# Capitolo 1

# L'insieme dei numeri macchina

## 1.1 Introduzione al Calcolo Numerico

Il Calcolo Numerico è una disciplina che fa parte di un ampio settore della Matematica Applicata che prende il nome di Analisi Numerica. Si tratta di una materia che è al confine tra la Matematica e l'Informatica poichè cerca di risolvere i consueti problemi matematici utilizzando però una via algoritmica. In pratica i problemi vengono risolti indicando un processo che, in un numero finito di passi, fornisca una soluzione numerica e soprattutto che sia implementabile su un elaboratore. I problemi matematici che saranno affrontati nelle pagine seguenti sono problemi di base: risoluzione di sistemi lineari, approssimazione delle radici di funzioni non lineari, approssimazione di funzioni e dati sperimentali, calcolo di integrali definiti. Tali algoritmi di base molto spesso non sono altro se non un piccolo ingranaggio nella risoluzione di problemi ben più complessi.

## 1.2 Rappresentazione in base di un numero reale

Dovendo considerare problemi in cui l'elaboratore effettua computazioni esclusivamente su dati di tipo numerico risulta decisivo iniziare la trattazione degli argomenti partendo dalla rappresentazione di numeri. Innanzitutto è opportuno precisare che esistono due modi per rappresentare i numeri: la cosiddetta notazione posizionale, in cui il valore di una cifra dipende dalla posizione

in cui si trova all'interno del numero, da quella notazione non posizionale, in cui ogni numero è rappresentato da uno, o da un insieme di simboli (si pensi come esempio alla numerazione usata dai Romani). La motivazione che spinge a considerare come primo problema quello della rappresentazione di numeri reali è che ovviamente si deve sapere il livello di affidabilità dei risultati forniti dall'elaboratore. Infatti bisogna osservare che i numeri reali sono infiniti mentre la memoria di un calcolatore ha una capacità finita che ne rende impossibile la rappresentazione esatta. Una seconda osservazione consiste nel fatto che un numero reale ammette molteplici modi di rappresentazione. Per esempio scrivere

$$x = 123.47$$

è la rappresentazione, in forma convenzionale, dell'espressione

$$x = 123.47 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times \times 10^{-1} + 7 \times 10^{-2},$$

da cui, mettendo in evidenza $10^2$:

$$x = 10^2 \times \left(1 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2} + 4 \times \times 10^{-3} + 7 \times 10^{-4}\right)$$

mentre, mettendo in evidenza $10^3$ lo stesso numero viene scritto come

$$x = 10^3 \times \left(1 \times 10^{-1} + 2 \times 10^{-2} + 3 \times 10^{-3} + 4 \times \times 10^{-4} + 7 \times 10^{-5}\right)$$

deducendo che ogni numero, senza una necessaria rappresentazione convenzionale, può essere scritto in infiniti modi. Il seguente teorema è fondamentale proprio per definire la rappresentazione dei numeri reali in una determinata base $\beta$.

**Teorema 1.2.1** *Sia $\beta \in \mathbb{N}$, $\beta \geq 2$, allora ogni numero reale $x$, $x \neq 0$, può essere rappresentato univocamente in base $\beta$ nel seguente modo*

$$x = \pm\ \beta^p \sum_{i=1}^{\infty} d_i \beta^{-i}$$

*dove $p \in \mathbb{Z}$, e i valori $d_i \in \mathbb{N}$ (detti cifre), verificano le seguenti proprietà:*
*1. $d_i \in \{0, 1, 2, 3, \ldots, \beta - 1\}$;*
*2. $d_1 \neq 0$;*
*3. le cifre $d_i$ non sono definivamente uguali a $\beta - 1$.*

Evitiamo la dimostrazione del Teorema 1.2.1 ma osserviamo che la la terza ipotesi è essenziale per l'unicità della rappresentazione. Consideriamo infatti il seguente esempio (in base $\beta = 10$).

$$\begin{aligned} x &= 0.999999999\ldots \\ &= 9 \times 10^{-1} + 9 \times 10^{-2} + 9 \times 10^{-3} + \ldots \\ &= \sum_{i=1}^{\infty} 9 \cdot 10^{-i} = 9 \sum_{i=1}^{\infty} \left(\frac{1}{10}\right)^i \\ &= 9 \left(\frac{1}{10}\right) \left(1 - \frac{1}{10}\right)^{-1} \\ &= 9 \left(\frac{1}{10}\right) \left(\frac{10}{9}\right) = 1. \end{aligned}$$

L'ultima uguaglianza deriva dalla convergenza della serie geometrica

$$\sum_{i=0}^{\infty} q^i = \frac{1}{1-q}$$

quando $0 < q < 1$, da cui segue

$$1 + \sum_{i=1}^{\infty} q^i = \frac{1}{1-q}$$

e

$$\sum_{i=1}^{\infty} q^i = \frac{1}{1-q} - 1 = \frac{q}{1-q}.$$

In conclusione, senza la terza ipotesi del Teorema 1.2.1, al numero 1 corrisponderebbero due differenti rappresentazioni in base.
Considerato un numero reale $x \in \mathbb{R}$, $x \neq 0$, l'espressione

$$x = \pm\, \beta^p \times 0.d_1 d_2 \ldots d_k \ldots$$

prende il nome di rappresentazione in base $\beta$ di $x$. Il numero $p$ viene detto esponente (o caratteristica), i valori $d_i$ sono le cifre della rappresentazione, mentre il numero decimale $0.d_1 d_2 \ldots d_k \ldots$ si dice mantissa. Il numero

$x$ viene normalmente rappresentato con la cosiddetta notazione posizionale $x = segno(x)(.d_1 d_2 d_3 \dots) \times \beta^p$, che viene detta normalizzata. In alcuni casi è ammessa una rappresentazione in notazione posizionale tale che $d_1 = 0$, che viene detta denormalizzata. La basi più utilizzate sono $\beta = 10$ (sistema decimale), $\beta = 2$ (sistema binario, che, per la sua semplicità, è quello utilizzato dagli elaboratori elettronici), e $\beta = 16$ (sistema esadecimale) e comunque la base è sempre un numero pari. Nel sistema esadecimale le cifre appartengono all'insieme

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\} .$$

Bisogna ternere presente che un qualunque numero reale $x \neq 0$ può essere rappresentato con infinite cifre nella mantissa e inoltre l'insieme dei numeri reali ha cardinalità infinita. Poichè un elaboratore è dotato di memoria finita non è possibile memorizzare:
a) gli infiniti numeri reali
b) le infinite (in generale) cifre di un numero reale.

