

Appunti Calcolo Numerico

Matteo Menichetti

6 dicembre 2024

Questi appunti sono da intendersi come base di studio. Eventuali errori di battitura, anche nella notazione matematica, possono essere corretti avvertendo.

Indice

1 Aritmetica finita	4
1.1 Errori di discretizzazione	4
1.2 Errori di convergenza	5
1.3 Errori di round-off	5
1.3.1 Numeri interi	6
1.3.2 Numeri reali	6
1.3.3 Overflow e Underflow	10
1.3.4 Standard IEEE 754	10
1.3.5 Aritmetica Finita	12
1.4 Condizionamento di un problema	13
2 Radici di un'equazione (nonlineare)	18
2.1 Metodo di bisezione	19
2.2 Criterio di Arresto	19
2.2.1 Criterio di arresto per il metodo di bisezione	19
2.2.2 Condizionamento di una radice	21
2.3 Convergenza di un metodo iterativo	23
2.4 Ordine di Convergenza di un metodo iterativo	24
2.4.1 Metodo di bisezione	25
2.5 Metodo di Newton	25
2.5.1 Convergenza del metodo di Newton	28
2.5.2 Studio della convergenza locale	30
2.5.3 Criteri d'arresto per il metodo di Newton (e non solo)	32
2.5.4 Caso delle radici multiple	34
2.5.4.1 Molteplicità $m > 1$ nota	34
2.5.5 Molteplicità $m > 1$ ignota	34
2.6 Metodo di Aitken	35
2.7 Metodi quasi-Newton	35
2.7.1 Metodo delle corde	37
2.7.2 Metodo delle secanti	37

3 Sistemi Lineari e Nonlineari	40
3.1 Sistemi lineari: casi semplici	40
3.1.1 Caso \mathbf{A} diagonale	40
3.1.2 Caso \mathbf{A} triangolare	41
3.1.3 Caso \mathbf{A} ortogonale	46
3.2 Fattorizzazione LU di una matrice	47
3.3 Costo computazionale	55
3.4 Matrici a diagonale dominante	56
3.5 Matrici Simmetriche e Definite Positive	58
3.6 Pivoting	66
3.6.1 Fattorizzazione LU con pivoting parziale	67
3.7 Condizionamento del problema	72
3.8 Sistemi lineari sovradianimensionati	76
3.8.1 Esistenza della fattorizzazione QR	79
3.8.2 Analisi complessità dell'algoritmo di fattorizzazione QR (Householder)	83
3.9 Risoluzione di sistemi nonlineari	85
3.10 Intermezzi	86
3.10.1 Norme su vettori	87
3.10.2 Norme indotte su matrice	87
4 Approssimazioni di funzioni	89
4.1 Polinomio Interpolante	89
4.2 Forma di Lagrange e di Newton	91
4.3 Interpolazione di Hermite	102
4.4 Errore nell'interpolazione polinomiale	106
4.5 Condizionamento del problema	115
4.5.1 Connessioni tra condizionamento ed errore dell'interpolazione polinomiale	117
4.6 Ascisse di Chebyshev	120
4.7 Interpolazione mediante funzioni spline	124
4.8 Spline cubiche	129
4.8.1 Spline cubica naturale	130
4.8.2 Spline cubica completa	130
4.8.3 Spline cubica interpolante periodica	130
4.8.4 Spline cubica interpolante not-a-knot	131
4.9 Calcolo (pratico) di una spline cubica	133
4.10 Approssimazione polinomiale nel senso dei minimi quadrati	136
4.11 Risoluzione di un sistema tridiagonale	140
5 Formule di quadratura numerica (approssimazione di integrali definiti)	143
5.1 Formule di Newton-Cotes	143
5.1.1 Condizionamento di una formula di Newton-Cotes	145
5.2 Errore (di quadratura) e formule composite	147
5.2.1 Errore di quadratura	147
5.2.2 Formula dei trapezi composita	148
5.2.3 Formula di Simpson composita	149
5.3 Formule di Newton-Cotes adattive	152
5.3.1 Formula dei trapezi	152

5.3.2	Formula di Simpson	153
6	Esercitazione capitoli 1 e 2	156
6.1	A.A. 2022/23	156
7	Esercitazione capitolo 3	160
7.1	A.A. 2022/23	160
8	Esercitazione capitoli 4 e 5	167
8.1	A.A. 2022/23	167
9	Esonero 1	175
10	Esonero 2	179
11	Argomenti trattati che sono intermezzi	181

1 Aritmetica finita

Supposto di avere un problema matematico, questo ha soluzione esatta $x \in \mathbb{R}^n$. L'utilizzo di un metodo numerico fornisce un'approssimazione (\tilde{x}) del risultato esatto (x). L'approssimazione genera un errore dovuto alla stessa.

È possibile misurare l'errore introdotto dall'approssimazione attraverso:

- Errore Assoluto: $\Delta x = \tilde{x} - x$;
- Errore Relativo: $\varepsilon_x = \frac{\tilde{x} - x}{x}$.

È possibile notare ciò che segue:

$$\begin{aligned}\varepsilon_x &= \frac{\Delta x}{x} = \frac{\tilde{x} - x}{x} \Rightarrow \tilde{x} = (1 + \varepsilon_x)x \\ \tilde{x} &= x + \Delta x\end{aligned}$$

Esempi di errore:

- $\Delta x = 10^{-3}$, $x = 10^{-3} \Rightarrow \varepsilon_x = 1$;
- $\Delta x = 10^{-3}$, $x = 10^{-6} \Rightarrow \varepsilon_x = 10^{-9}$;
- $x = \pi$, $\tilde{x} = 3.14 \Rightarrow \Delta x \approx -1.6 \cdot 10^{-3}$.

Gli errori assimilabili ad un metodo numerico sono individuabili in tre categorie distinte:

1. Errori di discretizzazione;
2. Errori di convergenza;
3. Errori di round-off.

1.1 Errori di discretizzazione

¹ Al fine di ottenere un metodo numerico che possa essere risolto da un calcolatore è necessario trasformare il problema **continuo**² in problema **discreto**³. È possibile notare che questo tipo di errore è dovuto alla definizione del metodo numerico utilizzato.

Supposto il problema da risolvere sia il calcolo della derivata $f'(x_0)$, dove $f : \mathbb{R} \rightarrow \mathbb{R}$, $x_0 \in (a, b)$, $f \in C^{(2)}[a, b]$, allora è possibile utilizzare lo sviluppo di Taylor di secondo ordine, per ottenere $f'(x_0)$, nel seguente modo:

Sia

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(\xi)}{2}(x - x_0)^2, \quad \xi \in (x_0, x),$$

allora segue che, sostituendo $x = x_0 + h$ (con $h > 0$ è quantità piccola ma finita), da

$$f(x_0 + h) = f(x_0) + f'(x_0)(x_0 + h - x_0) + \frac{f''(\xi)}{2}(x_0 + h - x_0)^2 = f(x_0) + f'(x_0)h + \frac{f''(\xi)}{2}h^2, \quad \xi \in (x_0, x_0 + h)$$

¹Slide 2 PDF lez1, PG 4.

²Ovvero dove la funzione da approssimare è definita come $f(x) : \mathbb{R} \rightarrow \mathbb{R}$.

³Ovvero dove la funzione da approssimare è definita come $f(x) : \mathbb{N} \rightarrow \mathbb{N}$.

⁴Tramite lo sviluppo di Taylor di $f(x)$ di punto iniziale x_0 con resto al secondo ordine, è ottenuto $f(x_0) + f'(x_0)h + \frac{f''(\xi)}{2}h^2 \rightarrow f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} - \frac{f''(\xi)}{2}h$, spostando i membri e dividendo per h come in (1.1). La serie è: $\sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$.

il problema dell'approssimazione della derivata può essere risolto mediante l'utilizzo del seguente rapporto incrementale:

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{f''(\xi)}{2}h = f'(x_0) - \frac{f''(\xi)}{2}h \approx \frac{f(x_0 + h) - f(x_0)}{h} = \varphi(x_0), \quad \xi \in (x_0, x_0 + h). \quad (1.1)$$

Pertanto, l'errore di discretizzazione è denotato come segue:

$$|f'(x_0) - \varphi(x_0)| = \frac{|f''(\xi)|}{2}h = O(h) \leq kh.$$

1.2 Errori di convergenza

⁵ I metodi numerici sono spesso iterativi, questo significa che non forniscono un risultato diretto ma una serie di risultati intermedi, una successione di approssimazioni del tipo $\{x_n\}$, definiti mediante una procedura iterativa

$$x_n = \Phi(x_{n-1}), \quad n = 1, 2, \dots, \quad (1.2)$$

dove n è il numero di iterazioni. Il metodo in questione converge se

$$\lim_{n \rightarrow \infty} x_n = x^*, \quad (1.3)$$

dove x^* è la soluzione esatta. La soluzione del problema quindi è fornita con infinite iterazioni.

È necessario definire un criterio d'arresto per la procedura (1.2), ovvero un numero finito di iterazioni n affinché il metodo numerico sia utilizzabile. In generale, qualunque sia n , $x_n \neq x^*$.

L'errore assoluto di convergenza è definito $x_n - x^*$ ed il criterio d'arresto è stabilito arrestando l'iterazione quando $n = N$ in modo che x_N sia sufficientemente accurata. Un'approssimazione è sufficientemente accurata se $|x_N - x^*| \leq \varepsilon$, con ε la tolleranza stabilita.

È possibile affermare che questo tipo di errore è dovuto dal metodo numerico utilizzato.

Esempio 1.1. La procedura iterativa che definisce un metodo convergente per calcolare $\sqrt{2}$ è

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{2}{x_n} \right), \quad n = 0, 1, 2, \dots, \quad x_0 = 2. \quad (1.4)$$

Precisazione sui due errori precedenti: Gli errori di discretizzazione e di convergenza sono dovuti al metodo numerico applicato, il primo, ed alla sua modalità di utilizzo, il secondo tipo di errore.

1.3 Errori di round-off

⁶ Nella Sezione sono trattati gli errori di rappresentazione ed è tralasciata l'analisi della loro propagazione durante il calcolo. Gli errori di round-off sono dovuti all'impiego di un calcolatore, quindi con aritmetica finita, per ottenere un risultato. Questi errori sono dovuti alla rappresentazione finita di una quantità numerica, la quale richiede infinite informazioni per essere rappresentate esattamente (ad esempio i numeri irrazionali come π).

Saranno trattati i due casi seguenti:

- numeri interi,
- numeri reali.

In entrambi i casi è utilizzata una notazione posizionale che utilizza potenze di base $b \in \mathbb{N}$ ($b \geq 2$).

⁵Slide 4 PDF lez1, PG 5.

⁶Slide 6 PDF lez1, PG 6.

1.3.1 Numeri interi

⁷ Un numero intero è memorizzato come stringa del tipo

$$\alpha_0 \alpha_1 \cdots \alpha_N, N \in \mathbb{N} \text{ numero di cifre}, \quad (1.5)$$

in cui, assegnata la base $b \in \mathbb{N}$, $b \geq 2$, $\alpha_0 \in \{+, -\}$, $\alpha_i \in \{0, 1, \dots, b-1\}_{i=1, \dots, N}$.

È possibile rappresentare la stringa (1.5) come segue:

$$n = \begin{cases} \sum_{i=1}^N \alpha_i b^{N-i}, & \text{se } \alpha_0 = + \\ \sum_{i=1}^N \alpha_i b^{N-i} - b^N, & \text{se } \alpha_0 = - \end{cases} \quad (1.6)$$

Tramite la notazione (1.5)-(1.6) è possibile rappresentare **senza errore** tutti i numeri interi dell'insieme

$$\{-b^N, b^N - 1\}. \quad (1.7)$$

Dettagli aggiuntivi:

- Per motivi di efficienza è assunto, anche in seguito, che le basi utilizzate siano pari. In particolare, nel caso della base binaria, i numeri negativi sono efficientemente rappresentati utilizzando il complemento 2;
- Il caso in cui è necessario rappresentare un numero all'esterno dell'insieme (1.7) crea una condizione di errore non facilmente diagnosticabile.

Esempio 1.2. $b = 10$, $N = 4$, $\alpha_0 \alpha_1 \alpha_2 \alpha_3 \alpha_4 = n = 1 \cdot 10^3 + 7 \cdot 10^2 + 4 \cdot 10^1 + 2 \cdot 10^0$.
 $\quad \quad \quad + \quad 1 \quad 7 \quad 4 \quad 2$

1.3.2 Numeri reali

⁸ Un numero "reale" è memorizzato in un calcolatore mediante una stringa del tipo

$$\alpha_0 \cdots \alpha_m \beta_1 \cdots \beta_s, \quad (1.8)$$

in cui $b \in \mathbb{N}$, $b \geq 2$, $\alpha_0 \in \{+, -\}$, $\alpha_i, \beta_j \in \{0, \dots, b-1\}_{i=1, \dots, m, j=1, \dots, s}$.

Definizione 1.1 (Notazione scientifica normalizzata). La stringa (1.8), sotto la condizione $\alpha_1 \neq 0$, rappresentabile come la notazione scientifica normalizzata in base b nella seguente forma

$$r = \pm \frac{\left(\sum_{i=1}^m \alpha_i b^{1-i} \right)}{\alpha_0} b^{e-\nu}, \quad e = \sum_{j=1}^s \beta_j b^{s-j} \in \mathbb{N}, \quad (1.9)$$

⁹ con fissati $\nu \in \mathbb{N}$ shift, m il numero di cifre della mantissa ed s il numero di cifre dell'esponente. Le quantità

$$\rho = \sum_{i=1}^m \alpha_i b^{1-i}, \quad \eta = e - \nu, \quad (1.10)$$

sono rispettivamente **mantissa** ed **esponente** del numero reale.

⁷Slide 6 PDF lez1, PG 7.

⁸Slide 8 PDF lez1, PG 7.

Esempio 1.3.¹⁰ Dati $b = 10$, $m = 3$, $s = 2$, $\nu = 0$, allora è possibile rappresentare il seguente numero:

$$\alpha_0 \alpha_1 \alpha_2 \alpha_3 \beta_1 \beta_2 = r = + \left(\sum_{i=1}^3 \alpha_i 10^{1-i} \right) 10^e = +(3 \cdot 10^0 + 4 \cdot 10^{-1} + 7 \cdot 10^{-2}) \cdot 10^{1 \times 10^{2-1} + 1 \times 10^{2-2}} = +3.47 \cdot 10^{11},$$

dove $e = \sum_{j=1}^2 \beta_j 10^{2-j} = 10 + 1 = 11$.

Esempio 1.4.¹¹ Supposto l'utilizzo della notazione denormalizzata,

$$\alpha_1 = 0, b = 10, m = 5, s = 5, \nu = 2, x = 1.543 \cdot 10^\eta,$$

con $\eta = e - \nu = e - 2 = 1$, quindi $e = 3$. Inoltre, la rappresentazione di x non è unica, è possibile la seguente: $x = 0.1543 \cdot 10^2$.

Definizione 1.2.¹² L'insieme di numeri di macchina, o floating-point, è definito come:

$$M = \{0\} \cup \{\text{Numeri rappresentabili come (1.9) (fissati } m, s, \nu, b\text{)}\}. \quad (1.11)$$

Teorema 1.1.¹⁴ M ha un numero **finito** di elementi. (Ovvero: M è un insieme discreto.)

Teorema 1.2.¹⁵ $1 \leq |\rho| = \sum_{i=1}^m \alpha_i b^{1-i} < b$

Dimostrazione. Dalle (1.8)-(1.10) segue che ($|\rho| \geq 1$)

$$|\rho| = \underbrace{\alpha_1}_{\substack{\text{mantissa} \\ 0}} \underbrace{\alpha_2 \cdots \alpha_m}_{\substack{\text{17} \\ 16}} \geq \alpha_1 \underbrace{0 \cdots 0}_{m-1} = \alpha_1 \geq 1.$$

Inoltre (è necessario dimostrare che $|\rho| < b$):

$$|\rho| = \alpha_1 \alpha_2 \cdots \alpha_m \leq (b-1) \underbrace{(b-1) \cdots (b-1)}_{m-1} = b \underbrace{(1-b^{-m})}_{<1} < b.$$

□

Esempio 1.5. $b = 10$, $m = 4$, $9.999 = 10 \cdot (1 - 0.0001) = 10 \cdot \underbrace{(1 - 10^{-9})}_{0.9999}.$

⁹Utile per rappresentare i numeri minori di 1.

¹⁰Slide 10 PDF lez1.

¹¹Slide 1 PDF 2.

¹²Slide 2 PDF lez2, Definizione 1.1 PG 8.

¹³Aggiunto perché nella notazione scientifica normalizzata non è rappresentabile.

¹⁴Slide 2 PDF lez2, Teorema 1.1 PG 8.

¹⁵Slide 2 PDF lez2, Teorema 1.2 PG 8.

¹⁶Per la definizione di notazione scientifica normalizzata.

¹⁷Ogni $\alpha_i \in \{0, b-1\}$ ed ipotizzando che siano tutti 0 allora accade ciò che segue.

¹⁸Dimostrando $|\rho| \geq 1$.

¹⁹ $\forall i \in \{1, \dots, m\}$, $1 \leq \alpha_i \leq b \rightarrow$ sostituzione degli α_i con $b-1$.

Teorema 1.3.²⁰ Il più piccolo ed il più grande numero macchina positivi e diversi da 0 sono, rispettivamente:

$$\begin{aligned} r_1 &= b^{-\nu}, \\ r_2 &= (1 - b^{-m})b^\varphi, \quad \varphi = b^s - \nu \end{aligned} \quad (1.12)$$

con b^s massimo numero ottenibile con la notazione (1.8).

Osservazione 1.1.²¹ Lo shift è scelto in modo che $r_1 \approx r_2^{-1}$, ovvero $\nu \approx b^{\frac{s}{2}}$.

Definizione 1.3 (Reali rappresentabili). Dato il Teorema 1.3 e sia

$$I = [-r_2, -r_1] \cup \{0\} \cup [r_1, r_2], \quad (1.13)$$

dove $[-r_2, -r_1]$ e $[r_1, r_2]$ sono insiemi infiniti. I è l'insieme dei numeri reali rappresentabili con s, m, ν fissati.

Definizione 1.4 (Insieme numeri macchina). $M \subseteq I$ è l'insieme dei numeri di macchina ed è un sottoinsieme finito di I .

Definizione 1.5 (Funzione floating). La funzione floating è definita come

$$\begin{aligned} fl &: I \rightarrow M \\ x &\mapsto fl(x) \end{aligned}$$

dove, in generale, $fl(x) \neq x$.

La funzione floating fl trasforma un numero reale in numero macchina (con l'aggiunta di un errore, dato che $fl(x) \neq x$). Un numero per il quale non è introdotto un errore nella rappresentazione è 0 ($fl(0) = 0$). Quindi, i numeri tra $[-r_2, -r_1]$ e $[r_1, r_2]$ possono essere rappresentati con la funzione fl , la quale associa ad ogni $i \in I$ una rappresentazione macchina.

L'utilizzo della funzione floating introduce errori di rappresentazione del tipo round-off, data la rappresentazione non precisa (come specificato nella definizione $fl(x) \neq x$).

Dato $x \in I$, ovvero un numero esatto, definito come (1.9), della forma

$$x = (\alpha_0 \alpha_1 \alpha_2 \cdots \alpha_m \alpha_{m+1} \cdots) b^{e-\nu} \in I, \quad (1.14)$$

esistono due possibili implementazioni della funzione $fl(x) = (\alpha_0 \underbrace{\alpha_1 \alpha_2 \cdots \alpha_{m-1}}_{m-1} \tilde{\alpha}_m) b^{e-\nu}$ per approssimare $\tilde{\alpha}_m$:

- rappresentazione con troncamento,

$$\tilde{\alpha}_m = \alpha_m. \quad (1.15)$$

Quindi vengono ignorate le cifre successive alla m -esima.

- rappresentazione con arrotondamento,

$$\tilde{\alpha}_m = \begin{cases} \alpha_m, & \text{se } \alpha_{m+1} < \frac{b}{2} \\ \alpha_m + 1, & \text{se } \alpha_{m+1} \geq \frac{b}{2} \end{cases} \quad (1.16)$$

Nel caso in cui $\tilde{\alpha}_m \geq b$ allora ci sarà un riporto delle cifre precedenti alla m -esima.

²⁰Slide 3 PDF lez2, Teorema 1.3 PG 8.

²¹Slide 4 PDF lez2, Osservazione 1.2 PG 9.

Principalmente sarà usata la rappresentazione con arrotondamento, utilizzando alcune sofisticazioni.

Esempio 1.6. ²² $b = 10, m = 2, s = 2$

	TRONCAMENTO	ARROTONDAMENTO
$x = 3.14$	3.1	$3.1 \quad 4 < \frac{10}{2} = 5$
$x = 3.18$	3.1	$3.2 \quad 8 \geq \frac{10}{2} = 5$

Il seguente Teorema è importante, afferma che il massimo errore relativo da poter commettere (con base, mantissa, ecc. fissati) è u ed è possibile esprimere con questo ultimo valore la precisione della rappresentazione. La precisione dipende fortemente dal numero di cifre rappresentate (maggiori è il numero di cifre maggiore è la precisione).

Teorema 1.4. ²³ Se $x \in I, x \neq 0 \Rightarrow fl(x) = x(1 + \varepsilon_x)$, con $|\varepsilon_x| \leq u$, dove

$$u = \begin{cases} b^{1-m}, & \text{in caso di troncamento;} \\ \frac{1}{2}b^{1-m}, & \text{in caso di arrotondamento.} \end{cases} \quad (1.17)$$

Dimostrazione. È riportato il caso del troncamento, simili argomenti sono applicati all'arrotondamento.

$x = (\alpha_0 \alpha_1 \dots \alpha_m \dots) b^{e-\nu}$ ed $fl(x)$ sono dati, da (1.14) e (1.15) con la normalizzazione $\alpha_1 \neq 0$, allora:

$$\begin{aligned} |\varepsilon_x| &= \frac{|fl(x) - x|}{x} & \stackrel{24}{=} & \frac{|(\alpha_1 \alpha_2 \dots \alpha_m) b^{e-\nu} - (\alpha_1 \alpha_2 \dots \alpha_m \dots) b^{e-\nu}|}{(\alpha_1 \alpha_2 \dots \alpha_m \dots) b^{e-\nu}} & = & \frac{|(\alpha_1 \alpha_2 \dots \alpha_m) - (\alpha_1 \alpha_2 \dots \alpha_m \dots)| b^{e-\nu}}{(\alpha_1 \alpha_2 \dots \alpha_m \dots) b^{e-\nu}} \\ &\stackrel{25}{=} \frac{0.00 \dots \alpha_{m+1} \dots}{\alpha_1 \alpha_2 \dots \alpha_{m+1} \dots} & \leq & 0.00 \dots \alpha_{m+1} \dots & \stackrel{26}{=} & (\alpha_{m+1} \alpha_{m+2} \dots) b^{-m} \\ &\leq & ((b-1)(b-1) \dots) b^{-m} & < & b b^{-m} & = b^{1-m} \end{aligned}$$

□

Definizione 1.6 (Precisione macchina). u è detto **precisione macchina**.

u è il più piccolo reale, diverso da 0, in notazione scientifica che sommato ad n numero qualsiasi dà un numero diverso da n . Infatti, qualsiasi numero più piccolo di u è considerato 0 (?).

Definizione 1.7 (Errore relativo di rappresentazione). $\varepsilon_x = \frac{|fl(x) - x|}{x}$ è l'**errore relativo di rappresentazione** relativo ad x .

Per rappresentare un elemento di I (vedere (1.13)), attraverso la funzione floating point, è commesso, al più, un errore relativo ε_x maggiorato dalla precisione macchina u , quindi u è un limite per l'errore.

²²Slide 6 PDF lez3.

²³Slide 6 PDF lez3, Teorema 1.4 PG 10.

²⁴È ignorato α_0 perché, dato il valore assoluto, il segno è ininfluente.

²⁵Al numeratore è presente il risultato della sottrazione precedente passaggio. Al denominatore è presente un termine sicuramente diverso da 0, se $\alpha_1 \neq 0$, quindi il denominatore $(\frac{1}{\alpha_1 \dots \alpha_m \dots})$ può essere maggiorato da 1, permettendo una successiva maggiorazione.

²⁶Spostamento degli $\alpha_{m+1} \dots$ a sinistra della virgola.

²⁷Derivato da $fl(x) = x(1 + \varepsilon_x)$ del Teorema 1.4.

1.3.3 Overflow e Underflow

²⁸ Può capitare di dover rappresentare numeri reali non contenuti in I , in questo caso è sollevata una condizione d'errore. Le condizioni d'errore derivate dalla rappresentazione di un reale $x \notin I$, con I definito come in (1.13), sono:

- $|x| > r_2 \Rightarrow$ overflow. Tale condizione è rappresentata in base al sistema di calcolo utilizzato (esempio: nello standard IEEE 754 il simbolo **Inf** rappresenta ∞);
- $0 < |x| < r_1 \Rightarrow$ underflow. In questo caso esistono due tipi di recovery, indipendenti ed associate al sistema di calcolo utilizzato (ed è possibile utilizzare solo una delle due):
 - $fl(x) = 0$;
 - gradual underflow: non è più richiesto $\alpha_1 \neq 0$, quindi è utilizzata la notazione scientifica denormalizzata per rappresentare i numeri di macchina M , invalidando il Teorema 1.4.

Esempio 1.7. $b = 10$, $\nu = 30$, $m = 4 \Rightarrow r_1 = 10^{-30}$. $x = 10^{-32}$ può essere espresso in aritmetica finita come $x = 0.0010 \times 10^{-30}$ (gradual underflow).

L'implementazione della funzione fl può essere riassunta come segue:

$$fl(x) = \begin{cases} 0, & \text{se } x = 0; \\ \tilde{x} \equiv x(1 + \varepsilon_x), |\varepsilon_x| \leq u & \text{se } r_1 \leq |x| \leq r_2 \\ \text{underflow}, & \text{se } 0 < |x| < r_1 \\ \text{overflow}, & \text{se } |x| > r_2 \end{cases}$$

Osservazione 1.2. Può essere utile vedere *eps*, *realmin* e *realmax* di Matlab:

- $eps = 2.2204e - 16$, rappresenta la distanza fra 1 ed il numero in doppia precisione più vicino ad 1. In Matlab $1 - eps = 9.99999999999998e - 01$, mentre $1 - 1e - 17 = 1$;
- $realmin = 2.2251e - 308$ è il più piccolo numero finito rappresentabile con la funzione di floating point nello standard IEEE in doppia precisione;
- $realmax = 1.7977e + 308$ è il più grande numero finito rappresentabile con la funzione di floating point in nello standard IEEE in doppia precisione.

1.3.4 Standard IEEE 754

²⁹ Nella Sezione è trattato lo standard ANSI/IEEE 754-1985, per la rappresentazione di numeri reali sugli elaboratori. Questo standard è stato definito per poter garantire che programmi identici, eseguiti su piattaforme di calcolo differenti, producano gli stessi risultati ed è adottato dalla maggior parte dei calcolatori esistenti.

Lo standard IEEE 754 è particolarmente adatto alla base binaria ($b = 2$).

La rappresentazione dei numeri reali è effettuata tramite la funzione di floating per arrotondamento, caratteristica che permette di commettere un errore minore che per troncamento. Inoltre, fl è round to even, ovvero: dato il numero macchina $fl(x)$ che più si avvicina ad x , allora, in caso di ambiguità, se vi fossero due numeri macchina equidistanti da x , è selezionato quello il cui ultimo bit della mantissa è pari (ossia uguale a 0). Nonostante quest'ultima proprietà della funzione di floating, il Teorema 1.4 rimane valido.

I formati di base per i dati trattati e previsti dallo standard sono:

²⁸Slide 8 PDF lez2, PG 11.

²⁹Slide 1 PDF lez3, PG 11.

- singola precisione (32 bit),
- doppia precisione (64 bit).

Per entrambi i formati la mantissa è assunta in base 2 e vale quanto segue:

- $\rho = 1.f$ nel caso della notazione scientifica normalizzata ($\alpha_1 \neq 0$);
- $\rho = 0.f$ nel caso della notazione scientifica denormalizzata ($\alpha_1 = 0$, quindi lo standard implementa il gradual-underflow).

Per entrambe le notazioni non sarà memorizzata la prima cifra, sarà memorizzata solo la frazione f , con un evidente risparmio di 1 bit. Il tipo di notazione sarà dedotto dalle convenzioni fra poco definite.

Singola precisione. ³⁰ La lunghezza della mantissa ρ è suddivisa fra 32 bit $\begin{cases} 1, & \text{segno;} \\ 23, & \text{frazione;} \\ 8, & \text{esponente;} \end{cases} \Rightarrow m = 24, s = 8.$ ³¹

Inoltre, è possibile configurare shift, mantissa ed esponente (ovvero $e \in \mathbb{N}$) come segue:

- se $0 < e < 255 = 2^8 - 1$, la notazione è normalizzata, $\nu = 127 = 2^7 - 1$;
- se $e = 0$, $f \neq 0$, la notazione è denormalizzata, $\nu = 126$ (caso gradual-underflow);
- se $e = 0$, $f = 0 \rightarrow 0$ con eventuale segno (non rappresentabile con notazione scientifica normalizzata);
- se $e = 255$, $\alpha_0 = 0$, $f = 0 \rightarrow +Inf$ (rappresenta l'overflow);
- se $e = 255$, $\alpha_0 = 1$, $f = 0 \rightarrow -Inf$ (rappresenta l'underflow);
- se $e = 255$, $f \neq 0 \rightarrow NaN$ (Not a Number, esempio divisione per 0, $0/0$, $Inf - Inf$, $0 \times Inf$).

Nei primi due punti è inclusa l'informazione riguardo il tipo di notazione (de/normalizzata).

Doppia precisione. ³² La lunghezza della mantissa ρ è suddivisa fra 64 bit $\begin{cases} 1, & \text{segno;} \\ 52, & \text{frazione;} \\ 11, & \text{esponente;} \end{cases} \Rightarrow m = 53, s = 11.$

Inoltre, è possibile configurare shift, mantissa ed esponente come segue:

- se $0 < e < 2047 = 2^{11} - 1$, la notazione è normalizzata, $\nu = 1023 = 2^{10} - 1$;
- se $e = 0$, $f \neq 0$, la notazione è denormalizzata, $\nu = 1022$ (caso gradual-underflow);
- se $e = 0$, $f = 0 \rightarrow 0$ con eventuale segno;
- se $e = 2047$, $\alpha_0 = 0$, $f = 0 \rightarrow +Inf$;
- se $e = 2047$, $\alpha_0 = 1$, $f = 0 \rightarrow -Inf$;
- se $e = 2047$, $f \neq 0 \rightarrow NaN$.

In questo caso la **precisione macchina** può essere approssimata come $u \approx 10^{-16} \approx 2^{-53}$ (vedere il Teorema 1.4) e le stesse valutazioni fatte per i primi due punti della singola precisione sono fatte per la doppia.

³⁰Slide 2 PDF lez3, PG 12.

³¹Lunghezza della mantissa.

³²Slide 3 PDF lez3, PG 13.

1.3.5 Aritmetica Finita

³³ È interessante capire cosa accade in aritmetica finita quando sono svolte operazioni su numeri macchina. Date le implementazioni delle operazioni algebriche (+, −, ×, /) in aritmetica finita, è necessario distinguere tra numeri reali ed interi.

Numeri interi: Se operandi e risultato appartengono all'insieme dei numeri di macchina (1.7), le operazioni coincidono con le corrispondenti algebriche.

Numeri reali: In questo caso le operazioni ed il risultato sono, rispettivamente, definite tra numeri macchina e numero macchina.

Un esempio di implementazione della somma algebrica in aritmetica finita, sia \oplus , risulta essere il seguente:

$$x \oplus y = \text{fl}(\text{fl}(x) + \text{fl}(y)), \quad x, y \in \mathbb{R}.$$

Osservazione 1.3. ³⁵ Non sempre le proprietà algebriche delle operazioni coincidono con quelle in aritmetica finita. Generalmente proprietà distributiva, associativa, ecc, non valgono.

Esempio 1.8. Dato r_2 definito come in (1.12) e 2 (numero intero), allora:

$$\text{fl}(r_2) = r_2, \quad f(2) = 2, \quad (r_2 - r_2) \times 2 = r_2 \times 2 - r_2 \times 2 = 0.$$

$$(r_2 \ominus r_2) \otimes 2 \stackrel{?}{=} \underset{36}{r_2 \otimes 2} \ominus \underset{36}{r_2 \otimes 2}$$

$$(r_2 \ominus r_2) \otimes 2 = \text{fl}(\text{fl}(\underbrace{r_2 - r_2}_0 \cdot 2)) = \text{fl}(0 \cdot 2) = \text{fl}(0) = 0 \neq \text{NaN} \stackrel{37}{=} \text{fl}(\text{fl}(r_2 \cdot 2) - \text{fl}(r_2 \cdot 2)) = r_2 \otimes 2 \ominus r_2 \otimes 2.$$

Esempio 1.9. Dato $r_2 = \underbrace{9.999\cdots 9}_{m} \cdot 10^{10^s-1-\nu}$, ovvero definito come in (1.12) ed $1 = 1.000\cdots 0 \cdot 10^0 = \underbrace{0.000\cdots 0}_{m} \cdot 1 \cdot 10^{10^s-1-\nu}$, allora: ³⁸

$$\begin{aligned} r_2 - r_2 + 1 &= (r_2 - r_2) + 1 = r_2 - (r_2 - 1) \\ (r_2 \ominus r_2) \oplus 1 &\stackrel{?}{=} \underset{39}{r_2 \ominus (r_2 \ominus 1)} \end{aligned}$$

$$(r_2 \ominus r_2) \oplus = \text{fl}(\text{fl}(r_2 - r_2) + 1) = \text{fl}(0 + 1) = 1 \neq 0 = \text{fl}(r_2 - r_2) = \text{fl}(r_2 - \text{fl}(r_2 - 1)) = r_2 \ominus (r_2 \ominus 1).$$

³³Slide 4 PDF lez3, PG 13.

³⁴È necessaria la doppia applicazione della funzione di floating fl perché x e y non sono numeri macchina e la somma di numeri macchina non è un numero macchina.

³⁵Slide 5 PDF lez3, Osservazione 1.7 PG 13

³⁶Non c'è la distributività.

³⁷L'overflow non man tiene l'ordine, quindi l'applicazione di operazioni sull'infinito non è permesso ed è ottenuto NaN come risultato perché $+Inf = r_2 \cdot 2 > r_2$.

³⁸Quando è applicata fl , a meno di scegliere s molto piccolo l'1 è molto dopo gli m 0.

³⁹Non c'è l'associatività.

⁴⁰ $fl(r_2 - 1) = r_2$ perché $2^{64}, 2^{52}$ o qualsivoglia sia il numero da rappresentare, è così grande che non cambia quando viene rappresentato, se gli è sottratto 1.

1.4 Condizionamento di un problema

⁴¹ Con "problema" non è fatto riferimento al metodo di calcolo, ma ad un generico problema matematico schematizzabile con

$$y = f(x), \quad (1.18)$$

dove:

- $x \in \mathbb{R}$, sono i dati in ingresso;
- $f : \mathbb{R} \rightarrow \mathbb{R}$, $f \in C^2$, la descrizione formale del problema;
- $y \in \mathbb{R}$, denota la soluzione al problema.

Un metodo numerico che risolve il problema (1.18), ovvero lo approssima, può essere formalizzato come

$$\tilde{y} = \tilde{f}(\tilde{x}), \quad (1.19)$$

in cui:

- $\tilde{x} \in \mathbb{R}$ denota i dati in ingresso affetti da errore, ovvero perturbati (questi sono sempre affetti da errore di rappresentazione);
- \tilde{f} denota il metodo numerico implementato utilizzando aritmetica finita, introducendo così possibili errori discretizzazione e/o convergenza;
- $\tilde{y} \in \mathbb{R}$ denota la soluzione, il dato in uscita, affetta da errore (ovvero sarà perturbata).

È interessante studiare l'errore ($y - \tilde{y}$) del metodo numerico (1.19), ovvero la sua misura, in funzione degli errori sui dati in ingresso, $\tilde{x} - x$, oltre alla misura dell'errore relativo. Studiare l'errore sui dati iniziali $\tilde{x} - x$ e l'effettiva implementazione del metodo numerico è spesso complesso (non rientra nel corso).

Sarà studiato, indipendentemente dal metodo, il condizionamento del problema (1.18), ovvero quella parte dell'errore che è incluso nel risultato e che dipende solo dalla natura del problema e dai dati iniziali. Tale studio è una analisi semplificata di quanto citato e permette di stabilire l'attendibilità del risultato del metodo numerico. In dettaglio, considerata l'amplificazione sul risultato finale di perturbazioni sui dati di ingresso, è supposto di risolvere il problema esattamente (quindi in aritmetica esatta).

Formalmente è studiato

$$\tilde{y} = f(\tilde{x}), \quad (1.20)$$

quindi il condizionamento del problema è lo studio della relazione fra l'errore sui dati iniziali ($x - \tilde{x}$) e l'errore sulla soluzione ($y - \tilde{y}$). Ciò è interessante per evitare che un piccolo errore sui dati iniziali porti ad un errore grande nella soluzione e per questo il problema è detto:

- ben condizionato, se l'errore sui dati iniziali è "poco" amplificato;
- malcondizionato, se l'errore sui dati iniziali è molto amplificato.

Data $y = f(x)$, $y \in C^2$ e gli errori relativi

$$\tilde{x} = x(1 + \varepsilon_x), \quad \tilde{y} = y(1 + \varepsilon_y),$$

⁴¹Slide 7-10 PDF lez3 + lez4, PG 14-18.

allora è possibile sviluppare (1.20), tramite lo sviluppo di Taylor al secondo ordine centrato in x (dati esatti), come segue:

$$\tilde{y} = f(\tilde{x}) = f(x) + f'(x)(\tilde{x} + x\varepsilon_x - \tilde{x}) + \frac{f''(\xi)}{2}(\tilde{x} + x\varepsilon_x - \tilde{x})^2 \Rightarrow y(1 + \varepsilon_y) = \tilde{y} \stackrel{f(x)=y}{=} y + f'(x)x\varepsilon_x + \underbrace{\frac{f''(\xi)}{2}x^2\varepsilon_x^2}_{O(\varepsilon_x^2)}. \quad (1.21)$$

Tenendo di conto di (1.21), allora

$$y + y\varepsilon_y = y + f'(x)x\varepsilon_x + O(\varepsilon_x^2) \rightarrow y\varepsilon_y = f'(x)x\varepsilon_x + O(\varepsilon_x^2) \Rightarrow y\varepsilon_y \stackrel{42}{\approx} f'(x)x\varepsilon_x \rightarrow |\varepsilon_y| \approx \left| f'(x) \frac{x}{y} \right| \cdot |\varepsilon_x| \equiv \kappa |\varepsilon_x|.$$

Definizione 1.8 (Numero di condizionamento della radice). Il fattore κ misura quanto gli errori iniziali possono amplificarsi sul risultato ed è noto come numero di condizione del problema (1.18).

È possibile distinguere i seguenti casi:

- $\kappa \approx 1 \Rightarrow |\varepsilon_y| \approx |\varepsilon_x| \rightarrow$ il problema è ben condizionato;
- $k >> 1 \Rightarrow |\varepsilon_y| >> |\varepsilon_x| \rightarrow$ il problema è mal condizionato.

È possibile notare ciò che segue:

1. nel caso in cui è utilizzata precisione macchina u e $\kappa \approx u^{-1}$, qualunque risultato sarà privo di significato, in quanto i dati sono affetti da errore di rappresentazione, per cui $|\varepsilon_x| \approx u$. Inoltre, se $|\varepsilon_y| \approx u^{-1}u = 1$, è presente una completa perdita di informazione ed il problema risulta malcondizionato, l'errore ε_y è dello stesso ordine del risultato;
2. nel caso di problemi malcondizionati, l'unica possibilità per ottenere un risultato attendibile è quello di riformulare il problema in modo che abbia proprietà di condizionamento favorevoli;
3. nel caso di problemi ben condizionati, occorre scegliere metodi ben condizionati che preservino il buon condizionamento del problema originario (tali metodi sono detti metodi numericamente stabili).

Il limite fra problema ben e malcondizionato non è preciso, è necessario un criterio per distinguere i due casi. È possibile affermare che se il numero di condizionamento è di ordine unità ($\kappa \in [1, 10]$) allora è ben condizionato, in altri casi il problema rimane ben condizionato anche se il numero di condizionamento è dell'ordine delle decine (ovvero quando l'errore iniziale viene amplificato di ordine 10 sul risultato).

Osservazione qualitativa: Valutare quantitativamente il numero di condizionamento è importante ed è necessario osservare come κ varia all'aumentare della dimensione del problema (se è costante entro certi limiti è buono).

⁴² $O(\varepsilon_x^2)$ è di ordine 2 quindi è trascurabile rispetto a $f'(x)\varepsilon_x$, allora è possibile l'approssimazione.

⁴³Gli errori sono dello stesso ordine di grandezza.

Condizionamento della somma algebrica: Il problema è studiare il condizionamento di

$$y = x_1 + x_2 \neq 0, \quad x_1, x_2 \in \mathbb{R}, \quad (1.22)$$

dove x_1, x_2 sono gli addendi esatti ed y la soluzione esatta.

Dati gli errori relativi sui dati iniziali $\varepsilon_1 = \frac{\tilde{x}_1 - x_1}{x_1}$ e $\varepsilon_2 = \frac{\tilde{x}_2 - x_2}{x_2}$, con $\tilde{x}_1 = x_1(1 + \varepsilon_1)$ e $\tilde{x}_2 = x_2(1 + \varepsilon_2)$ addendi perturbati, assumendo che non venga introdotto alcun nuovo errore nel calcolo di (1.22), allora, con

$$\tilde{y} = \tilde{x}_1 + \tilde{x}_2 = y(1 + \varepsilon_y) \quad \text{e} \quad \varepsilon_y = \frac{\tilde{x}_1 + \tilde{x}_2 - (x_1 + x_2)}{x_1 + x_2},$$

è ottenuto quanto segue:

$$\begin{aligned} y(1 + \varepsilon_y) &= \tilde{y} &= \tilde{x}_1 + \tilde{x}_2 &= x_1(1 + \varepsilon_1) + x_2(1 + \varepsilon_2) &= x_1 + x_1\varepsilon_1 + x_2 + x_2\varepsilon_2 \\ \rightarrow y + y\varepsilon_y &= \underbrace{x_1 + x_2}_{y} + x_1\varepsilon_1 + x_2\varepsilon_2 \\ \rightarrow y + y\varepsilon_y &= \tilde{y} + x_1\varepsilon_1 + x_2\varepsilon_2 \\ \rightarrow y\varepsilon_y &= x_1\varepsilon_1 + x_2\varepsilon_2 \\ \rightarrow \varepsilon_y &= \frac{x_1\varepsilon_1 + x_2\varepsilon_2}{x_1 + x_2}. \end{aligned}$$

Considerando (1.22), è ottenuto quanto segue:

$$\begin{aligned} |\varepsilon_y| &= \frac{|x_1\varepsilon_1 + x_2\varepsilon_2|}{|x_1 + x_2|} \leq \frac{|x_1\varepsilon_1| + |x_2\varepsilon_2|}{\underbrace{|x_1| + |x_2|}_{k}} = \frac{|x_1||\varepsilon_1| + |x_2||\varepsilon_2|}{|x_1 + x_2|} \\ &\leq \frac{|x_1|\varepsilon_x + |x_2|\varepsilon_x}{|x_1| + |x_2|} = \underbrace{\frac{|x_1| + |x_2|}{|x_1 + x_2|}}_k \varepsilon_x \equiv \kappa \varepsilon_x \end{aligned}$$

allora,

$$|\varepsilon_y| \leq \frac{|x_1| + |x_2|}{|x_1 + x_2|} \varepsilon_x, \quad \varepsilon_x = \max \{|\varepsilon_1|, |\varepsilon_2|\}$$

Pertanto, il numero di condizionamento della somma algebrica è esprimibile come

$$\kappa = \frac{|x_1| + |x_2|}{|x_1 + x_2|}.$$

Seguono due casi significativi:

- se $x_1 x_2 > 0 \Rightarrow |x_1| + |x_2| = |x_1 + x_2| \Rightarrow k = 1$, quindi è concluso che la somma di numeri concordi di segno è sempre ben condizionata;
- se $x_1 \approx -x_2 \Rightarrow \frac{|x_1 + x_2|}{|x_1| + |x_2|} \approx \frac{1}{2|x_1|} \Rightarrow k \gg 1$, quindi il problema è malcondizionato quando è svolta la somma di due numeri quasi opposti. In aritmetica finita questo malcondizionamento porta al fenomeno della **cancellazione numerica**.

⁴⁴ Errore relativo sui dati iniziali.

Esempio 1.10.⁴⁵ Dati $x_1 = 1.000$, $\tilde{x}_1 = 1.000$, $y = -0.001 = -10^{-3}$, $x_2 = -1.000$, $\tilde{x}_2 = -0.999$, $\tilde{y} = 0.001 = 10^{-3}$, allora,

$$\begin{aligned}\kappa &= \frac{|x_1| + |x_2|}{|x_1 + x_2|} = \frac{1+1.001}{10^{-3}} = \frac{2.001}{10^{-3}} = \mathbf{2.001 \cdot 10^3} \\ |\varepsilon_1| &= \frac{|\tilde{x}_1 - x_1|}{|x_1|} = \frac{0}{1} = 0, \quad |\varepsilon_2| = \frac{|\tilde{x}_2 - x_2|}{|x_2|} = \frac{|-0.999 + 1.001|}{1.001} = \frac{2 \cdot 10^{-3}}{1.001} \approx \mathbf{2 \cdot 10^{-3}} \\ \varepsilon_x &= \max\{0, 2 \cdot 10^{-3}\} = \mathbf{2 \cdot 10^{-3}} \quad |\varepsilon_y| = \frac{|10^{-3} + 10^{-3}|}{|10^{-3}|} = \frac{2 \cdot 10^{-3}}{10^{-3}} = \mathbf{2}\end{aligned}$$

Condizionamento della moltiplicazione:⁴⁶ È esaminato il condizionamento del problema

$$y = x_1 x_2 \neq 0, \quad x_1, x_2 \in \mathbb{R}, \quad (1.23)$$

con x_1 e x_2 dati esatti ed y soluzione esatta.

Siano $\tilde{x}_1 = x_1(1 + \varepsilon_1)$ e $\tilde{x}_2 = x_2(1 + \varepsilon_2)$ i dati perturbati, $\tilde{y} = \tilde{x}_1 \tilde{x}_2 = y(1 + \varepsilon_y)$ il risultato perturbato, ε_1 e ε_2 gli errori relativi sui dati ed e_y l'errore sul risultato. È ottenuto quanto segue:

$$\begin{aligned}y(1 + \varepsilon_y) &= \tilde{y} &= \tilde{x}_1 \tilde{x}_2 &= x_1(1 + \varepsilon_1)x_2(1 + \varepsilon_2) \\ &= (x_1 + x_1 \varepsilon_1)(x_2 + x_2 \varepsilon_2) &= x_1 x_2 + x_1 x_2 \varepsilon_2 + x_1 x_2 \varepsilon_1 + x_1 x_2 \varepsilon_1 \varepsilon_2 &= x_1 x_2 (1 + \varepsilon_2 + \varepsilon_1 + \varepsilon_1 \varepsilon_2) \\ &\stackrel{47}{\approx} x_1 x_2 (1 + \varepsilon_1 + \varepsilon_2) &= x_1 x_2 + x_1 x_2 (\varepsilon_1 + \varepsilon_2).\end{aligned}$$

Pertanto, è ottenuto ciò che segue:

$$\begin{aligned}y(1 + \varepsilon_y) &\approx x_1 x_2 + x_1 x_2 (\varepsilon_1 + \varepsilon_2) \stackrel{x_1 x_2 = y}{=} y + y(\varepsilon_1 + \varepsilon_2) \\ &\Rightarrow \frac{y + y(\varepsilon_1 + \varepsilon_2)}{y} \approx \frac{y + y(\varepsilon_1 + \varepsilon_2)}{y} \\ &\Rightarrow \frac{|\varepsilon_y|}{|\varepsilon_y|} \approx \frac{|\varepsilon_1 + \varepsilon_2|}{|\varepsilon_1 + \varepsilon_2|} \leq \frac{|\varepsilon_1| + |\varepsilon_2|}{|\varepsilon_1 + \varepsilon_2|} \leq \frac{|\varepsilon_1| + |\varepsilon_2|}{|\varepsilon_1| + |\varepsilon_2|} = 2\varepsilon_x.\end{aligned}$$

Quindi,

$$|\varepsilon_y| \leq 2\varepsilon_x, \quad \varepsilon_x = \max\{|\varepsilon_1|, |\varepsilon_2|\},$$

il numero di condizionamento è $\kappa = 2$, quindi la moltiplicazione è sempre ben condizionata.

Condizionamento della divisione:⁴⁸ È esaminato il condizionamento del problema

$$y = \frac{x_1}{x_2}, \quad x_1, x_2 \in \mathbb{R}, \quad x_1 x_2 \neq 0 \quad (1.24)$$

con x_1 e x_2 dati esatti ed y soluzione esatta.

Siano $\tilde{x}_1 = x_1(1 + \varepsilon_1)$ e $\tilde{x}_2 = x_2(1 + \varepsilon_2)$ i dati perturbati, $\tilde{y} = y(1 + \varepsilon_y) = \frac{\tilde{x}_1}{\tilde{x}_2}$ il risultato perturbato. Considerando lo sviluppo di Taylor centrato in 0, è possibile la seguente approssimazione:

$$f(x) = \frac{1}{1+x} \approx f(0) + f'(0)(x-0) = 1-x, \quad f'(x) = -\frac{1}{(1+x)^2} \Rightarrow \varepsilon < 1, \quad f(\varepsilon) = \frac{1}{1+\varepsilon} \approx 1-\varepsilon$$

⁴⁵Slide 6 PDF lez4.

⁴⁶Slide 7 PDF lez4, PG 17.

⁴⁷Dati $\varepsilon_1 < 1$, $\varepsilon_2 < 1 \Rightarrow \varepsilon_1 \varepsilon_2 < \varepsilon_1$, $\varepsilon_1 \varepsilon_2 < \varepsilon_2$, quindi $\varepsilon_1 \varepsilon_2$ è trascurabile rispetto a $\varepsilon_1, \varepsilon_2$.

⁴⁸Slide 9 PDF lez4, PG 18.

dalla quale è ottenuto

$$\begin{aligned}
 y(1 + \varepsilon_y) = \tilde{y} = \frac{\tilde{x}_1}{\tilde{x}_2} &= \frac{x_1(1+\varepsilon_1)}{x_2(1+\varepsilon_2)} = \frac{x_1(1+\varepsilon_1)}{x_2} \cdot \frac{1}{1+\varepsilon_2} \\
 &\stackrel{49}{\approx} \frac{x_1}{x_2}(1 + \varepsilon_1)(1 - \varepsilon_2) = \frac{x_1}{x_2}(1 + \varepsilon_1 - \varepsilon_2 - \varepsilon_1\varepsilon_2) \approx \frac{x_1}{x_2}(1 + \varepsilon_1 - \varepsilon_2) \\
 \Rightarrow y + y\varepsilon_y &= y(1 + \varepsilon_y) \approx y(1 + \varepsilon_1 - \varepsilon_2) \\
 &\Rightarrow \frac{y + y\varepsilon_y}{y} \approx \frac{y(1 + \varepsilon_1 - \varepsilon_2)}{y} \\
 \rightarrow \varepsilon_y &\approx (1 + \varepsilon_1 - \varepsilon_2).
 \end{aligned}$$

Pertanto, è possibile ottenere

$$|\varepsilon_y| \approx |\varepsilon_1 - \varepsilon_2| \leq |\varepsilon_1| + |\varepsilon_2| \leq 2\varepsilon_x, \quad \varepsilon_x = \max\{|\varepsilon_1|, |\varepsilon_2|\},$$

Il numero di condizionamento è $\kappa = 2$, quindi la divisione è sempre ben condizionata.

⁴⁹Come per il condizionamento della moltiplicazione: Dati $\varepsilon_1 < 1$, $\varepsilon_2 < 1 \Rightarrow \varepsilon_1\varepsilon_2 < \varepsilon_1$, $\varepsilon_1\varepsilon_2 < \varepsilon_2$, quindi $\varepsilon_1\varepsilon_2$ è trascurabile rispetto a $\varepsilon_1, \varepsilon_2$.

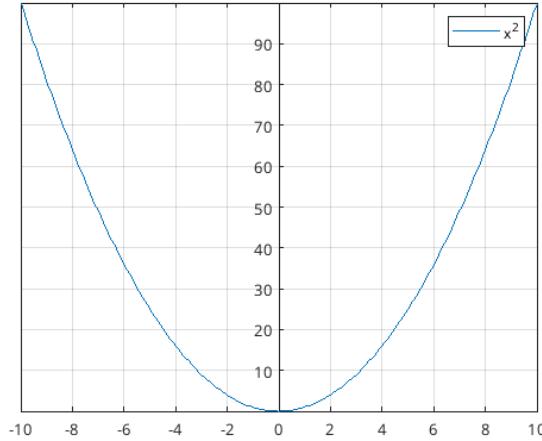


Figura 1: Grafico della radice della funzione x^2

2 Radici di un'equazione (nonlineare)

⁵⁰ Il problema da risolvere è determinare

$$x^* \in \mathbb{R} : f(x^*) = 0, \quad f : \mathbb{R} \rightarrow \mathbb{R}. \quad (2.1)$$

Per risolvere il problema è necessario determinare, se esiste/ono, la/e radice/i x^* della funzione f , distinguendo i casi in base al numero di soluzioni del problema (2.1). Il problema (2.1):

1. ammette un numero finito di soluzioni (esempio: $f(x) = (x - 1)(x^2 - 4) = (x - 1)(x - 2)(x + 2)$);
2. non ha soluzioni reali (esempio: $f(x) = e^x$);
3. ammette infinite soluzioni reali (esempio: $f(x) = \sin(x)$, la quale si annulla in $\pi, 2\pi, \dots, k\pi, \dots, k \in \mathbb{N}$).

Affinché il problema (2.1) sia risolvibile è necessario che ricada negli scenari 1. o 2. appena indicati.

Assunzione importante: in seguito sarà supposto che esista almeno una soluzione reale per (2.1) e che valga il Teorema degli zeri.

Teorema 2.1 (degli zeri). Sia $f : I = [a, b] \rightarrow \mathbb{R}$, $f \in C([a, b])$, $f(a)f(b) < 0 \Rightarrow \exists x^* \in [a, b] : f(x^*) = 0$.

N.B.: È possibile fornire un intervallo di confidenza $[a, b]$ per la radice x^* . La condizione $f(a)f(b) < 0$ può essere verificata all'inizio dell'algoritmo che ricerca le radici.

⁵⁰PG 19-40, PDF lez6.

⁴⁹L'implicazione è dovuta al Teorema degli zeri.

⁵⁰ f assume valori di segno opposto ai due estremi.

2.1 Metodo di bisezione

⁵¹ Data l'assunzione precedente, $[a, b]$ è un intervallo di confidenza per la radice x^* di (2.1) (aumentandone così la precisione). Dovendo scegliere un punto nell'intervallo $[a, b]$, in modo tale che sia la migliore approssimazione della radice, è possibile porre $x^* \approx x_1 = \frac{a+b}{2}$, ovvero selezionare il punto medio dell'intervallo.

Solo uno dei tre seguenti casi può verificarsi:

1. $f(x_1) = 0$, la soluzione è determinata ($x_1 = x^*$);
2. $f(a)f(x_1) < 0$, il procedimento descritto può essere ripetuto sull'intervallo $[a, x_1]$ ($x^* \in [a, x_1]$);
3. $f(x_1)f(b) < 0$, il procedimento descritto può essere ripetuto sull'intervallo $[x_1, b]$ ($x^* \in [x_1, b]$).

Osservazione 2.1. È possibile osservare che nei casi 2. e 3. l'ampiezza dell'intervallo di confidenza si dimezza ad ogni passo perché sostituito, ad ogni passo, l'estremo opportuno.

Osservazione 2.2. Dati $[a_1, b_1] \equiv [a, b]$, $x_1 = \frac{a_1+b_1}{2} \Rightarrow x_n = \frac{a_n+b_n}{2}$, dove x_n è l'approssimazione di x^* . Ulteriori chiarimenti sono forniti nella Sezione 2.2.

È possibile che esistano più radici per la funzione f , un metodo iterativo come quello di bisezione ne approssima solamente una (vedere Figura 2).

Esempio 2.1. Sia $p(x) = (x - 1.1)^{20}(x - \pi)$, in Matlab $p(\pi) \approx 10^{-5}$. È possibile, in altri termini, ottenere il seguente risultato:

```
p = poly([1.1*ones(1,20) pi]);
polyval(p,pi);
ans = -5.521324170132402e-05
```

Implementazione del metodo di bisezione: Vedere Algoritmo 2.1.

2.2 Criterio di Arresto

Quando è possibile fermare la successione di approssimazioni x_n ? Un possibile criterio d'arresto può essere $f(x_n) = 0$. Questa condizione è improbabile che si verifichi, quindi inutilizzabile, perché è possibile arrivare alla radice x^* solo quando $x^* = x_n$, quindi in un numero infinito di passi. Questo non è compatibile con un elaboratore in aritmetica finita, dove $f(x^*) = f(x_n) \neq 0$, a causa dell'aritmetica finita stessa.

Realisticamente è possibile richiedere che

$$|x_n - x^*| \leq \text{tolx}, \quad (2.2)$$

dove tolx è noto, in quanto scelto dall'utente e rappresenta il massimo errore (assoluto) che sarà commesso. Questa diseguaglianza è utile al fine di determinare se esiste un numero finito di iterazioni che permetta un livello di tolleranza dell'errore accettabile (tolx).

2.2.1 Criterio di arresto per il metodo di bisezione

Dato un intervallo di confidenza iniziale $[a, b] \equiv [a_1, b_1]$ e l'approssimazione iniziale $x_1 = \frac{(a_1+b_1)}{2}$, allora, al passo i -esimo, sarà ottenuto l'intervallo di confidenza $[a_i, b_i]$ e l'approssimazione $x_i = \frac{(a_i+b_i)}{2}$. Al passo n -esimo sarà ottenuto l'intervallo di confidenza $[a_n, b_n]$ e l'approssimazione $x_n = \frac{(a_n+b_n)}{2}$.

⁵¹PG. 19-22.

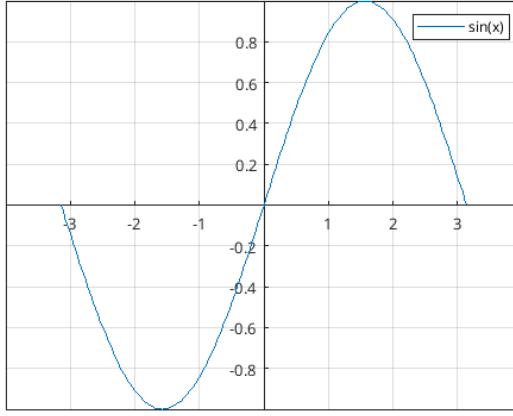


Figura 2: Grafico della funzione con più radici $\sin(x)$

In quanti passi è approssimata la radice x^* ? Per il metodo di bisezione è possibile affermare che

Numero iterazione	Errore	Intervallo di confidenza
1	$ x_1 - x^* \leq \frac{b-a}{2}$	$[a, b]$
2	$ x_2 - x^* \leq \frac{x_1-a}{2} = \frac{b-a}{4}$	$[a_2, b_2]$
\vdots	\vdots	\vdots
i	$ x_i - x^* \leq \frac{b-a}{2^i}$	$[a_i, b_i]$
\vdots	\vdots	\vdots
n	$ x_n - x^* \leq \frac{b-a}{2^n}$	$[a_n, b_n]$

dove $\frac{b-a}{2^i}$ è **stima dell'errore** al passo i -esimo.

L'obiettivo è determinare n in modo tale $|x_n - x^*| \leq tolx$. Fermandosi all'iterazione n allora $|x_n - x^*| \leq \frac{b-a}{2^n} \leq tolx$, è soddisfatto $\frac{b-a}{2^n} \leq tolx$ quindi:

$$\begin{aligned}
 & |x_n - x^*| \leq tolx \\
 \stackrel{52}{\Rightarrow} & \frac{b-a}{2^n} \leq tolx \\
 \Rightarrow & 2^n \geq \frac{b-a}{tolx} \\
 \Rightarrow & n \geq \lceil \log_2 \left(\frac{b-a}{tolx} \right) \rceil \stackrel{53}{=} \lceil \log_2(b-a) - \log_2(tolx) \rceil \equiv itmax.
 \end{aligned}$$

Dalla precedente serie di implicazioni è possibile affermare che la condizione (2.2) si verificherà in, al più, *itmax*.

È possibile soddisfare $|x_n - x^*| \leq tolx$ con meno di *itmax* passi: È possibile, per effettuare meno di *itmax*, approssimare $|x_i - x^*|$ come segue:⁵⁴

$$\begin{aligned} f(x) &\stackrel{55}{\equiv} f(x^*) + f'(x^*)(x - x^*) = f'(x^*)(x - x^*) \\ &\stackrel{56}{\Rightarrow} f(x_i) \approx f'(x^*)(x_i - x^*) \\ &\stackrel{57}{\Rightarrow} |x_i - x^*| \approx \frac{|f(x_i)|}{|f'(x^*)|} \\ &\Rightarrow |x_n - x^*| \approx \frac{|f(x_n)|}{|f'(x^*)|} \end{aligned}$$

e quindi è possibile utilizzare $\frac{|f(x_i)|}{|f'(x^*)|} \leq tolx$ come criterio d'arresto ($i = 1, \dots, n$).

Al fine di utilizzare $f'(x^*)$ nel criterio di arresto, questa può essere approssimata, dato l'intervallo di confidenza $[a_i, b_i]$, come segue:

$$f'(x^*) \approx \frac{f(b_i) - f(a_i)}{b_i - a_i} = \frac{f(a_i + \overbrace{b_i - a_i}^h) - f(a_i)}{\underbrace{b_i - a_i}_h} \rightarrow \lim_{i \rightarrow \infty} \frac{f(b_i) - f(a_i)}{b_i - a_i}. \quad (2.3)$$

Osservazione 2.3 (Non ufficiale). Il costo computazionale dell'approssimazione $f'(x^*)$ calcolata come (2.3) è nullo perché $f(a_i)$ e $f(b_i)$ sono già calcolati nel metodo di bisezione.

Inoltre,

$$\frac{|f(x_i)|}{|f'(x^*)|} \leq tolx \iff |f(x_i)| \leq |f'(x^*)| \cdot tolx = tol f (\rightarrow |f(x_i)| \leq tol f),$$

con $tol f$ quantità precisa⁵⁸. In altre parole: per avere $|x_n - x^*| \leq tolx$ occorre utilizzare una tolleranza $tol f \approx |f'(x^*)| \cdot tolx$.

2.2.2 Condizionamento di una radice

⁵⁹ L'analisi appena condotta per definire un criterio d'arresto più efficiente per il metodo di bisezione è altresì legata al condizionamento della radice x^* e rimarrà valida anche per i metodi che saranno trattati successivamente.

Definizione 2.1 (Numero di condizionamento di una radice). Sia f ed x^* la radice trovata. Il numero di condizionamento della radice x^* è dato da

$$\kappa = \frac{1}{|f'(x^*)|}. \quad (2.4)$$

⁵² n è scelto in modo tale che si verichi ciò che segue.

⁵³ $\lceil \rceil$ significa arrotondamento all'intero successivo.

⁵⁴ Ricordando che di f è ricercata la radice ovvero $x^* : f(x^*) = 0$, allora è possibile quanto segue.

⁵⁵ Sviluppo di Taylor.

⁵⁶ È ricercata $f(x_i)$, ovvero è effettuata la sostituzione $x = x_i$.

⁵⁷ Essendo $|x_i - x^*|$ la quantità ricercata allora la conseguenza è logica.

⁵⁸ Se $tol f >> 1$ (vedi Figura 3), significa che x_i è vicino a x^* e ciò è un buon indicatore per determinare l'affidabilità legata all'errore.

⁵⁹ PG 23, slide 6 PDF lez7.

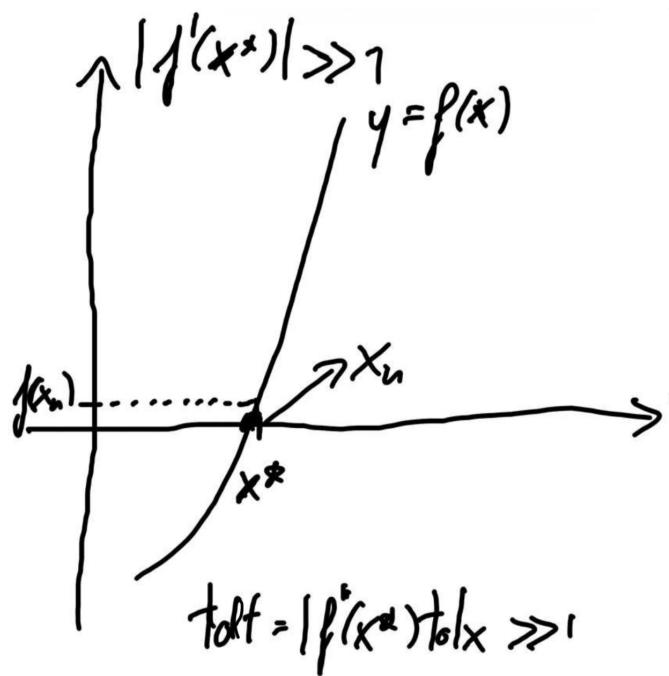


Figura 3: $f'(x^*) \gg 1$

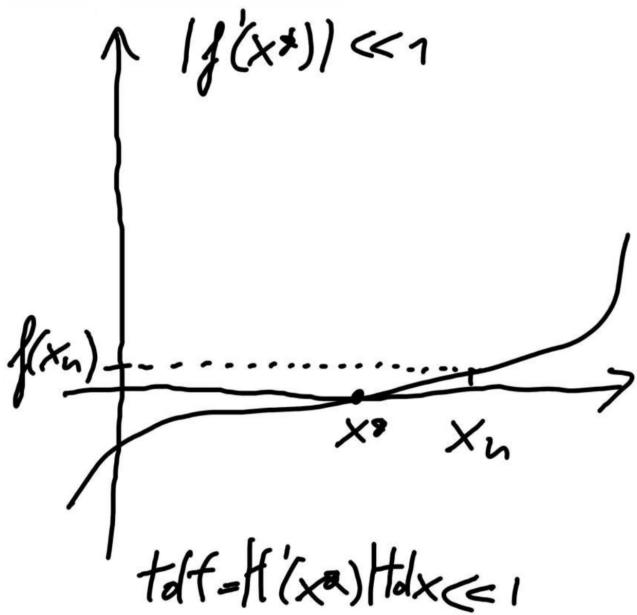


Figura 4: $f'(x^*) \ll 1$

Tale valore amplifica (di un fattore κ) l'errore $x_i - x^*$ commesso su x^* .

