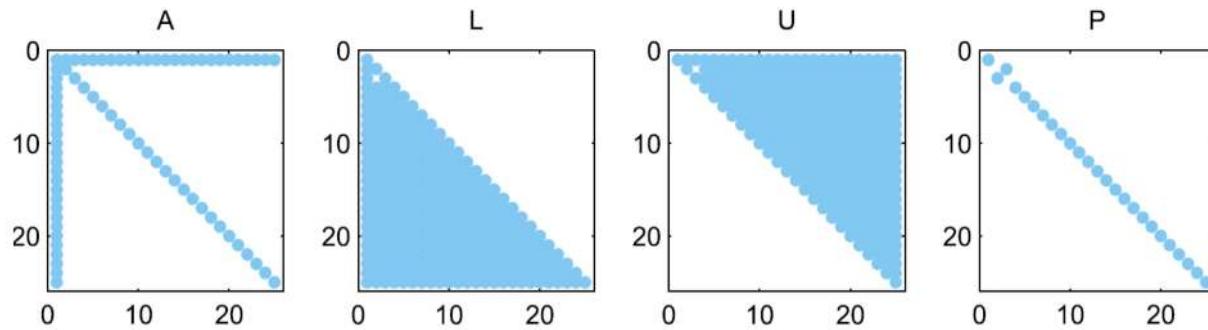
$$B = \begin{pmatrix} & & & 0 \\ & & & 0 \\ & & & 0 \\ & & & 0 \\ & & & 0 \end{pmatrix}$$

es: di pattern

$$\# \text{ELEM} \neq 0 = n + 2(n-1) = 3n - 2 \sim n$$

↳ B sparse • $n = 100$ =

DEF: Il fill-in consiste nel fatto che i fattori L e U di A sparsa non sono sparsi.



L'implementazione del MEG è fatta in modo da memorizzare L e U sovrascrivendo sullo spazio dedicato ad A.

Perciò, l'occupazione di memoria allocata per $A(\sim n)$ è insufficiente per memorizzare L e U che richiedono n^2 celle di memoria.

Il MEG è altamente sconsigliabile per matrici sparse \rightarrow FATT. LU non adatta dal punto di vista dell'occupazione di memoria.

2B. METODI ITERATIVI PER SISTEMI LINEARI

MEG / FATT LU (con pivoting) ha qualche limitazione :

(1) NON efficiente per n molto grande :

es : $n = 10^7$, calcolatore con 10^{15} flops/s
 $\# \text{flops} = 2/3 n^3 \rightarrow \sim 11$ giorni

problemi di efficienza per n grande

(2) Se $K_2(A) \gg 1 \rightarrow$ Grossi errori

problemi di accuratezza

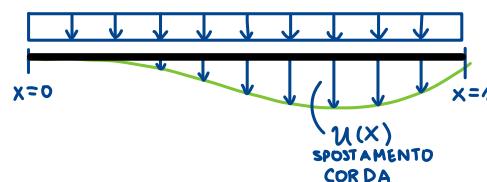
(3) Se A sparsa \rightarrow Fill-in

problemi di memoria

Sono metodi alternativi al MEG, che è un metodo esatto nonostante presenti dei problemi dovuti alla struttura della matrice, mentre i metodi iterativi non sono metodi esatti (bisognerebbe iterare all'infinito per avere una soluzione esatta)

Esempio : matrice sparsa (eventualmente di grosse dimensioni) e mal condizionata

- CORDA ELASTICA 1D
- FISSA AGLI ESTREMI
- SOGGETTA A FORZA f

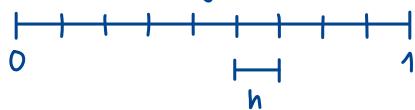


dalla II legge di Newton:

$$\begin{cases} -u''(x) = f(x) & x \in (0,1) \\ u(0) = u(1) = 0 \end{cases}$$

Introduco approssimazione :

$$x_j = jh; j = 0, \dots, N$$



divido il dominio in nodi

scrivo l'equazione approssimata nei nodi

$$-u''(x_j) = f(x_j) \Rightarrow -\frac{U_{j+1} - 2U_j + U_{j-1}}{h^2} = f(x_j)$$

$$U_j \approx u(x_j)$$

$$\sim -u''$$

$$\forall j$$

Riscrivo l'equazione $A\mathbf{x} = \mathbf{b}$ e ottengo: $\frac{1}{h^2} A \mathbf{U} = \mathbf{F}$

A sparsa e con $K_2(A) \sim \frac{1}{h^2} \gg 1$

$$A = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

Le matrici sparse hanno grande utilità in campo ingegneristico e le devo saper risolvere → introduco metodi alternativi al MEG.

Introduco metodi iterativi:

parto da $\underline{x}^{(0)}$ e introduco una legge di aggiornamento che genera la successione $\underline{x}^{(k)}$

$$\underline{x}^{(0)} \xrightarrow{\text{LEGGE}} \underline{x}^{(1)} \xrightarrow{\text{LEGGE}} \underline{x}^{(2)} \xrightarrow{\text{LEGGE}} \dots \longrightarrow \underline{x}^{(k)} \longrightarrow$$

speranza è che metodo converga:

$$\lim_{k \rightarrow +\infty} \underline{x}^{(k)} = \underline{x}$$

Il processo iterativo si arresterà quando un opportuno criterio d'arresto sarà soddisfatto.

Quindi, i metodi iterativi commettono un errore $\underline{x} - \underline{x}^{(K_{\text{FINALE}})}$ anche in aritmetica esatta → c'è **errore numerico** (a differenza di FATT LU).

METODO DI JACOBI

$A \underline{x} = \underline{b} \rightarrow$ i-esima riga: $\sum_{j=1}^n a_{ij} x_j = b_i$

$$x_i = \frac{b_i - \sum_{j \neq i} a_{ij} x_j}{a_{ii}}$$

vera ma inutilizzabile:
 x_j non noti

Ciò suggerisce la seguente legge di aggiornamento:

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

Dato $\underline{x}^{(0)}$, trovo $\underline{x}^{(1)}$, da cui $\underline{x}^{(2)}, \dots$
↳ lo conosco perché lo decido io

↳ posso anche farlo in parallelo su n processori

- **CONVERGENZA** (vedi in seguito)
- **MEMORIA**: la stessa di $A + 3$ vettori ($\underline{x}^{(k+1)}, \underline{x}, \underline{b}$)
→ Jacobi beneficia di sparsità (esente da FILL-IN)
- **EFFICIENZA**: ogni iterata consiste di n steps (il calcolo di $x_i^{(k+1)}$) ognuno dei quali costa: $1 + 1 + n - 1 + n - 2 \sim n$
→ Ogni iterazione costa $\sim n^2$
Jacobi è competitivo con MEG se **#ITER << n** (così costo totale $< \frac{2}{3} n^3$)

se A è sparsa, ogni step per calcolare x_i costa $1 + 1 + 1 \sim 1$
 → costo singola iterazione $\sim n$
 → Jacobi super competitivo anche se faccio n iterazioni

ASINTOTICO

Esempio :

$$B = \begin{pmatrix} & & a_{ii} & a_{i,i+1} \\ & & 0 & \\ & a_{i,i-1} & & \\ & 0 & & \end{pmatrix}_i$$

Jacobi :

$$x_i^{(k+1)} = \frac{b_i - \sum_{j \neq i} a_{ij} x_j^{(k)}}{a_{ii}} = \frac{b_i - a_{i,i-1} x_{i-1}^{(k)} - a_{i,i+1} x_{i+1}^{(k)}}{a_{ii}}$$

$$\rightarrow \text{OPER} = 1 + 1 + 3 \sim 1$$

Esempio :

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 3 & 6 & 1 \\ 1 & 2 & 4 \end{pmatrix}; \quad \underline{x} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}; \quad \underline{b} = A \underline{x} = \begin{pmatrix} 3 \\ 10 \\ 7 \end{pmatrix}$$

$$\underline{x}^{(0)} = \begin{pmatrix} 1/2 \\ 1/2 \\ 1/2 \end{pmatrix} \xrightarrow{\text{JACOBI}} x_1^{(1)} = \frac{b_1 - a_{12} x_2^{(0)} - a_{13} x_3^{(0)}}{a_{11}} = \frac{3 - 1 \cdot 1/2 - 0 \cdot 1/2}{2} = 5/4$$

$$x_2^{(1)} = \frac{b_2 - a_{21} x_1^{(0)} - a_{23} x_3^{(0)}}{a_{22}} = \frac{10 - 3 \cdot 1/2 - 1 \cdot 1/2}{6} = 4/3$$

$$\underline{x}^{(1)} = \begin{pmatrix} 5/4 \\ 4/3 \\ 11/8 \end{pmatrix} \xleftarrow{} x_3^{(1)} = \frac{b_3 - a_{31} x_1^{(0)} - a_{32} x_2^{(0)}}{a_{33}} = \frac{7 - 1 \cdot 1/2 - 2 \cdot 1/2}{4} = 11/8$$

$$\xrightarrow{} x_1^{(2)} = \frac{b_1 - a_{12} x_2^{(1)} - a_{13} x_3^{(1)}}{a_{11}} = \frac{3 - 1 \cdot 4/3 - 0 \cdot 11/8}{2} = \frac{5}{6}$$

$$x_2^{(2)} = \frac{b_2 - a_{21} x_1^{(1)} - a_{23} x_3^{(1)}}{a_{22}} = \frac{10 - 3 \cdot 5/4 - 1 \cdot 11/8}{6} = \frac{13}{16}$$

$$\underline{x}^{(2)} = \begin{pmatrix} 5/6 \\ 13/16 \\ 35/48 \end{pmatrix} \xleftarrow{} x_3^{(2)} = \frac{b_3 - a_{31} x_1^{(1)} - a_{32} x_2^{(1)}}{a_{33}} = \frac{7 - 1 \cdot 4/3 - 2 \cdot 11/8}{4} = \frac{35}{48}$$

GAUSS - SEIDEL

Jacobi : $x_i^{(k+1)} = \frac{b_i - \sum_{j \neq i} a_{ij} x_j^{(k)}}{a_{ii}}$ → spezzo in due contributi
noti

$$x_i^{(k+1)} = \frac{b_i - \sum_{j > i} a_{ij} x_j^{(k)} - \sum_{j < i} a_{ij} x_j^{(k+1)}}{a_{ii}}$$

Non è parallelizzabile,
a differenza di Jacobi

Non è detto
che sia meglio

ES : $A = \begin{pmatrix} 2 & 1 & 0 \\ 3 & 6 & 1 \\ 1 & 2 & 4 \end{pmatrix}$; $\underline{x} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ costruisco $b = A \underline{x} = \begin{pmatrix} 3 \\ 10 \\ 7 \end{pmatrix}$

JACOBI

$$\underline{x}^{(0)} = (1/2 \ 1/2 \ 1/2)^T$$

$$x_1^{(1)} = 5/4$$

$$x_2^{(1)} = 4/3$$

$$x_3^{(1)} = 11/8$$

GAUSS - SEIDEL

$$\underline{x}^{(0)} = (1/2 \ 1/2 \ 1/2)^T$$

$$x_1^{(1)} = 5/4$$

$$x_2^{(1)} = \frac{b_2 - a_{21} x_1^{(1)} - a_{23} x_3^{(0)}}{a_{22}} = \frac{10 - 3 \cdot 5/4 - 1 \cdot 1/2}{6} = \frac{27}{24}$$

$$x_3^{(1)} = \frac{b_3 - a_{31} x_1^{(1)} - a_{32} x_2^{(1)}}{a_{33}} = \frac{7 - 1 \cdot 5/4 - 2 \cdot 27/24}{4} = \frac{7}{8}$$

• Occupazione di memoria GS come Jacobi

PARALLELLIZZABILE

$$y = 2$$

$$\begin{cases} x = y + 3 \\ z = y - 4 \end{cases}$$

Parallelizzare un algoritmo vuol dire che invece di eseguire le operazioni in serie, divido le operazioni tra i vari processori e le faccio eseguire assieme.
È vantaggioso perché impiego meno tempo di elaborazione.

• costo computazionale GS come Jacobi

NON PARALLELLIZZABILE

$$\begin{cases} x = y + 3 \\ z = x + 4 \end{cases}$$

CONVERGENZA DI METODI ITERATIVI

Mi concentro sui metodi della famiglia

(*) $\underline{x}^{(k+1)} = \underline{\beta} \underline{x}^{(k)} + \underline{f}$, con $\underline{\beta} \in \mathbb{R}^{n \times n}$ e $\underline{f} \in \mathbb{R}^n$ opportuni
MATRICE DI ITERAZIONE

DOMANDA : quali sono le condizioni su $\underline{\beta}$ e \underline{f} che garantiscono la convergenza ($\underline{x}^{(k+1)} \xrightarrow{k \rightarrow \infty} \underline{x}$, con $A \underline{x} = \underline{b}$)?

- Scriviamo J e GS nella forma generica (*)

scompongo A :

$$A = \begin{pmatrix} & -F \\ \searrow D & \\ -E & \swarrow \end{pmatrix}$$

DIAGONALE
↓
 $A = D - E - F$
↑
TRIANGOLARE
INFERIORE

ES: $A = \begin{pmatrix} 3 & 2 & 1 \\ 0 & 4 & -7 \\ 2 & 1 & -1 \end{pmatrix}$;

$$D = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

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

$$F = \begin{pmatrix} 0 & -2 & -1 \\ 0 & 0 & 7 \\ 0 & 0 & 0 \end{pmatrix}$$

$$A\underline{x} = \underline{b} \rightarrow (D - E - F)\underline{x} = \underline{b}$$

$$D\underline{x} = (E + F)\underline{x} + \underline{b}$$

- Introduco metodo iterativo

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

- Nel caso di JACOBI abbiamo:

$$i\text{-esima riga: } a_{ii}x_i^{(k+1)} = b_i - \sum_{j \neq i} a_{ij}x_j^{(k)}$$

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

$$\rightarrow \text{Riconosciamo } B_J = D^{-1}(E+F), f_J = D^{-1}\underline{b}$$

- Nel caso di GAUSS-SEIDEL abbiamo:

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

$$\rightarrow \text{Riconosciamo } B_{GS} = (D - E)^{-1}F; f_{GS} = (D - E)^{-1}\underline{b}$$

- studiamo convergenza in generale di $\underline{x}^{(k+1)} = B\underline{x}^{(k)} + f$

1) Devo garantire la **consistenza**: metodo numerico applicato ad \underline{x} restituisce \underline{x} .
 Il metodo si deve accorgere che se ha la soluzione esatta non deve fare niente (non deve fare aggiornamenti).
 $\rightarrow \underline{x} = B\underline{x} + f$

Requisito: $f = (I - B)\underline{x} = (I - B)A^{-1}\underline{b}$ Relazione di consistenza fra f e B

Data una, ricavo l'altro. Non posso sceglierle entrambe arbitrariamente.

- 2) Devo garantire anche la **stabilità**

=> CONVERGENZA (generale) = consistenza + stabilità

→ Indago le condizioni di stabilità (l'errore non esplode)

- Introduco la norma: $\|\cdot\|: \mathbb{R}^n \rightarrow \mathbb{R}$

- Definisco errore: $\underline{e}^{(k)} = \underline{x} - \underline{x}^{(k)}$

- Convergenza: $\lim_{k \rightarrow +\infty} \underline{x}^{(k)} = \underline{x} \Leftrightarrow \lim_{k \rightarrow +\infty} \|\underline{e}^{(k)}\| = 0$

- Ottengo: $\underline{x}^{(k+1)} = B\underline{x}^{(k)} + \underline{f}$ + $\underline{f} = (I - B)\underline{x}$

metodo iterativo

consistenza

$$\begin{aligned} \|\underline{e}^{(k+1)}\| &= \|\underline{x} - \underline{x}^{(k+1)}\| = \|\underline{x} - B\underline{x}^{(k)} - \underline{f}\| = \|\underline{x} - B\underline{x}^{(k)} - (I - B)\underline{x}\| = \\ &= \|B(\underline{x} - \underline{x}^{(k)})\| = \|B\underline{e}^{(k)}\| \leq \|B\| \|\underline{e}^{(k)}\| \end{aligned}$$

Ricorsivamente: $\|\underline{e}^{(k+1)}\| \leq \|B\| \|\underline{e}^{(k)}\| \leq \|B\|^2 \|\underline{e}^{(k-1)}\| \leq \|B\|^3 \|\underline{e}^{(k-2)}\| \leq \dots$

vedi appunti del prof per capire perché compaiono le potenze sulla norma di B

$$\leq \dots \leq \|B\|^{k+1} \|\underline{e}^{(0)}\|$$

$\|\underline{x} - \underline{x}^{(0)}\|$ è sicuramente finito

Poiché $\underline{e}^{(0)}$ è fisso e finito, si ha: $\lim_{k \rightarrow \infty} \|\underline{e}^{(k+1)}\| \leq \left(\lim_{k \rightarrow \infty} \|B\|^{k+1} \right) \underline{e}^{(0)}$

La convergenza è garantita se: $\lim_{k \rightarrow \infty} \|B\|^{k+1} = 0 \Leftrightarrow \|B\| < 1$

Noto che è sufficiente considerare una norma $\|\cdot\|$ che garantisca $\|B\| < 1$

condizione sufficiente di convergenza ←

Di solito, come norma, si usa il **RAGGIO SPETTRALE** $\|B\| = \rho(B) = \max_j |\lambda_j(B)|$

il massimo dei moduli degli autovalori della matrice B

Proprietà: per ogni matrice $C \in \mathbb{R}^{n \times m}$ si ha $\rho(C) \leq \|C\|$ per qualsiasi altra norma $\|\cdot\|$

Quindi: (i) se $\|C\| < 1 \rightarrow \rho(C) < 1$ (se $\exists \|\cdot\|$)

(ii) se $\rho(C) > 1 \rightarrow \|C\| > 1 \quad \forall \|\cdot\|$

$$\underline{f} = (I - B)\underline{x}$$

$$\rho(B) < 1$$

$$\lim_{k \rightarrow \infty} \underline{x}^{(k)} = \underline{x}$$

condizione necessaria e sufficiente di convergenza

assieme a
consistenza (*)

discutiamo convergenza di Jacobi

- verifichiamo consistenza : $B_J = D^{-1}(E+F)$; $f_J = D^{-1}\underline{b}$ $\rightarrow f_J = (I - B_J)A^{-1}\underline{b}$?

$$D^{-1}\underline{b} = (I - D^{-1}(E+F))A^{-1}\underline{b}$$

$$= (I - D^{-1}(D-A))A^{-1}\underline{b} =$$

$$= (I - I + D^{-1}A)A^{-1}\underline{b} = D^{-1}\underline{b}$$

OK!