## 1.3 L'insieme dei numeri macchina

Assegnati i numeri $\beta, t, m, M \in \mathbb{N}$ si definisce insieme dei numeri di macchina con rappresentazione normalizzata in base $\beta$ con $t$ cifre significative

$$\mathbb{F}(\beta, t, m, M) = \left\{ x \in \mathbb{R} : \ x = \pm\beta^p \sum_{i=1}^{t} d_i \beta^{-i} \right\} \cup \{0\}$$

dove
1. $t \geq 1, \ \beta \geq 2, \ m, M > 0$;
2. $d_i \in \{0, 1 \dots, \beta - 1\}$;
3. $d_1 \neq 0$;
4. $p \in \mathbb{Z}$, $-m \leq p \leq M$.

È stato necessario aggiungere il numero zero all'insieme in quanto non ammette rappresentazione in base normalizzata.
Osserviamo che un elaboratore la cui memoria abbia le seguenti caratteristiche (riportate anche in Figura 1.1):

- $t$ campi di memoria per la mantissa, ciascuno dei quali può assumere $\beta$ differenti configurazioni (e perciò può memorizzare una cifra $d_i$),

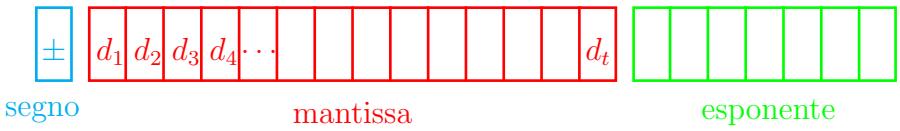



Figura 1.1: Locazione di memoria.

- un campo di memoria che può assumere $m + M + 1$ differenti configurazioni (e perciò può memorizzare i differenti valori $p$ dell'esponente),
- un campo che può assumere due differenti configurazioni (e perciò può memorizzare il segno + o −),

è in grado di rappresentare tutti gli elementi dell'insieme $\mathbb{F}(\beta, t, m, M)$. In realtà poichè se $\beta = 2$ $d_1 = 1$, allora determinati standard non memorizzano la prima cifra della mantissa. Il più piccolo numero positivo appartenente all'insieme $\mathbb{F}(\beta, t, m, M)$ si ottiene prendendo la più piccola mantissa (ovvero 0.1) ed il più piccolo esponente

$$x = 0.1 \times \beta^{-m}$$

mentre il più grande ha tutte le cifre della mantissa uguali alla cifra più grande (ovvero $\beta - 1$) ed il massimo esponente

$$x = 0.\underbrace{dd\ldots dd}_{t}\ \beta^M, \qquad d = \beta - 1.$$

Un'ultima osservazione riguarda il fatto che non è necessario rappresentare il segno dell'esponente poichè questo viene memorizzato utilizzando un'opportuna traslazione, detta offset, che lo rende sempre posiitvo. Consideriamo ora come esempio l'insieme $\mathbb{F}(2, 2, 2, 2)$, cioè i numeri binari con mantissa di due cifre ed esponente compreso tra -2 e 2. Enumeriamo gli elementi di questo insieme. Poichè il numero zero non appartiene all'insieme dei numeri macchina viene rappresentato solitamente con mantissa nulla ed esponente

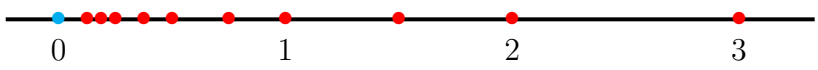



Figura 1.2: Elementi dell'insieme $\mathbb{F}(2,2,2,2)$.

$-m$.

$$
\begin{aligned}
p=-2 \quad & x = 0.10 \times 2^{-2} = 2^{-1} \times 2^{-2} = 2^{-3} = 0.125; \\
& x = 0.11 \times 2^{-2} = (2^{-1} + 2^{-2}) \times 2^{-2} = 3/16 = 0.1875; \\
p=-1 \quad & x = 0.10 \times 2^{-1} = 2^{-1} \times 2^{-1} = 2^{-2} = 0.25; \\
& x = 0.11 \times 2^{-1} = (2^{-1} + 2^{-2}) \times 2^{-1} = 3/8 = 0.375; \\
p=0 \quad & x = 0.10 \times 2^{0} = 2^{-1} \times 2^{0} = 2^{-1} = 0.5; \\
& x = 0.11 \times 2^{0} = (2^{-1} + 2^{-2}) \times 2^{0} = 3/4 = 0.75; \\
p=1 \quad & x = 0.10 \times 2^{1} = 2^{-1} \times 2^{1} = 1; \\
& x = 0.11 \times 2^{1} = (2^{-1} + 2^{-2}) \times 2^{1} = 3/2 = 1.5; \\
p=2 \quad & x = 0.10 \times 2^{2} = 2^{-1} \times 2^{2} = 2; \\
& x = 0.11 \times 2^{2} = (2^{-1} + 2^{-2}) \times 2^{2} = 3;
\end{aligned}
$$