Osservazione 2.4 (Non ufficiale). È possibile osservare che:

- se $|f'(x^*)| \approx 0 \Rightarrow tol f \ll tol x \left(\frac{1}{|f'(x^*)|} \gg 1 \right)$, x^* è una radice malcondizionata;
- se $|f'(x^*)| \geq 1 \Rightarrow tol f \gg tol x \left(\frac{1}{|f'(x^*)|} \leq 1 \right)$, x^* è una radice bencondizionata.

Esempio 2.2 (Condizionamento radice di un polinomio).⁶⁰ Nel caso del polinomio $p(x) = (x - 1.1)^{20}(x - \pi)$ è ottenuta $p'(\pi) = (\pi - 1.1)^{20} \approx 1.6 \cdot 10^6$. Il numero di condizionamento vale $\kappa \approx 6 \cdot 10^{-7}$ e quindi la radice $x^* = \pi$ è ben condizionata.

Definizione 2.2 (Errore (perturbazione) della radice). L'errore (perturbazione) sul risultato (di approssimazione della radice) è

$$|x_n - x^*| \approx \frac{|f(x_n)|}{|f'(x^*)|} = \frac{1}{|f'(x^*)|} |f(x_n)|,$$

dove:

- $|f(x_n)|$ è la perturbazione rispetto al valore esatto $|f(x^*)| = 0$,
- $\frac{1}{|f'(x^*)|}$ il condizionamento della radice.

N.B.: È necessario osservare che il condizionamento riguarda il problema e non il metodo numerico.

Definizione 2.3 (Radice con molteplicità m).⁶¹ x^* è una radice del problema (2.1) ed ha molteplicità esatta $m \geq 1$, se è verificato quanto segue:

$$f(x^*) = f'(x^*) = f''(x^*) = \dots = f^{(m-1)}(x^*) = 0, \quad f^{(m)}(x^*) \neq 0.$$

Se $m = 1$ allora x^* si dice radice semplice, altrimenti (per $m \geq 2$) si dice multipla.

Il metodo di bisezione è malcondizionato con radici multiple: Dalla Definizione precedente e dalla Definizione 2.1 di condizionamento di una radice è possibile affermare che il problema della determinazione di radici multiple è sempre malcondizionato, quindi non è consigliata l'applicazione del metodo di bisezione, in quanto il numero di condizionamento risulterebbe essere $\kappa = \infty$ (sarà ottenuto un risultato, la radice, non preciso).

Convergenza del metodo di bisezione: Vedere Sezione 2.4.1.

2.3 Convergenza di un metodo iterativo

⁶² Lo scopo dell'argomento introdotto è quello di misurare l'errore di approssimazione della radice, quindi che il metodo iterativo sia corretto nel funzionamento. Ciò che sarà stabilito è la convergenza della successione alla radice.

Definizione 2.4 (Errore di approssimazione). Supposto di voler risolvere l'equazione (2.1) e sia x_i l'approssimazione fornita al passo i -esimo, è possibile definire l'errore come

$$e_i := x_i - x^*. \tag{2.5}$$

⁶⁰PG 25.

⁶¹PG 25, slide 7 PDF lez 7.

⁶²PG 26-28, slide 8-9 + 1-2 PDF lez7-8.

Algoritmo 2.1 Implementazione ottimale del metodo di bisezione.

```
function x = bisezione(f, a, b) % da applicare controllo f(a)f(b)<0 (
    Teorema degli zeri)
fa = feval(f, a);
fb = feval(f, b);
imax = ceil(log2(b-a) - log2(tolx));
for i = 1 : imax
    x = (a + b)/2;
    fx = feval(f, x);
    f1x = abs((fb - fa)/(b - a));
    if abs(fx) <= tolx * f1x
        break
    elseif fa * fx < 0
        b = x;
        fb = fx;
    else
        a = x;
        fa = fx;
    end
end
return
```

Definizione 2.5 (Metodo iterativo convergente). Un metodo iterativo è convergente s.se

$$\lim_{i \rightarrow \infty} e_i = 0. \quad (2.6)$$

2.4 Ordine di Convergenza di un metodo iterativo

Il passo successivo alla convergenza è stabilire la velocità computazionale del metodo (quanto velocemente x_n si avvicina a x^* ? Quanti calcoli servono?). Per questo è necessaria l'introduzione di **Definizione 2.6** di ordine di convergenza e **Osservazione 2.6**.

Definizione 2.6 (Ordine di convergenza di un metodo iterativo). Se un metodo iterativo ha $p \in \mathbb{R}^+$ come valore più grande per cui

$$\lim_{i \rightarrow \infty} \frac{|e_{i+1}|}{|e_i|^p} = c < \infty, \quad (2.7)$$

allora tale metodo ha **ordine di convergenza p** , con **costante asintotica dell'errore** pari a c .

Osservazione 2.5. Se $e_i < 1 \Rightarrow |e_i|^p$ è sempre più piccolo al crescere di p .

Osservazione 2.6. Per i sufficientemente grande, nel caso di convergenza lineare ($p = 1$), è ottenuto (data (2.7)):

$$\frac{|e_{i+1}|}{|e_i|^p} \approx c \Rightarrow |e_{i+1}| \approx c|e_i|^p.$$

Osservazione 2.7. Un metodo lineare ($p = 1$) è **convergente s.se $0 \leq c < 1$** .

Perché l'Osservazione 2.7 vale? Dato $|e_{i+1}| \approx c|e_i|^p$ allora, se $p = 1$ $|e_{i+1}| \approx c|e_i| \Rightarrow |e_{i+k}| \approx c^k|e_{i-k}| \Rightarrow |e_i| \approx c^i|e_0|$ quindi

$$\lim_{i \rightarrow \infty} |e_i| = \lim_{i \rightarrow \infty} c^i |e_0| \iff c < 1.$$

Più è elevato l'ordine di un metodo convergente, più le approssimazioni generate dal metodo convergono verso la radice x^* . Questo è accentuato nel prossimo esempio.

Esempio 2.3.⁶³ Considerati 2 metodi iterativi convergenti alla stessa radice x^* , con ordine di convergenza rispettivamente $p = 1$ e $p = 2$, entrambi con costante $c = 0.1$, se per entrambi $e_0 = 0.1$, allora

i	$p = 1$	$p = 2$
0	0.1	0.1
1	$c e_0 ^1 = 10^{-2}$	$c e_0 ^2 = 10^{-3}$
2	$c e_1 ^1 = 10^{-3}$	$c e_1 ^2 = 10^{-7}$
3	$c e_2 ^1 = 10^{-4}$	$c e_2 ^2 = 10^{-15}$
4	$c e_3 ^1 = 10^{-5}$	$c e_3 ^2 = 10^{-31}$

È possibile osservare come la tabella sia ottenuta dall'applicazione della Definizione 2.6 e che $e_i = c|e_{i-1}|^1$, $\forall i = 1, \dots, n$ (dove, nella tabella, $n = 4$).

2.4.1 Metodo di bisezione

Osservazione 2.8. (Convergenza metodo di bisezione) Il metodo di bisezione è sempre convergente (convergenza globale):

$$|e_i| = |x_i - x^*| \leq \frac{b-a}{2^i} \Rightarrow 0 \leq \lim_{i \rightarrow \infty} |e_i| \leq \lim_{i \rightarrow \infty} \frac{b-a}{2^i} = 0 \Rightarrow \lim_{i \rightarrow \infty} e_i = 0.$$

Inoltre, il metodo di bisezione ha ordine di convergenza assimilabile a $p = 1$ e costante asintotica dell'errore $c = \frac{1}{2}$, ovvero converge linearmente⁶⁴. Questo è dovuto a $\lim_{i \rightarrow \infty} \frac{|e_{i+1}|}{|e_i|} = \frac{1}{2}$, utilizzando la stima assegnata al metodo di bisezione $|e_i| \leq \frac{b-a}{2^i}$.

2.5 Metodo di Newton

⁶⁵ Il metodo di Newton è un metodo di approssimazione della radice con ordine migliore del metodo di bisezione. Questo metodo si basa su un'approssimazione lineare della funzione a partire dalla soluzione approssimata corrente, ovvero: supposto di conoscere un'approssimazione y della funzione, allora il sistema generale per determinare x^* tale che $f(x^*) = 0$ è

$$\begin{cases} y = f(x), \\ y = 0, \end{cases}$$

dal quale, attraverso l'utilizzo della retta tangente al grafico di f nel punto $(x_0, f(x_0))$

$$\begin{cases} y = f(x_0) + f'(x_0)(x - x_0), \\ y = 0, \end{cases} \quad (2.8)$$

⁶³PG 27, slide 1 PDF lez8.

⁶⁴Quindi il metodo di bisezione converge R -linearmente (informazione non richiesta).

⁶⁵Slide 3-8, 1-2 PDF 8,11 PDF 9,10.

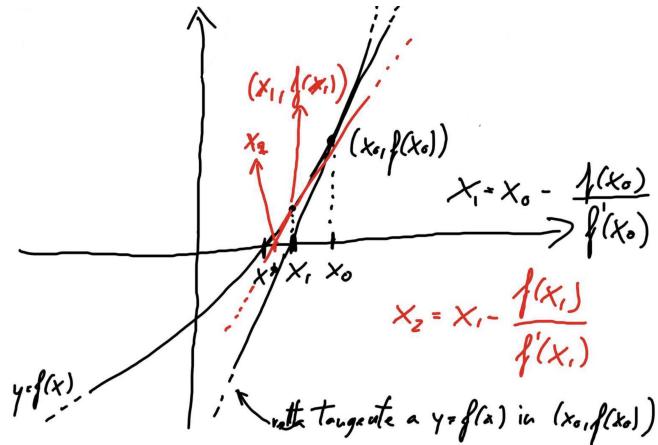


Figura 5: Grafico $y = f(x)$ con relative approssimazioni ottenute tramite Newton (è ricercata l'intersezione).

è ricavata, tramite calcoli intermedi⁽⁶⁶⁾, l'approssimazione (definita dall'approssimazione di tale retta con l'asse delle ascisse)

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)},$$

la quale è definita per $f'(x_0) \neq 0$, con x_0 approssimazione iniziale di x^* . Reiterando per i passi successivi, è possibile ottenere l'espressione funzionale del metodo di Newton:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad i = 0, 1, 2, \dots \quad (2.9)$$

Più passi vengono calcolati, più x_i si avvicina a x^* .

Osservazioni sul metodo: La soluzione al problema (2.1) è ottenuta risolvendo equazioni lineari. Nel caso in cui $f(x)$ sia una funzione lineare allora il metodo di Newton fornisce la soluzione in un solo passo.

Costo computazionale: È possibile osservare che il metodo di Newton ha un costo per iterazione di 2 valutazioni funzionali (il metodo di bisezione ne richiede una): è necessario calcolare $f(x)$ ed $f'(x)$, in quanto ad ogni passo/iterazione cambia l'input (x_0, x_1, x_2, \dots) . Inoltre, mentre il metodo di bisezione richiede solo la continuità della funzione f , per il metodo di Newton è richiesto che f sia, oltre che continua, derivabile (ovvero $f \in C^{(2)}$). Il costo computazionale maggiore e maggiori requisiti del metodo, rispetto al metodo di bisezione, sono compensati dall'elevato ordine di convergenza del metodo.

Teorema 2.2 (Metodo di Newton converge quadraticamente, nome non ufficiale). Se $f(x)$ è sufficientemente regolare, il **metodo di Newton converge quadraticamente** (ovvero con ordine $p = 2$) verso radici semplici⁶⁷.

⁶⁶Da (2.8) $y = 0 \Rightarrow \frac{f(x_0)}{f'(x_0)} + f'(x_0)(x - x_0) = 0 \Rightarrow x - x_0 = -\frac{f(x_0)}{f'(x_0)} \Rightarrow x = x_0 - \frac{f(x_0)}{f'(x_0)}$.

⁶⁷Ovvero radici di molteplicità 1, le quali sono definite come $x^* : f(x^*) = 0, f'(x^*) \neq 0$.

Dimostrazione. Assunto che $f \in C^{(2)}$ in un intorno della radice x^* , preso un opportuno ξ_i (non noto) compreso fra x^* e x_i , tramite lo sviluppo di Taylor di ordine 2 centrato in x_i

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{f''(\xi_i)}{2}(x - x_i)^2, \quad \xi_i \text{ compreso fra } x, x_i,$$

è valutata f tramite lo sviluppo di Taylor centrato in x^*

$$\begin{aligned} 0 &= f(x^*) \\ &= f(x_i) + f'(x_i)(x^* - x_i) + \frac{f''(\xi_i)}{2}(x^* - x_i)^2 \\ {}^{68} &= f'(x_i) \left[\frac{f(x_i)}{f'(x_i)} - x_i + x^* \right] + \frac{f''(\xi_i)}{2}(x^* - x_i)^2 \\ {}^{69} &= f'(x_i)(-x_{i+1} + x^*) + \frac{f''(\xi_i)}{2}(x^* - x_i)^2 \\ {}^{70} &= -f'(x_i)e_{i+1} + \frac{f''(\xi_i)}{2}e_i^2 \quad \rightarrow \quad \frac{f''(\xi_i)}{2}e_i^2 = -f'(x_i)e_{i+1} \\ &\Rightarrow \frac{|e_{i+1}|}{|e_i|^2} = \frac{1}{2} \frac{|f''(\xi_i)|}{|f'(x_i)|}, \quad x^* < \xi_i < x_i. \end{aligned}$$

Allora, supponendo che il metodo converga (ovvero che $\lim_{i \rightarrow +\infty} x_i = x^*$):

$$\lim_{i \rightarrow \infty} \frac{|e_{i+1}|}{|e_i|^2} = \lim_{i \rightarrow \infty} \frac{1}{2} \frac{f''(\xi_i)}{f'(x_i)} \stackrel{72}{=} \frac{1}{2} \frac{f'' \left(\lim_{i \rightarrow \infty} \xi_i \right)}{f' \left(\lim_{i \rightarrow \infty} x_i \right)} = \frac{1}{2} \frac{f''(x^*)}{f'(x^*)},$$

quindi, per $f'(x^*) = 0$ (x^* radice semplice)⁷³:

$$\lim_{i \rightarrow \infty} \frac{|e_{i+1}|}{|e_i|^2} = \frac{1}{2} \frac{f''(x^*)}{f'(x^*)} < \infty,$$

e il metodo ha ordine di convergenza $p = 2$.⁷⁴ □

Teorema 2.3 (Convergenza lineare di Newton, nome non ufficiale). ⁷⁵ Se $f(x)$ è sufficientemente regolare in un intorno di x^* , radice di molteplicità $m > 1$, il metodo di Newton converge linearmente ($p = 1$) verso una radice di molteplicità $m > 1$, con costante asintotica dell'errore $c = \frac{m-1}{m}$.

Dimostrazione. ⁷⁶ Sia x^* radice con molteplicità $m > 1$, è possibile esprimere f come $f(x) = (x - x^*)^m g(x)$, con g funzione tale che $g(x^*) \neq 0$ e sviluppabile in serie di Taylor centrato in x^* . È possibile ottenere quanto segue, per

⁶⁸Raccolto $f'(x_i)$ perché interessante la q.tà $\frac{f(x_i)}{f'(x_i)} - x_i$ (ovvero x_{i+1} cambiato di segno).

⁶⁹Sostituzione di $\frac{f(x_i)}{f'(x_i)} - x_i$ con $-x_{i+1}$.

⁷⁰Sostituzione $x^* - x_{i+1} = -e_{i+1}$ e $x^* - x_i = -e_i$.

⁷¹L'obbiettivo è valutare quanto segue.

⁷² $f \in C^{(2)}$, quindi f'' è continua.

⁷³Ipotesi importante perché significa che $f'(x^*) \neq 0$ e che quindi $\frac{f''(x^*)}{f'(x^*)} \neq \infty$.

⁷⁴L'ordine è almeno 3 se $f''(x^*) = 0$.

⁷⁵Teorema utilizzato per radici non semplici.

⁷⁶Slide 8, 1-2 PDF lez8, lez9.

$x_i \approx x^*$ (vedi (2.5), (2.9) e l'Osservazione 11.1):

$$\begin{aligned}
\frac{e_{i+1}}{e_i} &= \frac{\frac{x_{i+1}-x^*}{x_i-x^*}}{\frac{(x_i-x^*)^m g(x_i)}{m(x_i-x^*)^{m-1} g(x_i)+(x_i-x^*)^m g'(x_i)}} & \stackrel{77}{=} & \frac{x_i - \frac{f(x_i)}{f'(x_i)} - x^*}{x_i - x^*} \\
&\stackrel{78}{=} \frac{\cancel{x_i - x^*} - \frac{(x_i-x^*)^m g(x_i)}{m(x_i-x^*)^{m-1} g(x_i)+(x_i-x^*)^m g'(x_i)}}{\cancel{x_i - x^*}} & = & 1 - \frac{(x_i-x^*)^{m-1} g(x_i)}{m(x_i-x^*)^{m-1} g(x_i)+(x_i-x^*)^m g(x_i)} \\
&= \frac{\frac{(m-1)g(x_i)+e_i g'(x_i)}{mg(x_i)+e_i g'(x_i)}}{\frac{m(x_i-x^*)^{m-1} g(x_i)+(x_i-x^*)^m g'(x_i)}{mg(x_i)+e_i g'(x_i)}} & \stackrel{79}{=} & \frac{mg(x_i)+(x_i-x^*)g'(x_i)-g(x_i)}{mg(x_i)+(x_i-x^*)g'(x_i)} \\
&\stackrel{80}{=} \frac{(m-1)g(x_i)+e_i g'(x_i)}{mg(x_i)+e_i g'(x_i)} & \stackrel{81}{\Rightarrow} & \frac{e_{i+1}}{e_i} = \frac{(m-1)g(x_i)+e_i g'(x_i)}{mg(x_i)+e_i g'(x_i)}.
\end{aligned}$$

Allora, dalla (2.6):

$$\lim_{i \rightarrow \infty} \frac{|e_{i+1}|}{|e_i|} = \lim_{i \rightarrow \infty} \frac{|(m-1)g(x_i) + e_i g'(x_i)|}{|mg(x_i) + e_i g'(x_i)|} \stackrel{82}{=} \lim_{i \rightarrow \infty} \frac{|(m-1)\cancel{g(x_i)}|}{|m\cancel{g(x_i)}|} = \frac{m-1}{m}. \quad (2.10)$$

È dimostrata così la tesi, ovvero: il metodo di Newton converge linearmente con costante asintotica $\frac{m-1}{m}$. \square

2.5.1 Convergenza del metodo di Newton

⁸³ L'ordine di convergenza di un metodo convergente quantifica la "velocità" di avvicinamento delle approssimazioni alle radici. È necessario stabilire le condizioni che garantiscono la convergenza del metodo, ovvero che il metodo generi una successione di approssimazioni $\{x_i\}$ che soddisfi (2.5)-(2.6).

Nel caso del metodo di bisezione, assumendo che $f \in C([a, b])$ t.c. $f(a)f(b) < 0$, è possibile affermare che il metodo abbia proprietà di convergenza globale. Tale proprietà garantisce sempre la convergenza della serie di approssimazioni verso una radice di f . Purtroppo questa è pressoché proprietà esclusiva del metodo di bisezione.

Il metodo di Newton è convergente in un opportuno intorno della radice che dipende dalla scelta dell'approssimazione iniziale x_0 .

Esempio 2.4 (Controesempio di non convergenza). Data l'approssimazione $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$, è applicato il metodo di Newton a $f(x) = x^3 - 5x$, con approssimazione iniziale $x_0 = 1$. Calcolata f' di f (prima cosa da calcolare), ovvero $f'(x) = 3x^2 - 5$, è possibile il calcolo le seguenti approssimazioni:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 1 - \frac{1-5}{3-5} = -1; \quad x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 1 - \frac{-1+5}{3-5} = 1; \quad x_3 = -1; \quad x_4 = 1, \quad \dots$$

La successione di approssimazioni ottenuta non converge ad alcuna radice (ed x_0 non è una radice).

Tramite la Figura 6 è possibile osservare quanto segue:

⁷⁷ Sostituzione di x_{i+1} con $x_i - \frac{x_i}{f'(x_i)}$ per applicare la definizione del metodo di Newton.

⁷⁸ Dato che $f(x) = (x - x^*)^m g(x)$ allora $f'(x) = m(x - x^*)^{m-1} g(x) + (x - x^*)^m g'(x)$.

⁷⁹ Divisione di numeratore e denominatore per $(x_i - x^*)^{m-1}$.

⁸⁰ Raccoglimento $g(x_i)$ e sostituzione di $(x_i - x^*)$ con e_i .

⁸¹ È dimostrato quanto segue.

⁸² È supposto che il problema di Newton converga quindi, per il fatto stesso di convergere (vedi (2.6)), l'errore (e_i) tende a 0, quindi $\lim_{i \rightarrow \infty} e_i g'(x_i) \rightarrow 0$.

⁸³ Sul libro è "Convergenza Locale", slide 3-10, 1-3 PDF lez9, lez10 PG 30-33.

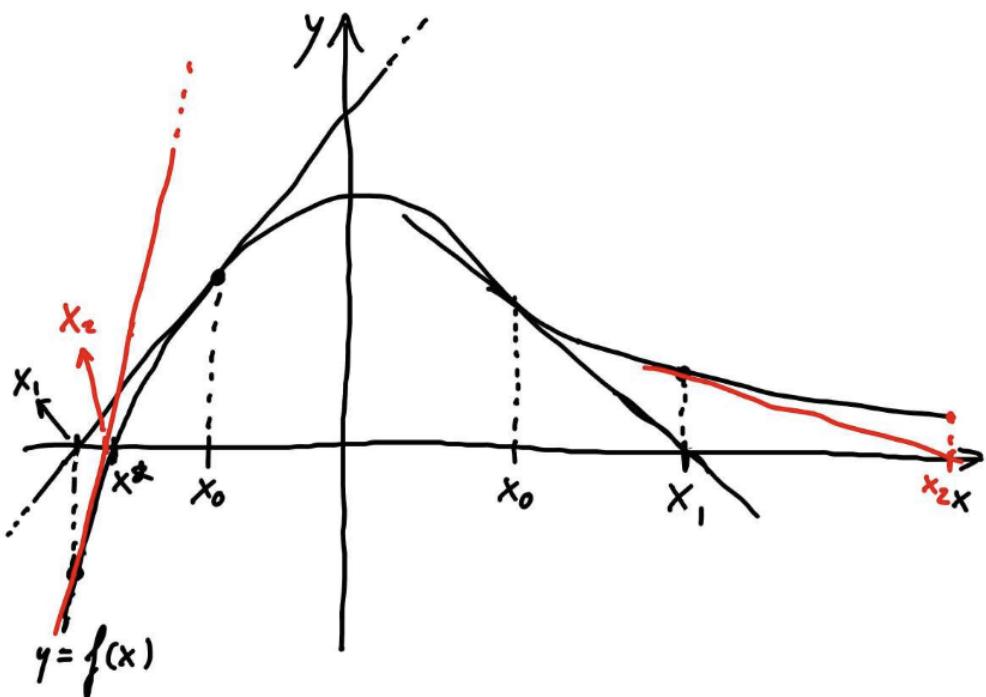


Figura 6: Esempio dal punto di vista geometrico della convergenza.

- più le rette tangenti (le linee che toccano il grafico di $y = f(x)$) si avvicinano a x^* , più il metodo converge ed è preciso;
- il problema di convergenza sorge se x_0 è scelto come nel primo quadrante perché i successivi calcoli (x_1, x_2, \dots) si allontanano da x^* ; quindi è necessario scegliere x_0 nell'intervallo locale della radice in base alla funzione.

2.5.2 Studio della convergenza locale

Questa Sezione è una formalizzazione della Figura 6. Lo studio della convergenza locale in un contesto generale può essere formalizzata come segue: considerato un metodo iterativo per l'approssimazione di una radice, l'approssimazione della radice x^* di f , è definita da

$$x_{i+1} = \Phi(x_i) \quad i = 0, 1, 2, \dots, \quad (2.11)$$

con $\Phi(x_i)$ funzione d'iterazione, la quale caratterizza il metodo di iterazione generico ⁸⁴.

Affinché il metodo iterativo abbia senso, supposto che $x_i \rightarrow x^*$ per $i \rightarrow \infty$, è richiesto che x^* sia il **punto fisso** della funzione di iterazione $\Phi(x)$, ovvero:

$$x^* = \Phi(x^*).$$

Definizione 2.7 (Funzione d'iterazione del metodo di Newton). Per il metodo di Newton la funzione di iterazione del metodo è

$$\Phi(x) = x - \frac{f(x)}{f'(x)}, \quad (2.12)$$

quindi

$$\Phi(x^*) = x^* - \frac{f(x^*)}{f'(x^*)} \stackrel{f(x^*)=0}{=} x^*.$$

Questo argomento permette di trattare le proprietà di convergenza del metodo iterativo, mediante lo studio delle proprietà di stabilità del punto fisso corrispondente della funzione di iterazione. La convergenza locale dei metodi può essere studiata mediante declinazioni del seguente teorema.

Teorema 2.4 (del punto fisso). ⁸⁵ Sia $\Phi(x)$ la funzione d'iterazione (2.11) che definisce il metodo numerico. Supposto che $\exists \delta > 0$, $0 \leq L < 1$, costanti tali che:

$$|\Phi(x) - \Phi(y)| \leq L|x - y| \quad \forall x, y \in I = (x^* - \delta, x^* + \delta),$$

allora:

1. x^* è l'unico punto fisso di Φ in I ;
2. se $x_0 \in I \Rightarrow x_{i+1} = \Phi(x_i) \in I$, $i = 0, 1, 2, \dots$;
3. se $x_0 \in I \Rightarrow \lim_{i \rightarrow \infty} x_i = x^*$. (proprietà più interessante)

Dimostrazione. La dimostrazione avverrà per punti:

⁸⁴Tale funzione è una regola che fornisce una buona approssimazione applicando Φ a partire da x_i .

⁸⁵Slide 6 PDF lez9, PG 31-32.

1. Per assurdo (è supposto che esistano due punti fissi): $\exists x^*, \bar{x} \in I : \Phi(x^*) = x^*, \Phi(\bar{x}) = \bar{x}$. Poichè $x \neq \bar{x}$, segue:

$$|x^* - \bar{x}| = |\Phi(x^*) - \Phi(\bar{x})| \stackrel{86}{\leq} L|x^* - \bar{x}| < |x^* - \bar{x}| \Rightarrow \text{Assurdo } (|x^* - \bar{x}| \not\prec |x^* - \bar{x}|).$$

Quindi il punto fisso in I è unico (e tale punto è x^*);

2. Per induzione: dato $x_0 \in I = (x^* - \delta, x^* + \delta)$ è supposto che $x_i \in I \iff |x_i - x^*| < \delta$, quindi è possibile dimostrare che $x_i \in I$ come segue:

$$|x_{i+1} - x^*| = |\Phi(x_i) - \Phi(x^*)| \stackrel{88}{\leq} L|x_i - x^*| < L\delta < \delta \stackrel{89}{\Rightarrow} |x_{i+1} - x^*| < \delta \Rightarrow x_{i+1} \in I;$$

3. È necessario dimostrare

$$\lim_{i \rightarrow \infty} x_i = x^* \iff \lim_{i \rightarrow \infty} |x_i - x^*| = 0.$$

Ciò che interessa calcolare è $|x_i - x^*|$, ovvero l'errore:

$$\begin{aligned} |x_i - x^*| &\stackrel{90}{=} |\Phi(x_{i-1}) - \Phi(x^*)| \leq L|x_{i-1} - x^*| = L|\Phi(x_{i-2}) - \Phi(x^*)| \\ &\leq L \cdot L|x_{i-2} - x^*| = L^2|x_{i-2} - x^*| \leq \dots \leq L^i|x_0 - x^*| \\ &\Rightarrow |x_i - x^*| \leq L^i|x_0 - x^*|. \end{aligned}$$

Quindi è possibile dimostrare la tesi come segue:

$$0 \leq \underbrace{\lim_{i \rightarrow \infty} |x_i - x^*|}_{91} \leq \lim_{i \rightarrow \infty} L^i|x_0 - x^*| = 0 \Rightarrow \lim_{i \rightarrow \infty} |x_i - x^*| = 0 \Rightarrow \lim_{i \rightarrow \infty} x_i = x^*.$$

□

⁹² Inoltre, è possibile dimostrare che la funzione (2.12) soddisfa le ipotesi del Teorema del punto fisso (Teorema 2.4), come segue: assumendo $f(x) \in C^{(2)}$ in un intorno di x^* , dove x^* è la radice semplice di f e definita $\Phi(x) \in C^{(1)}$ in un intorno di x^* (come in (2.12)) con $\Phi'(x) = 1 - \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2}$, allora

$$\Phi'(x^*) = 1 - \frac{[f'(x^*)]^2 - f(x^*)f''(x^*)}{[f'(x^*)]^2} = 1 - \frac{[f'(x^*)]^2}{[f'(x^*)]^2} = 1 - 1 = 0.$$

Quindi, $\Phi \in C^{(1)} \Rightarrow \Phi' \in C^{(0)}$ e $\Phi'(x^*) = 0$.

Applicando la definizione di funzione continua⁹³ a $\Phi = g$, $\varepsilon = L$ e fissato $0 < L < 1$:

$$\exists \delta > 0 : |\Phi'(x) - \Phi'(x^*)| < L \quad \forall x \in (x^* - \delta, x^* + \delta).$$

⁸⁶Applicazione del teorema.

⁸⁷ $|x^* - \bar{x}|$ non può essere minore di se stesso.

⁸⁷Sfruttando le ipotesi del teorema.

⁸⁸ $x_i, x^* \in I$.

⁸⁹ $L < 1$, quindi è valido ciò che segue.

⁹⁰ $x_{i-1}, x^* \in I$ perché $x_0 \in I$ e vale 2.

⁹¹Limite calcolato con la disuguaglianza precedente.

⁹²Slide 1 PDF lez10, PG 32.

⁹³Definizione di funzione continua: g è continua in x^* s.se $\forall \varepsilon > 0 : \exists \delta > 0 : |g(x) - g(x^*)| < \varepsilon \quad \forall x \in I = (x^* - \delta, x^* + \delta)$.

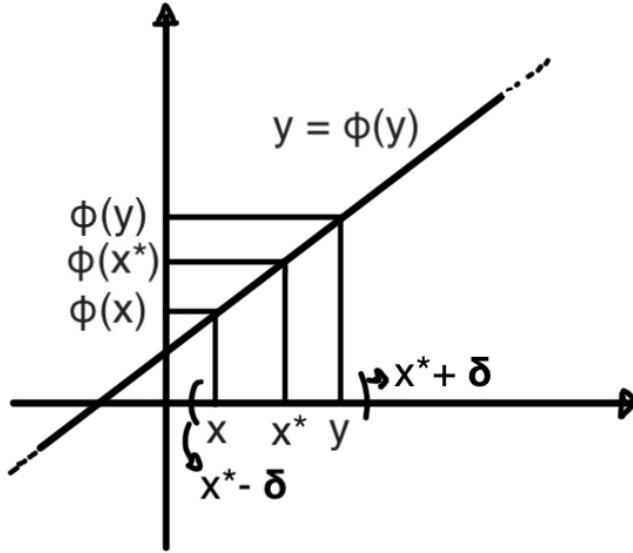


Figura 7: Esempio del grafico di una funzione d'iterazione per il Teorema 2.4.

L'ipotesi è verificata con L e δ scelte, utilizzando lo sviluppo di Taylor, in quanto

$$\begin{aligned} \forall x, y \in I = (x^* - \delta, x^* + \delta) : |\Phi(x) - \Phi(y)| &\stackrel{94}{=} |\Phi(x) - \Phi(x^*) - \Phi'(x^*)(x - x^*)| && \stackrel{95}{=} |\Phi'(x^*)(x - x^*)| \\ &= |\Phi'(x^*)||x - x^*| && < L|x - x^*| \\ &\Rightarrow |\Phi(x) - \Phi(y)| && \leq L|x - y|. \end{aligned}$$

2.5.3 Criteri d'arresto per il metodo di Newton (e non solo)

⁹⁶ Gli argomenti trattati nella Sezione 2.2, riguardo la definizione di un criterio d'arresto basato sul controllo di $|f(x)|$ (ovvero $|f(x_i)| \leq |f'(x^*)| tol x$), sono applicati anche al caso del metodo di Newton (ed ai metodi quasi-Newton trattati in seguito). A causa della convergenza del metodo non è possibile determinare a priori il numero massimo di iterazioni entro le quali il criterio di accuratezza sulla approssimazione calcolata sarà soddisfatto. Dunque, in prossimità della radice x^* , nel caso di ordine di convergenza $p > 1$,

$$|x_{i+1} - x_i| = \left| \underbrace{x_{i+1} - x^*}_{e_{i+1}} + \underbrace{x^* - x_i}_{e_i} \right| \stackrel{97}{=} |e_i - e_{i+1}| \approx |e_i|. \quad (2.13)$$

Pertanto, un **criterio di arresto appropriato**, per metodi di **ordine di convergenza $p > 1$** è del tipo

$$|x_{i+1} - x_i| \leq tol x. \quad (2.14)$$

⁹⁴È necessario dimostrare che sia minore di $L|x - y|$. Inoltre, $\Phi(y) = \Phi(x) + \Phi'(\xi)(x - y)$ con $\xi \in (x, y) \rightarrow \xi \in I$.

⁹⁵Approssimazione di $f(x)$ con sviluppo di Taylor.

⁹⁶Slide 4 PDF lez10, PG 33.

⁹⁷ $e_{i+1} \approx ce_i$. $\lim_{i \rightarrow \infty} \frac{|e_{i+1}|}{|e_i|^p} = c < \infty \Rightarrow |e_{i+1}| \approx c|e_i|^p \stackrel{p \geq 1}{\Rightarrow} e_{i+1}$ è trascurabile rispetto a e_i .

Questo è dovuto al fatto che l'errore $|x^* - x_i|$ non è noto, quindi è necessario stimarlo, come in precedenza.

Osservazione 2.9 (Criterio d'arresto per il metodo di Newton). Risulta essere sempre applicabile il seguente criterio d'arresto

$$|\mathbf{f}(x_i)| \leq |\mathbf{f}'(x_i)| \cdot \mathbf{tolx} \equiv \frac{|f(x_i)|}{|f'(x_i)|} \leq \mathbf{tolx}, \quad (2.15)$$

il quale è ottenuto dalla Sezione 2.2 tramite la diseguaglianza (2.2) (\mathbf{tolx} ha lo stesso significato che ha in (2.2)).

Osservazione 2.10 (Criterio d'arresto ideale per il metodo di Newton). ⁹⁸ Oltre alla tolleranza sul massimo errore assoluto, \mathbf{tolx} , è possibile utilizzare la tolleranza sull'errore relativo, \mathbf{rtolx} (o sull'accuratezza dell'approssimazione dello zero). In questo caso il controllo di arresto "ideale" diviene

$$\frac{e_i}{\mathbf{tolx} + \mathbf{rtolx}|x^*|} \leq 1$$

e mediante le approssimazioni $x^* \approx x_{i+1}$ e $|e_i| \approx |x_{i+1} - x_i|$ il criterio da considerare è

$$\frac{|x_{i+1} - x_i|}{\mathbf{tolx} + \mathbf{rtolx}|x_{i+1}|} \leq 1. \quad (2.16)$$

dove la diseguaglianza $e_i \leq \mathbf{tolx} + \mathbf{rtolx}|x^*|$ può essere rappresentata come

$$\frac{1}{2}|e_i| \leq \mathbf{tolx} \wedge \frac{1}{2}|e_i| \leq \mathbf{rtolx}|x^*| \iff \frac{1}{2}\frac{|e_i|}{x^*} \leq \mathbf{rtolx}$$

permettono di dividere in due il controllo (mediante ciò che è in grassetto). È spesso considerata la scelta $\mathbf{rtolx} = \mathbf{tolx}$, trasformando il criterio d'arresto (2.16) in

$$\frac{|x_{i+1} - x_i|}{1 + |x_{i+1}|} \leq \mathbf{tolx}.$$

Nel caso in cui la convergenza sia lineare (ovvero l'ordine di convergenza è $p = 1$), il criterio d'arresto (2.14) può essere modificato nel seguente (come 2.16), date (2.6)-(2.7) e (2.10)):

$$|x_{i+1} - x_i| = |e_{i+1} - e_i| = |e_i - \underbrace{e_{i+1}}_{c \cdot e_i}| \approx |e_i|(1 - c) \Rightarrow |e_i| \approx \frac{|x_{i+1} - x_i|}{(1 - c)}. \quad (2.17)$$

Se c non è nota (nel caso del metodo di bisezione è $c = \frac{1}{2}$), è possibile, tramite (2.17), la seguente stima della costante asintotica c :

$$\frac{|x_{i+1} - x_i|}{|x_i - x_{i-1}|} \approx \frac{\cancel{|e_{i-1}|} \cancel{c(1-c)}}{\cancel{|e_{i-1}|} \cancel{(1-c)}} = c,$$

la quale è ottenuta tramite i iterazioni del tipo

$$\begin{aligned} i = 0 & : |x_1 - x_0| \approx |e_0|(1 - c); \\ i = 1 & : |x_2 - x_1| \approx |e_1|(1 - c) \stackrel{100}{\approx} |e_0| c(1 - c); \\ & \vdots \quad \vdots \quad \vdots \\ i & : |x_i - x_{i-1}| \approx |e_{i-1}|(1 - c); \\ i + 1 & : |x_{i+1} - x_i| \approx |e_i|(1 - c) = |e_{i-1}| c(1 - c); \end{aligned}$$

⁹⁸Osservazione 2.3 PG 33, Slide 5 PDF lez10.

⁹⁹Senza valore assoluto perché, per definizione di convergenza, i metodi convergenti hanno costante asintotica c minore di 1.

per renderla più precisa. Tale stima è utile per avere una buona approssimazione dell'errore. Pertanto, è necessario considerare il costo computazionale delle i iterazioni (sono necessarie almeno due iterazioni).

2.5.4 Caso delle radici multiple

¹⁰¹ Nel caso in cui la molteplicità di x^* sia $m > 1$, è noto che il problema di determinare le radici sia malcondizionato. Inoltre, è noto che il metodo di Newton risulti essere solo lineare. Tuttavia, è possibile modificare tale metodo per ripristinare la convergenza quadratica, distinguendo i seguenti casi.

2.5.4.1 Molteplicità $m > 1$ nota

Avere m nota non è una caratteristica banale. Applicando l'Osservazione 11.1, per semplicità, al metodo di Newton, per determinare la radice (2.9), allora, è ottenuta

$$x_{i+1} = x_i - \frac{x_i - x^*}{m}, \quad i = 0, 1, 2, \dots$$

Pertanto, il metodo di Newton è modificato come segue:

$$x_{i+1} = x_i - m \frac{f(x_i)}{f'(x_i)}, \quad i = 0, 1, 2, \dots, \quad (2.18)$$

ripristinando la convergenza quadratica del metodo, qualora converga verso una radice di molteplicità esatta m .

Nota: Saranno presenti riferimenti a questo paragrafo ed a (2.18) come metodo di Newton modificato.

Convergenza: Nel caso generale la **convergenza** rimane **quadratica**.

2.5.5 Molteplicità $m > 1$ ignota

In questo caso il metodo di Newton converge linearmente ($p = 1$), quindi (da (2.7))

$$\lim_{i \rightarrow \infty} \frac{|e_{i+1}|}{|e_i|} = c < \infty.$$

Allora, per i sufficientemente grande:

$$\begin{aligned} \frac{|e_{i+1}|}{|e_i|} \approx c &\Rightarrow \frac{e_i}{e_{i+1}} \approx \frac{c \cdot e_{i-1}}{c \cdot e_i} \\ &\Rightarrow \frac{e_i}{e_{i+1}} \approx \frac{e_{i-1}}{c} \\ &\stackrel{102}{\Rightarrow} \frac{e_i^2}{e_i} = e_i e_i \approx \cancel{c} \cdot e_{i-1} \cdot \frac{e_{i+1}}{\cancel{c}} = e_{i-1} \cdot e_{i+1} \\ &\Rightarrow e_i^2 \approx e_{i-1} \cdot e_{i+1}. \end{aligned}$$

È possibile riscrivere la precedente approssimazione, l'ultima implicazione, come

¹⁰⁰ $e_{i+1} \approx c \cdot e_i$.

¹⁰¹ Slide 1-3 PDF lez11, PG 34-36.

¹⁰² Dalle due relazioni (approssimazioni) è possibile ottenere ciò che segue.

$$\underbrace{(x_i - x^*)^2}_{e_i^2} \approx \underbrace{(x_{i-1} - x^*)}_{e_{i-1}} \underbrace{(x_{i+1} - x^*)}_{e_{i+1}}$$

e da questa segue che

$$\begin{aligned} x_i^2 - 2x_i x^* + (x^*)^2 &\approx x_{i-1} x_{i+1} - x_{i-1} x^* - x_{i+1} x^* + (x^*)^2 \\ &\rightarrow -2x_i x^* + x_{i-1} x^* + x_{i+1} x^* \approx -x_i^2 + x_{i-1} x_{i+1} \\ &\rightarrow (-2x_i + x_{i-1} + x_{i+1}) x^* \approx -x_i^2 + x_{i-1} x_{i+1}. \end{aligned}$$

Da quest'ultima approssimazione è possibile la seguente approssimazione della radice:

$$x^* \approx x_i^* \equiv \frac{x_i^2 - x_{i-1} \cdot x_{i+1}}{2x_i - x_{i-1} - x_{i+1}}, \quad (2.19)$$

dove x_i^* rappresenta l'approssimazione di x^* al passo i -esimo.

2.6 Metodo di Aitken

¹⁰³ Il metodo utilizza una procedura a due livelli:

1. Vengono eseguiti due passi del metodo di Newton (2.9):

- $x_{i-1} = x_{i-1}^*$;
- $x_i = x_{i-1} - \frac{f(x_{i-1})}{f'(x_{i-1})}$;
- $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$;

2. Passo di Aitken: esecuzione del passo (2.19) di accelerazione, approssimare al meglio la radice. Questo passo fornirà il nuovo punto iniziale per il livello interno.

La successione $\{x_i^*\}_{i=1,2,\dots}$ generata con Aitken converge quadraticamente verso x^* .

Costo computazionale Pertanto, il prezzo di avere un'iterazione a due livelli è pagato con un costo computazionale doppio rispetto al classico Newton, ad ogni iterazione è applicato Newton due volte (per le quali sono necessarie le valutazioni di f e f').

2.7 Metodi quasi-Newton

¹⁰⁴ È possibile ridurre il costo di ogni iterazione del tipo (2.9) attraverso un'approssimazione di $f'(x_i)$. I metodi in questa Sezione, variazioni del metodo di Newton, calcolano x_{i+1} come segue:

$$x_{i+1} = x_i - \frac{f(x_i)}{\varphi_i}, \quad i = 0, 1, 2, \dots, \quad \varphi_i \approx f'(x_i). \quad (2.20)$$

¹⁰³ Slide 4-5 PDF lez11, PG 36-37.

¹⁰⁴ Slide 6-10 PDF 11, PG 37-40.

Algoritmo 2.2 Implementazione efficiente del metodo di Newton.

```
function x = newton(f, f1, x0, tol, itmax)
%
%   x = newton(f, f1, x0, tol, itmax)
%
%   Metodo di Newton per la ricerca della radice di una funzione
%
% Input:
%   f   -   function che implementa f(x);
%   f1  -   function che implementa f'(x);
%   x0  -   punto iniziale;
%   tol -   tolleranza richiesta (default 1e-12);
%   itmax - numero massimo di iterazioni (default 1000);
%
% Output:
%   x   -   soluzione approssimata.
%
% Dalla riga successiva fino a x = x0 sono controlli strutturali.
if nargin < 5
    itmax=1000;
else
    if itmax < 1, error('itmax errato'); end
end
if nargin < 4
    tol = 1e-12;
else
    if tol < 0, error('tolleranza negativa'); end
    tol = max(tol, 10*eps);
end
if nargin < 3, error('numero argomenti di ingresso errato'); end
x = x0;
for i = 1 : itmax
    xold = x;
    fx = feval(f,x);
    f1x = feval(f1, x);
    if f1x == 0, error('il metodo non converge'); end
    x = x - fx/f1x;
    err = abs(x-xold);
    if err <= tol, break; end
end
if err > tol, warning('tolleranza richiesta non soddisfatta'), end
return
```

Algoritmo 2.3 Implementazione metodo di Aitken.

```
function x = aitken(f, f1, x, tolx, itmax)
x = x0;
for i = 1 : itmax
    x0 = x;
    fx = feval(f, x0);
    f1x = feval(f1, x0);
    x1 = x0 - fx/f1x;
    fx = feval(f, x1);
    f1x = feval(f1, x1);
    x = x1 - fx/f1x;
    x = (x*x0 - x1^2)/(x - 2*x1 + x0);
    if abs(x-x0) <= tolx, break, end
end
if abs(x-x0) > tolx, warning('il metodo non converge'), end
```

2.7.1 Metodo delle corde

È assunto che $f(x)$ sia sufficientemente regolare ed è presupposto che la derivata vari di poco in prossimità della radice. Pertanto, se x_0 è vicino alla radice, allora è possibile approssimare f' come

$$f'(x_i) \approx f'(x_0) \equiv \varphi_i, \quad i = 0, 1, 2, \dots$$

In questo modo è ottenuta l'iterazione

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_0)}, \quad i = 0, 1, 2, \dots, \tag{2.21}$$

la quale definisce il metodo delle corde.

È necessario notare che l'approssimazione della derivata sia una costante (ovvero $f'(x_0)$).

Costo computazionale: ¹⁰⁵ 1 valutazione funzionale ($f(x_i)$).

Covergenza: Si, locale.

Ordine di convergenza: È lineare ($p = 1$) qualsiasi sia la molteplicità della radice.

Il metodo è utilizzato per la risoluzione di sistemi lineari per il basso costo di computazionale.

2.7.2 Metodo delle secanti

¹⁰⁶ Supposto siano già calcolati x_{i-1} ed x_i , allora $f(x) \approx f(x_i) + (x - x_i)f'(x_i)$. La derivata può essere approssimata come

$$f'(x_i) \approx \frac{f(x) - f(x_i)}{x - x_i} \xrightarrow{x=x_{i-1}} f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \equiv \varphi_i.$$

¹⁰⁵ Il costo computazionale è considerato per iterazione.

¹⁰⁶ Slide 7-10 PDF lez11.

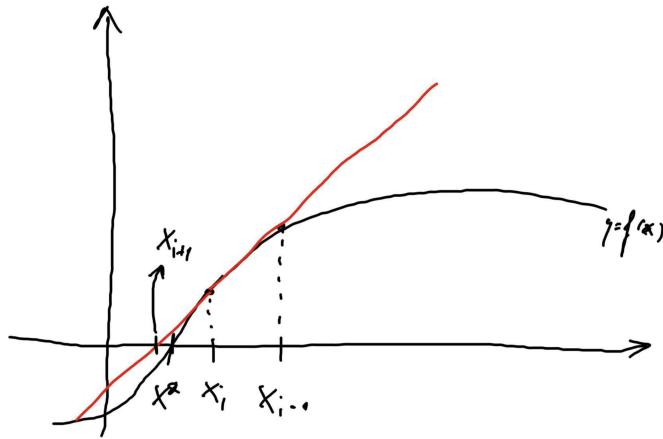


Figura 8: Metodo delle secanti (2.22). La linea rossa dovrebbe passare per x_{i+1} , x_i e x_{i-1} .

L'iterazione (2.20) diviene

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})} = \frac{f(x_i)x_{i-1} - f(x_{i-1})x_i}{f(x_i) - f(x_{i-1})}, \quad i = 1, 2, \dots, \quad (2.22)$$

con x_0 ed x_1 approssimazioni iniziali assegnate (sono richiesti per il calcolo di x_2). Assegnato x_0 è possibile calcolare x_1 , tramite il metodo di Newton, come $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$.

Costo computazionale:¹⁰⁷ L'approssimazione (2.22) richiede una valutazione funzionale per iterazione, in quanto $f(x_{i-1})$ è calcolato al passo $i-1$. La prima iterazione ne richiede due, ma sono trascurabili.

Convergenza:¹⁰⁸ locale. Inoltre, il metodo delle secanti è superlineare, ovvero: la successione delle approssimazioni converge alla radice della funzione con un tasso di convergenza maggiore rispetto al metodo delle tangenti e al metodo della bisezione, ma minore rispetto al metodo di Newton.

Ordine di convergenza: $p = \frac{\sqrt{5}+1}{2} \approx 1.618$ per radici semplici. Anche se l'ordine di convergenza è minore rispetto a quello del metodo di Newton è preferibile a quest'ultimo, in quanto non viene effettuata una valutazione della derivata ad ogni iterazione.

Osservazione 2.11. Nella conversione dei precenti metodi in algoritmi è buona norma eseguire controlli sul denominatore, affinché non siano svolte divisioni per 0. Nel caso dell'approssimazione di radici multiple, mediante il metodo di Newton modificato o il metodo di accelerazione Aitken, quando la derivata si annulla ed il valore della funzione è molto piccolo, significa che è stata raggiunta la migliore approssimazione possibile della soluzione.

¹⁰⁷Slide 9 PDF lez11.

¹⁰⁸Slide 10 PDF lez11.

Algoritmo 2.4 Implementazione del metodo delle secanti.

```
function x = secanti(f, f1, x0, tol, itmax)
fx = feval(f, x0);
f1x = feval(f1, x0);
x = x0 - fx/f1x;
for i = 1 : itmax
    if abs(x-x0) <= tol, break, end
    fx0 = fx;
    fx = feval(f, x);
    x1 = (fx*x0 - fx0*x)/(fx - fx0);
    x0 = x;
    x = x1;
end
if abs(x-x0) > tol, warning('metodo non converge'), end
```

3 Sistemi Lineari e Nonlineari

Il problema di questa sezione è risolvere sistemi di equazioni lineari. Tali sistemi sono della forma

$$A\underline{x} = \underline{b}, \quad (3.1)$$

dove $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ è la matrice dei coefficienti, $\underline{x} = (x_i) \in \mathbb{R}^n$ il vettore delle incognite (da determinare) e $\underline{b} \in \mathbb{R}^m$ il vettore dei termini noti. A e \underline{b} sono noti. Sarà assunto che $m \geq n$ e che il rango di A sia massimo, ovvero $\text{rank}(A) = n$.

Nel caso in cui $m > n$ allora sono presenti più equazioni che incognite ed in questo caso il sistema è detto **sistema sovradimensionato**. Questo caso sarà trattato nella Sezione 3.8 e nella 3.9. Inizialmente, fino alla Sezione 3.8, sarà considerato il caso in cui la matrice dei coefficienti è definita come $A \in \mathbb{R}^{n \times n}$, quindi è quadrata, ovvero $m = n$.

Definizione 3.1 (Matrice nonsingolare). La matrice $A \in \mathbb{R}^{n \times n}$, quadrata, è nonsingolare se $\text{rank}(A) = n$ (quindi $\det(A) \neq 0$) ed il sistema (3.1) ha soluzione unica del tipo

$$\underline{x} = A^{-1}\underline{b}. \quad (3.2)$$

La soluzione (3.2) è una risoluzione del problema che non definisce, in genere, un algoritmo efficiente di risoluzione dal punto di vista computazionale.

Osservazione 3.1. Con "costo computazionale" è da intendersi in termini di occupazione di memoria e numero di operazioni floating-point richieste.

3.1 Sistemi lineari: casi semplici

¹⁰⁹ La risoluzione del sistema lineare (3.1) risulta agevole nel caso in cui A abbia proprietà strutturali particolari, ovvero quando A è:

- diagonale;
- triangolare;
- ortogonale.

Queste caratteristiche saranno utilizzate per definire opportuni metodi di fattorizzazione che consentiranno di risolvere il problema generale.

3.1.1 Caso A diagonale

¹¹⁰ In questo caso gli elementi non appartenenti alla diagonale di A sono nulli, ovvero A è della forma

$$\begin{pmatrix} a_{11} & & \\ & \ddots & \\ & & a_{nn} \end{pmatrix},$$

dove gli elementi $a_{ii} \neq 0$, $a_{ij} = 0$, $j \neq i$, $i = 1, \dots, n$, sono gli elementi della **diagonale principale** di A . La matrice dei coefficienti si fatta può essere memorizzata come un singolo vettore.

¹⁰⁹Slide 5 PDF 1, PG 41.

¹¹⁰Slide 6 PDF 1, PG 42.

Osservazione 3.2. Per matrici grandi e con molti elementi nulli può essere utile l’Osservazione 3.34 e il successivo esempio.

Il determinante di A , se diagonale, è dato da $\det(A) = \prod_{i=1}^n a_{ii}$. Pertanto, assunto $\det(A) \neq 0$, allora segue che $a_{ii} \neq 0, \forall i = 1, \dots, n$. Inoltre, la forma component-wise del sistema lineare, data A diagonale, è del tipo

$$\begin{aligned} a_{11}x_1 &= b_1 \\ a_{22}x_2 &= b_2 \\ &\vdots \\ a_{nn}x_n &= b_n \end{aligned}$$

quindi la soluzione può essere calcolata come

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

Questo algoritmo è ben definito, in quanto $a_{ii} \neq 0, \forall i = 1, \dots, n$ per ipotesi.

La complessità di questo algoritmo è n flops e $O(n)$ memoria (si dice lineare perché richiede la memorizzazione di un vettore di lunghezza n).

Un’implementazione naïve di questo algoritmo è l’Algoritmo 3.1.

Algoritmo 3.1 Risoluzione sistema lineare bidiagonale.

```
function x = diag(a, b)
    x = b ./ a;
    return
end
```

3.1.2 Caso A triangolare

Questo caso è a sua volta diviso in due, in quanto è necessario specificare dove si trovano gli elementi significativi della matrice, ovvero:

- A è **triangolare inferiore** $\iff a_{ij} = 0, i < j$, ovvero la matrice è della forma

$$A = \begin{pmatrix} a_{11} & & \\ \vdots & \ddots & \\ a_{n1} & \dots & a_{nn} \end{pmatrix};$$

- A è **triangolare superiore** $\iff a_{ij} = 0, i > j$, ovvero la matrice è della forma

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ & \ddots & \vdots \\ & & a_{nn} \end{pmatrix}.$$

Per una generica matrice $A \in \mathbb{R}^{n \times n}$ triangolare (inferiore o superiore), è noto che $\det(A) = \prod_{i=1}^n a_{ii}$.

Osservazione 3.3.¹¹¹ Se A è contemporaneamente triangolare inferiore e superiore, allora è diagonale.

Nel caso in cui A è triangolare inferiore, il sistema lineare (3.1) è della forma

$$\begin{array}{lll} a_{11}x_1 & = b_1 \\ a_{21}x_1 + a_{22}x_2 & = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 & = b_3 \\ \vdots & \ddots & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n & = b_n \end{array}$$

e gli elementi della soluzione possono essere ottenuti mediante sostituzioni in avanti:

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j}{a_{ii}}, \quad i = 1, \dots, n. \quad (3.3)$$

Anche in questo caso l'algoritmo è ben definito sotto l'ipotesi $\det(A) \neq 0$ ($\iff a_{ii} \neq 0, i = 1, \dots, n$).

Esaminando il costo computazionale per risolvere il sistema lineare (3.1), mediante (3.3), allora: per calcolare il numeratore sono necessarie $i - 1$ moltiplicazioni, $i - 1$ somme ed 1 divisione finale (per calcolare x_i) per un totale di $2i - 1$ flops, quindi il costo totale è dato da

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

L'occupazione di memoria è dovuta alla porzione triangolare della matrice ed ai due vettori (\underline{x} e \underline{b}), quindi:

$$\sum_{i=1}^n i = \frac{n(n+1)}{2} + \overset{\text{vettori}}{2n} \approx \frac{n^2}{2} \text{ locazioni di memoria.}$$

È possibile codificare di soluzione al sistema triangolare inferiore (3.3) con l'Algoritmo 3.2. Per accedere alla matrice per colonne è sufficiente scambiare i cicli (vedere Algoritmo 3.3).

Il ciclo interno dell'Algoritmo 3.2 è un prodotto scalare (*scal*). Il ciclo interno dell'Algoritmo 3.3 è una *axpy* (ovvero un'operazione del tipo vettore = scalare \times vettore + vettore). Pertanto, è necessario osservare che il risultato di *scal* e *axpy* è simile ma non uguale, le operazioni sono diverse ed in aritmetica finita l'errore si propaga.

Nel caso in cui A sia triangolare superiore, ovvero

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} \\ & a_{22} & \dots & \dots & a_{2n} \\ & & \ddots & & \vdots \\ & & & a_{n-1,n-1} & a_{n-1,n} \\ & & & & a_{nn} \end{bmatrix},$$

con $\det(a) = \prod_{i=1}^n a_{ii} \neq 0 \iff a_{ii} \neq 0, i = 1, \dots, n$, il sistema di equazioni lineari (3.1) è della forma

$$\begin{array}{lll} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n & = b_1 \\ a_{22}x_2 + \dots + a_{2n}x_n & = b_2 \\ \vdots & \vdots & \vdots \\ a_{nn}x_n & = b_n \end{array}$$

¹¹¹Slide 9 PDF 1.

Algoritmo 3.2 Sistema triangolare inferiore.

```
% a <- matrice dei coefficienti
% b <- vettore dei termini noti
% x <- vettore soluzione
%
% x <- b rappresenta b_i in x_i
for i = 1 : n
    for j = 1 : i-1
        x(i) = x(i) - a(i,j) * x(j);
    end
    x(i) = x(i) / a(i,i);
end
```

Algoritmo 3.3 Sistema triangolare inferiore per colonne.

```
% x <- b
for j = 1 : n
    x(j) = x(j) / a(j,j);
    for i = j+1 : n
        x(i) = x(i) - a(i,j) * x(j);
    end
end
```

e gli elementi della soluzione possono essere ottenuti mediante sostituzioni in avanti:

$$x_i = \underbrace{b_i - \sum_{j=i+1}^{i-1} a_{ij}x_j}_{\begin{array}{l} 0 \text{ se } i=1 \\ a_{ii} \end{array}}, \quad i = n, n-1, \dots, 1. \quad (3.4)$$

La codifica di questo algoritmo (vedi Algoritmo 3.4) segue considerazioni simili a quelle viste per il caso triangolare inferiore riguardo ai costi.

Algoritmo 3.4 Sistema triangolare superiore.

```
x = b
for i = n : -1 : 1
    for j = i+1 : n
        x(i) = x(i) - a(i,j) * x(j); % scal (prodotto scalare)
    end
    x(i) = x(i) / a(i,i);
end
```

Algoritmo 3.5 Sistema triangolare superiore con accesso per colonne.

```
x = b
for j = n : -1 : 1
    x(j) = x(j) / a(j,j);
    for i = 1 : j-1
        x(i) = x(i) - a(i,j) * x(j);
    end
end
```

Nota sul ciclo interno dell'Algoritmo 3.4: ¹¹² Data (3.4), invece di calcolare prima la sommatoria in un ciclo, sottrarla a b_i e dividerla per a_{ii} , $b_i - \sum_{j=i+1}^{i-1} a_{ij}x_j$ è calcolata nel seguente modo: $x(i)$ è aggiornato con il valore precedente sottratto allo scalare $a(i,j) * x(j)$, dove, per $i = n$, inizialmente $x(i)$ contiene $b(i) (= b(n))$ con j che varia ed i che rimane fisso. Inoltre è un prodotto scalare.

Il relativo costo in termini di spazio e numero di operazioni è **identico** all'algoritmo del caso triangolare inferiore, ovvero rispettivamente $\frac{n^2}{2}$ e n^2 .

L'Algoritmo 3.4 accede agli elementi della matrice A **per righe**. È possibile ottenere la corrispondente versione che accede ad A **per colonne** scambiando l'ordine dei due cicli for (vedere Algoritmo 3.5).

Il ciclo interno dell'Algoritmo 3.5 è una *axpy*.

Osservazione 3.4. ¹¹³ Una *axpy* con vettori di lunghezza n ha lo stesso costo di una *scal* con vettori della stessa dimensione: **2n flops**, dove 2 sono il numero di operazioni per ogni ciclo ed n è il numero di cicli.

¹¹²Slide 3 PDF 2.

¹¹³Slide 4 PDF 2.

Pertanto, sono enunciate alcune importante proprietà (enunciati\lemmi) delle matrici triangolari, le quali saranno utilizzate (quindi è necessario conoscerle):¹¹⁴

1. ¹¹⁵ La somma di matrici triangolari inferiori (\superiori) è una matrice triangolare inferiore (\superiore);
2. Il prodotto di due matrici triangolari inferiori (\superiori) è una matrice triangolare inferiore (\superiore). Gli elementi diagonali del prodotto sono dati dal prodotto degli elementi diagonali omologhi dei due fattori;
3. ¹¹⁶ La matrice inversa di una matrice triangolare inferiore (\superiore) nonsingolare è triangolare inferiore (\superiore). I suoi elementi diagonali sono dati dai reciproci degli elementi diagonali omologhi;
4. ¹¹⁷ Il prodotto di due matrici triangolari inferiori (\superiori) a diagonale unitaria (ovvero con elementi diagonali tutti uguali ad 1) è una matrice triangolare inferiore (\superiore) a diagonale unitaria;¹¹⁸
5. La matrice inversa di una matrice triangolare inferiore (\superiore) a diagonale unitaria è una matrice triangolare inferiore (\superiore) a diagonale unitaria;

Dimostrazione della proprietà 2. ¹¹⁹ Sarà esaminato il caso triangolare inferiore, il caso triangolare superiore è analogo. Siano $A = (a_{ij})$, $B = (b_{ij}) \in \mathbb{R}^{n \times n}$, con $a_{ij} = 0 = b_{ij}$, $j > i$ (ovvero A e B sono triangolari inferiori) e sia $C = (c_{ij}) = A \cdot B$. È necessario dimostrare che, $\forall i = 1, \dots, n$:

- 1) $c_{ij} = 0$, $j > i$;
- 2) $c_{ii} = a_{ii} \cdot b_{ii}$, $i = j$.

Dalla 1) è ottenuto

$$c_{ij} \stackrel{120}{=} \underline{e}_i^T C \underline{e}_j = \underbrace{(\underline{e}_i^T A) \cdot (B \underline{e}_j)}_{121} = (\overbrace{\underbrace{a_{i1} \cdots a_{ii}}^i \underbrace{0 \cdots 0}_{n-i}})(\underbrace{0 \cdots 0}_{j-1} \underbrace{b_{jj} \cdots b_{nj}}_{n-j+1})^T \stackrel{122}{=} 0.$$

Dalla 2), ovvero il caso $i = j$, con argomenti analoghi è ottenuto

$$c_{ii} = (\overbrace{\underbrace{a_{i1} \cdots a_{ii}}^i \underbrace{0 \cdots 0}_{n-i}})(\underbrace{0 \cdots 0}_{i-1} \underbrace{b_{ii} \cdots b_{ni}}_{n-i+1})^T \stackrel{123}{=} a_{ii} \cdot b_{ii}.$$

□

¹¹⁴Slide 4 PDF 2

¹¹⁵Lemma 3.1 PG 45.

¹¹⁶Proprietà derivata dalla 2. Lemma 3.3 PG 45.

¹¹⁷Lemma 3.2 PG 45.

¹¹⁸Il motivo è dovuto al fatto che la diagonale sia 1 e che gli elementi della diagonale siano calcolati dal prodotto degli elementi omonimi delle diagonali delle matrici (i quali sono tutti 1).

¹¹⁹Slide 5 PDF 2.

¹²⁰Tramite il prodotto $e_i^T C$ è ottenuta l' i -esima riga di C . Tramite il prodotto $C \cdot e_j$ è ottenuta l' j -esima riga di C .

¹²¹Prodotto scalare tra riga i -esima di A e la riga j -esima di B .

¹²²Oppure $j - 1 \geq i$.

¹²³ ii perché le moltiplicazioni sono svolte tra a con indici precedenti ad ii sono per 0 (ovvero gli $(i-1) \cdot 0$), così anche per le b con indici maggiori di ii .

3.1.3 Caso A ortogonale

Definizione 3.2. ¹²⁴ A è ortogonale se $A^T A = AA^T = I$.

Pertanto, se A è ortogonale, allora $A^{-1} = A^T$. In questo caso, la soluzione a (3.1) è data da $\underline{x} = A^T \underline{b}$. Quindi il problema è risolto mediante un **prodotto matrice-vettore (matvec)**.

Divagazione sul costo ed accesso ai dati per l'esecuzione di una matvec. Supponendo di voler calcolare $\underline{y} = A\underline{x}$, dove $A \in \mathbb{R}^{m \times n}$, $\underline{x} \in \mathbb{R}^n$, $\underline{y} \in \mathbb{R}^m$ ¹²⁵ è possibile distinguere il caso in cui A è rappresentato per righe e per colonne.

- A per righe è della forma $(a_{ij}) = A = \begin{bmatrix} \underline{r}_1^T \\ \underline{r}_2^T \\ \vdots \\ \underline{r}_m^T \end{bmatrix}$, con $\underline{r}_i^T = (a_{i1}, \dots, a_{in})$ la i -esima riga di A , $i = 1, \dots, m$;
- A per colonne è della forma $A = [\underline{c}_1 \ \underline{c}_2 \ \cdots \ \underline{c}_n]$, con $\underline{c}_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix}$ la j -esima colonna di A , $j = 1, \dots, n$.

Siano $y_{i \in \{1, \dots, m\}}$ la i -esima componente di \underline{y} e con $x_{j \in \{1, \dots, n\}}$ la j -esima componente di \underline{x} , \underline{y} può essere calcolata come $y_i = \underline{r}_i^T \underline{x}$, $i = 1, \dots, m$. L'implementazione del calcolo di y_i è l'Algoritmo 3.6 (il quale ha un costo di **2mn flops**).