- convergenza di Jacobi : $\rho(B_J) < 1 \quad \downarrow$

$$\max_j |\lambda_j(I - (D^{-1}A))| < 1$$

In generale non abbiamo metodi che convergono o no a priori, ma la convergenza dipende dalla matrice A presa in considerazione.

\rightarrow convergenza di J dipende da A

$$\text{Es: } A = \begin{pmatrix} 3 & -2 \\ 7 & 5 \end{pmatrix} ; \quad \underline{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} ; \quad \underline{b} = \begin{pmatrix} 1 \\ 12 \end{pmatrix}$$

$$\begin{aligned} B_J &= D^{-1}(E+F) \\ &= D^{-1}(D-A) \\ &= I - D^{-1}A \end{aligned}$$

Verifico se $\rho(B_J) < 1$ (NOTA: J sempre consistente!)

$$D^{-1}A = \begin{pmatrix} 1/3 & 0 \\ 0 & 1/5 \end{pmatrix} \begin{pmatrix} 3 & -2 \\ 7 & 5 \end{pmatrix} = \begin{pmatrix} 1 & -2/3 \\ 7/5 & 1 \end{pmatrix}$$

diagonale
con i reciproci
(è semplice)

$$I - D^{-1}A = \begin{pmatrix} 0 & 2/3 \\ -7/5 & 0 \end{pmatrix}$$

$$\lambda_{1,2} \Rightarrow \det(\lambda I - B_J) = \det \left[\begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} - \begin{pmatrix} 0 & 2/3 \\ -7/5 & 0 \end{pmatrix} \right] = \det \begin{pmatrix} \lambda & -2/3 \\ 7/5 & \lambda \end{pmatrix} = \lambda^2 + \frac{14}{15}$$

$$\Rightarrow \lambda_{1,2} = \pm i\sqrt{\frac{14}{15}} \Rightarrow |\lambda_{1,2}| = \sqrt{\frac{14}{15}} < 1 \quad \longrightarrow \boxed{\text{J converge}}$$

discutiamo convergenza di Gauss-Seidel

Analogamente trovo che GS è consistente; è inoltre convergente $\Leftrightarrow \rho(B_{GS}) < 1$

$$\max_j |\lambda_j((D-E)^{-1}F)| < 1$$

CONDIZIONI SUFFICIENTI DI CONVERGENZA J e GS

La condizione sul raggio spettrale appena discusa è una Condizione Necessaria e Sufficiente di convergenza. Tuttavia, richiede il calcolo di un determinante per arrivare agli autovalori, il che può essere oneroso. Vediamo delle condizioni alternative, più immediate, ma solo sufficienti per la convergenza.

i) Matrici strettamente dominanti diagonali per righe:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \quad \forall i \quad \rightarrow J, GS \text{ convergono}$$

ii) Matrici strettamente dominanti diagonali per colonne:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \quad \forall i \quad \rightarrow J, GS \text{ convergono}$$

iii) Matrici simmetriche definite-positive: $\forall z \neq 0 \rightarrow z^T A z > 0 \rightarrow GS \text{ converge}$

iv) Matrici tridiagonali: due alternative:

$$\begin{pmatrix} & & 0 \\ & \diagup & \\ 0 & \diagdown & \end{pmatrix}$$

- non convergono né J né GS
oppure
- convergono entrambi e GS più veloce

Esempio: $\begin{pmatrix} 2 & 1 & 0 \\ 3 & 6 & 1 \\ 0 & 2 & 4 \end{pmatrix}$

- (i) è a dominanza diagonale stretta per righe $\rightarrow J, GS \text{ convergono}$
- (ii) non è a dominanza diagonale stretta per colonne
 \rightarrow nessuna implicazione
- (iv) TRIDIAGONALE \rightarrow assieme a i $\rightarrow GS \text{ più veloce}$

Dato $\underline{x}^{(k)}$, definisco la seguente quantità:

$$\underline{r}^{(k)} = \underline{b} - A\underline{x}^{(k)} \rightarrow \text{Residuo all' iterata } k$$

Nota: il residuo è di fatto calcolabile (a differenza dell' errore)

$$\text{Noto che se per caso } \underline{x}^{(k)} = \underline{x} \rightarrow \underline{r}^{(k)} = \underline{b} - A\underline{x} = 0$$

\Rightarrow In corrispondenza della soluzione esatta, il residuo è nullo.

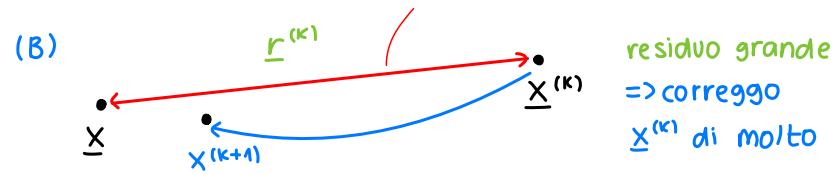
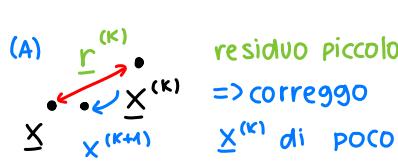
Più in generale: essendo il problema lineare, posso affermare che

- più sono vicini ad \underline{x} , più \underline{r} è piccolo;
- più sono lontano ad \underline{x} , più \underline{r} è grande.

Allora, introduco il seguente metodo iterativo: dato $\underline{x}^{(0)}$ e $\alpha \in \mathbb{R}$:

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha \underline{r}^{(k)} \quad \text{Richardson stazionario}$$

uso il residuo come fattore di iterazione



$$\text{Equivalentemente: } \underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha \frac{(\underline{b} - A\underline{x}^{(k)})}{\underline{r}^{(k)}} = (\underline{I} - \alpha A) \underline{x}^{(k)} + \alpha \underline{b}$$

$\underline{B}\alpha$ MATRICE DI ITERAZIONE $\underline{f}\alpha$

\rightarrow segue lo schema generale

cioè mi permette di studiare la convergenza di Richardson stazionario. D'ora in avanti mi concentro su matrici simmetriche definite positive.

1. consistenza: $\underline{f}\alpha = (\underline{I} - \underline{B}\alpha) \underline{x} ?$
 $\alpha \underline{b} = (\underline{I} - \underline{I} + \alpha A) \underline{x}$
 $\underline{b} = A\underline{x}$ OK

2. stabilità: $\rho(\underline{B}\alpha) < 1 ?$

$$\max_j |\lambda_j(\underline{B}\alpha)| < 1 \leftrightarrow \boxed{\max_j |\lambda_j(\underline{I} - \alpha A)| < 1} \leftrightarrow \max_j |1 - \alpha \lambda_j(A)| < 1$$

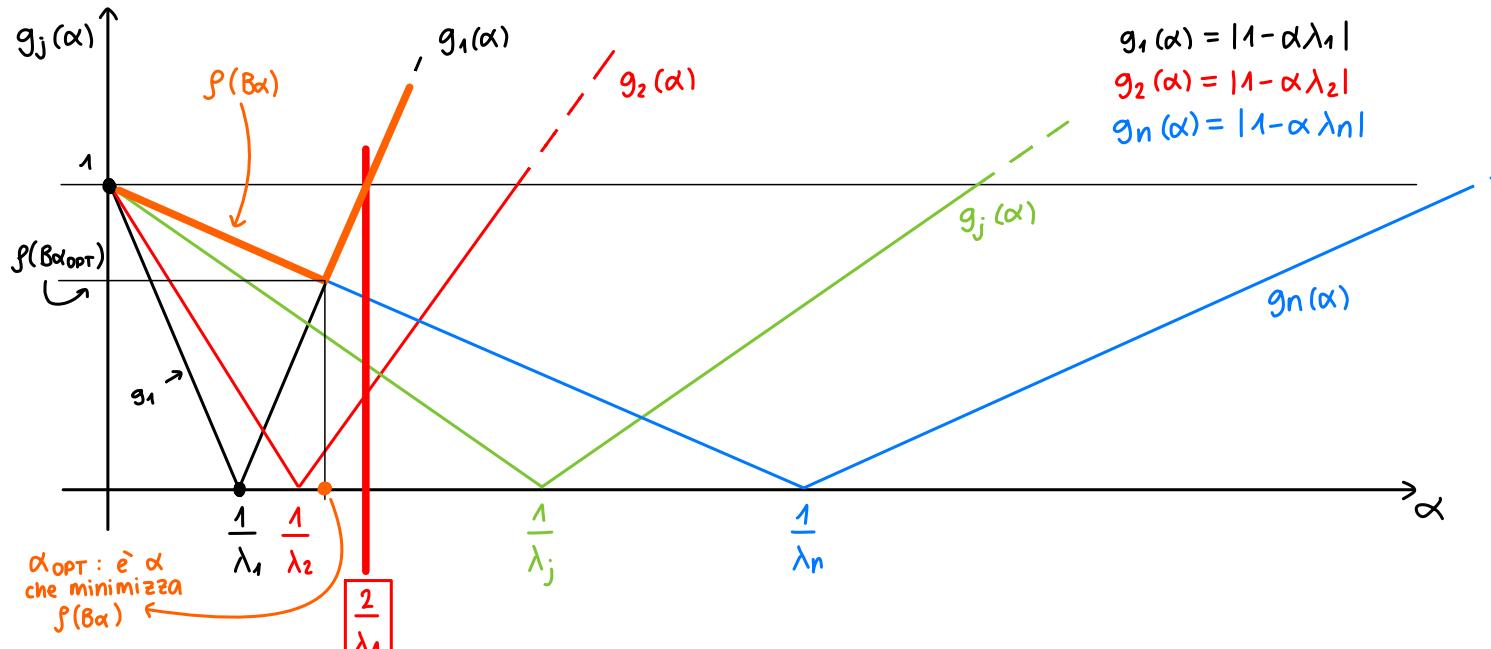
$\Leftrightarrow \lambda_j(A) \in \mathbb{R}^+$ $\rightarrow \boxed{|1 - \alpha \lambda_j(A)| < 1 \quad \forall j = 1, \dots, n}$

Quali valori di $\alpha \in \mathbb{R}$ soddisfano queste relazioni?

Ordino i $\lambda_j(A)$ come segue: $\lambda_{\max} = \lambda_1(A) \geq \lambda_2(A) \geq \dots \geq \lambda_n(A) = \lambda_{\min} > 0$

chiamo $g_j(\alpha) = |1 - \alpha \lambda_j(A)|$, $j = 1, \dots, n$ (sono tante funzioni di α)

Disegno le $g_j(\alpha)$ e guardo quando sono tutte < 1 .



Noto che $g_j(\alpha) < 1 \quad \forall j$ quando $\alpha < \frac{2}{\lambda_{\max}}$

Effettivamente si ha che Richardson stazionario converge solo se $0 < \alpha < \frac{2}{\lambda_{\max}}$

Mi chiedo ora quale sia fra tutti gli $\alpha < 2/\lambda_{\max}$ quello che massimizza la velocità di convergenza.

Ricordo che $\|e^{(k+1)}\| < \rho(B\alpha) \|e^{(k)}\|$ (v. lez precedente : $\|B\| = \rho(B)$)

\Rightarrow cerco α che minimizza $\rho(B\alpha)$

$$\rho(B\alpha) = \max_j |\lambda_j(B\alpha)| = \max_j |1 - \alpha \lambda_j(A)| = \max_j (g_j(\alpha))$$

α_{opt} è il punto di intersezione fra $g_1(\alpha)$ e $g_n(\alpha)$:

$$1 - \alpha \lambda_n = \alpha \lambda_1 - 1 \quad (\text{occhio alle pendenze delle rette!})$$

$$\alpha_{\text{opt}} = \frac{2}{\lambda_{\max} + \lambda_{\min}} \quad \begin{array}{l} \text{Miglior } \alpha, \text{ che garantisce} \\ \text{massima velocità di convergenza} \end{array}$$

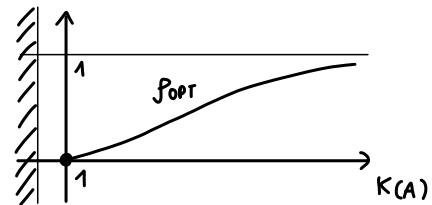
$$\text{Calcolo } \boxed{\rho(B\alpha_{\text{opt}}) = 1 - \alpha_{\text{opt}} \lambda_{\min} = 1 - \frac{2 \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}} \quad \begin{array}{l} \text{più piccolo } \rho(B\alpha) \\ \text{possibile} \end{array}$$

La velocità max dipende da A . In particolare si ha che per matrici SDP,

$$K(A) = \frac{\lambda_{\max}}{\lambda_{\min}} \rightarrow \boxed{\rho_{\text{opt}} = \rho(B\alpha_{\text{opt}}) = \frac{K(A) - 1}{K(A) + 1}}$$

Quindi, matrici A malcondizionate ($K(A) \gg 1$) hanno un ρ_{opt} "grande", molto vicino a 1.

\hookrightarrow converge ma a fatica



Nota: al limite se $K(A) \approx 1 \rightarrow \rho_{\text{opt}} \approx 0 \rightarrow$ convergenza in 1 iterazione. In generale, la convergenza è tanto più veloce tanto più gli autovalori di A sono confinati in una piccola regione.

PRECONDIZIONAMENTO → possiamo migliorare le cose se $K(A)$ è molto grande?

$$\rho_{\text{OPT}} = \frac{K(A) - 1}{K(A) + 1} ; \quad \text{IDEA: INTRODUKO } P \in \mathbb{R}^{n \times n} \text{ SDP invertibile e al posto di risolvere } A \underline{x} = \underline{b}, \text{ risOLVO } P^{-1}A \underline{x} = P^{-1}\underline{b} \quad (*)$$

Noto che la soluzione \underline{x} di $(*)$ è la stessa del sistema originario
→ Applico Richardson stazionario a $(*)$.

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

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha \underline{r}^{(k)} \rightarrow \text{AGGIORNAMENTO}$$

$$\underline{r}^{(k)} = \underline{b} - A \underline{x}^{(k)} \rightarrow \text{RESIDUO}$$

$$\alpha < \frac{2}{\lambda_{\max}(A)} \rightarrow \text{CONVERGENZA}$$

$$\alpha_{\text{OPT}} = \frac{2}{\lambda_{\min}(A) + \lambda_{\max}(A)}$$

$$\rho_{\text{OPT}} = \frac{K(A) - 1}{K(A) + 1}$$

$$P^{-1}A \underline{x} = P^{-1}\underline{b}$$

SISTEMA PRECONDIZIONATO

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha \underline{z}^{(k)}$$

$$\underline{z}^{(k)} = P^{-1}\underline{b} - P^{-1}A \underline{x}^{(k)}$$

$$\alpha < \frac{2}{\lambda_{\max}(P^{-1}A)}$$

$$\alpha_{\text{OPT}} = \frac{2}{\lambda_{\min}(P^{-1}A) + \lambda_{\max}(P^{-1}A)}$$

$$\rho_{\text{OPT}} = \frac{K(P^{-1}A) - 1}{K(P^{-1}A) + 1}$$

Noto che se $K(P^{-1}A) \ll K(A)$ → Numero di iterazioni per convergenza si è molto ridotto. Qual è il prezzo da pagare?

METODO DI RICHARDSON STAZIONARIO PRECONDIZIONATO

Dato $\underline{x}^{(0)} \in \mathbb{R}^n$, per ogni $k \geq 0$:

$$(i) \alpha_{\text{OPT}} = \frac{2}{\lambda_{\max}(P^{-1}A) + \lambda_{\min}(P^{-1}A)}$$

Lo uso per calcolare \underline{z}

$$(ii) \underline{r}^{(k)} = \underline{b} - A \underline{x}^{(k)} \quad \text{RESIDUO non precondizionato}$$

$$(iii) \underline{z}^{(k)} = P^{-1}\underline{b} - P^{-1}A \underline{x}^{(k)} = P^{-1}(\underline{b} - A \underline{x}^{(k)}) \quad \text{non conosco l'inverso di } P!$$

$$\underline{z}^{(k)} = P^{-1}\underline{r}^{(k)} \rightarrow P \underline{z}^{(k)} = \underline{r}^{(k)} \rightarrow \text{Extra costo rispetto a Richardson non precondizionato: devo risolvere un altro sistema lineare prima}$$

La scelta di P deve cercare di soddisfare i seguenti due requisiti:

↳ dovrò stare "in between"

1) P "buona immagine" di A . Infatti solo così ho che $K(P^{-1}A) \ll K(A)$
→ # ITERAZIONI si abbassa

Limite estremo: $P = A \rightarrow K(P^{-1}A) = K(I) = 1 \rightarrow \rho_{\text{OPT}} = 0$ MA GROSSO extra-costo

2) P "FACILE DA RISOLVERE": così calcolo (extra-costo) di $\underline{z}^{(k)}$ è poco oneroso: $P \underline{z}^{(k)} = \underline{r}^{(k)}$

Limite estremo: $P = I$ NO extra-costo MA $K(A)$ non si abbassa

(1) PRECONDIZIONATORE DI JACOBI

$$P = D \quad \text{MATRICE DIAGONALE DI } A$$

$$\Rightarrow K(D^{-1}A) \ll K(A) \Rightarrow \# \text{ ITERAZIONI per arrivare a CONVERGENZA è RIDOTTO}$$

$$\begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & a_{jj} & \dots & a_{nn} \end{pmatrix} \rightarrow \text{RISOLUZIONE IMMEDIATA: le equazioni sono disaccoppiate}$$

RICHARDSON CON $P = J$:

$$(i) \alpha_{\text{OPT}} = \frac{2}{\lambda_{\max}(D^{-1}A) + \lambda_{\min}(D^{-1}A)} \sim n^2 \quad (n \text{ se } A \text{ è sparsa})$$

$$(ii) \underline{r}^{(k)} = \underline{b} - A\underline{x}^{(k)} \sim n^2 \quad (n \text{ se } A \text{ è sparsa})$$

$$(iii) D\underline{z}^{(k)} = \underline{r}^{(k)} \sim n \rightarrow \text{EXTRA-COSTO ACCETTABILE (rispetto a Richardson normale)}$$

$$(iv) \underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha_{\text{OPT}} \underline{z}^{(k)} \sim n$$

↓
COSTO DELLA SINGOLA
ITERAZIONE ACCETTABILE

(2) LU INESATTA (vale solo per A sparse)

Fill-in : Se A è sparsa \rightarrow L e U in FATT. LU non sono sparsi

Idea: Introduciamo un MEG "modificato" che preservi la sparsità : cioè che fissi a priori $l_{ij}=0$ o $u_{ij}=0$ laddove $a_{ij}=0$.