Nella Figura 1.2 è rappresentato l'insieme dei numeri macchina positivi appartenenti a $\mathbb{F}(2,2,2,2)$ (i numeri negativi sono esattamente simmetrici rispetto allo zero). Dalla rappresentazione dell'insieme dei numeri macchina si evincono le seguenti considerazioni:

1. L'insieme è discreto;
2. I numeri rappresentabili sono solo una piccola parte dell'insieme $\mathbb{R}$;
3. La distanza tra due numeri macchina consecutivi è $\beta^{p-t}$, infatti, considerando per semplicità numeri positivi, sia

$$x = +\beta^{p} \times (0.d_1 d_2 \dots d_{t-1} d_t)$$

il successivo numero macchina è

$$y = +\beta^p \times (0.d_1 d_2 \dots d_{t-1}\tilde{d}_t)$$

dove

$$\tilde{d}_t = d_t + 1.$$

La differenza è pertanto

$$y - x = +\beta^p(0.\underbrace{00\dots00}_{t-1}1) = \beta^{p-t}.$$

Nello standard IEEE (Institute of Electric and Electronic Engineers) singola precisione una voce di memoria ha 32 bit, dei quali 1 riservato al segno, 8 all'esponente e 23 alla mantissa. Allora $\beta = 2$, $t = 23, m = 127$ e $M = 128$. In questo caso il valore dell'offset è 127 quindi per esempio l'esponente $-30$ viene rappresentato come il numero 93 $(= -30 + 127)$. Nella realtà spesso non tutte le rappresentazioni dell'esponente sono ammesse (per esempio gli esponenti 0 e 255 sono riservati ad alcune situazioni particolari, ma su questo non è opportuno soffermarsi ulteriormente).
Per la doppia precisione si utilizzano 64 bit, di cui 1 per il segno, 11 per l'esponente e 52 per la mantissa. Dunque $\beta = 2$, $t = 52, m = 1023$ e $M = 1024$. Dopo aver compreso la struttura dell'insieme $\mathbb{F}(\beta, t, m, M)$ resta da capire come, assegnato un numero reale $x$ sia possibile rappresentarlo nell'insieme dei numeri macchina, ovvero quale elemento $\tilde{x} \in \mathbb{F}(\beta, t, m, M)$ possa essergli associato in modo da commettere il più piccolo errore di rappresentazione possibile. Supponiamo ora che la base $\beta$ sia un numero pari. Possono presentarsi diversi casi:

- Sia
$$x = \pm\ \beta^p\ \sum_{i=1}^{n} d_i \beta^{-i}$$
con $d_1 \neq 0$, $n \leq t$, e $-m \leq p \leq M$. Allora è evidente che $x \in \mathbb{F}(\beta, t, m, M)$ e pertanto verrà rappresentato esattamente su un qualunque elaboratore che utilizzi $\mathbb{F}(\beta, t, m, M)$ come insieme dei numeri di macchina.

- Sia
$$x = \pm\ \beta^p\ \sum_{i=1}^{n} d_i \beta^{-i}$$

con $n \leq t$ ma supponiamo che $p \notin [-m, M]$. Se $p < -m$ allora $x$ è più piccolo del più piccolo numero di macchina: in questo caso si dice che si è verificato un underflow (l'elaboratore interrompe la sequenza di calcoli e segnala con un messaggio l'underflow). Se $p > M$ allora vuol dire che $x$ è più grande del più grande numero di macchina e in questo caso si dice che si è verificato un overflow (anche in questo caso l'elaboratore si ferma e segnala l'overflow, anche se tale eccezione può anche essere gestita via software in modo tale che l'elaborazione continui).

- Sia
$$x = \pm\, \beta^p \sum_{i=1}^{n} d_i \beta^{-i}$$
con l'esponente $-m \leq p \leq M$ ma $n > t$ ed inoltre esiste un indice $k$, $t < k \leq n$, tale che $d_k \neq 0$. In questo caso, poichè la mantissa di $x$ ha più di $t$ cifre decimali, $x \notin \mathbb{F}(\beta, t, m, M)$. È però possibile rappresentare $x$ mediante un numero in $\mathbb{F}(\beta, t, m, M)$ con un'opportuna operazione di taglio delle cifre decimali che seguono la $t$-esima. Per questo si possono utilizzare due diverse tecniche di approssimazione:

  1. troncamento di $x$ alla $t$-esima cifra significativa
  $$\tilde{x} = \mathrm{tr}(x) = \beta^p \times 0.d_1 d_2 \ldots d_t$$
  2. arrotondamento di $x$ alla $t$-esima cifra significativa
  $$\tilde{x} = \mathrm{arr}(x) = \beta^p \times 0.d_1 d_2 \ldots \tilde{d}_t$$

dove
$$\tilde{d}_t = \begin{cases} d_t + 1 & \text{se } d_{t+1} \geq \beta/2 \\ d_t & \text{se } d_{t+1} < \beta/2. \end{cases}$$
Per esempio se $\beta = 10$, $t = 5$ e $x = 0.654669235$ allora
$$tr(x) = 0.65466, \qquad arr(x) = 0.65467$$
In pratica quando il numero reale $x$ non appartiene all'insieme $\mathbb{F}(\beta, t, m, M)$ esistono sicuramente due numeri $a, b \in \mathbb{F}(\beta, t, m, M)$, tali che
$$a < x < b. \tag{1.1}$$

Supponendo per semplicità $x > 0$ si ha che

$$tr(x) = a$$

mentre se $x \geq (a+b)/2$ allora

$$arr(x) = b$$

altrimenti

$$arr(x) = a.$$

L'arrotondamento è un'operazione che fornisce sicuramente un risultato più preciso (come risulterà evidente nel prossimo paragrafo), ma può dar luogo ad overflow. Infatti se

$$x = 0.\underbrace{ddddd\ldots ddd}_{t+1}\cdots \times \beta^M$$

con $d = \beta - 1$, allora

$$arr(x) = 1.0\beta^M = 0.1\beta^{M+1} \notin \mathbb{F}(\beta, t, m, M).$$

La rappresentazione di $x \in \mathbb{R}$ attraverso $\tilde{x} \in \mathbb{F}(\beta, t, m, M)$ si dice rappresentazione in virgola mobile di $x$ o rappresentazione floating point, con troncamento se $\tilde{x} = \mathrm{tr}(x)$, con arrotondamento se $\tilde{x} = \mathrm{arr}(x)$. Talvolta il numero macchina che rappresenta $x \in \mathbb{R}$ viene indicato con $fl(x)$.