Algoritmo 3.6 $y_i = \underline{r}_i^T \underline{x}$ in pseudo-codice.

```

 $y \leftarrow 0$ 
for i = 1 : m do
  for j = 1 : n do
     $y(i) = y(i) + \underline{a}(i, j) * x(j)$  | scal
    126
  end for
end for

```

È possibile ottenere una corrispondente implementazione con accesso per colonna scambiando i cicli for e, quindi, svolgendo una *axpy*, osservando che: $\underline{y} = \underline{c}_1 x_1 + \dots + \underline{c}_n x_n = \sum_{j=1}^n c_j x_j$.

Metodi di fattorizzazione ¹²⁷ Nel caso in cui A sia una generica matrice nonsingolare allora è utile che questa sia fattorizzata nel prodotto di un numero, k (in genere $k = 1, 2$), di fattori $F_1, F_2, \dots, F_k \in \mathbb{R}^{n \times n}$ semplici. Questo significa che:

1. $A = F_1 \cdots F_k$;

¹²⁴Slide 6 PDF 2, PG 45.

¹²⁵Il caso in cui $m = n$ è un caso particolare di $\mathbb{R}^{m \times n}$.

¹²⁶Accesso per riga.

¹²⁷Slide 9-11 PDF 2, PG 46.

$$2. \quad F_i \begin{cases} \text{diagonale} \\ \text{triangolare} & \text{con } i = 1, \dots, k. \\ \text{ortogonale} \end{cases}$$

Pertanto, è possibile risolvere $A\underline{x} = \underline{b}$ è possibile risolvere $F_1 \cdots F_k \underline{x} = \underline{b}$. Questo equivale a risolvere, ponendo $\underline{x}_0 = \underline{b}$, i k sistemi lineari

$$F_i \underline{x}_i = \underline{x}_{i-1}, \quad i = 1, \dots, k,$$

quindi, $\underline{x}_k \equiv \underline{x}$.¹²⁸

Esempio 3.1. Con $k = 2$, $F_1 F_2 \underline{x} = \underline{b} \stackrel{129}{\Rightarrow} F_1 \underline{x}_1 = \underline{x}_0$, $F_2 \underline{x}_2 = \underline{x}_1$ ¹³⁰

I due sistemi lineari (in genere saranno k) sono di tipo semplice e quindi sono possono essere risolti agilmente.

I metodi che consistono nel determinare dei fattori che soddisfano i precedenti punti, 1. e 2., sono detti **metodi di fattorizzazione**.

Inoltre, è possibile osservare che non è necessario memorizzare tutte le soluzioni intermedie $x_1, \dots, x_k \equiv x$. Una volta calcolata la soluzione intermedia x_i , le precedenti non saranno utilizzate. Pertanto, è possibile utilizzare un unico vettore che, inizialmente, contiene il vettore dei termini noti e viene sovrascritto con le soluzioni intermedie, fino ad ottenere la soluzione finale del sistema lineare (3.1).

3.2 Fattorizzazione LU di una matrice

Definizione 3.3 (Matrice fattorizzabile LU).¹³² Data A , matrice dei coefficienti del sistema (3.1), questa è **fattorizzabile LU** rappresentabile nella forma

$$A = L \cdot U, \tag{3.5}$$

in cui:

- L è una matrice **triangolare inferiore a diagonale unitaria**, ovvero:¹³³

$$L = (l_{ij}), \quad l_{ij} = 0, \quad j > i, \quad \forall i, j = 1, \dots, n;$$

- U è **triangolare superiore**, ovvero $U = (u_{ij})$, $u_{ij} = 0$, $i > j$ (condizione restrittiva).

Osservazione 3.5.¹³⁴ Se A è fattorizzabile LU allora, risolvere il sistema lineare (3.1), significa risolvere

$$L \underbrace{U \underline{x}}_y = \underline{b}.$$

Il problema sarà risolto mediante la risoluzione, **nell'ordine**, dei seguenti sistemi lineari di tipo **semplice**:

$$Ly = \underline{b}, \quad U\underline{x} = y.$$

¹²⁸Soluzione equivalente ad x .

¹²⁹È necessario risolvere prima $F_1 x_1 = x_0$ e poi $F_2 \underline{x} = x_1$, altrimenti cambia il risultato.

¹³⁰Dal prodotto risulta un vettore con m righe. m è comune alla matrice F_2 ed al vettore \underline{x} , altrimenti non sarebbe possibile effettuare il prodotto.

¹³¹ $F_2 \underline{x} = \underline{x}_1$ e $b = \underline{x}_0$.

¹³²Slide 2 PDF 3, PG 47.

¹³³ A è a diagonale unaria se $\forall i = 1, \dots, n$, $a_{ii} = 1$.

¹³⁴Slide 3 PDF 3.

Teorema 3.1 (Unicità della fattorizzazione LU).¹³⁵ Se A è fattorizzabile LU (ed è nonsingolare), la fattorizzazione è unica.

*Dimostrazione.*¹³⁶ Supposto $A = L \cdot U = L_1 \cdot U_1$, con:

- L, L_1 triangolari inferiori a diagonale unitaria;
- U, U_1 triangolari superiori.

È necessario verificare che

$$L = L_1, \quad U = U_1.$$

Poiché

$$0 \neq \det(A) = \det(L \cdot U) = \det(L) \cdot \det(U) = \det(U),$$

è possibile affermare che U è nonsingolare. In modo analogo U_1 è nonsingolare.

Dall'uguaglianza $L \cdot U = L_1 \cdot U_1$ è ottenuto, moltiplicando a sinistra, membro a membro, per L_1^{-1} (a sinistra):

$$L_1^{-1}L \cdot U = \underbrace{L_1^{-1}L_1}_I \cdot U_1 = U_1 \longrightarrow L_1^{-1}L \cdot U = U_1.$$

Moltiplicando, da destra, membro a membro per U^{-1} è ottenuto $\mathbf{L}_1^{-1}\mathbf{L} = L_1^{-1}L \cdot \underbrace{U \cdot U^{-1}}_I = \mathbf{U}_1\mathbf{U}^{-1}$.

In conclusione:

$$\mathbf{L}_1^{-1}\mathbf{L} = \mathbf{U}_1\mathbf{U}^{-1}.$$

È possibile osservare che:

- **a secondo membro**, U è triangolare superiore, allora U^{-1} è triangolare superiore; poiché anche U_1 è triangolare superiore, è possibile concludere che $\mathbf{U}_1\mathbf{U}^{-1}$ è **triangolare superiore**;
- **a primo membro**, L_1 è triangolare inferiore a diagonale unitaria allora L_1^{-1} è triangolare inferiore a diagonale unitaria; poiché anche L è triangolare inferiore a diagonale unitaria $\mathbf{L}_1^{-1}\mathbf{L}$ è **triangolare inferiore a diagonale unitaria**.

Da questo è concluso che $L_1^{-1}L = I = U_1U^{-1}$, ovvero:

- $L_1^{-1}L = I \Rightarrow L = L_1$,
- $U_1U^{-1} = I \Rightarrow U_1 = U$.

Quindi è provata l'unicità della fattorizzazione di una matrice nonsingolare. \square

Rimane da stabilirne l'esistenza e per questo è definito quanto segue.

Osservazione 3.6. Affinché A sia fattorizzabile $LU \Rightarrow \det(A) \neq 0$.

¹³⁵Slide 3 PDF 3, Teorema 3.1 PG 47.

¹³⁶Slide 4 PDF 3, PG 47. È importante per successive trattazioni.

Al fine di definire un algoritmo per ottenere la fattorizzazione LU di A , è considerato il seguente problema: dato un vettore $\underline{v} \in \mathbb{R}^n$, è supposto di voler definire una matrice $L \in \mathbb{R}^{n \times n}$ **triangolare inferiore a diagonale unitaria** tale che:

$$L\underline{v} \equiv L \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = [\underbrace{\underline{v}_1, \dots, \underline{v}_k}_{k}, \underbrace{\underline{0}, \dots, \underline{0}}_{n-k}]^T, \quad (3.6)$$

dove v_i è la i -esima componente di \underline{v} e k è un indice fissato a priori. L deve azzerare in modo selettivo le componenti di \underline{v} , a partire dalla $(k+1)$ -esima in poi, lasciando inalterate le prime k componenti del vettore.

Supponendo $\underline{v}_k \neq \underline{0}$, ovvero l'ultima componente di \underline{v} che rimane inalterata, è definito il corrispondente **vettore elementare di Gauss**:

$$\underline{g}_k = \frac{1}{\underline{v}_k} [\underbrace{0 \cdots 0}_{k}, \underbrace{\underline{v}_{k+1} \cdots \underline{v}_n}_{n-k}]^T \quad (3.7)$$

¹³⁷

e la relativa **matrice elementare di Gauss**,

$$L = I - \underline{g}_k \underline{e}_k^T, \quad (3.8)$$

¹³⁸

dove \underline{e}_k è il k -esimo vettore della base canonica di \mathbb{R}^n definito come $\underline{e}_k = [0 \cdots 0 \ 1 \ 0 \cdots 0]^T$.

È necessario dimostrare che:

1. L è triangolare inferiore a diagonale unitaria;
2. L soddisfa (3.6).

Dimostrazione di 1. ¹³⁹ Da (3.8) è ottenuto

$$L = \begin{bmatrix} 1 & & & & \\ \ddots & 1 & & & \\ & & 1 & & \\ & & -\frac{v_{k+1}}{v_k} & \ddots & \\ & & \vdots & & \ddots \\ & & -\frac{v_n}{v_k} & & 1 \end{bmatrix}, \quad L\underline{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

che è triangolare inferiore con diagonale unitaria. □

¹³⁷Numero di componenti da azzerare.

¹³⁸ k -esimo elemento del vettore.

¹³⁹Slide 8 PDF 3, PG 48.

¹⁴⁰Sotto l'1 c'è il vettore \underline{g}_k .

$$\text{Dimostrazione della 2.}^{141} L\underline{v} = \left(I - \underline{g}_k \underline{e}_k^T \right) \underline{v} = \underline{v} - \underline{g}_k \underbrace{\left(\underline{e}_k^T \underline{v} \right)}_{v_k}^{142} = \underline{v} - \underline{g}_k v_k = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ v_{k+1} \\ \vdots \\ v_n \end{bmatrix}^{143} = \begin{bmatrix} v_1 \\ \vdots \\ v_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}^{144} \quad \square$$

Teorema 3.2. ¹⁴⁶ Sia L la matrice elementare di Gauss definita come (3.8), allora:

$$L^{-1} = I + \underset{147}{g}_k \underline{e}_k^T. \quad (3.9)$$

È possibile ora definire l'algoritmo di fattorizzazione LU . Questo algoritmo è **semi-iterativo**, ovvero ottiene la fattorizzazione, se esiste, in $n - 1$ passi. Sarà utilizzata la notazione $a_{ij}^{(k)}$ per denotare il passo più recente dell'algoritmo, il k -esimo, in cui l'elemento (i, j) di A è stato modificato.

Data la matrice di partenza

$$A = \begin{bmatrix} a_{11}^{(1)} & \cdots & a_{1n}^{(1)} \\ \vdots & & \vdots \\ a_{n1}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix} \equiv A^{(1)}, \quad (3.10)$$

al primo passo di fattorizzazione è necessario definire una matrice triangolare inferiore a diagonale unitaria, in modo tale che $L_1 A^{(1)} = A^{(2)}$ abbia la prima colonna strutturalmente simile a quella di una matrice triangolare superiore (ovvero è necessario azzerare gli elementi dal secondo in poi).

Se $a_{11}^{(1)} \neq 0$ (condizione necessaria e sufficiente), è possibile definire il primo vettore di Gauss,

$$\underline{g}_1 = \frac{1}{a_{11}^{(1)}} [0 \ a_{21}^{(1)} \cdots a_{n1}^{(1)}]^T \quad (3.11)$$

e la corrispondente prima matrice elementare di Gauss,

$$L_1 = I - \underline{g}_1 \underline{e}_1^T = \begin{bmatrix} 1 & & & \\ -\frac{a_{21}^{(1)}}{a_{11}^{(1)}} & 1 & & \\ \vdots & & \ddots & \\ -\frac{a_{n1}^{(1)}}{a_{11}^{(1)}} & & & 1 \end{bmatrix}, \quad (3.12)$$

tali che

$$L_1 A = \underset{148}{\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}} \equiv A^{(2)}.$$

¹⁴¹Slide 7 PDF 3, PG 48.

¹⁴²Sostituzione di L con $(I - \underline{g}_k \underline{e}_k^T)$.

¹⁴³Sostituzione di $\underline{e}_k^T \underline{v}$ con v_k .

¹⁴⁴È possibile la semplificazione di $\frac{1}{v_k} [0 \cdots 0 \ v_{k-1} \cdots v_n]^T$ con v_k .

¹⁴⁵Il vettore sottratto è \underline{g}_k .

¹⁴⁶Slide 8 PDF 3, PG 48.

¹⁴⁷Importante che sia $+$ per ottenere L^{-1} invertendo solamente il segno della parte inferiore della k -esima colonna.

Se $a_{22}^{(2)} \neq 0$ è possibile definire il secondo vettore elementare di Gauss,

$$\underline{g}_2 = \frac{1}{a_{22}^{(2)}} \left[\overbrace{0 \dots 0}^2 a_{32}^{(2)} \dots a_{n2}^{(2)} \right]^T, \quad (3.13)$$

e la corrispondente seconda matrice elementare di Gauss,

$$L_2 = I - \underline{g}_2 e_2^T = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & -\frac{a_{32}^{(2)}}{a_{22}^{(2)}} & \ddots & & \\ & \vdots & & \ddots & \\ & -\frac{a_{n2}^{(2)}}{a_{22}^{(2)}} & & & 1 \end{bmatrix},$$

tale che

$$L_2 L_1 A = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \dots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \dots & a_{nn}^{(3)} \end{pmatrix} \equiv A^{(3)}.$$

Procedendo in modo analogo, dopo un totale di $n - 1$ passi, sarà ottenuto

$$\underbrace{L_{n-1} \cdot L_{n-2} \cdot \dots \cdot L_1}_{L^{-1}} A = \begin{bmatrix} a_{11}^{(1)} & \dots & \dots & \dots & a_{1n}^{(1)} \\ a_{22}^{(2)} & \dots & \dots & \dots & a_{2n}^{(2)} \\ a_{33}^{(3)} & \dots & \dots & \dots & a_{3n}^{(3)} \\ \ddots & & & & \vdots \\ & & & & a_{nn}^{(n)} \end{bmatrix} \equiv A^{(n)} \equiv U. \quad (3.14)$$

Condizioni per l'algoritmo di fattorizzazione: L'algoritmo è utilizzabile (vedere il Teorema 3.3 definito in seguito) se, all' i -esimo passo,

$$\underline{a}_{ii}^{(i)} \neq 0, \quad i = 1, \dots, n - 1, \quad (3.15)$$

¹⁴⁸Gli elementi in grassetto sono modificati rispetto al passo precedente.
¹⁴⁹Elementi diagonali di U .

il che permetterà di costruire i corrispondenti vettori di Gauss,

$$\underline{g}_i = \frac{1}{a_{ii}^{(i)}} \left[\overbrace{0 \dots 0}^i a_{i+1,i}^{(i)} \dots a_{ni}^{(i)} \right]^T, \quad (3.16)$$

e la i -esima matrice elementare di Gauss,

$$L_i \stackrel{^{150}}{\equiv} I - \underline{g}_i \underline{e}_i^T = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & -\frac{a_{i+1,i}^{(i)}}{a_{ii}^{(i)}} & \ddots & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}, \quad (3.17)$$

tali che

$$L_i A^{(i)} \stackrel{^{151}}{=} L_i \cdots L_1 A = \begin{bmatrix} a_{11}^{(1)} & \dots & \dots & \dots & \dots & a_{1n}^{(1)} \\ 0 & \ddots & & & & \vdots \\ \vdots & \ddots & a_{ii}^{(i)} & \dots & \dots & a_{in}^{(i)} \\ \vdots & & 0 & a_{i+1,i+1}^{(i+1)} & \dots & a_{i+1,n}^{(i+1)} \\ \vdots & & & \vdots & & \vdots \\ 0 & \dots & 0 & a_{n,i+1}^{(i+1)} & \dots & a_{nn}^{(i+1)} \end{bmatrix} \equiv A^{(i+1)}, \quad i = 1, \dots, n-1. \quad (3.18)$$

Teorema 3.3. La procedura (3.14) è definita s.se $a_{ii}^{(i)} \neq 0$, $i = 1, \dots, n$.

Dimostrazione. Se A è nonsingolare, U deve essere nonsingolare e, quindi, è necessario che $a_{nn}^{(n)} \neq 0$, oltre alle condizioni (3.15). \square

Osservazione 3.7. Date le matrici elementari di Gauss L_1, \dots, L_{n-1} in (3.14), è possibile osservare che:

1. sono triangolari inferiori a diagonale unitaria;
2. il loro prodotto è ancora una matrice triangolare inferiore a diagonale unitaria;
3. la matrice inversa di questo prodotto è ancora una matrice triangolare inferiore a diagonale unitaria.

Inoltre, è possibile $L^{-1} = L_{n-1} \cdot L_{n-2} \cdots L_2 \cdot L_1$ da cui, in virtù della (3.14), è ottenuta la fattorizzazione $A = LU$.

È necessario trattare l'organizzazione dei dati della fattorizzazione LU :

- Il fattore U , nella sua porzione significativa, ovvero gli elementi della parte triangolare superiore, può essere memorizzato nella porzione triangolare superiore di A , come esposto in (3.14);
- Il fattore L può essere rappresentato come $L = (L_{n-1} \cdots L_1)^{-1} = L_1^{-1} \cdots L_{n-1}^{-1}$. Inoltre, se

$$L_i = I - \underline{g}_i \underline{e}_i^T \Rightarrow L_i^{-1} = I + \underline{g}_i \underline{e}_i^T,$$

¹⁵⁰Azzera gli elementi di $A^{(i)}$, in colonna i -esima, al di sotto dell'elemento diagonale.

¹⁵¹Questa moltiplicazione ha un costo di $2n^4$ flops perché entrambe le matrici hanno dimensione $n \times n$. La moltiplicazione $A\underline{v}$ ha un costo di $2n^2$ perché \underline{v} è un vettore colonna.

è possibile concludere, per un generico n , che

$$\begin{aligned} \mathbf{L} &= \stackrel{^{152}}{(I + \underline{g}_1 \underline{e}_1^T)(I + \underline{g}_2 \underline{e}_2^T) \cdots (I + \underline{g}_{n-1} \underline{e}_{n-1}^T)} = I + \underline{g}_1 \underline{e}_1^T + \cdots + \underline{g}_{n-1} \underline{e}_{n-1}^T \\ &= \begin{bmatrix} 1 & & & & \\ g_{21} & 1 & & & \\ \vdots & \ddots & \ddots & & \\ g_{n1} & \cdots & g_{n,n-1} & 1 \end{bmatrix} = \mathbf{I} + \sum_{i=1}^{n-1} \underline{g}_i \underline{e}_i^T. \end{aligned} \quad (3.19)$$

Esempio 3.2.¹⁵³ Per comprendere (3.19) è valutato il caso più semplice, ovvero con $n = 3$:

$$L = (I + \underline{g}_1 \underline{e}_1^T)(I + \underline{g}_2 \underline{e}_2^T) = \underbrace{I \cdot I}_{\underline{g}_1 \underline{e}_1^T} + \underbrace{I \cdot \underline{g}_1 \underline{e}_1^T + \underline{g}_2 \underline{e}_2^T}_{0} + \underline{g}_1 \underbrace{(\underline{e}_1^T \underline{g}_2) \underline{e}_2^T}_{0} = I + \underline{g}_1 \underline{e}_1^T + \underline{g}_2 \underline{e}_2^T = I + \sum_{i=2}^2 \underline{g}_i \underline{e}_i^T.$$

Pregi della fattorizzazione LU: Poiché al passo i -esimo è necessario azzerare gli elementi $a_{i+1,i}^{(i)}, \dots, a_{ni}^{(i)}$, ovvero la sezione triangolare inferiore in colonna i , allora è possibile sovrascriverli come $\frac{a_{i+1,i}^{(i)}}{a_{ii}^{(i)}}, \dots, \frac{a_{ni}^{(i)}}{a_{ii}^{(i)}}$, i quali sono gli elementi significativi del vettore \underline{g}_i . Questi ultimi costituiscono gli elementi significativi, in colonna i , del fattore L . In conclusione, A sarà riscritta con l'informazione relativa ai suoi fattori L ed U come:

$$\begin{bmatrix} a_{11}^{(1)} & \cdots & \cdots & \cdots & a_{1n}^{(1)} \\ g_{21} & a_{22}^{(2)} & \cdots & \cdots & a_{2n}^{(2)} \\ g_{31} & g_{32} & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ g_{n1} & \cdots & \cdots & g_{n,n-2} & a_{nn}^{(n)} \end{bmatrix},$$

dove la diagonale principale di L è "ignorata" perché nota. \square

Per trattare meglio l'esistenza della fattorizzazione LU , stabilita dal Teorema 3.3, occorre qualche definizione.

Definizione 3.4 (Sottomatrice principale di ordine).¹⁵⁴ Sia

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \in \mathbb{R}^{n \times n}. \quad (3.20)$$

È definita **sottomatrice principale di ordine k** di A :

$$\mathbf{A}_k = \begin{bmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} \end{bmatrix} \in \mathbb{R}^{k \times k}. \quad (3.21)$$

$\det(\mathbf{A}_k)$ è chiamato **minore principale di ordine k** .

¹⁵²L'ordine di moltiplicazione è importante.

¹⁵³Slide 5 PDF 4.

¹⁵⁴Slide 7 PDF 4, Definizione 3.1 PG 51.

Osservazione 3.8. ¹⁵⁵

1. A_k è l'intersezione delle prime k righe e k colonne di A ;
2. $A_1 = (a_{11})$, $A_n = A$.

Lemma 3.4. ¹⁵⁶ Sia $U = (u_{ij}) \in \mathbb{R}^{n \times n}$ triangolare superiore. Allora:

$$\det(U) \neq 0 \iff \det(U_k) \neq 0, \quad \forall k = 1, \dots, n.$$

Dimostrazione. \Leftarrow ovvia ($k = n$);

\Rightarrow

$$\det(U) \neq 0 \iff \prod_{i=1}^n u_{ii} \neq 0, \Rightarrow u_{ii} \neq 0, \forall i = 1, \dots, n \Rightarrow \prod_{i=1}^k u_{ii} \neq 0, \quad \forall k = 1, \dots, n.$$

Segue la tesi osservando che $\prod_{i=1}^k u_{ii} = \det(U_k)$. □

Il Lemma precedente stabilisce che una matrice triangolare superiore è nonsingolare s.se tutti i suoi minori principali sono nonnulli.

Finora è stato stabilito che se A è nonsingolare, allora

$$A = LU \iff \det(U) \neq 0 \iff \det(U_k) \neq 0, \quad \forall k = 1, \dots, n. \quad (3.22)$$

Lemma 3.5. ¹⁵⁷ Se A è nonsingolare ed $A = LU$, allora $\det(A_k) = \det(U_k)$, $\forall k = 1, \dots, n$.

Dimostrazione. Siano $A = LU \in \mathbb{R}^{n \times n}$, $I_k \in \mathbb{R}^{k \times k}$ e $O \in \mathbb{R}^{n-k \times k}$, segue che

$$A_k = [I_k \ O_{k,n-k}] A \begin{bmatrix} I_k \\ O_{n-k,k} \end{bmatrix} = \underbrace{[I_k \ O_{k,n-k}] L}_{158} \overbrace{U \begin{bmatrix} I_k \\ O_{n-k,k} \end{bmatrix}}^{159} = \underbrace{[L_k \ O_{k,n-k}]}_{160} \overbrace{\begin{bmatrix} U_k \\ O_{n-k,k} \end{bmatrix}}^{161} = L_k U_k + O_{k,k} \quad (3.23)$$

Pertanto, è stabilito che

$$A_k = L_k \cdot U_k, \quad \forall k = 1, \dots, n,$$

e da questo segue

$$\det(A_k) = \det(L_k U_k) = \overbrace{\det(L_k)}^1 \cdot \det(U_k) = \det(U_k), \quad \forall k = 1, \dots, n. \quad \square$$

Allora, unendo tutto, dalla (3.22) segue il Teorema di esistenza della fattorizzazione LU .

¹⁵⁵Slide 8 PDF 4, PG 52.

¹⁵⁶Slide 8, PDF 4, Lemma 3.5 PG 52.

¹⁵⁷Slide 9 PDF 4, Lemma 3.6 PG 52.

¹⁵⁸Prime k righe di L , con L triangolare inferiore.

¹⁵⁹Prime k colonne di U , con U triangolare superiore.

¹⁶⁰Poiché L è triangolare inferiore.

¹⁶¹Poiché U è triangolare superiore.

Teorema 3.6 (Esistenza della fattorizzazione della fattorizzazione LU).¹⁶² Sia $A \in \mathbb{R}^{n \times n}$ un matrice nonsingolare, allora:

$$A = LU \iff \det(A_k) \neq 0, \quad \forall k = 1, \dots, n.$$

Ovvvero: A è fattorizzabile LU s.se tutti i minori principali di A sono non nulli.

Osservazione 3.9. Il Teorema 3.6 è una proprietà restrittiva (A nonsingolare), la quale sarà ereditata dalle sottomatrici di ordine k di A . Nelle Sezioni 3.4 e 3.5 sarà visto come non è restitutiva per le classi trattate, per le quali:

- la nonsingolarità di A deriva da una sua specifica proprietà strutturale;
- la proprietà di essere fattorizzabile LU è goduta da tutte le sue sottomatrici principali, le quali sono nonsingolari.

3.3 Costo computazionale

È esaminato il costo computazionale in termine di operazioni algebriche (*flops*) e di occupazione di memoria del metodo i eliminazione di Gauss. Per questo (3.18) è ridefinito, esplicitando la struttura L_i , come:

$$A^{(i+1)} = L_i A^{(i)} = \left(I - \underline{g}_i \underline{e}_i^T \right) A^{(i)} = A^{(i)} - \frac{\underline{g}_i}{163} \left(\underline{e}_i^T A^{(i)} \right). \quad (3.24)$$

Tuttavia:

- le prime i componenti di \underline{g}_i sono nulle, quindi le prime i righe nella somma in (3.24) non sono modificate;
- $\underline{e}_i^T A^{(i)}$ è l' i -esima riga di $A^{(i)}$, la quale ha le prime $i - 1$ componenti nulle. Inoltre, in colonna i , dato che al passo successivo gli elementi sotto $a_{ij}^{(k)}$ sono 0, non è necessario svolgere operazioni perché è noto che queste avranno come risultato l'azzeramento degli elementi al di sotto di quello diagonale.

Pertanto, saranno processate solo le ultime $(n - i)$ righe e colonne di $A^{(i)}$, svolgendo 2 operazioni per componente, per un totale di $2(n - i)^2$ flops e, sommando gli $n - 1$ passi, è ottenuto:

$$2 \sum_{i=1}^{n-1} (n - i)^2 = 2 \sum_{i=1}^{n-1} i^2 \stackrel{165}{\approx} \frac{2}{3} n^3 \text{ flops.} \quad (3.25)$$

In termini di locazioni di memoria occupata, il metodo di eliminazione di Gauss non richiede memoria addizionale, in quanto la matrice A in ingresso sarà riscritta con le informazioni relative ai fattori L ed U .

È possibile verificare quanto specificato per il costo computazionale esaminando lo pseudo-codice dell'Algoritmo 3.7 (vedere l'Algoritmo 3.9 per controlli significativi).

La soluzione di un sistema lineare con $A = LU$ è implementata nell'Algoritmo 3.10.

¹⁶²Slide 10 PDF 4, Teorema 3.2 PG 52.

¹⁶³Vettore colonna.

¹⁶⁴Vettore riga ($i -$ esima di $A^{(i)}$).

¹⁶⁵Utilizzando la somma notevole $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6} \Rightarrow \sum_{i=1}^{n-1} i^2 = \frac{(n-1)(n-1+1)((2n-1)+1)}{6} = \frac{(n-1)n(2n-1)}{6}$

Algoritmo 3.7 Fattorizzazione LU di una matrice.

```
for i = 1 : n - 1 %passi della fattorizzazione
    if A(i,i) == 0, error('Matrice non fattorizzabile LU'); end
    A(i+1:n,i) = A(i+1:n, i)/A(i,i); %gi
    A(i+1:n,i+1:n) = A(i+1:n,i+1:n) - A(i+1:n,i) * A(i,i+1:n);
end
```

3.4 Matrici a diagonale dominante

Definizione 3.5 (Matrici diagonale dominante).¹⁶⁶ Sia $A = (a_{ij}) \in \mathbb{R}^{n \times n}$, questa è

- **diagonale dominante per righe**, se:

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|, \quad \forall i = 1, \dots, n; \quad (3.26)$$

- **diagonale dominante per colonne**, se:

$$|a_{ii}| > \sum_{k=1, k \neq i}^n |a_{ki}|, \quad \forall i = 1, \dots, n.$$

Esempio 3.3. •

$$A = \begin{bmatrix} -4 & 2 & 1 \\ 0 & -3 & 2 \\ 7 & -5 & 14 \end{bmatrix}$$

è diagonale dominante per righe e non è diagonale dominante per colonne;

-

$$A = \begin{bmatrix} -4 & 2 & 1 \\ 0 & -5 & 2 \\ 3 & 0 & -5 \end{bmatrix}$$

è sia diagonale dominante per righe che per colonne;

-

$$A = \begin{bmatrix} 7 & 6 & 1 \\ 0 & 6 & 1 \\ 7 & 7 & 6 \end{bmatrix}$$

non è diagonale dominante.

È necessario ricordare che la famiglia di matrici *d.d.* e *sdp* soddisfano le ipotesi del Teorema 3.6 e, pertanto, sono fattorizzabili LU .

Valgono le seguenti proprietà delle matrici diagonali dominanti.

¹⁶⁶Slide 3 PDF 5, Definizione 3.2 PG 55.

Teorema 3.7. ¹⁶⁷ A è d.d. per righe s.se A^T è d.d. per colonne. Rispettivamente per il caso opposto.

Teorema 3.8. ¹⁶⁸ A è d.d. per righe (\ colonne), s.se $\forall k = 1, \dots, n : A_k$ è d.d. per righe (\ colonne).

Dimostrazione. È trattato il caso per righe, il caso per colonne è analogo.

\Leftarrow ovvio (se $k = n \Rightarrow A_k = A$);

$\Rightarrow A$ d.d. per righe $\Leftrightarrow \forall i = 1, \dots, n : |a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}| \Rightarrow$ per un generico $k \in \{1, \dots, n\}$:

$$\forall i = 1, \dots, k : |a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}| \geq \sum_{j=1, j \neq i}^k |a_{ij}| \Leftrightarrow A_k \text{ è d.d. per righe.}$$

□

Se A è una matrice diagonale dominante allora è fattorizzabile LU , ovvero A d.d. $\Rightarrow \forall k = 1, \dots, n : A_k$ d.d. $\Rightarrow A = LU$ se vale il seguente risultato:

Teorema 3.9. ¹⁶⁹ A d.d. $\Rightarrow A$ nonsingolare.

Dimostrazione. È considerata A d.d. per righe, altrimenti è considerata A^T (Teorema 3.7). Per assurdo, inoltre, è assunto che A sia singolare $\Rightarrow \exists \underline{x} \in \mathbb{R}^n, \underline{x} \neq \underline{0} : A\underline{x} = \underline{0}$. Inoltre, poiché questa uguaglianza vale per ogni multiplo scalare di \underline{x} , è possibile normalizzare \underline{x} in modo tale che

$$\frac{1}{170} = \max_{i=1, \dots, n} |x_i| \equiv x_k, \quad (3.27)$$

con x_i la componente i -esima di \underline{x} . Allora, è esaminata la **k -esima** equazione del sistema $A\underline{x} = \underline{0}$. Formalmente, moltiplicando, membro a membro, per \underline{e}_k^T , il trasposto del k -esimo versore della base canonica $\underbrace{(\underline{e}_k^T A)}_{171} \underline{x} = \underline{e}_k^T \underline{0} = \underline{0}$.

È ottenuto $(a_{k1}, a_{k2}, \dots, a_{kn}) = \underline{0}$, ovvero,

$$\sum_{j=1}^n a_{kj} x_j \underset{172}{=} 0.$$

Segue (una somma equivalente alla precedente sommatoria)

$$a_{kk} x_k + \sum_{\substack{j=1 \\ \parallel}}^{n-1} a_{kj} x_j = 0,$$

da cui

$$a_{kk} = - \sum_{j=1, j \neq k}^n a_{kj} x_j.$$

Applicando i valori assoluti:

$$|a_{kk}| = \left| \sum_{j=1, j \neq k}^n a_{kj} x_j \right| \leq \sum_{j=1, j \neq k}^n |a_{kj}| \underbrace{|x_j|}_{\leq 1} \leq \sum_{j=1, j \neq k}^n |a_{kj}|.$$

¹⁶⁷Slide 4 PDF 5, Lemma 3.8 PG 55.

¹⁶⁸Slide 4 PDF 5, Lemma 3.7 PG 55.

¹⁶⁹Slide 5 PDF 5, Lemma 3.9 PG 55.

Questo contraddice l'ipotesi che A sia *d.d.* per righe. L'assurdo è aver assunto che A fosse singolare, è concluso che A deve essere nonsingolare. \square

Corollario 3.9.1. ¹⁷³ Se A è *d.d.* (per righe e/o colonne), allora $A = LU$.

Il Corollario 3.9.1 è derivato dai Teoremi 3.7, 3.8 e 3.9 ed è dimostrabile enunciando i teoremi stessi con le corrispondenti dimostrazioni.

Osservazione 3.10. Le matrici diagonali dominanti compaiono in importanti applicazioni.

3.5 Matrici Simmetriche e Definite Positive

¹⁷⁴

Definizione 3.6. Sia $A \in \mathbb{R}^{n \times n}$, questa è simmetrica e definita positiva (*sdp*) se:

- A è simmetrica $\iff A = A^T$;
- A è definita positiva $\iff \forall \underline{x} \in \mathbb{R}^n, \underline{x} \neq \underline{0} : \underbrace{\underline{x}^T A \underline{x}}_{175} > 0$.

Intermezzo:
$$\left[\begin{array}{l} A \text{ sdp} \iff A_k \text{ sdp}, \forall k = 1, \dots, n \\ \Downarrow \\ A \text{ nonsingolare} \end{array} \right] \stackrel{176}{\Rightarrow} A = LU$$

Teorema 3.10. $A \text{ sdp} \Rightarrow A$ nonsingolare.

Dimostrazione. Se, per assurdo, A fosse singolare, allora $\exists \underline{x} \in \mathbb{R}^n \setminus \{\underline{0}\} : A\underline{x} \neq \underline{0} \Rightarrow \underline{x}^T A \underline{x} = \underline{x}^T \underline{0} = 0$, il che contraddice il fatto che A sia definita positiva. Pertanto, A è non singolare. \square

Teorema 3.11. $A \in \mathbb{R}^{n \times n}$ è *sdp* $\iff \forall k = 1, \dots, n : A_k$ è *sdp*.

Dimostrazione. \Leftarrow ovvia (se $k = n \Rightarrow A_k = A$);

\Rightarrow Considerato un generico $k \in \{1, \dots, n\}$, allora:

$$A = \left[\begin{array}{c|c} A_k & B \\ \hline C & D \end{array} \right], \text{ con } A_k \in \mathbb{R}^{k \times k}, D \in \mathbb{R}^{(n-k) \times (n-k)}, \text{ da cui è evinto che } B \in \mathbb{R}^{k \times (n-k)} \text{ e } C \in \mathbb{R}^{(n-k) \times k}.$$

Poiché i blocchi diagonali sono quadrati, allora

$$A^T = \left[\begin{array}{c|c} A_k^T & C^T \\ \hline B^T & D^T \end{array} \right].$$

¹⁷¹ k -esima riga di A .

¹⁷² Per (3.27) $|x_j| \leq 1, \forall j = 1, \dots, n$.

¹⁷³ Slide 6 PDF 5, Teorema 3.3 PG 56

¹⁷⁴ Slide 7-12 PDF 5, 6.

¹⁷⁵ È uno scalare perché $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} \Rightarrow \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} (x_1, \dots, x_n) = \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n = \alpha > 0$.

¹⁷⁶ Da \Rightarrow e \iff .

Data la simmetria, allora $A = A^T$.

Uguagliando i blocchi omologhi è ottenuto che:

- $A_k = A_k^T \Rightarrow A_k$ è simmetrica; (punto dimostrato)
- $D = D^T$;
- $B = C^T$.

Rimane da dimostare che A_k è anche definita positiva, ovvero

$$\forall \underline{y} \in \mathbb{R}^k, \underline{y} \neq \underline{0} : \underline{y}^T A_k \underline{y} > 0.$$

A questo fine, considerando, dato un generico $\underline{y} \in \mathbb{R}^k$, $\underline{y} \neq \underline{0}$, il vettore a blocchi: $\underline{x} = \begin{pmatrix} \underline{y} \\ \underline{0} \end{pmatrix} \in \mathbb{R}^n$ ¹⁷⁷.

Segue che, poichè $\underline{y} \neq \underline{0} \Rightarrow \underline{x} \neq \underline{0}$. Pertanto, essendo A sdp:

$$0 < \underline{x}^T A \underline{x} = (\underline{y}^T, \underline{0}^T) \left(\left[\begin{array}{c|c} A_k & B \\ \hline C & D \end{array} \right] \begin{pmatrix} \underline{y} \\ \underline{0} \end{pmatrix} \right)^{178} = (\underline{y}^T, \underline{0}^T) \begin{pmatrix} A_k \underline{y} \\ C \underline{y} \end{pmatrix} = \underline{y}^T A_k \underline{y}.$$

Questo dimostra l'asserto. □

Osservazione 3.11. I Teoremi 3.11 e 3.10 permettono di affermare che una A sdp è fattorizzabile LU . Per dimostrare che ciò è vero è necessario enunciare le dimostrazioni di entrambi i Teoremi.

Quindi è possibile concludere che, se A è sdp, allora $A = LU$. La fattorizzazione LU non tiene di conto della proprietà di simmetria di A .

Al fine di individuare una fattorizzazione più conveniente per A sdp, sono enunciati i seguenti risultati.

Teorema 3.12. ¹⁷⁹ Sia $A = (a_{ij}) \in \mathbb{R}^{n \times n}$, una matrice sdp. Allora, $\forall i = 1, \dots, n : a_{ii} > 0$.

Dimostrazione. Dato $\underline{e}_i \in \mathbb{R}^{n \times n}$, l' i -esimo vettore della base canonica, è evidente che $\underline{e}_i \neq \underline{0}$. Inoltre, essendo A definita positiva: $a_{ii} = \underline{e}_i^T A \underline{e}_i > 0$. □

Un fatto derivante dal precedente Teorema è il seguente: data una matrice simmetrica, se sono presenti elementi diagonali nulli o negativi, allora la matrice non è definita positiva.

Teorema 3.13. ¹⁸⁰ A sdp s.se

$$A = LDL^T, \tag{3.28}$$

con

1. D diagonale ad elementi diagonali positivi;
2. L triangolare inferiore a diagonale unitaria.

¹⁷⁷ $\underline{0} \in \mathbb{R}^{n-k}$, dove k è la dimensione di \underline{y} , ed ha dimensione diversa da $\underline{0}$ di $\underline{y} \neq \underline{0}$ poco sopra.

¹⁷⁸ B e D non ci sono perché sono moltiplicati per il vettore nullo. $\left(\left[\begin{array}{c|c} A_k & B \\ \hline C & D \end{array} \right] \begin{pmatrix} \underline{y} \\ \underline{0} \end{pmatrix} \right) = \begin{pmatrix} A_k \underline{y} \\ 2C \underline{y} \end{pmatrix}$.

¹⁷⁹ Slide 10 PDF 5, Teorema 3.5 PG 57.

¹⁸⁰ Slide 10 PDF 5, Teorema 3.6 PG 57

Dimostrazione. \Leftarrow

$$A = LDL^T, D = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix}, d_i > 0, i = 1, \dots, n,$$

allora A è *sdp*. Infatti:

$$\mathbf{A}^T = (LDL^T)^T = LD^T L^T = LDL^T = \mathbf{A},$$

essendo D diagonale (dimostrando così che A è simmetrica).

È necessario dimostrare che, $\forall \underline{x} \in \mathbb{R}^n \setminus \{\underline{0}\}$:

$$\underline{x}^T A \underline{x} = \underbrace{(\underline{x}^T L)}_{\underline{y}^T} \underbrace{D \overbrace{(L^T \underline{x})}^{\underline{y}}}^{y} > 0.$$

Se $y = L^T \underline{x} \in \mathbb{R}^n$, allora $y \neq \underline{0}$ perché L^T è nonsingolare. Inoltre:

$$\underline{x}^T A \underline{x} = \underline{y}^T D \underline{y} = \sum_{i=1}^n d_i y_i^2,$$

essendo y_i la i -esima componente di \underline{y} .

Poiché $\underline{y} \neq \underline{0} \Rightarrow \exists k : y_k \neq 0$.

Pertanto,

$$\underline{x}^T A \underline{x} = \underline{y}^T D \underline{y} = \sum_{i=1}^n d_i y_i^2 \geq d_k y_k^2 > 0,$$

poiché $d_k > 0$ per ipotesi. Pertanto, A è definita positiva.

$\Rightarrow U$ può essere espressa nella forma

$$U = D \widehat{U},$$

con D diagonale (contenente, come elementi diagonali, gli elementi di U) ed \widehat{U} triangolare superiore a diagonale unitaria.

Pertanto, è ottenuto

$$A = LU = LD \widehat{U} \stackrel{181}{=} \widehat{U}^T (DL^T) = (LD \widehat{U})^T = A^T \quad (3.29)$$

dove:

- \widehat{U}^T triangolare inferiore a diagonale unitaria;
- DL^T triangolare superiore.

Quindi è stata ottenuta una riscrittura della fattorizzazione $A = LU$ e, per l'unicità di questa fattorizzazione, è ottenuto che

$$L = \widehat{U}^T \iff \widehat{U} = L^T.$$

¹⁸¹ A è simmetrica.

Sostituendo nella prima espressione (ovvero nella seconda metà di (3.29)) è ottenuto ciò che $A = LDL^T$ (dimostrando la parte iniziale del teorema).

Per completare l'asserto, rimane da dimostrare che (gli elementi diagonali di D , matrice simmetrica, siano positivi e non che la matrice sia diagonale perché lo è già per definizione):

$$\text{se } D = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \stackrel{182}{\Rightarrow} d_i > 0, \forall i = 1, \dots, n.$$

Fissato un generico $i \in \{1, \dots, n\}$, $\exists! \underline{x} \in \mathbb{R}^n : \mathbf{L}^T \underline{x} = \underline{e}_i$, l' i -esimo versore di \mathbb{R}^n . Essendo L nonsingolare, è evidente che $\underline{x} \neq \underline{0}$.

Pertanto, è ottenuto:

$$0 < \underline{x}^T A \underline{x} = \underline{x}^T L D L^T \underline{x} = (\mathbf{L}^T \underline{x})^T D (\mathbf{L}^T \underline{x}) = \underline{e}_i^T D \underline{e}_i = d_i. \stackrel{183}{=}$$

Questo conclude l'asserto. □

Osservazione 3.12. $\widehat{U}^T D L^T$ è una fattorizzazione LU di A .

Esempio 3.4.

$$U = \begin{bmatrix} 2 & 3 & 4 \\ 0 & 5 & 6 \\ 0 & 0 & 7 \end{bmatrix} = \underbrace{\begin{bmatrix} 2 & & \\ & 5 & \\ & & 7 \end{bmatrix}}_D \underbrace{\begin{bmatrix} 1 & 3/2 & 2 \\ 0 & 1 & 6/5 \\ 0 & 0 & 1 \end{bmatrix}}_{\widehat{U}}$$

Osservazione 3.13. ¹⁸⁴ Quanto esposto dal Teorema 3.13 permette di generare facilmente matrici *sdp*, considerando una matrice triangolare inferiore a diagonale unitaria casuale ed una matrice diagonale ad elementi casuali positivi. Con il prodotto LDL^T sarà ottenuta **sicuramente** una matrice *sdp*.

Il grande pregio della fattorizzazione LDL^T ((3.28)) consiste nell'ottenere un algoritmo di risoluzione per sistemi lineari più efficiente dal punto di vista computazionale, non è necessario il calcolo del fattore U , e dal punto di vista dell'occupazione di memoria rispetto alla fattorizzazione LU classica.

Per ottenere la fattorizzazione (3.28) sarà sufficiente uguagliare gli elementi delle due matrici ai due membri dell'uguaglianza, relativi ad una porzione **triangolare**, poiché gli altri derivano dalla simmetria di A .

Dal punto di vista di occupazione di memoria, una matrice *sdp* può essere memorizzata in forma compatta memorizzandone solo una parte triangolare, inferiore o superiore. Nello specifico, sarà considerata la porzione **triangolare inferiore**. Queste locazioni di memoria possono essere riscritte con la porzione strettamente triangolare di L (o L^T) e la diagonale D .

Sono derivate delle formule che permettono di ottenere in modo efficiente la fattorizzazione (3.28) uguagliando, se $A \in \mathbb{R}^{n \times n}$, gli elementi

$$(A)_{ij} = (LDL^T)_{ij}, \quad \left. \begin{array}{l} j = 1, \dots, n, \\ i = j, \dots, n. \end{array} \right\} \stackrel{185}{=}$$

¹⁸²Per il Teorema 3.12 (?).

¹⁸³Dovuto al fatto che A è definita positiva.

¹⁸⁴Slide 4 PDF 6.

A questo fine, denotando con a_{ij} l'elemento (i, j) di A ,

$$L = \begin{bmatrix} l_{11} & & & \\ \vdots & \ddots & & \\ l_{n1} & \dots & l_{nn} \end{bmatrix}, \quad l_{jj} = 1, \quad j = 1, \dots, n$$

e

$$D = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix}, \quad d_i > 0, \quad i = 1, \dots, n.$$

È ottenuto, per $j = 1, \dots, n$ e $i = j, \dots, n$:

$$\begin{aligned} \mathbf{a}_{ij} &= \underbrace{\underline{e}_i^T A \underline{e}_j}_{186} \stackrel{187}{=} \underline{e}_i^T L D L^T \underline{e}_j = (\underline{e}_i^T L) D (\underline{e}_j^T L)^T \\ &= (l_{i1} l_{i2} \dots l_{ii} \overbrace{0 \dots 0}^{n-i}) D (l_{j1} l_{j2} \dots l_{jj} \overbrace{0 \dots 0}^{n-j})^T = \sum_{k=1}^{\min\{i,j\}} l_{ik} d_k l_{jk} \stackrel{188}{=} \sum_{\mathbf{k}=1}^j l_{ik} l_{jk} d_k. \end{aligned}$$

Quindi $\mathbf{a}_{ij} = \sum_{\mathbf{k}=1}^j l_{ik} l_{jk} d_k$, $j = 1, \dots, n$, $i = j, \dots, n$.

È possibile distinguere i due casi $i = j$ e $i = j + 1, \dots, n$, con a_{jj} ottenuto tramite la precedente sommatoria:

$$\begin{aligned} d_j &\stackrel{189}{=} a_{jj} - \sum_{k=1}^{j-1} l_{jk} \overbrace{(l_{jk} d_k)}^{v_k} \\ l_{ij} &= \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} \overbrace{(l_{jk} d_k)}^{v_k}}{d_j}, \quad i = j + 1, \dots, n. \end{aligned} \tag{3.30}$$

I precedenti v_k non sono ricalcolati più volte perché sono lo stesso elemento, in sommatorie diverse.

Al fine di valutare il costo computazionale è implementato l'Algoritmo 3.8, il quale implementa il precedente algoritmo di fattorizzazione LDL^T .

Il costo principale dell'Algoritmo 3.8, per ogni iterazione, è il termine $A(j+1:n, 1:j-1)*v$ (ultima riga utile). Questo termine rende quadratico l'algoritmo, è un prodotto matrice-vettore, con una matrice $(n-j) \times (j-1)$ e rappresenta $\sum_{k=1}^{j-1} l_{ik} (l_{jk} d_k)$, $\forall i = j + 1, \dots, n$.

È possibile dimostrare che il costo dell'Algoritmo 3.8, considerando $2(nj - j^2 + j) \approx 2(nj - j^2)$ flops da sommare n volte, sia:

$$2 \sum_{j=1}^n (nj - j^2) = 2n \sum_{j=1}^n j - 2 \sum_{j=1}^n j^2 \approx 2n \frac{n^2}{2} - 2 \frac{2n^3}{3} = n^3 - \frac{2}{3}n^3 = \frac{1}{3}n^3 \text{ flops.} \tag{3.31}$$

Pertanto, il costo computazionale è dimezzato rispetto a quello della fattorizzazione LU classica (vedere (3.25)).

¹⁸⁵Scorrendo j sono scandite le colonne e fissato j A è scandito fino all'ultima riga.

¹⁸⁶ i -esima riga.

¹⁸⁷ i -esimo elemento della i -esima riga.

¹⁸⁸ $\min\{i, j\} = j$ perché i valori successivi sono nulli, quindi sono trascurati nei calcoli della sommatoria.

¹⁸⁹ d_k maggiore di 0, altrimenti la matrice non è sdp.

Algoritmo 3.8 Fattorizzazione LDL^T di una matrice (implementa le formule (3.30)).

```
%A e' la matrice n*n in ingresso. La parte triangolare inferiore di
%questa contiene l'informazione dei fattori L e D
for j = 1 : n %passi della fattorizzazione
    if j > 1
        v = (A(j, 1:j-1).*diag(A(1:j-1, 1:j-1)))';
        A(j,j) = A(j,j) - A(j, 1:j-1)*v; %d_j
    end
    if A(j,j) <= 0, error('A non sdp'), end
    A(j+1:n, j) = (A(j+1:n, j) - A(j+1:n, 1:j-1)*v)/A(j,j);
end
```

Osservazione 3.14. Applicando quanto scritto per ridurre la memoria occupata, l’Algoritmo 3.8 riscrive la porzione triangolare inferiore di A come segue:

$$A = \begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \Rightarrow \begin{bmatrix} d_1 & & & \\ l_{21} & d_2 & & \\ l_{31} & l_{32} & d_3 & \\ \vdots & \ddots & \ddots & \ddots \\ l_{n1} & \dots & \dots & l_{n,n-1} & d_n \end{bmatrix}, l_{jj} = 1, \quad j = 1, \dots, n.$$

Questa struttura dati è utilizzata per risolvere i sistemi lineari derivati dalla fattorizzazione, ovvero:

$$Ax = \underline{b} \stackrel{190}{\iff} \underbrace{\begin{matrix} \xrightarrow{x_1} \\ LD L^T \underline{x} \end{matrix}}_{191} = \underline{b} \stackrel{190}{\iff} \begin{matrix} \xrightarrow{x_2} \\ L\underline{x}_1 \end{matrix} = \underline{b} \stackrel{191}{\iff} \begin{matrix} \xrightarrow{x_2} \\ Dx_2 \end{matrix} = \underline{x}_1 \stackrel{192}{\iff} \begin{matrix} \xrightarrow{x_2} \\ L^T \underline{x} \end{matrix} = \underline{x}_2 \quad (3.32)$$

Per la soluzione dei sistemi lineari (3.32) è possibile utilizzare un unico vettore per il termine noto e le soluzioni intermedie (\underline{b}). In particolare, l’ultimo sistema lineare sarà risolto considerando che è nota L e non L^T .

Raffinatezza Algoritmo 3.9: Il fatto che A sia nonsingolare garantisce che, facendo il ciclo, anche l’ultimo elemento del fattore LU sia diverso da 0.

Errore da non fare nell’implementazione della fattorizzazione LU : calcolare il determinante della matrice da fattorizzare e controllare che questo sia diverso da 0. Matlab, per calcolare il determinante, utilizza la fattorizzazione LU e calcola il determinante come prodotto degli elementi sulla diagonale. Il controllo sul determinante costa quanto la fattorizzazione della *function*.

¹⁹⁰Equivale a risolvere ciò che segue (dall’alto verso il basso).

¹⁹¹Necessario il calcolo perché è noto L e non L^T .

¹⁹²Soluzione cercata.

Algoritmo 3.9 Fattorizzazione LU.

```
function LU = LUfatt(A)
% LU = LUfatt(A)
% Scrittura della matrice LU con l'informazioni dei fattori L e U di A,
% se e' fattorizzabile.
% Input:
%   A - matrice quadrata fattorizzabile;
% Output:
%   LU - matrice quadrata che contiene le informazioni sul fattore L (
%         parte triangolare inferiore) ed il fattore U (parte triangolare
%         superiore).
[m,n] = size(A);
if m ~= n, error('matrice non quadrata'), end
LU = A;
for i = 1: n-1
    if LU(i,i) == 0 %condizione necessaria per la fattorizzazione
        error('matrice non fattorizzabile LU')
    end
    LU(i+1:n, i) = LU(i+1:n, i) / LU(i,i); %vettore di Gauss
    LU(i+1:n, i+1:n) = LU(i+1:n, i+1:n) - LU(i+1:n, i) * LU(i, i+1:n);
end
if LU(n,n) == 0, error('matrice in input singolare'), end
return
%Il for riscrive gli elementi del vettore di Gauss.
```

Algoritmo 3.10 Risolutore sistema triangolare LU.

```
function x = LUsolve(LU, b)
% x=LUsolve(LU, b)
% solves the triangular linear systems stored in LU, for the right-hand
% -side b.
% Input:
%   LU - matrix created by LUfatt, containing the triangular factors;
%   b - right-hand-side of the linear system.
% Output:
%   x - solution vector.
% i primi due if conferiscono robustezza.
%
[m, n] = size(LU);
if m ~= n, error('matrice in ingresso non quadrata'), end
m = length(b);
if m ~= n, error('vettore in ingresso non compatibile con la matrice'),
    end
x = b(:); %trasforma b in vettore colonna
for i = 1 : n-1 % risoluzione fattore L
    x(i+1:n) = x(i+1:n) - LU(i+1:n,i)*x(i)
end
for i = n : -1 : 1
    if LU(i,i) == 0, error('matrice in input non valida'),
    else
        x(i) = x(i)/LU(i,i);
    end
    x(1:i-1) = x(1:i-1) - LU(1:i-1, i) * x(i);
end
return
```

Cose importanti sulla fattorizzazione LU :¹⁹³

- utilizzabile nel caso di matrici *d.d.*;
- nella variante LDL^T è applicabile al caso di matrici *sdp*;
- negli altri casi, ovvero il caso generale, è necessario parlare di Pivoting.

3.6 Pivoting

¹⁹⁴ È esaminato il caso in cui le ipotesi Teorema 3.6 non sono soddisfatte, anche se A è nonsingolare, quindi quando la fattorizzazione LU non esiste. È possibile modificare la fattorizzazione LU , in modo da definire una nuova matrice che soddisfi le ipotesi del Teorema 3.6.

Supposto di voler permutare le componenti del vettore \underline{x} come segue:

$$\underline{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \rightarrow \underline{y} = \begin{bmatrix} 3 \\ 4 \\ 1 \\ 2 \end{bmatrix}.$$

Per questo, se è considerato e_i , l' i -esimo versore di \mathbb{R}^n , allora: $e_i^T \underline{x} = i$, la i -esima componente di \underline{x} . Pertanto, per svolgere la precedente trasformazione è necessario definire la matrice P tale che $P\underline{x} = \underline{y}$, questa dovrà essere

$$P = \begin{bmatrix} e_3^T \\ e_4^T \\ e_1^T \\ e_2^T \end{bmatrix},$$

ovvero una matrice che contiene le righe dell'identità permutate. **P è una matrice di permutazione.**

Utilizzando **matrici di permutazione elementari**, ovvero matrici che scambiano solo due componenti tra loro, è possibile ottenere lo stesso risultato di $P\underline{x} = \underline{y}$, tramite 2 moltiplicazioni. Le matrici di permutazione elementari necessarie per lo scambio possono essere definite come

$$P_1 = \begin{bmatrix} e_3^T \\ e_2^T \\ e_1^T \\ e_4^T \end{bmatrix} \quad \text{e} \quad P_2 = \begin{bmatrix} e_1^T \\ e_4^T \\ e_3^T \\ e_2^T \end{bmatrix}.$$

Quindi $P = P_1 \cdot P_2 = P_2 \cdot P_1$ e $P_2 P_1 \underline{x} = \underline{y}$. Le matrici elementari P_1 e P_2 sono definite come:

$$P_1 = \begin{bmatrix} 0 & 0 & \boxed{1} & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ \boxed{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} \end{bmatrix}, \quad P_2 = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \boxed{1} \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & \boxed{1} & 0 & 0 \end{bmatrix}.$$

Gli 1 **cerchiati** sono elementi simmetrici, gli 1 in **grassetto** sono gli elementi dell'identità.
Le matrici di permutazione elementari P_1 e P_2 (in generale P_i) hanno le seguenti proprietà:

¹⁹³Slide 7 PDF 7.

¹⁹⁴Slide 7 PDF 7, PG 59-64.

- sono simmetriche;
- sono ortogonali:

$$\begin{aligned} P_1 &= P_1^T = P_1^{-1}, \\ P_2 &= P_2^T = P_2^{-1}. \end{aligned}$$

È possibile osservare che la matrice di permutazione $P = P_1 \cdot P_2$, non sarà più simmetrica. P sarà **ortogonale** sse $P^T P = I$: (ovvero se è verificato quanto segue)

$$P^T P = (P_2 \cdot P_1)^T P_2 P_1 = P_1^T \cdot P_2^T \cdot P_2 \cdot P_1 = P_1 \cdot \underbrace{P_2 \cdot P_2}_{I} \cdot P_1 = \underbrace{P_1 \cdot P_1}_{I} = I^{195}$$

In generale, ogni matrice di permutazione di permutazione P sarà decomponibile nel prodotto di matrici di permutazioni elementari (P_i) e risulta essere ortogonale (anche se può non essere unica):

$$P = P_k \cdot P_{k-1} \cdot \dots \cdot P_1 \Rightarrow P^{-1} = P^T = P_1 \cdot P_2 \cdot \dots \cdot P_k.$$

IMPORTANTE (il senso dell'argomento è il seguente:) il prodotto di più matrici di permutazione elementari da origine ad una matrice di permutazione, la quale è una matrice ortogonale.

Domanda: Cosa serve per tenere conto dell'informazione relativa ad una matrice P ,

$$P \begin{bmatrix} 1 \\ 2 \\ \vdots \\ n \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} = p,$$

permutazione del vettore iniziale? È sufficiente p . Infatti, se è permutato un generico vettore $\underline{x} \in \mathbb{R}^n$, il vettore permutato sarà $\underline{x}(p)$ (inteso come istruzione Matlab).

3.6.1 Fattorizzazione LU con pivoting parziale

¹⁹⁶ Questo metodo consente di risolvere un sistema lineare,

$$A\underline{x} = \underline{b}, \quad A \in \mathbb{R}^{n \times n}, \quad \det(A) \neq 0$$

con una propagazione dell'errore minore rispetto alla fattorizzazione LU classica.

È utilizzata la stessa notazione dell'algoritmo di eliminazione di Gauss, (3.10). Quindi, se A è nonsingolare, allora vi è sicuramente un elemento nonnullo nella sua prima colonna. Pertanto, è ricercato

$$\left| a_{k_1 1}^{(1)} \right| = \max_{k=1, \dots, n} \left| a_{k 1}^{(1)} \right| \stackrel{197}{>} 0,$$

dove $a_{k_1 1}^{(1)}$ è l'elemento di massimo modulo sulla prima colonna (in genere, sarà l'elemento di massimo modulo a partire dall'elemento diagonale).

¹⁹⁵Per la simmetria $P_1^T \cdot P_2^T = P_1 \cdot P_2$.

¹⁹⁶PDF 8, PG 60-64.

¹⁹⁷Se fosse nullo, A sarebbe singolare.

Osservazione 3.15. ($k =$) $k_1 \geq 1$.

Definendo la matrice di permutazione elementare, la quale è simmetrica e ortogonale, P_1 , che scambia gli elementi 1 e $k_1 (\geq 1)$ di un generico vettore di \mathbb{R}^n , è ottenuto che

$$P_1 A^{(1)} = P_1 A = \begin{bmatrix} a_{k_1 1}^{(1)} & \dots & \dots & a_{k_1 n}^{(1)} \\ a_{21}^{(1)} & \dots & \dots & a_{2n}^{(1)} \\ \vdots & & & \vdots \\ a_{11}^{(1)} & \dots & \dots & a_{1n}^{(1)} \\ \vdots & & & \vdots \\ a_{n1}^{(1)} & \dots & \dots & a_{nn}^{(1)} \end{bmatrix}^{198}.$$

È possibile definire il primo vettore elementare di Gauss (3.11) come

$$\underline{g}_1 = \frac{1}{a_{k_1 1}^{(1)}} [0 \ a_{21}^{(1)} \cdots a_{11}^{(1)} \cdots a_{n1}^{(1)}]^T$$

\uparrow
 k_1

Osservazione 3.16. Gli elementi di \underline{g}_1 sono uniformemente limitati, quindi hanno modulo minore uguale ad 1.

La prima matrice elementare di Gauss è definita come

$$L_1 = I - \underline{g}_1 \underline{e}_1^T,$$

tale che

$$L_1 P_1 A = \begin{bmatrix} a_{k_1 1}^{(1)} & a_{k_1 2}^{(1)} & \dots & a_{k_1 n}^{(1)} \\ \mathbf{0} & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & a_{n2}^{(2)} & \dots & a_{nn}^{(2)} \end{bmatrix} \equiv A^{(2)}.$$

È ricercato, in seconda colonna, a partire dall'elemento diagonale, l'elemento di massimo modulo:

$$\left| a_{k_2 2}^{(2)} \right| = \max_{k=2, \dots, n} \left| a_{k 2}^{(2)} \right|.$$

Svolto il prodotto $L_1 P_1 A$, non è più importante quale righe sono state scambiate perché dalla seconda alla n -esima sono state modificate.

Osservazione 3.17. 1. $k_2 \geq 2$;

2. se $\left| a_{k_2 2}^{(2)} \right| = 0 \Rightarrow A$ singolare.

¹⁹⁸Scambio riga k_1 -esima con riga 1.

La matrice di permutazione elementare P_2 , la quale permuta la riga 2 con la riga k_2 , nell'ordinamento naturale è definita come

$$P_2 L_1 P_1 A \stackrel{199}{=} \begin{bmatrix} a_{k_1 1}^{(1)} & a_{k_1 2}^{(1)} & \dots & a_{k_1 n}^{(1)} \\ \mathbf{0} & a_{k_2 2}^{(2)} & \dots & a_{k_2 n}^{(2)} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & a_{n2}^{(2)} & \dots & a_{nn}^{(2)} \end{bmatrix}.$$

È possibile definire il secondo vettore di Gauss (3.13) come

$$\underline{g}_2 = \frac{1}{a_{22}^{(2)}} [0 \ 0 \ a_{32}^{(2)} \ \dots \ a_{22}^{(2)} \ a_{n2}^{(2)}]^T, \quad \begin{matrix} \uparrow \\ k_2 \end{matrix}$$

il quale ha elementi di modulo ≤ 1 .

La seconda matrice elementare di Gauss è

$$L_2 = I - \underline{g}_2 e_2^T,$$

tale che:

$$L_2 P_2 L_1 P_1 A = \begin{bmatrix} a_{k_1 1}^{(1)} & a_{k_1 2}^{(1)} & a_{k_1 3}^{(1)} & \dots & a_{k_1 n}^{(1)} \\ \mathbf{0} & a_{k_2 2}^{(2)} & a_{k_2 3}^{(2)} & \dots & a_{k_2 n}^{(2)} \\ \mathbf{0} & \mathbf{0} & a_{33}^{(3)} & \dots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & a_{n3}^{(3)} & \dots & a_{nn}^{(3)} \end{bmatrix} \equiv A^{(3)}.$$

Ripetendo questa procedura fino al passo $(n-1)$ -esimo, sarà ottenuta la matrice

$$L_{n-1} P_{n-1} \cdots L_1 P_1 A = \begin{bmatrix} a_{k_1 1}^{(1)} & \dots & \dots & a_{k_1 n}^{(1)} \\ \mathbf{0} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & a_{k_n n}^{(n)} \end{bmatrix} \equiv A^{(n)} \equiv U, \quad (3.33)$$

in cui, $\forall i = 1, \dots, n-1 :$

$$\left| a_{k_i i}^{(i)} \right| = \max_{k=i, \dots, n} \left| a_{ki}^{(i)} \right|$$

dove

- $k_i \geq i$,
- se $\left| a_{k_i i}^{(i)} \right| = 0 \Rightarrow A$ è non singolare,

¹⁹⁹Scambio riga 2 con riga k_2 .

ed il vettore elementare i -esimo di Gauss definito come

$$\underline{g}_i = \frac{1}{a_{k_i i}^{(i)}} \left[\underbrace{0 \dots 0}_{\substack{\uparrow \\ k_i}} a_{i+1,1}^{(i)} \cdots a_{ii}^{(i)} \cdots a_{ni}^{(i)} \right]^T.$$

P_i è la matrice di permutazione elementare che permuta la riga i con la k_i , e

$$L_i = I - \underline{g}_i e_i^T \quad (3.34)$$

è la i -esima matrice elementare di Gauss che azzera gli elementi in colonna i , al di sotto dell'elemento diagonale.

Prima di affrontare ogni passo elementare è necessario moltiplicare per una corrispondente permutazione P_i .

Osservazione 3.18. Le prime $i - 1$ righe di P_i coincidono con quelle dell'identità.