Ovviamente, alla fine non ottengo L e U : $A = LU$, bensì \tilde{L} e \tilde{U} triangolari; inf e sup e sparse.

$\Rightarrow \tilde{L}\tilde{U} \approx A$ perché \tilde{L} e \tilde{U} sono ottenute a partire da A.

\Rightarrow USO $P = \tilde{L}\tilde{U}$. Mi aspetto che $K((\tilde{L}\tilde{U})^{-1}A) \ll K(A) \Rightarrow \# \text{ ITERAZIONI DIMINUISCE}$

Extra-costo : $(\tilde{L}\tilde{U})\underline{z}^{(k)} = \underline{r}^{(k)} \sim n$ (sostituzioni in avanti e indietro con \tilde{L} e \tilde{U} sparse)

È un metodo più evoluto di quello di Jacobi: non mi limito a prendere la diagonale, ma devo fare un MEG modificato. È più complesso ma in generale è anche più efficiente.

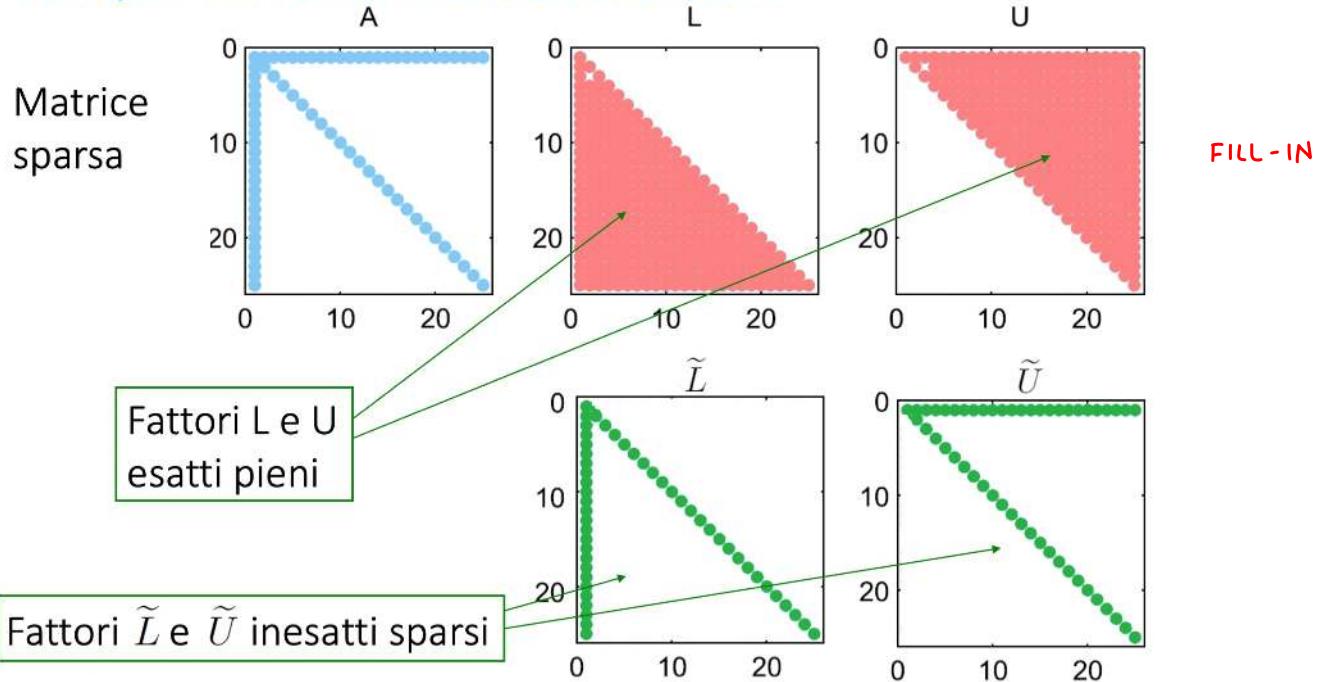
MEG Modificato :

```

if k=1, ..., n-1
  if i=k+1, ..., n
    if i, k ≠ 0
       $l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ 
    else
       $l_{ik} = 0$  [...]
  
```

Extra-costo $\rightarrow \tilde{L}\tilde{U}\underline{z}^{(k)} = \underline{r}^{(k)} \rightarrow \begin{cases} \tilde{L}\underline{y}^{(k)} = \underline{r}^{(k)} \\ \tilde{U}\underline{z}^{(k)} = \underline{y}^{(k)} \end{cases}$

Esempio di Fattorizzazione LU iesatta:



(3) METODO DEL GRADIENTE (Richardson dinamico) :

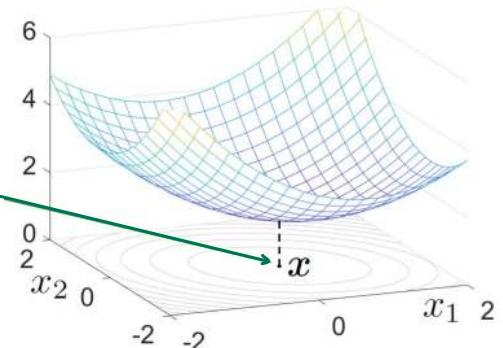
Analizziamo il caso di A SDP. Introduco un'energia associata ad $A\underline{x} = \underline{b}$:

$$\forall \underline{y} \in \mathbb{R}^n : \Phi(\underline{y}) = \frac{1}{2} \underline{y}^\top A \underline{y} - \underline{y}^\top \underline{b}$$

Poiché A è SDP $\Rightarrow \Phi$ ammette un unico minimo ($\bar{\underline{x}}$)

Tale minimo annulla il gradiente : $\nabla \Phi(\bar{\underline{x}}) = 0$

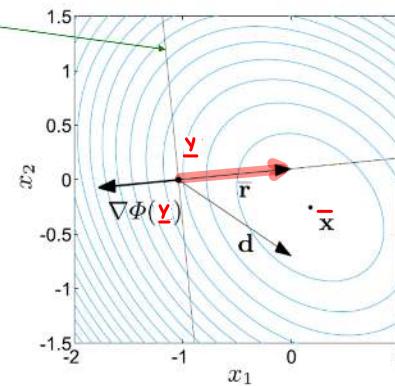
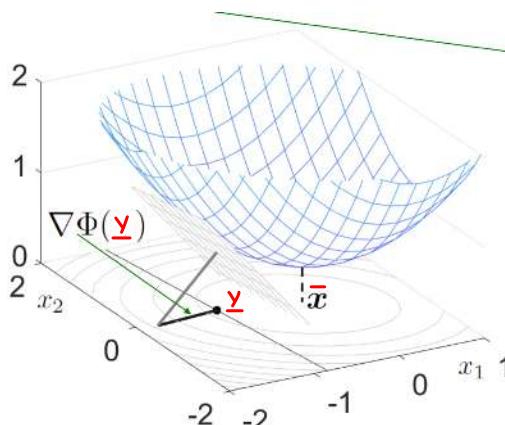
Ma : $\nabla \Phi(\underline{y}) = A \underline{y} - \underline{b} \rightarrow$ punto di minimo $\bar{\underline{x}}$ annulla $A \underline{y} - \underline{b}$



$$\Rightarrow \bar{\underline{x}} = \underline{x}$$

PUNTO DI MINIMO DI Φ E'
PROPRIO SOLUZIONE \underline{x} DI $A\underline{x} = \underline{b}$

Idea : Introduco un metodo iterativo che minimizzi l'energia $\Phi(\underline{y})$. Per fare ciò ricordo che la direzione di massima pendenza di Φ a partire da \underline{y} è $-\nabla \Phi(\underline{y})$.



Il $-\nabla \Phi(\underline{y})$ dà la direzione di massima discesa dell'energia

\Rightarrow Aggiorno $\underline{x}^{(k)}$ muovendomi lungo la direzione di max discesa: $\underline{x}^{(k+1)} = \underline{x}^{(k)} - \alpha_k \nabla \Phi(\underline{x}^{(k)})$

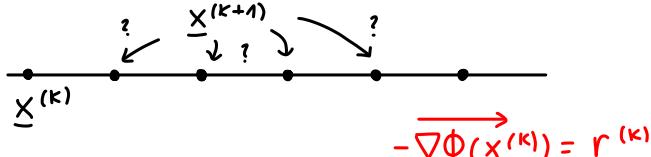
$$\text{MA } \nabla \Phi(\underline{x}^{(k)}) = A \underline{x}^{(k)} - \underline{b} = -\underline{r}^{(k)}$$

$$\Rightarrow \underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha_k \underline{r}^{(k)}$$

Metodo del gradiente
(RICHARDSON DINAMICO)

PARAMETRO DINAMICO

NOTA: α_{OPT} richiede il calcolo (spesso oneroso) degli autovalori λ di A . Con α_k mi svincolo da questo calcolo.



Ho scelto la direzione e ho fissato il mio punto di partenza.
Di quanto deve essere ampio il mio passo?

seleziono α_k che minimizza $\Phi(\underline{x}^{(k+1)}) = \underline{x}^{(k)} + \alpha_k \underline{r}^{(k)} = \Phi(\alpha_k) \Rightarrow$

$$\text{IMPONGO che } \frac{d\Phi(\underline{x}^{(k+1)})}{d\alpha_k} = 0$$

svolgendo i conti \Rightarrow

$$\alpha_k = \frac{(\underline{r}^{(k)})^T \underline{r}^{(k)}}{(\underline{r}^{(k)})^T A \underline{r}^{(k)}}$$

→ Mi svincolo dalla determinazione λ_{MAX} e λ_{MIN} (come per α_{OPT})

METODO GRADIENTE: Dato $\underline{x}^{(0)} \in \mathbb{R}^n$, calcolo $\underline{r}^{(0)} = \underline{b} - A \underline{x}^{(0)}$ e per ogni $k \geq 0$:

$$(i) \alpha_k = \frac{(\underline{r}^{(k)})^T \underline{r}^{(k)}}{(\underline{r}^{(k)})^T A \underline{r}^{(k)}} \sim n^2 \quad (n \text{ se } A \text{ sparsa})$$

$$(ii) \underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha_k \underline{r}^{(k)} \sim n$$

\Rightarrow costo singola

iterata $\sim n^2$

(n se A sparsa)

$$(iii) \underline{r}^{(k+1)} = \underline{b} - A \underline{x}^{(k+1)}$$

$$= \underline{b} - A(\underline{x}^{(k)} + \alpha_k \underline{r}^{(k)})$$

$$= \underline{r}^{(k)} - \alpha_k A \underline{r}^{(k)} = (I - \alpha_k A) \underline{r}^{(k)}$$

$\cancel{\sim n^2}$ meglio

$\sim n$ (perché $A \underline{r}^{(k)}$ già noto da (i))

Efficienza: Introduco la norma indotta da A :

$$\underline{z} \in \mathbb{R}^n : \|\underline{z}\|_A = \sqrt{\underline{z}^T A \underline{z}} ; \|\underline{z}\|_A = \sqrt{(\underline{z}, \underline{z})_A} ; (\underline{z}, \underline{w})_A = \underline{z}^T A \underline{w}$$

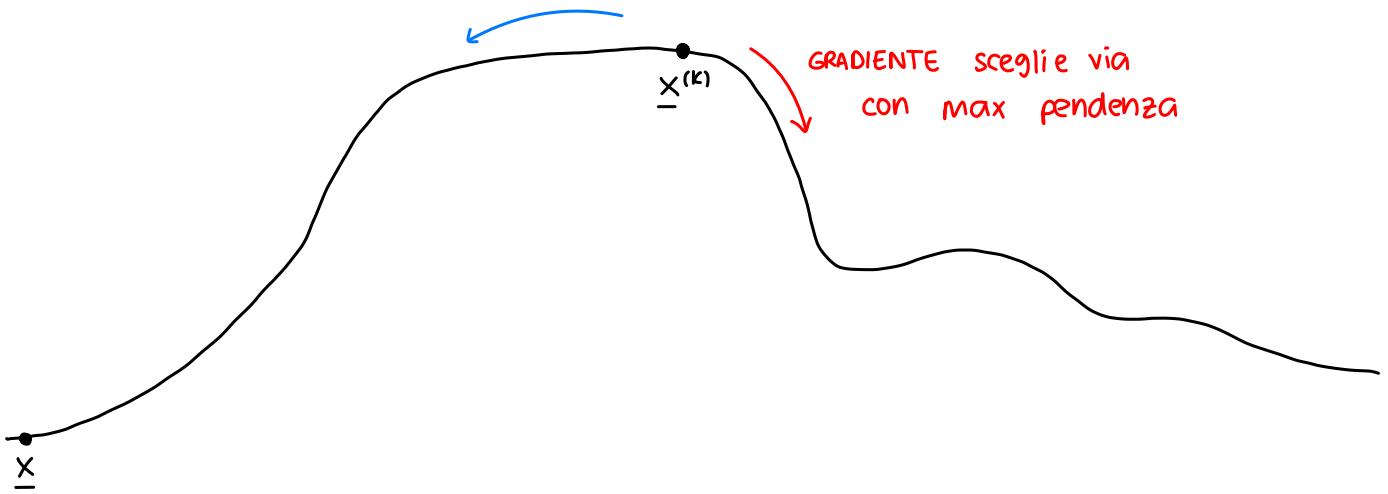
Si può dimostrare che per gradiente:

$$\|\underline{e}^{(k+1)}\|_A \leq \frac{K(A) - 1}{K(A) + 1} \|\underline{e}^{(k)}\|_A$$

L'ERRORE IN GRADIENTE SCALA
COME PER RICHARDSON STAZIONARIO
(RS e RD hanno stessa efficienza)

Per migliorare l'efficienza noto che $\nabla \Phi(\underline{x}^{(k)}) \perp \nabla \Phi(\underline{x}^{(k-1)})$ per costruzione geometrica

$$\Rightarrow \underline{r}^{(k)} \perp \underline{r}^{(k-1)} \quad \text{MA } \underline{r}^{(k)} \cancel{\perp} \underline{r}^{(j)} \quad j < k-1 \quad \text{cioè} \quad \begin{cases} (\underline{r}^{(k)}, \underline{r}^{(j)}) \neq 0 & \text{se } j < k-1 \\ (\underline{r}^{(k)}, \underline{r}^{(k-1)}) = 0 & \end{cases}$$



Cerco di muovermi lungo una direzione nuova che preservi l'**ortogonalità** rispetto a tutte le precedenti, in modo da massimizzare il carattere esplorativo del metodo. L'ortogonalità va intesa nel senso di $(\cdot, \cdot)_A \rightarrow$ [PROD. SCALARE in $A \neq$ EUCLIDEO]

→ Introduco le **direzioni coniugate** $\underline{d}^{(k)} : (\underline{d}^{(k)}, \underline{d}^{(j)})_A = 0 \quad j \leq k$
ortogonali a tutti i precedenti

METODO GRADIENTE CONIUGATO

Dato $\underline{x}^{(0)}$, calcolo $\underline{r}^{(0)} = \underline{b} - A\underline{x}$, pongo $\underline{d}^{(0)} = \underline{r}^{(0)}$ e per $k \geq 0$:

$$(i) \alpha_k = \frac{(\underline{d}^{(k)})^T \underline{r}^{(k)}}{(\underline{d}^{(k)})^T A \underline{d}^{(k)}} \rightarrow \text{parametro dinamico}$$

$$(ii) \underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha_k \underline{d}^{(k)} \rightarrow \text{Aggiornamento}$$

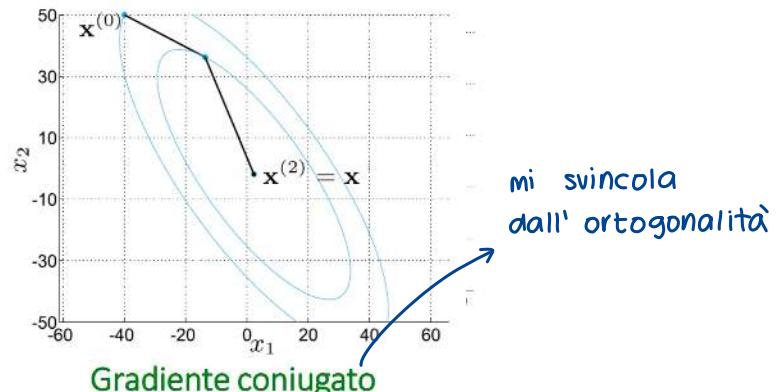
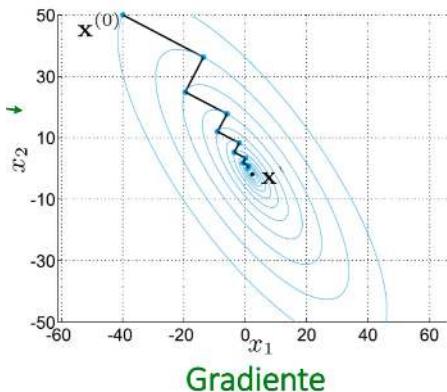
$$(iii) \underline{r}^{(k+1)} = \underline{b} - A \underline{x}^{(k+1)}$$

$$(iv) \beta_k = \frac{(A \underline{d}^{(k)})^T \underline{r}^{(k+1)}}{(A \underline{d}^{(k)})^T \underline{d}^{(k)}} \rightarrow \text{parametro per aggiornamento di } \underline{d}^{(k)}$$

$$(v) \underline{d}^{(k+1)} = \underline{r}^{(k+1)} - \underbrace{\beta_k \underline{d}^{(k)}}_{\text{CORREZIONE}} \rightarrow \text{Aggiornamento direzione coniugata}$$

RISPETTO A GRADIENTE

Rispetto a gradiente ho costo in più di (iv) $\sim n$



TEOREMA : Il metodo GC in aritmetica esatta (conti a mano) converge esattamente in n iterazioni.