## 1.4 Errore Assoluto ed Errore Relativo

Una volta definite le modalità per associare ad un numero reale $x$ la sua rappresentazione macchina $\tilde{x}$ si tratta di stabilire l'errore che si commette in questa operazione di approssimazione. Si possono definire due tipi di errori, l'errore assoluto e l'errore relativo.
Se $x \in \mathbb{R}$ ed $\tilde{x}$ è una sua approssmazione allora si definisce errore assoluto la quantità

$$E_a = |\tilde{x} - x|$$

mentre se $x \neq 0$ si definisce errore relativo la quantità

$$E_r = \frac{|\tilde{x} - x|}{|x|}.$$

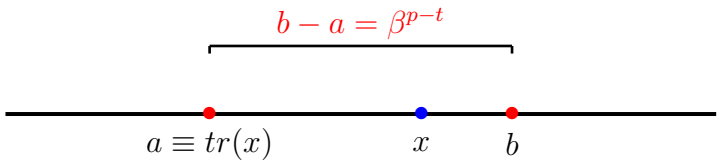



Figura 1.3: Stima dell'errore di rappresentazione nel caso di troncamento.

Se $E_r \leq \beta^{-q}$ allora si dice che $\tilde{x}$ ha almeno $q$ cifre significative corrette. Nel seguito assumeremo $x > 0$ e supporremo anche che la rappresentazione di $x$ in $\mathbb{F}(\beta, t, m, M)$ non dia luogo ad underflow o overflow. Calcoliamo ora una maggiorazione per tali errori nel caso in cui $\tilde{x}$ sia il troncamento di $x > 0$. Nella Figura 1.3 $a$ e $b$ rappresentano i due numeri macchina tali che sia vera la relazione (1.1). È evidente ch risulta

$$|tr(x) - x| < b - a = \beta^{p-t}.$$

Per maggiorare l'errore relativo osserviamo che

$$|x| = +\beta^p \times 0.d_1 d_2 d_3 \cdots \geq \beta^p \times 0.1 = \beta^{p-1}.$$

da cui

$$\frac{1}{|x|} \leq \beta^{1-p}$$

e quindi

$$\frac{|tr(x) - x|}{|x|} \leq \beta^{p-t} \times \beta^{1-p} = \beta^{1-t}. \tag{1.2}$$

Passiamo ora alla valutazione degli errori quando

$$\tilde{x} = arr(x).$$

Nella Figura 1.4 $a$ e $b$ rappresentano i due numeri macchina tali che sia vera la relazione (1.1). Se $x > 0$ si trova a sinistra del punto medio $(a + b)/2$ allora l'arrotondamento coincide con il valore $a$, se si trova nel punto medio oppure alla sua destra allora coincide con $b$. È evidente che il massimo errore si ottiene quando $x$ coincide con il punto medio tra $a$ e $b$ risulta

$$|arr(x) - x| \leq \frac{1}{2}(b - a) = \frac{1}{2}\beta^{p-t}.$$

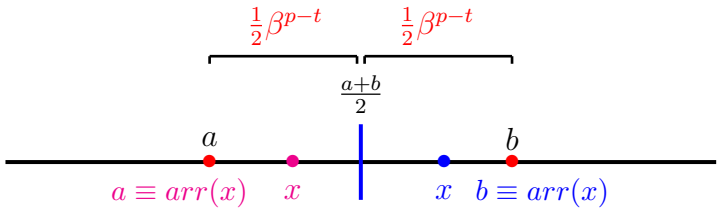



Figura 1.4: Stima dell'errore di rappresentazione nel caso di arrotondamento.

Per maggiorare l'errore relativo procediamo come nel caso del troncamento di $x$:

$$\frac{|arr(x) - x|}{|x|} \leq \frac{1}{2}\beta^{p-t} \times \beta^{1-p} = \frac{1}{2}\beta^{1-t}. \tag{1.3}$$

Le quantità che compaiono a destra delle maggiorazioni (1.2) e (1.3), ovvero

$$u = \beta^{1-t}$$

oppure

$$u = \frac{1}{2}\beta^{1-t}$$

sono dette precisione di macchina o zero macchina per il troncamento (o per l'arrotondamento, in base alla tecnica in uso).
Posto

$$\varepsilon_x = \frac{\tilde{x} - x}{x}, \qquad |\varepsilon_x| \leq u$$

risulta

$$\tilde{x} = x(1 + \varepsilon_x) \tag{1.4}$$

che fornisce la relazione tra un numero $x \in \mathbb{R}$ e la sua rappresentazione macchina.