Esempio 3.5. Per meglio comprendere il primo membro di (3.33) ($L_{n-1} P_{n-1} \cdots L_1 P_1$) è portato l'esempio $n = 4$:²⁰⁰

$$L_3 P_3 L_2 P_2 L_1 P_1 A = (L_3)(P_3 L_2 \underbrace{P_3}_{I})(P_3 P_2 L_1 P_2 P_3)(P_3 P_2 P_1) A = U.$$

Sono possibili le seguenti definizioni:

$$\begin{aligned} \widehat{L}_3 &= L_3 & \rightarrow & \text{matrice elementare di Gauss,} \\ \widehat{L}_2 &= P_3 L_2 P_3 & \rightarrow & \text{struttura analoga a } L_2, \\ \widehat{L}_1 &= P_3 P_2 L_1 P_2 P_3 & \rightarrow & \text{struttura analoga a } L_1, \\ P &= P_3 P_2 P_1 & \rightarrow & \text{è una matrice di permutazione.} \end{aligned}$$

È possibile riscrivere \widehat{L}_1 e \widehat{L}_2 affinché sia possibile notare che queste hanno struttura analoga a L_1 e L_2 :

$$\bullet \quad \widehat{L}_1 = P_3 P_2 L_1 P_2 P_3 = P_3 P_2 (I - \underline{g}_1 e_1^T) P_2 P_3 \stackrel{201}{=} I_{202} - \underbrace{(P_3 P_2 \underline{g}_1)}_{203} (\underbrace{e_1^T P_2 P_3}_{e_1^T}) = I - \widehat{g}_1 e_1^T,$$

dove $\widehat{g}_1 = P_3 P_2 \underline{g}_1$, ha la stessa struttura di \underline{g}_1 . Pertanto, anche \widehat{L}_1 e L_1 hanno la stessa struttura. Nel caso $n = 4$ è ottenuto

$$\underbrace{\widehat{L}_3 \cdot \widehat{L}_2 \cdot \widehat{L}_1}_{L^{-1}} \cdot P A = U \Rightarrow P \cdot A = LU.$$

$$\bullet \quad \widehat{L}_2 = P_3 L_2 P_3 = P_3 (I - \underline{g}_2 e_2^T) P_3 = I - (P_3 \underline{g}_2) (e_2^T P_3) = I - (P_3 \underline{g}_2) e_2^T \equiv I - \widehat{g}_2 e_2^T.$$

Poiché \widehat{g}_2 ha la stessa struttura di \underline{g}_2 (primi due elementi nulli), segue che L_2 e \widehat{L}_2 hanno la stessa struttura (triangolare inferiore a diagonale unaria ed elementi significativi in seconda colonna); \square

²⁰⁰Significa che le matrici considerate nell'esempio sono 4×4 .

²⁰¹Proprietà distributiva.

²⁰² $P_3 P_2 I P_2 P_3$.

²⁰³In genere alle g_i sono moltiplicati i P_{i+1} .

Il caso del precedente esempio può essere generalizzato. Infatti (3.33) può essere riscritta come

$$\widehat{L}_{n-1} \cdot \widehat{L}_{n-2} \cdot \dots \cdot \widehat{L}_1 \cdot P \cdot A = U, \quad (3.35)$$

dove

$$\begin{aligned}\widehat{L}_{n-1} &= L_{n-1}, \\ \widehat{L}_i &= P_{n-1} \cdot \dots \cdot P_{i+1} L_i P_{i+1} \cdot \dots \cdot P_{n-1}, \quad i = 1, \dots, n-2, \\ P &= P_{n-1} \cdot P_{n-2} \cdot \dots \cdot P_1,\end{aligned}$$

con \widehat{L}_i avente la stessa struttura della corrispondente matrice elementare di Gauss (3.17).

Quindi, ponendo

$$L^{-1} = \widehat{L}_{n-1} \cdot \dots \cdot \widehat{L}_1,$$

(3.35) può essere espressa come

$$L^{-1}PA = U.$$

Questo dimostra il seguente Teorema:

Teorema 3.14. ²⁰⁴Se A è nonsingolare, allora $\exists P$ matrice di permutazione, tale che:

$$P \cdot A = LU. \quad (3.36)$$

Il significato del Teorema è il seguente: se A non è fattorizzabile LU allora, per una permutazione, può diventarlo.

Osservazione 3.19. ²⁰⁵ Se è necessario risolvere il sistema lineare $A\underline{x} = \underline{b}$ allora può essere risolto al suo posto $\underbrace{PA\underline{x}}_{LU\underline{x}} = \underbrace{P\underline{b}}_{206}$.

Dato l'Algoritmo 3.11, allora:

- A è riscritta con l'informazione relativa ai suoi fattori L ed U ;
- il vettore p , di lunghezza n , conterrà tutte le informazioni della matrice di permutazioni P , ovvero le informazioni riguardo le permutazioni elementari svolte per ottenere la fattorizzazione (3.36);
- la porzione di codice aggiuntivo rispetto all'implementazione della fattorizzazione LU senza pivoting (Algoritmo 3.9) è dalla prima riga del for all'end del secondo if.

Pertanto, il costo in termini di operazioni algebriche elementari rimane

$$\approx \frac{2}{3}n^3 flops, \quad (3.37)$$

alle quali è necessario aggiungere $(n - 1)$ permutazioni di righe di A e $\approx \frac{1}{2}n^2$ confronti per l'individuazione del pivot.

²⁰⁴Slide 8 PDF 8, TH 3.7 PG 63. Sul libro il Teorema 3.14 è come segue: Se A è una matrice nonsingolare, allora esiste una matrice di permutazione P tale che PA è fattorizzabile LU .

²⁰⁵Slide 9 PDF 8, PG 63.

²⁰⁶In Matlab può essere scritto come $b(p)$, il quale è un vettore contenente la permutazione definita da P .

Algoritmo 3.11 Fattorizzazione LU con pivoting parziale di una matrice.

```
%A n*n matrice da fattorizzare
p = 1 : n;
for i = 1 : n - 1 %passi della fattorizzazione
    [mi, ki] = max(abs(A(i:n, i)));
    if mi == 0, error('matrice singolare'), end
    ki = ki + i - 1;
    if ki > i %controllo poco costoso che permette di risparmiare
        permutazioni inutili
        p([i, ki]) = p([ki, i]);
        A([i, ki], :) = A([ki, i], :); %Permutazione righe
    end
    if A(i,i) == 0, error('Matrice non fattorizzabile LU'), end
    A(i+1:n,i) = A(i+1:n, i)/A(i,i); %vettore di Gauss
    A(i+1:n,i+1:n) = A(i+1:n,i+1:n) - A(i+1:n,i) * A(i,i+1:n);
end
```

3.7 Condizionamento del problema

²⁰⁷ Considerato il sistema lineare

$$\begin{bmatrix} 1 & & & & & & & & & \\ 100 & 1 & & & & & & & & \\ & 100 & 1 & & & & & & & \\ & & 100 & 1 & & & & & & \\ & & & 100 & 1 & & & & & \\ & & & & 100 & 1 & & & & \\ & & & & & 100 & 1 & & & \\ & & & & & & 100 & 1 & & \\ & & & & & & & 100 & 1 & \\ & & & & & & & & 100 & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} = \boldsymbol{\alpha} \quad (3.38)$$

la sua soluzione è $x_i = \alpha$, $i = 1, \dots, 10$. Essendo bidiagonale inferiore, è possibile risolvere il sistema con

$$\begin{cases} x_1 = \boldsymbol{\alpha}, \\ x_i = 101 \cdot \boldsymbol{\alpha} - 100 \cdot x_{i-1}, \quad i = 2, \dots, 10, \end{cases}$$

e poiché il sistema lineare è

$$\begin{cases} x_1 = \boldsymbol{\alpha}, \\ x_i + 100 \cdot x_{i-1} = 101 \cdot \boldsymbol{\alpha}, \quad i = 2, \dots, 10, \end{cases}$$

è possibile codificare la soluzione del sistema con l'Algoritmo 3.12.

L'esecuzione dell'Algoritmo 3.12 restituisce i risultati di Tabella 1. Ciò che è evinto dalla precedente tabella è che il sistema lineare (3.1), la cui soluzione è rappresentata con i precedenti dati, è perturbato.

²⁰⁷PDF 9, Slide 2-4 PDF 10, PG 64-66.

Algoritmo 3.12 Risoluzione sistema bidiagonale

```

function x = bidia(alfa)
format long e
n = 10;
x = alfa*[1; 101*ones(n-1,1)];
for i = 2 : n
  x(i) = x(i) - 100 * x(i-1);
end
return
  
```

$x \setminus \alpha$	2.1	1	1.1	0.5	0.51	0.25	0.24
	2.100000000000000e+00	1	1.1	0.5	0.51	0.25	0.24
	2.10000000000023e+00	1	1.099999999999994	0.5	0.509999999999998	0.25	0.2399999999999984
	2.09999999997749e+00	1	1.100000000000577	0.5	0.510000000000197	0.25	0.2400000000001548
	2.100000000225123e+00	1	1.099999999942312	0.5	0.5099999999803018	0.25	0.2399999999845228
	2.09999977487755e+00	1	1.100000005768763	0.5	0.5100000019698214	0.25	0.2400000015477168
	2.100002251224510e+00	1	1.09999942312372	0.5	0.5099998030178554	0.25	0.239999845228315
	2.099774877549066e+00	1	1.100057687628052	0.5	0.5100196982144567	0.25	0.2400154771685017
	2.122512245093390e+00	1	1.094231237194819	0.5	0.5080301785543284	0.25	0.2384522831498295
	-1.512245093389311e-01	1	1.676876280518101	0.5	0.7069821445671565	0.25	0.3947716850170515
	2.272224509338931e+02	1	-56.58762805181011	0.5	-19.18821445671565	0.25	-15.23716850170515

Tabella 1: Risultati esecuzione Algoritmo (3.12)

È necessario studiare il **condizionamento di un sistema lineare** del tipo (3.1), con $m = n$ ed \underline{x} soluzione esatta. Se sono perturbati i dati in ingresso, A e \underline{b} , è studiato come questo si ripercuote sul risultato, ovvero:

$$(A + \Delta A)(\underline{x} + \Delta \underline{x}) = \underline{b} + \Delta \underline{b}, \quad (3.39)$$

in cui:

- ΔA è una matrice che contiene le perturbazioni degli elementi omologhi di A ;
- $\Delta \underline{b}$ è un vettore che contiene le perturbazioni degli elementi omologhi di \underline{b} ;
- $\Delta \underline{x}$ ha lo stesso ruolo per il vettore soluzione \underline{x} .

Al posto di (3.1) è considerato il problema (3.39). Per studiare (3.39) sarà trattato il calcolo del numero di condizionamento del problema, ovvero sarà studiato come le perturbazioni sui dati iniziali, ΔA e $\Delta \underline{b}$, determinano la perturbazione $\Delta \underline{x}$ sulla soluzione (3.2). Inoltre, per studiare (3.39) è necessario introdurre le **norme su vettori ed indotte su matrici** e quindi studiare le **Sezioni 3.10.1 e 3.10.2** (le quali possono essere argomento di esame).

In seguito sarà considerata una generica norma su vettore, e la corrispondente indotta su matrice, che indicheremo con $\|\cdot\|$. Per semplificare la trattazione è supposto che $\Delta A = \varepsilon \cdot F$, con $\varepsilon \in \mathbb{R}$, $\varepsilon \approx 0$ e $F \in \mathbb{R}^{n \times n}$. Similmente è posto $\Delta \underline{b} = \varepsilon \cdot \underline{f}$, con ε lo stesso parametro ed $\underline{f} \in \mathbb{R}^n$.

Quindi:

- $A(\varepsilon) = A + \varepsilon F \Rightarrow A(0) = A \wedge \dot{A}(\varepsilon) \equiv F$;
- $b(\varepsilon) = b + \varepsilon f \Rightarrow b(0) = b \wedge \dot{b}(\varepsilon) \equiv f$.

(3.39) può essere scritto come

$$A(\varepsilon)x(\varepsilon) = b(\varepsilon), \quad \varepsilon \approx 0, \quad (3.40)$$

dove $x(\varepsilon)$ denota la soluzione perturbata determinata dai dati $A(\varepsilon)$ e $b(\varepsilon)$.

Quindi:

- $x(0) = x$, la soluzione del sistema originario (non perturbato);
- $x(\varepsilon) = x(0) + \varepsilon \cdot \dot{x}(0) + O(\varepsilon^2)$. Quindi, per $\varepsilon \approx 0$, è possibile l'approssimazione

$$x(\varepsilon) \approx x + \varepsilon \cdot \dot{x}(0) \Rightarrow \Delta x \equiv x(\varepsilon) - x \underset{209}{\approx} \varepsilon \cdot \dot{x}(0).$$

Il problema da risolvere è trovare $\dot{x}(0)$. Per fare ciò è possibile osservare che (3.40) è valida identicamente in un intorno di $\varepsilon = 0$. Pertanto, da (3.40):

$$\frac{\partial}{\partial \varepsilon} (A(\varepsilon) \cdot x(\varepsilon)) = \dot{A}(\varepsilon)x(\varepsilon) + A(\varepsilon)\dot{x}(\varepsilon) \underset{F}{\approx} \frac{\partial}{\partial \varepsilon} b(\varepsilon) = f.$$

Quindi è ottenuto

$$F \cdot x(\varepsilon) + A(\varepsilon)\dot{x}(\varepsilon) = f.$$

²⁰⁸Significa che ε è sufficientemente piccolo per ciò che segue.

²⁰⁹Approssimazione al primo ordine.

Calcolando in $\varepsilon = 0$, ricordando che $x(0) = x$ e $A(0) = A$, è ottenuto:

$$F \cdot x + Ax(0) = f \Rightarrow \underset{210}{A} \dot{x}(0) = f - Fx \Rightarrow \dot{x}(0) = A^{-1}(f - Fx),$$

moltiplicando per ε membro a membro, è ottenuto:

$$\varepsilon \cdot \dot{x}(0) = A^{-1}(\varepsilon f - \varepsilon F \cdot x), \quad (3.41)$$

ovvero:

$$\Delta x \approx A^{-1} \left(\underbrace{\Delta b}_{\substack{\text{vett.} \\ \text{vett.}}} - \underbrace{\Delta A \cdot x}_{\substack{\text{vettore} \\ \text{matr.}}} \right).$$

Passando alle norme, è ottenuto (utilizzando la compatibilità tra norma su vettore e norma indotta su matrice):

$$\begin{aligned} \|\Delta x\| &\underset{211}{\approx} \|A^{-1}(\Delta b - \Delta A \cdot x)\| \leq \|A^{-1}\| \cdot \|\Delta b - \Delta A \cdot x\| \\ &\leq \|A^{-1}\| (\|\Delta b\| + \|\Delta A \cdot x\|) \leq \|A^{-1}\| (\|\Delta b\| + \|\Delta A\| \cdot \|x\|) \\ &\Rightarrow \frac{\|\Delta x\|}{\|x\|} \leq \|A^{-1}\| \left(\frac{\|\Delta b\|}{\|x\|} + \|\Delta A\| \right) \\ &= \|A\| \cdot \|A^{-1}\| \left(\frac{\|\Delta b\|}{\|A\| \cdot \|x\|} + \frac{\|\Delta A\|}{\|A\|} \right) \leq \|A\| \cdot \|A^{-1}\| \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right), \end{aligned}$$

pertanto:

- $\frac{\|\Delta x\|}{\|x\|}$ è una sorta di misura dell'errore relativo sul risultato;
- $\frac{\|\Delta b\|}{\|b\|}$, precisione macchina, e $\frac{\|\Delta A\|}{\|A\|}$ sono una sorta di misura relativa (rispettivamente di b ed A) sui dati in ingresso.

È possibile notare ciò che segue:

$$b = Ax \Rightarrow \|b\| = \|Ax\| \leq \|A\| \cdot \|x\| \Rightarrow \frac{1}{\|b\|} \geq \frac{1}{\|A\| \cdot \|x\|}.$$

Definizione 3.7. [Numero di condizionamento di una matrice] ²¹² $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ è denominato **numero di condizionamento** (o fattore di ampliamento) della matrice A .

Esempio 3.6. Data A in (3.38) allora $\begin{cases} \kappa(A) = 10^{200}, & \text{se misurato in } \|\cdot\|_1 \text{ o } \|\cdot\|_\infty, \\ \kappa(A) \approx 10^{200}, & \text{se misurato in } \|\cdot\|_2. \end{cases}$

Poiché $\kappa(A) \cdot u$, dove $u (\approx 10^{-16})$ è la precisione di macchina, è maggiore di 1, è necessario aspettarsi (in generale) risultati privi di senso nella risoluzione del sistema lineare.

Questo spiega ciò che è stato osservato nel caso in cui α non è un numero macchina. Se α è uguale 0.24 non è un numero di macchina, c'è un errore di rappresentazione sull'ultima cifra della sua mantissa. Questo errore viene propagato di un fattore 100 ad ogni iterazione (come si vede nel risultato in Tabella 1). 0,51, 3,9, 1,1, non sono numeri di macchina perché 0.1 è una cifra decimale in base 10 ma è un numero irrazionale se espresso in base 2 (è utilizzato un elaboratore con aritmetica finita).

²¹⁰ A nonsingolare, se fosse singolare allora il problema sarebbe mal posto.

²¹¹ Trasformazione del - della norma a sinistra del \leq in + a destra del \leq perché il - può essere rappresentato come (-1) e portato fuori dal modulo.

²¹² Slide 22 PDF 9, Definizione 3.4 PG 66.

Osservazione 3.20. ²¹³ Nei casi in cui è necessario risolvere $A\underline{x} = \underline{b}$, con $\kappa(A) \gg 1$, sono utilizzate **matrici di precondizionamento**, trasformando il sistema in

$$PA\underline{x} = P\underline{b}, \quad (3.42)$$

dove $P \stackrel{214}{\approx} A^{-1}$ e tale che $\kappa(PA) = O(1)$. Il sistema (3.42) è equivalente a quello di partenza.

Una proprietà importante del numero di condizionamento è la seguente:

Osservazione 3.21. Per ogni norma indotta su matrice, dalla Definizione 3.7, vale

$$1 = \|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = \kappa(A).$$

Il numero di condizionamento è legato alla dimensione della matrice (legge del numero di condizionamento). Se:

- $\kappa(A) \approx 1 \Rightarrow A$ è detta **ben condizionata**;
- $\kappa(A) \gg 1 \Rightarrow A$ è detta **mal condizionata**;
 - $\kappa(A) \approx u^{-1}$, se u è la precisione di macchina utilizzata. (Questo accade se il numero di cifre perse è tale per cui il risultato ottenuto è senza senso.)

Osservazione 3.22. ²¹⁶ In Matlab la *function cond* calcola il numero di condizionamento di una matrice. Se utilizzata con un solo parametro (ovvero solo la matrice) allora il risultato è ottenuto attraverso la norma 2, altrimenti è necessario specificare il secondo argomento.

IMPORTANTE: Se è richiesto, come esercizio, di costruire una matrice 2×2 con un numero di condizionamento pari a $\frac{1}{2}$ è necessario affermare che non si può fare per la precedente osservazione.

3.8 Sistemi lineari sovradimensionati

²¹⁷ È talvolta necessario risolvere un sistema di equazioni lineari sovradimensionato, ovvero con più equazioni che incognite, in cui la matrice dei coefficienti ha rango massimo, come specificato ad inizio del capitolo. In forma vettoriale, è necessario risolvere il seguente sistema lineare

$$A\underline{x} = \underline{b}, \quad A \in \mathbb{R}^{m \times n}, \quad m > n = \text{rank}(A), \quad (3.43)$$

con $\underline{b} \in \mathbb{R}^m$ e $\underline{x} \in \mathbb{R}^n$.

Un caso rilevante del problema (3.43) è trattato nella Sezione 4.10, dove, in genere, $m \gg n$.

²¹³Slide 3 PDF 10.

²¹⁴Approssimazione economica.

²¹⁵Significa che è del tipo 10^{15} .

²¹⁶Slide 3 PDF 10, Osservazione 3.4 PG 66.

²¹⁷Slide 3-10 PDF 10, PDF 11, Slide 2-4 PDF 12, PG 67-74.

²¹⁸ A ha rango massimo (n).

Il sistema lineare (3.43) ammette soluzione se $\underline{b} \in \text{ran}(A)$, dove $\text{ran}(A) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n, y = Ax\}$. È possibile osservare che il vettore $A\underline{x} \in \text{ran}(A)$: infatti, se

$$A = [\underline{c}_1, \dots, \underline{c}_n], \underline{c}_j \in \mathbb{R}^m, \underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

allora

$$A\underline{x} = \sum_{j=1}^n c_j x_j \wedge \text{ran}(A) = \{\underline{y} \in \mathbb{R}^m : \underline{y} = A\underline{x}, \underline{x} \in \mathbb{R}^n\}, \dim(\text{ran}(A)) = \text{rank}(A) = n. \quad ^{219}$$

$\underline{b} \in \mathbb{R}^m$, con $m > n$, è generico vettore e nel caso in cui $\underline{b} \in \text{ran}(A) \Rightarrow \exists! \underline{x} \in \mathbb{R}^n : A\underline{x} = \underline{b}$. L'unicità di \underline{x} deriva dal fatto che $\text{null}(A) = \{0\}$, in genere questo non avviene. Pertanto, non esiste, in genere una soluzione classica a (3.43).

Tuttavia, il sistema di risoluzione del sistema cambia per ogni vettore $\underline{x} \in \mathbb{R}^n$. Inoltre, è possibile definire il corrispondente **vettore residuo** \underline{r} come

$$\underline{r} = A\underline{x} - \underline{b} \in \mathbb{R}^m. \quad (3.44)$$

Osservazione 3.23. ²²⁰ Se \underline{x} è soluzione, in senso classico, di (3.43), allora $\underline{r} = \underline{0}$

Quando \underline{x} non è una soluzione in senso classico, allora è ricercato \underline{x} che rende \underline{r} il più piccolo possibile, cosicché $A\underline{x}$ sia il più vicino possibile a \underline{b} . Questo traduce nella ricerca della soluzione che minimizzi il residuo, ovvero:

$$\underline{x} \text{ t.c. } \|\underline{r}\|_2^2 = \min! \quad (3.45)$$

Osservazione 3.24. Se $\underline{r} = \begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix}$ è il vettore residuo, allora

$$\|\underline{r}\|_2^2 = \|A\underline{x} - \underline{b}\|_2^2 = \sum_{i=1}^m r_i^2, \quad ^{221} \quad (3.46)$$

ovvero una somma di quadrati.

Definizione 3.8 (Soluzione ai minimi quadrati). Il vettore \underline{x} soluzione di (3.45) è detto **soluzione ai minimi quadrati** (o nel senso dei minimi quadrati) del sistema lineare (3.43).

Per comprendere la scelta del nome "soluzione ai minimi quadrati" è necessario comprendere il motivo della scelta della **norma 2** (ovvero euclidea). Per farlo è possibile osservare che $\|\underline{r}\|_2^2 = \underline{r}^T \underline{r}$. Considerando una matrice **ortogonale** $U \in \mathbb{R}^{m \times m}$ (ovvero tale che $U^T U = I$), è noto che

$$\|U\underline{r}\|_2^2 = (U\underline{r})^T (U\underline{r}) = \underbrace{\underline{r}^T U^T U}_{I} \underline{r} = \underline{r}^T \underline{r} = \|\underline{r}\|_2^2.$$

È stato dimostrato il seguente risultato, il quale è il motivo per cui è scelta la norma euclidea.

²¹⁹Quindi $\text{ran}(A)$ è uno spazio vettoriale di dimensione n .

²²⁰Slide 5 PDF 10.

²²¹La radice di $\sqrt{\sum_{i=2}^m r_i^2}$ è annullata perché $\|r_i\|_2^2 = (\sqrt{\sum_{i=2}^m r_i^2})^2$.

Teorema 3.15.²²² La norma euclidea di un vettore è **invariante** rispetto alla moltiplicazione alla moltiplicazione per una matrice ortogonale.

Lo strumento principale per risolvere sistemi del tipo (3.43) è la fattorizzazione QR della matrice A , descritta dal seguente Teorema e dimostrata in seguito.

Teorema 3.16 (Esistenza della fattorizzazione QR).²²³ Data $A \in \mathbb{R}^{m \times n}$ (definita come in (3.43)) allora esistono

1. $Q \in \mathbb{R}^{m \times m}$, **ortogonale**;
2. $\widehat{R} \in \mathbb{R}^{n \times n}$, **triangolare superiore e nonsingolare**;

tale che, data $R = \begin{bmatrix} \widehat{R} \\ O \end{bmatrix} \in \mathbb{R}^{m \times n}$, da cui segue che $O \in \mathbb{R}^{(m-n) \times n}$, allora

$$A = QR. \quad (3.47)$$

\widehat{R} e Q sono nonsingolari, rispettivamente per specifica e per l'ortogonalità ($\det(Q) = 1 \vee \det(Q) = -1$). La matrice R è triangolare superiore.

Osservazione 3.25.²²⁴ Poiché Q è **nonsingolare**, allora:

$$n = \text{rank}(A) = \text{rank}(QR) = \text{rank}(R) = \text{rank}(\widehat{R}) \Rightarrow \det(\widehat{R}) \neq 0.$$

È possibile calcolare la soluzione ai minimi quadrati di (3.43), ovvero è cercata la soluzione definita dalla Definizione 3.8, tramite:

$$\begin{aligned} \|r\|_2^2 &\stackrel{225}{=} \|Ax - b\|_2^2 &\stackrel{226}{=} \|QR\underline{x} - b\|_2^2 &\stackrel{227}{=} \left\| Q(R\underline{x} - Q^T b) \right\|_2^2 \\ &\stackrel{228}{=} \|R\underline{x} - Q^T b\|_2^2 &\stackrel{229}{=} \|R\underline{x} - g\|_2^2 &= \left\| \begin{bmatrix} \widehat{R} \\ O \end{bmatrix} \underline{x} - \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \right\|_2^2 \\ &= \left\| \begin{bmatrix} \widehat{R}\underline{x} - g_1 \\ -g_2 \end{bmatrix} \right\|_2^2 &= \begin{bmatrix} \widehat{R}\underline{x} - g_1 \\ -g_2 \end{bmatrix}^T \begin{bmatrix} \widehat{R}\underline{x} - g_1 \\ -g_2 \end{bmatrix} &= (\widehat{R}\underline{x} - g_1)^T (\widehat{R}\underline{x} - g_1) + g_2^T g_2 \\ &\stackrel{230}{=} \left\| \widehat{R}\underline{x} - g_1 \right\|_2^2 + \|g_2\|_2^2 &\stackrel{231}{=} \|g_2\|_2^2 &= \min! \end{aligned} \quad (3.48)$$

Scegliendo \underline{x} come soluzione del sistema lineare,

$$\widehat{R}\underline{x} = g_1, \quad (3.49)$$

il quale ammette soluzione, e questa è unica, essendo \widehat{R} **nonsingolare**. Inoltre, il sistema lineare (3.49) è di **facile soluzione**, poiché \widehat{R} è **triangolare superiore**.

²²²Slide 7 PDF 10.

²²³Slide 7 PDF 10, Teorema 3.8 PG 68.

²²⁴Slide 7 PDF 10.

²²⁵Sostituzione (3.46).

²²⁶Sostituzione $A = QR$.

²²⁷Messa in evidenza di Q . Dato che b non è moltiplicato per Q allora è necessario immaginare che lo sia moltiplicata per QQ^T , ovvero la matrice identità.

²²⁸La norma euclidea è invariante per moltiplicazione per una matrice ortogonale.

²²⁹Ponendo $\underline{g} = Q^T b$, la quale è l'unica funzione di Q .

²³⁰ $\underline{g}_1 \in \mathbb{R}^n$, $\underline{g}_2 \in \mathbb{R}^{m-n}$.

²³¹ $(R\underline{x} - g_1)^T (\widehat{R}\underline{x} - g_1) = \|\widehat{R}\underline{x} - g_1\|_2^2$, $g_2^T g_2 = \|g_2\|_2^2$.

Teorema 3.17.²³² Se $A \in \mathbb{R}^{m \times n}$, $m > n = \text{rank}(A)$, la soluzione ai minimi quadrati (Definizione 3.8) del sistema lineare $A\underline{x} = \underline{b}$ (del tipo (3.43)) **esiste ed è unica**.

Dimostrazione. Il Teorema 3.16, l'Osservazione 3.25 e quanto scritto per (3.48)-(3.49) forniscono la dimostrazione del Teorema. \square

Osservazione 3.26. La norma del residuo, corrispondente alla soluzione ai minimi quadrati, corrisponde a quella del vettore \underline{g}_2 , definita da (3.48).

Esercizio potenziale: Calcolare la soluzione nel senso dei minimi quadrati e poi calcolare la norma del residuo. Per farlo è necessario restituire la norma del residuo e non il residuo stesso.

Quasi tutti gli studenti hanno calcolato correttamente la soluzione ai minimi quadrati risolvendo $\widehat{R}\underline{x} = \underline{g}_1$, calcolando successivamente la norma del residuo, tramite $A\underline{x} = \underline{b}$, per poi applicare la *function norm* (la quale ha un costo $2n \times n$) anche se la norma è già stata calcolata. Altri studenti, per calcolare la norma del residuo, hanno commesso l'orrore di assemblare la matrice Q (per calcolare il prodotto $Q^T \underline{b}$), arrivando a complessità quarta.

Svolgendo i passi per ottenere la soluzione è ottenuta anche la norma del residuo (ovvero le ultime $m - n$ componenti del vettore processato), con complessità meno che lineare ($m - n$).

3.8.1 Esistenza della fattorizzazione QR

²³³ Il Teorema 3.16 sarà dimostrato in modo costruttivo attraverso un algoritmo, il quale servirà a costruire quantitativamente la rappresentazione stessa.

Il fattore \widehat{R} utilizzato per definire la fattorizzazione QR (ovvero la (3.47)) è necessario per ottenere la soluzione del sistema lineare nel senso dei minimi quadrati.

Osservazione 3.27.²³⁴ Il fattore Q in (3.47) non è richiesto esplicitamente, è necessario solo per poter effettuare il prodotto $\underline{g} = Q^T \underline{b}$.

Al fine di definire l'algoritmo di fattorizzazione è considerato il seguente problema: dato un generico vettore $\underline{z} \in \mathbb{R}^n \setminus \{\underline{0}\}$, determinare una matrice ortogonale H tale che

$$H\underline{z} = \alpha \underline{e}_1, \quad (3.50)$$

con $\alpha \in \mathbb{R}$ ed $\underline{e}_1 \in \mathbb{R}^n$.²³⁵

Calcolando la norma euclidea dei vettori ad ambo i membri di (3.50) è ottenuto che:

$$\|H\underline{z}\|_2^2 = (H\underline{z})^T H\underline{z} = \underline{z}^T \underbrace{H^T H}_{I} \underline{z} = \underline{z}^T \underline{z} = \|\underline{z}\|_2^2 = \|\alpha \underline{e}_1\|_2^2 = \alpha^2 \underbrace{\|\underline{e}_1\|_2^2}_1 = \alpha^2.$$

Pertanto,

$$\alpha^2 = \|\underline{z}\|_2^2 \Rightarrow \alpha = \pm \|\underline{z}\|_2. \quad (3.51)$$

H è della seguente forma:

$$H = I - \frac{2}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T, \quad \underline{v} \in \mathbb{R}^n \setminus \{\underline{0}\}, \quad (3.52)$$

²³²Slide 9 PDF 10.

²³³PDF 11, PG 69-73

²³⁴Slide 2 PDF 11.

²³⁵Ciò che è ricercato è il vettore risultante dal prodotto del vettore \underline{z} e la matrice ortogonale H , in forma di multiplo del primo versore della base canonica (ovvero \underline{e}_1). Questa ricerca porta ad azzerare tutte le componenti della seconda colonna (in questo caso la prima componente non rimane uguale a quella del vettore originale).

dove v è scelto in modo da soddisfare (3.50) ($\|v\|_2 > 0$). H definita come (3.52) è **simmetrica** e **ortogonale**. Infatti:

$$H^T H = H^2 = \left(I - \frac{2}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T \right) \left(I - \frac{2}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T \right) \stackrel{238}{=} I - \underbrace{\frac{4}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T}_{239} + \underbrace{\frac{4}{(\underline{v}^T \underline{v})^2} \underline{v} (\underline{v}^T \underline{v}) \underline{v}^T}_{239} = I,$$

dimostrando che H ortogonale.

È necessario scegliere \underline{v} affinché (3.50) sia verificato ed a questo fine è scelto nella forma:

$$\underline{v} = \underline{z} - \alpha \underline{e}_1. \quad (3.53)$$

È possibile verificare che la scelta di \underline{v} come (3.53) assolve al compito richiesto, ovvero soddisfa (3.50), come segue:

$$\begin{aligned} H \underline{z} &\stackrel{(3.52)}{=} \left(I - \frac{2}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T \right) \underline{z} \stackrel{240}{=} \underline{z} - \frac{2}{\underline{v}^T \underline{v}} \widehat{\underline{v}^T \underline{z}} \underline{v} \\ &\stackrel{241}{=} \underline{z} - \frac{2}{\underline{v}^T \underline{v}} \underline{v}^T \underline{z} (\underline{z} - \alpha \underline{e}_1) \stackrel{242}{=} \left(1 - \frac{2}{\underline{v}^T \underline{v}} \underline{v}^T \underline{z} \right) \underline{z} + \alpha \underline{e}_1 \left(\frac{2}{\underline{v}^T \underline{v}} \underline{v}^T \underline{z} \right) = \alpha \underline{e}_1, \end{aligned}$$

nel caso fosse $\frac{2}{\underline{v}^T \underline{v}} \underline{v}^T \underline{z} = 1$, ovvero: $2 \underline{v}^T \underline{z} = \underline{v}^T \underline{v}$.

Quindi:

$$\bullet \underline{v}^T \underline{v} \stackrel{(3.53)}{=} (\underline{z} - \alpha \underline{e}_1)^T (\underline{z} - \alpha \underline{e}_1) = \underbrace{\underline{z}^T \underline{z}}_{243} - 2\alpha \underbrace{\underline{e}_1^T \underline{z}}_{244} + \alpha^2 (= \alpha^2 \cdot 2 \underline{e}_1^T + \alpha^2) = 2(\alpha^2 - \alpha z_1),$$

$$\bullet 2 \underline{v}^T \underline{z} \stackrel{(3.53)}{=} 2(\underline{z} - \alpha \underline{e}_1)^T \underline{z} = 2(\underline{z}^T \underline{z} - \alpha z_1) = 2(\alpha^2 - \alpha z_1) \text{ (ovvero quanto ottenuto al punto precedente).}$$

Inoltre, è possibile osservare che:

$$\underline{v} = \underline{z} - \alpha \underline{e}_1 = \begin{bmatrix} z_1 - \alpha \\ z_2 \\ \vdots \\ z_n \end{bmatrix},$$

pertanto, è svolta un'unica operazione sulla prima componente del vettore. Al fine di rendere questa operazione **ben condizionata**, è richiesto che z_1 e $-\alpha$, in prima riga di \underline{v} , siano concordi. Pertanto, $\alpha = -\text{sign}(z_1) \cdot \|\underline{z}\|_2$, dove

$$\text{sign}(z_1) = \begin{cases} 1, & \text{se } z_1 \geq 0, \\ -1, & \text{se } z_1 < 0. \end{cases}$$

²³⁶Perché H è una somma tra I ed una matrice di rango 1 ($\frac{2}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T$ è uno scalare).

²³⁷Proprietà non ovvia, necessita di essere descritta (dopo il punto).

²³⁸Proprietà distributiva.

²³⁹Somma tra $\left(-\frac{2}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T \right) I$ e $I \left(-\frac{2}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T \right)$.

²⁴⁰Scalare.

²⁴¹Sostituzione di \underline{v} con (3.53).

²⁴²Raccolto \underline{z} .

²⁴³ $\|\underline{z}\|_2^2 = \alpha^2$, quindi è svolta la somma con α^2 ed è raccolto un 2 (risultato $2(\alpha^2 - \alpha z_1)$).

²⁴⁴Prima componente di \underline{z} .

Definizione 3.9. Il vettore \underline{v} definito come (3.53) prende il nome di **vettore elementare di Householder**. La matrice H corrispondente, definita come (3.52), prende il nome di **matrice elementare di Householder**.

Il vettore \underline{v} scelto in forma (3.53), con α è definita come in (3.51). La scelta del segno di α è importante perché fa differenza. Tipicamente fa la differenza quando z_1 è l'elemento di massimo modulo e, di conseguenza, la norma di \underline{z} è grosso modo uguale al modulo di a_1 . In questo caso la sottrazione (3.53), con il segno di α scelto in modo opportuno, da origine alla cancellazione.

Il vettore \underline{v} e le matrici di Householder hanno un ruolo omologo a quello delle matrici triangolari di Gauss. La differenza sta nel fatto che le matrici di Gauss sono triangolari inferiori a diagonale unitaria mentre le matrici di Householder ortogonali.

N.B.: Per effettuare il prodotto $H\underline{z}$, il quale richiede $2n^2$ flops, se H fosse assemblata utilizzando (3.52), allora:

$$\left(I - \frac{2}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T \right) \underline{z} = \underline{z} - \frac{2 \underline{v}^T \underline{z}}{\underline{v}^T \underline{v}} \underline{v},$$

che ha un costo di $5n$ flops. Il costo è indicativo del fatto che le matrici di Householder non sono costruite in modo esplicito ed è sufficiente il vettore di Householder per svolgere il prodotto $\frac{2 \underline{v}^T \underline{z}}{\underline{v}^T \underline{v}} \underline{v}$.

Per memorizzare H sarà sufficiente memorizzare il vettore \underline{v} , ovvero n locazioni di memoria. La scelta di α , come già esposto, fa sì che $\underline{z} \neq \underline{0}$, quindi $\beta = z_1 - \alpha \neq 0$ (la quale è la componente di massimo modulo di \underline{v}). Pertanto, dividendo \underline{v} per β_1 sarà ottenuto che la prima componente del vettore $\frac{1}{\beta} \underline{v}$ sarà $\mathbf{1}$. In tal caso, dato che è noto che vale 1, è possibile non memorizzarlo portando ad $(n-1)$ il numero di locazioni di memoria necessarie.

Inoltre, la matrice di Householder definita dal vettore normalizzato $\frac{1}{\beta} \underline{v}$ è data da:

$$I - \frac{2 \beta^2}{\underline{v}^T \underline{v}} \frac{\underline{v} \underline{v}^T}{\beta^2} = I - \frac{2}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T \equiv H. \quad (3.54)$$

Questo significa che la matrice di Householder è **invariante** per uno scalamento del corrispondente vettore di Householder. Ciò significa che \underline{v} è memorizzato con un suo multiplo scalare qualunque. Sarà memorizzato $\frac{1}{\beta} \underline{v}$ normalizzato (il quale ha primo componente uguale ad 1), ottenendo la stessa matrice di Householder senza necessità di memorizzare la prima componente del vettore normalizzato.

L'equivalenza (3.54) è una proprietà utile al fine di ottimizzare il costo computazionale, in termini di occupazione di memoria, per ottenere la fattorizzazione (3.47).

È possibile definire l'algoritmo di fattorizzazione QR (di Householder). È utilizzata la notazione $a_{ij}^{(k)}$ per denotare l'elemento (i, j) della matrice al passo k , ovvero il più recente in cui l'elemento è stato modificato, come per il metodo di Gauss.

Dimostrazione del Teorema 3.16. ²⁴⁵ Data

$$A = \begin{bmatrix} a_{11}^{(0)} & \dots & a_{1n}^{(0)} \\ \vdots & & \vdots \\ a_{m1}^{(0)} & \dots & a_{mn}^{(0)} \end{bmatrix} \equiv A^{(0)},$$

²⁴⁵Slide 9 PDF 11. Ciò che sarà fatto è un processo semi-iterativo, con un numero minimo di passi (n se n è il numero di colonne della matrice). Sarà fatto ciò che è stato fatto con il metodo di Gauss, ovvero trasformare la matrice in una che, al passo i -esimo, ha la colonna i -esima di una matrice triangolare superiore (quindi saranno azzerati tutti gli elementi sotto la diagonale). Per fare questo saranno utilizzate le matrici ortogonali.

la prima colonna sarà un vettore nonnullo, altrimenti la matrice non avrebbe massimo rango.

La matrice elementare di Householder (ovvero il suo vettore di Householder normalizzato) è definita come H_1 tale che:

$$H_1 \cdot \begin{bmatrix} a_{11}^{(0)} \\ \vdots \\ a_{m1}^{(0)} \end{bmatrix} = \begin{bmatrix} a_{11}^{(1)} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad a_{11}^{(1)} \neq 0.$$

Senza $a_{11}^{(1)} \neq 0$ la prima colonna non sarebbe nulla. Considerando la matrice intera A , allora:

$$H_1 \cdot A \stackrel{246}{=} \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{m2}^{(1)} & \cdots & a_{mn}^{(1)} \end{bmatrix} \equiv A^{(1)}.$$

Considerando la sottomatrice di dimensione $(m - 1) \times (n - 1)$ in **grassetto** è visibile come la prima colonna della sottomatrice debba essere diversa da 0, poichè la matrice originaria ha rango massimo (altrimenti A non sarebbe fattorizzabile QR).

È possibile definire la matrice elementare di Householder, di dimensione $(m - 1) \times (m - 1)$, come $H^{(2)}$, tale che:

$$H^{(2)} \begin{bmatrix} a_{22}^{(1)} \\ \vdots \\ a_{m2}^{(1)} \end{bmatrix} = \begin{bmatrix} a_{22}^{(2)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{m-1}.$$

Definendo la matrice

$$H_2 = \left[\begin{array}{c|c} 1 & \underline{0}^T \\ \hline \underline{0} & H_2^{(2)} \end{array} \right] \in \mathbb{R}^{m \times m},$$

la quale è a sua volta simmetrica e ortogonale in quanto

$$H_2^T H_2 = H_2^2 = \left[\begin{array}{c|c} 1^2 & \underline{0}^T \\ \hline \underline{0} & (H^{(2)})^2 \end{array} \right] = H_2,$$

e, inoltre:

$$H_2 A^{(1)} = H_2 \cdot H_1 A = \begin{bmatrix} a_{11}^{(1)} & \cdots & \cdots & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{m3}^{(2)} & \cdots & a_{mn}^{(2)} \end{bmatrix} \equiv A^{(2)}, \quad a_{22}^{(2)} \neq 0.$$

²⁴⁶La prima colonna della matrice, gli elementi sotto $a_{11}^{(1)}$, può essere utilizzata per memorizzare le $n - 1$ componenti significative del vettore di Householder.

Procedendo in modo analogo, per altri $(n - 2)$ passi, sarà ottenuto:

$$\underbrace{H_n \cdot H_{n-1} \cdots \cdots H_1}_{Q^T} \cdot A = \begin{bmatrix} a_{11}^{(1)} & \dots & \dots & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & \dots & a_{2n}^{(2)} \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ & & & \ddots & a_{nn}^{(n)} \\ \vdots & & & & 0 \\ \vdots & & & & \vdots \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix} \stackrel{247}{\equiv} A^{(n)} \equiv R, \quad (3.55)$$

con

$$a_{jj}^{(j)} \neq 0, \quad j = 1, \dots, n \quad (3.56)$$

e

$$H_i = \left[\begin{array}{c|c} I_{i-1} & 0 \\ \hline 0 & H^{(i)} \end{array} \right], \quad H^{(i)} \in \mathbb{R}^{(m-i+1) \times (m-i+1)}.$$

□

Osservazione 3.28. $H^{(i)}$ è la matrice elementare di Householder che azzera gli elementi al di sotto di quello diagonale, in colonna i .

Osservazione 3.29. Dall'equazione (3.55) è possibile dedurre che $A = QR$.

Osservazione 3.30. ²⁴⁸ Verificare la condizione (3.56), ad ogni passo, garantisce che A abbia rango massimo.

Osservazione 3.31. Le prime n righe di (3.55) sono la matrice \widehat{R} del Teorema 3.16.

La dimostrazione al Teorema 3.16 descrive l'algoritmo di fattorizzazione QR di Householder. A termine della procedura di fattorizzazione gli elementi significativi di A (appartenente al sistema (3.43)) saranno sovrascritti con il fattore R , ovvero saranno sovrascritte le prime n righe con la porzione triangolare di $\widehat{R} \in \mathbb{R}^{n \times n}$. Le rimanenti componenti saranno, via via, utilizzate per memorizzare gli elementi significativi dei vettori di Householder normalizzati, in modo tale che la prima componente sia 1.

Il costo e l'implementazione dell'algoritmo è analizzato nella seguente Sezione.

3.8.2 Analisi complessità dell'algoritmo di fattorizzazione QR (Householder)

²⁴⁹ Per derivare il costo della fattorizzazione QR è esaminato lo pseudo-codice Matlab che implementa l'algoritmo in questione, ovvero l'Algoritmo 3.13. È assunto che $A \in \mathbb{R}^{m \times n}$, con $m > n$, la quale sarà sovrascritta con l'informazione della sua fattorizzazione QR .

Dato l'Algoritmo 3.13 è possibile affermare che il **costo dell'algoritmo di fattorizzazione QR** è il seguente:

$$\sum_{i=1}^n \underbrace{2 \times 2(m-i)(n-i)}_{250} \approx \frac{2}{3} n^2 (3m-n) \text{ flops.}$$

²⁴⁷Le prime n righe della matrice formano \widehat{R} .

²⁴⁸Slide 12 PDF 11.

²⁴⁹Slide 2-4 PDF 12, PG 73-74

Algoritmo 3.13 Fattorizzazione QR di Householder.

```

function A = QRfact(A)
[m,n] = size(A);
for i = 1 : n
    alfa = norm(A(i:m, i)); %norma porzione colonna i-esima a partire
    dalla diagonale
    if alfa == 0, error('matrice non ha rango massimo'), end
    if A(i,i)>= 0, alfa = -alfa; end
    v1 = A(i, i) - alfa; %prima componente del vettore di Householder
    A(i, i) = alfa;
    A(i+1:m, i)=A(i+1:m, i)/v1; %vettore normalizzato
    beta = -v1/alfa;
    A(i:m, i+1:n) = A(i:m, i+1:n) - (beta * [1; A(i+1:m, i)]) * ([1 A(i
    +1:m, i)']*A(i:m, i+1:n));
end

```

A riguardo l'Algoritmo 3.13 è possibile affermare quanto segue:

- è necessario decidere il segno di alfa in base alla prima componente del vettore, a partire dall'elemento diagonale fino alla fine. Se tale componente è maggiore uguale a 0 allora cambia segno;
- per l'ultima riga: è necessario calcolare il prodotto della matrice di Householder per la matrice A, dalla riga i alla m e dalla colonna i alla n . H è considerata nella versione $H = I - (\beta \hat{v}) \hat{v}^T$ per facilitare i calcoli. Questo perché assemblare una matrice di Householder ha costo cubico (in termini di flops), mentre svolgere i calcoli come nell'ultima riga utile a costo quadratico;
- la matrice A sarà trasformata in $H = I - (\beta \hat{v}) \hat{v}^T$ dove

$$\hat{v}_{251}^{252} = \frac{v}{v_1} = \frac{z - \alpha e_1}{v_1} = \frac{z - \alpha e_1}{z_1 - \alpha} \Rightarrow \beta = -\frac{z_1 - \alpha}{\alpha};$$

$$\beta = \frac{2}{\hat{v}^T \hat{v}} = \frac{2v_1^2}{v^T v} = \frac{2(z_1 - \alpha)^2}{(z - \alpha e_1)^T (z - \alpha e_1)} = \frac{2(z_1 - \alpha)^2}{||z||_2^2 + \alpha^2 - 2\alpha z_1} = \frac{2(z_1 - \alpha)^2}{2(\alpha^2 - \alpha z_1)} = \frac{2(z_1 - \alpha)^2}{2\alpha(\alpha - z_1)} = \frac{-\frac{1}{2}(z_1 - \alpha)^{\frac{1}{2}}}{\frac{1}{2}\alpha(z_1 - \alpha)} = \frac{(z_1 - \alpha)}{\alpha} = \frac{-v_1}{\alpha}$$

Osservazione 3.32.²⁵⁴ L'Algoritmo 3.13 può essere utilizzato anche nel caso di matrici quadrate: $m = n$. Infatti, in questo caso, dato che $m = n$ e $R = \hat{R}$, il costo è di $\frac{4}{3}n^3$ flops (+ spiccioli). Nel caso quadrato è preferibile utilizzare l'algoritmo di fattorizzazione LU con pivoting parziale, il quale ha un costo di $\frac{2}{3}n^3$ flops (vedere (3.37)).

²⁵⁰Data l'ultima riga dell'Algoritmo 3.13 allora è possibile affermare che il primo 2 sia il costo dovuto alla sottrazione, $(m - i)$ è costo di $A(i:m, i)$, quindi il $2(m - i)$ è dovuto al calcolo due volte e $2(m - i)(n - i)$ è il costo delle moltiplicazioni.

²⁵¹Vettore normalizzato.

²⁵²Sostituzione di $v = z - \alpha e_1$ e $\alpha^2 = ||z||_2^2$.

²⁵³Sostituzione di \hat{v} con $\frac{v}{v_1}$.

²⁵⁴Slide 4 PDF 12, Osservazione 3.10 PG 73.

3.9 Risoluzione di sistemi nonlineari

²⁵⁵ Per determinare la radice di un'equazione scalare, del tipo $f(x) = 0$, il metodo di Newton è uno dei metodi più efficaci. Questo metodo è formalmente definito dall'espressione funzionale (2.9), a partire da un'approssimazione x_0 e può essere esteso agevolmente anche al caso di **sistemi di equazioni nonlineari**. In questo caso, se

$$\underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n, \quad F : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad F(\underline{x}) = \begin{bmatrix} f_1(\underline{x}) \\ \vdots \\ f_n(\underline{x}) \end{bmatrix}, \quad (3.57)$$

con $f_i : \mathbb{R}^n \rightarrow R$, la i -esima funzione componente, il metodo di Newton diviene

$$\underline{x}_{n+1} = \underline{x}_n - \underbrace{(F'(\underline{x}_n))^{-1}}_{256} F(\underline{x}_n), \quad n = 0, 1, 2, \dots, \quad (3.58)$$

dove $F'(\underline{x}_n)$ è la matrice Jacobiana di $F(\underline{x})$ valutata in \underline{x}_n :

$$F'(\underline{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\underline{x}) & \dots & \frac{\partial f_1}{\partial x_n}(\underline{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\underline{x}) & \dots & \frac{\partial f_n}{\partial x_n}(\underline{x}) \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

Osservazione 3.33. ²⁵⁷ $(F'(\underline{x}))_{ij} = \frac{\partial f_i}{\partial x_j}(\underline{x}), \quad i, j = 1, \dots, n$.

Esempio 3.7. ²⁵⁸ Se $n = 2$ e $F(x_1, x_2) = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} \cos(x_1) + e^{x_2} \\ \sin(x_1) + x_2 \end{bmatrix} \Rightarrow F'(x_1, x_2) = \begin{bmatrix} -\sin(x_1) & e^{x_2} \\ \cos(x_1) & 1 \end{bmatrix}$.

Tuttavia, l'equazione (3.58) è equivalente, moltiplicando membro a membro per $F'(\underline{x}_n)$, a:

$$F'(\underline{x}_n) \underline{x}_{n+1} = F'(\underline{x}_n) \underline{x}_n - F(\underline{x}_n),$$

ovvero, ponendo $\Delta \underline{x}_n = \underline{x}_{n+1} - \underline{x}_n$,

$$\begin{cases} F'(\underline{x}_n) \Delta \underline{x}_n = -F(\underline{x}_n), \\ \underline{x}_{n+1} = \underline{x}_n + \Delta \underline{x}_n, \end{cases} \quad n = 0, 1, \dots \quad (3.59)$$

dove $F'(\underline{x}_n)$ è una matrice conosciuta ed $\Delta \underline{x}_n$ è un vettore sconosciuto. Pertanto, (3.59) significa (risolvere un sistema lineare) che:

²⁵⁹

Per $n = 0, 1, 2, \dots$ è risolto, tramite il metodo di Newton, il sistema lineare $F'(\underline{x}_n) \Delta \underline{x}_n = -F(\underline{x}_n)$, aggiornando $\underline{x}_{n+1} = \underline{x}_n + \Delta \underline{x}_n$.

È necessario definire un criterio d'arresto. Innanzitutto, è trasformato un ciclo condizionale ($n = 0, 1, 2, \dots$) in uno con iterazioni massime prefissate. Se un criterio è soddisfatto, il ciclo è terminato anzitempo.

²⁵⁵ Slide 4-8 PDF 12, PG 74-76.

²⁵⁶ Calcolare l'inversa di una matrice è molto costoso al livello computazionale.

²⁵⁷ Slide 5 PDF 12.

²⁵⁸ Slide 6 PDF 12.

²⁵⁹ Rappresenta un ciclo e quindi è necessario un criterio d'arresto.

Esempio criterio d'arresto: il numero massimo di iterazioni può essere legato alla dimensione n del problema e all'accuratezza tol richiesta. Il problema della scelta dell'accuratezza è concreto e dipende dal problema affrontato: misurare la traiettoria di un asteroide richiede una tolleranza diversa rispetto a quella di un intervento chirurgico.

Criterio d'arresto: può essere generalizzato dal caso scalare, ovvero (2.15), con:

$$\|\Delta x \cdot (1 + |x|)\|_\infty \leq tol. \quad (3.60)$$

Costo computazionale: Il costo per iterazione di (3.59) ammonta a:

1. 1 valutazione funzionale di $F(\underline{x})$;
2. **1 valutazione funzionale di $F'(\underline{x})$ (!!!);**
3. **1 fattorizzazione di $F'(\underline{x})$;**
4. 2 risoluzioni con i fattori + 1 aggiornamento.

I punti 2. e 3. sono i più onerosi dell'algoritmo. Per semplificare il costo è possibile fare ricorso al metodo delle corde (o Newton semplificato), la cui iterazione è:

$$\begin{cases} F'(\underline{x}_0) \Delta \underline{x}_n = -F(\underline{x}_n) \\ \underline{x}_{n+1} = \underline{x}_n + \Delta \underline{x}_n, \quad n = 0, 1, 2, \dots \end{cases}$$

In questo caso, i costi dei punti 2. e 3. precedentemente esposti, sono sostenuti una sola volta, prima di iniziare l'iterazione. Per questo motivo, questa iterazione è molto utilizzata nelle applicazioni.

3.10 Intermezzi

Osservazione 3.34. ²⁶¹ Data

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad (3.61)$$

allora $(a_{11}, a_{22}, a_{33}, a_{44})$ è la diagonale principale, (a_{12}, a_{23}, a_{34}) è la prima sopradiagonale, (a_{21}, a_{32}, a_{43}) è la prima sottodiagonale, (a_{13}, a_{24}) è la seconda sopradiagonale, (a_{31}, a_{42}) è la seconda sottodiagonale, (a_{14}) è la terza sopradiagonale e (a_{41}) è la terza sottodiagonale.

Esempio 3.8. Dato un generico elemento $a_{ij} \in A$, si verificano le seguenti differenze:

$$j - i = \begin{cases} k, & \text{sulla } k\text{-esima sopradiagonale;} \\ 0, & \text{sulla diagonale principale;} \\ -k, & \text{sulla } k\text{-esima sottodiagonale;} \end{cases}$$

²⁶⁰Calcolata solo una volta, ovvero nell'approssimazione iniziale x_0 .

²⁶¹Slide 6 PDF 1. Questa osservazione serve per le matrici grandi e quindi con molti elementi nulli. Gli elementi diversi da 0 sono sulla diagonale. Per salvare qual è la diagonale corretta, ovvero quella sulla quale sono presenti i numeri, è possibile utilizzare gli indici $j - i$ per salvare l'indice della diagonale: in questo caso è necessario un offset per individuare qual è la diagonale giusta.

3.10.1 Norme su vettori

Definizione 3.10. ²⁶² Una funzione

$$\|\cdot\| : V \rightarrow \mathbb{R},$$

con V spazio vettoriale, è detta **norma**, se soddisfa le seguenti proprietà:

1. $\forall \underline{v} \in V : \|\underline{v}\| \geq 0 \wedge \|\underline{v}\| = 0 \Rightarrow \underline{v} = \underline{0} \in V;$
2. $\forall \underline{v} \in V, \forall \alpha \in \mathbb{R} : \|\alpha \cdot \underline{v}\| = |\alpha| \cdot \|\underline{v}\|;$
3. $\forall \underline{u}, \underline{v} \in V : \|\underline{u} + \underline{v}\| \leq \|\underline{u}\| + \|\underline{v}\|.$

Se lo spazio vettoriale è $V = \mathbb{R}^n$, allora le norme sono generalmente norme- p : se $\underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$, allora

$$\|\underline{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, \quad p = 1, 2, \dots$$

Le norme più utilizzate in Analisi Numerica sono:

- $\|\underline{x}\|_1 = \sum_{i=1}^n |x_i|$ (**norma 1**);
- $\|\underline{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$ (**norma 2**, o Euclidea);
- $\|\underline{x}\|_\infty = \lim_{p \rightarrow \infty} \|\underline{x}\|_p = \max_{i=1,\dots,n} |x_i|$ (**norma ∞**).

Esempio 3.9. ²⁶³ $x = \begin{pmatrix} 1 \\ -3 \\ 2 \end{pmatrix}$

$$\begin{aligned} \|\underline{x}\|_1 &= 1 + 3 + 2 = 6; \\ \|\underline{x}\|_2 &= \sqrt{1 + 4 + 9} = \sqrt{14}; \\ \|\underline{x}\|_\infty &= 3. \end{aligned}$$

Osservazione 3.35. Tutte le norme sono **equivalenti**, nel senso che considerate $\|\cdot\|_{p_1}$ e $\|\cdot\|_{p_2}$, allora $\exists \alpha, \beta > 0$, indipendenti da \underline{x} , t.c.:

$$\alpha \|\underline{x}\|_{p_1} \leq \|\underline{x}\|_{p_2} \leq \beta \|\underline{x}\|_{p_1}.$$

3.10.2 Norme indotte su matrice

²⁶⁴

Definizione 3.11 (Norma indotta su matrice). Se $A \in \mathbb{R}^{m \times n}$,

$$\|A\|_p = \max_{\|\underline{x}\|_p=1} \underbrace{\|A\underline{x}\|_p}_{265}, \quad A \in \mathbb{R}^{m \times n}, \quad A\underline{x} \in \mathbb{R}^m, \quad \underline{x} \in \mathbb{R}^n, \quad (3.62)$$

è detta **norma p su matrice, indotta dalla norma p su vettore ($A\underline{x}$)**. (m ed n possono essere diversi)

²⁶²Slide 15 PDF 9, PG 136-139.

²⁶³Slide 16 PDF 9.

²⁶⁴Slide 16-19 PDF 9, PG 138-139.

È dimostrato, tramite la definizione:

1. $\|A\|_p \geq 0 \wedge \|A\|_p = 0 \Rightarrow A = 0 \in \mathbb{R}^{m \times n};$
2. $\|\alpha A\|_p = |\alpha| \cdot \|A\|_p;$
3. $\|A + B\|_p \leq \|A\|_p + \|B\|_p;$
4. $\|Ay\|_p \leq \|A\|_p \cdot \|y\|_p.$

Dimostrazione. Il vettore $\frac{y}{\|y\|_p}$ ha norma 1. Infatti:

$$\left\| \frac{y}{\|y\|_p} \right\| = \frac{1}{\|y\|_p} \cdot \|y\|_p = 1. \quad (3.63)$$

Pertanto,

$$\|Ay\|_p = \|y\|_p \cdot \left\| A \cdot \frac{y}{\|y\|_p} \right\|_p \leq \|y\|_p \cdot \underbrace{\max_{\|x\|_p=1} \|Ax\|_p}_{\|A\|_p} = \|y\|_p \cdot \|A\|_p = \|A\|_p \cdot \|y\|_p.$$

□

5. $\|I\|_p = 1$ (s.se la norma è indotta).

Dimostrazione.

$$\|I\|_p = \max_{\|x\|_p=1} \|I \cdot x\|_p = \max_{\|x\|_p=1} \|x\|_p = 1.$$

□

La proprietà 4. è importante per il condizionamento del problema e denota come **proprietà di compatibilità tra la norma su vettore e quella indotta su matrice**.

Sia $(a_{ij}) = A \in \mathbb{R}^{m \times n}$:

$$\begin{aligned} \|A\|_1 &= \max_{j=1,\dots,n} \sum_{i=1}^m |a_{ij}|; \\ \|A\|_\infty &= \max_{i=1,\dots,m} \sum_{j=1}^n |a_{ij}|; \\ \|A\|_2 &\leq \sqrt{\|A\|_1 \cdot \|A\|_\infty}. \end{aligned}$$

Esempio 3.10. ²⁶⁷ $A = \begin{bmatrix} 1 & -2 \\ -3 & 4 \\ 5 & -6 \end{bmatrix} \in \mathbb{R}^{3 \times 2}$

$$\begin{aligned} \|A\|_1 &= \max\{9, 12\} = 12; \\ \|A\|_\infty &= \max\{3, 7, 11\} = 11; \\ \|A\|_2 &\leq \sqrt{11 \cdot 12} \quad (= \text{ media geometrica } 11 \text{ e } 12). \end{aligned}$$

Osservazione 3.36. Matlab dispone della function built-in *norm* per calcolare la norma di un/a vettore/matrice.

²⁶⁵ Norma sul vettore Ax .

²⁶⁶ Commutazione prodotto scalare.

²⁶⁷ Slide 18 PDF 9.

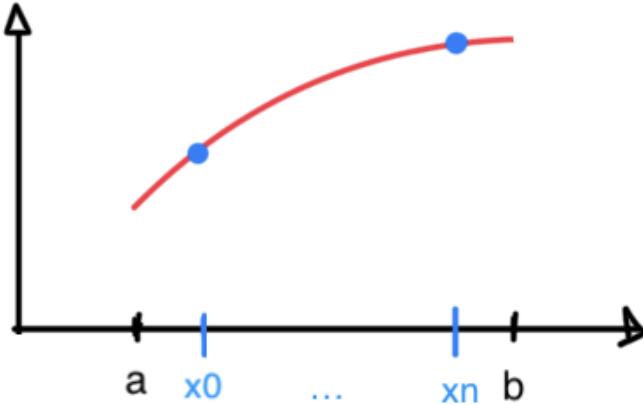


Figura 9: Esempio delle ascisse (4.1)

4 Approssimazioni di funzioni

²⁶⁸ In molte applicazioni è spesso richiesto di approssimare una funzione $f : [a, b] \rightarrow \mathbb{R}$, in quanto questa potrebbe essere troppo complessa o non nota. Per la funzione sono conosciute le ascisse (tra loro distinte):

$$a \leq x_0 < x_1 < \dots < x_n \leq b, \quad x_i \neq x_j, \quad i \neq j, \quad \forall i, j = 0, \dots, n. \quad (4.1)$$

Con (4.1) sono rappresentate $n + 1$ ascisse distinte.

4.1 Polinomio Interpolante

In seguito sarà considerata la notazione

$$f_i = f(x_i), \quad (x_i, f_i), \quad i = 0, \dots, n,$$

per ricercare un polinomio interpolante $f(x)$ sulle ascisse (4.1), in modo che, con $p(x_i) \in \Pi_n$, valgano le condizioni di interpolazione

$$p(x_i) = f_i, \quad i = 0, \dots, n. \quad (4.2)$$

Divagazione su Π_n : **Π_n è lo spazio vettoriale dei polinomi di grado al più n .** Uno spazio vettoriale è un insieme per il quale una combinazione lineare dei suoi elementi produce un elemento dello spazio stesso. Infatti, $\forall q_1, q_2 \in \Pi_n$ e $\forall \alpha, \beta \in \mathbb{R}$, è ottenuto che $\alpha \cdot q_1(x) + \beta \cdot q_2(x) \in \Pi_n$. $\alpha \cdot q_1(x)$ e $\beta \cdot q_2(x)$ sono moltiplicazioni che non aumentano il grado della somma, al massimo lo diminuiscono.

Se $p(x) \in \Pi_n$ è il polinomio interpolante $f(x)$, allora è possibile scriverlo come

$$p(x) = \sum_{k=0}^n a_k x^k. \quad (4.3)$$

²⁶⁸Slide 2 PDF 15, PG 77-104. L'interpolazione polinomiale è l'interpolazione di una serie di valori (ad esempio dei dati sperimentali) con una funzione polinomiale che passa per i punti dati.

I polinomi $x^0, x^1, x^2, \dots, x^n \in \Pi_n$ sono linearmente indipendenti e questi costituiscono la base canonica di Π_n . Pertanto, $\dim(\Pi_n) = n + 1$. Questo significa che, per individuare univocamente un polinomio di Π_n , saranno necessarie $n + 1$ condizioni linearmente indipendenti.

L'esistenza e l'unicità del polinomio interpolante è enunciata dal seguente teorema.

Teorema 4.1 (Esistenza ed Unicità del polinomio interpolante).²⁶⁹ $\exists! p(x) \in \Pi_n : p(x_i) = f_i, i = 0, \dots, n$, date le coppie di dati $(x_i, f_i), i = 0, \dots, n$, soddisfacenti (4.1).

Dimostrazione. Considerato il polinomio incognito in forma (4.3) ed imposte le condizioni di interpolazione (4.2):

$$p(x_i) \equiv \sum_{k=0}^n a_k x_i^k = f_i, \quad i = 0, \dots, n. \quad (4.4)$$

Questo è un sistema di equazioni lineari $(n + 1)$, nelle $n + 1$ incognite a_0, \dots, a_n . Il sistema può essere denotato in forma vettoriale come

$$V \underline{a} = \underline{b}, \quad (4.5)$$

dove

$$\underline{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix} \in \mathbb{R}^{n+1}, \quad V = \begin{bmatrix} x_0^0 & x_0^1 & \dots & x_0^n \\ x_1^0 & x_1^1 & \dots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ x_n^0 & x_n^1 & \dots & x_n^n \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}.$$

V è la matrice dei coefficienti, definita univocamente dalle ascisse, e trasposta di una matrice di Vandermonde.

Definizione 4.1 (Matrice di Vandermonde). Una matrice di Vandermonde ha la caratteristica di essere definita con attraverso una progressione geometrica, ovvero gli elementi di V sono del tipo $v_{i,j} = v_i^{j-1}$.

Osservazione 4.1. È nota la seguente proprietà delle matrici di Vandermonde, quindi di V :

$$\det(V) \stackrel{270}{=} \prod_{\substack{i > j \\ i \neq j}} (x_i - x_j). \stackrel{271}{=}$$

272

In virtù dell'ipotesi (4.1), segue che $\det(V) \neq 0 \Rightarrow \exists!$ la soluzione per (4.5) $\iff \exists! p(x) \in \Pi_n$ che soddisfa le condizioni di interpolazione 4.2. \square

Il problema discreto (4.5), per la determinazione di (4.4), deriva dall'aver scelto la base delle potenze, ovvero x^0, x^1, \dots, x^n , come base di Π_n .

Definizione 4.2 (Polinomio Interpolante). Il polinomio (4.4) è noto come **polinomio interpolante** la funzione sulle ascisse assegnate.

Il calcolo del polinomio interpolante tramite la risoluzione di (4.5) è inefficiente per il costo computazionale elevato e per il numero di condizionamento di V , il quale cresce rapidamente al crescere di n .

È necessario cercare un modo alternativo per il calcolo del polinomio interpolante $p(x)$, ovvero utilizzare una base diversa da quella delle potenze per il polinomio interpolante. Per questo sono definiti i polinomi della seguente Sezione, atti a calcolare il polinomio interpolante con una base diversa.