In pratica, in aritmetica finita, CG non converge in n iterazioni, no errore numerico $\underline{e}^{(k)} = \underline{x} - \underline{x}^{(k)}$, però ho un metodo molto più veloce dei Richardson. Infatti : $\|\underline{e}^{(k)}\|_A \leq \frac{K(A)-1}{K(A)+1} \|\underline{e}^{(k-1)}\|_A$

$$\Rightarrow \|\underline{e}^{(k)}\|_A \leq \left(\frac{K(A)-1}{K(A)+1} \right)^k \|\underline{e}^{(0)}\|_A \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{GRADIENTE}$$

Si può mostrare : $\|\underline{e}^{(k)}\|_A \leq \frac{2C^k}{1+2C^{2k}} \|\underline{e}^{(0)}\|_A ; C = \frac{\sqrt{K(A)} - 1}{\sqrt{K(A)} + 1} \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{GRADIENTE CONIUGATO}$

$$\frac{\sqrt{K(A)} - 1}{\sqrt{K(A)} + 1} << \frac{K(A) - 1}{K(A) + 1} \quad \Rightarrow \quad \begin{array}{l} \text{GC ha una dipendenza} \\ \text{della convergenza da} \\ K(A) più tenue \rightarrow \text{maggiore efficienza} \end{array}$$

Il metodo del gradiente coniugato (rispetto al metodo del gradiente) risulta essere più efficiente in termini di numero di iterazioni necessarie per giungere a convergenza. Tuttavia, abbiamo comunque una dipendenza dal numero di condizionamento $K(A)$, si può quindi introdurre un precondizionamento.

GRADIENTE CONIUGATO PRECONDIZIONATO

Nonostante la migliore velocità di convergenza, il GC comunque soffre per $K(A) \gg 1$.
 → VSO precondizionatori

Spesso, data P SDP, si lavora su:

$$A \underline{x} = \underline{b} \rightarrow \underbrace{P^{-1} A P^{-T}}_I \underbrace{\underline{x}}_{\tilde{\underline{x}}} = P^{-1} \underline{b} \rightarrow \tilde{A} \tilde{\underline{x}} = \tilde{\underline{b}}$$

Ho certezza
che \tilde{A} sia SDP
(perché ho moltiplicato
a sx e dx per una
matrice SDP)

Applico GC a $\tilde{A} \tilde{\underline{x}} = \tilde{\underline{b}}$ → $\tilde{\underline{x}}$ → $\underline{x} = P^{-T} \tilde{\underline{x}}$

Si ottiene: $c = \frac{\sqrt{K(P^{-1}A)} - 1}{\sqrt{K(P^{-1}A)} + 1}$

DOPPIO EFFETTO DATO DAL GC-PRECONDIZIONATO:

- (i) minor numero di iterazioni : $\sqrt{K(A)} \leftrightarrow K(A)$ rispetto a GC
- (ii) rispetto a GC ho ulterioriamente diminuito # iterazioni

CRITERI D'ARRESTO

(1) CRITERIO SUL RESIDUO : $\|\underline{r}^{(k)}\| < \varepsilon \rightarrow$ tolleranza assegnata

$$\frac{\|\underline{r}^{(k)}\|}{\|\underline{b}\|} < \varepsilon \quad o \quad \frac{\|\underline{r}^{(k)}\|}{\|\underline{r}^{(0)}\|} < \varepsilon \quad \begin{array}{l} \text{versioni} \\ \text{normalizzate} \end{array}$$

Da discussione sul MEG sappiamo che: $\frac{\|\underline{x} - \hat{\underline{x}}\|}{\|\underline{x}\|} < K(A) \frac{\|\underline{b} - A \hat{\underline{x}}\|}{\|\underline{b}\|}$

Adattando con $\hat{\underline{x}} = \underline{x}^{(k)}$ → $\frac{\|\underline{e}^{(k)}\|}{\|\underline{x}\|} \leq K(A) \frac{\|\underline{r}^{(k)}\|}{\|\underline{b}\|}$

→ Residuo buon criterio d'arresto se $K(A)$ piccolo
 → OK per metodi precondizionati

(2) CRITERIO SU INCREMENTO : $\|\underline{\delta}^{(k+1)}\| < \varepsilon$

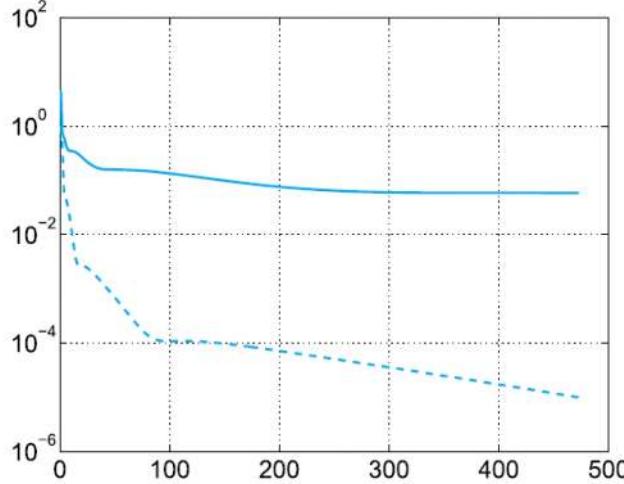
$$\underline{\delta}^{(k+1)} = \underline{x}^{(k+1)} - \underline{x}^{(k)}, \quad \frac{\|\underline{\delta}^{(k+1)}\|}{\|\underline{x}^{(k+1)}\|} < \varepsilon \quad \begin{array}{l} \text{versione} \\ \text{normalizzata} \end{array}$$

NOTA: non è l'errore!

Vale che: $\|\underline{e}^{(k)}\| = \|\underline{x} - \underline{x}^{(k)}\| = \|\underline{x} - \underline{x}^{(k+1)} + \underline{x}^{(k+1)} - \underline{x}^{(k)}\| \leq \|\underline{e}^{(k+1)}\| + \|\underline{\delta}^{(k+1)}\| \leq \rho(B) \|\underline{e}^{(k)}\| + \|\underline{\delta}^{(k+1)}\|$
 $\Rightarrow \|\underline{e}^{(k)}\| \leq \frac{1}{1-\rho(B)} \|\underline{\delta}^{(k+1)}\|$

→ buon criterio se $\rho(B) \ll 1$

Esempi di criterio sul residuo



Andamento (al variare di k) del residuo relativo (in linea tratteggiata) e dell'errore $\|\mathbf{x} - \mathbf{x}^{(k)}\|/\|\mathbf{x}\|$ (in linea continua) per il metodo di Gauss-Seidel applicato al sistema di Hilbert →

che sappiamo essere mal condizionata

Consideriamo un sistema con matrice $A \in \mathbb{R}^{50 \times 50}$ tridiagonale simmetrica avente elementi sulla diagonale principale pari a 2.001 e quelli sulla sopra e sottodiagonale pari a 1. Al solito, il termine noto del sistema verrà scelto in modo che il vettore $(1, \dots, 1)^T$ sia la soluzione esatta. Essendo A tridiagonale a dominanza diagonale stretta, il metodo di Gauss-Seidel convergerà più rapidamente di quello di Jacobi.

Partendo da un vettore iniziale di componenti $(\mathbf{x}_0)_i = 10 \sin(100i)$ (per $i = 1, \dots, n$) e richiedendo una tolleranza $\text{tol} = 10^{-5}$, il programma restituisce dopo ben 859 iterazioni una soluzione affetta da un errore $\|\mathbf{e}^{(859)}\| \simeq 0.0021$. La convergenza è molto lenta e l'errore è piuttosto grande poiché il raggio spettrale della matrice di iterazione è pari a 0.9952, cioè molto vicino a 1. Se gli elementi diagonali fossero stati pari a 3, avremmo invece ottenuto convergenza in 17 iterazioni con un errore $\|\mathbf{e}^{(17)}\| \simeq 8.96 \cdot 10^{-6}$: in questo caso infatti il raggio spettrale della matrice di iterazione è pari a 0.4428 ■

2C. METODI NUMERICI PER SISTEMI NON LINEARI

Spesso i problemi ingegneristici sono **NON-LINEARI** → Es: onda che si proga non linearmente, flusso turbolento nei vasi sanguigni, elettrofisiologia (su slide), ...

Data $\underline{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$, cerco radici $\underline{x} \in \mathbb{R}^n$ del sistema di equazioni non lineari $\underline{f}(\underline{x}) = \underline{0}$, equivalentemente:

$$\begin{cases} f_1(x_1, \dots, x_n) = 0 \\ \vdots \\ f_j(x_1, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, \dots, x_n) = 0 \end{cases} \quad \begin{matrix} n \text{ equazioni non-lineari} \\ \text{in } \underline{x} = (x_1, \dots, x_n)^T \end{matrix}$$

IDEA: estendere metodo di Newton al caso di sistemi.

RIPASSO CASO scalare: per trovare radici $x \in \mathbb{R}$ di $f(x)$, dato $x^{(0)} \in \mathbb{R}$

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})} \quad \text{se } f'(x^{(k)}) \neq 0$$

Equivalentemente, lo riscrivo:

$$\text{Dato } x^{(0)}, \quad \begin{cases} \delta x^{(k)} = x^{(k+1)} - x^{(k)} \\ f'(x^{(k)}) \delta x^{(k)} = -f(x^{(k)}) \\ x^{(k+1)} = x^{(k)} + \delta x^{(k)} \end{cases} \quad (*) \quad \begin{matrix} \text{NUOVA VARIABILE} \\ \text{AUXILIARIA} \end{matrix}$$

Voglio estendere (*) al caso vettoriale. A tal fine, introduco per ogni $\underline{y} \in \mathbb{R}^n$ la **MATRICE JACOBIANA**:

$$J(\underline{y}) = \nabla f(\underline{y}) \in \mathbb{R}^{n \times m}$$

$$\text{cioè } J_{il}(\underline{y}) = \frac{\partial f_i}{\partial x_l}(\underline{y}) \quad , \quad i, l = 1, \dots, n$$

Esempio:

$$f(\underline{x}) = \begin{cases} x_1^2 + 2x_2 x_3 + \sin(x_3) \\ x_2^3 + x_3 \\ 1/x_1 + x_2^2 + x_1 x_3 \end{cases} \quad n = 3$$

$$J(\underline{x}) = \begin{bmatrix} 2x_1 & 2x_3 & 2x_2 + \cos(x_3) \\ 0 & 3x_2^2 & 1 \\ -1/x_1^2 & 2x_2 & x_1 \end{bmatrix}$$

Ora valuto J in $\underline{y} = (4 \ -7 \ 1)^T$

$$J(\underline{y}) = \begin{bmatrix} 8 & 2 & -14 + \cos(1) \\ 0 & 49 \cdot 3 & 1 \\ -1/16 & -14 & 4 \end{bmatrix}$$

Introduco metodo di Newton per sistemi

**NEWTON
SCALARE**

$$\begin{array}{l} \text{Dato } \underline{x}^{(0)} \in \mathbb{R} \\ \text{se } f'(\underline{x}^{(k)}) \neq 0 \end{array} \Rightarrow \begin{cases} f'(\underline{x}^{(k)}) \delta \underline{x}^{(k)} = -f(\underline{x}^{(k)}) \\ \underline{x}^{(k+1)} = \underline{x}^{(k)} + \delta \underline{x}^{(k)} \end{cases}$$

**NEWTON
VETTORIALE**

$$\begin{array}{l} \text{Dato } \underline{x}^{(0)} \in \mathbb{R} \\ \text{SE } J(\underline{x}^{(k)}) \text{ e'} \text{ INVERTIBILE} \end{array} \Rightarrow \begin{cases} J(\underline{x}^{(k)}) \delta \underline{x}^{(k)} = -f(\underline{x}^{(k)}) \\ \underline{x}^{(k+1)} = \underline{x}^{(k)} + \delta \underline{x}^{(k)} \end{cases}$$

Ogni iterazione
richiede di
risolvere 1
sistema lineare

Quindi, ad ogni iterazione :

- (i) Risolvo un sistema lineare con matrice $J(\underline{x}^{(k)}) \rightarrow$ matrice nota ($\underline{x}^{(k)}$ noto)
uso una delle tecniche viste (MEG, GC, PRECONDIZ, ...)
- (ii) Sistema lineare mi permette di avere $\delta \underline{x}^{(k)}$
 \rightarrow Aggiornamento $\underline{x}^{(k+1)} = \underline{x}^{(k)} + \delta \underline{x}^{(k)}$
- (iii) $k \rightarrow k+1$ fino a soddisfacimento di un criterio d'arresto.

PROPRIETA'

- E' un metodo locale : convergenza $\lim_{k \rightarrow +\infty} \|\underline{x}^{(k)} - \underline{x}\| = 0$

solo per $\underline{x}^{(0)}$ sufficientemente vicino a \underline{x} :

$$\exists \delta > 0 : \|\underline{x}^{(k)} - \underline{x}\| < \delta$$

NOTA: ora sono norme e non val. ass., perché sono vettori !

- E' un metodo del 2^o ordine :

se converge $\rightarrow \|\underline{x}^{(k+1)} - \underline{x}\| \leq C \|\underline{x}^{(k)} - \underline{x}\|^2, C > 0$

nuovo errore scala come costante per vecchio errore al quadrato
 \rightarrow metodo molto veloce

Esempio :

$$f(\underline{x}) = 0 : \begin{cases} x_1^2 - 2x_1x_2 - 2 = 0 & f_1 = 0 \\ x_1 + x_2^2 + 1 = 0 & f_2 = 0 \end{cases}$$

$$J(\underline{x}) = \begin{pmatrix} 2(x_1 - x_2) & -2x_1 \\ 1 & 2x_2 \end{pmatrix}; \rightarrow \begin{cases} J(\underline{x}^{(k)}) \delta \underline{x}^{(k)} = -f(\underline{x}^{(k)}) \\ \underline{x}^{(k+1)} = \underline{x}^{(k)} + \delta \underline{x}^{(k)} \end{cases}$$

$$\begin{pmatrix} 0 & -2 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} \delta x_1^{(0)} \\ \delta x_2^{(0)} \end{pmatrix} = -\begin{pmatrix} -3 \\ 3 \end{pmatrix} \rightarrow \begin{cases} -2 \delta x_2^{(0)} = 3 \\ \delta x_1^{(0)} + 2 \delta x_2^{(0)} = -3 \end{cases}$$

$$\begin{cases} \delta x_2^{(0)} = -3/2 \\ \delta x_1^{(0)} = 0 \end{cases} \rightarrow \underline{x}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ -3/2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1/2 \end{pmatrix}$$

$$\bullet \quad k=1 \quad J(\underline{x}^{(1)}) = \dots, \quad f(\underline{x}^{(1)}) = \dots \rightarrow \underline{x}^{(2)} = \dots$$

CRITERI DI ARRESTO

(i) **RESIDUO**: $\| \underline{f}(\underline{x}^{(k)}) \| < \varepsilon ; \frac{\| \underline{f}(\underline{x}^{(k)}) \|}{\| \underline{f}(\underline{x}^{(0)}) \|} < \varepsilon$

(ii) **INCREMENTO**: $\| \underline{x}^{(k+1)} - \underline{x}^{(k)} \| < \varepsilon ; \frac{\| \underline{x}^{(k+1)} - \underline{x}^{(k)} \|}{\| \underline{x}^{(k+1)} \|} < \varepsilon$

COSTI COMPUTAZIONALI

Ad ogni iterazione k devo:

- (i) Costruire $J(\underline{x}^{(k)})$
 - (ii) Risolvere sistema lineare
- $\left. \begin{array}{l} \\ \end{array} \right\}$ Mi aspetto costo elevato

Entrambi i passi hanno un costo computazionale

costo complessivo : $CPU\ Time = \#ITER (C_{cos} + C_{SL})$

$\downarrow \quad \downarrow$

Numero iterazioni di Newton Costo per costruire J

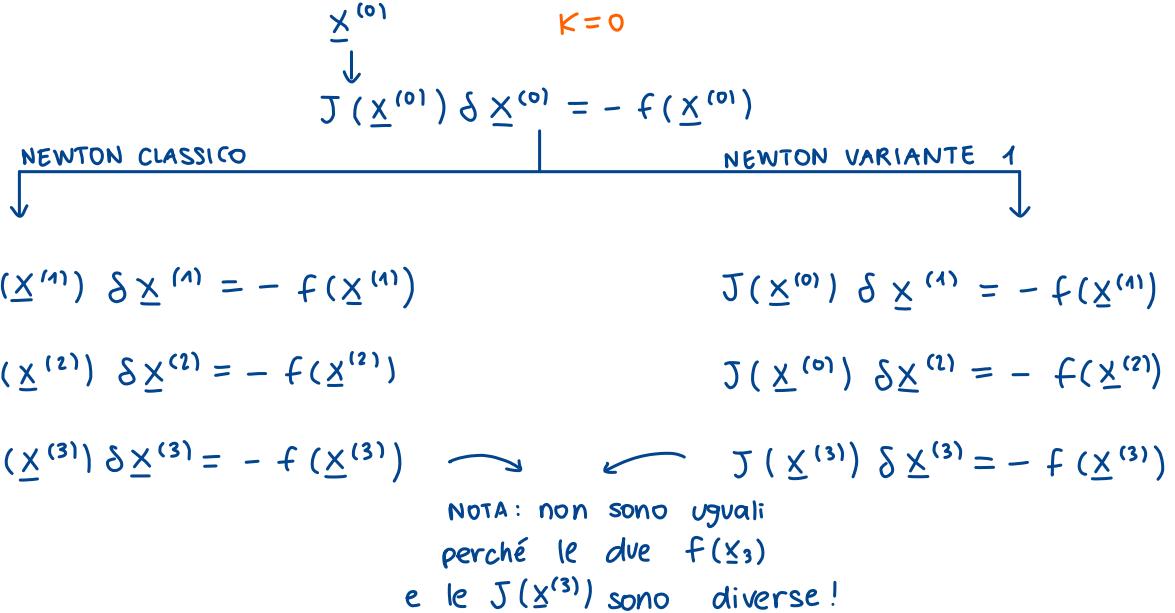
Costo per risolvere il sistema lineare

Per velocizzare Newton ho 2 tecniche:

- (1) **Aggiornamento di J solo ogni $p \geq 2$ iterazioni**

Fissato $p \geq 2$, Aggiorno $J(\underline{x}^{(k)})$ solo ogni p iterazioni e la lascio "congelata" nelle altre.