### 1.4.1 Operazioni Macchina

Se $x, y \in \mathbb{F}(\beta, t, m, M)$ non è detto che il risultato di un'operazione aritmetica tra $x$ e $y$ non è detto che sia un numero macchina. Per esempio se $x, y \in \mathbb{F}(10, 2, m, M)$ e $x = 0.11\ 10^0$ e $y = 0.11\ 10^{-2}$, allora

$$x + y = 0.1111 \notin \mathbb{F}(10, 2, m, M).$$

Si pone il problema di definire le operazioni aritmetiche in modo tale che ciò non accada. Se $\cdot$ è una delle quattro operazioni aritmetiche di base allora il risultato è un numero macchina se

$$x \cdot y = fl(x \cdot y). \tag{1.5}$$

L'operazione definita dalla relazione (1.5) è detta operazione macchina. L'operazione macchina associata a $\cdot$ viene indicata con $\odot$ e deve soddisfare anch'essa la relazione (1.4), ovvero dev'essere:

$$x \odot y = (x \cdot y)(1 + \varepsilon), \qquad |\varepsilon| < u \tag{1.6}$$

per ogni $x, y \in \mathbb{F}(\beta, t, m, M)$ tali che $x \odot y$ non dia luogo ad overflow o underflow. Si può dimostrare che

$$x \odot y = \mathrm{tr}(x \cdot y)$$

e

$$x \odot y = \mathrm{arr}(x \cdot y)$$

soddisfano la (1.6) e dunque danno luogo ad operazioni di macchina. Le quattro operazioni così definite danno luogo alla aritmetica di macchina o aritmetica finita. La somma algebrica macchina (addizione e sottrazione) tra due numeri $x, y \in \mathbb{F}(\beta, t, m, M)$ richiede le seguenti fasi:

1. Si scala la mantissa del numero con l'esponente minore in modo tale che i due addendi abbiano lo stesso esponente (ovvero quello dell'esponente maggiore);

2. Si esegue la somma tra le mantisse;

3. Si normalizza il risultato aggiustando l'esponente in modo tale che la mantissa sia un numero minore di 1.

4. Si arrotonda (o si tronca) la mantissa alle prime $t$ cifre;

Consideriamo per esempio i numeri $x, y \in \mathbb{F}(10, 5, m, M)$

$$x = 0.78546 \times 10^2, \qquad y = 0.61332 \times 10^{-1}$$

e calcoliamo il numero macchina $x \oplus y$.
1. Scaliamo il numero $y$ fino ad ottenere esponente 2 (quindi si deve spostare

il punto decimale di 3 posizioni), $y = 0.00061332 \times 10^2$;
2. Sommiamo le mantisse $0.78546 + 0.00061332 = 0.78607332$;
3. Questa fase non è necessaria perchè la mantissa è già minore di 1;
4. Si arrotonda alla quinta cifra decimale ottenendo

$$x \oplus y = 0.78607 \times 10^2.$$

Un fenomeno particolare, detto cancellazione di cifre significative, si verifica quando si effettua la sottrazione tra due numeri reali all'incirca uguali. Consideriamo per esempio la differenza tra i due numeri

$$x = 0.75868531 \times 10^2, \qquad y = 0.75868100 \times 10^2$$

nell'insieme $\mathbb{F}(10, 5, m, M)$. Risulta

$$fl(x) = 0.75869 \times 10^2, \qquad fl(y) = 0.75868 \times 10^2$$

e quindi

$$fl(fl(x) - fl(y)) = 0.1 \times 10^{-2}$$

mentre

$$x - y = 0.431 \times 10^{-3}$$

Calcolando l'errore relativo sul risultato dell'operazione si trova

$$E_r \simeq 1.32019$$

che è un valore piuttosto alto.
Per esemplificare il fenomeno appena descritto consideriamo il problema di calcolare (per esempio in MatLab) le radici dell'equazione di secondo grado

$$p(x) = ax^2 + bx + c$$

applicando la consueta formula

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \qquad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}. \tag{1.7}$$

In alternativa si potrebbe calcolare la radice più grande in modulo

$$r_1 = \frac{-b - segno(b)\sqrt{b^2 - 4ac}}{2a} \tag{1.8}$$

e poi, sfruttando la proprietà che il prodotto tra le radici è pari a $c/a$, ottenere la seconda radice ponendo

$$r_2 = \frac{c}{ar_1}. \tag{1.9}$$

Considerando il polinomio

$$p(x) = x^2 - (10^7 + 10^{-7})x + 1$$

che ammette come radici $10^7$ e $10^{-7}$, applicando le formule (1.7), si ottiene

$$x_1 = 10^7, \qquad x_2 = 9.9652e - 008$$

mentre utilizzando le formule (1.8) e (1.9) i risultati sono esatti

$$r_1 = 10^7, \qquad r_2 = 10^{-7}.$$

Nel primo caso il calcolo della radice $x_1$ avviene effettuando la differenza tra due numeri (ovvero $-b$ e $\sqrt{b^2 - 4ac}$) che sono molto vicini tra loro e pertanto generano il suddetto fenomeno. Nel secondo caso non viene effettuata alcuna differenza e pertanto il risultato è corretto.
Il prodotto macchina tra due numeri $x, y \in \mathbb{F}(\beta, t, m, M)$ richiede le seguenti fasi:

1. Si esegue il prodotto tra le mantisse;
2. Si esegue l'arrotondamento (o il troncamento) alle prime $t$ cifre normalizzando, se necessario, la mantissa;
3. Si sommano gli esponenti.

Consideriamo per esempio il prodotto tra i due numeri

$$x = 0.11111 \times 10^3, \qquad y = 0.52521 \times 10^2$$

nell'insieme $\mathbb{F}(10, 5, m, M)$.
1. Il prodotto delle mantisse produce 0.05835608;
2. L'arrotondamento a 5 cifre produce $0.58356 \times 10^{-1}$;
3. La somma degli esponenti fornisce come risultato

$$x * y = 0.58356 \times 10^{3+2-1} = 0.58356 \times 10^4.$$

La divisione macchina tra due numeri $x, y \in \mathbb{F}(\beta, t, m, M)$ richiede le seguenti fasi:

1. Si scala il dividendo $x$ finchè la sua mantissa non risulti minore di quella del divisore $y$;
2. Si esegue la divisione tra le mantisse;
3. Si esegue l'arrotondamento (o il troncamento) alle prime $t$ cifre;
4. Si sottraggono gli esponenti.