²⁶⁹Slide 2 PDF 15, TH 4.1 PG 78

²⁷¹Produttoria di tutte le coppie di elementi ascisse x con indice $i > j$. Le ascisse x_0, \dots, x_n soddisfano (4.1).

²⁷²Non nullo dato che le ascisse si interpolazione x_0, \dots, x_n sono distinte.

4.2 Forma di Lagrange e di Newton

Definizione 4.3 (Base di Lagrange).²⁷³ La **base di Lagrange** è definita dai seguenti **polinomi di Lagrange**:

$$L_{in} = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}, \quad i = 0, 1, \dots, n, \quad (4.6)$$

dove i stabilisce quale polinomio è considerato ed n determina il grado del polinomio.

Date le ascisse definite come (4.1), allora i polinomi di Lagrange (4.6) sono ben definiti ed hanno le seguenti proprietà:

P1) I polinomi sono ben distinti se $x_i - x_j \neq 0 (\rightarrow x_i \neq x_j)$, $i \neq j$ ²⁷⁴;

P2) ²⁷⁵ $L_{in}(x) \in \Pi_n$, $\forall i = 0, \dots, n$;

P3) ²⁷⁶ $L_{in}(x_k) = \begin{cases} 1 & \text{se } k = i \text{ (quindi nom=denom);} \\ 0 & \text{se } k \neq i. \end{cases}$

P4) I polinomi $\{L_{0n}(x), \dots, L_{nn}(x)\}$ sono tra loro linearmente indipendenti²⁷⁸. Infatti, se

$$\sum_{i=0}^n \alpha_i L_{in}(x) = 0, \forall x \implies \alpha_i = 0, \quad i = 0, \dots, n.$$

È possibile dimostrare la tesi per un generico $k \in \{0, \dots, n\}$ valutando $0 = \sum_{i=0}^n \alpha_i L_{in}(x_k) = \underbrace{\alpha_k L_{kn}(x_k)}_{1 \text{ per P3}} = \alpha_k$.

Pertanto, $\underbrace{\{L_{0n}(x), \dots, L_{nn}(x)\}}_{n+1}$ costituiscono una base per Π_n , detta **base di Lagrange**.

Un'importante conseguenza dei punti P3) e P4) è il prossimo Teorema.

Teorema 4.2 (Forma di Lagrange). Il polinomio interpolante $p(x) \in \Pi_n$, che soddisfa le condizioni (4.2), è dato da

$$p(x) = \sum_{k=0}^n f_k L_{kn}(x). \quad (4.7)$$

Dimostrazione. Definita la funzione di Kronecker (da conoscere):

$$L_{in}(x_i) = \delta_{ik} = \begin{cases} 1 & i = k; \\ 0 & i \neq k. \end{cases} \quad (4.8)$$

²⁷³Slide 7 PDF 15, PG 79.

²⁷⁴Ipotesi di (4.1).

²⁷⁵La proprietà, il grado dei polinomi L_{in} , non è determinata dal denominatore di L_{in} , ma dal nominatore di questa. $L_{in}(x)$ è una moltiplicazione di polinomi monici di grado n moltiplicati n volte.

²⁷⁶Dimostrazione di P2).

²⁷⁷ $x = x_j$ in almeno una moltiplicazione, quindi è 0.

²⁷⁸Significa che una combinazione lineare per p , polinomio nullo, deve essere a coefficienti nulli.

²⁷⁹Coefficienti di rappresentazione del polinomio interpolante rispetto alla base interpolante.

Quindi

$$L_{kn}(x_i) = \delta_{ki} \Rightarrow p(x_i) = \sum_{k=0}^n f_k L_{kn}(x_i) = \sum_{k=0}^n f_k \delta_{ki} = f_i \underset{1}{\underset{\parallel}{\delta_{ii}}} = f_i, \quad \forall i = 0, \dots, n.$$

□

Definizione 4.4. (4.7) costituisce la **forma di Lagrange** del polinomio interpolante.

La precedente Definizione significa che (4.7) è espresso rispetto alla base di Lagrange, anche se il polinomio interpolante non varia.

Algoritmo 4.1 Implementazione del polinomio interpolante nella forma di Lagrange.

```

function y = lagrange (xi, fi, x)
%   function y = lagrange (xi, fi, x)
%   Implementazione del polinomio interpolante nella forma di Lagrange.
%
%   Input:
%   xi - vettore delle ascisse di interpolazione (utili per calcolare f)
%   fi - vettore delle immagini delle xi
%   x - vettore delle ascisse
%
%   Output:
%   y - vettore sui dati interpolati
%
if length(xi) ~= length(unique(xi)), error('ascisse non distinte'), end
if length(xi) - length(fi) ~= 0
    error('xi e fi hanno lunghezze diverse')
end
n = length(xi); %grado del polinomio n-1 (in Matlab i contatori sono da 1
    ad n-1);
if n < 1, error('numero di xi e fi insufficienti'), end
y = zeros(size(x)); %y e' una matrice quadrata con dimensione indipendente
    da n;
for i = 1 : n
    y = y + fi(i) * Lin(i, xi, x);
end
return

```

È importante utilizzare manipolazioni vettoriali al posto di cicli, in quanto le prime sono migliori per prestazioni (oltre ad essere più compatte).

Date le $n+1$ coppie (x_i, f_i) , $i = 0, \dots, n$, con $x_i \neq x_j$, se $i \neq j$ e $f_i \equiv f(x_i)$,

$$\exists! p(x) \in \Pi_n : p(x_i) = f_i, \quad i = 0, \dots, n,$$

dove $p(x)$ è il polinomio interpolante di $f(x)$ sulle ascisse assegnate.

²⁸⁰Condizione ideale perché non è sempre vero che f possa essere calcolata (nel nostro lo è possibile).

Algoritmo 4.2 Implementazione della base del polinomio interpolante nella forma di Lagrange.

```
function L = Lin(i, xi, x)
%
%   function L = Lin(i, xi, x)
%   Lin function che calcola i-esima inversa del polinomio di Lagrange.
%   Input:
%       i      - indice della funzione
%       xi    - vettore ascisse sulle quali e' calcolata f
%       x     - stesso della function lagrange
%
%   Output:
%       L    - vettore con la base ricercata
%
%   Non sono inclusi controlli perche' svolti nel layer superiore
L = ones(size(x));
zi=xi(i);
xi(i)=[];
n = length(xi);
for j = 1 : n
    L = L.*(x-xi(j));
end
L = L / prod(zi - xi);
return
```

Inoltre, è utile ribadire che le rappresentazioni del polinomio interpolante (4.3) e (4.7) sono algebricamente uguali, quindi segue che:

$$p(x) = \sum_{i=0}^n \mathbf{a}_i x^i \equiv \sum_{i=0}^n f_i L_{in}(\mathbf{x}), \quad (4.9)$$

dove x_i rappresenta la base delle potenze ed $L_{in}(x)$ la base di Lagrange.

Questa semplicità formale non si concilia con requisiti computazionali per il calcolo, in modo incrementale, del polinomio.

Il problema è capire come modificare la rappresentazione del polinomio $p(x)$ rispetto alla base utilizzata, se aggiunta un'ulteriore ascissa di interpolazione $x_{n+1} \notin \{x_0, \dots, x_n\}$, in modo da calcolare f_{n+1} (ovvero $f(x_{n+1})$). Data $\hat{p}(x) \in \Pi_{n+1} : \hat{p}(x_i) = f_i, i = 0, \dots, n + 1$, è interessante capire in che relazione sono $\hat{p}(x)$ e $p(x)$, considerando che sono considerate due basi per i polinomi. Come visto per (4.9), allora

$$\begin{aligned} \hat{p}(x) &= \sum_{i=0}^{n+1} \hat{\mathbf{a}}_i x^i \equiv \sum_{i=0}^{n+1} f_i L_{i,n+1}(\mathbf{x}), \\ p(x) &= \sum_{i=0}^n a_i x^i \equiv \sum_{i=0}^n f_i L_{in}(\mathbf{x}), \end{aligned} \quad (4.10)$$

con rispettivamente $\hat{\mathbf{a}}_i$ ed $L_{i,n+1}(\mathbf{x})$, x^i ed $L_{in}(\mathbf{x})$ collegate fra loro. Le funzioni di base sono le stesse, ma i coefficienti sono diversi, sono soluzione di due sistemi lineari di dimensione diversa, con matrici di coefficienti diverse (le quali non si prestano bene alla costruzione incrementale del polinomio).

È necessario definire una base di rappresentazione, in modo tale che $\hat{p}(x)$ sia ottenuta da $p(x)$, aggiungendo una funzione di base ed il relativo coefficiente, mentre tutti gli altri rimangono gli stessi di $p(x)$. Questo è possibile utilizzando, come base di rappresentazione, i **polinomi di base di Newton**:

$$\begin{cases} \omega_0(x) \equiv 1; \\ \omega_r(x) = \underbrace{(x - x_{r-1})}_{281} \omega_{r-1}(x), \quad r = 1, \dots, n. \end{cases} \quad (4.11)$$

Se gli altri coefficienti delle funzioni di base rimangono gli stessi di $p(x)$ è possibile definire il nuovo polinomio ottenendo una costruzione incrementale del polinomio interpolante.

ω_r definito in (4.11) è una relazione ricorsiva utilizzata per il calcolo, in modo efficiente, di $p(x)$ e può essere rappresentato come

$$\omega_{k+1}(x) = (x - x_k) \omega_k(x), \quad k = 0, \dots, n.$$

oppure è possibile scriverlo come polinomio monico, ovvero il polinomio con coefficiente del termine di grado più alto uguale ad 1, in forma esplicita come

$$w_k(x) = \prod_{i=0}^{k-1} (x - x_i), \quad k = 1, \dots, n. \quad (4.12)$$

Utilizzando il principio di induzione è possibile verificare le seguenti **proprietà di ω_r** :

1. $w_r(x)$ è un polinomio monico di grado r , $\forall r \geq 0$ (ovvero $\omega_r \in \Pi_r$);
2. $\forall r \geq 0 : \underbrace{\{\omega_0(x), \dots, \omega_r(x)\}}_{r+1}$ sono linaremente indipendenti ²⁸³;

²⁸¹Moltiplicazione dell'elemento $\omega_{r-1}(x)$ per un polinomio monico; x_{r-1} è la $(r-1)$ -esima ascissa di interpolazione.

²⁸² $(r-1)$ -esima ascissa di interpolazione.

²⁸³Conseguenza del punto precedente dato che i gradi sono distinti.

3. $\{\omega_0(x), \dots, \omega_r(x)\}$ sono una base per Π_r ²⁸⁴;
4. $\forall r \geq 1 : w_r(x) = \prod_{j=0}^{r-1} (x - x_j)$ ²⁸⁵;
5. Date le ascisse distinte x_0, \dots, x_n : $\begin{cases} w_r(x_j) = 0, & j \leq r-1, \\ w_r(x_j) \neq 0, & j \geq r. \end{cases}$

È possibile denotare

$$p_r(x) \in \Pi_r : p_r(x_i) = f_i, \quad i = 0, \dots, r. \quad (4.13)$$

È possibile definire la famiglia dei polinomi $\{p_r\}_{r=0, \dots, n}$ (dove n è il grado da raggiungere), in modo incrementale, con la seguente modalità.

Teorema 4.3 (Forma di Newton). ²⁸⁶ La famiglia di polinomi interpolanti definita in (4.13) è ottenuta ricorsivamente come:

$$\begin{cases} p_0(x) \equiv f_0 \in \Pi_0 \\ p_r(x) = p_{r-1}(x) + f[x_0, \dots, x_r] \omega_r(x), \quad r = 1, 2, \dots \end{cases} \quad (4.14)$$

dove $f[x_0, \dots, x_r]$ è la differenza divisa di ordine r della funzione f sulle ascisse x_0, \dots, x_r , definita come:

$$f[x_0, \dots, x_r] \stackrel{\text{def}}{=} \sum_{i=0}^r \frac{f_i}{\prod_{j=0, j \neq i}^r (x_i - x_j)}. \quad (4.15)$$

Dimostrazione. ²⁸⁷ Per induzione la tesi è vera, con $r = 0$, poiché $p_0(x) \equiv f_0 \equiv f[x_0]$. Supposto vero per $r - 1$ sarà dimostrato per r .

Per ipotesi è supposto che

$$p_{r-1}(x) \in \Pi_{r-1} : p_{r-1}(x_i) = f_i, \quad i = 0, \dots, r-1$$

e che $p(x)$ sia definito come in (4.14). Sarà dimostrato quanto segue:

1. $p_r(x_i) = f_i, \quad i = 0, \dots, r$ ²⁸⁸;
2. $f[x_0, \dots, x_r]$ è definita come in (4.15).

Dimostrazione di 1.

$$\forall i = 0, \dots, r-1 : p_r(x_i) = \underbrace{p_{r-1}(x_i)}_{f_i} + \underbrace{f[x_0, \dots, x_r]}_{289} \underbrace{\omega_r(x_i)}_0 = f_i.$$

Per $i = r$, essendo le ascisse distinte, $\omega_r(x_r) \neq 0$ ed imponendo $p_r(x_r) = p_{r-1}(x_r) + f[x_0, \dots, x_r] \omega_r(x_r) = f_r$ è ricavato che

$$f[x_0, \dots, x_r] \stackrel{\text{def}}{=} \frac{f_r - p_{r-1}(x_r)}{\omega_r(x_r)}, \quad (4.16)$$

²⁸⁴Conseguenza del punto precedente.

²⁸⁵Ottenuta induttivamente dalla seconda riga del sistema (4.11) con gli zeri dei polinomi conosciuti. Questa relazione ricorsiva è utilizzata per il calcolo efficiente del polinomio.

²⁸⁶Slide 5 PDF 16, TH 4.3 PG 80.

²⁸⁷Dimostrazione riconosciuta del sistema (4.14) e di (4.15).

²⁸⁸Quindi è necessario dimostrare che p_r è calcolabile come p_{r-1} .

²⁸⁹Non è definito.

che è ben definito essendo $\omega_r(x_r) \neq 0$. È necessario dimostrare che la differenza divisa espressa come (4.16) coincide con (4.15). A questo fine è possibile osservare che $f[x_0, \dots, x_r]$ è il coefficiente principale di $p_r(x)$. È definito l'algoritmo per la soluzione in due passi:

1. scrittura di $p_r(x)$ in forma di Lagrange;
2. calcolo del coefficiente principale di $p_r(x)$ in forma di Lagrange.

Quindi i due coefficienti dovranno coincidere, per il principio di identità dei polinomi.

La forma di Lagrange di $p_r(x)$ è espressa come segue:

$$\begin{aligned}
p_r(x) &= \sum_{i=0}^r f_i L_{ir}(x) \\
&\stackrel{290}{=} \sum_{i=0}^r f_i \prod_{j=0, j \neq i}^r \frac{x - x_j}{x_i - x_j} \\
&\stackrel{291}{=} \sum_{i=0}^r \frac{f_i}{\prod_{j=0, j \neq i}^r (x_i - x_j)} \cdot \prod_{j=0, j \neq i}^r (x - x_j) \\
&\stackrel{292}{=} x^r \sum_{i=0}^r \frac{f_i}{\prod_{j=0, j \neq i}^r (x_i - x_j)} + (\text{termini di ordine inferiore in } x) \\
&\stackrel{293}{\equiv} x^r f[x_0, \dots, x_r] + (\text{termini di ordine inferiore in } x).
\end{aligned}$$

□

Da questo è possibile concludere che la (4.15) vale.

Dimostrazione 2. La seconda parte della dimostrazione del Teorema è svolta come dimostrazione del punto P5), esposto fra poco. □

□

Osservazione 4.2 (Forma di Newton del polinomio interpolante). ²⁹⁴ Sia $p(x) \in \Pi_n$, polinomio interpolante $f(x)$ sulle ascisse $\{x_0, \dots, x_n\}$, $p(x)$ coincide con $p_n(x)$ in (4.13). Per induzione è ottenuta la **forma di Newton del polinomio interpolante**:

$$p(x) \equiv p_n(x) = \sum_{i=0}^n f[\underbrace{x_0, \dots, x_i}_{i+1}] \omega_i(x), \quad (4.17)$$

dove $\omega_i(x)$ è la dunzione di base.

N.B.: La forma di Lagrange e la forma di Newton di $p(x)$ definiscono lo stesso polinomio interpolante, il quale esiste ed è unico, se le ascisse di interpolazione sono tra loro distinte. Ciò significa che per $p(x)$ varia solo la sua rappresentazione. Questo ha il pregio di prestarsi alla costruzione incrementale del polinomio aggiornandosi in modo dinamico.

²⁹⁰ Scrittura del polinomio di Lagrange di grado r in forma esplicita ($r \rightarrow n$).

²⁹¹ $\prod_{j=0, j \neq i}^r (x - x_j)$ è un polinomio monico di grado r , è una produttoria di r polinomi di grado 1.

²⁹² Forma di Lagrange.

²⁹³ Forma diversa da Lagrange.

²⁹⁴ Slide 8 PDF 16.

Esempio di caso d'uso polinomio interpolante: se fosse richiesto di approssimare ciò che è sotto una certa curva questa viene calcolata con un numero prefissato di punti, poi se ne sono aggiungono uno ad uno fino a che non è più possibile effettuare una stima accurata.

La definizione in forma di Newton di $p(x)$ come (4.17) permette di ottenere funzioni di base in funzione delle precedenti. Pertanto, questo non permette di calcolare la differenza divisa e la rappresentazione $f[x_0, \dots, x_r]$ non è algoritmicamente preferibile. Per arrivare al fine di ottenere un algoritmo efficiente, sono esaminate alcune proprietà delle differenze divise.

Proprietà differenze divise (4.15):

P1) (Linearità dell'operatore) Se f e g sono funzioni in variabili reali $\alpha, \beta \in \mathbb{R}$, allora

$$(\alpha \cdot f + \beta \cdot g)[x_0, \dots, x_r] = \alpha \cdot f[x_0, \dots, x_r] + \beta \cdot g[x_0, \dots, x_r];$$

P2) (Simmetria dell'operatore) Se $\{i_0, \dots, i_r\}$ è una permutazione di $\{0, \dots, r\}$, allora:

$$f[x_0, \dots, x_r] = f[x_{i_0}, \dots, x_{i_r}];$$

P3) Se $f(x) = \sum_{i=0}^k a_i x^i$, allora $f[x_0, \dots, x_r] = \begin{cases} a_k, & r = k; \\ 0, & r > k. \end{cases}$

P4) Se $f \in C^{(r)}[a, b]$, $a = \min_{i=0, \dots, r} x_i$, $b = \max_{i=0, \dots, r} x_0$, allora

$$f[x_0, \dots, x_r] = \frac{f^{(r)}(\xi)}{r!}, \quad \xi \in [a, b], \quad (4.18)$$

dove $[a, b]$ è il più piccolo intervallo che contiene le ascisse di interpolazione. Questa proprietà vale anche nel caso in cui 2 o più ascisse coincidano. In questo caso, la definizione di differenza divisa vale come limite.

P5)

$$\underbrace{f[x_0, \dots, x_r]}_{r+1} = \frac{\overbrace{f[x_1, \dots, x_r]}^r - \overbrace{f[x_0, \dots, x_{r-1}]}^r}{x_r - x_0}. \quad (4.19)$$

(4.19) è una proprietà algoritmica importante, è utilizzata per il calcolo efficiente della differenze divisa ed è possibile dimostrarla come segue.

²⁹⁵Slide 9-11 PDF 16, TH 4.4 PG 82.

²⁹⁶Sia $g(x) \in \Pi_n$, allora il polinomio $p(x) \in \Pi_n$ che interpola $g(x)$ sulle ascisse distinte x_0, \dots, x_n è $g(x)$ stesso ($p(x) = g(x)$), per l'unicità del polinomio interpolante. Il polinomio $p(x) \in \Pi_r$, con $r > n$, che interpola $g(x)$ sulle $r+1$ ascisse distinte x_0, \dots, x_r , è sempre $g(x)$ per lo stesso motivo.

²⁹⁷Le due differenze divise al denominatore differiscono solo per l'ascissa x_0 . Il concetto è che, se è applicata iterativamente questa proprietà, allora è possibile scrivere le differenze divise come combinazione di differenze divise, con $r-1$ argomenti, fino ad ottenere un solo argomento. Quando l'argomento è unico allora la differenza divisa coinciderà con il valore calcolato nell'argomento.

Dimostrazione P5).²⁹⁸

$$\begin{aligned}
& \frac{1}{x_r - x_0} (f[x_1, \dots, x_r] - f[x_0, \dots, x_{r-1}]) = \frac{1}{x_r - x_0} \left(\sum_{k=1}^r \frac{f_k}{\prod_{j=1, j \neq k}^r (x_k - x_j)} - \sum_{k=0}^{r-1} \frac{f_k}{\prod_{j=0, j \neq k}^{r-1} (x_k - x_j)} \right) \\
& = \frac{1}{x_r - x_0} \left(\underbrace{\frac{f_r}{\prod_{j=1, j \neq r}^r (x_r - x_j)} - \frac{f_0}{\prod_{j=0, j \neq 0 \rightarrow j=1}^{r-1} (x_0 - x_j)}}_{\text{Elementi comuni alle due sommatorie}} + \underbrace{\sum_{k=1}^{r-1} f_k \left(\frac{1}{\prod_{j=1, j \neq k}^r (x_k - x_j)} - \frac{1}{\prod_{j=0, j \neq k}^{r-1} (x_k - x_j)} \right)}_{\substack{(c) \\ 299}} \right) \\
& = (a) + (b) + (c),
\end{aligned} \tag{4.20}$$

dove $\frac{1}{x_r - x_0}$ è inglobata in (a), (b) e (c) come segue:

$$\begin{aligned}
(a) &= \frac{1}{x_r - x_0} \frac{f_r}{\prod_{j=1, j \neq r}^r (x_r - x_j)} \stackrel{300}{=} \frac{f_r}{\prod_{j=0, j \neq r}^r (x_r - x_j)}; \\
(b) &= \frac{1}{x_0 - x_r} = \frac{f_0}{\prod_{j=0, j \neq 0 \rightarrow j=1}^{r-1} (x_0 - x_j)} = \frac{f_0}{\prod_{j=0, j \neq 0}^{r-1} (x_0 - x_j)}; \\
(c) &= \frac{1}{x_r - x_0} \sum_{k=1}^{r-1} \frac{f_k}{\prod_{j=1, j \neq k}^{r-1} (x_k - x_j)} \underbrace{\left(\frac{1}{x_k - x_r} - \frac{1}{x_k - x_0} \right)}_{\substack{301}} = \frac{1}{x_r - x_0} \sum_{k=1}^{r-1} \frac{f_k}{\prod_{j=1, j \neq k}^{r-1} (x_k - x_j)} \left(\frac{x_k - x_0 - x_k + x_r}{(x_k - x_r)(x_k - x_0)} \right) = \\
&\quad \sum_{k=1}^{r-1} \frac{f_k}{\prod_{j=0, j \neq k}^{r-1} (x_k - x_j)};
\end{aligned}$$

Pertanto, unendo i precedenti risultati:

$$(4.20) = \sum_{k=0}^r \frac{f_k}{\prod_{j=0, j \neq k}^r (x_k - x_j)} \stackrel{302}{=} f[x_0, x_1, \dots, x_r].$$

□

²⁹⁸Tramite la definizione di differenza divisa. Sono calcolate due differenze divise, è effettuata la somma ed è ottenuta l'espressione ricercata.

²⁹⁹Dato che gli indici da 1 ad $r - 1$ sono comuni ad entrambe le produttorie, saranno raggruppati gli addendi con indici comuni e lasciati gli addendi non comuni. Addendi con indice uguale ad r sono sommatorie, con indice uguale a 0 sono la somma $a + b$.

³⁰⁰Ultimo termine della sommatoria della differenza divisa delle ascisse x_0, \dots, x_r .

³⁰¹Termini necessari per le caratteristiche degli indici.

³⁰²Ovvero ciò che è necessario dimostrare.

Esempio di utilizzo della proprietà (4.19): Considerando il caso $n = 3$:

$$p(x) = f[x_0] \omega_0(x) + f[x_0, x_1] \omega_1(x) + f[x_0, x_1, x_2] \omega_2(x) + f[x_0, x_1, x_2, x_3] \omega_3(x),$$

\parallel
1

allora è possibile generalizzarlo in la seguente tabella triangolare:

	303 0	304 1	2	3
x_0	$f_0 = f[x_0]$			
x_1	$f_1 = f[x_1]$	$f[x_0, x_1]$		
x_2	$f_2 = f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$	
x_3	$f_3 = f[x_3]$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$

Gli elementi sulla diagonale principale, ovvero quelli in grassetto, sono i coefficienti del polinomio interpolante nella forma di Newton (4.17).

La tabella è calcolata dal basso verso l'alto per evitare di memorizzare tutti i dati della tabella. Non è necessario memorizzarla tutta perché è possibile sovrascrivere gli elementi a sinistra con quelli a destra (quindi in fine saranno presenti le differenze divise in grassetto). Per questo è necessario calcolare prima la colonna 0 e poi la 1, 2 e 3 sovrascrivendone, via via, gli elementi. Quanto appena scritto non è valido se gli elementi sono calcolati dall'alto verso il basso.

Utilizzando il metodo di calcolo più conveniente è possibile utilizzare due vettori: uno per le ascisse di interpolazione (x_0, \dots, x_3) ed uno per i dati del problema da sovrascrivere con i coefficienti dei polinomi.

La tabella ha complessità quadratica, come la forma di Lagrange, e minore rispetto a risolvere il sistema lineare con la matrice di Vandermonde (la quale ha complessità $\frac{2}{3}n^3$).

Le colonne della precedente tabella, dalla 1 alla 3, sono determinate (calcolate) dal basso verso l'alto come segue:

$$\begin{aligned} & \mathbf{1} \quad \left\{ \begin{array}{l} f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2} \\ f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1} \\ f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} \end{array} \right. \\ & \mathbf{2} \quad \left\{ \begin{array}{l} f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} \\ f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \end{array} \right. \\ & \mathbf{3} \quad \left\{ f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} \right. \end{aligned}$$

È possibile notare che al denominatore le ascisse di iterpolazione differiscono rispettivamente, dall'alto al basso, di un indice 1, 2 e 3 e che tali ascisse non sono comuni alle differenze divise al denominatore delle rispettive frazioni.

Osservazione 4.3.³⁰⁶ Nel caso generale, la precedente tabella arriverà fino a colonna n (l'indice partìra da 0) ed avrà struttura analoga.

Osservazione 4.4. Il costo computazionale dell'Algoritmo (4.3) è

$$\bullet \sum_{j=1}^n 3(n-j+1) \stackrel{307}{=} 3 \sum_{j=1}^n j = 3 \cdot \frac{n(n+1)}{2} \approx \frac{3}{2} n^2 \text{ flops};$$

³⁰³Differenza tra l'indice delle ascisse (utilizzata la proprietà incrementale delle differenze divise).

³⁰⁴Differenze divise su due ascisse.

³⁰⁵Differiscono di un'ascissa non comune.

³⁰⁶Slide 6 PDF 17.

³⁰⁷Somma naturale.

Algoritmo 4.3 Calcolo delle differenze divise.

```
% x - ascisse di interpolazione
% f - valori della funzione nelle ascisse
% x ed f sono vettori di dimensione n+1 perche' Matlab utilizza indici
% che partono da 1
n = length(x)-1; % grado del polinomio
for j = 1 : n
    for i = n+1 : -1 : j+1
        f(i)=(f(i)-f(i-1))/(x(i)-x(i-j)); % saranno modificati in
        % colonna j gli elementi che vanno dall'ultimo al j+1-esimo.
        % Il j-esimo contiente la differenza divisa, la quale e'
        % calcolata nella riga della nota
    end
end
```

- **2 vettori (x, f)** (x memorizza le ascisse interpolanti ed f le differenze divise);

Problema: È necessario considerare il calcolo efficiente del polinomio di Newton. Del polinomio sono noti i coefficienti ed è necessario calcolare la sommatoria. Sarà utilizzata la proprietà ricorsiva del polinomio di Newton. Al fine del calcolo efficiente del polinomio di Newton è valutato un problema più semplice rispetto alla base delle potenze, il calcolo del polinomio (4.3) ($p(x) = \sum_{k=0}^n a_k x^k$).

Partendo da un caso semplice, ovvero con $n = 3$:

$$p(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \stackrel{308}{=} a_0 + x(a_1 + x(a_2 + x a_3)).$$

Supposto di avere il vettore $a = \underbrace{[a_0 \ a_1 \ \dots \ a_n]}_{309}$ ^{n+1 elementi} è possibile calcolare il polinomio tramite l'Algoritmo 4.4.

Algoritmo 4.4 Algoritmo di Horner per il calcolo di un polinomio.

```
p = a(n+1); % al termine dell'algoritmo conterra' il valore del
            % polinomio
for i = n : -1 : 1
    p = p .* x + a(i);
end
```

Osservazione 4.5 (Costo dell'Algoritmo 4.4). Il costo dell'algoritmo è $2n$ flops (per componente di x).

L'Algoritmo 4.4 (di Horner) ha un costo minimale: sono svolte due operazioni algebriche elementari per calcolare il vettore $p(x)$, per iterazione. È, inoltre, importante osservare che questo algoritmo si presta ad essere vettorizzato

³⁰⁸Raggruppamento.

³⁰⁹In Matlab è necessario rappresentarli come $a(1) \ a(2) \ \dots \ a(n+1)$.

in Matlab, attraverso la moltiplicazione vettoriale \cdot^* , dove x , in riga 3 dell'Algoritmo 4.4, può essere di qualsivoglia forma (vettore, elemento singolo, matrice). Utilizzare le capacità vettoriali di Matlab rende più efficiente il codice (quindi è necessario utilizzarlo nell'elaborato).

Inoltre, è possibile generalizzare la differenza divisa al caso della base di Newton attraverso l'algoritmo di Horner generalizzato (Algoritmo 4.5).

Esempio 4.1. Con $n = 3$:

$$\begin{aligned} p(x) & \stackrel{310}{=} f[x_0] + f[x_0, x_1]\omega_1(x) + f[x_0, x_1, x_2]\omega_2(x) + f[x_0, x_1, x_2, x_3]\omega_3(x) \\ & \stackrel{311}{=} f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) \\ & = ((f[x_0, x_1, x_2, x_3](x - x_2) + f[x_0, x_1, x_2])(x - x_1) + f[x_0, x_1])(x - x_0) + f[x_0]. \end{aligned}$$

Algoritmo 4.5 Algoritmo di Horner generalizzato (per il calcolo di un polinomio).

```
% f(1), ..., f(n+1) - calcolato dal codice precedente per le
% differenze divise
% x(1), ..., x(n+1) - contiene le ascisse di interpolazione
p = f(n+1)
for i = n : -1 : 1
    p = p .* (x - x(i)) + f(i)
end
```

Osservazione 4.6. L'operazione \cdot^* in riga 5 dell'Algoritmo 4.5 è vettorizzabile.

Osservazione 4.7 (Costo dell'Algoritmo 4.5). Il costo dell'algoritmo è $3n$ flops per componente.

Intermezzo per la definizione della "Interpolazione di Hermite"³¹² Dato il polinomio interpolante in forma di Newton (4.17) allora è possibile utilizzare l'Algoritmo 4.6 per calcolare $p(x)$.

Algoritmo 4.6 Pseudo-codice calcolo $p(x)$.

```
w0 ≡ 1
p = 0
for i = 0 : n do
    p = p + ωi * ai
    ωi+1 = ωi * (x - xi)
end for
```

L'Algoritmo 4.6 utilizza il fatto che le funzioni di base possono essere ottenute in modo iterativo, moltiplicando un termine di grado uno (il polinomio monico). Pertanto, è meno efficiente dell'Algoritmo (di Horner generalizzato) 4.5 in quanto effettua $4n$ flops ed utilizza una variabile di appoggio (ω , la quale rappresenta $\omega_0, \omega_1, \omega_{i+1}$).

³¹⁰È noto come calcolare le differenze divise (ovvero i coefficienti).

³¹¹Scrittura di $\omega_i(x)$ (polinomio di Newton) in forma estesa in modo da appurare efficientemente il valore in un punto, o in più di uno. La forma estesa è la forma in cui i polinomi in base di Newton sono usati perché la loro espressione è nota.

³¹²Slides 1-3 PDF 18.

Tuttavia, l'Algoritmo 4.6 si presta facilmente per derivare un algoritmo per il calcolo della derivata prima di $p(x)$, la quale sarà significativa per gli argomenti che saranno trattati, ovvero:

$$p'(x) = \sum_{i=0}^n \text{coeff. } a_i \omega'_i(x) = \sum_{\substack{i=1 \\ 313}}^n a_i \omega'_i(x).$$

L'obiettivo è derivare un algoritmo per il calcolo delle derivate $\omega'(x)$ e $p'(x)$. Ciò significa che le righe precedenti si riducono a derivare un algoritmo, relativamente efficiente, per il calcolo delle derivate prime dei polinomi di base di Newton.

È possibile osservare che il calcolo di $\omega_i(x)$ come

$$\omega_i(x) = \prod_{k=0}^{i-1} (x - x_k) \Rightarrow \omega'_i(x) = \sum_{k=0}^{i-1} \prod_{j=0, j \neq k}^{i-1} (x - x_j)$$

è dispendioso e poco efficiente, se calcolato in questo modo.

Esempio 4.2. $\omega_2(x) = (x - x_0)(x - x_1) \rightarrow \omega'_2(x) = (x - x_0) + (x - x_1)$.

Osservazione 4.8. È possibile ottenere i polinomi, in modo ricorsivo, dall'Algoritmo 4.7 (ultima riga del for), ovvero:

$$\begin{cases} \omega_0(x) \equiv 1, \omega'_0(x) \equiv 0, \\ i \geq 1 : \omega'_i(x) = \frac{\partial f}{\partial x}[\omega_{i-1}(x)(x - x_{i-1})] = \omega'_{i-1}(x)(x - x_{i-1}) + \omega_{i-1}(x). \end{cases}$$

Questa osservazione porta alla definizione dell'Algoritmo 4.7 (dove ω_i rimane una variabile). L'algoritmo utilizza una variabile per la derivata, ω'_i , ed una per il polinomio di base, p'_i . Pertanto, dato un polinomio, espresso con base di Newton, è noto come calcolarlo e con esso la sua derivata prima.

Algoritmo 4.7 Algoritmo calcolo $p'(x)$.

```
w0 = 1, omega'_0 = 0, p' = 0
for i = 1 : n do
    omega'_i = omega'_{i-1} + omega'_{i-1} * (x - xi_{i-1})
    omega_i = omega_{i-1} * (x - xi_{i-1})
    p' = p' + ai * omega'_i
end for
```

4.3 Interpolazione di Hermite

³¹⁴ [Problema:] Supposto di avere un polinomio interpolante una funzione su un numero pari $(2n + 2)$ di ascisse, quest'ultime sono numerate come segue:

$$a \leq \overbrace{x_0 < x_{\frac{1}{2}} < x_1 < x_{\frac{3}{2}} < \dots < x_n < x_{n+\frac{1}{2}}}^{2n+2 \text{ ascisse}} \leq b. \quad (4.21)$$

³¹³Trasformazione in $i = 1$ perché $\omega_0 \equiv 1$ (vedere (4.11)).

³¹⁴Slide 4-11 PDF 18, PG 84-86.

Pertanto, sotto la precedente ipotesi di ascisse distinte, $\exists! p(\mathbf{x}) \in \Pi_{2n+1}$ (condizione importante) tale che le condizioni del polinomio interpolante (4.2) diventano le seguenti:

$$\left. \begin{array}{l} p(x_i) = f(x_i) \\ p\left(x_{i+\frac{1}{2}}\right) = f\left(x_{i+\frac{1}{2}}\right) \end{array} \right\} i = 0, \dots, n. \quad (4.22)$$

Se $\forall i = 0, \dots, n : x_{i+\frac{1}{2}} \rightarrow x_i$, le ascisse che sono tra due interi, quelle con indice frazionario (ovvero $i + \frac{1}{2}$), sono fatte tendere all'ascissa a sinistra, quindi la condizione di interpolazione è duplicata. Per evitare la duplicazione è possibile riscrivere le condizioni di interpolazione (4.22), in modo equivalente, come:

$$\left. \begin{array}{l} p(x_i) = f(x_i) \\ \frac{p\left(x_{i+\frac{1}{2}}\right) - p(x_i)}{x_{i+\frac{1}{2}} - x_i} = \frac{f\left(x_{i+\frac{1}{2}}\right) - f(x_i)}{x_{i+\frac{1}{2}} - x_i} \end{array} \right\} i = 0, \dots, n, \quad (4.23)$$

per

$$x_{i+\frac{1}{2}} \rightarrow x_i : \begin{cases} \frac{p\left(x_{i+\frac{1}{2}}\right) - p(x_i)}{x_{i+\frac{1}{2}} - x_i} \rightarrow p'(x_i) & i = 0, \dots, n. \\ \frac{f\left(x_{i+\frac{1}{2}}\right) - f(x_i)}{x_{i+\frac{1}{2}} - x_i} \rightarrow f'(x_i) \end{cases}$$

Osservazione 4.9. La seconda condizione di (4.22) è stata modificata nella seconda condizione di (4.23), sottraendo la prima condizione alla seconda di (4.22) e dividendo membro a membro per $x_{i+\frac{1}{2}} - x_i$.

Data $f(x) \in C^{(1)}$ è possibile ottenere ciò che segue (vedere (4.18)):

$$\frac{f\left(x_{i+\frac{1}{2}}\right) - f(x_i)}{x_{i+\frac{1}{2}} - x_i} = f\left[x_i, x_{i+\frac{1}{2}}\right] \rightarrow f[x_i, x_i] \equiv f'(x_i), \quad i = 0, \dots, n. \quad (4.24)$$

È possibile concludere che se $x_{i+\frac{1}{2}} \rightarrow x_i$, $\forall i = \underbrace{0, \dots, n}_{315} \Rightarrow \exists! p_{\mathbf{H}}(x) \in \Pi_{2n+1}$ tale che:

$$316 \quad \begin{cases} p_{\mathbf{H}}(x_i) = f(x_i) \\ p'_{\mathbf{H}}(x_i) = f'(x_i) \end{cases} \quad i = 0, \dots, n. \quad (4.25)$$

Definizione 4.5 (Polinomio interpolante di Hermite). $p_{\mathbf{H}}(x)$ è il polinomio interpolante di Hermite.

Pertanto, date le $n + 1$ ascisse distinte (4.1), $\exists! p_{\mathbf{H}}(x) \in \Pi_{2n+1}$, che soddisa le condizioni di interpolazione (4.25). Il polinomio $p_{\mathbf{H}}(x)$ interpola la funzione $f(x)$ e la sua derivata, $f'(x)$, nelle ascisse di interpolazione.

³¹⁵Quindi le ascisse sono $n + 1$.

³¹⁶Condizioni su polinomio e sulla sua derivata prima.

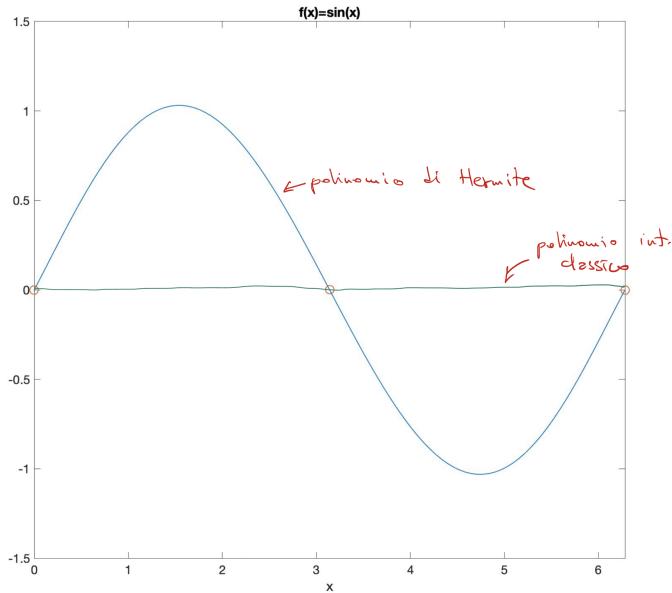


Figura 10: Esempio delle differenze di approssimazione

Esempio 4.3. ³¹⁷ Considerate le ascisse $x_i = i \cdot \pi$, $i = 0, 1, 2$ e considerando la funzione

$$f(x) = \sin(x) \stackrel{318}{\Rightarrow} f'(x) = \cos(x).$$

Spiegazione grafica dell'esempio in Figura 10 e 11.

Osservazione 4.10 (Calcolo $p_H(x)$, caso semplice). Per il calcolo del polinomio di Hermite è utilizzata la sua forma di Newton: un caso semplice, per estrapolare una situazione generale, è con $n = 2$, ovvero

$$\begin{aligned} p_H(x) &= f[x_0] \cdot 1 \\ &+ f[x_0, x_0](x - x_0) \\ &+ f[x_0, x_0, x_1](x - x_0)^2 \\ &+ f[x_0, x_0, x_1, x_1](x - x_0)^2(x - x_1) \\ &+ f[x_0, x_0, x_1, x_1, x_2](x - x_0)^2(x - x_1)^2 \\ &+ f[x_0, x_0, x_1, x_1, x_2, x_2](x - x_0)^2(x - x_1)^2(x - x_2). \end{aligned}$$

Il risultato dell'osservazione è ottenuto immaginando di avere il doppio delle ascisse mediante l'utilizzo dell'indice $i + \frac{1}{2}$, dove le ascisse con questo indice tendono ad x_i (analogo per le funzioni di base di Newton).

È importante osservare che da $f[x_0]$ è possibile arrivare a $f[x_0, x_0, x_1, x_1, x_2, x_2]$. Quest'ultima differenza divisa ha tutte le ascisse raddoppiate ed è possibile che l'ultimo polinomio di base di Newton abbia tutte le ascisse al quadrato, tranne l' n -esima (altrimenti avrebbe grado $2n + 2$ invece di $2n + 1$).

³¹⁷Slide 5 PDF 18.

³¹⁸Il polinomio interpolante classico è il polinomio costante 0 perché interpola $f(x)$ nelle 3 ascisse di interpolazione. Il polinomio di Hermite interpola anche la deriva prima, ciò rende l'approssimazione più accurata rispetto a quella del polinomio classico.

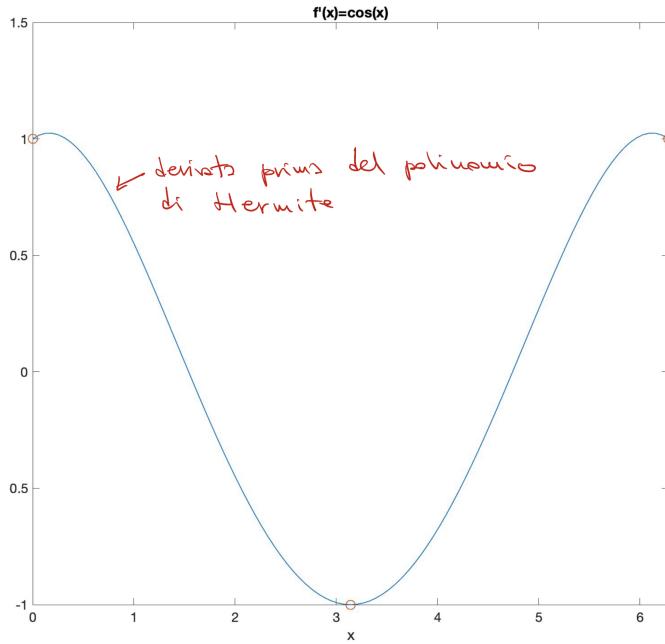


Figura 11: Esempio delle differenze di approssimazione

Osservazione 4.11 (Calcolo $p_H(x)$, caso generale). In generale, per n generico, il polinomio di Hermite sarà ottenuto tramite

$$p_H(x) = f[x_0] \cdot 1 + f[x_0, x_0] \cdot (x - x_0) + \dots + f[x_0, x_0, \dots, x_n, x_n] (x - x_0)^2 \cdot \dots \cdot (x - x_{n-1})^2 (x - x_n). \quad (4.26)$$

È possibile calcolare i coefficienti del polinomio, ovvero le differenze divise, con l'Algoritmo 4.8, una variante dell'Algoritmo 4.3, tenendo di conto di (4.24). Nell'Algoritmo, il vettore in ingresso, \mathbf{f} contiene i valori $f(x_0), f'(x_0), f(x_1), f'(x_1), \dots, f(x_n), f'(x_n)$ ed in uscita, essendo sovrascritto, le differenze divise in (4.26).

Algoritmo 4.8 Polinomio di Hermite: calcolo delle differenze divise

```

for i = (2*n+1):-2:3
    f(i) = (f(i)-f(i-2))/(x(i)-x(i-1))
end
for j = 2 : 2*n+1
    for i = (2*n+2) : -1 : j+1
        f(i) = (f(i)-f(i-1))/(x(i)-x(i-j))
    end
end

```

Inoltre, è possibile calcolare $p_H(x)$ in Matlab tramite una versione modificata dell'algoritmo di Horner generalizzato (Algoritmo 4.5), nella quale è utilizzato un vettore delle ascisse [319] $x = [x_0 \ x_0 \ x_1 \ x_1 \ \dots \ x_{n-1} \ x_{n-1} \ x_n]$, a patto di

conoscere le differenze divise (ovvero i coefficienti del polinomio).

È possibile capire come calcolare le differenze divise tramite il seguente esempio.

Esempio 4.4. Considerando il caso in cui $n = 1$, il polinomio ha grado 3 seguendo la "formula" del calcolo del grado $(2n+1)$, e ritornando al caso in cui sono presenti le ascisse $x_{i+\frac{1}{2}}$ (quindi al caso in cui le ascisse sono duplicate):

	320 0	1	2	3
x_0	$f[x_0]$			
$x_{\frac{1}{2}}$	$f[x_{\frac{1}{2}}]$	$f[x_0, x_{\frac{1}{2}}]$		
x_1	$f[x_1]$	$f[x_{\frac{1}{2}}, x_1]$	$f[x_0, x_{\frac{1}{2}}, x_1]$	
$x_{\frac{3}{2}}$	$f[x_{\frac{3}{2}}]$	$f[x_1, x_{\frac{3}{2}}]$	$f[x_{\frac{1}{2}}, x_1, x_{\frac{3}{2}}]$	$f[x_0, x_{\frac{1}{2}}, x_1, x_{\frac{3}{2}}]$

Le ascisse in **grassetto** sono quelle la cui differenza va a denominatore nel calcolo della rispettiva differenza divisa.

Operando il **limite** per $x_{i+\frac{1}{2}} \rightarrow x_i$:

	0	1	2	3
x_0	$f[x_0]$			
x_0	$f[x_0]$	$f[x_0, x_0]$		
x_1	$f[x_1]$	$f[x_0, x_1]$	$f[x_0, x_0, x_1]$	
x_1	$f[x_1]$	$f[x_1, x_1]$	$f[x_0, x_1, x_1]$	$f[x_0, x_0, x_1, x_1]$

Le differenze divise cerchiate sono le differenze divise per le quali non è noto il metodo di calcolo.

Per un generico n non è possibile calcolare direttamente le differenze divise in colonna 1 $f[x_i, x_i], i = 0, \dots, n$, tuttavia è possibile quanto segue:

$$f[x_i, x_i] = \lim_{x_{i+\frac{1}{2}} \rightarrow x_i} \frac{f(x_{i+\frac{1}{2}}) - f(x_i)}{x_{i+\frac{1}{2}} + x_i} = f'(x_i).$$

Pertanto, è possibile modificare leggermente l'algoritmo classico per il calcolo delle differenze divise (ovvero l'Algoritmo 4.3), in modo che, dove sono presenti ascisse ripetute, sia utilizzato il corrispondente valore della derivata della funzione $f(x)$, come appena esposto. In questo modo sono presenti tutti gli elementi per implementare efficientemente il calcolo del polinomio interpolante di Hermite nella sua forma di Newton.

Osservazione 4.12. Nella tabella precedente, utilizzata per il calcolo delle differenze divise, il vettore delle ascisse utilizzato è del tipo $x = [x_0 \ x_0 \ x_1 \ x_1 \ \dots \ x_n \ x_n]$.

4.4 Errore nell'interpolazione polinomiale

³²¹ È necessario studiare il comportamento della funzione errore e definita come

$$e(x) = f(x) - p(x), \quad x \in [a, b], \tag{4.27}$$

per la quale è possibile osservare che $f(x) = p(x) + e(x)$, con $e(x)$ errore commesso nell'approssimazione di $f(x)$ tramite il polinomio interpolante $p(x)$.

³¹⁹ Le ascisse sono duplicate e per questo, nel ciclo implementato per il calcolo del polinomio, sarà contato fino ad $2n+1$. Il +1 è dovuto al fatto che x_n è ripetuta solo una volta.

³²⁰ Differenza tra l'indice delle ascisse utilizzando la proprietà incrementale delle differenze divise.

³²¹ Slide 2-23 PDF 19, PG 86-87.

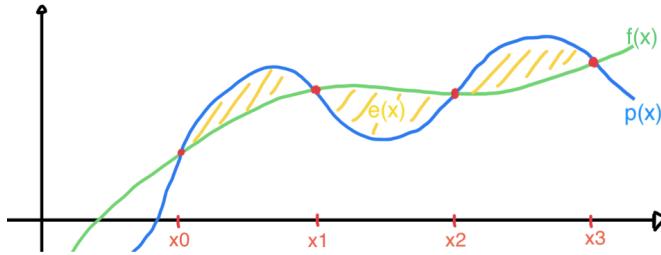


Figura 12: Esempio dell'errore e del polinomio Hermite p rispetto a f .

Osservazione 4.13. ³²² $e(x_i) = f(x_i) - p(x_i) = f(x_i) - f(x_i) = 0$, $i = 0, \dots, n$.

Pertanto, è noto che nelle ascisse di interpolazione l'errore si annulla. È necessario stabilire quanto $e(x)$ sia "distanza" da 0, per $x \notin \{x_0, \dots, x_n\}$.

Teorema 4.4. ³²³ Dato $p(x)$, polinomio interpolante $f(x)$, definito come (4.17), sulle ascisse distinte (definite come in (4.1)), vale:

$$e(x) = f[x_0, x_1, \dots, x_n, x] \omega_{n+1}(x), \quad w_{n+1}(x) = \prod_{j=0}^n (x - x_j). \quad (4.28)$$

Dimostrazione. ³²⁴ Fissato $\hat{x} \notin \overbrace{\{x_0, \dots, x_n\}}^{\text{ascisse d'interp.}}$ un punto generico, il polinomio $\hat{p}(x) \in \Pi_{n+1}$ è costruito come segue:

$$\begin{aligned} \hat{p}(x_i) &= f(x_i), \quad i = 0, \dots, n, \\ \hat{p}(\hat{x}) &= f(\hat{x}), \end{aligned}$$

Ovvero, $\hat{p}(x)$ interpola $f(x)$ anche in \hat{x} , oltre che nelle ascisse x_0, \dots, x_n .

(È imposta la seguente condizione:) Utilizzando la forma di Newton del polinomio interpolante (Teorema 4.3) è ottenuto che

$$\hat{p}(x) = p(x) + f[x_0, \dots, x_n, \hat{x}] \omega_{n+1}(x), \quad (325)$$

la quale soddisfa le condizioni di interpolazione

$$\hat{p}(x_i) = p(x_i) + f[x_0, \dots, x_n, \hat{x}] \underbrace{\omega_{n+1}(x_i)}_{\substack{0 \\ \| \\ f(x_i)}} = f(x_i), \quad i = 0, \dots, n,$$

ed inoltre:

$$\hat{p}(\hat{x}) = p(\hat{x}) + f[x_0, \dots, x_n, \hat{x}] \omega_{n+1}(\hat{x}) = f(\hat{x}).$$

Da quest'ultima uguaglianza è ottenuto che

$$e(\hat{x}) \equiv \underbrace{f(\hat{x}) - p(\hat{x})}_{\text{errore di } \hat{x}} = f[x_0, \dots, x_n, \hat{x}] \omega_{n+1}(\hat{x}).$$

L'aaserto discende dal fatto che \hat{x} è un punto generico. ^[326]

□

³²²Slide 3, PDF 19. Da (4.2) segue l'osservazione.

³²³Slide 3 PDF 19, TH 4.3 PG 86.

³²⁴Sfrutta il fatto che è possibile costruire il polinomio interpolante nella forma di Newton in modo incrementale.

Osservazione 4.14. Il Teorema 4.4 definisce la forma dell'errore di interpolazione polinomiale (e di conseguenza è necessario impararlo).

Corollario 4.4.1.³²⁷ Utilizzando le ipotesi del Teorema 4.4 e supposto che $f \in C^{(n+1)}$ sul più piccolo intervallo contenente le ascisse in argomento alla differenza divisa in (4.28), allora (vedere (4.18)):

$$e(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \omega_{n+1}(x), \quad \xi_x \in [\min_i \{x_i, x\}, \max_i \{x_i, x\}] \equiv I(\mathbf{x}). \quad (4.29)$$

Osservazione 4.15 (Errore polinomio di Hermite).³²⁸ Per il polinomio di Hermite (4.26) interpolante la funzione $f(x)$ ($\in C^{(2n+2)}[I(x)]$), sulle ascisse x_0, \dots, x_n , l'errore corrispondente al polinomio è definito come

$$e_H(x) = f(x) - p_H(x) \stackrel{329}{=} f[\overbrace{x_0, x_0, \dots, x_n, x_n}^{2n+3}] \omega_{n+1}^2(x) \equiv \frac{f^{(2n+2)}(\hat{\xi}_x)}{(2n+2)!} \omega_{n+1}^2(x). \quad (4.30)$$

$e_H(x)$ è di grado $2n+2$.

Osservazione 4.16.³³⁰ Da quanto esposto nell'Osservazione 4.15 è ottenuta una controprova del fatto che:

1. $p(x) \equiv f(x)$, se $f(x) \in \Pi_n$ ($f^{(n+1)} \equiv 0$);
2. $p_H(x) \equiv f(x)$, se $f(x) \in \Pi_{2n+1}$ ($f^{(2n+2)} \equiv 0$).

Osservazione 4.17.³³¹ Osservando la struttura dell'errore (4.29)

$$e(x) = \boxed{\frac{f^{(n+1)}(\xi_x)}{(n+1)!}} \boxed{\omega_{n+1}(x)}$$

(analoghe considerazioni varranno per $e_H(x)$), è possibile osservare che questo è costituito da due parti:

- la prima che dipende da $\boxed{f(x)}$, per cui, più $f(x)$ è regolare, tanto più velocemente questo termine tende a 0, per $n \rightarrow \infty$ (dovuto a $(n+1)!$ al denominatore);
- la seconda, $\boxed{\omega_{n+1}(x) = \prod_{j=0}^n (x - x_j)}$, dipende solo dalla scelta delle ascisse di interpolazione.

È necessario osservare che, per $x > \max_i \{x_i\}$ o per $x < \min_i \{x_i\}$, $\omega_{n+1}(x) \approx x^{n+1}$ (vedere Figura 13). Da questa considerazione è evinto che x debba essere scelta nel più piccolo intervallo che contiene le ascisse di interpolazione (in Figura 13 tra x_0 e x_2). **Quindi sarà scelta $x \in [a, b]$ che contiene le ascisse.**

³²⁵Quando è calcolato in una delle ascisse x_i , si annulla sia il polinomio di interpolazione che la funzione. Il coefficiente $f[x_0, \dots, x_n, \hat{x}]$ permette di ottenere le seguenti condizioni di accuratezza (ovvero le condizioni di interpolazione che seguono).

³²⁶Se al posto di \hat{x} è sostituito x allora $\hat{p}(x)$ continua a variare.

³²⁷Slide 5 PDF 19, Corollario 4.1 PG 87; Importante perché esiste un legame tra le differenze divise e le derivate della funzione f , se questa è sufficientemente regolare.

³²⁸Da sapere. Slide 5 PDF 19, Oss. 4.4 PG 87.

³²⁹Le funzioni di base del polinomio di Newton hanno ascisse raddoppiate.

³³⁰Slide 5 PDF 19.

³³¹Slide 6 PDF 19.

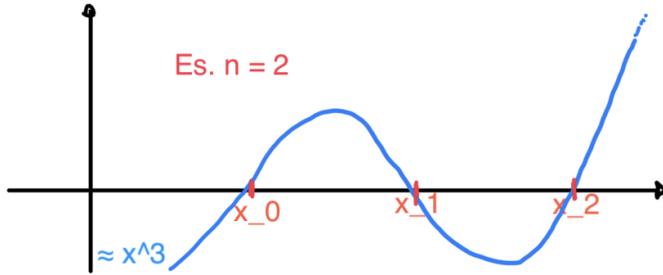


Figura 13: Esempio con numero di ascisse $n = 2$.

Il polinomio interpolante coincide con la funzione f , come per il riquadro principale di $e(x)$ nell'Osservazione 4.17, se f è un polinomio di grado al più n con la sua derivata $(n+1)$ -esima che si annulla identicamente. Quindi, se la funzione f è un polinomio di grado al più n l'errore è identicamente nullo, il che significa che il polinomio interpolante coincide con la funzione di Hermite (ovvero l'uguaglianza (4.30)).

Data l'interpolazione di Hermite su $n+1$ ascisse (il polinomio interpolante di Hermite è di grado $2n+1$), se la funzione interpolante è un polinomio di grado al più $2n+1$, ancora una volta, il polinomio di Hermite coinciderà con la funzione interpolante. Ciò viene confermato dalla parte riquadrata in (4.30), in quanto la derivata $(2n+1)$ -esima di un polinomio di grado $2n+1$ è identicamente nulla.

Con funzioni "buone" è possibile aspettarsi che $p(x)$ approssimi sempre meglio $f(x)$, per $n \rightarrow \infty$.

Come misura dell'errore, negli esempi che seguono, è considerata la norma infinito:

$$\|e\| = \overbrace{\max_{a \leq x \leq b} |f(x) - p(x)|}^{332} \approx \max_{x=a+i \frac{b-a}{N}, i=0, \dots, N} |e(x_i)|, \quad N \gg 1 (N = 10000).$$

$\|e\|$ sarà approssimata, dato che non è possibile calcolarla, con la differenza in valore assoluto, calcolato su un numero molto ampio di ascisse. $\|e\|$ è una sottostima ed è buona se è utilizzato un numero accettabile di punti ($N > 10000$).

Progetto e scelta di N : Per il progetto N deve essere molto grande. Negli esempi richiesti nell'elaborato non saranno ammessi $N < 10000$ (saranno considerati sbagliati). Inoltre, per un polinomio di grado 5 assegnare $N = 3$ sarà considerato sbagliato in quanto la sottostima dell'errore è importante.

Esempio 4.5. Per $f(x) = \sin(x)$ è possibile ottenere ottimi risultati (vedere Figure 14-17), utilizzando un numero di ascisse crescenti ed equidistanti, dato che tutte le derivate di $\sin(x)$ sono limitate.

Date le Figure 14-17, per $f(x) = \sin(x)$, è ottenuto un numero crescente di ascisse equidistanti, dato che le derivate di $\sin(x)$ sono tutte limitate.

È possibile provare, come nel precedente esempio, applicare un numero di ascisse crescenti ed equidistanti, applicando la formula di Runge, alla seguente funzione (grafico Figura 18):

$$f(x) = \frac{1}{1+x^2}, \quad x \in [-5, 5]. \quad (4.31)$$

³³²Norma uniforme, ovvero continua nell'intervallo $[a, b]$.

³³³Che è una norma in $C^{(0)}[a, b]$.

³³⁴Simmetrica rispetto all'asse y e massimo in $x = 0$, vedere Figura 18.

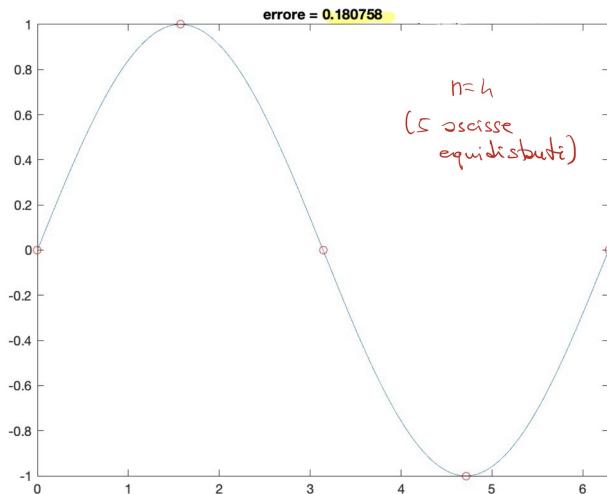


Figura 14: Esempio con numero di ascisse $n = 4$.

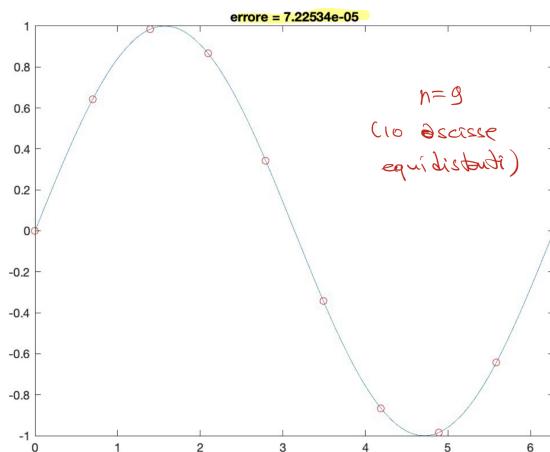


Figura 15: Esempio con numero di ascisse $n = 9$.

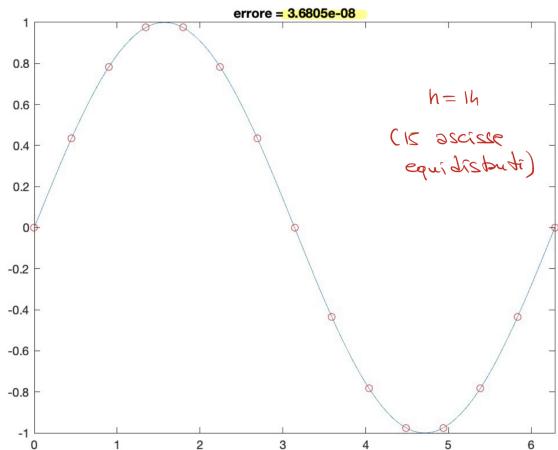


Figura 16: Esempio con numero di ascisse $n = 14$.

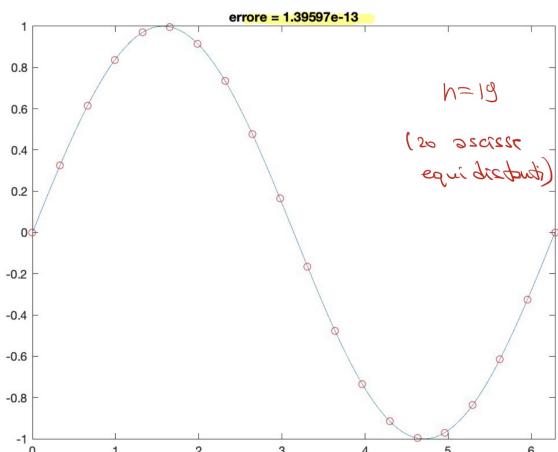


Figura 17: Esempio con numero di ascisse $n = 19$.

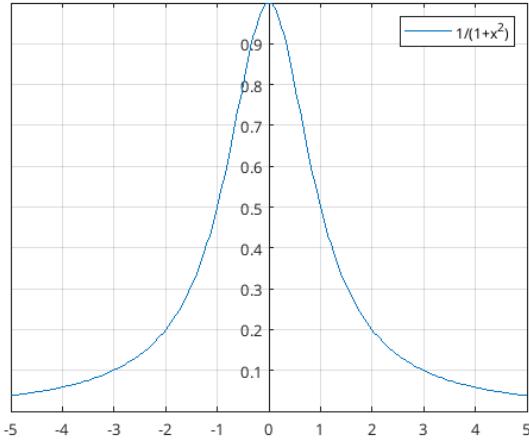


Figura 18: Grafico di (4.31)

Nelle Figure 19-22 [335] è utilizzata la funzione di Runge per approssimare (4.31), con un numero crescente di ascisse equidistanti.

Osservazione 4.18 (Dettagli importanti per il progetto). Distinguere fra numeri pari e dispari di ascisse è importante. Se n è pari allora le ascisse sono dispari. Un esempio è graficare il polinomio (4.31) per un valore pari di n , quindi con un numero dispari di ascisse (ovvero $n + 1$), così da trovare i punti cerchiati in rosso delle Figure 19-22 con un metodo affidabile. Avere un numero di ascisse dispari permette di trovare il massimo della funzione in modo più efficace rispetto ad avere un numero pari di ascisse (vedere Figura 20). Inoltre, nel caso del numero di ascisse pari, l'errore diventa importante (una parte di funzione viene tagliata fuori).

In genere è possibile verificare che $e(x) \rightarrow \infty$, $n \rightarrow \infty$, quindi il problema è malcodizionato. Tuttavia, è necessario studiare il condizionamento del problema dell'interpolazione dove, per ciascuna delle classi di problemi considerati, è noto quale sia il coefficiente di perturbazione del problema.

È necessario studiare il problema del condizionamento dell'interpolazione polinomiale, ovvero: date le ascisse di interpolazione (4.1), è definito il polinomio $p(x) \in \Pi_n$ interpolante la funzione $f(x) \in \Pi_n$ come

$$p(x_i) = f(x_i), \quad i = 0, \dots, n.$$

Considerando una funzione $\tilde{f}(x)$, perturbazione di $f(x)$, e costruito $\tilde{p}(x)$, il polinomio interpolante $\tilde{f}(x)$ sulle ascisse (4.1), come

$$\tilde{p}(x_i) = \tilde{f}(x_i), \quad i = 0, \dots, n,$$

allora sono introdotte le seguenti misure degli errori:

- $\|f - \tilde{f}\|$, misura degli errori sui dati di ingresso;
- $\|p - \tilde{p}\|$, misura degli errori sul risultato (minore è migliore è il condizionamento del problema).

Il problema riguarda anche come sono scelte le ascisse di interpolazione.

³³⁵Nelle figure i puntini rappresentano la funzione di Runge.