Es: $p = 3$



C_{cos} è spalmato su p iterazioni, però perdo ordine 2
 $\rightarrow \#ITER_{VARIANTE\ 1} > \#ITER_{NEWTON}$

Attenzione: variante 1 converge (ma più lentamente)

Speranza è che $\#ITER_{VARIANTE\ 1} \cdot \left(\frac{C_{cos}}{p} + C_{SL} \right) < \#ITER_{NEWTON} (C_{cos} + C_{SL})$

Se inoltre usiamo FATT LU per J → spalmo su p iter. anche C_{SL} → Spero che

$$\# \text{ITERVARIANTE1} \left(\frac{C_{\cos} + C_{\text{SL}}}{p} \right) < \# \text{ITER}_{\text{NEWTON}} (C_{\cos} + C_{\text{SL}})$$

P : compromesso (~ 5 o 10) \leadsto se troppo grande devo fare molte iterazioni

(2) Approssimare J ad ogni k

$J(\underline{x}^{(k)}) \rightarrow J^{(k)}$ "Facile" da risolvere

\rightarrow Abbatto C_{SL}

Attenzione $\# \text{ITERVARIANTE2} > \# \text{ITER}_{\text{NEWTON}}$

3. APPROXIMAZIONE DI DATI E FUNZIONI

Conosco $n+1$ coppie di dati (x_i, y_i) , $i = 0, \dots, n$.

I dati sono di due tipi :

- MISURE Sperimentali
- VALORI DI UNA FUNZIONE $f(x)$. In questo caso $y_i = f(x_i)$

PROBLEMA : Cerco una funzione approssimante $\tilde{f}(x)$ che bene approssima i dati.

Cio' mi permette di avere informazioni anche "lontano" dalle misure.

- Dati dei dati, esistono più approssimanti possibili
- Se cambio i dati, l'approssimante cambia

Interpolazione lagrangiana

DEF: Una funzione \tilde{f} approssimante i dati $(x_i, y_i), i=0, \dots, n$, è detta **interpolante** se:

$$\tilde{f}(x_i) = y_i \quad i=0, \dots, n \rightarrow \text{se la } \tilde{f} \text{ passa per i punti}$$

Introduciamo i polinomi caratteristici di Lagrange, associati ai dati $(x_i, y_i), i=0, \dots, n$.

$$\varphi_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}, \quad i=0, \dots, n \rightarrow \text{sono } n+1 \text{ polinomi di grado } n$$

Valutiamo le φ_i nei nodi x_j :

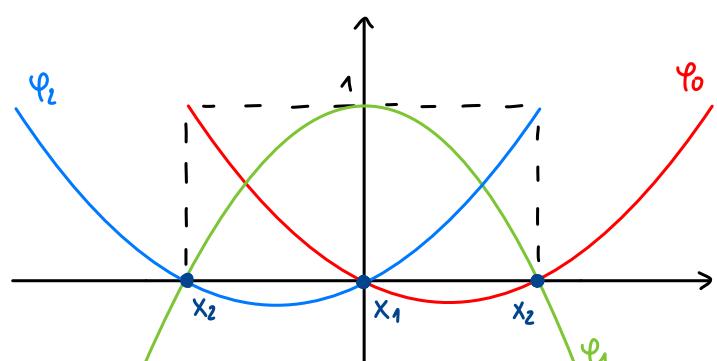
$$\varphi_i(x_j) = \begin{cases} 0 & j \neq i \\ 1 & j = i \end{cases} \quad \text{ciascuna } \varphi_i \text{ vale 1 se valutata nel nodo } i\text{-esimo e 0 altrove}$$

Si indica con il delta di Kronecker: $\varphi_i(x_j) = \delta_{ij}$

Esempio $n=2$; $x_0 = -1$; $x_1 = 0$; $x_2 = 1$; y_0, y_1, y_2, \dots

avrò $n+1 = 3$ polinomi di grado $n=2$.

- $\varphi_0(x) = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2} = \frac{x(x-1)}{2} \quad i=0$
- $\varphi_1(x) = \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2} = (x+1)(1-x) \quad i=1$
- $\varphi_2(x) = \frac{x - x_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1} = \frac{(x+1)x}{2} \quad i=2$



Costruiamo l'interpolatore lagrangiano:

Dati (x_i, y_i) , $i=0, \dots, n$ esso è dato:

$$\Pi_n(x) = \sum_{j=0}^n y_j \varphi_j(x)$$

↓ MISURE ↓ POLINOMI CARATTERISTICI
DI LAGRANGE

PROPRIETA'

(i) Π_n è un interpolatore. Infatti : $\Pi_n(x_i) = \sum_{j=0}^n y_j \varphi_j(x_i) = \sum_{j=0}^n y_j \delta_{ij} = y_i$ ok

(ii) $T\ln(x)$ è un polinomio di grado n , perché somma di polinomi di grado n (le ψ_j).

(iii) T_{1n} è unico polinomio di grado n che interpola $n+1$ dati. Infatti, supponiamo che ce ne sia un altro $\Psi_n(x)$. Allora, $D_n(x) = T_{1n}(x) - \Psi_n(x)$ è un polinomio di grado n . In più:

$$D_n(x_i) = \bar{T}l_n(x_i) - \Psi_n(x_i) = y_i - y_i = 0$$

$\rightarrow D_n$ è polinomio di grado n con $n+1$ zeri $\rightarrow D_n = 0$

$$\underline{\text{Esempio}} \quad \left\{ \begin{array}{l} x_0 = -1 \\ y_0 = 5 \end{array} \right. , \quad \left\{ \begin{array}{l} x_1 = 0 \\ y_1 = 2 \end{array} \right. , \quad \left\{ \begin{array}{l} x_2 = 1 \\ y_2 = 4 \end{array} \right.$$

$$\Pi_2(x) = \sum_{j=0}^2 y_j \varphi_j(x) = y_0 \varphi_0(x) + y_1 \varphi_1(x) + y_2 \varphi_2(x) = 5 \frac{x(x-1)}{2} + 2(x+1)(1-x) + 4 \frac{x(x+1)}{2} =$$

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

Qual è l'accuratezza di $T_{ln}(x)$ "lontano dai nodi"? (\rightarrow cioè in mezzo ai nodi)

Considero il caso $y_i = f(x_i)$, $i = 0, \dots, n$

In questo caso l'interpolatore di Lagrange è $T_1 f(x)$. Qual è l'errore $f(x) - T_1 f(x)$?

Nei nodi so che l'errore è nullo, ma lontano?

$H_0: f(x) \rightarrow$ considero $n+1$ dati $(x_i, f(x_i)) \rightarrow$ costruisco $\prod f(x)$ \rightarrow valuto errore $f - \prod f$

TEOREMA Se $f(x)$ è derivabile fino all'ordine $n+1$, allora $\forall x \exists \xi_x :$

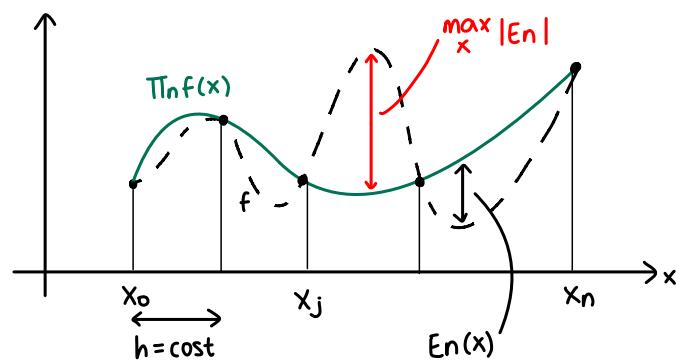
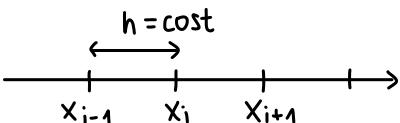
$$E_n(x) = f(x) - T_n f(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^n (x-x_i)$$

NOTA : Nei nodi $E_n(x_j) = 0$, ma $E_n(x) \neq 0$ per $x \neq x_j$ lontano dai nodi.

Il precedente è un risultato puntuale, che però non è quantificabile perché non conosco Σx . Per ottenere una stima dell'errore utilizzabile, passo ad un risultato globale:

$$\max_x |E_n(x)| \leq \max_x \left| \frac{f^{(n+1)}(x)}{4(n+1)} \right| h^{n+1}$$

→ STIMA DELL'ERRORE MASSIMO
NEL CASO EQUISPAZIATO



CONVERGENZA DI $T_{n,f}(x)$

Vorremo che aumentando il numero delle informazioni ($n \uparrow$), l'approssimazione migliori.
Vorrei che $\lim_{n \rightarrow +\infty} E_n(x) = 0$. Verifico ciò.

Tornando alla stima: $\max_x |E_n(x)| \leq \boxed{\max_x f^{(n+1)}(x)} \cdot \frac{h^{n+1}}{4(n+1)} \xrightarrow{n \rightarrow +\infty} 0$

Ho due scenari :

$$(1) \lim_{n \rightarrow +\infty} \max_x |f^{(n+1)}(x)| < +\infty \Rightarrow \max_x |E_n| \xrightarrow{n \rightarrow +\infty} 0 \quad \text{OK convergenza}$$

$$(2) \lim_{n \rightarrow +\infty} \max_x |f^{(n+1)}(x)| = +\infty \Rightarrow \max_x |E_n| \xrightarrow{n \rightarrow +\infty} ???$$

Sfortunatamente esistono funzioni nello scenario 2, tali per cui $\max_x |E_n(x)| \xrightarrow{n \rightarrow +\infty} +\infty$.

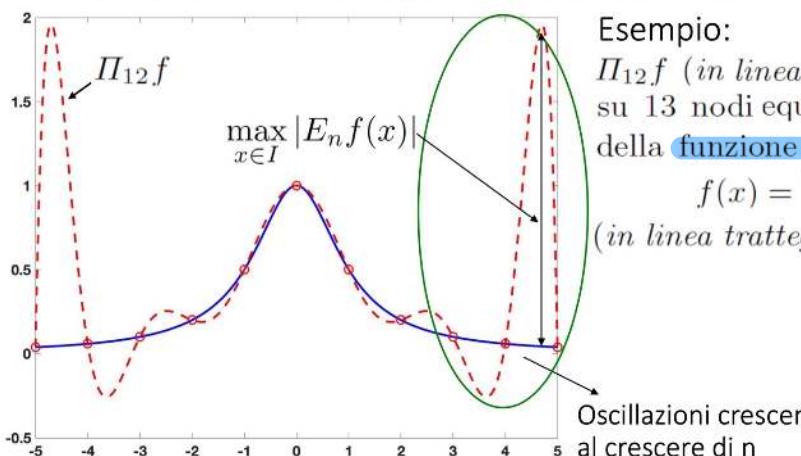
FENOMENO DI RUNGE : possibile non convergenza di $T_{n,f}(x)$ nel caso equispaziato ($h = \text{cost}$)

In particolare, posso avere oscillazioni alle estremità del dominio che si accentuano sempre di più all'infittirsi dei nodi :

Esempio $f(x) = \frac{1}{1+x^2} \quad x \in [-5, 5]$

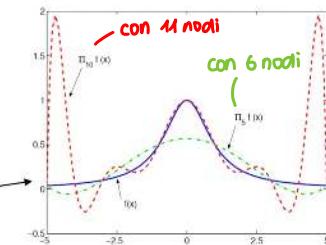
La possibile mancata convergenza dell'interpolatore Lagrangiano prende il nome di fenomeno di Runge

In particolare, in presenza di questo fenomeno, la funzione errore E_n presenta delle oscillazioni ai nodi estremi che crescono con il crescere di n



Esempio:
 $P_{12}f$ (in linea continua) calcolato su 13 nodi equispaziati nel caso della **funzione di Runge**
 $f(x) = 1/(1+x^2)$ (in linea tratteggiata)

Aumentando in numero di informazioni aumenta il grado del polinomio, il quale oscilla di più perché deve avere più zeri. Bisogna trovare un compromesso.



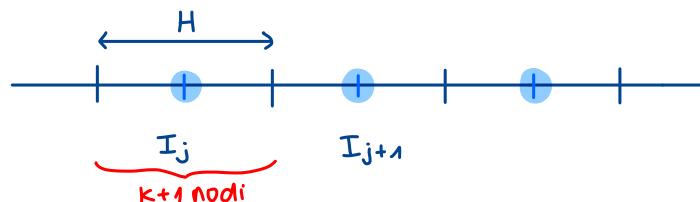
Interpolazione lagrangiana composita

Considero, all'interno del dominio di interesse, dei sottodomini. All'interno di ognuno di essi costruisco l'interpolatore lagrangiano. In particolare, avrò 2 requisiti:

- (i) globalmente l'interpolatore sia continuo
- (ii) il grado degli interpolatori lagrangiani deve essere basso ($k=1,2,3$)

 evito Runge

Esempio $k=2$



Raggruppo i nodi a $k+1$ a $k+1$ → ogni sottodominio I_j contiene $k+1$ nodi (nell'esempio 3).

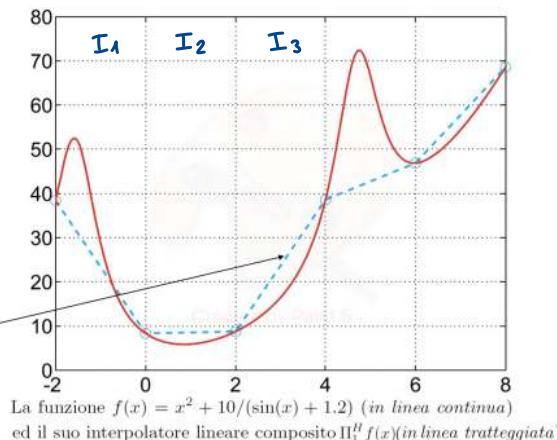
chiamo l'unione di tutti gli interpolatori lagrangiani di ordine k come **INTERPOLATORE LAGRANGIANO COMPOSITO** $\Pi_k^H(x)$ o $\Pi_k^H f(x)$. Esso è **CONTINUO** e **NON DERIVABILE**

Esempio : Interpolatore Lagrangiano composito lineare

Consideriamo nel seguente esempio il caso $k=1$

Vista la sua semplicità questo interpolatore è molto utilizzato nelle applicazioni

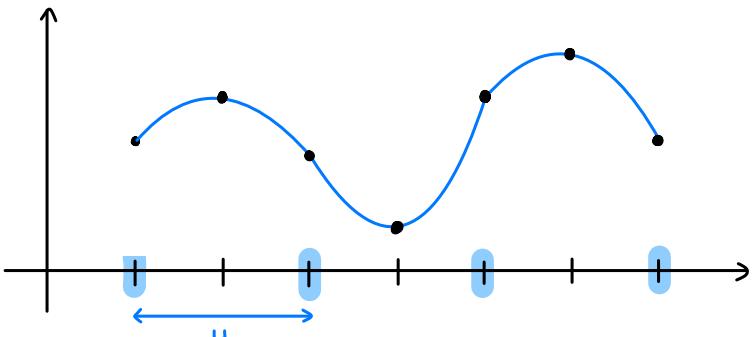
Interpolatore locale lineare

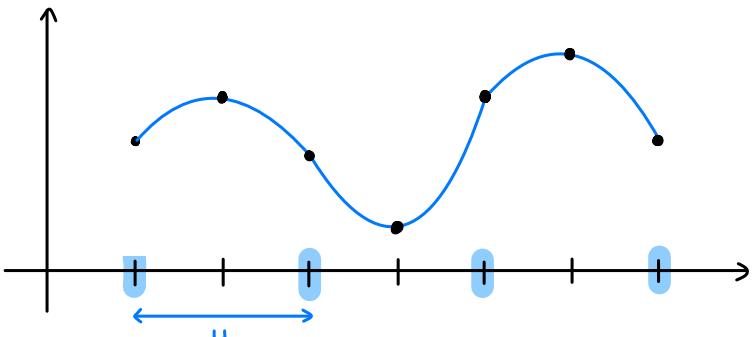


La funzione $f(x) = x^2 + 10/(\sin(x)) + 1.2$ (in linea continua) ed il suo interpolatore lineare composito $\Pi_1^H f(x)$ (in linea tratteggiata)

Esempio : ($k=2$) → Raggruppo i nodi a 3 a 3

$\Pi_2^H(x)$

 = Parabole ($k=2$)



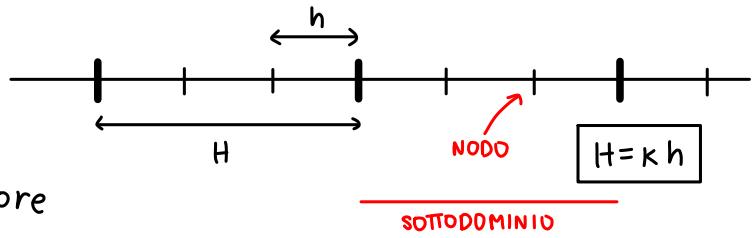
NOTA : Se n cresce (ho a disposizione più dati) :

- Per $\Pi_n(x)$ aumenta $n \rightarrow$ grado del polinomio
- Per $\Pi_k^H(x)$ aumenta numero di sottodomini mentre k rimane costante.