Consideriamo la divisione tra i due numeri

$$x = 0.12100 \times 10^5, \qquad y = 0.11000 \times 10^2$$

nell'insieme $\mathbb{F}(10, 5, m, M)$.
1. Scaliamo il dividendo di una cifra decimale 0.012100; l'esponente diventa 6;
2. Dividiamo le mantisse $0.012100/0.11000 = 0.11000$;
3. Il troncamento fornisce lo stesso numero 0.11000;
4. Si sottraggono gli esponenti ottenendo il risultato

$$x \oslash y = 0.11000 \times 10^4.$$

Si può dimostrare che valgono le seguenti proprietà:

1. L'insieme $\mathbb{F}(\beta, t, m, M)$ non è chiuso rispetto alle operazioni macchina;
2. L'elemento neutro per la somma non è unico: infatti consideriamo i due numeri macchina

   $$x = 0.15678 \times 10^3, \qquad y = 0.25441 \times 10^{-2},$$

   appartenenti all'insieme $\mathbb{F}(10, 5, m, M)$, innanzitutto si scala $y$

   $$y = 0.0000025441 \times 10^3,$$

   sommando le mantisse si ottiene 0.1567825441 mentre l'arrotondamento fornisce il risultato finale

   $$x \oplus y = 0.15678 \times 10^3 = x.$$

3. L'elemento neutro per il prodotto non è unico;
4. Non vale la proprietà associativa di somma e prodotto;
5. Non vale la proprietà distributiva della somma rispetto al prodotto.

# Capitolo 2

# Equazioni non Lineari

## 2.1 Introduzione

Le radici di un'equazione non lineare $f(x) = 0$ non possono, in generale, essere espresse esplicitamente e anche se ciò è possibile spesso l'espressione si presenta in forma talmente complicata da essere praticamente inutilizzabile. Di conseguenza per poter risolvere equazioni di questo tipo siamo obbligati ad utilizzare metodi numerici che sono, in generale, di tipo iterativo, cioè partendo da una (o in alcuni casi più) approssimazioni della radice, producono una successione $x_0$, $x_1$, $x_2, \ldots$, convergente alla radice. Per alcuni di questi metodi per ottenere la convergenza è sufficiente la conoscenza di un intervallo $[a, b]$ che contiene la soluzione, altri metodi richiedono invece la conoscenza di una buona approssimazione iniziale. Talvolta è opportuno utilizzare in maniera combinata due metodi, uno del primo tipo e uno del secondo.
Prima di analizzare alcuni metodi per l'approssimazione delle radici dell'equazione $f(x) = 0$ diamo la definizione di molteplicità di una radice.

**Definizione 2.1.1** *Sia $f \in \mathcal{C}^r([a, b])$ per un intero $r > 0$. Una radice $\alpha$ di $f(x)$ si dice di molteplicità $r$ se*

$$\lim_{x \to \alpha} \frac{f(x)}{(x - \alpha)^r} = \gamma, \qquad \gamma \neq 0, \gamma \neq \pm\infty. \tag{2.1}$$

Se $\alpha$ è una radice della funzione $f(x)$ di molteplicità $r$ allora risulta

$$f(\alpha) = f'(\alpha) = \cdots = f^{(r-1)}(\alpha) = 0, \qquad f^{(r)}(\alpha) = \gamma \neq 0.$$

## 2.2 Localizzazione delle radici

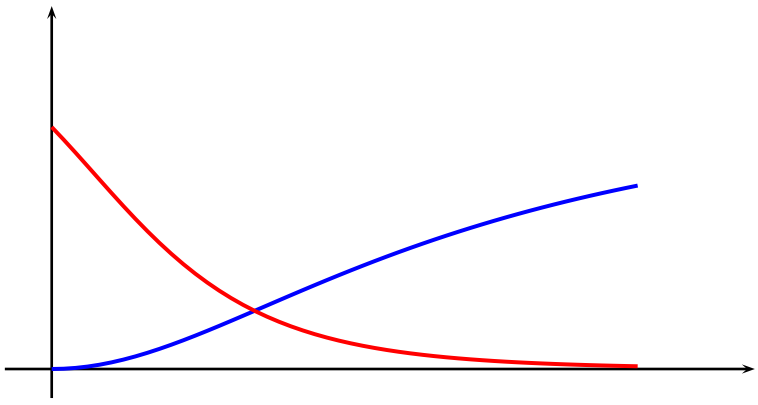
Nei successivi paragrafi saranno descritti alcuni metodi numerici per il calcolo approssimato delle radici di un'equazione non lineare. Tali metodi numerici sono di tipo iterativo, ovvero consistono nel definire una successione (o più successioni), che, a partire da un'assegnata approssimazione iniziale (nota), converga alla radice $\alpha$ in un processo al limite. Infatti poichè non esistono tecniche generali che consentano di trovare l'espressione esplicita di $\alpha$ in un numero finito di operazioni, allora questa può essere calcolata in modo approssimato solo in modo iterativo. Questa perculiarità tuttavia richiede che sia nota appunto un'approssimazione iniziale o, almeno, un intervallo di appartenenza. Il problema preliminare è quello di localizzare la radice di una funzione, problema che viene affrontato in modo grafico. Per esempio considerando la funzione

$$f(x) = \sin\big(\log(x^2+1)\big) - \frac{e^{-x}}{x^2+1}$$

risulta immediato verificare che il valore dell'ascissa in cui si annulla è quello in cui si intersecano i grafici delle funzioni

$$g(x) = \sin\big(\log(x^2+1)\big) \qquad\qquad h(x) = \frac{e^{-x}}{x^2+1}.$$

Un modo semplice per stimare tale valore è quello di tracciare i grafici delle due funzioni, come riportato nella seguente figura in cui il grafico di $h(x)$ è in rosso, mentre quello di $g(x)$ è blu, e l'intervallo di variabilità di $x$ è $[0, 2.5]$.