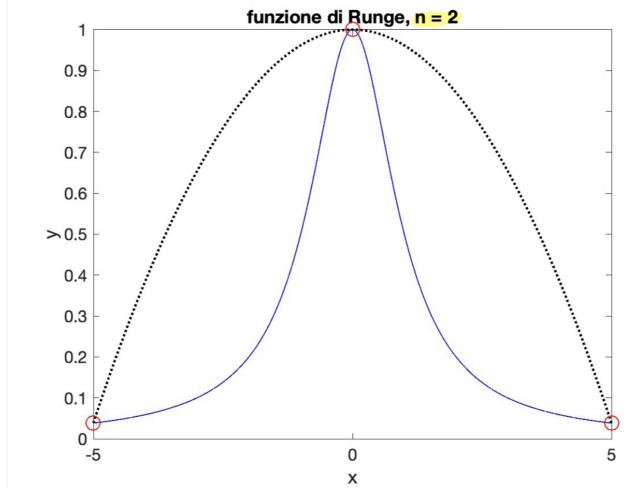


Figura 19: Approssimazione di (4.31)

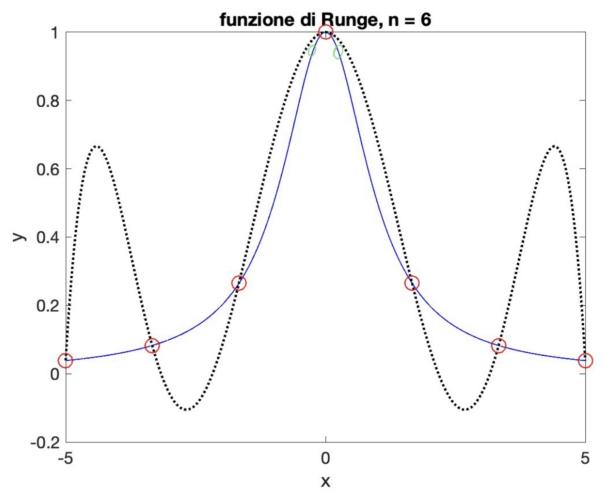


Figura 20: Approssimazione di (4.31)

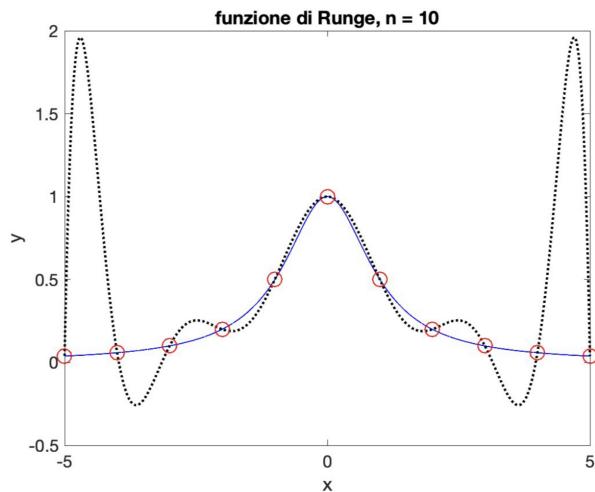


Figura 21: Approssimazione di (4.31)

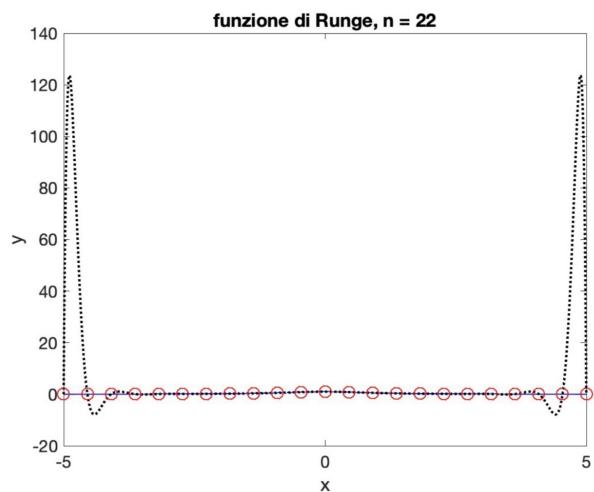


Figura 22: Approssimazione di (4.31)

4.5 Condizionamento del problema

Il problema del condizionamento dell'errore prima citato può essere "trasformato" nello studio di come $\|p - \tilde{p}\|$ dipenda da $\|f - \tilde{f}\|$. Studiare il condizionamento del problema significa stabilire in che modo piccole perturbazioni del problema, ovvero $\|f - \tilde{f}\|$, si riflettano con perturbazioni sul risultato, ovvero su $\|p - \tilde{p}\|$:

- se le perturbazioni sul risultato sono piccole allora il problema è ben condizionato;
- se piccole perturbazioni ne provocano di grandi sul risultato il problema è malcondizionato.

Ciò che sarà studiato è il problema della valutazione del polinomio interpolante.

Intermezzo Lo studio del condizionamento è svolto in aritmetica esatta. Non è svolto in aritmetica finita in quanto sarebbe necessario studiare le singole operazioni elementari e verificare come gli errori si propagano per ognuna di essa. Lo studio del condizionamento è uno studio semplificato della modalità con la quale gli errori si propagano.

Quindi, lo studio deò condizionamento di un problema sarà fatto in aritmetica esatta. Questo significa che qualunque rappresentazione algebrica del polinomio interpolante è equivalente. Ciò non è vero in aritmetica finita, il metodo di calcolo del polinomio influisce sul risultato finale, quindi due metodi di calcolo diversi non forniranno il medesimo risultato.

Al fine di avere una misura degli errori è necessario introdurre la norma infinito su $C^{(0)}[a, b]$:

$$\forall g \in C^{(0)}[a, b] : \|g\| = \max_{a \leq x \leq b} |g(x)|. \quad (4.32)$$

Da quanto appena scritto, per $x \in [a, b]$, è ottenuto che

$$\begin{aligned} |p(x) - \tilde{p}(x)| &\stackrel{337}{=} \left| \sum_{i=0}^n f_i L_{in}(x) - \sum_{i=0}^n \tilde{f}_i L_{in}(x) \right| \stackrel{338}{=} \left| \sum_{i=0}^n (f_i - \tilde{f}_i) L_{in}(x) \right| \\ &\stackrel{339}{\leq} \sum_{i=0}^n |f_i - \tilde{f}_i| |L_{in}(x)| \leq \underbrace{\left(\sum_{i=0}^n |L_{in}(x)| \right)}_{340} \underbrace{\max_{i=0, \dots, n} |f_i - \tilde{f}_i|}_{\lambda_n(x)} \\ &\leq \lambda_n(x) \cdot \|f - \tilde{f}\|, \end{aligned}$$

dove $\lambda_n(x)$ è la **funzione di Lebesgue**. Tale funzione è positiva per definizione e dipende solo dalla scelta delle ascisse di interpolazione. Pertanto, dalla precedente diseguaglianza è ottenuto che

$$\forall x \in [a, b] : |p(x) - \tilde{p}(x)| \leq \lambda_n(x) \|p - \tilde{p}\| \Rightarrow \|p - \tilde{p}\| \leq \|\lambda_n \cdot \underbrace{\|f - \tilde{f}\|}_{341}\| = \underbrace{\|\lambda_n\|}_{\Lambda_n} \cdot \|f - \tilde{f}\|,$$

³³⁶Applicazione del Teorema di Weistrass: esiste un massimo per g perché questa è limitata tra a e b .

³³⁷Utilizzata la forma di Lagrange perché il contributo della funzione è evidenziato rispetto ad un numero generico delle ascisse, i polinomi di base di Lagrange ne usufruiscono.

³³⁸Raccoglimento di $L_{in}(x)$.

³³⁹Applicazione della diseguaglianza triangolare: il modulo di una somma è minore uguale della somma dei loro moduli.

³⁴⁰Massimo dei valori assoluti della differenza $f_i - \tilde{f}_i$, $i = 0, \dots, n$ (ovvero la norma $\|f - \tilde{f}\|$), anche se $x_1, \dots, x_n \in [a, b]$. La differenza $f_i - \tilde{f}_i$ è minore della norma $\|f_i - \tilde{f}_i\|$ che segue nella diseguaglianza.

Inoltre è possibile affermare quanto segue: $\max_{i=0, \dots, n} |f_i - \tilde{f}_i| = \max_{i=0, \dots, n} |f(x_i) - \tilde{f}(x_i)| \leq \max_{a \leq x \leq b} |f(x) - \tilde{f}(x)| = \|f - \tilde{f}\|$.

³⁴¹Costante positiva da portare fuori.

dove $\Lambda_n = \|\lambda_n\|$ è detta **costante di Lebesgue**. Questa costante misura la massima amplificazione sul risultato dell'errore sui dati di ingresso e definisce il numero di condizionamento. Per questo è ottenuto quanto segue:

$$\boxed{\underbrace{\| \frac{f(x)}{p} - \frac{\tilde{f}(x)}{\tilde{p}} \|}_{342} \leq \Lambda_n \underbrace{\| f - \tilde{f} \|}_{343}} \quad (4.33)$$

Data l'importanza della costante di Lebesgue è utile sturdiarne alcune importanti proprietà.

Λ_n è indipendente dal particolare intervallo $[a, b]$ considerato. Questa proprietà è importante perché se i risultati sono contenuti in un uno specifico intervallo di riferimento, allora le ascisse di interpolazione dell'intervallo specifico sono estendibili ad uno generico intervallo che le contiene.

Date $a \leq \underbrace{x_0 < \dots < x_n}_{344} \leq b$, allora

$$\xi = \frac{x-a}{b-a}, \quad (4.34)$$

dove ξ è una trasformazione lineare, per $x = a \Rightarrow \xi = 0$ e per $x = b \Rightarrow \xi = 1$. Pertanto, se $x \in [a, b] \Rightarrow \xi \in [0, 1]$; viceversa, se $\xi \in [0, 1] \Rightarrow x = a + (b-a)\xi$, $x \in [a, b]$.

Alle ascisse di interpolazione $x_i \in [a, b]$ corrispondono

$$\xi_i = \frac{x_i-a}{b-a} \in [0, 1], \quad \forall i = 0, \dots, n. \quad (4.35)$$

È possibile provare che i polinomi di base di Lagrange, definiti sulle ascisse $\{x_i\}$, coincidano con quelli costruiti sulle ascisse $\{\xi_i\}$, attraverso le prossime uguaglianze:

$$\begin{aligned} L_{in}(x) &= \prod_{j=0, j \neq i}^n \frac{x-x_j}{x_i-x_j} & 345 \\ L_{in}(\xi) &= \prod_{j=0, j \neq i}^n \frac{\xi-\xi_j}{\xi_i-\xi_j} & 346 \\ \prod_{j=0, j \neq i}^n \frac{x-a-(x_j-a)}{x_i-a-(x_j-a)} &= \prod_{j=0, j \neq i}^n \frac{\frac{x-a}{b-a}-\frac{x_j-a}{b-a}}{\frac{x_i-a}{b-a}-\frac{x_j-a}{b-a}} & 347 \\ &= L_{in}(x). \end{aligned}$$

□

L'intervallo di riferimento non è importante, in quanto è possibile definire le seguenti proprietà, le quali valgono considerando solo il numero delle ascisse, ovvero:

- P1) ridefinendo la norma sull'intervallo $[0, 1]$, la costante di Lebesgue rimane invariata. Questo è dovuto dal fatto che, se i polinomi coincidono, allora anche la costante di Lebesgue, ottenuta come somma dei due polinomi, non dipende dall'intervallo $[a, b]$. L'unica differenza è nella definizione di norma perché definita per $x \in [a, b]$;
- P2) qualunque sia la scelta delle ascisse (distinte tra loro), è noto che $\Lambda_n \geq O(\log n)$;

³⁴²Misura dell'errore sul risultato.

³⁴³Misura degli errori in ingresso.

³⁴⁴Saranno trasformati linearmente sull'intervallo $[0, 1]$.

³⁴⁵Sarà trasformato in $x = a + (b-a)\xi$, ovvero $\xi = \frac{x-a}{b-a}$.

³⁴⁶Per (4.34) $\stackrel{e}{+}$ (4.35).

³⁴⁷Evidenziato $(b-a)$ e semplificato.

- P3) dalla precedente proprietà deriva che $\Lambda_n \rightarrow \infty$, $n \rightarrow \infty$, quindi il problema diviene malcondizionato al crescere di n ;
- P4) la scelta di ascisse equidistanti da una costante di Lebesgue genera una successione $\{\Lambda_n\}$, la quale diverge esponenzialmente con $n \rightarrow \infty$. Pertanto, la scelta di ascisse equidistanti non è, in generale, appropriata.

4.5.1 Connessioni tra condizionamento ed errore dell'interpolazione polinomiale

³⁴⁸Allo scopo di studiare le connessioni tra condizionamento ed errore dell'interpolazione polinomiale sarà assunto di aver fissato un intervallo di riferimento $[a, b]$, la corrispondente norma $\|\cdot\|$ ed una funzione $f(x) \in C^{(0)}[a, b]$. Lo studio che sarà trattato varrà per ogni intervallo, in quanto la proprietà P3) del precedente elenco è invariante rispetto all'intervallo considerato.

Definizione 4.6 (Errore e polinomio di migliore approssimazione). ³⁴⁹ $\forall n \geq 0$, $\exists p^* \in \Pi_n$, un polinomio, tale che:

$$\|f - p^*\| = \min_{p \in \Pi_n} \|f - p\|, \quad (4.36)$$

dove $p^*(x)$ è detto **polinomio di migliore approssimazione di $f(x)$, di grado n (sull'intervallo $[a, b]$)** e $\|f - p^*\|$ è detto **l'errore di migliore approssimazione**.

Osservazione 4.19. È possibile trattare (4.36) come massimo errore commesso, approssimando f con il polinomio di migliore approssimazione di f . L'errore d'interpolazione polinomiale con il polinomio dello stesso grado è legato all'errore di miglior approssimazione.

La Definizione 4.6 denota l'esistenza di una funzione con un minimo sull'intervallo $[a, b]$ e fissa il grado n , ovvero il grado dell'insieme di polinomi (Π_n) con i quali f sarà approssimata. Fra tutti i polinomi di grado n che approssimano la norma, $\|f - p\|$ è non minore di $\|f - p^*\|$ (dove $p^* \in \Pi_n$). Esiste un polinomio della migliore approssimazione che approssima f meglio degli altri.

Dato $p(x)$, polinomio interpolante $f(x)$ di grado n , è trattato come l'errore di interpolazione sia legato all'errore di migliore approssimazione (4.36). A questo scopo è definito $\|e\| = \|f - p\|$, ovvero **l'errore massimo di interpolazione**.

Teorema 4.5. ³⁵⁰ Sia $p^*(x)$ il polinomio di migliore approssimazione di grado n di $f(x)$, allora, per l'errore di interpolazione (4.27), vale:

$$\|e\| \leq (1 + \Lambda_n) \overbrace{\|f - p^*\|}^{351}, \quad (4.37)$$

dove Λ_n è la costante di Lebesgue, definita sulle ascisse di interpolazione.

Dimostrazione. Considerando $p^* \in \Pi_n$:

$$\begin{aligned} \|e\| &= \|f - p^* + p^* - p\| \\ &\leq \|f - p^*\| + \|p^* - p\| \\ &\leq \|f - p^*\| + \Lambda_n \|f - p^*\| = (1 + \Lambda_n) \|f - p^*\|. \end{aligned}$$

³⁴⁸Cose da non dimostare (tranne il teorema di Jackson), ma da conoscere perché utili per le future trattazioni di diversi tipi di interpolazione, le quali si giustificano per i risultati diversi di questa analisi.

³⁴⁹Slide 7 PDF 20, Definizione 4.2 + Teorema 4.6 PG 90.

³⁵⁰Slide 8 PDF 20, Teorema 4.7 PG 90

³⁵¹Quantità che tende a 0 quando n cresce (lentamente in generale).

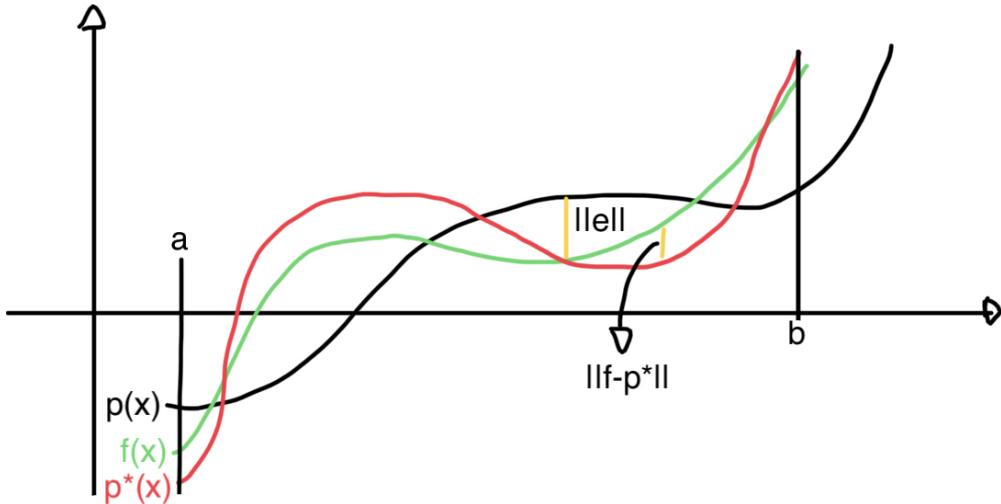


Figura 23: Esempio grafico della approssimazione p e dell'errore (4.36) conseguente.

Su $p^*(x)$ sono possibili le seguenti considerazioni:

- coincide con il suo polinomio interpolante sulle $n + 1$ ascisse sul quale è definito $p(x)$;
- può essere interpretato come una perturbazione $\hat{f}(x)$ di $f(x)$.

□

È necessaria un'approssimazione dell'errore di migliore approssimazione (4.36) in quanto l'errore crescerà al crescere di n , se la costante di Lebesgue diverge esponenzialmente. Per questo è fornita una maggiorazione per (4.36), introducendo il modulo di continuità di una funzione.

Definizione 4.7 (Modulo di continuità di una funzione). Data $f \in C^{(0)}[a, b]$ è definito il modulo di continuità di f , per $h > 0$ (variazione), come:

$$\omega(f; h) = \left\{ \sup_{x, y \in [a, b]} |f(x) - f(y)| : |x - y| \leq h \right\}. \quad (4.38)$$

Teorema 4.6.³⁵³ Se $f \in C^{(0)}[a, b] \Rightarrow \omega(f; h) \rightarrow 0$, per $h \rightarrow 0$.

Definizione 4.8 (Polinomio di migliore approssimazione). Data $f \in C^{(0)}[a, b]$ allora è definito polinomio di migliore approssimazione di grado n di f su $[a, b]$:

$$p^* = \underset{p \in \Pi_n}{\arg \min} \|f - p\|. \quad (4.39)$$

Dato $p(x) \in \Pi_n$, ovvero un polinomio interpolante $f(x)$ su $n + 1$ ascisse in $[a, b]$, valgono i due seguenti risultati.

³⁵²Aggiunto il polinomio p^* per il quale vale la diseguaglianza triangolare.

³⁵³Slide 10 PDF 20, primo punto PG 91

³⁵⁴ $\arg \min_{x \in X} f(x) = \{x \mid \forall y : f(y) \geq f(x)\}$. In altre parole, è l'insieme dei valori di x per i quali $f(x)$ raggiunge il suo più alto valore M .

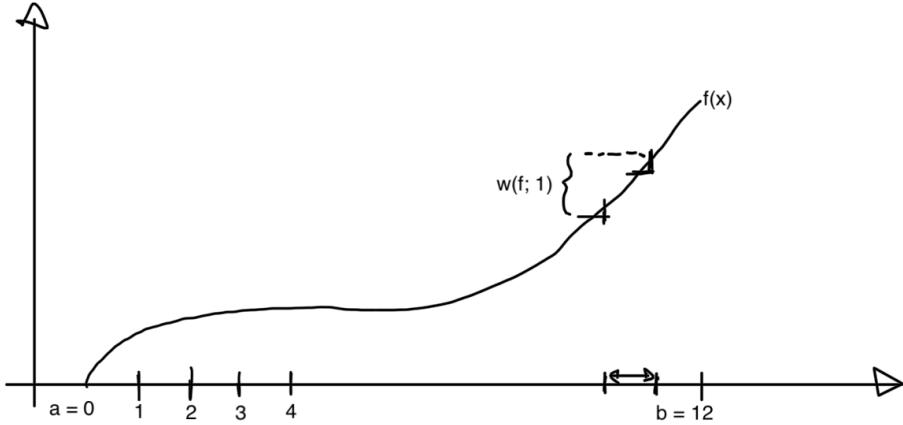


Figura 24: Esempio modulo di continuità. $\omega(f; 1)$ è la massima distanza sul grafico di f tra i punti.

Teorema 4.7 (Jackson).³⁵⁵ Se $f \in C^{(0)}[a, b]$ e p^* è il suo polinomio di migliore approssimazione di grado n , allora $\exists \alpha$, indipendente da n , tale che:

$$\|f - p^*\| \leq \alpha \cdot \omega \left(f; \frac{b-a}{n} \right). \quad (4.40)$$

Corollario 4.7.1. Se $f \in C^{(0)}[a, b]$, allora

$$\|f - p_n^*\| \rightarrow 0, \quad n \rightarrow \infty,$$

essendo $\{p_n^*\}$ la successione dei problemi di migliore approssimazione di f di grado n .

È possibile affermare che esiste una relazione fra l'errore nell'approssimazione polinomiale ed il condizionamento del problema (dato il Teorema di Jackson).

Corollario 4.7.2.³⁵⁸ $\forall n \geq 0$, data una funzione $f(x)$ generica, $p(x)$ il polinomio interpolante di grado n su $n+1$ ascisse assegnate e Λ_n la corrispondente costante di Lebesgue, vale che:

$$\boxed{\|e\| = \|f - p\| \leq \underbrace{\alpha_{359} (1 + \Lambda_n) \omega \left(f; \frac{b-a}{n} \right)}_{360}.} \quad (4.41)$$

³⁵⁵Slide 10 PDF 20, Teorema 4.8 PG 91

³⁵⁶Aampiezza dell'intervallo sul quale approssimiamo f .

³⁵⁷Grado del polinomio di migliore approssimazione. Più questo cresce più $\omega \rightarrow 0$, quindi $\|f - p\| > 0$.

³⁵⁸Slide 10 PDF 20, PG 91. Dalle (4.37)-(4.40) vale quanto segue.

³⁵⁹Non dipende da n , ovvero il grado del polinomio approssimante.

³⁶⁰Definita nella dimostrazione del Teorema 4.5.

Quest'ultima espressione sarà utile per introdurre nuovi metodi di interpolazione. Inoltre, è necessario che la costante Λ_n non cresca più rapidamente della quantità per la quale è moltiplicata, ma tenda a 0, al crescere del numero di ascisse di interpolazione.

Osservazione 4.20. ³⁶¹ Al crescere del numero delle ascisse di interpolazione ($n + 1$) è necessario fare in modo che Λ_n cresca in modo ottimale, affinché $\|e\|$ diminuisca.

4.6 Ascisse di Chebyshev

³⁶² Le Sezioni 4.4 e 4.5 sono riassumibili nei seguenti punti:

- Λ_n è indipendente dal particolare intervallo considerato; cresce esponenzialmente con n se sono utilizzate ascisse equidistanti, altrimenti ha una crescita moderta (quella ottimale è di tipo logaritmico) rispetto ad n ; ³⁶³

- Data la maggiorazione $\|e\| \leq \overbrace{\frac{\|f^{(n+1)}\|}{(n+1)!}}^{364} \|\omega_{n+1}\|$, con $\omega_{n+1}(x) = \prod_{j=0}^n (x - x_j)$, è necessario capire come questa sia collegata con (4.41).

L'obiettivo della Sezione è quello di scegliere le ascisse di interpolazione x_j in modo da minimizzare $\|\omega_{n+1}\|$. Per fare questo è necessario rendere la costante di Lebesgue piccola o almeno con una precisione accettabile.

Considerando l'intervallo di riferimento $[-1, 1]$, è necessario scegliere le ascisse x_0, \dots, x_n in modo tale da risolvere il problema del **minmax**, ovvero il seguente: ³⁶⁵

$$\min_{a=-1 \leq x_0 < \dots < x_n \leq b} \|w_{n+1}\| = \min_{a=-1 \leq x_0 < \dots < x_n \leq 1=b} \max_{-1 \leq x \leq 1} |\overbrace{w_{n+1}(x)}^{366}|, \quad (4.42)$$

con $w_{n+1}(x)$ calcolata come in (4.12), ovvero $\prod_{j=0}^n (x - x_j)$.

A questo scopo è introdotta la seguente Definizione:

Definizione 4.9 (Polinomi di Chebyshev di I specie). Assunto $x \in [-1, 1]$, la **famiglia di polinomi di Chebyshev di I specie** è definita come segue:

$$\begin{cases} T_0(x) \equiv 1; \\ T_1(x) = x; \\ T_{k+1}(x) = 2x \cdot T_k(x) - T_{k-1}(x), \quad k \geq 1. \end{cases} \quad (4.43)$$

Esempio 4.6.

$$\begin{aligned} T_2 &= 2x \cdot \underbrace{x}_{T_1(x)} - \underbrace{1}_{T_0(x)} = 2x^2 - 1, \\ T_3(x) &= 2x \cdot T_2(x) - T_1(x) = 2x(2x^2 - 1) - x = 4x^3 - 2x - x = 4x^3 - 3x, \\ &\vdots \end{aligned}$$

□

³⁶¹ Slide 10 PDF 20.

³⁶² PDF 21, PG 91-93.

³⁶³ n è il grado del polinomio interpolante.

³⁶⁴ Indipendente dalle ascisse, non come $\|\omega_{n+1}\|$.

³⁶⁵ Slide 5 PDF 21, PG 91.

³⁶⁶ Polinomio monico che ha come radici le ascisse, rispetto alle quali è svolta la minimizzazione.

Alcune proprietà dei polinomi di Chebyshev di I specie sono le seguenti:

P1) $T_k(x)$ è un **polinomio di grado esatto k** , $\forall k = 0, 1, \dots$;

P2) ³⁶⁷Il coefficiente principale di $T_k(x)$ è 2^{k-1} , $\forall k = 1, 2, \dots$;

P3) ³⁶⁸I polinomi $\{\widehat{T}_k\}_{k \geq 0}$ definiti come

$$\begin{cases} \widehat{T}_0(x) = T_0 \equiv 1, \\ \widehat{T}_k(x) = 2^{1-k} T_k(x), \quad k = 1, 2, \dots \end{cases}$$

sono una **famiglia di polinomi monici** (perché 2^{1-k} è il reciproco del precedente coefficiente 2^{k-1});

P4)

$$\forall k \geq 1 : \widehat{T}_k(x) = \prod_{j=0}^{k-1} (x - x_j^{(k)}),$$

con $T_k(x_j^{(k)}) = 0$, $j = 0, \dots, k-1$, dove $x_j^{(k)}$ sono le **radici** di $T_k(x)$. ³⁶⁹

Osservazione 4.21. ³⁷⁰ Considerata $\widehat{T}_{n+1}(x) = \prod_{j=0}^n (x - x_j^{(n+1)})$, dove $x_j^{(n+1)}$, $j = 0, \dots, n$, sono le radici di $T_{n+1}(x)$ e date le seguenti condizioni sulle radici:

1. $x_i^{(n+1)} \neq x_j^{(n+1)}$, $i \neq j$, $i, j = 0, \dots, n$; ³⁷¹

2. $x_j^{(n+1)} \in [-1, 1]$, $\forall j = 0, \dots, n$; ³⁷²

allora è possibile sceglierle come ascisse di interpolazione, per cui:

$$\omega_{n+1} \equiv \widehat{T}_{n+1} = \prod_{j=0}^n (x - x_j^{(n+1)}).$$

Per verificare che 1. e 2. siano soddisfatte, sarà ottenuta un'espressione esplicita delle ascisse stesse. Poiché $x \in [-1, 1]$, è posto $x = \cos(\theta)$, $\theta \in [0, \pi]$, allora $\theta = \arccos x$ (vedere Figura 25).

P5) Se $x = \cos \theta \Rightarrow T_k(x) = T_k(\cos \theta) = \cos k\theta$ ³⁷³.

³⁶⁷A cosa serve questa proprietà? Dato il polinomio $4x^2 - 1$, è necessario dividerlo per 4 per renderlo monico (4 è il coefficiente principale del polinomio). Data una famiglia di polinomi di grado k , se è necessario derivare una famiglia di polinomi monici è necessario conoscere qual è il coefficiente principale di ciascuno di loro.

³⁶⁸Dati i polinomi di grado k con 8 coefficienti principali, è possibile derivare da questa una famiglia di polinomi monici.

³⁶⁹Se le radici $x_j^{(k)}$ fossero tutte distinte e nell'intervallo $[-1, 1]$, allora per il polinomio $\widehat{T}_k(x)$ è possibile scegliere un'ascissa di interpolazione.

³⁷⁰Slide 7 PDF 21.

³⁷¹Definisce le ascisse distinte.

³⁷²Definisce le ascisse in un intervallo.

³⁷³Polinomio di Chebyshev, se x è parametrizzata in modo idoneo.

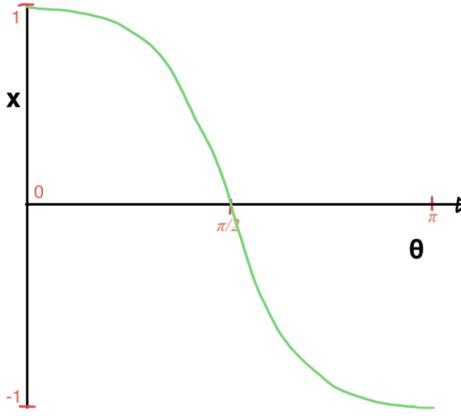


Figura 25: Grafico osservazione 4.21.

Dimostrazione. È svolta per induzione su k .

$$\begin{aligned} k=0 &\Rightarrow T_0(x) = T_0(\cos \theta) = \cos(0 \cdot \theta) \equiv 1, \\ k=1 &\Rightarrow T_1(x) = T_1(\cos \theta) = \cos \theta = x, \end{aligned}$$

supposto vero per k e $k-1$ è dimostrato per $k+1$ ciò che segue:

$$\begin{aligned} T_{k+1}(x) &= 2x T_k(x) - T_{k-1}(x) \\ &= 2\cos\theta \cdot \cos(k\theta) - \cos\theta \cdot \cos(k\theta) - \sin\theta \sin(k\theta) \\ &= \cos\theta \cdot \cos(k\theta) - \sin\theta \cdot \sin(k\theta) \\ &= \cos((k+1)\theta) \end{aligned}$$

□

P6) ³⁷⁴ $\|T_k\| = \max_{\theta \in [0, \pi]} |\cos k\theta| \stackrel{375}{=} 1, \forall k \geq 0;$

P7) $\|\widehat{T}_k\| = 2^{1-k} \|T_k\| = 2^{1-k}, \forall k \geq 1$, con \widehat{T}_k detto polinomio monicizzato;

P8) Gli zeri di $T_k(x)$ sono ottenuti imponendo che $x = \cos\theta, \theta \in [0, \pi]$ e

$$\begin{aligned} T_k(x) &= T_k(\cos\theta) = \cos(k\theta) = 0 \\ &\stackrel{376}{\Rightarrow} k\theta = \frac{\pi}{2} + i\pi \\ &\Rightarrow \theta = \frac{(2i+1)\pi}{2k}, i = 0, \dots, k-1 \\ &\Rightarrow x_i^{(k)} = \cos\left(\frac{(2i+1)\pi}{2k}\right), i = 0, \dots, k-1. \end{aligned}$$

con le ascisse risultanti reali, distinte tra loro e appartenenti a $(-1, 1)$.

³⁷⁴Deriva dalla precedente proprietà.

³⁷⁵In $[0, \pi]$ il massimo vale 1.

³⁷⁶ $\cos(k\theta)$ si annulla per $k\theta$ come segue.

Definizione 4.10 (Ascisse di Chebyshev). ³⁷⁷ Dato il polinomio interpolante f di grado n , le ascisse di Chebyshev sono definite come

$$x_i = \cos\left(\frac{(2i+1)\pi}{2(n+1)}\right), \quad i = 0, \dots, n, \quad (4.44)$$

le quali sono le radici di $\widehat{T}_{n+1}(x)$.

Dato che le ascisse di Chebyshev, appena definite, sono radici di $\widehat{T}_{n+1}(x)$ allora la norma

$$\|\omega_{n+1}\| = \|\widehat{T}_{n+1}\| = 2^{-n} \|T_{n+1}\| = 2^{-n} 1 = 2^{-n}$$

è **minima** tra tutti **polinomi monici di grado $n + 1$** .³⁷⁸ Pertanto, il polinomio di Chebyshev monicizzato di grado $n + 1$ è soluzione del prodotto min max (4.42).

Inoltre, con la scelta delle ascisse di Chebyshev come ascisse di interpolazione è minimizzata la maggiorazione dell'errore (4.40), portando la costante di Lebesque a $\Lambda_n \approx \frac{2}{\pi} \log n$. Quindi, la scelta delle ascisse di Chebyshev permette una crescita **ottimale** della costante di Lebesque, per $n \rightarrow \infty$.

Ricapitolando: Sono scelte le ascisse di interpolazione, nell'intervallo $[-1, 1]$, distinte fra loro e che utilizzano la norma del polinomio monico (il quale ha come radici le ascisse del tipo x_i definite in (4.44)).

È stato dimostrato che $\|w_{n+1}\| = 2^{-n}$, con n ottenuto attraverso le ascisse di interpolazione e le ascisse del polinomio di Chebyshev di grado n .

Le ascisse del polinomio di Chebyshev di grado $n + 1$ ha $n + 1$ radici (necessario che un polinomio di Chebyshev di grado $n + 1$ abbia $n + 1$ radici); con $n + 1$ ascisse di interpolazione è possibile definire il polinomio interpolante di grado n . Quindi, se è richiesta una function per calcolare le ascisse relative al grado n , è neceessario ricordare di contare fino ad $n + 1$ per Matlab, altrimenti sono contati $n - 1$ (quindi il polinomio interpolante sarebbe di grado $n - 1$).

Dalla Figura 26 è possibile notare che sopra il grado 40 non vengono considerati i polinomi. Questo è dovuto al fatto che in aritmetica finita sono introdotti errori e molcondizionamento nelle operazioni. Inoltre, è introdotta la cancellazione numerica nel caso di differenza con ascisse di interpolazione sempre più vicine.

Osservazione 4.22. Le ascisse di Chebyshev forniscono delle ascisse di interpolazione ottimali nell'intervallo $[-1, 1]$. Per riportare ad un generico intervallo $[a, b]$ quanto descritto per (4.44), è possibile verificare facilmente che è sufficiente utilizzare la trasformazione lineare:

$$x_{n-i} = \underbrace{\frac{a+b}{2}}_{379} + \frac{b-a}{2} \cos\left(\frac{(2i+1)\pi}{2(n+1)}\right), \quad i = 0, \dots, n,$$

dove è utilizzato l'indice di enumerazione in notazione $n - i$, quindi rovesciato, per ottenere le ascisse in ordine crescente.

Utilizzando le ascisse di Chebyshev nell'intervallo $[-5, 5]$ sono ottenute le Figure 27-30. Nelle figure è possibile notare che, da $n = 10$ (vedere Figura 28), la precisione di approssimazione è buona. Nella precedente approssimazione di $f(x) = \frac{1}{1+x^2}$ tramite la funzione di Runge, è possibile notare come, con 10 e più ascisse, erano presenti oscillazioni enormi agli estremi (vedere Figura 21 e 22). Inoltre, è possibile notare come le ascisse addensate agli estremi dell'intervallo sono più vicine l'una alla altra di quanto lo siano quelle centrali.

È possibile trovare un'implementazione dell'algoritmo che calcola le ascisse di Chebyshev, per costruire il polinomio interpolante di grado n , su un generico intervallo $[a, b]$, nell'Algoritmo 8.1.

³⁷⁷Slide 10 PDF 21, TH 4.9 PG 92.

³⁷⁸Comunque siano date le ascisse in $[-1, 1]$ è ottenuto un valore più grande del min max (4.42).

³⁷⁹Punto medio dell'intervallo.

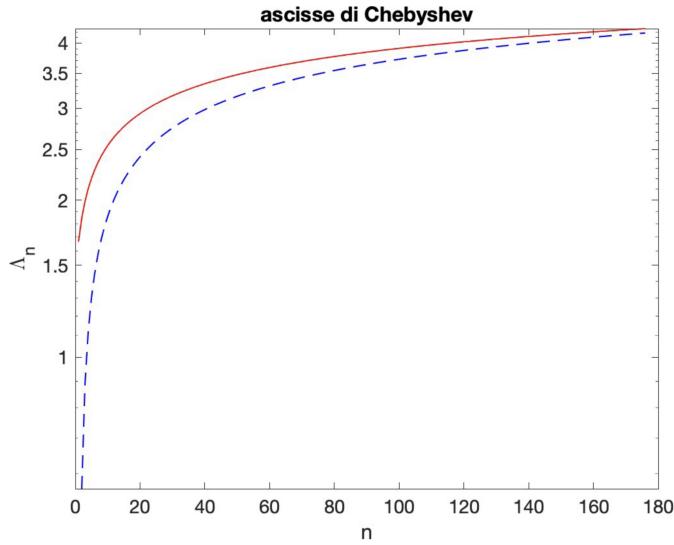


Figura 26: Esempio di crescita di Λ_n al crescere di n .

Osservazione 4.23. È necessario ricordare che al crescere del grado n del polinomio interpolante, la scelta delle ascisse di interpolazione non deve fare crescere troppo la costante di Lebesque Λ_n .

4.7 Interpolazione mediante funzioni spline

³⁸⁰ Dalle Figure 27-30 è possibile ottenere che l'approssimazione migliora all'aumentare del numero di ascisse. Inoltre, è necessario notare che 40 è il numero limite di ascisse per il quale l'approssimazione non migliora, anche se queste sono aumentate. Questo è dovuto al fatto che il numero di condizionamento, pur non essendo particolarmente grande, è influenzato dagli errori di round-off.

In analisi matematica i problemi sono concepiti in aritmetica infinita (ovvero esatta), ma gli algoritmi sono in aritmetica finita (il problema rappresentato è perturbato, in quanto non è quello reale). Al crescere della grado del polinomio interpolante, le ascisse di Chebyshev evidenziano i limiti dell'utilizzo dell'aritmetica finita, dove il problema è quello di valutare un polinomio di grado elevato in aritmetica finita.

La necessità, quindi, è quella di definire un algoritmo poco sensibile a perturbazioni. Se il numero di ascisse è fatto tendere all'infinito allora l'errore tenderà a 0, ma se è considerato l'errore di round-off accade che gli errori, dovuti all'approssimazione, sono più piccoli degli errori di round-off e sotto questo livello non è possibile scendere. Lo scopo è quello di fornire una soluzione a questo problema tramite ciò che sarà introdotto.

Il punto di partenza per la definizione dell'algoritmo prima citato è il Teorema di Jackson (Teorema 4.7), il quale lega l'errore di approssimazione al condizionamento del problema tramite la maggiorazione (4.41). Il problema con la maggiorazione (4.41) è il seguente: aumentando il grado n del polinomio, il modulo di continuità ω è reso sempre più piccolo e non è assicurato che non possa essere migliore (così facendo è possibile che venga modificata la decrescenza del modulo di continuità). Tuttavia, se n è piccolo e l'intervallo $[a, b]$ è fissato, ω non potrà tendere a 0.

In alternativa all'approccio classico è possibile definire quanto segue.

³⁸⁰Slide 10-15 PDF 22, Slide 2-3 PDF 23, PG 94 - 100.

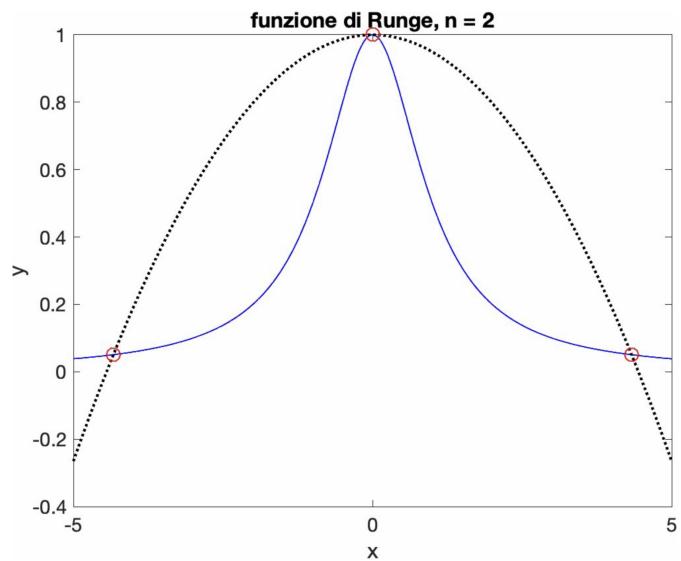


Figura 27: Esempio di crescita di Λ_n al crescere di n .

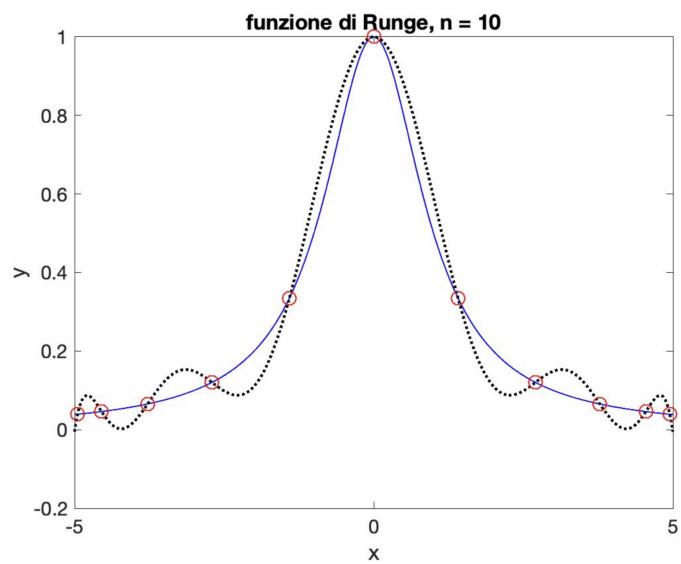


Figura 28: Esempio di crescita di Λ_n al crescere di n .

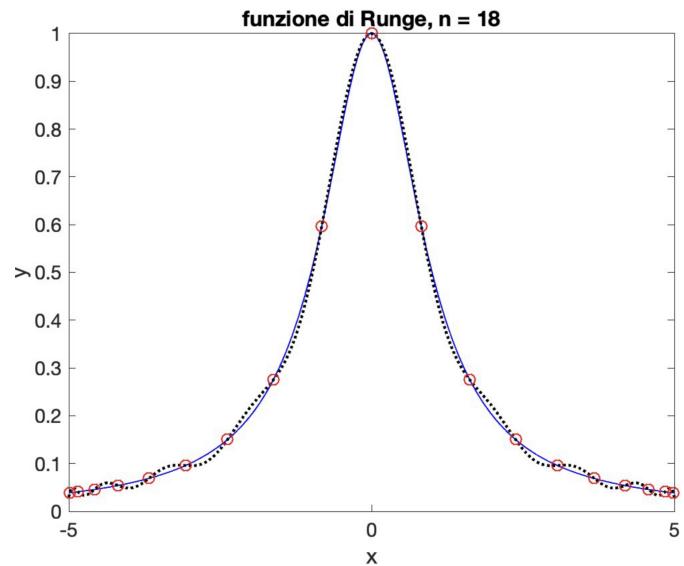


Figura 29: Esempio di crescita di Λ_n al crescere di n .

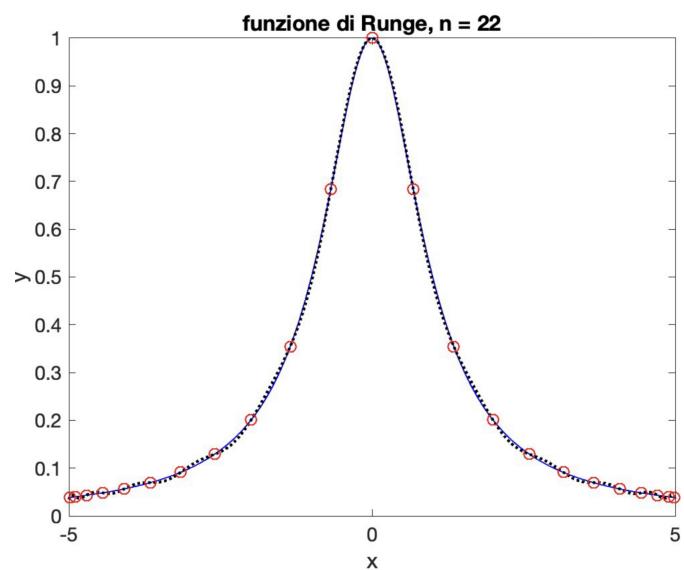


Figura 30: Esempio di crescita di Λ_n al crescere di n .

Definizione 4.11 (Partizione di insieme). Dato $[a, b]$ dominio di f , una partizione Δ sull'intervallo $[a, b]$ è definita come

$$\Delta = \{\mathbf{a} = x_0 < x_1 < \dots < x_n = \mathbf{b}\}, \quad (4.45)$$

la quale contiene $n + 1$ punti (ovvero ascisse).

Definizione 4.12 (Condizione di uniformità della partizione). La condizione di uniformità della partizione Δ (definita come (4.45)) è definita come

$$h = \max_{i=1,\dots,n} (x_i - x_{i-1}). \quad (4.46)$$

È assunto che $h \rightarrow 0$, $n \rightarrow \infty$. Inoltre, è assunto che su ciascun sottointervallo $[x_{i-1}, x_i]$, $i = 1, \dots, n$, di Δ è utilizzato un polinomio di **grado m fissato**, interpolante $f(x)$ agli estremi del sottointervallo. Quindi, la nuova funzione interpolante è polinomiale a tratti (ovvero, in ogni sottointervallo è presente un polinomio). Così facendo, (4.41) diviene

$$||e|| \leq \underbrace{\alpha(1 + \Lambda_m)}_{\text{non varia}} \underbrace{\omega\left(f; \frac{h}{m}\right)}_{\text{tende a } 0}, \quad (4.47)$$

dove:

- m è fissato,
- $h \rightarrow 0$, se $n \rightarrow \infty$,

con m ed n svincolati.

³⁸³ Date le due precedenti definizioni, il problema del condizionamento diviene meno importante perché m è fissato mentre, se $f \in C^{(0)}$ (vedi (4.38)), vale il Teorema 4.6.

Con una lente più formale è possibile definire quanto segue:

Definizione 4.13 (Spline di m su Δ). ³⁸⁴ $s_m(x)$ è una funzione definita come spline di grado m sulla partizione Δ , la quale è definita come (4.45), se:

1. $s_m(x) \in C^{(m-1)}[a, b];$ ³⁸⁵
2. $s_m|_{[x_{i-1}, x_i]}(x) \in \Pi_m, \forall i = 1, \dots, n.$

1. e 2. sono condizioni diverse: 1. definisce cos'è una spline di grado m e 2. afferma che la spline coincide con il grado assegnato in ciascun sottointervallo. Una spline interpolante soddisfa le condizioni di interpolazione solite (ovvero $n + 1$). È necessario imporre sulla 2. la condizione (4.49).

Osservazione 4.24. ³⁸⁶ Denotando con $S_m(\Delta)$ l'insieme delle spline di grado m sulla partizione Δ , contenente $n + 1$ ascisse, questo è uno **spazio vettoriale** di dimensione $\mathbf{m} + \mathbf{n}$.

³⁸¹ Massima ampiezza tra i sottointervalli e la partizione. Ciò che è assunto è: aumentando il numero dei punti $h \rightarrow 0$, quindi non rimangano zone dell'intervallo $[a, b]$ che, se il numero delle ascisse tende all'infinito, sono senza punti. Se sono aggiunti punti questi sono aggiunti per tutti. Una partizione uniforme garantisce queste proprietà, ovvero (4.46).

³⁸²Se f e p hanno grado m , significa che la costante di Lebesgue si trasforma in Λ_m , la quale è fissata, non cambia, ed al posto di $\frac{b-a}{m}$ è inserita la massima ampiezza dell'intervallo.

³⁸³La strategia è dividere il problema in tanti sottoproblemi, ciascuno di essi utilizza un polinomio interpolante di grado fissato m solo agli estremi dell'intervallo.

³⁸⁴Definizione 1 Slide 11 PDF 22, Definizione 4.3 PG 94.

³⁸⁵ s_m deve essere derivabile $m - 1$ volte e le $m - 1$ derivate devono essere continue. Se s_m è un polinomio allora soddisferà la condizione perché un polinomio è C^∞ (le derivate successive sono nulle).

³⁸⁶Slide 12 PDF 22, Teorema 4.11 PG 95.

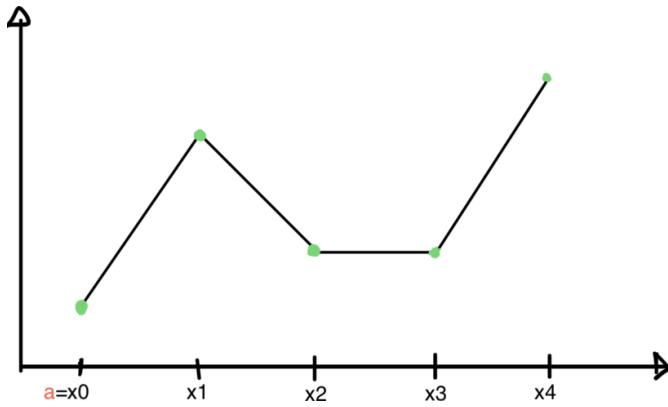


Figura 31: Esempio di spline con $n = 4$.

La dimensione di uno spazio vettoriale è importante per individuare una spline univocamente. Un risultato importante dell’Osservazione 4.24 è il seguente: è possibile individuare in modo univoco una spline di grado m sulla partizione Δ , con $m + n$ condizioni distinte, quindi indipendenti. Inoltre, la dimensione di uno spazio vettoriale è utilizzata per determinare quante condizioni sono necessarie per individuare un oggetto.

Osservazione 4.25. ³⁸⁷ $\Pi_m \subset S_m(\Delta)$.

Dalla precedente osservazione è possibile dedurre che lo spazio vettoriale dei polinomi di grado al più m è contenuto nell’insieme delle spline di grado m . Inoltre, è necessario ricordare che il polinomio di grado m è una funzione C^∞ e, rispetto a ciascun sottointervallo, coincide con se stesso.

Definizione 4.14. ³⁸⁸ Una spline $s_m(x)$ sulla partizione Δ intercala una funzione $f(x)$ (nei nodi della partizione) se (valgono le seguenti condizioni di interpolazione):

$$s_m(x_i) = f(x_i) \equiv f_i, \quad i = 0, \dots, n. \quad (4.47)$$

La definizione aggiunge alle due condizioni necessarie affinché $s_m(x)$ sia definita spline, le condizioni necessarie affinché $s_m(x)$ intercali $f(x)$.

Osservazione 4.26. ³⁸⁹ Le sole condizioni di interpolazione $(n + 1)$ permettono di individuare univocamente le spline di grado 1, anche dette spline lineari. In questo caso una spline lineare è la spezzata che congiunge i punti di interpolazione (x_i, f_i) , per $i = 0, \dots, n$. La spline lineare interpolante è data da

$$s_1|_{[x_{i-1}, x_i]}(x) = \frac{(x - x_{i-1})f_i + (x_i - x)f_{i-1}}{x_i - x_{i-1}}, \quad i = 1, \dots, n. \quad (4.48)$$

Quindi $s_1(x)$ è il segmento che congiunge i punti in coordinate (x_{i-1}, f_{i-1}) e (x_i, f_i) . Questo sarà utile quando saranno calcolate spline diverse dalle lineari.

³⁸⁷ Slide 12 PDF 22.

³⁸⁸ Definizione 2 Slide 12 PDF 22, Definizione 4.4 PG 95. Ciò che è tra parentesi è un aggiunta maggiore chiarezza. Date le condizioni 1. e 2. della Definizione 4.13 è possibile dare la seguente definizione.

³⁸⁹ Slide 13 PDF 22, PG 95.

La definizione di spline di grado m (Definizione 4.13) prescinde dal fatto che la spline sia interpolante la funzione. All'interno dello spazio vettoriale l'obbiettivo è quello di individuare una spline interpolante, possibilmente unica. Quando il grado della spline aumenta allora esistono diverse spline di grado superiore, per le quali ciascuna sarà una spline di grado m interpolante. Queste spline differendo per poco, dato che le condizioni mancanti saranno imposte e ciascuna scelta riguardo le condizioni poste.

La condizione 1. della Definizione 4.13 implica che la spline s_m debba essere una funzione di classe $C^{(m-1)}[a, b]$. Questo requisito richiede di impostare delle condizioni sui punti interni della partizione x_1, \dots, x_{n-1} , ovvero i punti di contiguità tra i sottointervalli $[x_{i-1}, x_i]$ e $[x_i, x_{i+1}]$, per $i = 1, \dots, n-1$. Ovvero, le condizioni da impostare sono le seguenti:

$$s_m^{(j)}|_{[x_{i-1}, x_i]}(x_i) = s_m^{(j)}|_{[x_i, x_{i+1}]}(x_i), \quad j = 0, \dots, m-1, \quad i = 1, \dots, n-1. \quad (4.49)$$

³⁹⁰In altri termini, i due polinomi devono raccordarsi fino alla derivata $m-1$ nel punto x_i . Quando $j=0$ è richiesta solo la continuità, se è ricercata una spline interpolante in quei punti "viene gratis". Entrambi i polinomi devono assumere il valore della funzione che sta essendo interpolata. Il problema sono le derivate successive.

Vale il seguente risultato.

Teorema 4.8. ³⁹¹Se s_m è una spline di grado $m \geq 2$ sulla partizione Δ , allora $s'_m(x)$ è una spline di grado $m-1$ sulla stessa partizione.

Dimostrazione. Se $s_m(x)$ è una spline di grado $m \geq 2$ sulla partizione Δ , allora:

1. $s_m(x) \in C^{(m-1)}[a, b] \stackrel{392}{\Rightarrow} s'_m(x) \in C^{(m-2)}[a, b];$
2. $s_m|_{[x_{i-1}, x_i]}(x) \in \Pi_m, \forall i = 1, \dots, n \Rightarrow s'_m|_{[x_{i-1}, x_i]}(x) \in \Pi_{m-1}, \forall i = 1, \dots, n.$

Pertanto, $s'_m \in S_{m-1}(\Delta)$. □

Dato che $m \geq 2 (> 0)$, per il Teorema, quindi la funzione è continua (questo vale anche per le funzioni lineari). (C^{-1} significa che la funzione è costante.)

4.8 Spline cubiche

Osservazione 4.27. ³⁹³Nella pratica computazionale assumono particolare importanza le **spline cubiche ($m = 3$)**. Se è ricercata la spline cubica interpolante sono necessarie due ulteriori condizioni, ognuna delle quali darà origine ad una spline cubica interpolante generalmente diversa, oltre alle condizioni definite in (4.47). Una funzione di grado 3 sarà interpolata da un polinomio di grado 3 e può accadere che più spline coincidano.

I motivi per i quali è scelta la spline cubica sono le condizioni solitamente asimmetriche rispetto agli estremi dell'intervallo. La derivata prima di una spline quadratica è continua, ha un profilo smooth e per questo sono scelte per l'approssimazione di funzioni. Sono utilizzate le spline cubiche e non quadratiche dato che l'unica condizione è la non simmetria.

³⁹⁰Significa che sono presenti due sottointervalli contigui: a sinistra del primo sottointervallo la spline coincide con un polinomio, nel secondo sottointervallo con un altro polinomio di grado n . La j rappresenta la derivata j -esima di s_m e x_i il punto continguo degli insiemi.

³⁹¹Slide 14 PDF 22, Teorema 4.10 PG 95

³⁹²Data una funzione di una qualsiasi classe m allora la sua derivata è di $m-1$. È eliminata un'equazione ma le successive derivate rimangono.

³⁹³Slide 15 PDF 22.

Per individuare univocamente una spline cubica definita sulla partizione Δ , occorrono $n + 3$ ($m = 3$) condizioni indipendenti tra loro. Se è ricercata una spline interpolante una data funzione $f(x)$, le condizioni di interpolazione (4.47) forniscono $n + 1$ condizioni. Pertanto, rimangono da fissare **2 condizioni addizionali** per determinare univocamente una spline cubica, dove ciascuna coppia di condizioni addizionali darà origine ad una spline interpolante, le quali sono **generalmente diverse tra loro**. Le condizioni di interpolazione sono importanti ed esiste un certo numero di modi per imporle, ciascuno di questi modi genererà una spline interpolante, generalmente le spline generate sono diverse.

Sono esaminate **4 possibili implementazioni** delle condizioni aggiuntive e necessarie da imporre per determinare una spline cubica interpolante.

4.8.1 Spline cubica naturale

³⁹⁴ Questa spline è determinata dalle due seguenti condizioni:

$$s_3''(a) = 0, \quad s_3''(b) = 0. \quad (4.50)$$

La spline cubica naturale è quella che minimizza la curvatura totale della curva (non sarà trattato). Questa spline è scelta perché porta all'algoritmo di misurazione più efficiente tra tutti (non sarà trattato).

4.8.2 Spline cubica completa

³⁹⁵ Questa spline è determinata dalle due seguenti condizioni:

$$s_3'(a) = f'(a), \quad s_3'(b) = f'(b) \quad (4.51)$$

Negli estremi della partizione Δ è imposta una condizione di interpolazione di Hermite. Dal punto di vista computazionale è implementata con variazioni della spline cubica naturale.

4.8.3 Spline cubica interpolante periodica

³⁹⁶ Questo tipo di spline è utilizzata supponendo che la funzione $f(x)$ sia una funzione periodica sull'intervallo $[a, b]$. Infatti vale la seguente osservazione:

Osservazione 4.28. Quanto scritto significa che $b - a$, che è l'ampiezza dell'intervallo, deve essere un multiplo intero del periodo T della funzione ^[397].

Esempio 4.7. Se $f(x) = \sin(x) \Rightarrow b$ deve essere un multiplo di 2π , altrimenti la funzione non è periodica (ad esempio con $[0, \pi]$).

Dalle condizioni di interpolazioni e per la periodicità di $f(x)$, è noto quanto segue:

$$s_3(a) = f(a) = f(b) = s_3(b).$$

Se $f(x) \in C^{(2)}[a, b]$, ovvero f è più che continua, **come funzione periodica**, allora:

$$f'(a) = f'(b), \quad f''(a) = f''(b). \quad (4.52)$$

³⁹⁴Slide 4 PDF 23, PG 96.

³⁹⁵Nota precedente.

³⁹⁶Nota precedente. Questa non è applicata ad una funzione interpolanda ma a funzioni periodiche.

³⁹⁷È possibile un'efficienza maggiore esprimendo b come $b = T \cdot a$.

Dato che è richiesto che la spline cubica $s_3 \in C^{(2)}[a, b]$, per la definizione di spline (Definizione 4.13), è imposto che lo sia anche come funzione periodica. Analogamente a (4.52), sono imposte le seguenti condizioni aggiuntive:

$$s'_3(a) = s'_3(b), \quad s''_3(a) = s''_3(b). \quad (4.53)$$

Osservazione 4.29. Questo tipo di spline è applicato esclusivamente al caso di funzioni periodiche sull'intervallo $[a, b]$.

Intermezzo: È possibile costruire spline cubiche interpolanti una funzione periodica del tipo naturale, periodica e completa ottenendo risultati generalmente diversi fra loro.

4.8.4 Spline cubica interpolante not-a-knot

Osservazione 4.30.³⁹⁸ Questa spline è implementata nella *function* *spline* di Matlab.

Questa spline non necessita di ulteriori informazioni oltre a quelle di interpolazione, le condizioni aggiuntive sono imposte in modo implicito.

È noto che nei primi due sottointervalli siano presenti due polinomi di grado 3 (e così anche negli ultimi due). Ciò che è richiesto è che i due polinomi ed i primi due sottoinsiemi, tramite l'unione ($[x_0, x_1] \cup [x_1, x_2] = [x_0, x_1]$), coincidano. In questo modo è imposta una condizione in meno in modo esplicito.

La spline not-a-knot fa in modo che il nodo 1 (x_1) non sia il nodo che definisce una spline in senso classico. Questo è dovuto al fatto che è presente un unico polinomio nel primo e secondo sottoinsieme, in genere sono distinti. La stessa cosa avviene anche negli ultimi due sottointervalli.

Le condizioni aggiuntive sono imposte in modo implicito come segue:

1. lo stesso polinomio cubico definisce la spline s_3 sui primi 2 sottointervalli $[x_0, x_1]$ e $[x_1, x_2]$;
2. ³⁹⁹simmetricamente, lo stesso polinomio cubico definisce la spline s_3 sugli ultimi due sottointervalli $[x_{n-2}, x_{n-1}]$ e $[x_{n-1}, x_n]$.

Imporre questa condizione per la 1. significa quanto segue (similmente per 2.):

$$p_1(x) = s_3|_{[x_0, x_1]}(x), \quad p_2(x) = s_3|_{[x_1, x_2]}(x),$$

⁴⁰⁰

dove è ricercato $p_1(x) = p_2(x) \in \Pi_3$. In x_1 , tramite le condizioni di interpolazione, è ottenuto $p_1(x_1) = p_2(x_1)$. Il raccordo deve valere anche per le derivate prima e seconda, ovvero:

$$p'_1(x_1) = p'_2(x_1) \quad \wedge \quad p''_1(x_1) = p''_2(x_1).$$

³⁹⁸Slide 5 PDF 23, PG 97

³⁹⁹La condizione di simmetria è importante perché, data la funzione $f(x)$, è definita $g(x)$ come f rovesciando la direzione di x . $g(a) = f(b)$ se $f(x)$ e $g(x)$ hanno lo stesso verso. Le due funzioni hanno lo stesso grafico ed è importante che abbiano la stessa approssimazione. Sotto le stesse condizioni è ottenuto lo stesso risultato.

⁴⁰⁰Un polinomio di grado 3 ha 4 coefficienti, 4 gradi libertà e Π_3 è uno spazio vettoriale di dimensione 4.

⁴⁰¹ Se è imposto che $p_1'''(x_1) = p_2'''(x_1)$ allora $p_1(x) = s_3(x) = p_2(x)$, $x \in [x_0, x_2]$. Quindi, definita la spline $s_3(x)$ e le sue derivate come

$$\begin{aligned} s_3|_{[x_0, x_2]}(x) &\in \Pi_3, \\ s'_3|_{[x_0, x_2]}(x) &\in \Pi_2, \\ s''_3|_{[x_0, x_2]}(x) &\in \Pi_1, \\ s'''_3|_{[x_0, x_2]}(x) &\in \Pi_0, \end{aligned}$$

⁴⁰² ⁴⁰³ ⁴⁰⁴

è possibile imporre condizione sui primi due sottointervalli della partizione Δ , la quale è

$$\boxed{\frac{s''_3(\mathbf{x}_1) - s''_3(x_0)}{x_1 - x_0} = \frac{s''_3(x_2) - s''_3(\mathbf{x}_1)}{x_2 - x_1}.}$$

Simmetricamente, sugli ultimi due sottointervalli, sarà imposto:

$$\boxed{\frac{s''_3(x_n) - s''_3(\mathbf{x}_{n-1})}{x_n - x_{n-1}} = \frac{s''_3(\mathbf{x}_{n-1}) - s''_3(x_{n-2})}{x_2 - x_1}.}$$

\mathbf{x}_1 e \mathbf{x}_{n-1} precedenti non sono considerati nodi della partizione in quanto sono imposte le condizioni di interpolazione e le condizioni di default.

Ciò che è stato appena definito nei riquadri è una conveniente espressione, in virtù del Teorema 4.8, delle condizioni aggiuntive della spline cubica not-a-knot, ovvero:

$$s'''_3|_{[x_0, x_1]}(x_1) = s'''_3|_{[x_1, x_2]}(x_1), \quad s'''_3|_{[x_{n-2}, x_{n-1}]}(x_{n-1}) = s'''_3|_{[x_{n-1}, x_n]}(x_{n-1}). \quad (4.54)$$

La rappresentazione delle condizioni (4.54) tramite le derivate seconde è utile al fine di definire l'unicità della spline interpolante. Dal punto di vista computazionale è importante che le condizioni siano imposte sulla derivata seconda.

Osservazione 4.31. ⁴⁰⁵ L'ampiezza dell'intervalle i -esimo della partizione Δ , definita come (4.45), è definita come

$$h_i = x_i - x_{i-1}, \quad i = 1, \dots, n, \quad (4.55)$$

dove la massima ampiezza definita come

$$h = \max_{i=1, \dots, n} h_i,$$

allora è possibile dimostrare, se $f \in C^{(4)}[a, b]$, che per tutte le spline cubiche esaminate vale

$$\left\| f^{(i)} - s_3^{(i)} \right\| = O(h^{4-i}), \quad i = 0, 1, 2.$$

⁴⁰¹ Restrizione sul primo sottointervallo.

⁴⁰² È presente un raccordo $C^{(2)}$ nel punto di continuità ed è noto il raccordo della derivata terza. Inoltre, la funzione ha derivate che coincidono in più punti, quando due polinomi sono lo stesso, per $x \in [x_0, x_1] \cup [x_1, x_2] = [x_0, x_2]$. Le precedenti sono 4 condizioni distinte con 4 gradi di libertà.

⁴⁰³ Restrizione sul secondo sottointervallo.

⁴⁰⁴ Questo polinomio di grado 0 è una costante, il coefficiente lineare della retta derivata seconda, ed è la stessa a sinistra ed a destra di x_1 . Questo significa che il rapporto incrementale della derivata seconda su x_0, x_1 e x_2 deve essere lo stesso.

⁴⁰⁵ Slide 2 PDF 24, Osservazione 4.5 PG 97.

Quanto osservato significa che le spline cubiche consentono di approssimare efficientemente funzioni regolari, senza preoccuparsi troppo della scelta della partizione (ad esempio quella uniforme è sufficiente).

Approssimare la funzione significa approssimare uniformemente anche la derivata prima e seconda. Questo risultato è esatto per tutte le spline cubiche tranne per la naturale, dove, se forzata, la derivata seconda negli estremi è 0. Questo non è sempre vero in quanto è possibile introdurre errori sistematici. È possibile affermare che l'errore diminuisce rapidamente quando è ottenuto dagli estremi dell'intervallo.

4.9 Calcolo (pratico) di una spline cubica

⁴⁰⁶ Per trattare questa Sezione è utile la Sezione 4.11.

Il problema da affrontare è quello del calcolo della spline cubica $s_3(x)$ interpolante $f(x)$ sulla partizione, definita come in (4.45), $\Delta = \{a = x_0 < x_1 < \dots < x_n = b\}$.