Mi aspetto di evitare il fenomeno di Runge perché lavoro sempre con k basso.

Definisco errore:

$$E_k^H f(x) = f(x) - T_k^H f(x)$$



So che su ogni I_j vale la stima dell'errore dell'interpolatore lagrangiano:

$$\text{su } I_j : E_k^H f(x) = \max_{x \in I_j} \frac{|f^{(k+1)}(x)|}{4(k+1)} h^{k+1}$$

→ L'errore globale su tutto l'intervalle sarà il massimo di tutti i precedenti errori sugli I_j .

$$\rightarrow |E_k^H f(x)| \leq \max_j \left(\max_{x \in I_j} \frac{|f^{(k+1)}(x)|}{4(k+1)} h^{k+1} \right) = \frac{\max_x |f^{(k+1)}(x)|}{4(k+1) H^{k+1}} \cdot H^{k+1}$$

$H = h/k$

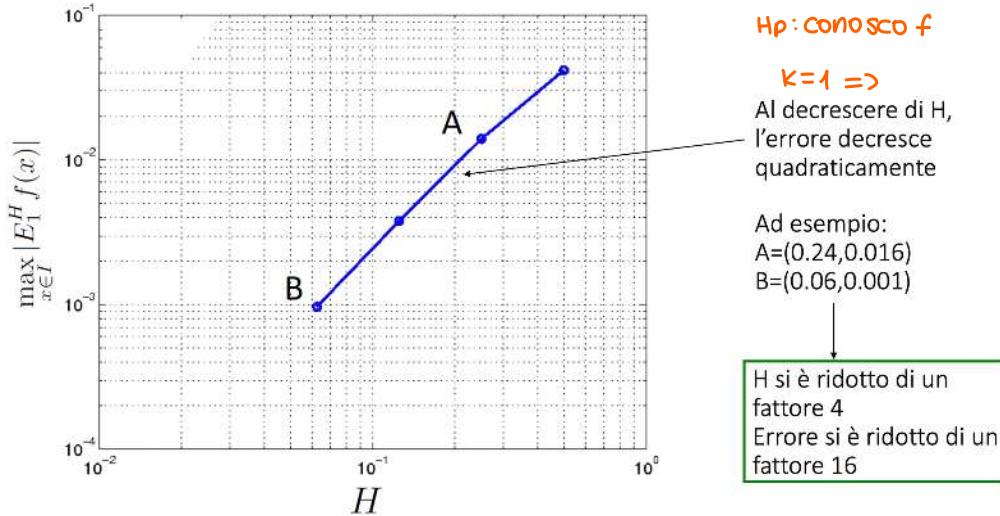
per $n \rightarrow +\infty$
perché per $n \rightarrow +\infty$
si ha $H \rightarrow 0$

È SEMPRE LIMITATO
(perché $f \in C^{k+1}$)

→ E` un metodo di ordine $k+1$: $|E_k^H f(x)| = \mathcal{O}(H^{k+1})$

Esempio

Ad esempio, considerando l'interpolatore Lagrangiano composito di ordine 1, la convergenza rispetto ad H per la funzione di Runge si comporta come segue

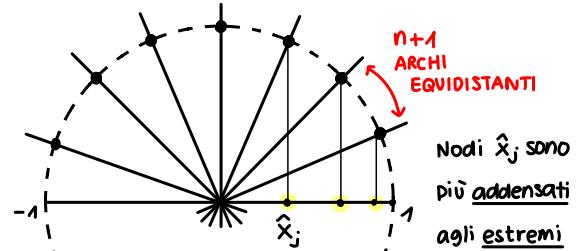


Interpolazione su nodi di Chebichev

IDEA : Poiché oscillazioni di $T_n(x)$ quando avviene il fenomeno di Runge si manifestano alle estremità, infatti i nodi alle estremità

Dettagliamo il caso Chebichev : $x \in [-1; 1]$

Prendo come nodi : $\hat{x}_j = -\cos\left(\frac{\pi j}{n}\right)$, $j=0, \dots, n$

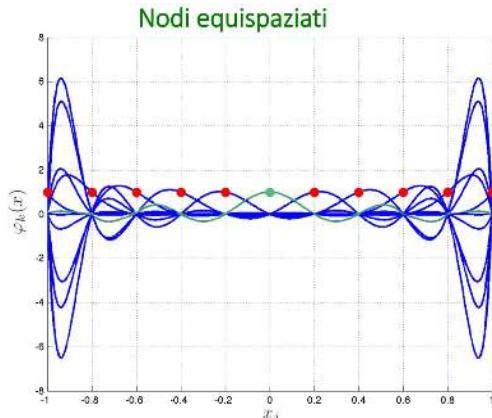
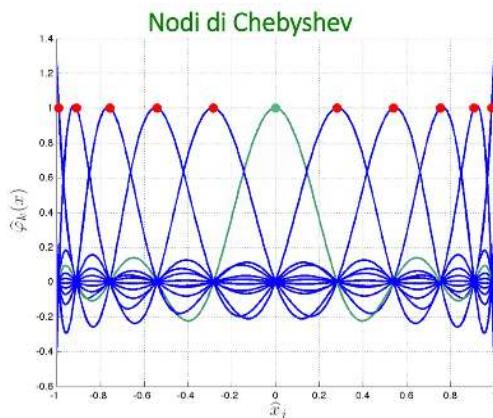


Costruisco l'interpolazione lagrangiana su \hat{x}_j :

- Polinomi caratteristici : $\hat{\varphi}_k(x) = \prod_{j \neq k} \frac{x - \hat{x}_j}{\hat{x}_k - \hat{x}_j}$ ← NON VARIA
- costruisco interpolatore : $\Pi_n^c(x) = \sum_{j=0}^n y_j \hat{\varphi}_j(x)$

Confronto polinomi caratteristici

Consideriamo 11 nodi ($n=10$) e i corrispondenti polinomi caratteristici



Si notano le assenze di oscillazioni nel caso Chebychev in prossimità degli estremi, dovute ad un infittimento dei nodi in quelle regioni

CONVERGENZA : se $f \in C^{s+1}([-1; 1])$ →
 $\max_{x \in [-1, 1]} |f(x) - \Pi_n^c f(x)| \leq \tilde{C} \left[\frac{1}{n^s} \right] \xrightarrow{n \rightarrow \infty} 0$ sempre

Interpolatore stabile
e al crescere di n
smorzo le oscillazioni

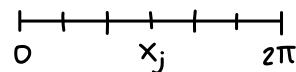
NOTA : se $f \in C^{s+1}$ $\forall s \rightarrow \max_{x \in [-1, 1]} |f - \Pi_n^c f| \leq \tilde{C} \cdot e^{-n}$
INFINITE DER. CONTINUE
convergenza esponenziale → rapidissima

LIMITE DEL METODO : non sempre posso ubicare i nodi con il cos → dati nel tempo

ESTENSIONE AL DOMINIO $[a, b]$

Interpolazione trigonometrica

Supponiamo che voglia approssimare dati o funzioni periodici, ad esempio di periodo 2π su $[0, 2\pi]$. Nodi: $x_j = \frac{2\pi j}{n+1}$, $j=0, \dots, n$.



es: tempo

IDEA: costruisco interpolatore $\tilde{f}(x)$ non polinomiale, bensì trigonometrico:

$$\tilde{f}(x) = \frac{a_0}{2} + \sum_{k=1}^M [a_k \cos(kx) + b_k \sin(kx)] \quad M = \frac{n}{2}, \quad n \text{ PARI}$$

Ciò è equivalente, grazie alla formula di Euler $e^{ikx} = \cos(kx) + i \sin(kx)$

$$\tilde{f}(x) = \sum_{k=-M}^M c_k e^{ikx} \quad \text{con } c_k \text{ incogniti}$$

funzioni note \rightarrow equivalenti ai polinomi lagrangiani

Ho $2M+1 = n+1$ incognite. Ottengo $n+1$ equazioni imponendo il vincolo di interpolazione:

$$\tilde{f}(x_j) = \sum_{k=-M}^M c_k e^{ikx_j} = f(x_j) \quad \text{con } j=0, \dots, n$$

NOTA: $h = \frac{2\pi}{n+1} \rightarrow x_j = jh \rightarrow \sum_{k=-M}^M c_k e^{ikjh} = f(x_j), \forall j$

Il precedente sistema lineare mi permette di passare dal dominio spaziale (delle misure $f(x_j)$) al dominio delle frequenze (ai coefficienti c_k).

Tale sistema ha per soluzione:

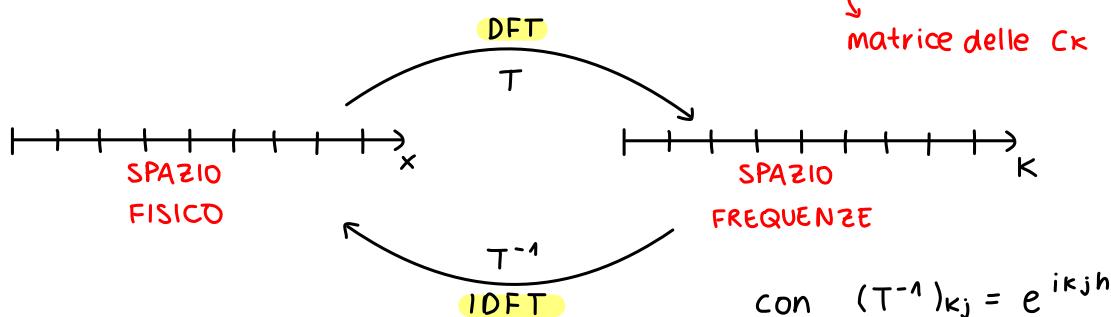
$$c_k = \frac{1}{n+1} \sum_{j=0}^n f(x_j) e^{-ikjh} \quad \text{TRASFORMATURA DISCRETA DI FOURIER (DFT)}$$

DFT è una mappa dal mondo spaziale al mondo delle frequenze. Invece:

$$f(x_j) = \sum_{k=-M}^M c_k e^{ikjh} \quad \text{INVERSE DFT (IDFT)}$$

La IDFT è una mappa dal dominio delle frequenze al dominio spaziale.

Formalmente: $T_{kj} = \frac{1}{n+1} e^{-ikjh}$, $T \in \mathbb{R}^{(n+1) \times (n+1)} \Rightarrow \mathbb{R}^{n+1} \ni \underline{f} = T \underline{c}$, con $(\underline{f})_j = f(x_j)$



COSTO: Per applicare la DFT devo eseguire un prodotto matrice - vettore
 \rightarrow COSTO $\sim (n+1)^2$ PROIBITIVO

Negli anni '60 è stata introdotta la celebre Fast Fourier Transform (FFT)
 \rightarrow COSTO $\sim (n+1) \log(n+1)$

ACCURATEZZA: Esponenziale : $\max_{x \in [0, 2\pi]} |f(x) - \tilde{f}(x)| \leq C \frac{1}{n^3}$

purché $f \in C^{s+1}([0, 2\pi])$
 \rightarrow CONVERGENZA ESPONENZIALE
 se s è grande

Il fenomeno dell'Aliasing

13 - 10/05/23

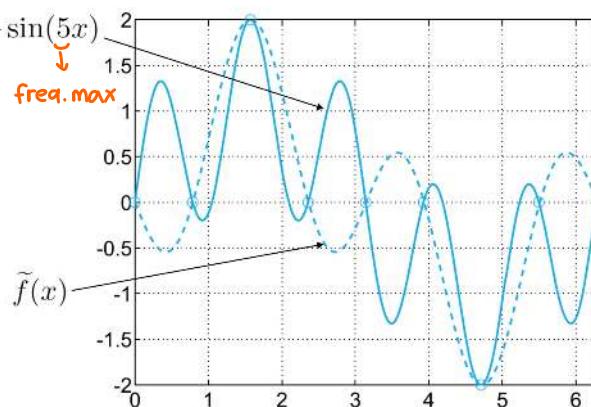
Qualora il numero di dati $n+1$ non sia sufficientemente elevato, l'interpolatore trigonometrico non è in grado descrivere le frequenze k più alte

Sia k_{max} la frequenza massima della funzione $f(x)$. Allora, se $n \leq 2k_{max}$, la frequenza massima dell'interpolatore $\tilde{f}(x)$ è minore di k_{max} (aliasing). Nell'esempio a destra, $n=7$ e $2k_{max} = 10$.

Si deve quindi avere

$$n > 2k_{max}$$

(Teorema di Shannon)



Gli effetti dell'aliasing. Confronto tra la funzione $f(x) = \sin(x) + \sin(5x)$ (in linea continua) ed il suo interpolatore trigonometrico $\tilde{f}(x)$ con $M = 3$ (linea tratteggiata)

Metodo dei minimi quadrati

Supponiamo di avere $n+1$ misure (x_i, y_i) $i = 0, \dots, n$ e di volere **estrapolare altri dati** fuori dal dominio. Abbandono vincolo interpolatorio. → Problema di **previsione**
 → cerco una curva semplice globale (retta, parabola, ...)

IDEA: cerco polinomio di grado $m < n$ che approssimi le misure in un senso opportuno.

In particolare, cerco un polinomio di grado m che minimizza lo scarto quadratico medio.

MINIMI QUADRATI : cerco $q \in \mathbb{P}^m$: $\sum_{i=0}^n (q(x_i) - y_i)^2 \leq \sum_{i=0}^n (p(x_i) - y_i)^2 \quad \forall p \in \mathbb{P}^m$

insieme di tutti i polinomi possibili di grado $\leq m$

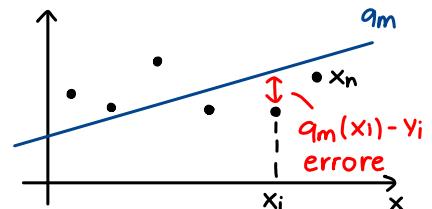
cerco il polinomio q di grado m tale che la somma, con i che va da 0 a n , di q valutato nei nodi meno le mie misure (al quadrato), sia più piccolo dell'analogo errore che commetto se calcolo lo scarto quadratico medio per ogni altro polinomio di grado m .

Esempio : $m=1 \rightarrow$ RETTA DI REGRESSIONE . Determiniamo $q_1(x) = a_0 + a_1 x$

$$\text{chiamo } \Phi(b_0, b_1) = \sum_{i=0}^n (y_i - p(x_i))^2 = \sum_{i=0}^n (y_i - (b_0 + b_1 x_i))^2 \quad p \in \mathbb{P}^1$$

Chiamerò (a_0, a_1) il punto di minimo :

$$\Phi(a_0, a_1) = \min_{b_0, b_1} \Phi(b_0, b_1)$$



$$\Phi(b_0, b_1) = \sum_{i=0}^n (y_i^2 + b_0^2 + b_1^2 x_i^2 + 2b_0 b_1 x_i - 2b_0 y_i - 2b_1 x_i y_i)$$

Impongo $\frac{\partial \Phi}{\partial b_0} = 0$ e $\frac{\partial \Phi}{\partial b_1} = 0$ per trovare minimo a_0, a_1

$$\left\{ \begin{array}{l} \sum_{i=0}^n 2(b_0 + b_1 x_i - y_i) = 0 \\ \sum_{i=0}^n 2(b_1 x_i^2 + b_0 x_i - x_i y_i) = 0 \end{array} \right.$$

$$a_0 = \frac{1}{D} \sum_{i=0}^n y_i \cdot \sum_{j=0}^n x_j^2 - \sum_{j=0}^n x_j \cdot \sum_{i=0}^n x_i y_i$$

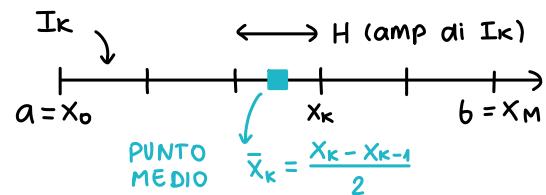
$$b_0 = \frac{1}{D} (n+1) \sum_{i=0}^n x_i y_i - \sum_{j=0}^n x_j \cdot \sum_{i=0}^n y_i$$

$$D = (n+1) \sum_{i=0}^n x_i^2 - (\sum_i x_i)^2$$

4. INTEGRAZIONE NUMERICA

Formule di quadratura

Voglio calcolare l'integrale $I(f) = \int_a^b f(x) dx \rightarrow$ suddivido $[a, b]$ in sottointervalli I_k



DEFINIZIONI

Una **formula di quadratura** (F.Q)

(approssimazione integrale) fornisce
l'approssimazione di $I(f)$ con $I_H(f)$.

$$H = \frac{b-a}{M}; x_k = a + kH; k=0, \dots, M$$

DEF1 : La formula di quadratura si dice di **ordine p** se $E_H = |I(f) - I_H(f)| \leq C H^p$

\downarrow ERRORE COSTANTE \downarrow

più p è grande, più la convergenza
è veloce

DEF2 : La formula di quadratura si dice avere **grado di esattezza pari a r** se, applicata ai polinomi di grado minore o uguale a r , essa è esatta. Cioè :

$$E_H = I(f) - I_H(f) = 0 \quad \forall f \in P^r$$

1. Formula del punto medio composita

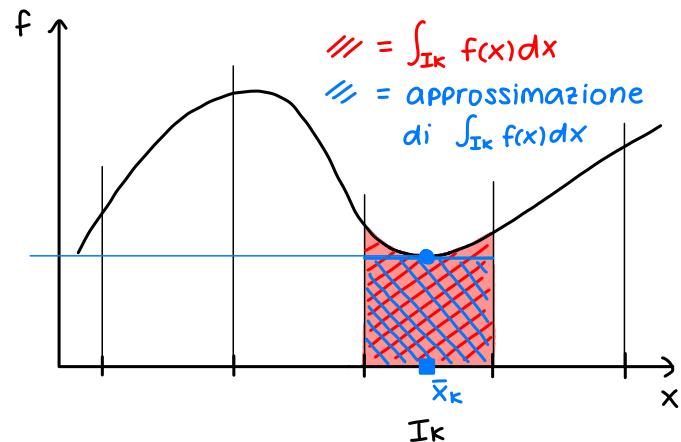
Parto dall'additività dell'integrale :

$$I(f) = \int_a^b f(x) dx = \sum_{k=1}^M \int_{I_k} f(x) dx$$

IDEA : Approssimo questi integrali $\int_{I_k} f(x) dx$ e poi sommo.