Calcolando le funzioni in valori compresi in tale intervallo di variabilità si può restringere lo stesso intervallo, infatti risulta

$$g(0.5) = 0.2213 < h(0.5) = 0.48522$$

e

$$g(1) = 0.63896 > h(1) = 0.18394,$$

da cui si deduce che $\alpha \in ]0.5, 1[$.

## 2.3 Il Metodo di Bisezione

Sia $f : [a, b] \to \mathbb{R}$, $f \in \mathcal{C}([a, b])$, e sia $f(a)f(b) < 0$. Sotto tali ipotesi esiste sicuramente almeno un punto nell'intervallo $[a, b]$ in cui la funzione si annulla. L'idea alla base del Metodo di Bisezione (o metodo delle bisezioni) consiste nel costruire una successione di intervalli $\{I_k\}_{k=0}^{\infty}$, con $I_0 = [a_0, b_0] \equiv [a, b]$, tali che:

1. $I_{k+1} \subset I_k$;
2. $\alpha \in I_k$, $\forall k \geq 0$;
3. l'ampiezza di $I_k$ tende a zero per $k \to +\infty$.

La successione degli $I_k$ viene costruita nel seguente modo. Innanzitutto si pone

$$I_0 = [a_0, b_0] = [a, b]$$

e si calcola il punto medio

$$c_1 = \frac{a_0 + b_0}{2}.$$

Se $f(c_1) = 0$ allora $\alpha = c_1$, altrimenti si pone:

$$I_1 = [a_1, b_1] \equiv \begin{cases} a_1 = a_0 \quad b_1 = c_1 & \text{se } f(a_0)f(c_1) < 0 \\ a_1 = c_1 \quad b_1 = b_0 & \text{se } f(a_0)f(c_1) > 0. \end{cases}$$

Ora, a partire da $I_1 = [a_1, b_1]$, si ripete la stessa procedura. In generale al passo $k$ si calcola

$$c_{k+1} = \frac{a_k + b_k}{2}.$$

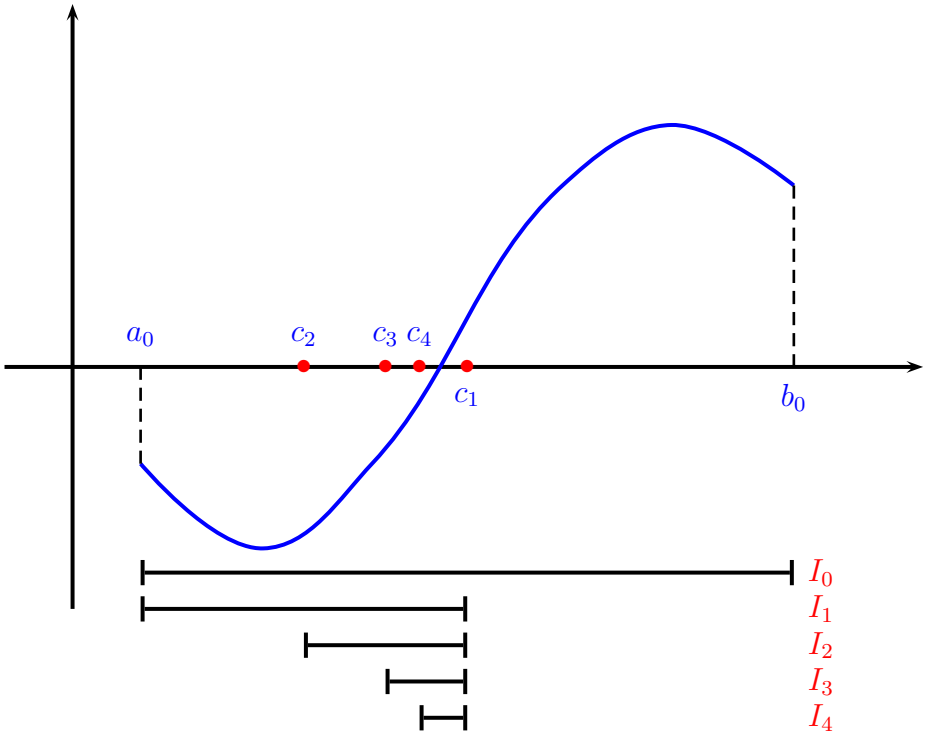
Se $f(c_{k+1}) = 0$ allora $\alpha = c_{k+1}$, altrimenti si pone:

$$I_{k+1} = [a_{k+1}, b_{k+1}] \equiv \begin{cases} a_{k+1} = a_k \quad b_{k+1} = c_{k+1} & \text{se } f(a_k)f(c_{k+1}) < 0 \\ a_{k+1} = c_{k+1} \quad b_{k+1} = b_k & \text{se } f(a_k)f(c_{k+1}) > 0. \end{cases}$$

La successione di intervalli $I_k$ così costruita soddisfa automaticamente le condizioni 1) e 2). Per quanto riguarda la 3) abbiamo:

$$b_k - a_k = \frac{b_{k-1} - a_{k-1}}{2} = \frac{b_0 - a_0}{2^k}$$

e dunque l'ampiezza di $I_k$ tende a zero quando $k \to +\infty$.