Per determinare un algoritmo efficiente per il calcolo di una spline cubica, oltre a (4.55), sarà usata la notazione seguente:

$$m_i = s_3''(x_i), \quad i = 0, \dots, n. \quad (4.56)$$

In particolare saranno esaminati gli algoritmi per il calcolo delle spline cubiche naturali e not-a-knot. Sarà altresì trattato come, utilizzando questi argomenti, con piccoli cambiamenti, possono essere ottenute le spline naturali e complete. È necessario individuare quali siano le condizioni, in termini di m_i definiti tramite (4.56), che caratterizzano l'un l'altra.

Il calcolo della spline cubica naturale è semplice ed, inoltre, è stato trattato come le condizioni imposte riguardino l'annullamento della derivata seconda, negli estremi a e b . L'annullamento della derivata seconda e la notazione (4.56) implicano che, per una **spline cubica naturale**, le condizioni diventino

$$m_0 = 0 = m_n. \quad (4.57)$$

Imponendo (4.55), le **condizioni aggiuntive della spline not-a-knot**, ovvero degli estremi sinistro e destro (4.54), diventano

$$\frac{m_1 - m_0}{h_1} = \frac{m_2 - m_1}{h_2}, \quad \frac{m_{n-1} - m_{n-2}}{h_{n-1}} = \frac{m_n - m_{n-1}}{h_n}$$

ovvero $h_2(m_1 - m_0) = h_1(m_2 - m_1)$, $h_n(m_{n-1} - m_{n-2}) = h_{n-1}(m_n - m_{n-1})$, dalle quali sono ottenute le condizioni

$$h_2 m_0 - (h_1 + h_2) m_1 + h_1 m_2 = 0 \quad h_{n-1} m_n - (h_{n-1} + h_n) m_{n-1} + h_n m_{n-2} = 0. \quad (4.58)$$

(4.58) sono le **condizioni aggiuntive per le spline not-a-knot**.

Riguardo le altre condizioni $\{m_i\}$, è possibile osservare che, se $s_3(x)$ è una spline cubica sulla partizione Δ , allora $s'_3(x)$ è una spline di grado 2 sulla stessa partizione, e $s''_3(x)$ è una spline lineare su Δ . Pertanto, essendo $s''(x)$ una spline lineare, da (4.48) e (4.56), è ottenuto che

$$s''_3(x) \underset{407}{=} \begin{cases} \frac{(x - x_{i-1})m_i + (x_i - x)m_{i-1}}{h_i}, & x \in [x_{i-1}, x_i], \\ \parallel & \\ x_i - x_{i-1} & \end{cases} i = 1, \dots, n.$$

⁴⁰⁶ Slide 2-13 PDF 24, PG 97-101.

⁴⁰⁷ Le m_i non sono note. Sono noti i valori della derivata seconda della spline che assume nelle assise di interpolazione. Quindi la derivata seconda della spline è la spline lineare che interpola gli m_i .

Integrando membro a membro la precedente equazione è ottenuto

$$s'_3(x) = \frac{(x - x_{i-1})^2 m_i - (x_i - x)^2 m_{i-1}}{2h_i} + q_i, \quad x \in [x_{i-1}, x_i], \quad i = 1, \dots, n. \quad (4.59)$$

⁴⁰⁹ Integrando nuovamente è ottenuta un'ulteriore spline:

$$s_3(x) = \frac{(x - x_{i-1})^3 m_i + (x_i - x)^3 m_{i-1}}{6h_i} + q_i(x - x_{i-1}) + r_i, \quad x \in [x_{i-1}, x_i], \quad i = 1, \dots, n. \quad (4.60)$$

⁴¹¹ Imponendo le condizioni di interpolazione a ciascun intervallo, è ottenuto quanto segue:

- $s_3(x_1) = \frac{h_i^2}{6} m_{i-1} + r_i = f(x_{i-1})$, dalla quale è ottenuta

$$r_i = f(x_{i-1}) - \frac{h_i^2}{6} m_{i-1}, \quad (4.61)$$

dove $f(x_{i-1})$ ed h_i^2 sono valori noti.

- $s_3(x_i) = \frac{h_i^2}{6} m_i + q_i h_i + r_i = f(x_i)$. Pertanto,

$$\begin{aligned} h_i q_i &= f(x_i) - r_i - \frac{h_i^2}{6} m_i && \stackrel{413}{=} f(x_i) - f(x_{i-1}) + \frac{h_i^2}{6} (m_{i-1} - m_i) \Rightarrow \\ &\Rightarrow q_i && = \underbrace{\frac{f(x_i) - f(x_{i-1})}{h_i}}_{h_i} + \frac{h_i}{6} (m_{i-1} - m_i). \end{aligned}$$

Da cui,

$$q_i = f[x_{i-1}, x_i] + \frac{h_i}{6} (m_{i-1} - m_i). \quad (4.62)$$

Ora è necessario calcolare gli $\{m_i\}$, per poter calcolare le costanti q_i e r_i di ciascun sottointervallo $[x_{i-1}, x_i]$ e quindi (4.60).

Imponendo che $s'_3(x)$ sia continua nei punti x_i , $i = 1, \dots, n$, ovvero imponendo che

$$s'_3|_{[x_{i-1}, x_i]}(x_i) = s'_3|_{[x_i, x_{i+1}]}(x_i),$$

è ottenuto, da (4.59):

$$\frac{h_i}{2} m_i + q_i = -\frac{h_{i+1}}{2} m_i + q_{i+1}, \quad i = 1, \dots, n-1.$$

⁴⁰⁸Costante d'integrazione.

⁴⁰⁹Non conoscendo le m_i non è possibile calcolare la derivata seconda. Integrando membro a membro sarà ottenuta la derivata prima della spline e questa sarà la restrizione all' i -esimo sottointervallo $[x_{i-1}, x_i]$.

⁴¹⁰Nuova costante di integrazione.

⁴¹¹Fossero noti gli m_i e fosse possibile determinare le costanti di integrazione r_i e q_i , allora sarebbe possibile calcolare il polinomio di grado 3, il quale costituisce la restrizione della spline nell'intervallo d'interesse. Ciò che sarà richiesto nell'elaborato, quando sarà necessario calcolare la spline nei punti assegnati, sarà calcolare la spline nei punti interpolanti. Adesso saranno eliminate le costanti q_i e r_i imponendo condizione di interpolazione agli estremi di ciascun sottointervallo.

⁴¹²Non calcolabile perché le m_i non sono note.

⁴¹³Con questa uguaglianza sono ricavate le g_i perché le r_i sono state calcolate.

⁴¹⁴Sostituzione di r_i con (4.61).

Quindi [415]:

$$3h_i m_i + h_i(m_{i-1} - m_i) + 3h_{i+1}m_i - h_{i+1}(m_i - m_{i+1}) = 6(f[x_i, x_{i+1}] - f[x_{i-1}, x_i]), \quad i = 1, \dots, n-1.$$

Raggruppando gli m_i con gli stessi indici a primo membro è ottenuto quanto segue:

$$h_i m_{i-1} + 2(h_i + h_{i+1})m_i + h_{i+1}m_{i+1} = 6(f[x_i, x_{i+1}] - f[x_{i-1}, x_i]), \quad i = 1, \dots, n-1.$$

Dividendo membro a membro per $h_i + h_{i+1} = x_{i+1} - x_{i-1}$, è ottenuto:

$$\begin{array}{c} \varphi_i \\ \parallel \\ \boxed{\frac{h_i}{h_i + h_{i+1}}} m_{i-1} + 2m_i + \boxed{\frac{h_{i+1}}{h_i + h_{i+1}}} m_{i+1} \end{array} = 6 \frac{f[x_i, x_{i+1}] - f[x_{i-1}, x_i]}{x_{i+1} - x_{i-1}} \stackrel{416}{=} 6 f[x_{i-1}, x_i, x_{i+1}], \quad i = 1, \dots, n-1. \quad (4.63)$$

Osservazione 4.32. ⁴¹⁷ $\varphi_i, \xi_i > 0, \varphi_i + \xi_i = 1$.

Le equazioni (4.63) sono un sistema di $n-1$ equazioni nelle $n+1$ incognite $\{m_0, \dots, m_n\}$.

Nel caso di spline naturali, tenendo di conto delle condizioni (4.57), è **ottenuto il sistema tridiagonale**

$$\begin{pmatrix} 2 & \xi_1 & & & \\ \varphi_2 & 2 & \xi_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \xi_{n-2} \\ & & & \varphi_{n-1} & 2 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ \vdots \\ m_{n-1} \end{pmatrix} = 6 \begin{pmatrix} f[x_0, x_1, x_2] \\ f[x_1, x_2, x_3] \\ \vdots \\ \vdots \\ f[x_{n-2}, x_{n-1}, x_n] \end{pmatrix}.$$

Dall'Osservazione 4.32 è possibile notare che la matrice dei coefficienti è diagonale dominante sia per righe che per colonne e quindi la sua fattorizzazione LU è definita. **Calcolati i valori $\{m_i\}$ incogniti, la spline naturale è calcolata sostituendoli, assieme a (4.61)-(4.62), in (4.60).**

Ragionamento analogo vale per la spline cubica not-a-knot ottenuta imponendo le relative condizioni aggiuntive, per la quale le condizioni (4.58) e (4.63) danno origine al sistema lineare

$$\begin{pmatrix} \xi_1 & -1 & \varphi_1 & & & \\ \varphi_1 & 2 & \xi_1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \varphi_{n-1} & 2 & \xi_{n-1} & \\ & & \xi_{n-1} & -1 & \varphi_{n-1} & \end{pmatrix} \begin{pmatrix} m_0 \\ m_1 \\ \vdots \\ \vdots \\ m_n \end{pmatrix} = 6 \begin{pmatrix} 0 \\ f[x_0, x_1, x_2] \\ \vdots \\ f[x_{n-2}, x_{n-1}, x_n] \\ 0 \end{pmatrix}. \quad (4.64)$$

Sostituendo alla prima equazione la somma delle prime due, all'ultima la somma delle ultime due e moltiplicando a destra la matrice dei coefficienti per

$$\begin{pmatrix} 1 & -1 & -1 & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & 1 & 1 \end{pmatrix} \equiv I_{n+1},$$

⁴¹⁵Moltiplicando per 6 e portando a sinistra dell'uguale tutto ciò che dipende da m , è ottenuto quanto segue.

⁴¹⁶Per (4.19).

⁴¹⁷Slide 9 PDF 24.

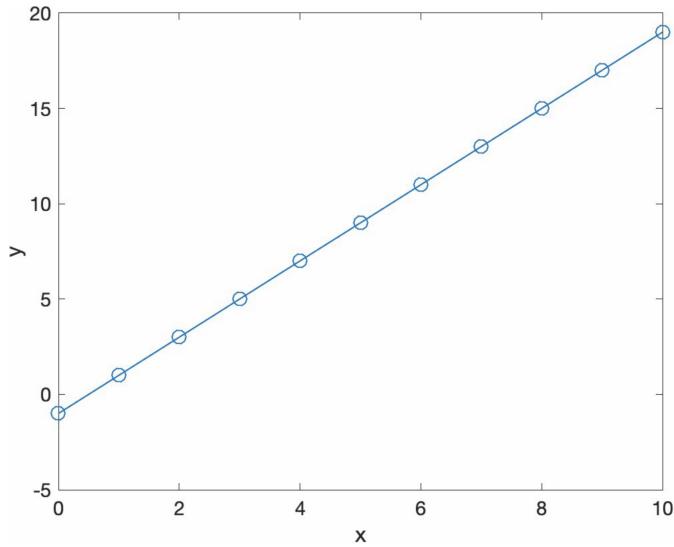


Figura 32: Esempio di 10 dati esatti.

il sistema lineare (4.64) è equivalente a

$$\begin{pmatrix} 1 & 0 & & & \\ \varphi_1 & 2 - \varphi_1 & \xi_1 - \varphi_1 & & \\ & \varphi_2 & 2 & \xi_2 & \\ & & \ddots & \ddots & \ddots \\ & & & \varphi_{n-2} & 2 & \xi_{n-2} \\ & & & & \varphi_{n-1} - \xi_{n-1} & 2 - \xi_{n-1} & \xi_{n-1} \\ & & & & & 0 & 1 \end{pmatrix} \begin{pmatrix} m_0 + m_1 + m_2 \\ m_1 \\ \vdots \\ m_{n-1} \\ m_n + m_{n-1} + m_{n-2} \end{pmatrix} = 6 \begin{pmatrix} f[x_0, x_1, x_2] \\ f[x_0, x_1, x_2] \\ \vdots \\ f[x_{n-2}, x_{n-1}, x_n] \\ f[x_{n-2}, x_{n-1}, x_n] \end{pmatrix}, \quad (4.65)$$

che risulta essere ancora tridiagonale e, avendo tutti i minori principali non nulli, fattorizzabile LU .

4.10 Approssimazione polinomiale nel senso dei minimi quadrati

⁴¹⁸ L'argomento trattato in questa Sezione è un diverso tipo di approssimazione. Spesso sono presenti troppi dati da approssimare affetti da errore, tipicamente senza bias, con una distribuzione gaussiana. Il concetto è che se il dato è affetto da errore interpolare non ha significato. È necessario trovare un'altra approssimazione ed a titolo di esempio sono presenti le Figure 32-35.

Inoltre, è bene sottolineare che **è stato utilizzato** il termine **"Approssimazione"** e non **"Interpolazione"**.

⁴¹⁸Slide 6-9 PDF 25, PG 101-104.

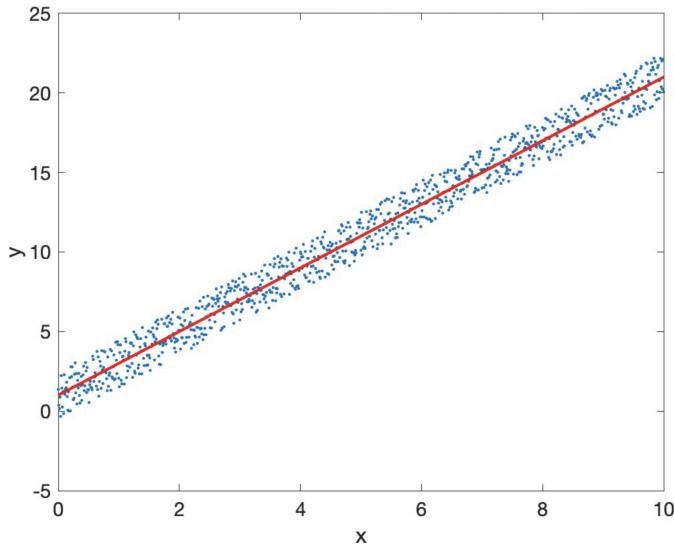


Figura 33: Esempio di 1000 dati affetti da errore.

Per la Figura 32 è possibile affermare che la retta che passa per i 10 punti necessiterebbe solo di due punti per definirla e che i dati sono collineari. Se fossero interpolanti con una spline o con un polinomio interpolante sarebbe ottenuta una retta.

Supposto di avere 1000 dati effetti da errore non sistematico, è ottenuta Figura 33. I dati, i puntini nella figura, sono ottenuti partendo dalla retta ed aggiungendo un termine d'errore, non lo stesso della retta, da un distribuzione gaussiana. Se fosse necessario interpolare allora il grafico oscillerebbe tra gli estremi dei puntini blu (più o meno periodicamente), non fornendo informazioni utili. La retta rossa, rappresentante la retta in Figura 32, approssima bene i dati.

Per la Figura 35 non ha senso calcolare il polinomio interpolante perché un polinomio di grado 10000 è impossibile da calcolare ed un polinomio si fatto non ha dati utili. Inoltre, la retta rossa della figura rappresenta la parabola di Figura 34, la quale approssima in modo soddisfacente i dati.

Dalle Figure 32-35 è possibile affermare che è un caso che un punto utile sia trovato (perché sono troppi), infatti in genere non è trovato.

Supposto di avere un fenomeno da rilevare, dipendente da un variabile indipendente x , che soddisfa una legge di tipo polinomiale (non inventata):

$$y = p(x), \quad p(x) \in \Pi_m.$$

Osservazione 4.33. In genere il grado m è noto a priori.

Tuttavia, y , data l'uguaglianza sopra, è un polinomio del tipo

$$p(x) = \sum_{k=0}^m a_k x^k, \quad (4.66)$$

per il quale non sono noti i coefficienti del polinomio ma sono note le seguenti misurazioni del fenomeno:

$$(x_i, y_i), \quad i = 1, \dots, n, \quad n \geq m, \quad n \gg k + 1. \quad (4.67)$$

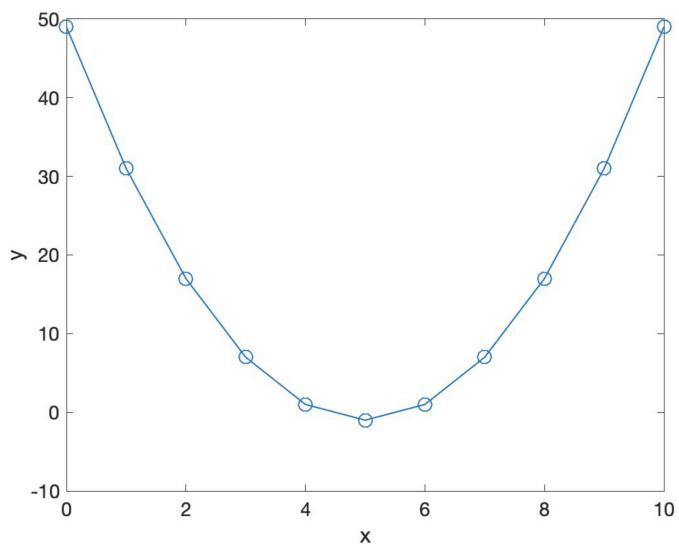


Figura 34: Esempio di spline con $n = 4$.

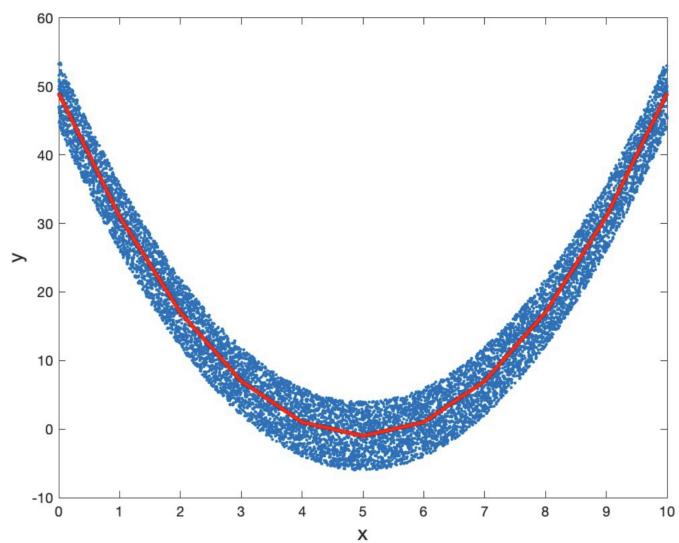


Figura 35: Esempio di spline con 10000 dati.

Il problema dell'approssimazione polinomiale nel senso dei minimi quadrati è determinare il polinomio (4.66), ovvero determinare i coefficienti a_0, \dots, a_n , che meglio approssima le coppie di dati definite come (4.67). Inoltre, è possibile osservare che il problema da risolvere è un sistema sovradianimensionato ($n \geq m$).

Osservazione 4.34. È assunto che almeno $k + 1$ delle ascisse $\{x_i\}$ siano distinte tra loro (dato k il grado del polinomio da determinare)⁴²⁰.

Per determinare i coefficienti incogniti del polinomio è necessario definire la migliore approssimazione dei dati. Il problema è rappresentato in forma vettoriale, quindi, dati i vettori

$$\underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \underline{z} = \begin{bmatrix} p(x_1) \\ \vdots \\ p(x_n) \end{bmatrix} \equiv \begin{bmatrix} \sum_{k=0}^m a_k x_1^k \\ \vdots \\ \sum_{k=0}^m a_k x_n^k \end{bmatrix}, \quad (4.68)$$

contenenti rispettivamente i valori attesi ed i valori misurati in corrispondenza delle ascisse x_0, \dots, x_n , è determinato il vettore $\underline{a} = (a_0, \dots, a_m)^T$, il quale minimizza la norma $\|\underline{r}\|_2^2 = \sum_{k=0}^n |y_i - z_i|^2$ (ovvero la norma euclidea al quadrato del vettore residuo $\underline{r} = \underline{z} - \underline{y}$). Tuttavia, è possibile esprimere \underline{z} come

$$\underline{z} \stackrel{421}{=} \begin{bmatrix} x_1^0 & \dots & x_1^m \\ \vdots & & \vdots \\ x_n^0 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_m \end{bmatrix} \equiv V \underline{a},$$

con $V \in \mathbb{R}^{n \times k+1}$ matrice del tipo di Vandermonde. Quindi, ricercando il vettore \underline{r} di minima norma euclidea tale che $V \underline{a} = \underline{y} + \underline{r}$, il problema dell'approssimazione polinomiale si traduce nella ricerca della soluzione del sistema lineare sovradianimensionato

$$V \underline{a} = \underline{y}, \quad (4.69)$$

nel senso dei minimi quadrati.

⁴²³

Osservazione 4.35. ⁴²² $V \in \mathbb{R}^{n \times k+1}$, con $n > k + 1$. Questo significa che sono presenti più equazioni (n) che incognite ($k + 1$).

$V \underline{a} = \underline{y}$ è risolto tramite fattorizzazione QR, sotto la condizione che la V abbia rango massimo ($m + 1$) affinché esista soluzione e questa sia unica. Quindi, almeno $m + 1$ delle ascisse $\{x_i\}$ devono essere distinte fra loro.

Osservazione 4.36. ⁴²⁴ La matrice V ha rango massimo ($k + 1$) sotto le ipotesi dell'Osservazione 4.34.

Data V , del tipo di Vandermonde e le $k + 1$ righe corrispondenti alle $k + 1$ ascisse distinte, è ottenuta una matrice quadrata, la quale è una matrice di Vandermonde definita da ascisse distinte (quindi è nonsingolare). Se esiste una sottomatrice di dimensione $k + 1$, ma singolare, questa ha rango massimo.

⁴¹⁹ n è tipicamente molto maggiore di $k + 1$, quindi il numero di misurazioni deve essere almeno pari al numero di coefficienti. Se $n = k + 1$, ovvero n è pari al numero di coefficienti, allora è calcolata l'interpolazione.

⁴²⁰ Le $\{x_i\}$ possono essere utilizzate per effettuare più misurazioni in un punto in cui è necessario. È necessario che la condizione ($k + 1$ ascisse distinte) sia soddisfatta per ammettere che la formulazione abbia soluzione unica.

⁴²¹ La moltiplicazione riga di V per \underline{a} da la forma base di Lagrange.

⁴²² OSS 1 Slide 8 PDF 25.

⁴²³ Sul libro, pagina 103, ha notazione diversa perché n è indicizzato da 0, quindi è $n + 1$.

⁴²⁴ OSS 2 Slide 8 PDF 25, Teorema 4.12 PG 103.

Fuori dalla lezione: il vettore \underline{a} in (4.69) esiste ed è unico. Di conseguenza esiste ed è unico il polinomio di approssimazione ai minimi quadrati (4.66). Riguardo al costo computazionale per ottenere il polinomo di approssimazione ai minimi quadrati, è possibile affermare che valgono le stesse considerazioni riguardo alla fattorizzazione QR fatte nelle Sezioni 3.8 e 3.8.2.

Osservazione 4.37. ⁴²⁵ Pertanto, il problema è risolvibile utilizzando la fattorizzazione $V = QR$, con $Q \in \mathbb{R}^{n \times n}$ ortogonale e $R = \begin{pmatrix} \widehat{R} \\ 0 \end{pmatrix} \in \mathbb{R}^{n \times k+1}$, dove $\widehat{R} \in \mathbb{R}^{(k+1) \times (k+1)}$, triangolare superiore e non singolare.

Osservazione 4.38. ⁴²⁶ Se $n = k + 1$ è ottenuto il polinomio di grado k che interpola (x_i, y_i) , $i = 0, \dots, k$.

Per $n = k + 1$, V è quadrata ed ha rango massimo, quindi il sistema $V\underline{a} = \underline{y}$ è risolvibile.

Osservazione 4.39. ⁴²⁷ Questa tecnica di risoluzione è implementata nella funzione **polyfit** di Matlab.

Osservazione 4.40. ⁴²⁸ Dato un sistema lineare quadrato

$$V\underline{a} = \underline{y} \quad (4.70)$$

e D una matrice nonsingolare, allora (4.70) ha la stessa soluzione di

$$DV\underline{a} = Dy. \quad (4.71)$$

⁴²⁹

⁴³⁰

Questo **non è più vero** se la matrice D è rettangolare (perché (4.70) e (4.71) non avrebbero più la solita soluzione).

(4.71) è utilizzato, nella pratica, calcolando $D = diag(\underline{p})$, con \underline{p} vettore contenente una **distribuzione di probabilità discreta** ($\iff \underline{p} \geq 0 \wedge \text{sum}(\underline{p}) = 1$), per dare un diverso peso, p_i , a ciascuna equazione.

4.11 Risoluzione di un sistema tridiagonale

⁴³¹ È necessario, ai fini della Sezione 4.9, risolvere il sistema lineare

$$A\underline{x} = \underline{q} \in \mathbb{R}^n,$$

dove

$$A = \begin{bmatrix} a_1 & c_1 & & & & & \\ b_2 & a_2 & c_2 & & & & \\ & b_3 & a_3 & c_3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & c_{n-1} & \\ & & & & b_n & a_n & \end{bmatrix} \in \mathbb{R}^{n \times n}$$

⁴²⁵OSS 3 Slide 9 PDF 25

⁴²⁶OSS 4 Slide 8 PDF 25

⁴²⁷OSS 5 Slide 9 PDF 25, Osservazione 4.8 PG 104.

⁴²⁸OSS6 Slide 9 PDF 25.

⁴²⁹È utilizzato "Questo" perché alcune volte, nella ricerca del polinomio che approssima i dati e che soddisfa nel senso dei minimi quadrati il sistema, alcuni dati sono più attendibili di altri. I dati sono pesati in base all'attendibilità, attraverso una distribuzione discreta sulle diagonali di \underline{p} , dove le righe corrispondenti a distribuzioni più grandi saranno più importanti di altre.

⁴³⁰Ricavare la soluzione nel senso dei minimi quadrati del sistema lineare è diverso dal ricavare la soluzione nel senso dei minimi quadrati del sistema.

⁴³¹Slide 8-10 PDF 23

è una matrice triangolare. È supposto che questa sia fattorizzabile LU .

Osservazione 4.41. ⁴³² È possibile **memorizzare A con 3 vettori (complessità lineare dei dati)**.

Risolvere per fattori LU ha complessità lineare. Invece di avere $\frac{2}{3}n^3$ flops per la fattorizzazione ed n^2 operazioni per i fattori triangolari. In questo caso la complessità è n per la fattorizzazione ed n per la risoluzione.

Errori da non fare: con complessità lineare è possibile risolvere problemi di qualsivoglia grandezza, con n^2 fino ad un certo punto e con ordine superiore al secondo no. Per risolvere il sistema tridiagonale alcune persone utilizzano una matrice piena, assegnano i tre vettori alle tre diagonali e le forniscono in input alla function di risoluzione, tramite fattorizzazione LU . In questo caso l'esercizio è valutato 0.

È necessario verificare A che sia fattorizzabile LU , ovvero ⁴³³ $A = LU$, con

$$L_{434} = \begin{bmatrix} 1 & & & \\ l_2 & 1 & & \\ & l_3 & 1 & \\ & & \ddots & \ddots \\ & & l_n & 1 \end{bmatrix}, U^{435} = \begin{bmatrix} d_1 & c_1 & & \\ & d_2 & c_2 & \\ & & \ddots & \ddots \\ & & & c_{n-1} \\ & & & d_n \end{bmatrix},$$

$$L \cdot U = \begin{bmatrix} d_1 & c_1 & & & \\ l_2 d_1 & (d_2 + l_2 c_1) & c_2 & & \\ & l_3 d_2 & (d_3 + l_3 c_2) & c_3 & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & c_{n-1} \\ & & & l_n d_{n-1} & (d_n + l_n c_{n-1}) \end{bmatrix}$$

uguagliando gli elementi omologhi, ottenendo:

$$\left. \begin{array}{l} d_1 = a_1 \\ l_i = b_i / d_{i-1}, \\ d_i = a_i - l_i \cdot c_{i-1}, \quad i = 2, \dots, n. \end{array} \right\} \text{al costo di } 3n \text{ flops dove è possibile sovrascrivere } l_i \rightarrow b_i, d_i \rightarrow a_i.$$

Il sistema bidiagonale per L può essere risolto tramite l'utilizzo dell'Algoritmo 4.9, con $\underline{x} \leftarrow \underline{q}$.

Algoritmo 4.9 Algoritmo risoluzione sistema bidiagonale per L

```
for i = 2 : n
    x_i = x_i - e_i x_(i - 1) % (2n flops)
end
```

Il sistema bidiagonale per U può essere risolto con l'Algoritmo 4.10.

Quindi al posto di $\frac{2}{3}n^3$ flops c'è $3n$ ed al posto di n^2 $2n$. Totale $8n$ flops, mentre se è simmetrica 7 flops.

Morale: se è richiesto di risolvere un sistema tridiagonale fattorizzabile LU allora è necessario utilizzare l'Algoritmo 4.10, non assegnando i 3 vettori ad una matrice e richiamando la function per la risoluzione fattorizzabile LU .

⁴³²Slide 9 PDF 23.

⁴³³ A è diagonale dominante quindi fattorizzabile LU .

⁴³⁴Matrice bidiagonale memorizzabile con un vettore.

⁴³⁵La diagonale dei d_i cambia rispetto ad A mentre la diagonale dei c_i è uguale ad A .

Algoritmo 4.10 Algoritmo risoluzione sistema bidiagonale per U

```
x_n = x_n / d_n
for i = n - 1 : -1 : 1
    x_i = (x_i - c_i x_{i+1}) / d_i %(3n flops)
end
```

5 Formule di quadratura numerica (approssimazione di integrali definiti)

⁴³⁶ Il problema da risolvere è quello di approssimare il valore dell'integrale nella forma

$$I(f) = \int_a^b f(x)dx, \quad -\infty < a < b < \infty, \quad (5.1)$$

dove $f : [a, b] \rightarrow \mathbb{R}$, almeno continua. In questa trattazione di approssimazione di integrali definiti è considerato solo il caso in cui f è continua, per evitare singolarità integrabili nell'intervallo $[a, b]$ e non sarà considerato il caso di integrali definiti su insiemi illimitati. I metodi di approssimazione trattati saranno definiti mediante l'integrale di una approssimazione polinomiale a tratti di $f(x)$, in quanto l'integrale di un polinomio può essere calcolato facilmente ed in modo esatto.

Prima di trattare i metodi di approssimazione è studiato il condizionamento del problema, in cui la perturbazione è sulla funzione integranda $f(x)$. Studiare il condizionamento significa stabilire se al posto di f è presente una sua perturbazione, ovvero quando il risultato è perturbato.

Definizione 5.1 (Numero di condizionamento). Supposta $\tilde{f}(x) \in C^{(0)}[a, b]$, ovvero una perturbazione di $f(x)$, allora (vedi (4.32)):

$$\begin{aligned} \overbrace{|I(f) - I(\tilde{f})|}^{438} &= \left| \int_a^b f(x)dx - \int_a^b \tilde{f}(x)dx \right| = \left| \int_a^b (f(x) - \tilde{f}(x))dx \right| \\ &\stackrel{439}{\leq} \int_a^b |f(x) - \tilde{f}(x)|dx \leq \|f - \tilde{f}\| \cdot \int_a^b dx = (\mathbf{b} - \mathbf{a}) \cdot \|f - \tilde{f}\|, \end{aligned}$$

dove

$$\kappa = (\mathbf{b} - \mathbf{a}), \quad \|f - \tilde{f}\| = \max_{a \leq x \leq b} |f(x) - \tilde{f}(x)|$$

definiscono, rispettivamente, il **numero di condizionamento del problema** (5.1) e la misura l'**errore sui dati d'ingresso**.

5.1 Formule di Newton-Cotes

⁴⁴⁰ Le formule più semplici per l'approssimazione di $I(f)$ nell'intervallo $[a, b]$, dette **formule di Newton-Cotes**, sono ottenute calcolando l'integrale esatto del polinomio interpolante la funzione $f(x)$ su $n + 1$ ascisse equidistanti, ovvero:

$$\begin{aligned} x_i &= a + ih, & i &= 0, \dots, n, \\ p_n(x_i) &= f_i \equiv f(x_i), & h &= \frac{b-a}{n}. \end{aligned} \quad (5.2)$$

⁴³⁶Slide 10 PDF 25, 26-27, PG 105-114.

⁴³⁷Se $a > b$ allora è applicato Riemann, a e b sono invertiti di segno. Se non è conosciuta la derivata prima di f , l'integrale è calcolato in forma chiusa. Spesso è necessario fornire un metodo numerico in questo intervallo. Questi metodi si chiamano "Forma di quadratura" (i quali consentono di determinare l'integrale esatto di una approssimazione della funzione f , per la quale è possibile l'integrazione). Prima è necessario studiare il condizionamento del problema dell'integrazione.

⁴³⁸Errore del risultato.

⁴³⁹Il valore assoluto di un integrale di una funzione è minore uguale dell'integrazione del valore assoluto della funzione. Questo è specificato anche dal criterio di integrabilità.

⁴⁴⁰Slide 10 PDF 25-26, PG 106-108.

Sia $p(x)$ il polinomio interpolante $f(x)$, è considerata la sua forma di Lagrange (4.6)-(4.8) per l'approssimazione di $I(f)$. Segue la definizione della **formula di Newton-Cotes di grado n** (esprimibile in (5.3)):

$$I(f) \approx \int_a^b p(x)dx = \underbrace{\sum_{i=0}^n f_i \int_a^b L_{in}(x)dx}_{441} \equiv I_n(f).$$

Una formulazione più conveniente della precedente, poste le seguenti condizioni, per $x = a + th$ e $t \in [0, n]$:

1. $\underbrace{x_i = a + ih}_{442} \Rightarrow x_i \rightarrow i, \quad i = 0, \dots, n,$
2. $dx = hdt \quad (h = \frac{b-a}{n}, \text{ ovvero } h \text{ di (5.2)}),$

è la seguente:

$$\begin{aligned} \int_a^b L_{in}(x)dx &= h \int_0^n L_{in}(a + th)dt \stackrel{x=a+th}{=} h \int_0^n \prod_{j=0, j \neq i}^n \frac{(a + th) - \overbrace{(a + ih)}^{x_i}}{(a + ih) - \overbrace{(a + jh)}^{x_j}} dt \\ &\stackrel{443}{=} h \underbrace{\int_0^n \prod_{j=0, j \neq i}^n \frac{t - j}{i - j} dt}_{c_{in}} \equiv \frac{b-a}{n} c_{in} = h c_{in}. \end{aligned}$$

Pertanto, la **formula di Newton-Cotes di grado n** è esprimibile come:

$$I_n(f) \approx h \sum_{i=0}^n f_i c_{in}, \quad c_{in} = \int_0^n \prod_{j=0, j \neq i}^n \frac{t - j}{i - j} dt. \quad (5.3)$$

Alcune proprietà dei coefficienti $\{c_{in}\}_{i \in \{0, \dots, n\}}$ sono le seguenti:

1. ⁴⁴⁴Dalla simmetria delle ascisse $\{x_i\}$ nell'intervallo $[a, b]$ segue che:

$$c_{in} = c_{n-i,n}, \quad i = 0, \dots, n. \quad (5.4)$$

2. ⁴⁴⁵

$$\frac{1}{n} \sum_{i=0}^n c_{in} = 1. \quad (5.5)$$

⁴⁴¹Questa è una formulazione ingombrante, cambiando intervallo è necessario ricalcolare gli f_i (ovvero gli $f(x_i)$). Tramite la sommatoria è ottenuta una combinazione lineare dei valori della funzione f pesata dagli integrali (i quali sono integrali del polinomio di Lagrange).

⁴⁴²Trasformazione lineare di $x = a + th$.

⁴⁴³Semplificando a e raggruppando h .

⁴⁴⁴Slide 13 PDF 25, PG 107. Questa proprietà deriva dal fatto che le ascisse di interpolazione sono simmetriche. Affinché la 1. valga è sufficiente che le ascisse siano simmetriche (non è richiesto che siano equidistanti).

⁴⁴⁵Slide 13 PDF 25, Teorema 5.1 PG 106.

Dimostrazione 2. Se considerato $\mathbf{a} = \mathbf{0}$, $\mathbf{b} = \mathbf{1}$ e $\mathbf{f}(x) \equiv \mathbf{1}$ segue che, poiché $p(x) \equiv f(x) \equiv 1$:

$$\int_0^1 1 \cdot dx = 1 \equiv I_n(1) = \frac{1}{n} \sum_{i=0}^n 1 \cdot c_{in} = \frac{1}{n} \sum_{i=0}^n c_{in}. \quad ^{446}$$

□

Riassumendo: $I(f) \approx I(p_n) \equiv I_n(f) = \frac{b-a}{n} \sum_{i=0}^n f_i \int_0^n \prod_{j=0, j \neq i}^n \frac{t-j}{i-j} dt$.

Utilizzando (5.3)-(5.5):

1. **$n = 1$** : c_{01}, c_{11} sono i coefficienti (da calcolare, anche se non realmente) e dalle precedenti proprietà 1. e 2. segue che $c_{01} = c_{11} = \frac{1}{2}$. Pertanto, quando il gardo è 1, è **definita la formula dei trapezi**:

$$I_1(f) = \frac{b-a}{2} (f(a) + f(b)), \quad (5.6)$$

la quale ha un importante significato geometrico, visibile in Figura 36, ovvero quello di approssimare, nel caso di funzioni non negative, l'area sottesa dal grafico di $f(x)$ con quella del trapezio avente come vertici i punti $(a, 0), (a, f(a)), (b, f(b)), (b, 0)$.

2. **$n = 2$** : In questo caso sono presenti 3 coefficienti, ovvero c_{02}, c_{12}, c_{22} , t.c.:

$$P1: c_{02} + c_{12} + c_{22} = 2;$$

$$P2: c_{02} = c_{22} \text{ (per la sommatoria).}$$

Quindi è necessario calcolare un solo elemento, c_{22} , e non tre. È possibile calcolarlo come segue:

$$\begin{aligned} c_{22} &= \int_0^2 \frac{\frac{t}{2} \frac{t-1}{1}}{\frac{1}{2}} dt = \frac{1}{2} \int_0^2 (t^2 - t) dt = \\ &= \frac{1}{2} \left[\frac{t^3}{3} - \frac{t^2}{2} \right]_0^2 = \frac{1}{2} \left(\frac{8}{3} - 2 \right) = \frac{1}{3}. \end{aligned}$$

Quindi $c_{02} = c_{22} = \frac{1}{3} \Rightarrow c_{12} = 2 - \frac{2}{3} = \frac{4}{3}$. La conseguenza di quanto scritto è la **formula di Simpson**, dato $x_0 = a, x_2 = b$ e $x_1 = \frac{a+b}{2}$:

$$I_2(f) = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right), \quad (5.7)$$

5.1.1 Condizionamento di una formula di Newton-Cotes

⁴⁴⁷ Data una perturbazione $\tilde{f}(x)$ di $f(x)$ e c_{in} , definita come in (5.3), è possibile studiare come $\tilde{f}(x)$ influisca su (5.6), studiandone il condizionamento. In analogia con quanto visto per il condizionamento di (5.1), è ottenuto quanto segue:

⁴⁴⁶Se $n = 1$ sono presenti due coefficienti e non è necessario calcolarli perché è noto che siano uguali (la somma c vale $\frac{1}{2}$). Se $n = 2$ sono presenti tre coefficienti, è calcolato il primo (il quale è uguale all'ultimo) e l'altro è ottenuto tramite differenza tra il risultato e i due coefficienti uguali. Il calcolo dei coefficienti è spiegato tra poche righe.

⁴⁴⁷Slide 5-6 PDF 26, PG 108.

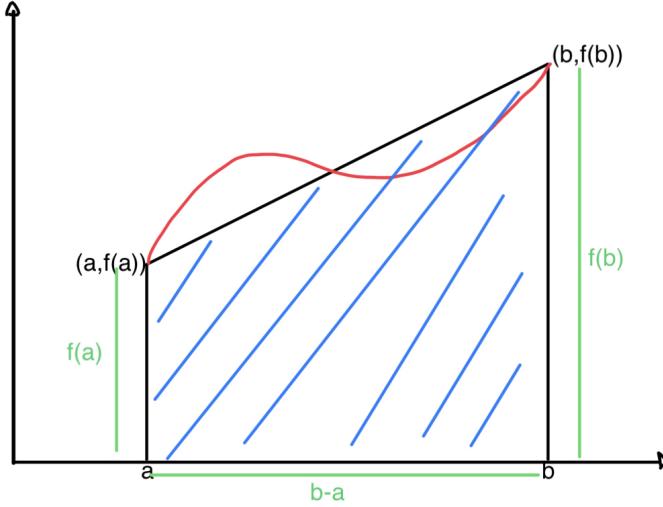


Figura 36: Esempio del metodo dei trapezi ($f(a)$, $b - a$ ed $f(b)$ sono lunghezze).

$$\begin{aligned}
 |I_n(f) - I_n(\tilde{f})| &\stackrel{448}{=} \frac{b-a}{n} \left| \sum_{i=0}^n c_{in} f_i - \sum_{i=0}^n c_{in} \tilde{f}_i \right| = \\
 &\stackrel{449}{\leq} \frac{b-a}{n} \sum_{i=0}^n \left| c_{in} (f_i - \tilde{f}_i) \right| = \\
 &\leq \left(\frac{b-a}{n} \sum_{i=0}^n |c_{in}| \right) \max_{i=0,\dots,n} |f_i - \tilde{f}_i| \leq \\
 &\underbrace{\left(\frac{b-a}{n} \sum_{i=0}^n |c_{in}| \right)}_{\kappa_n} \cdot \underbrace{\|f - \tilde{f}\|}_{\max_{a \leq x \leq b} \|f(x) - \tilde{f}(x)\|}
 \end{aligned}$$

dove κ_n è il **numero di condizionamento del problema** e $\|f - \tilde{f}\|$ misura l'**errore in ingresso**.

Osservazione 5.1. ⁴⁵⁰ Se

$$\forall i = 0, \dots, n : c_{in} \leq 0, \quad (5.8)$$

allora:

$$\kappa_n = \frac{b-a}{n} \sum_{i=0}^n |c_{in}| \stackrel{451}{=} \frac{b-a}{\mathcal{K}} \sum_{i=0}^n c_{in} \stackrel{452}{=} b-a \equiv \kappa. \quad (5.9)$$

Se i coefficienti della formula di Newton-Cotes sono non negativi allora il numero di condizionamento della formula di Newton-Cotes coincide con quello dell'integrale. Questo non significa che è ben condizionato: se l'integrale è ben condizionato allora lo è anche la funzione, se l'integrale è mal condizionato lo è anche la funzione. Ciò può

⁴⁴⁸Supposto di aver trasformato il problema in modo tale che $b > a$.

⁴⁴⁹Diseguaglianza triangolare.

⁴⁵⁰OSS Slide 6 PDF 26.

⁴⁵¹Eliminato il modulo perché $c_{in} \geq 0$.

⁴⁵²Per (5.5) ci sono le eliminazioni.

essere riformulato affermando che **il mal condizionamento coincide con quello del problema continuo** (l'integrale $I(f)$).

Tuttavia, la proprietà (5.8) vale solo per $n = 1, 2, \dots, 7, 9$. Per tutti gli altri valori di n compaiono pesi negativi ed il rapporto $\frac{\kappa_n}{\kappa} = \frac{1}{n} \sum_{i=0}^n |c_{in}|$ cresce molto rapidamente al crescere di n . Pertanto non è raccomandabile l'utilizzo di formule di Newton-Cotes di grado 8 o, genericamente, maggiori uguali a 10 (vedere Figura 37).

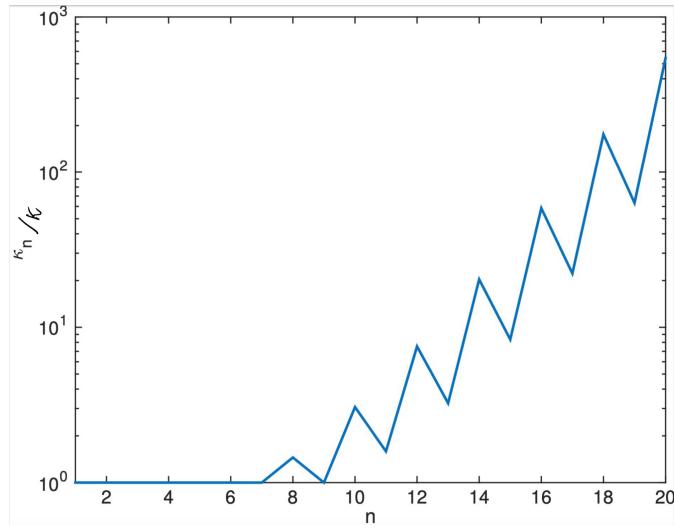


Figura 37: numero di condizionamento (5.9) delle formule di Newton-Cotes nel caso $b - a = 1$.

5.2 Errore (di quadratura) e formule composite

5.2.1 Errore di quadratura

⁴⁵³ Discutere l'accuratezza di $I_n(f)$ significa quantificare l'errore di quadratura

$$\begin{aligned}
 E_n(f) &\equiv I(f) - I_n(f) && \stackrel{454}{=} \int_a^b f(x) dx - \int_a^b p_n(x) dx \\
 &= \underbrace{\int_a^b (f(x) - p_n(x)) dx}_{455} && = \int_a^b f[x_0, \dots, x_n, x] \omega_{n+1}(x) dx \\
 & && \stackrel{456}{=} \nu_n \underbrace{f^{(n+\mu)}(\xi)}_{457} \left(\frac{b-a}{n}\right)^{n+\mu+1}, \quad \xi \in [a, b],
 \end{aligned} \tag{5.10}$$

dove:

- ν_n è una costante che dipende solo da n (ed è limitata uniformemente);

⁴⁵³ Slide 8 PDF 26 PG 108.

⁴⁵⁴ È esattamente l'integrale del polinomio interpolante la f .

⁴⁵⁵ Errore di interpolazione.

⁴⁵⁶ L'errore ha una struttura che nel caso n sia dispari è facile da dimostrare, non è così per n pari.

⁴⁵⁷ Assunto che $f \in C^{(n+\mu)}[a, b]$.

- $\mu = \begin{cases} 1, & \text{se } n \text{ è dispari;} \\ 2, & \text{se } n \text{ è pari.} \end{cases}$

Se n è dispari, è calcolata la derivata $n+1$ (dove 1 è μ). Quindi la formula è esatta se è un polinomio di grado n , quando n è dispari. Questo fatto è noto perché il polinomio di grado n interpolante la funzione, la quale è un polinomio di grado n , coincide con la funzione stessa.

Se n è pari, l'intergrale è esatto per i polinomi di grado $n+1$ e la derivata $f^{(n+\mu)}(\xi)$, con $m=2 \rightarrow n+2$, si annulla se f è un polinomio di grado $n+1$.

Quanto appena descritto è l'osservazione seguente.

Osservazione 5.2. ⁴⁵⁸ Una formula di Newton-Cotes di grado n , calcolata su $n+1$ punti, è esatta per integrandi polinomiali fino al grado:

$$\begin{cases} n, & \text{se } n \text{ dispari;} \\ n+1, & \text{se } n \text{ pari.} \end{cases}$$

Dato $E_n(f)$ in forma (5.10) non è possibile affermare che $(\frac{b-a}{n})^{n+\mu+1} \rightarrow 0$, per $n \rightarrow \infty$, a causa del condizionamento del problema. Per risolvere questo problema è necessario decrementare il rapporto $\frac{b-a}{n}$ senza fare crescere n , utilizzando un approccio simile a quello delle spline nel caso di interpolazione, ovvero: l'intervallo $[a, b]$ è suddiviso in più sottointervalli di uguale ampiezza ed è utilizzata, su ciascun intervallo, la formula di Newton-Cotes di grado k fissato. Così facendo sono ottenute le formule di Newton-Cotes composite. ⁴⁵⁹

5.2.2 Formula dei trapezi composita

Definizione 5.2 (Formula dei trapezi composita). ⁴⁶⁰ Siano $\{x_i\}$ ascisse equidistanti definite come in (5.2), è ottenuta la **formula dei trapezi composita**:

$$\begin{aligned} I(f) &= \int_a^b f(x) dx && \stackrel{461}{=} \overbrace{\sum_{i=1}^n \int_{x_{i-1}}^{x_i} f(x) dx}^{462} \\ &\approx h \sum_{i=2}^n \frac{f(x_{i-1}) + f(x_i)}{2} && \stackrel{463}{=} I_1^{(n)}(f) \\ &= h \left(\frac{f_0}{2} + \frac{f_1}{2} + \frac{f_1}{2} + \dots + \frac{f_{n-1}}{2} + \frac{f_{n-1}}{2} + \frac{f_n}{2} \right) && = h \left(\frac{f_0}{2} + \sum_{i=1}^{n-1} f_i + \frac{f_n}{2} \right) \end{aligned} \quad (5.11)$$

Osservazione 5.3. ⁴⁶³ Se $f(x)$ è periodica allora $f_n = f_0 \Rightarrow I_1^{(n)}(f) = h \sum_{i=0}^{n-1} f_i$. ⁴⁶⁴

⁴⁵⁸Slide 10 PDF 26, Corollario 5.1 PG 110.

⁴⁵⁹L'idea è la seguente: date le ascisse è applicato su ciascun sottointervallo una formula di grado k fissato. Con il numero di questi sottointervalli che tende all'infinito allora, la formula di grado k viene applicata su intervalli sempre più piccoli. Ciò significa che il termine $\frac{b-a}{n}$ si trasformerà in $(\frac{b-a}{k})^n$, dove k è il grado della formula ed n il numero di punti.

⁴⁶⁰Slide 11 PDF 26.

⁴⁶¹L'intervallo è diviso in sottointervalli e quindi l'area totale è la somma delle aree parziali, le quali sono approssimate con le funzioni trapezi.

⁴⁶²Ogni integrale è una parte dell'area del sottografo.

⁴⁶³Slide 12 PDF 25.

⁴⁶⁴La complesità del calcolo della sommatoria è $\log(n)$ perché sommando 2 a 2 i risultati sommati 2 a 2 è possibile arrivare ad un'approssimazione in $\log(n)$ passi. Se la funzione è periodica allora è possibile approssimarla con sin e cos, per le quali l'approssimazione diviene esatta, per polinomi parametrici di grado sempre più elevato, man mano che cresce n . Ciò significa che è possibile ottenere delle approssimazioni molto accurate di un segnale periodico (con sin e cos).

Definizione 5.3 (Errore di quadratura formula dei trapezi). L'errore commesso dalla formula dei trapezi è il seguente:

$$\begin{aligned}
 E_1^{(n)}(f) &\stackrel{465}{=} I(f) - I_1^{(n)}(f) \\
 &\stackrel{466}{=} \nu_1 \left(\frac{b-a}{n}\right)^3 \sum_{i=1}^n f^{(2)}(\xi_i), \quad \xi_i \in [x_{i-1}, x_i] \\
 &= \nu_1 \left(\frac{b-a}{n}\right)^3 n f^{(2)}(\xi), \quad \xi \in [a, b] \\
 &\stackrel{467}{=} \nu_1 (b-a) f^{(2)}(\xi) \left(\frac{b-a}{n}\right)^2 \rightarrow 0, \quad n \rightarrow \infty.
 \end{aligned} \tag{5.12}$$

Osservazione 5.4. La formula dei trapezi composita può essere implementata come nell'Algoritmo 8.2.

5.2.3 Formula di Simpson composita

Definizione 5.4 (Formula di Simpson composita). Siano $\{x_i\}$ ascisse equidistanziate definite come in (5.2) e dato n pari, la **formula di Simpson composita** è

$$\begin{aligned}
 I(f) &= \int_a^b f(x) dx \\
 &= \sum_{i=1}^{\frac{n}{2}} \int_{x_{2(i-1)}}^{x_{2i}} f(x) dx \\
 &\stackrel{467}{\approx} \frac{b-a}{3n} \sum_{i=1}^{\frac{n}{2}} (f(x_{2(i-1)}) + 4f(x_{2i-1}) + f(x_{2i})) \\
 &\stackrel{468}{=} \frac{b-a}{3n} (f_0 + 4f_1 + f_2 + f_2 + 4f_3 + f_4 + \dots + f_{n-2} + 4f_{n-1} + f_n) \\
 &= \underbrace{\frac{b-a}{3n} \left[4 \sum_{i=1}^{\frac{n}{2}} f_{2i-1} + 2 \sum_{i=0}^{\frac{n}{2}-1} f_{2i} - f_0 - f_n \right]}_{469} \equiv I_2^{(n)}(f).
 \end{aligned} \tag{5.13}$$

Definizione 5.5 (Errore di quadratura formula di Simpson). L'errore commesso dalla formula di Simpson è il seguente:

$$E_2^{(n)}(f) = I(f) - I_2^{(n)}(f) = \nu_2 f^{(4)}(\xi) \left(\frac{b-a}{n}\right)^5 = \nu_2 f^{(4)}(\xi) \frac{b-a}{2} \left(\frac{b-a}{n}\right)^4 \rightarrow 0, \quad n \rightarrow \infty, \quad \xi \in [a, b], \tag{5.14}$$

dove $\nu_2 = -\frac{1}{180}$ (cosa non specificata a lezione).

⁴⁶⁵ Somma di $\sum_{i=1}^n \int_{x_{i-1}}^{x_i} f(x) dx$ e $h \sum_{i=2}^n \frac{f(x_{i-1})+f(x_i)}{2}$.

⁴⁶⁶ Come in seguito, vale il teorema della sommazione discreta, ovvero: data la derivata seconda continua $f^{(2)}$ (successiva all'uguale nella sommatoria) e sommando n valori che appartengono ad n sottointervalli contigui allora $f^{(2)}$ sarà uguale ad n volte il valore della derivata seconda in un parametro dell'intervallo. Inoltre, $\nu_1 = -\frac{1}{12}$, cosa non specificata a lezione. La scelta di ν_1 viene fatta risolvendo un sistema di equazioni che impone che l'errore di quadratura sia nullo per funzioni polinomiali di grado 0 e 1. In questo caso, è possibile arrivare alla conclusione che $\nu_1 = -\frac{1}{12}$ è la scelta giusta per garantire la convergenza di ordine $O((b-a)^3)$.

⁴⁶⁷ Approssimazione che necessita di 3 ascisse, ovvero da x_0 a x_1 , da x_2 a x_4 e così via, dove l'ascissa in mezzo è quella pesata con 4.

⁴⁶⁸ 4 è associato ai numeri con indice dispari (f_1, f_3, \dots, f_{n-1}). Gli elementi con indice pari, tranne il primo e l'ultimo, hanno coefficiente 2 perché appartenenti a due intervalli congiunti.

⁴⁶⁹ Quando le sommatorie dovranno essere implementate in Matlab sarà necessario sfruttare le capacità di calcolo vettoriale di Matlab. Non sono necessari cicli ma è possibile calcolare una valutazione vettoriale della f su tutte le ascisse per poi pesare le componenti in modo opportuno. Sarà visto un esempio nelle esercitazioni.

⁴⁷⁰ $\frac{n}{2} = \frac{2}{2} + \mu = 2 + 1$, con n numero di termini.

n	$I_1^{(n)}$	$E_1^{(n)}$	$I_2^{(n)}$	$E_2^{(n)}$
2	1.5708	4.2920e-1	2.0944	-9.4395e-2
4	1.8961	1.0388e-1	2.0046	-4.5598e-3
6	1.9541	4.5903e-2	2.0009	-8.6319e-4
8	1.9742	2.5768e-2	2.0003	-2.6917e-4
10	1.9835	1.6476e-2	2.0001	-1.0952e-4
12	1.9886	1.1436e-2	2.0001	-5.2624e-5
14	1.9916	8.3996e-3	2.0000	-2.8344e-5
16	1.9936	6.4297e-3	2.0000	-1.6591e-5
18	1.9949	5.0795e-3	2.0000	-1.0348e-5
20	1.9959	4.1140e-3	2.0000	-6.7844e-6

Figura 38: Approssimazione e corrispondenti errori di (5.16).

Questa procedura si generalizza per tutte le formule di Newton-Cotes di grado k , a patto che n sia un multiplo di k , ottenendo la formula di Newton-Cotes di grado k è espressa come

$$I_k^{(n)} = h \sum_{i=0}^n f_i c_{ik} = h \sum_{i=0}^n f_i \int_0^k \prod_{j=0, j \neq i}^k \frac{t-j}{i-j},$$

e l'errore associato come

$$E_k^{(n)}(f) = \nu_k f^{(k+\mu)}(\xi) \frac{b-a}{k} \left(\frac{b-a}{n} \right)^{k+\mu}, \quad \xi \in [a, b], \quad (5.15)$$

dove $E_k^{(n)}(f) \rightarrow 0$, $n \rightarrow \infty$.

Spiegazione notazione $E_k^{(n)}(f)$: k è il grado della formula base, n sono i punti utilizzati sui quali è applicata la formula di quadratura composita. Ciò significa che per far diminuire l'errore è possibile far aumentare n lasciando k fisso.

Esempio 5.1. Come esempio, esplicato tramite la Figura 38, è approssimato tramite la formula composita dei trapezi e con quella di Simpson il seguente integrale:

$$\int_0^\pi \sin x dx \quad (= 2). \quad (5.16)$$

Dalla Figura 38 è possibile notare che l'approssimazione di Simpson sia migliore di quella dei trapezi perché più precisa.

L'espressione (5.15) permette di derivare, **a costo nullo** (cosa importante), una stima dell'errore di quadratura, $E_k^{(n)}(f)$, nel caso in cui **n sia un multiplo pari di k** .

È necessario che n sia un multiplo di k , affinché lo sia anche $\frac{n}{2}$, cosicché sia possibile valutare $I_k^{(\frac{n}{2})}(f)$. $I_k^{(\frac{n}{2})}(f)$ è calcolato su ascisse di indice pari ($0, 2, 4, \dots$) e la valutazione di $I_k^{(\frac{n}{2})}(f)$ è ottenuta senza ulteriori valutazioni della

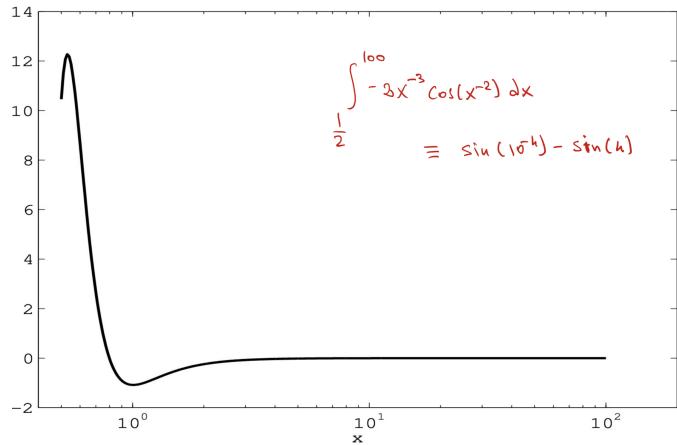


Figura 39: Approssimazione $\int_{\frac{1}{2}}^{100} -2x^{-3} \cos(x^{-2}) dx$.

funzione $f(x)$ perché è possibile utilizzare le valutazioni

$$f_{2i} \equiv f(x_{2i}), \quad i = 0, \dots, \frac{n}{2},$$

le quali sono già calcolate per la valutazione di $I_k^{(n)}(f)$.⁴⁷¹ Pertanto, analogamente a (5.14), vale:

$$I(f) - I_k^{(\frac{n}{2})}(f) = \nu_k f^{(n+\mu)}(\hat{\xi}) \frac{b-a}{k} \left(\frac{b-a}{n/2} \right)^{n+\mu} \stackrel{472}{\approx} \nu_k f^{(n+\mu)}(\xi) \frac{b-a}{k} \left(\frac{b-a}{n} \right)^{n+\mu} 2^{n+\mu}. \quad (5.17)$$

Sottraendo membro a membro (5.15) a (5.17) allora:

$$I_k^{(n)}(f) - I_k^{(\frac{n}{2})} \approx \nu_k f^{(n+\mu)}(\xi) \frac{b-a}{k} \left(\frac{b-a}{n} \right)^{n+\mu} (2^{n+\mu} - 1) \equiv (I(f) - I_k^{(n)}(f)) (2^{n+\mu} - 1).$$

In altri termini, è possibile ottenere una stima dell'errore di quadratura $E_k^{(n)}(f)$ come:

$$E_k^{(n)}(f) \approx \frac{I_k^{(n)}(f) - I_k^{(\frac{n}{2})}(f)}{2^{n+\mu} - 1} \equiv \widehat{E}_k^{(n)}(f).$$

Applicando quanto scritto all'esempio (5.16) sono derivate le Figure 39-40.

In Figura 39 il numero di condizionamento è 99.5, ovvero l'ampiezza dell'intervallo. Il problema è che nell'estremo inferiore, se $x \rightarrow 0$, allora x^{-2} e x^{-3} diventano molto grandi e $\cos(x^{-2})$, per $x \rightarrow 0$, oscilla molto. La parte sinistra del grafico è la parte in cui il grafico inizia ad oscillare molto. La questione importante è che la funzione è liscia poco dopo 10^0 e oscilla poco prima di 10^0 . Se fosse utilizzata una mesh costante accadrebbe che, per ottenere la

⁴⁷¹Sono calcolate le valutazioni funzionali di f per calcolare $I_n^k(f)$ e utilizzate quelle con indice pari così da calcolare I_n^k . Il costo della formula di quadratura è maggiorato in un numero di valutazioni di funzioni con costo nullo.

⁴⁷²Fingendo che $\hat{\xi} = \xi$ e trasformando $\frac{n}{2}$ in $2^{n+\mu}$ allora è possibile l'approssimazione.

n	$I_1^{(n)}$	$E_1^{(n)}$	$\hat{E}_1^{(n)}$	$I_2^{(n)}$	$E_2^{(n)}$	$\hat{E}_2^{(n)}$
2	1.5708	4.2920e-1	5.2360e-01	2.0944	-9.4395e-2	—
4	1.8961	1.0388e-1	1.0844e-01	2.0046	-4.5598e-3	-5.9890e-03
6	1.9541	4.5903e-2	4.6766e-02	2.0009	-8.6319e-4	—
8	1.9742	2.5768e-2	2.6038e-02	2.0003	-2.6917e-4	-2.8604e-04
10	1.9835	1.6476e-2	1.6586e-02	2.0001	-1.0952e-4	—
12	1.9886	1.1436e-2	1.1489e-02	2.0001	-5.2624e-5	-5.4038e-05
14	1.9916	8.3996e-3	8.4279e-03	2.0000	-2.8344e-5	—
16	1.9936	6.4297e-3	6.4462e-03	2.0000	-1.6591e-5	-1.6839e-05
18	1.9949	5.0795e-3	5.0899e-03	2.0000	-1.0348e-5	—
20	1.9959	4.1140e-3	4.1208e-03	2.0000	-6.7844e-6	-6.8489e-06

Figura 40: Confronto fra errore di quadratura approssimazione con formula di quadratura dei trapezi e di Simpson.

tolleranza desiderata, diventerebbero necessari punti molto vicini prima di 10^0 , i quali dovrebbero essere riportati anche nella parte liscia. Il problema posto è quello di definire un approccio adattivo per la definizione delle ascisse di interpolazione in modo adattivo, infatti sui punti laddove la funzione è più variabile saranno inseriti più punti ed invece meno dove la funzione è liscia. Per quanto appena scritto, sono introdotte le funzioni di Newton-Cotes adattive della Sezione 5.3. In Figura 40 è possibile notare che tanto più cresce n più l'errore diminuisce e quindi aumenta la precisione. È necessario trattare il fatto che, nell'implementazione di queste formule, le ascisse sono tutte equidistanti. Le ascisse equidistanti sono definite come (5.2).

5.3 Formule di Newton-Cotes adattive

⁴⁷³ Il problema è trovare un metodo per definire le ascisse in base al comportamento della funzione. Dal punto di vista algoritmico è interessante ciò che sarà visto nell'implementazione dei casi più semplici, ovvero dei trapezi e di Simpson, con Matlab, tramite function ricorsive. In ogni caso è sfruttato il fatto che esiste un'espressione asintotica dell'errore quadratico.

5.3.1 Formula dei trapezi

In questo caso, con $n = 1$, è ottenuto (da (5.10))

$$I(f) - I_1(f) = \nu_1 f^{(2)}(\xi)(b-a)^3 \quad \xi \in [a, b], \quad (5.18)$$

ed è ottenuto che, da (5.12), applicando la formula composita su due sottointervalli ($n = 2$), ciò che segue:

$$I(f) - I_1^{(2)}(f) = \nu_1 f^{(2)}(\xi)(b-a) \left(\frac{b-a}{2} \right)^2, \quad \xi \in [a, b], \quad (5.19)$$

dove $\nu_1 = -\frac{1}{12}$

Supponendo che gli ξ di (5.19) da (5.18) siano simili, sottraendo (5.19) da (5.18) è ottenuto che

$$I_1^{(2)}(f) - I_1(f) \approx \nu_1 f^{(2)}(\xi)(b-a) \left(\frac{b-a}{2} \right)^2 (4-1) \equiv (I(f) - I_1^{(2)}(f)) \frac{3}{474}$$

allora

$$E_1^{(2)}(f) \equiv I(f) - I_1^{(2)}(f) \approx \frac{I_1^{(2)}(f) - I_1(f)}{3}.$$

⁴⁷³Slide 7-12 PDF 27, PG 111-114.

La procedura adattiva è generata con il prossimo passo: Supponendo di dover calcolare $I(f)$ con una accuratezza tol (parametro prefissato, ovvero scelto dall'utente), se $|E_1^{(2)}| \leq \text{tol}$ è terminata la procedura, altrimenti è riapplicata la stessa procedura sui due sottointervalli $[a, \frac{a+b}{2}], [\frac{a+b}{2}, b]$, con tolleranza $\text{tol}/2$.

In questo modo è definito, in modo adattivo, l'insieme dei punti su quali sono applicate le formule base e la sua composita raddoppiata.