In particolare, approssimo $f(x)$ in I_k come una costante pari a $f(\bar{x}_k)$.

Poi approssimo $\int_{I_k} f(x) dx$ con l'area del rettangolo di base I_k e altezza $f(\bar{x}_k)$.



Quindi introduco l'approssimazione del punto medio composita :

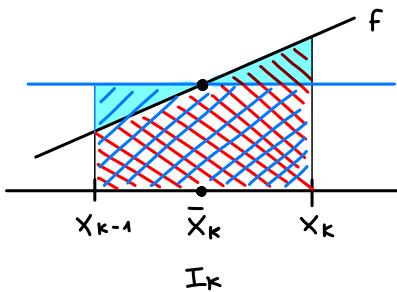
$$I_{PM}^H(f) = H \sum_{k=1}^M f(\bar{x}_k)$$

Si può mostrare (vedi slide) che l'errore è dato da, se $f \in C^2([a, b])$

$$E_{PM}^H = |I(f) - I_{PM}^H(f)| \leq \max_{x \in [a, b]} |f''(x)| \frac{b-a}{24} H^2 \rightarrow \text{METODO DI ORDINE } p=2$$

Inoltre, noto per le rette l'errore è identicamente nullo $\forall H$ perché $f''=0 \quad \forall f \in P^1 \rightarrow \text{GRADO DI ESATTEZZA } r=1$

NOTA : $r=1$ (G.d.E.) Infatti :

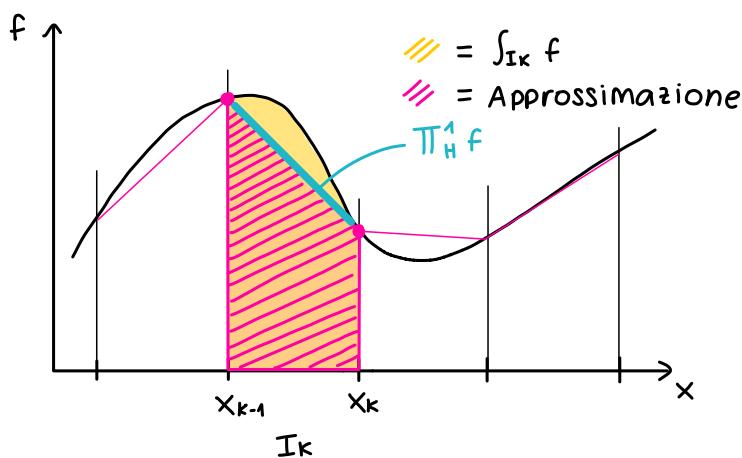


AREA = AREA
(dimostrazione grafica)

2. Formula dei trapezi composita

Questa volta x_{k-1} e x_k , non \bar{x}_k , per approssimare f .

IDEA : Approssimo $\int_{I_k} f(x) dx$ con l'area del trapezio costruito su $(x_{k-1}, f(x_{k-1}))$ e $(x_k, f(x_k))$



$$I_{TR}^H(f) = \frac{H}{2} \sum_{k=1}^M (f(x_{k-1}) + f(x_k))$$

A volte, si implementa così :

$$I_{TR}^H(f) = \frac{H}{2} (f(a) + f(b)) + \sum_{k=1}^{M-1} H f(x_k)$$

dimezzo circa i costi

NOTA : $I_{TR}^H = \int_a^b \Pi_H^1 f(x) dx$ può essere visto come l'integrale esatto di $\Pi_H^1 f(x)$.
Poiché conosco l'errore commesso da $\Pi_H^1 f$, allora so stimare E_{TR}^H :

$$E_{TR}^H = | I(f) - I_{TR}^H(f) | \leq \max_{x \in (a,b)} | f''(x) | \frac{b-a}{12} H^2$$

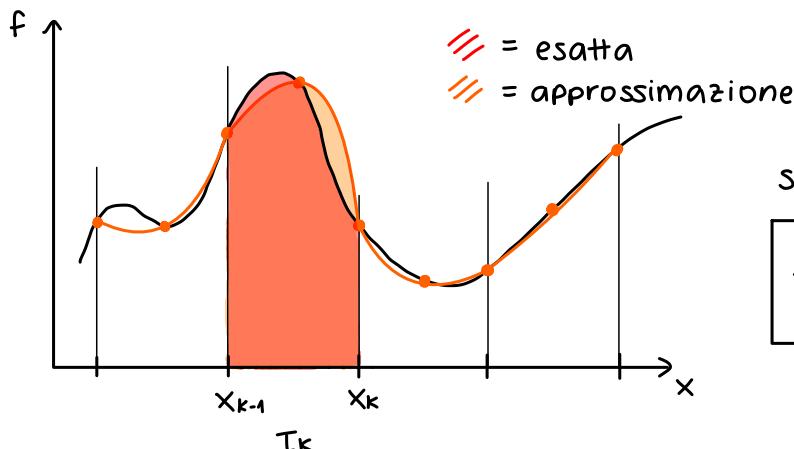
circa il doppio
del primo metodo

→ ORDINE $p=2$

→ GRADO DI ESATTEZZA $r=1$

3. Formula di Simpson composita

IDEA: Considero su I_k sia x_{k-1}, x_k che \bar{x}_k e approssimo f su I_k con la parabola che passa per tali 3 punti ($\Pi_H^2 f(x)$). Calcolo le aree sottese e sommo:



Si ottiene :

$$I_{\text{SIM}}^H(f) = \frac{H}{6} \sum_{k=1}^M f(x_{k-1}) + 4f(\bar{x}_k) + f(x_k)$$

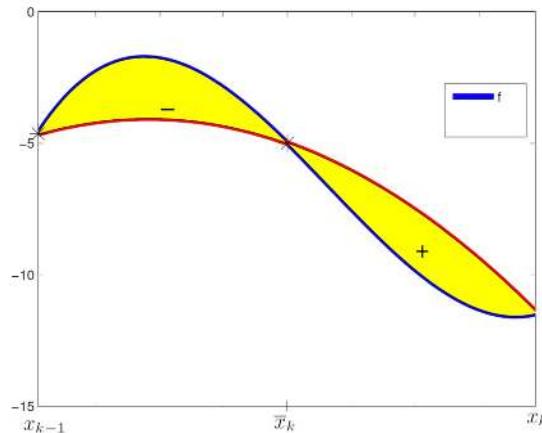
$I_{\text{SIM}}^H(f)$ è l'integrale esatto di $\Pi_H^2 f$. Si ottiene :

$$E_{\text{SIM}}^H = | I(f) - I_{\text{SIM}}^H(f) | \leq \max_x | f^{(iv)} | \cdot \frac{b-a}{2880} H^4$$

→ ORDINE $p=4$
→ GRADO $r=3$

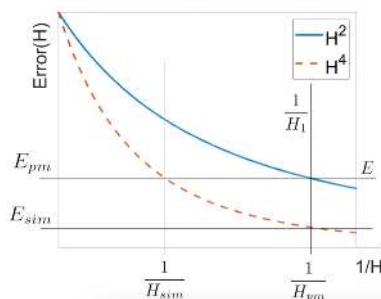
Tale formula quindi non integra esattamente soltanto i polinomi che sono globalmente di grado $r=1$ (rette) e grado $r=2$ (parabole), ma anche i polinomi che sono globalmente di grado 3 (cubiche).

Infatti, come si evince dalla figura, gli errori commessi su ogni intervallo si elidono.



Ricapitolando abbiamo:

FORMULA DI INTEGRAZIONE COMPOSITA	ORDINE p RISPETTO AD H	GRADO r DI ESATTEZZA
Punto medio	2	1
Trapezi	2	1
Simpson	4	3



Per $H=H_1$ fissato, l'errore E_{pm} commesso dalla formula del punto medio è maggiore dell'errore E_{sim} commesso dalla formula di Simpson.

Affinché l'errore scenda al di sotto di una certa soglia E , il valore di H_{sim} richiesto dalla formula di Simpson è maggiore del valore H_{pm} richiesto dal punto medio ($1/H_{\text{sim}} < 1/H_{\text{pm}}$)

TRAPEZI: Π_H^1 → $r=1$ rette → $r=2$ parabole

SIMP: Π_H^2 → $r=3$

Π_H^K CON K DISPARI → $r=K$
CON K PARI → $r=K+1$

5. APPROXIMAZIONE DI EQUAZIONI DIFFERENZIALI ORDINARIE (ODE)

Siamo interessati al problema di Cauchy:

Data $f: [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$, determinare $y: [0, T] \rightarrow \mathbb{R}$ tale che:

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(0) = y_0 \end{cases} \quad \begin{array}{l} \text{dipendenza in generale non} \\ \text{lineare di } f \text{ da } y \text{ e } t \end{array}$$

dato iniziale

Esempio $f(t, y) = 3y - 3t$, $y_0 = 1$
 → PB di Cauchy: $\begin{cases} y'(t) = 3y - 3t \\ y(0) = 1 \end{cases} \quad \rightarrow \quad y(t) = \frac{2}{3}e^{3t} + t + \frac{1}{3}$ CASO PARTICOLARE

Esempio $f(t, y) = \frac{t^2 - \sin(y^3)}{\ln(y)}$

$$\begin{cases} y'(t) = \frac{t^2 - \sin(y^3)}{\ln(y)} \\ y(0) = 3 \end{cases} \quad \begin{array}{l} \text{NON ESISTONO SOLUZIONI ANALITICHE} \\ \downarrow \\ \text{MA LA SOLUZIONE ESISTE} \\ \downarrow \\ \text{INTRODUCO I METODI NUMERICI} \end{array}$$

Per approssimare il PB di Cauchy, devo prima imparare ad approssimare le derivate:

APPROXIMAZIONE DERIVATE

PB DI CAUCHY → difficile

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(0) = y_0 \end{cases}$$

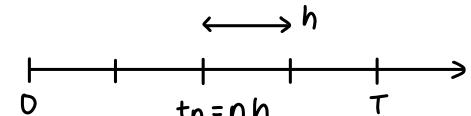
conosco $y'(t)$ (perché f è nota)
 e devo determinare $y(t)$
 Approssimo con U_n

APPROXIMAZIONE DERIVATE → facile

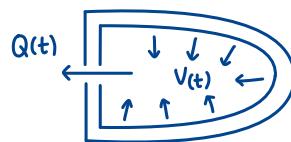
Data $y(t)$, trova $y'(t)$

conosco $y(t)$, voglio determinare $y'(t)$
 → è il contrario
 ↓
 Problema che sorge se conosco $y(t)$
 solo PER PUNTI

PB: conosco $y(t)$ in alcuni punti t_n , $n=0, \dots, N$



Esempio Flusso statico attraverso la valvola aortica

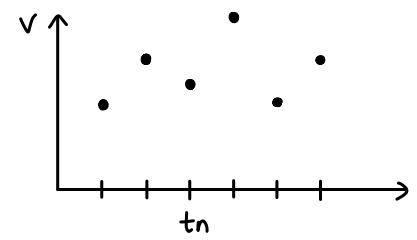


$Q(t) = \frac{dV(t)}{dt}$ ho misure di $V(t)$ in istanti t_n (solo 20 o 30) da IMMAGINI CINE-MRI
 → per calcolare $Q(t)$ devo approssimare una derivata perché conosco $V(t)$ solo in alcuni punti

Formula di Eulero in avanti e all'indietro

SVILUPPO IN SERIE DI TAYLOR per una generica $v(t)$:

$$v(t_{n+1}) = v(t_n) + h v'(t_n) + \frac{h^2}{2} v''(t_n) + \frac{h^3}{6} v'''(t_n) + \Theta(h^4)$$



$$\lim_{h \rightarrow 0} \frac{Q}{h^p} = \begin{cases} 0 & p < 1 \\ \text{oppure} & \\ L < +\infty & \end{cases}$$

Un $\Theta(h^4)$ è una quantità Q che va a zero almeno tanto velocemente quanto h^p
es: $S = 7h^4 + h^2 = \Theta(h^2)$

$$\rightarrow v'(t_n) = \frac{v(t_{n+1}) - v(t_n)}{h} \quad \boxed{-\frac{h}{2} v''(t_n) - \frac{h^2}{6} v'''(t_n) + \Theta(h^3)}$$

$\Theta(h)$ $\Theta(h^2)$

$\rightarrow \Theta(h)$

EULERO IN AVANTI :

$$v'(t_n) \simeq \frac{v(t_{n+1}) - v(t_n)}{h}$$

APPROXIMAZIONE DERIVATA } $D^+ v(t_n)$

ERRORE : $|v'(t_n) - D^+ v(t_n)| = \Theta(h)$ METODO DI ORDINE 1 → Formula conveniente perché errore va a zero per $h \rightarrow 0$ (ha convergenza)

Alternativamente : (EULERO ALL'INDIETRO)

$$v(t_{n-1}) = v(t_n) - h v'(t_n) + \frac{h^2}{2} v''(t_n) + \Theta(h^3)$$

$$\rightarrow v'(t_n) = \frac{v(t_n) - v(t_{n-1})}{h} + \Theta(h)$$

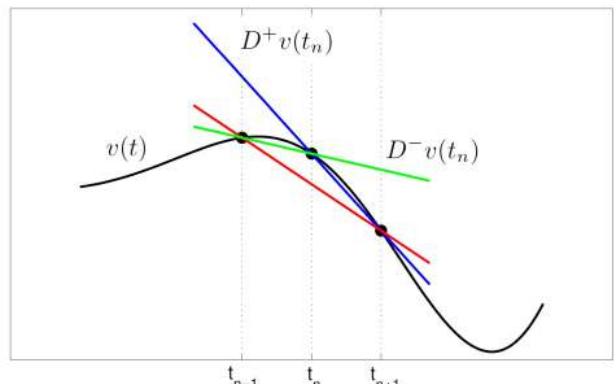
approssimazione di Eulero all'indietro } $D^- v(t_n)$

ERRORE : $|v'(t_n) - D^- v(t_n)| = \Theta(h)$ ORDINE 1

Posso anche introdurre un'approssimazione centrata :

$$D^c v(t_n) = \frac{v(t_{n+1}) - v(t_{n-1})}{2h};$$

Esempi di rappresentazione grafica:



$$|v'(t_n) - D^c v(t_n)| = \Theta(h^2)$$

convergente, ORDINE 2
(errore più piccolo)

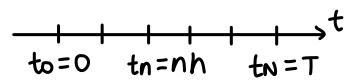
Approssimazioni in avanti (blu) e all'indietro (verde) della derivata prima, introdotte nelle prossime slides. In rosso una terza approssimazione (centrata) che non verrà qui utilizzata

APPLICAZIONE AL PB DI CAUCHY

$$\begin{cases} y'(t) = f(t, y(t)) & t \in [0, T] \\ y(0) = y_0 \end{cases}$$

REGOLA GENERALE

1. Scelgo una formula per approssimare la derivata prima (EA, EI, centrale, ...)
2. Mi concentro su t_n : $y'(t_n) = f(t_n, y(t_n))$
3. Sostituisco $y'(t_n)$ con formula approssimante scelta al punto 1.
→ Perdo l'uguaglianza
4. Ripristino l'uguaglianza sostituendo $y(t_n)$ con soluzione numerica u_n .
Quindi u_n sarà l'approssimazione di $y(t_n)$ → $u_n \approx y(t_n)$



Metodo di Eulero in avanti

1. Scelgo Eulero in avanti $v'(t_n) \approx \frac{v(t_{n+1}) - v(t_n)}{h}$

2. $y'(t_n) = f(t, y(t_n))$

3. $\frac{y(t_{n+1}) - y(t_n)}{h} \approx f(t, y(t_n))$

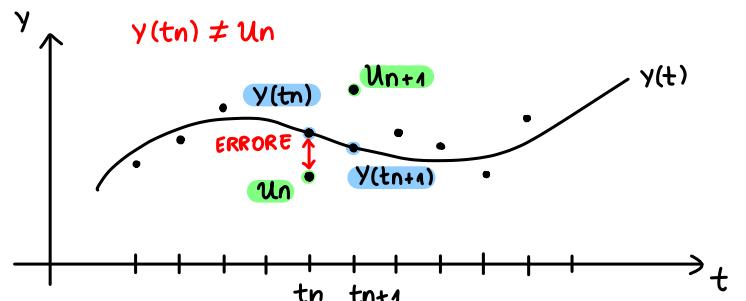
4. $\frac{u_{n+1} - u_n}{h} = f(t, u_n)$

METODO EA

$$u_0 = y_0$$

$$\frac{u_{n+1} - u_n}{h} = f(t_n, u_n), \quad \forall n \geq 0$$

Nota sulla nomenclatura: la formula di Eulero in avanti (ecc) si usa per risolvere il problema di approssimare le derivate, mentre il Metodo di Eulero in avanti (ecc) si usa per risolvere il problema di Cauchy. Sono problemi inversi e diversi.



NOTA: n è indice temporale, non di iterazione

Operativamente: $u_{n+1} = \boxed{u_n + h \cdot f(t, u_n)}$

Il Metodo di EA è **ESPLICITO** → ho un'espressione esplicita di u_{n+1} in funzione di u_n (che è noto)

cioè evita di risolvere le non-linearietà

Esempio $y'(t) = \frac{3e^{y(t)}}{y^2(t)-1} - \operatorname{tg}\left(\frac{1}{y(t)}\right)$ \rightsquigarrow f altamente non lineare rispetto a y

so valutare una funzione non-lineare in un PUNTO CHE CONOSCO!