Quindi il passo di raffinamento esplicito funziona nel seguente modo: se è soddisfatta l'accuratezza richiesta la procedura si ferma, altrimenti il problema è diviso in 2 sottoproblemi, ai quali è applicato separatamente la stessa procedura con tolleranza dimezzata. Il dimezzamento è applicato affinché la somma degli errori sia sempre piccola. Se questa procedura è applicata ricorsivamente laddove l'errore è maggiore allora questo sarà affinato maggiormente, mentre qualora l'errore soddisfa il requisito di accuratezza la procedura si ferma. Questa procedura definisce adattivamente le ascisse, sulle quali sono definite le formule di quadratura.

5.3.2 Formula di Simpson

Similmente, per la formula di Simpson ($n = 2$ su $[a, b]$), è ottenuto che:

$$I(f) - I_2(f) = \nu_2 f^{(4)}(\xi) \left(\frac{b-a}{2} \right)^5, \quad \xi \in [a, b]. \quad (5.20)$$

Se è applicata la formula di Simpson composita con $n = 4$ su $[a, b]$, è ottenuto:

$$I(f) - I_2^{(4)}(f) = \nu_2 f^{(4)}(\hat{\xi}) \frac{b-a}{2} \left(\frac{b-a}{4} \right)^4, \quad \hat{\xi} \in [a, b], \quad (5.21)$$

dove in (5.20) e (5.21) $\nu_2 = -\frac{1}{90}$.

Con $\hat{\xi} \approx \xi$ (5.20) diviene

$$I(f) - I_2(f) \approx \nu_2 f^{(4)}(\hat{\xi}) \frac{b-a}{2} \left(\frac{b-a}{4} \right)^4 \cdot 16. \quad (5.22)$$

Sottraendo (5.22) da (5.21) è ottenuto

$$I_2^{(4)}(f) - I_2(f) \approx \nu_2 f^{(4)}(\hat{\xi}) \frac{b-a}{2} \left(\frac{b-a}{4} \right)^4 (16 - 1) \equiv E_2^{(4)}(f) \cdot 15.$$

Da questo risultato è ottenuto che l'errore di quadratura $E_2^{(4)}(f)$ può essere stimato come

$$E_2^{(4)}(f) \approx \underbrace{\frac{I_2^{(4)}(f) - I_2(f)}{15}}_{\text{Numero}}. \quad (5.23)$$

⁴⁷⁶ Pertanto, è richiesta un'accuratezza tol per il calcolo di $I(f)$: se $|E_2^{(4)}(f)| \leq \text{tol}$ la procedura si arresta, altrimenti riapplicando stessa procedura sui due sottointervalli $[a, \frac{a+b}{2}]$ e $[\frac{a+b}{2}, b]$, con tolleranza $\text{tol}/2$.

⁴⁷⁴Dovuto a $\left(\frac{b-a}{2}\right)^2$ in (5.19) ed a $(b-a)^3$ in (5.18).

⁴⁷⁵ $\mu = 1$ perché k è pari.

⁴⁷⁶È importante notare, sarà fatto nell'esercitazione, che quando sono replicate le procedure, in modo ricorsivo, come vengono riciclate le valutazioni già calcolate, passandole come parametro, al fine di costruire l'albero ricorsivo.

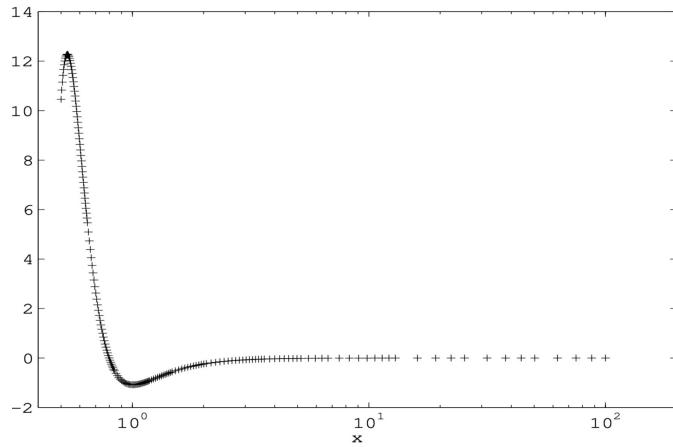


Figura 41: Simpson adattivo con $\text{tol} = 10e - 5$.

Ciò che è importante è notare che il procedimento è adattivo ed automatico. La formula dei trapezi è quella meno accurata. La formula di Simpson ha un ordine di accuratezza che è il quadrato di quello dei trapezi. Inoltre, osservando $E_1^{(n)}$ e $E_2^{(n)}$ è possibile notare che $E_1^{(n)}$ è quasi il quadrato di $E_2^{(n)}$. Questo deriva dal fatto che $\mu = 2$, per la formula di Simpson e $\mu = 1$ per la formula dei trapezi.

Osservazione 5.5. ⁴⁷⁷ help integral.

⁴⁷⁷Slide 10 PDF 27.

Algoritmo 5.1 Implementazione algoritmo adattivo dei trapezi.

```
function I2 = adaptrap(f, a, b, tol, fa, fb)
%
% Utilizzo: tab I2 = adaptrap(f, a, b, tol) [questa e' un'interfaccia utente
% , viene invocata la prima volta. fa e fb sono utilizzati per
% implementare la funzione ricorsivamente, quindi dalla seconda
% invocazione della function] [Nel caso della function che implementa la
% funzione Simpson e' necessario prevere 3 valutazioni funzionali, non
% come nelle implementazioni della formula dei trapezi, la quale richiede
% due valutazioni in input]
%
% Input:
%   f - function handler funzione integranda;
%   a,b - estremi intervallo di integrazione;
%   tol - accuratezza richiesta.
%
% Output:
%   I2 - approssimazione ottenuta.
if nargin < 4
    error('numero argomenti insufficienti')
end
if tol <= 0
    error('tolleranza nulla o minore di 0')
end
if nargin == 4
    fa = feval(f, a); fb = feval(f, b);
end
h = b - a;
x1 = (a + b)/2;
f1 = feval(f, x1); %valutazione funzionale nel punto medio
I1 = (h/2)*(fa+fb);
I2 = (I1 + h*f1)/2;
e = abs(I2-I1)/3; %stima dell'errore
if e > tol
    I2 = adaptrap(f, a, x1, tol/2, fa, f1)...
        + adaptrap(f, x1, b, tol/2, f1, fb);
end
return
```

6 Esercitazione capitoli 1 e 2

6.1 A.A. 2022/23

1. Approssimando $\pi = 3.14159265\dots$ con 3.14:

- qual è l'errore assoluto commesso?
- qual è quello relativo?

(approssimare la risposta a due cifre significative).

2. Come si definisce la precisione di macchina di un'aritmetica finita? Qual è il suo significato?

3. Qual è la precisione di macchina della singola e doppia precisione IEEE? Dedurre il numero di cifre decimali approssimativamente disponibili per la mantissa.

4. Cosa è il fenomeno della cancellazione numerica?

5. Come si definisce l'ordine di convergenza di un metodo per la ricerca degli zeri di una funzione?

6. Qual è il numero massimo di iterazioni che richiederà il metodo di bisezione per determinare la radice di una funzione assegnata con tolleranza (assoluta) 10^{-3} , se l'intervallo di confidenza iniziale è $[33, 37]$?

7. Derivare il metodo di Newton per la ricerca della radice di una funzione e dimostrare che esso converge quadraticamente a radici semplici.

8. Calcolare il numero di condizionamento della radice nulla di

$$f(x) = 3e^x - 2 \cos x - 1.$$

9. Definire la molteplicità di una radice. Calcolare la molteplicità della radice nulla di

$$f(x) = e^{x^2} - 1.$$

Perché il calcolo di una radice nulla è un problema malcondizionato?

10. Scrivere una function Matlab che implementi efficientemente il metodo di Newton.

1. ⁴⁷⁸ $x = 3.14159265\dots, \tilde{x} = 3.14 \Rightarrow \begin{cases} |\Delta x| = |\tilde{x} - x| = 0.0015926\dots \approx 1.6 \times 10^{-3}, \\ |\xi_x| = \frac{|\Delta x|}{|x|} = \frac{1.6 \times 10^{-3}}{3.1415\dots} \approx 5 \times 10^{-4}. \end{cases}$

2. ⁴⁷⁹ La precisione di macchina di un'aritmetica finita rappresenta una maggiorazione **uniforme** dell'errore relativo di rappresentazione, per numeri rappresentati da numeri di macchina normalizzati. Se è utilizzata una base b , con m cifre per la mantissa, essa vale: $u = \begin{cases} b^{1-m}, & \text{in caso di rappresentazione con troncamento;} \\ \frac{1}{2}b^{1-m}, & \text{in caso di rappresentazione con arrotondamento.} \end{cases}$

⁴⁷⁸Slide 3 PDF 13.

⁴⁷⁹Slide 3 PDF 13.

3. ⁴⁸⁰ Nella precisione IEEE è utilizzata la rappresentazione in base 2 con arrotondamento alla 24-esima cifra binaria. Pertanto la precisione di macchina è

$$u = \frac{1}{2} \cdot 2^{1-24} = 2^{-24} \approx \frac{10^{-6}}{16} \approx 0.7 \cdot 10^{-7},$$

ovvero, poco più di 7 cifre decimali significative.

Per la doppia precisione, l'unica differenza consiste nel numero di cifre binarie significative, sono 53. Pertanto,

$$u = \frac{1}{2} \cdot 2^{1-53} = 2^{-53} \approx 10^{-16},$$

ovvero, circa 16 cifre decimali.

4. ⁴⁸¹ La cancellazione numerica consiste nella perdita di cifre significative, nel risultato, derivante dalla somma di addendi quasi opposti. Ciò è dovuto al malcondizionamento di questa operazione. Dati x e y numeri da sommare, il numero di condizionamento della somma è rappresentato da $\kappa = \frac{|x|+|y|}{|x+y|}$, il quale non è limitato superiormente se $x \approx -y$.

5. ⁴⁸² Sia $x_{n+1} = \Phi(x_n)$, $n = 0, 1, \dots$, denota un generico metodo iterativo per la ricerca di una radice \bar{x} dell'equazione $f(x) = 0$. Supposto che $x_n \rightarrow \bar{x}$, per $n \rightarrow \infty$ e con $e_n = |x_n - \bar{x}|$ il corrispondente errore al passo n . Il metodo converge con ordine $p \geq 1$ alla radice, se p è il più grande valore reale per cui

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n^p} = c < \infty,$$

dove c è la costante asintotica dell'errore. Per $n \gg 1 \Rightarrow e_{n+1} \approx c \cdot e_n^p$.

6. ⁴⁸³ Il metodo di bisezione dimezza, ad ogni iterazione, l'ampiezza dell'intervallo di confidenza. Pertanto, l'approssimazione al passo n , avrà una accuratezza $2^{-n}(b-a)$, essendo $b-a$ l'ampiezza dell'intervallo iniziale. In questo caso $b-a=4$, per cui è ottenuto, imponendo $2^{2-n} \leq 10^{-3}$, il numero massimo di iterazioni. Dato che $10^{-3} \approx 2^{-10}$ è ottenuto quanto segue: $2^{2-n} \leq 2^{-10} \Rightarrow 2-n \leq -10 \Rightarrow n \geq 12 \Rightarrow n = 12 (= \lceil \log_2(37-34) - \log_2(10^{-3}) \rceil)$.

7. ⁴⁸⁴ Il metodo di Newton è ottenuto ricercando, ad ogni passo, la radice della retta tangente al grafico della funzione nell'approssimazione corrente (come in Figura 42).

La retta tangente il grafico di $f(x)$ nel punto $(x_n, f(x_n))$ è $y - f(x_n) = f'(x_n)(x - x_n)$, essendo $f'(x_n)$ la derivata di $f(x)$. Ponendo $y = 0$, è ricavata

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \dots$$

⁴⁸⁰Slide 4 PDF 13. Nella rappresentazione binaria le cifre sono normalizzate, tutto cambia se è denormalizzato. La prima cifra considerata 1 quindi non è memorizzata memorizzando con 23 bit 24. Stesso cosa vale per la precisione macchina in doppia precisione.

⁴⁸¹Slide 4 PDF 13.

⁴⁸²Slide 5 PDF 13.

⁴⁸³Slide 6 PDF 13.

⁴⁸⁴Slide 7 PDF 13.

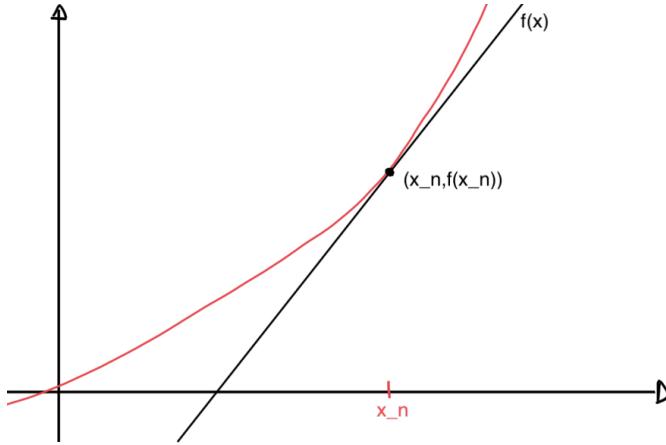


Figura 42: Approssimazione esercizio 10.

Dimostrare la convergenza quadratica ad una radice semplice di \bar{x} di $f(x)$ significa che $f'(x) \neq 0$, sotto ipotesi che $f \in C^{(2)}$ in un intorno di \bar{x} , per un opportuno intorno, per il Teorema della permanenza del segno, significa:

$$\begin{aligned} 0 &= f(\bar{x}) \\ &= f'(x_n) \left(\frac{f(x_n)}{f'(x_n)} - x_n + \bar{x} \right) + \frac{f''(\xi_n)}{2} (\bar{x} - x_n)^2 \quad \stackrel{485}{=} f(x_n) + (\bar{x} - x_n) f'(x_n) + \frac{f''(\xi_n)}{2} (\bar{x} - x_n)^2 \\ &\quad \stackrel{486}{=} f'(x_n)(x_{n+1} - \bar{x}) + \frac{f''(\xi_n)}{2} (\bar{x} - x_n)^2 \\ &\quad \xi_n \in I(\bar{x}, x_n) \end{aligned}$$

Da questo segue che $f'(x_n)(\bar{x} - x_{n+1}) = \frac{f''(\xi_n)}{2} (\bar{x} - x_n)^2$. Ponendo $e_n = \bar{x} - x_n$ e dato che $f'(\bar{x}) \neq 0$ è ottenuto ciò che segue:

$$\frac{e_{n+1}}{e_n^2} = \frac{f''(\xi_n)}{2f'(x_n)} \rightarrow \lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^2} = \lim_{n \rightarrow \infty} \left| \frac{f''(\xi_n)}{2f'(x_n)} \right| = \left| \frac{f''(\bar{x})}{2f'(\bar{x})} \right| < \infty.$$

8. ⁴⁸⁷ Il numero di condizionamento di una radice è dato da $\frac{1}{|f'(\bar{x})|}$, se \bar{x} è la radice. In questo caso $f'(x) = 3e^x + 2 \sin(x) \rightarrow f'(0) = 3$. Pertanto, il numero di condizionamento della radice nulla di $f(x)$ vale $\frac{1}{3}$

Osservazione sugli scritti: Meglio scrivere ciò di cui siamo sicuri in termini di passaggi intermedi, altrimenti è meglio scrivere solo il risultato finale.

9. ⁴⁸⁸ La radice \bar{x} di $f(x) = 0$ ha molteplicità m se: $\begin{cases} f(\bar{x}) = f'(\bar{x}) = \dots = f^{(m-1)}(\bar{x}) = 0, \\ f^{(m)}(\bar{x}) \neq 0. \end{cases}$

Se $m = 1$ la radice è **semplice**, se $m > 1$ la radice è **multipla**. La determinazione di una radice multipla è un problema malcondizionato, per il numero della radice, $\frac{1}{f'(\bar{x})}$, è infinito, essendo $f'(\bar{x}) = 0$.

⁴⁸⁵ $\frac{f(x_n)}{f'(x_n)} - x_n$ è Newton al passo $n + 1$, ovvero x_{n+1} .

⁴⁸⁶ Intervallo aperto nel quale gli estremi sono massimo e minimo. Se $x_n \rightarrow \bar{x}$ significa che anche $\xi_n \rightarrow \bar{x}$.

⁴⁸⁷ Slide 9 PDF 13.

⁴⁸⁸ Slide 9 PDF 13.

Data $f(x) = e^{x^2} - 1 \Rightarrow f(0) = 0$. Inoltre, $f'(x) = 2xe^{x^2} \Rightarrow f'(0) = 0$ ⁴⁸⁹. Ancora, $f''(x) = 2e^{x^2} + 4x^2e^{x^2} \Rightarrow f''(0) = 2 \neq 0$. Pertanto, la radice ha **molteplicità 2**.

10. L'implementazione richiesta è nell'Algoritmo 2.2.

⁴⁸⁹Quindi, da qui, è possibile capire che la radice non è semplice.

7 Esercitazione capitolo 3

7.1 A.A. 2022/23

1. Scrivere una function Matlab che risolva efficientemente un sistema triangolare inferiore.
2. Definire la fattorizzazione LU di una matrice nonsingolare. Dimostrare che, se fattorizzabile, la fattorizzazione LU di una matrice è unica.
3. Sotto quali condizioni esiste la fattorizzazione LU di una matrice?
4. Definire cosa si intende per matrice a diagonale dominante. Dimostrare che una matrice diagonale dominante è fattorizzabile LU .
5. Definire cosa si intende per matrice simmetrica e definita positiva (sdp). Dimostrare che una matrice sdp è fattorizzabile LU .
6. Dimostrare che una matrice simmetrica e definita positiva è fattorizzabile nella forma LDL^T , con L triangolare inferiore a diagonale unitaria, e D matrice diagonale.
7. Definire cosa si intende per norma indotta su matrice. Dare qualche esempio di tali norme.
8. Definire cosa è il numero di condizionamento di una matrice. Spiegarne il significato.
9. Definire la soluzione nel senso dei minimi quadrati di un sistema lineare sovra-determinato a rango pieno. Dimostrarne l'esistenza ed unicità.
10. Calcolare la matrice elementare di Householder H relativa al vettore:

$$\mathbf{z} = \begin{pmatrix} -1 \\ 2 \\ -2 \end{pmatrix}.$$

Quanto vale il prodotto $H\mathbf{z}$?

11. Definire il metodo di Newton per sistemi nonlineari, e dettagliarne l'implementazione.
1. ⁴⁹⁰ La function è implementata nell'Algoritmo 7.1.
2. ⁴⁹¹ Sia $A \in \mathbb{R}^{n \times n}$, $\det(A)$.
(Risposta alla prima parte di domanda:) A è fattorizzabile LU , se $A = LU$, con:

- L triangolare inferiore a diagonale unitaria;
- U triangolare superiore.

⁴⁹⁰Slide 3 PDF 14.

⁴⁹¹Slide 3 PDF 14.

Algoritmo 7.1 Implementazione efficiente risolutore sistema triangolare inferiore.

```
function y = trilow(L, b)
%
%   y = trilow(L, b)
%
%   Risolve un sistema triangolare inferiore
%
% Input:
%   L - matrice dei coefficienti;
%   b - termine noto;
%
% Output:
%   y - vettore soluzione.
%
% Le prossime tre righe sono controlli iniziali.
[m,n] = size(L);
if m ~= n, error('Matrice non quadrata'), end
if n~= length(b), error('Termine noto non consistente'), end

y=b(:); %trasformazione di b in vettore colonna

for i = 1 : n
    if L(i,i) == 0, error('Matrice singolare'), end
    y(i) = y(i)/L(i,i);
    y(i+1:n) = y(i+1:n) - L(i+1:n, i)*y(i);
end
return
```

(Risposta alla seconda parte) Unicità della fattorizzazione: È necessario dimostrare che se $A = L_1 U_1$ è un'ulteriore fattorizzazione LU di A , allora $L = L_1$, $U = U_1$. Infatti:

$$0 \neq \det(A) = \det(LU) = \underbrace{\det(L)}_{492} \det(U) = \det(U).$$

Pertanto, da $LU = L_1 U_1$, segue che, moltiplicando a destra per U^{-1} , $L = L_1 U_1 U^{-1}$. Da questa, moltiplicando a sinistra per L_1^{-1} , segue che:

$$L_1^{-1} L = U_1 U^{-1}. \quad (7.1)$$

È possibile osservare ciò che segue:

- L_1^{-1} è una matrice triangolare inferiore a diagonale unitaria;
- U^{-1} è una matrice triangolare superiore.

Quindi $L_1^{-1} L$ è una matrice triangolare inferiore a diagonale unitaria e $U_1 U^{-1}$ è una matrice triangolare superiore. Pertanto, le due matrici in (7.1) sono diagonali. Poiché la diagonale di $L_1^{-1} L$ è unitaria, questa è la matrice indentità. Dalle uguaglianze allora

- $L_1^{-1} L = I$,
- $U_1 U^{-1} = I$,

segue che $L = L_1 \wedge U_1 = U$, ovvero la fattorizzazione LU è unica. \square

IMPORTANTE: I passaggi come quelli sopra è necessario che siano il piu' dettagliati possibile affinché Luigi sappia che abbiamo studiato.

3. Sia $A \in \mathbb{R}^{n \times n}$, $\det(A) \neq 0$. Denotando con A_k la sottomatrice principale di ordine k di A , allora:

$$A = LU \iff \forall k = 1, \dots, n : \det(A_k) \neq 0.$$

Questa ultima riga non è sufficiente, è necessario specificare $A \in \mathbb{R}^{n \times n}$.

4. ⁴⁹³ Sia $A \in \mathbb{R}^{n \times n}$. Allora $A = (a_{ij})$ è:

- **diagonale dominante per righe**, se

$$|a_{ii}| > \sum_{j \neq i, j=1}^n |a_{ij}|, \quad \forall i = 1, \dots, n; \quad (7.2)$$

- **diagonale dominante per colonne**, se $|a_{jj}| > \sum_{i \neq j, i=1}^n |a_{ij}|$, $\forall j = 1, \dots, n$.

Valgono le seguenti proprietà:

P1) A è d.d. per righe $\iff A^T$ è d.d. per colonne;

⁴⁹²Uguale ad 1 perché a diagonale unitaria 1.

⁴⁹³Slide 6 PDF 14.

P2) Se A è d.d. per righe (o colonne), allora, detta A_k la sottomatrice principale di ordine k di A , essa sarà d.d. per righe (o colonne). (Dimostrazione:) Infatti, dato un generico $k \in \{1, \dots, n\}$, risulterà che dalla (7.2) segue che $\forall i = 1, \dots, k$:

$$|a_{ii}| > \sum_{j \neq i, j=1}^n |a_{ij}| \geq \sum_{j \neq i, j=1}^k |a_{ij}|,$$

ovvero, A_k è diagonale d.d. per righe. Analogamente per A d.d. per colonne. \square

P3) Se A è d.d. per righe (o per colonne), allora A è non singolare. Infatti, se A fosse singolare, esisterebbe $\underline{x} \in \mathbb{R}^n$, $\underline{x} \neq 0$: $A\underline{x} = \underline{0}$. Questa equazione vale per ogni multiplo di \underline{x} quindi è possibile normalizzare \underline{x} in modo che $x_k = \max_{i=1, \dots, n} |x_i| = 1$. Pertanto, la k -esima equazione di $A\underline{x} = \underline{0}$ diviene: $\sum_{j=1}^n a_{kj}x_j = 0$. Da questo segue che $a_{kk}x_k = -\sum_{j=1, j \neq k}^n a_{kj}x_j$. Pertanto:

$$|a_{kk}| = |a_{kk}x_k| = \left| \sum_{j=1, j \neq k}^n a_{kj}x_j \right| \leq \sum_{j=1, j \neq k}^n |a_{kj}| \cdot \underbrace{|x_j|}_{\leq 1} \leq \sum_{j=1, j \neq k}^n |a_{kj}|,$$

il che contraddice l'ipotesi d.d. per righe sulla riga k . Se A è d.d. per colonne, il tutto è riapplicato a A^T , che sarà d.d. per righe (per **P1**)).

Dalle proprietà **P2**) e **P3**), segue che:

- A è d.d. per righe (o colonne) $\iff \forall k = 1, \dots, n$;
- A_k è d.d. per righe (o colonne) $\iff \forall k = 1, \dots, n$;
- $\det(A_k) \neq 0 \iff A$ è fattorizzabile LU .

5. ⁴⁹⁴ Sia $A \in \mathbb{R}^{n \times n}$. A è simmetrica e definita positiva (sdp) se:

- $A = A^T$ (A è simmetrica);
- $\forall \underline{x} \in \mathbb{R}^n \setminus \{\underline{0}\} : \underline{x}^T A \underline{x} > 0$ (A è definita positiva).

Per dimostrare che, se A è sdp, allora $A = LU$, sarà dimostrato che:

1. $\forall k = 1, \dots, n$, A_k , detta sottomatrice principale di A di ordine k , è sdp;
2. A sdp $\Rightarrow A$ è nonsingolare.

Riguardo la 2), se A non fosse singolare, allora $\exists \underline{x} \in \mathbb{R}^n \setminus \{\underline{0}\} : A\underline{x} = \underline{0} \Rightarrow \underline{x}^T A \underline{x} = 0$, contraddicendo la definita positiva di A . Pertanto, A è nonsingolare.

Riguardo la 1):

$$\forall k = 1, \dots, n : A = \begin{bmatrix} A_k & B \\ C & D \end{bmatrix}, \text{ con } D \in \mathbb{R}^{(n-k) \times (n-k)}. \quad (7.3)$$

Dalla simmetria di A , segue che $A^T = \begin{bmatrix} A_k & C^T \\ B^T & D^T \end{bmatrix} = A$. Uguagliando i blocchi omologhi allora:

$$A_k = A_k^T, B = C^T, D = D^T \rightarrow A_k \text{ è simmetrica.}$$

⁴⁹⁴Slide 9 PDF 14.

Rimane da dimostrare che, $\forall \underline{y} \in \mathbb{R}^k \setminus \{\underline{0}\}$: $\underline{y}^T A_k \underline{y} > 0$. A questo fine, assegnato un generico $\underline{y} \in \mathbb{R}^k \setminus \{\underline{0}\}$, è considerato il vettore $\underline{x} = \begin{pmatrix} \underline{y} \\ \underline{0} \end{pmatrix} \in \mathbb{R}^n$. Chiaramente $\underline{x} \neq 0$. Dato che A è sdp allora da (7.3) segue che:

$$0 < \underline{x}^T A \underline{x} = (\underline{y}^T \underline{0}^T) \begin{bmatrix} A_k & B \\ C & D \end{bmatrix} \begin{pmatrix} \underline{y} \\ \underline{0} \end{pmatrix} = (\underline{y}^T \underline{0}^T) \begin{bmatrix} A_k \underline{y} \\ C \underline{y} \end{bmatrix} = \underline{y}^T A_k \underline{y}.$$

□

6. ⁴⁹⁵ È noto che, se A è sdp, allora $A = LU$ ed è noto che la fattorizzazione LU è unica. Pertanto, osservando il fattore U può essere scritto come $D\widehat{U}$, con D diagonale ed \widehat{U} triangolare superiore a diagonale unitaria, segue che $A = LU = LD\widehat{U}$. Essendo $A = A^T \Rightarrow A = LD\widehat{U} = (LD\widehat{U})^T = A^T$. Segue che $LD\widehat{U} = \widehat{U}^T DL^T$.
Essendo:

- \widehat{U}^T triangolare inferiore a diagonale unitaria,
- DL^T triangolare superiore,

per l'unicità della fattorizzazione LU segue che $L = \widehat{U}^T$. Da questo è possibile concludere che $A = LDL^T$. □

7. ⁴⁹⁶ Sia $\|\cdot\|$ una norma assegnata su vettore. È definito, data $A \in \mathbb{R}^{m \times n}$, la sua norma indotta dalla norma su vettore considerata: $\|A\| = \max_{\underline{x} \in \mathbb{R}^n, \|\underline{x}\|=1} \|A\underline{x}\|$.

Osservazione 7.1. $\|A\underline{x}\|$ è la norma di un vettore di \mathbb{R}^m .

Alcuni esempi: se $A = (a_{ij}) \in \mathbb{R}^{m \times n}$, allora:

- $\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^m |a_{ij}|$;
- $\|A\|_\infty = \max_{i=1,\dots,m} \sum_{j=1}^n |a_{ij}|$.

8. ⁴⁹⁷ Sia $A \in \mathbb{R}^{n \times n}$, $\det(A)$. Il suo numero di condizione, assegnato ad una generica norma indotta da una corrispondente norma su matrice, è definito come $\kappa(A) = \|A\| \cdot \|A^{-1}\|$. Questo misura il condizionamento del sistema lineare $A\underline{x} = \underline{b}$. Infatti, considerando perturbazioni ΔA e $\Delta \underline{b}$ su A e \underline{b} , rispettivamente, è ottenuto $(A + \Delta A)(\underline{x} + \Delta \underline{x}) = \underline{b} + \Delta \underline{b}$, con $\Delta \underline{x}$ perturbazione sul risultato. Vale che:

$$\frac{\|\Delta \underline{x}\|}{\|\underline{x}\|} \leq \kappa(A) \left(\frac{\|\Delta \underline{b}\|}{\|\underline{b}\|} + \frac{\|\Delta A\|}{\|A\|} \right) \quad \square$$

9. ⁴⁹⁸ Sia $A \in \mathbb{R}^{m \times n}$, con $m > n = \text{rank}(A)$. È definita **soluzione nel senso dei minimi quadrati** del sistema lineare $A\underline{x} = \underline{b}$, il vettore \underline{x} che minimizza la norma euclidea (al quadrato) del corrispondente vettore **residue** $\underline{r} = A\underline{x} - \underline{b}$. ⁴⁹⁹

⁴⁹⁵Slide 11 PDF 14.

⁴⁹⁶Slide 12 PDF 14.

⁴⁹⁷Slide 12 PDF 14.

⁴⁹⁸Slide 13 PDF 14.

⁴⁹⁹Necessario specificare la parte in grassetto affinché la risposta sia corretta.

Il motivo della scelta della norma euclidea è dovuto al fatto che questa è invariante per moltiplicazione del vettore in argomento per una matrice ortogonale Q :

$$\|Q\underline{v}\|_2^2 = (Q\underline{v})^T(Q\underline{v}) = \underline{v}^T \underbrace{Q^T Q}_{500} = \underline{v}^T \underline{v} = \|\underline{V}\|_2^2.$$

La soluzione ai minimi quadrati di $A\underline{x} = \underline{b}$ è ottenuta osservando che, se $A \in \mathbb{R}^{m \times n}$, $m > n = \text{rank}(A)$, allora esistono:

- $Q \in \mathbb{R}^{m \times m}$, Q ortogonale ;
- $\widehat{R} \in \mathbb{R}^{n \times n}$, \widehat{R} triangolare superiore e nonsingolare,

tali che: $A = QR$, con $R = \begin{pmatrix} \widehat{R} \\ 0 \end{pmatrix} \in \mathbb{R}^{m \times n}$. Per minimizzare:

$$\begin{aligned} \|r\|_2^2 &= \|A\underline{x} - \underline{b}\|_2^2 = \|QR\underline{x} - \underline{b}\|_2^2 = \|Q(R\underline{x} - Q^T \underline{b})\|_2^2 \stackrel{501}{=} \|R\underline{x} - \underline{g}\|_2^2 \\ &\stackrel{502}{=} \left\| \begin{bmatrix} \widehat{R} \\ 0 \end{bmatrix} \underline{x} - \begin{bmatrix} \underline{g}_1 \\ \underline{g}_2 \end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix} \widehat{R}\underline{x} - \underline{g}_1 \\ -\underline{g}_1 \end{bmatrix} \right\|_2^2 = \left\| \widehat{R}\underline{x} - \underline{g}_1 \right\|_2^2 + \left\| \underline{g}_2 \right\|_2^2 = \left\| \underline{g}_2 \right\|_2^2 \\ &= \min! \end{aligned}$$

Scegliendo \underline{x} come soluzione del sistema lineare $\widehat{R}\underline{x} = \underline{g}_1$, che esiste, ed è unica, essendo \widehat{R} non singolare.

10. ⁵⁰³ È noto che $H\underline{z} = \alpha \underline{e}_1 \in \mathbb{R}^3$, con $\alpha = \pm \|\underline{z}\|_2 = \pm 3$. La matrice H è nella forma

$$H = I - \frac{2}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T,$$

con

$$\underline{v} = \underline{z} - \alpha \underline{e}_1 = \begin{bmatrix} -1 - \alpha \\ 2 \\ -2 \end{bmatrix} = \begin{bmatrix} -1 - 3 \\ 2 \\ -2 \end{bmatrix}$$

avendo scelto $\alpha = 3$, in modo che la prima componente di \underline{v} sia ottenuta sommando quantità concordi.

Segue che $H\underline{z} = 3\underline{e}_1$. Il vettore di Householder è $\underline{v} = \begin{bmatrix} -4 \\ 2 \\ -2 \end{bmatrix}$, dal quale è ottenuto che

$$H = I - \frac{2}{\underline{v}^T \underline{v}} \underline{v} \underline{v}^T = I - \frac{2}{24} \underline{v} \underline{v}^T = I - \frac{1}{12} \underline{v} \underline{v}^T.$$

⁵⁰⁰Uguale ad I perché Q è ortogonale.

⁵⁰¹Ponendo $\underline{g} = Q^T \underline{b}$ e sfruttando l'invarianza di $\|\cdot\|_2$.

⁵⁰² $\underline{g}_1 \in \mathbb{R}^n$.

⁵⁰³Slide 15 PDF 14.

11.⁵⁰⁴ Sia $f : \mathbb{R}^m \rightarrow \mathbb{R}^m$, è definito il metodo di Newton per risolvere $f(\underline{x}) = \underline{0}$ fondamentalmente dall'iterazione seguente:

$$\underline{x}_{n+1} = \underline{x}_n - J_f(\underline{x}_n)^{-1} f(\underline{x}_n), \quad n = 0, 1, \dots \quad (7.4)$$

essendo $J_f(\underline{x}_n)$ la matrice Jacobiana di $f(\underline{x})$ calcolata in \underline{x}_n . Nella pratica (7.4) è implementata con l'obbiettivo di risolvere il sistema lineare $J_f(\underline{x}_n) \Delta \underline{x}_n = -f(\underline{x}_n)$ aggiornando $\underline{x}_{n+1} = \underline{x}_n + \Delta \underline{x}_n$, $n = 0, 1, \dots$. Pertanto, il costo per iterazione è il seguente:

1. 1 valutazione di J_f e la sua fattorizzazione;
2. 1 valutazione di f e soluzione per i fattori di J_f ;
3. 1 axpy per aggiornare l'approssimazione corrente.

Come criterio d'arresto è possibile utilizzare il seguente:

$$\|\Delta \underline{x}_n\| / (1 + |\underline{x}|) \leq tol,$$

per una tolleranza tol specificata.

⁵⁰⁴Slide 17 PDF 14.

8 Esercitazione capitoli 4 e 5

8.1 A.A. 2022/23

1. Formulare il problema dell'interpolazione polinomiale e dimostrare l'esistenza ed unicità del polinomio interpolante.
2. Scrivere la forma di Lagrange del polinomio interpolante le coppie di dati $(0, 1), (2, 3), (3, 7)$.
3. Determinare la forma di Newton del polinomio interpolante i dati $(0, 3), (2, 4), (3, 5)$. Sapendo che la derivata terza della funzione interpolanda ha norma minore o uguale a 5, dare una stima dell'errore di interpolazione nel punto $x = 4$.
4. Derivare l'espressione dell'errore nell'interpolazione polinomiale.
5. Costruire il polinomio di Hermite interpolante i dati, nella forma (x_i, f_i, f'_i) , $(0, 1, 7)$ e $(3, 4, -8)$.
6. Definire la costante di Lebesgue, e spiegarne esaurientemente il significato nell'ambito dell'interpolazione polinomiale.
7. Definire le ascisse di Chebyshev, e spiegarne il significato nell'ambito dell'interpolazione polinomiale.
8. Scrivere una *function* Matlab che calcoli le ascisse di Chebyshev per costruire il polinomio interpolante di grado n su un generico intervallo $[a, b]$.
9. Cosa è il polinomio di migliore approssimazione di una funzione, e come questo è collegato all'errore nell'interpolazione polinomiale?
10. Definire i polinomi di Chebyshev di prima specie, ed i corrispondenti polinomi monici. Qual è la caratteristica saliente di questi ultimi?
11. Definire una funzione *spline* di grado m su una partizione Δ assegnata.
12. Qual è l'unica funzione *spline* univocamente determinata dalle condizioni di interpolazione sui nodi della partizione assegnata? Scrivere esplicitamente l'espressione.
13. Quante condizioni (indipendenti tra loro) servono per individuare univocamente una funzione *spline*?
14. Qual è l'unica funzione spline univocamente determinata dalle condizioni di interpolazione sui nodi della partizione assegnata? Scrivere esplicitamente l'espressione.
15. Quante condizioni servono per determinare univocamente una spline cubica interpolante? Cosa si intende per spline cubica interpolante naturale (completa/periodica/not-a-knot)?
16. Scrivere una function Matlab che risolva efficientemente un sistema di equazioni tridiagonale (supporre che la matrice sia fattorizzabile *LU*).
17. Definire il polinomio approssimante ai minimi quadrati. Sotto quali condizioni esso esiste ed è unico?
18. Scrivere il problema algebrico che definisce il polinomio di approssimazione ai minimi quadrati di grado 2, per le coppie di dati $(1,2), (1,2.1), (2,3), (3,4), (3,3), (4,5)$.
19. Studiare il problema del condizionamento di un integrale definito.

20. Derivare le formule di quadratura di Newton-Cotes.
21. Calcolare $\int_0^\pi \cos(x)dx$ e la sua approssimazione con la formula dei trapezi e quella di Simpson, dettagliando i passaggi.
22. Qual è l'espressione dell'errore della formula di Newton-Cotes di grado k ? Quale quella della formula composita, supponendo che n sia multiplo di k ?
23. Come è possibile ottenere una stima dell'errore per la formula di Newton-Cotes di grado k , usata con n multiplo pari di k ?
24. Scrivere una function Matlab che calcoli efficientemente l'approssimazione di un integrale definito mediante la formula dei trapezi composita. Supporre che la function della funzione integranda accetti input vettoriali.

2.

$$p(x) = \sum_{i=0}^n f_i L_{in}(x), \quad L_{in}(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}, \quad i = 0, \dots, n.$$

Nel nostro caso: $\begin{cases} n = 2, \\ x_0 = 0, x_1 = 2, x_2 = 3, \\ f_0 = 1, f_1 = 3, f_2 = 7. \end{cases}$

Pertanto:

$$p(x) = 1 \cdot L_{02}(x) + 3 \cdot L_{12}(x) + 7 \cdot L_{22}(x),$$

con

$$\begin{aligned} L_{02}(x) &= \frac{(x-2)(x-3)}{(0-2)(0-3)} = \frac{(x-2)(x-3)}{6}; \\ L_{12}(x) &= \frac{(x-0)(x-3)}{(2-0)(2-3)} = -\frac{1}{2}x(x-3); \\ L_{22}(x) &= \frac{(x-0)(x-2)}{(3-0)(3-2)} = \frac{x(x-2)}{3}. \end{aligned}$$

È possibile che inserisca un esercizio trabocchetto del tipo: calcolare polinomio interpolante, una funzione (polinomio) di grado 3, fornendo 18 coppie di punti. Questo significa che è ricercato il polinomio di grado al più 17. È chiaro che con il calcolo della funzione, questo sarà di una funzione minimale.

3. In genere, si ha che

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] \omega_i(x),$$

dove $f[x_0, \dots, x_i]$ è la differenza divisa di f sulle ascisse in argomento, e

$$w_i(x) = \prod_{k=0}^{i-1} (x - x_k), \quad i = 0, \dots, n.$$

Nel nostro caso: $\begin{cases} n = 2, \\ x_0 = 0, x_1 = 2, x_2 = 3, \\ f_0 = 3, f_1 = 4, f_2 = 5. \end{cases}$

Pertanto:

a) $w_0(x) \equiv 1$, $w_1(x) = x$, $w_2(x) = x(x - 2)$,

	0	1	2
$x_0 = 0$	$f[x_0] = 3$		
$x_1 = 2$	$f[x_1] = 4$	$f[x_0, x_1] = \frac{1}{2}$	
$x_2 = 3$	$f[x_2] = 5$	$f[x_1, x_2] = 1$	$f[x_0, x_1, x_2] = \frac{1}{6}$

Concludiamo che:

$$p(x) = 3 + \frac{1}{2}x + \frac{1}{6}x(x - 2). \quad [\text{Fine prima parte del quesito}]$$

[Risposta alla seconda parte del quesito:] Sappiamo che l'errore di interpolazione è dato da:

$$e(x) = f[x_0, x_1, x_2, x] \omega_3(x), \quad \omega_3(x) = x(x - 2)(x - 3).$$

Inoltre,

$$f[x_0, x_1, x_2, x] = \frac{f^{(3)}(\xi_x)}{3!},$$

506

per un opportuno $\xi_x \in I(x, 0, 2, 3)$. Poiché $\|f^{(3)}\| \leq 5$, abbiamo che:

$$|e(4)| \stackrel{507}{\leq} \frac{\|f^{(3)}\|}{3!} |\omega_3(4)| \stackrel{508}{\leq} \frac{5}{6} \cdot \frac{8}{\|\omega_3(4)\|} = \frac{20}{3}.$$

5. In generale, si ha che:

$$\begin{aligned} p_H(x) &= f[x_0] + (x - x_0)f[x_0, x_0] + (x - x_0)^2 f[x_0, x_0, x_1] + \dots + \\ &\quad + (x - x_0)^2 \dots (x - x_{n-1})^2 (x - x_n) f[x_0, x_0, \dots, x_n, x_n]. \end{aligned}$$

Nel nostro caso: $\begin{cases} n = 1, x_0 = 0, x_1 = 3, \\ f(x_0) = 1, f(x_1) = 4, \\ f'(x_0) = 7, f'(x_1) = -8. \end{cases}$

Pertanto:

$$(x - x_0) = x, (x - x_0)^2 = x^2, (x - x_0)^2(x - x_1) = x^2(x - 3).$$

Inoltre: ⁵⁰⁹

	0	1	2	3
$x_0 = 0$	$f[x_0] = 1$			
$x_0 = 0$	$f[x_0] = 1$	$f[x_0, x_0] = f'(x_0) = 7$		
$x_1 = 3$	$f[x_1] = 4$	$f[x_0, x_1] = 1$	$f[x_0, x_0, x_1] = -2$	
$x_1 = 3$	$f[x_1] = 4$	$f[x_1, x_1] = f'(x_1) = -8$	$f[x_0, x_1, x_1] = -3$	$f[x_0, x_0, x_1, x_1] = -\frac{1}{3}$

Pertanto,

$$P_H(x) = 1 + 7x - 2x^2 - \frac{1}{3}x^2(x - 3).$$

⁵⁰⁶Massimo del valore assoluto.

⁵⁰⁷Dovuto alla norma, la quale è il massimo del valore assoluto.

⁵⁰⁸Dovuto all'ipotesi $\|f^{(3)}\| \leq 5$.

⁵⁰⁹I coefficienti in grassetto sono necessari.

8. Algoritmo 8.1.

Algoritmo 8.1 Implementazione esercizio 8.

```

function x = cheby(n, a, b)
%
% x = cheby(n, a, b)
%
% Ascisse di Chebyshev:
%
% n : grado del polinomio;
% a,b : estremi dell'intervallo.
%
if nargin < 3, error('numero argomenti insufficienti'), end
if n ~= fix(n) || n <= 0, error('grado non corretto'), end
x = cos((2*(0:n)+1)*pi/(2*(n+1))); %ascisse su [-1, 1]. Passo
%indispensabile. Se il polinomio e' di grado n, allora il numero di
%ascisse e' n+1.
x = (a+b)/2 + x*(b-a)/2;
return

```

9. Data una funzione $C[a, b]$, si definisce polinomio di migliore approssimazione di grado n di f su $[a, b]$:

$$p^* = \arg \min_{p \in \Pi_n} \|f - p\|.$$

È noto che, se $p(x) \in \Pi_n$ è un polinomio interpolante $f(x)$ su $n + 1$ ascisse in $[a, b]$, allora:

1. $\|f - p\| \leq (1 + \Lambda_n) \|f - p\|$, essendo Λ_n la costante di Lebesque definita sulle ascisse di interpolazione;
2. $\|f - p^*\| \leq \alpha \cdot \omega(f; \frac{b-a}{n})$, dove α è indipendente da n e $\omega(f; \frac{b-a}{n})$ è il modulo di continuità di $f(x)$.

Da 1. e 2. si deduce il Teorema di Jackson:

$$\|f - p\| \leq \alpha(1 + \Lambda_n) \omega\left(f; \frac{b-a}{n}\right).$$

10. I polinomi di Chebyshev di prima specie sono definiti dall'equazione di ricorrenza:

$$\begin{cases} T_0(x) \equiv 1; \\ T_1(x) = x; \\ T_{k+1}(x) = 2x \cdot T_k(x) - T_{k-1}(x), \quad k \geq 1 \end{cases} \quad x \in [-1, 1].$$

Sappiamo che:

1. $\|T_k\| = 1, \quad \forall k \geq 0$;

2. Il coefficiente principale di $T_k(x)$ è 2^{k-1} , $\forall k \geq 1$;

$$3. \widehat{T}_k(x) = \begin{cases} T_0(x), & k=0, \\ 2^{1-k}T_k(x), & k \geq 1, \end{cases}$$

è una famiglia di polinomi monici di grado k , $\forall k \geq 0$;

$$4. \forall k \geq 1 : ||\widehat{T}_k|| = 2^{1-k} = \min_{\substack{p \in \Pi_k \\ 510}} ||p||.$$

11. Assegnata la partizione $\Delta = \{x_0 < x_1 < \dots < x_n\}$ diremo che $s_m(x)$ è una spline di grado m su Δ , se:

$$1. \forall i = 1, \dots, n : s_m|_{[x_{i-1}, x_i]}(x) \in \Pi_m;$$

$$2. s_m(x) \in C^{(m-1)}[x_0, x_n].$$

[Specifica aggiuntiva:] La condizione 2. significa richiedere:

$$s_m^{(j)}|_{[x_{i-1}, x_i]}(x_i) = s_m^{(j)}|_{[x_i, x_{i+1}]}(x_i), \\ j = 0, \dots, m-1, i = 1, \dots, n-1.$$

[Aggiunta:] Se data una funzione $f(x)$, si ha $s_m(x_i) = f(x_i)$, $i = 0, \dots, n$, allora $s_m(x)$ è una spline di grado m interpolante $f(x)$ sulla partizione Δ .

13. Per determinare univocamente una spline di grado m sulla partizione $\Delta = \{x_0 < x_1 < \dots < x_n\}$, occorrono $m+n$ condizioni. Le condizioni di interpolazione sono, evidentemente, $n+1$.

Pertanto, esse individuano univocamente la spline lineare interpolante:

$$s_1(x_i) = f_i, \quad i = 0, \dots, n.$$

Trattandosi della spezzata congiungente i punti (x_i, f_i) , $i = 0, \dots, n$, la sua espressione sarà:

$$s_1(x) = \frac{f_i(x - x_{i-1}) + f_{i-1}(x_i - x)}{x_i - x_{i-1}}, \quad x \in [x_{i-1}, x_i], i = 1, \dots, n.$$

16. Si tratta di determinare i coefficienti del polinomio

$$p(x) = \sum_{k=0}^m a_k x^k \in \Pi_m,$$

che meglio approssima le coppie di dati (x_i, f_i) , $i = 1, \dots, n$, $n > m+1$. Questo significa che il valore f_i è approssimato con

$$f_i \approx \sum_{k=0}^m a_k x_i^k, \quad i = 1, \dots, n.$$

⁵¹⁰ p è un polinomio monico.

Scritto in forma vettoriale, abbiamo:

$$\begin{bmatrix} x_1^0 & \dots & x_1^m \\ \vdots & & \vdots \\ x_n^0 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix},$$

ovvero un sistema lineare sovradimesionato. È noto che questo ammette soluzione, mediante la fattorizzazione QR della matrice dei coefficienti, e questa è unica, s.se la matrice ha rango massimo (ovvero $m + 1$). Questo significa che **almeno $m + 1$ delle ascisse $\{x_i\}$ devono essere tra loro distinte.**

17. Preliminarmente, osserviamo che il problema ammette soluzione, e questa è unica, poiché abbiamo 4 ascisse distinte e $4 > m + 1 = 3$. Quindi:

$$\begin{bmatrix} 1^0 & 1^1 & 1^2 \\ 1^0 & 1^1 & 1^2 \\ 2^0 & 2^1 & 2^2 \\ 3^0 & 3^1 & 3^2 \\ 3^0 & 3^1 & 3^2 \\ 4^0 & 4^1 & 4^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2.1 \\ 3 \\ 4 \\ 3 \\ 5 \end{bmatrix}$$

è il problema algebrico che definisce i coefficienti del polinomio di approssimazione ai minimi quadrati:

$$p(x) = a_0 + a_1 x + a_2 x^2.$$

19. Il problema è quello di ottenere una approssimazione dell'integrale definito

$$I(f) = \int_a^b f(x) dx.$$

Questa si ottiene come l'integrale del polinomio, di grado n , interpolante $f(x)$ sulle ascisse, equidistanti:

$$x_i a + i h, \quad i = 0, \dots, n, \quad h = \frac{b - a}{n}.$$

Pertanto, $p(x_i) = f_i$, $i = 0, \dots, n$, e

$$I(p) = \int_a^b p(x) dx \equiv I_n(f),$$

che è la formula di Newton-Cotes di grado n . Otteniamo la sua espressione, utilizzando la formula di Lagrange del polinomio interpolante:

$$p(x) = \sum_{i=0}^n f_i L_{in}(x),$$

con

$$L_{in}(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}, \quad i = 0, \dots, n.$$

Osserviamo che, definendo la trasformazione $x = a + th$, allora:

1. $x \in [a, b] \iff t \in [0, n]$;

2. $x_i = a + ih$, $i = 0, \dots, n$;

3. $dx = hdt$;

$$4. L_{in}(x) = L_{in}(a + th) = \prod_{j=0, j \neq i}^n \frac{(d+th)-(d+ih)}{(d+ih)-(d+jh)} = \prod_{j=0, j \neq i}^n \frac{t-j}{i-j} \equiv \widehat{L}_{in}(t), i = 0, \dots, n.$$

Pertanto:

$$\begin{aligned} I_n(f) &= \int_a^b p(x)dx = \int_a^b \sum_{i=0}^n f_i L_{in}(x)dx \\ \stackrel{x=a+th}{=} h \int_0^h \sum_{i=0}^n f_i \widehat{L}_{in}(t)dt &= h \sum_{i=0}^n f_i \int_0^n \widehat{L}_{in}(t)dt = \underbrace{\frac{b-a}{n} \sum_{i=0}^n f_i \int_0^n \prod_{j=0, j \neq i}^n \frac{t-j}{i-j} dt}_{c_{in}}, \end{aligned}$$

che è la forma finale della formula.

23. È noto che, denotando con $I(f) = \int_a^b f(x)dx$ l'integrale da approssimare, e con $I_k^{(n)}(f)$ l'approssimazione fornita dalla composita di grado k :

$$I(f) - I_k^{(n)}(f) = \nu_k f^{(k+\mu)}(\xi) \frac{(b-a)}{k} \left(\frac{b-a}{n} \right)^{k+\mu},$$

$$\text{con } \mu = \begin{cases} 1, & k \text{ dispari}, \\ 2, & k \text{ pari}, \end{cases}$$

il coefficiente ν_k dipendente solo da k , e $\xi \in [a, b]$, opportuno.

Utilizzando solo le valutazioni funzionali sulle ascisse x_{2i} , $i = 0, \dots, \frac{n}{2}$, con $n = 2km$, $m \in \mathbb{N}$, otteniamo $I_k^{(\frac{n}{2})}(f)$, tali che:

$$I(f) - I_k^{(\frac{n}{2})}(f) = \nu_k f^{n+\mu}(\widehat{\xi}) \frac{b-a}{k} \left(\frac{b-a}{n/2} \right)^{k+\mu},$$

con $\widehat{\xi} \in [a, b]$, opportuno. Utilizzando l'approssimazione $\xi \approx \widehat{\xi}$, sottraendo membro a membro, otteniamo:

$$I_k^{(n)}(f) - I_k^{(\frac{n}{2})}(f) \approx \nu_k f^{(n+\mu)}(\xi) \frac{b-a}{k} \cdot \left(\frac{b-a}{n} \right)^{k+\mu} (2^{k+\mu} - 1) = [I(f) - I_k^{(n)}(f)] (2^{k+\mu} - 1)$$

Pertanto, otteniamo la stima

$$I(f) - I_k^{(n)}(f) \approx \frac{I_k^{(n)}(f) - I_k^{(\frac{n}{2})}(f)}{2^{k+\mu} - 1}.$$

24. Vedere l'Algoritmo 8.2

Algoritmo 8.2 Implementazione esercizio 24.

```
function If = trapez(fun, a, b, n)
%
%   If = trapez(fun, a, b, n)
%
%   Calcolo della formula composita dei trapezi.
%
% Input:
%   fun - identificatore function funzione integranda (deve accettare input
%         vettoriali);
%   a,b - estremi dell'intervallo di integrazione;
%   n   - numero sottointervalli (default=1).
%
% Output:
%   If - stima ottenuta.
%
if nargin < 3
    error('numero argomenti insufficiente')
elseif nargin == 3
    n = 1;
elseif n <= 0 || n~= fix(n)
    error('numero di sottointervalli non valido')
end
x = linspace(a, b, n+1);
h = (b-a)/n;
f = feval(fun, x);
If = h * (sum(f) - (f(1) + f(n+1))/2);
return
end
```

9 Esonero 1

1. Come si definisce la precisione di macchina di un'aritmetica finita? Quando vale la precisione macchina della doppia precisione IEEE?
 2. Cosa è il fenomeno della cancellazione numerica?
 3. Derivare il metodo di Newton per la ricerca della radice di una funzione e dimostrare che esso converge quadraticamente a radici semplici.
 4. Derivare il metodo di accelerazione di Aitken.
 5. Scrivere in modo professionale una *function* Matlab che risolva efficientemente un sistema triangolare superiore.
 6. Sotto quali condizioni esiste la fattorizzazione *LU* di una matrice nonsingolare? Definire cosa si intende per matrice a *diagonale dominante*. Dimostrare che una matrice diagonale dominante è fattorizzabile *LU*.
 7. Definire cosa è il numero di condizionamento di una matrice. Spiegarne il significato.
 8. Definire la soluzione nel senso dei minimi quadrati di un sistema lineare sovra-dimensionato a rango pieno. Dimostrare l'esistenza ed unicità.
1. Se un'aritmetica finita in base b utilizza m cifre per la mantissa di un numero di macchina normalizzato, allora la precisione di macchina u è definita come

$$u = \begin{cases} b^{1-m}, & \text{con troncamento alla } m\text{-esima cifra;} \\ \frac{1}{2}b^{1-m}, & \text{con arrotondamento alla } m\text{-esima cifra.} \end{cases}$$

Essa fornisce una maggiorazione uniforme per l'errore di rappresentazione, per i numeri di macchina normalizzati. La doppia precisione IEEE utilizza:

- base $b = 2$;
- $m = 53$ cifre;
- arrotondamento.

Pertanto, $u = \frac{1}{2}2^{1-53} = 2^{-53} \approx 1 \cdot 10^{-16}$.

2. La cancellazione numerica è dovuta al malcondizionamento della somma algebrica, nel caso i due numeri da sommare siano quasi opposti. In questo caso il numero di condizione della somma $x + y$ vale

$$\kappa = \frac{|x| + |y|}{|x + y|},$$

che non è limitato se $x \approx -y$.

3. Il metodo di Newton è un metodo iterativo per trovare la radice di f , con $f : \mathbb{R} \rightarrow \mathbb{R}$, basata su una linearizzazione locale della funzione $f(x)$ nell'approssimazione corrente x_n : data la retta tangente il grafico di $f(x)$ nel punto $(x_n, f(x_n))$,

$$r : y - f(x_n) = f'(x_n)(x - x_n),$$

x_{n+1} è ricavata come l'ascissa per cui

$$y = 0 : x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, \dots$$

Il metodo di Newton a radici semplici ha convergenza quadratica (Teorema 2.2).

4. Il metodo di accelerazione di Aitken è utilizzato per ripristinare la convergenza quadratica del metodo di Newton verso radici multiple, per cui la convergenza è lineare. Definito $e_n = x_n - \bar{x}$ l'errore al passo n , asintoticamente, $e_n \approx ce_{n-1}$ ed $e_{n+1} \approx ce_n$, con c costante non nota dell'errore. Dividendo membro a membro è ottenuto

$$\frac{e_{n+1}}{e_n} \approx \frac{e_n}{e_{n-1}}.$$

Data l'uguaglianza, $\bar{x}_n \approx \bar{x}$ il valore che la soddisfa, è ottenuta

$$\frac{\bar{x}_n - x_{n+1}}{\bar{x}_n - x_n} = \frac{\bar{x}_n - x_n}{\bar{x}_n - x_{n-1}},$$

dalla quale

$$(\bar{x}_n - x_{n+1})(\bar{x}_n - x_{n-1}) = (\bar{x}_n - x_n)^2.$$

Quindi

$$\bar{x}_n = \frac{x_n^2 - x_{n+1}x_{n-1}}{2x_n - x_{n+1} - x_{n-1}}.$$

Data questa approssimazione, con due passi di Newton, è ottenuta una nuova approssimazione, $\{\bar{x}_n\}$, che converge quadraticamente a \bar{x} . Il metodo di accelerazione di Aitken è preferibile al metodo di Newton modificato quando la molteplicità della radice non è nota.

5. Vedere Algoritmo 9.1.

6. Data una matrice nonsingolare $A \in \mathbb{R}^{n \times n}$, la fattorizzazione LU di A esiste se e solo se, data A_k la sottomatrice principale di ordine k di A , risulta: $\det(A_k) \neq 0$, $k = 1, \dots, n$. La matrice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ è detta diagonale dominante:

- per righe, se $\forall i = 1, \dots, n : |a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$;
- per colonne, se $\forall j = 1, \dots, n : |a_{jj}| > \sum_{i=1, i \neq j}^n |a_{ij}|$.

Una matrice diagonale dominante è fattorizzabile LU . E' sufficiente dimostrare che se A è diagonale dominante allora A è nonsingolare in quanto implicherà che $\forall k = 1, \dots, n : A_k$ è diagonale dominante e, quindi, nonsingolare.

Vale la seguente proprietà: A diagonale dominante per righe/colonne s.se A^T diagonale dominante per colonne/-righe; quindi è sufficiente dimostrare che A è diagonale dominante per righe per dimostrare che A è nonsingolare. Utile il Teorema 3.9.

Algoritmo 9.1 Implementazione esercizio 5.

```
function x = sollu(U, b)
%
%     function x = sollu(U, b)
%
%     Risoluzione del sistema triangolare superiore Ux = b.
%
% Input:
%     U - matrice dei coefficienti (di cui si utilizza la sola porzione
%          triangolare superiore);
%
% Output:
%     x - soluzione.
%
[m,n] = size(U);
if m ~= n || n ~= length(b), error, end
x = b(:);
for i = n : -1 : 1
    if U(i,i) == 0, error, end
    x(i) = x(i)/U(i,i);
    x(1:i-1) = x(i) * U(1:i-1, i);
end
return
```

7. Lo studio del condizionamento di un sistema lineare

$$Ax = b$$

consiste nel verificare in che modo la soluzione è perturbata. Sarà studiato

$$(A + \Delta A)(x + \Delta x) = b + \Delta b,$$

con ΔA e Δb perturbazioni note.

È ottenuto

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \cdot \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right),$$

dove, tramite l'utilizzo di una norma su vettore e la corrispondente indotta su matrice,

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

è il numero di condizione di A .

8. Dato il sistema lineare sovradimensionato

$$Ax = b, \quad A \in \mathbb{R}^{m \times n}, \quad m > n = \text{rank}(A), \quad x \in \mathbb{R}^n, \quad b \in \mathbb{R}^m, \quad (9.1)$$

è definita soluzione di (9.1) nel senso dei minimi quadrati il vettore x che minimizza

$$\|r\|_2^2 = \|Ax - b\|_2^2 = \sum_{i=1}^n r_i^2,$$

dove r_i è la i -esima componente di r .

Per dimostrare l'esistenza ed unicità di x è necessario il Teorema 3.16. Pertanto, rimane valida (3.48) scegliendo x come soluzione di (3.49). Poiché \hat{R} è nonsingolare, questa soluzione esiste ed è unica.

10 Esonero 2

1. Formulare il problema dell'interpolazione polinomiale e dimostrare l'esistenza ed unicità del polinomio di Newton.
2. Scrivere la forma di Lagrange del polinomio interpolante le coppie di dati $(0, 1), (2, 3), (4, 5)$.
3. Determinare la forma di Newton del polinomio interpolante i dati $(-1, 1), (1, 3), (4, -3)$.
4. Costruire il polinomio di Hermite interpolante i dati (della forma (x_i, f_i, f'_i)) $(-1, 3, 1)$ e $(1, 5, 5)$.
5. Definire le ascisse di Chebyshev per il polinomio di grado n su un generico intervallo $[a, b]$ e spiegarne l'importanza nell'ambito dell'interpolazione polinomiale.
6. Definire una funzione di *spline* di grado m su una partizione Δ assegnata. Quante condizioni servono per individuarla univocamente?
7. Scrivere il problema algebrico che definisce il polinomio di approssimazione ai minimi quadrati di grado 3, per le coppie di dati $(0, 2), (2, 0), (2, 3), (3, 2), (3, 3), (4, 5), (5, 4)$, e stabilire se esso esiste ed è unico.
8. Derivare le formule di quadratura di Newton-Cotes.

1. Teorema 4.1 con dimostrazione annessa.

2. $p(x) = 1 \cdot L_{02}(x) + 3 \cdot L_{12}(x) + 5 \cdot L_{22}(x)$, con i polinomi di Lagrange della forma $L_{i2}(x)$, $i = 0, 1, 2$, dati da:

$$L_{02}(x) = \frac{(x-2)(x-4)}{(0-2)(0-4)}, \quad L_{12}(x) = \frac{x(x-4)}{2(2-4)}, \quad L_{22}(x) = \frac{x(x-2)}{4(4-2)}.$$

3. $p(x) = f[-1] + f[-1, 1](x+1) + f[-1, 1, 4](x+1)(x-1)$, dove le differenze divise sono calcolate come segue:

	0	1	2
-1	$f[-1] = 1$		
1	$f[1] = 3$	$f[-1, 1] = 1$	
4	$f[4] = -3$	$f[1, 4] = -2$	$f[-1, 1, 4] = -\frac{3}{5}$

4. $p(x) = f[-1] + f[-1, -1](x+1) + f[-1, -1, 1](x+1)^2 + f[-1, -1, 1](x+1)^2(x-1)$, dove le differenze divise sono calcolate come segue:

	0	1	2	3
-1	$f[-1] = 3$			
-1	$f[-1] = 3$	$f[-1, -1] = 1$		
1	$f[1] = 5$	$f[-1, 1] = -1$	$f[-1, -1, 1] = 0$	
1	$f[1] = 5$	$f[1, 1] = 5$	$f[-1, 1, 1] = 2$	$f[-1, -1, 1, 1] = 1$

5. Le ascisse di Chebyshev richieste sono definite come segue:

$$x_{n-i} = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{2i+1}{2n+2}\right), \quad i = 0, 1, \dots, n.$$

La loro importanza è dovuta al fatto che permettono una crescita quasi ottimale della costante di Lebesgue, $\Lambda_n \approx O(\log n)$, che è il numero di condizionamento del problema.

6. $s_m(x)$ è la spline di grado m su una partizione $\Delta = \{x_0 < x_1 < \dots < x_n\}$ se soddisfa le seguenti proprietà:

1. $s_m|_{[x_{i-1}, x_i]}(x) \in \Pi_m, i = 1, \dots, n;$

2. $s_m(x) \in C^{(m-1)}[x_0, x_n].$

Sono necessarie $m + n$ condizioni indipendenti per individuare una particolare spline.

7. Se il polinomio in questione è $p(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, i suoi coefficienti sono individuati dal sistema lineare sovradeterminato

$$\begin{bmatrix} 0^0 & 0^1 & 0^2 & 0^3 \\ 2^0 & 2^1 & 2^2 & 2^3 \\ 2^0 & 2^1 & 2^2 & 2^3 \\ 3^0 & 3^1 & 3^2 & 3^3 \\ 3^0 & 3^1 & 3^2 & 3^3 \\ 4^0 & 4^1 & 4^2 & 4^3 \\ 5^0 & 5^1 & 5^2 & 5^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 3 \\ 2 \\ 3 \\ 5 \\ 4 \end{bmatrix}$$

nel senso dei minimi quadrati. La soluzione esiste ed è unica se la matrice dei coefficienti ha rango massimo. In questo caso la matrice ha rango massimo perché è del tipo di Vandermonde e vi sono almeno 4 ascisse distinte.

8. È necessario approssimare

$$I(f) = \int_a^b f(x) dx$$

con l'integrale del polinomio interpolante $f(x)$ sulle ascisse equidistanti $x_i = a + ih, i = 0, \dots, n, h = \frac{b-a}{n}$, con il polinomio espresso in forma di Lagrange come

$$p(x) = \sum_{i=0}^n f_i L_{in}(x).$$

È ottenuto quanto segue:

$$I(f) \approx I_n(f) \equiv I(p) = \int_a^b \sum_{i=0}^n f_i L_{in}(x) dx = \sum_{i=0}^n f_i \int_a^b L_{in}(x) dx.$$

Posto $x = a + th$, è ottenuto che $dx = hdt, x_i = a + ih, t \in [0, n]$ e

$$\int_a^b L_{in}(x) dx = h \int_0^n L_{in}(a + th) dt = h \underbrace{\int_0^n \prod_{j=0, j \neq i}^n \frac{t-j}{i-j} dt}_{c_{in}}.$$

Pertanto,

$$I_n(f) = \frac{b-a}{n} \sum_{i=0}^n c_{in} f_i,$$

che è la formula di Newton-Cotes di grado n .

11 Argomenti trattati che sono intermezzi

Osservazione 11.1. ⁵¹¹ Se $f(x)$ è sviluppabile in serie di Taylor in x^* , sua radice di molteplicità esatta m , allora essa è scrivibile nella forma:

$$(x - x^*)^m g(x), \quad (11.1)$$

con $g(x)$ funzione sviluppabile in serie di Taylor in x^* e tale che $g(x) \neq 0$.

⁵¹¹Osservazione 2.1 pg 26.