EA : $u_{n+1} = u_n + h \left(\frac{3e^{u_n}}{u_n^2 - 1} - \operatorname{tg}\left(\frac{1}{u_n}\right) \right)$

Espressione esplicita di u_{n+1} (nonostante la non-linearietà)

metodo altamente efficiente :

```

 $u_0 = y_0$ 
for  $n=1:N$ 
     $u_{n+1} = u_n + h * f(t_n, u_n)$ 
end
  
```

Alla fine ho il vettore degli u_n da plottare

Metodo di Eulero all'indietro

$$1. \text{ Scelgo Eulero all'indietro } v'(t_n) \simeq \frac{v(t_{n-1}) - v(t_n)}{h} \leftrightarrow v'(t_{n+1}) \simeq \frac{v(t_{n+1}) - v(t_n)}{h}$$

$$2. y'(t_{n+1}) = f(t_{n+1}, y(t_{n+1}))$$

$$3. \frac{y(t_{n+1}) - y(t_n)}{h} \simeq f(t_{n+1}, y(t_{n+1}))$$

$$4. \frac{u_{n+1} - u_n}{h} = f(t_{n+1}, u_{n+1})$$

METODO EI

$$u_0 = y_0$$

$$\frac{u_{n+1} - u_n}{h} = f(t_{n+1}, u_{n+1}) \quad \forall n \geq 0$$

NON NOTO

$$\text{Operativamente: } u_{n+1} = u_n + h \cdot f(t_{n+1}, u_{n+1})$$

Il metodo di EA è IMPLICITO (o NON ESPLICITO) \rightarrow non conosco u_{n+1}

cioè fa sì che si debba risolvere le NON-LINEARITÀ

$$\text{Esempio: } y'(t) = \frac{3e^{y(t)}}{y^2 - 1} - \operatorname{tg}\left(\frac{1}{y(t)}\right) ; \quad y(0) = 1$$

$$u_{n+1} = u_n + h \left(\frac{3e^{u_{n+1}}}{u_{n+1}^2 - 1} - \operatorname{tg}\left(\frac{1}{u_{n+1}}\right) \right)$$

cioè devo determinare la radice di: $g(x) = x - u_n - h \left(\frac{3e^x}{x^2 - 1} - \operatorname{tg}\left(\frac{1}{x}\right) \right) \rightarrow \text{NEWTON}$

\rightarrow Ad ogni n devo risolvere un PB non-lineare

\rightarrow Ad ogni n devo chiamare NEWTON

```

 $u_0 = y_0$ 
for  $n = 1 : N$ 
     $x^{(0)} = u_n$ 
    for  $k = 0 : K_{MAX}$ 
         $x^{(k+1)} = x^{(k)} - g(x^{(k)}) / g'(x^{(k)})$ 
        if  $|g(x^{(k+1)})| < \epsilon$ 
            then  $u_{n+1} = x^{(k+1)}$ 
            break
        end
    end
end

```

$\hookrightarrow n \rightarrow n+1$

\leftarrow Se Newton converge

$$\lim_{k \rightarrow +\infty} x^{(k)} = u_{n+1}$$

$$\text{con } g(x) = x - u_n - h f(t_{n+1}, x) \\ g'(x) = 1 - h \frac{\partial f}{\partial x}(t_{n+1}, x)$$

COSTO COMPUTAZIONALE ELEVATO (Ad ogni t_n)

EULERO ALL'INDIETRO - Fixed point Iterations

In alternativa a Newton, posso usare il metodo di iterazioni di punto fisso

$$g(x) = x - u_n - h f(t_{n+1}, x)$$

\downarrow CERCO PUNTI FISSI DI $\phi(x) = u_n + h f(t_{n+1}, x)$

Punti fissi soddisfano: $x = u_n + h f(t_{n+1}, x) \Rightarrow x = u_{n+1}$

SE ITERAZIONI CONVERGONO

$$\lim_{k \rightarrow +\infty} x^{(k)} = u_{n+1}$$

$$x^{(k+1)} = \frac{u_n + h f(t_{n+1}, x^{(k)})}{\Phi(x^{(k)})} \quad \text{FINO A CONVERGENZA}$$

Assoluta stabilità

15 - 24/05/23

Vogliamo discutere di seguito quando la soluzione numerica fornita dai metodi di Eulero sia esente da oscillazioni che aumentano indefinitamente producendo una soluzione non stabile e quindi non accettabile.

Ci chiediamo cosa succeda quando usiamo nei nostri codici un valore di h più grande. I metodi di Eulero producono ancora una soluzione stabile?

Esempio $\begin{cases} y'(t) = -2y(t) \\ y(0) = 1 \end{cases} \rightarrow f(t, y) \text{ lineare}$

$$\begin{aligned} EA \quad u_{n+1} &= u_n + h f(t_n, u_n) \\ &= u_n + h (-2u_n) \\ &= (1 - 2h) u_n \end{aligned}$$

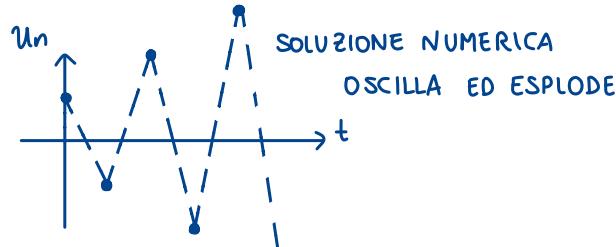
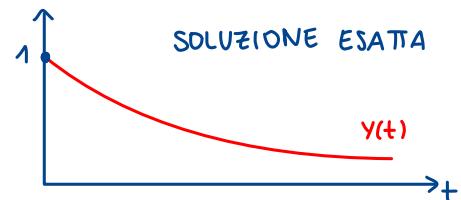
$h = 1.1 \rightarrow u_{n+1} = -1.2 u_n$

$$u_0 = 1$$

$$u_1 = -1.2(u_0) = -1.2$$

$$u_2 = -1.2(u_1) = 1.44$$

$$u_3 = -1.2(u_2) = -1.73$$



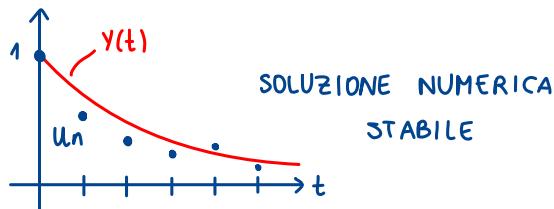
$h = 0.2 \rightarrow u_{n+1} = 0.6 u_n$

$$u_0 = 1$$

$$u_1 = 0.6(u_0) = 0.6$$

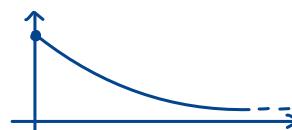
$$u_2 = 0.6(u_1) = 0.36$$

$$u_3 = 0.6(u_2) = 0.22$$



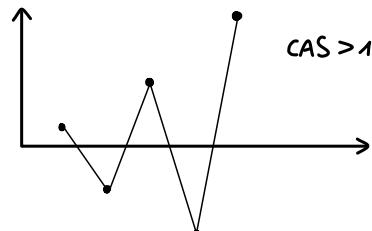
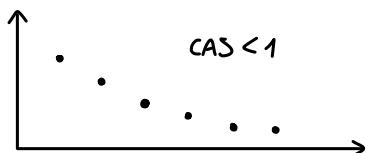
Per determinare il valore di h di "CUT-OFF" (stabilità vs Non stabilità) si analizza il PROBLEMA MODELLO.

$$\begin{cases} y'(t) = \lambda y(t), \quad h < 0 \\ y(0) = 1 \end{cases} \rightarrow y(t) = e^{\lambda t}$$



Un metodo numerico si dice ASSOLUTAMENTE STABILE se, applicato al problema modello, dà:

$$|u_{n+1}| \leq \text{CAS} |u_n| \quad \text{CAS} \leq 1 \quad (< 1) \rightarrow u \text{ decresce} \quad (\text{CAS} < 1) \text{ o non cresce} \quad (\text{CAS} \leq 1)$$



Se applico questa definizione al caso di dominio "illimitato" $(t_0, +\infty)$.

Allora A.S. $\lim_{n \rightarrow +\infty} |u_n| = 0$

ANALISI ASSOLUTA STABILITÀ DI EA

$$\begin{cases} y'(t) = \lambda y(t), \lambda < 0 \\ y(0) = 1 \end{cases} \xrightarrow{\text{EA}} \begin{aligned} u_{n+1} &= u_n + h f(t_n, u_n) \\ &= u_n + h \lambda u_n \\ &= (1 + h\lambda) u_n \end{aligned}$$

EA applicato al problema modello

HO A.S. se h soddisfa $|u_{n+1}| \leq C_{AS} u_n$ con $C_{AS} < 1$

$$C_{AS} = |1 + h\lambda| < 1 \Rightarrow -1 < 1 + h\lambda < 1$$

$$-1 < 1 + h\lambda$$

$$h\lambda > -2$$

$$\boxed{h < -2/\lambda}$$

$$1 + h\lambda < 1$$

$$h\lambda < 0$$

OK sempre

CONDIZIONE DI A.S. per EA

Nota: l'assoluta stabilità dipende dal valore di h scelto e non dal modello in se. Dire che un modello è assolutamente stabile, in generale, non ha senso. Ha invece senso affermare che il metodo è assolutamente stabile per un determinato valore di h . È un concetto diverso dalla convergenza, che invece dipende dal metodo in se.

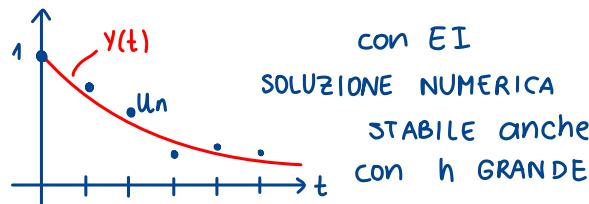
Infatti, nell'esempio di prima A.S. garantita per $h < 1$ (con $\lambda = -2$)

ANALISI ASSOLUTA STABILITÀ DI EI

Esempio $\begin{cases} y'(t) = -2y(t) \\ y(0) = 1 \end{cases}$

$$\begin{aligned} \text{E.I. : } u_{n+1} &= u_n + h f(t_{n+1}, u_{n+1}) \\ &= u_n - 2h u_{n+1} \\ &\downarrow \\ u_{n+1} &= \frac{1}{1+2h} u_n \end{aligned} \quad \text{è lineare} \rightarrow \text{NO NEWTON!}$$

$$\begin{aligned} h = 1.1 \rightarrow u_0 &= 1 \\ u_1 &= \frac{1}{1+2 \cdot 1.1} (u_0) = 0.3 \\ u_2 &= 0.09 \\ u_3 &= 0.027 \end{aligned}$$



In generale : $u_{n+1} = u_n + h \lambda u_{n+1}$

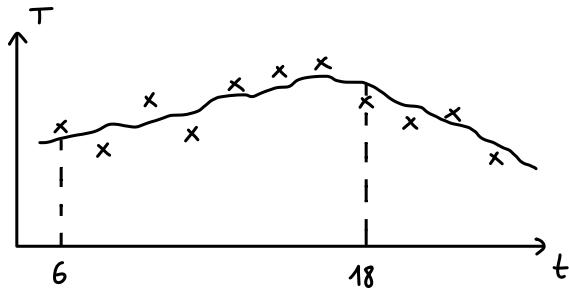
$$\text{PROBLEMA MODELLO : } u_{n+1} = \frac{1}{1-h\lambda} u_n \quad \text{con} \quad C_{AS} = \left| \frac{1}{1-h\lambda} \right| < 1 \quad \text{sempre } \forall h$$

Eulero all'indietro non esplode mai! Questo non vuol dire che EI sia accurato, se h è grande commetto grande errore!

Questo compensa il costo computazionale elevato di EI

E.A. vs E.I.

1) variazione temperatura in un giorno



x = soluzione numerica

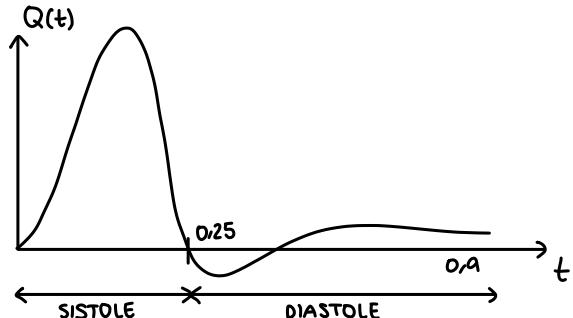
Non è necessario h piccolo perché $y(t)$ varia di poco \rightarrow prendo h grande ($h=1$ ora)

Non ho esigenza di catturare una dinamica veloce

E.I. è da preferirsi

Mi aspetto accuratezza buona

2) FLUSSO emodinamico in carotide



Ho bisogno di h piccolo se sono interessato alla sistole \rightarrow ho una dinamica veloce

E.A. è da preferirsi

ESTENSIONE AL CASO GENERALE (f non lineare)

Abbiamo introdotto il concetto di assoluta stabilità per il problema modello.

Cosa possiamo dire per un problema di Cauchy generale?

Diremo che un metodo numerico produce una soluzione stabile se a «piccole» perturbazioni sul dato iniziale si producono perturbazioni «piccole» e decrescenti per n crescente

Noto che se $f = \lambda y \rightarrow \frac{\partial f}{\partial y} = \lambda$

Estensione: prendi come soglia per h quella ottenuta per PB modello e sostituisci λ con

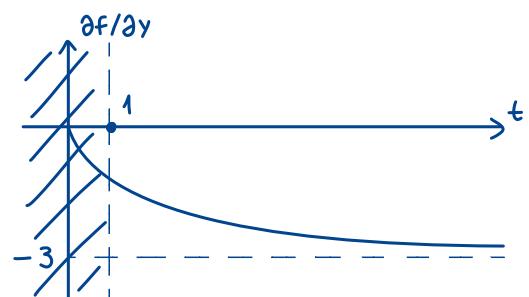
$$\bar{\lambda} = - \max_{t,y} \left| \frac{\partial f}{\partial y} \right|, \quad \frac{\partial f}{\partial y} < 0$$

Esempio EA è A.S. in generale per $h < -\frac{2}{\lambda}$

Esempio $\begin{cases} y'(t) = \underbrace{\arctan(3y)}_{f(t,y)} - 3y + t, \\ y(1) = 1 \end{cases} \quad t > 1$

$$\bar{\lambda} = - \max_{t,y} \left| \frac{\partial f}{\partial y} \right| = -3$$

E.A. A.S. per $h < 2/3$



Metodo di Crank-Nicolson

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(0) = 1 \end{cases}$$

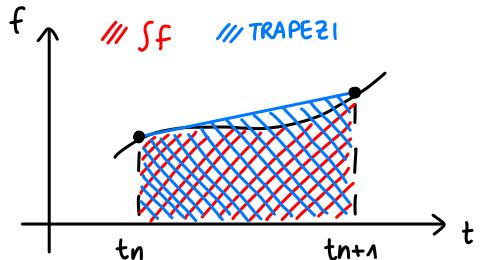
Dal teorema fondamentale del calcolo integrale su $[t_n, t_{n+1}]$ posso scrivere la soluzione esatta:

$$y(t_{n+1}) = y(t_n) + \underbrace{\int_{t_n}^{t_{n+1}} f(t, y(t)) dt}_{\text{USO formula di quadratura} \rightarrow \text{TRAPEZI (ad es.)}}$$

USO formula di quadratura \rightarrow TRAPEZI (ad es.)

$$y(t_{n+1}) \simeq y(t_n) + \frac{h}{2} [f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+1}))]$$

METODO CN : $U_{n+1} = U_n + \frac{h}{2} [f(t_n, U_n) + f(t_{n+1}, U_{n+1})]$



E` un metodo **Implicito**: ad ogni t_{n+1} devo risolvere PB non lineare:
cerco radice di $g(x) = x - U_n - \frac{h}{2} [f(t_n, U_n) + f(t_{n+1}, x)]$

?
Richiede
NEWTON / PUNTO FISSO

E` **incondizionatamente A.S.**:

$$y'(t) = \lambda y(t) \longrightarrow U_{n+1} = U_n + \frac{h}{2} (\lambda U_n + \lambda U_{n+1})$$

$$\longrightarrow U_{n+1} = \left(\frac{1}{1 - \frac{h\lambda}{2}} \right) \left(1 + \frac{\lambda h}{2} \right) U_n = \frac{2 + \lambda h}{2 - \lambda h} U_n$$

$$C_{AS} = \left| \frac{2 + h\lambda}{2 - h\lambda} \right| < 1 \quad \forall h \quad (\text{perché } \lambda < 1)$$

Metodo di Heun

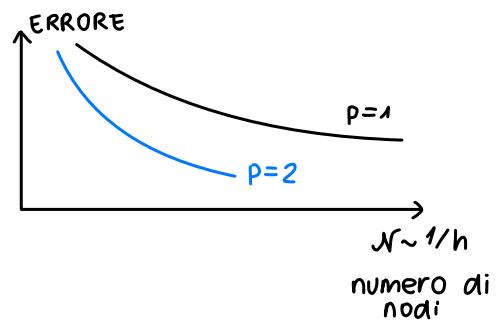
Partiamo da CN : $U_{n+1} = U_n + \frac{h}{2} (f(t_n, U_n) + f(t_{n+1}, \underline{U_{n+1}}))$

HEUN : $U_{n+1} = U_n + \frac{h}{2} (f(t_n, U_n) + f(t_{n+1}, \underline{U_n + h f(t_n, U_n)}))$ Io approssimo con E.A.

HEUN e' **esplicito** e A.S. per $h < -2/\lambda$

Accuratezza

convergenza : $\lim_{h \rightarrow 0} |y(t_n) - u_n| = \Theta(h^p)$, $\forall n$
 di ordine p



METODO	ESPLICITO O IMPLICITO	ORDINE	A.S.
EA	Esplicito	1	$h < -2/\bar{\lambda}$
EI	Implicito	1	$\forall h$
CN	Implicito	2	$\forall h$
HEUN	Esplicito	2	$h < -2/\bar{\lambda}$