Generalmente costruendo le successioni $\{a_k\}$ e $\{b_k\}$ accade che la condizione $f(c_k) = 0$, per un certo valore $k$, non si verifica mai a causa degli errori di arrotondamento. Quindi è necessario stabilire un opportuno criterio di stop che ci permetta di fermare la procedura quando riteniamo di aver raggiunto una precisione soddisfacente. Per esempio si può imporre:

$$b_k - a_k \leq \varepsilon \tag{2.2}$$

dove $\varepsilon$ è una prefissata tolleranza. La (2.2) determina anche un limite per il numero di iterate infatti:

$$\frac{b_0 - a_0}{2^k} \leq \varepsilon \qquad \Rightarrow \qquad k > \log_2 \left( \frac{b_0 - a_0}{\varepsilon} \right).$$

Poichè $b_k - \alpha \leq b_k - a_k$, il criterio (2.2) garantisce che $\alpha$ è approssimata da $c_{k+1}$ con un errore assoluto minore di $\varepsilon$. Se $0 \notin [a, b]$ si può usare come criterio di stop

$$\frac{b_k - a_k}{\min(|a_k|, |b_k|)} \leq \varepsilon \tag{2.3}$$

che garantisce che $\alpha$ è approssimata da $c_{k+1}$ con un errore relativo minore di $\varepsilon$. Un ulteriore criterio di stop è fornito dal test:

$$|f(c_k)| \leq \varepsilon. \tag{2.4}$$

È comunque buona norma utilizzare due criteri di stop insieme, per esempio (2.2) e (2.4) oppure (2.3) e (2.4).

### 2.3.1 Il metodo della falsa posizione

Una variante del metodo delle bisezioni è appunto il metodo della falsa posizione. Partendo sempre da una funzione $f(x)$ continua in un intervallo $[a, b]$ tale che $f(a)f(b) < 0$, in questo caso si approssima la radice considerando l'intersezione della retta passante per i punti $(a, f(a))$ e $(b.f(b))$ con l'asse $x$. L'equazione della retta è

$$y = f(a) + \frac{f(b) - f(a)}{b - a}(x - a)$$

pertanto il punto $c_1$, sua intersezione con l'asse $x$, è:

$$c_1 = a - f(a)\frac{b - a}{f(b) - f(a)}.$$

Si testa a questo punto l'appartenenza della radice $\alpha$ ad uno dei due intervalli $[a, c_1]$ e $[c_1, b]$ e si procede esattamente come nel caso del metodo delle bisezioni, ponendo

$$[a_1, b_1] \equiv \begin{cases} a_1 = a, \quad b_1 = c_1 & \text{se } f(a)f(c_1) < 0 \\ a_1 = c_1, \quad b_1 = b & \text{se } f(a)f(c_1) > 0. \end{cases}$$

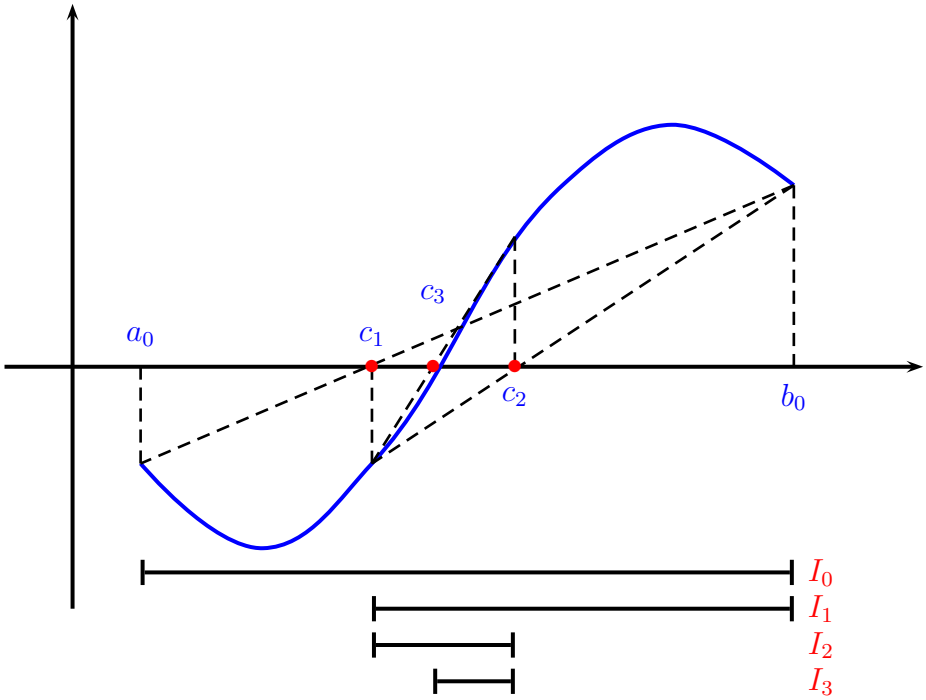
Ad un generico passo $k$ si calcola

$$c_{k+1} = a_k - f(a_k)\frac{b_k - a_k}{f(b_k) - f(a_k)}$$

e si pone

$$[a_{k+1}, b_{k+1}] \equiv \begin{cases} a_{k+1} = a_k \quad b_{k+1} = c_{k+1} & \text{se } f(a_k)f(c_{k+1}) < 0 \\ a_{k+1} = c_{k+1} \quad b_{k+1} = b_k & \text{se } f(a_k)f(c_{k+1}) > 0. \end{cases}$$

Anche per questo metodo è possibile dimostrare la convergenza nella sola ipotesi di continuità della funzione $f(x)$. Nella seguente figura è rappresentato graficamente il metodo della falsa posizione.




```
function [alfa,k]=bisezione(f,a,b,tol)
%
% La funzione approssima la radice con il metodo di bisezione
%
% Parametri di input
```

```
% f = funzione della quale calcolare la radice
% a = estremo sinistro dell'intervallo
% b = estremo destro dell'intervallo
% tol = precisione fissata
%
% Parametri di output
% alfa = approssimazione della radice
% k = numero di iterazioni
%
if nargin==3
    tol = 1e-8; % Tolleranza di default
end
fa = feval(f,a);
fb = feval(f,b);
if fa*fb>0
    error('Il metodo non e'' applicabile')
end
c = (a+b)/2;
fc = feval(f,c);
k = 0;
while (b-a)>tol | abs(fc)>tol
    if fa*fc<0
        b = c;
        fb = fc;
    else
        a = c;
        fa = fc;
    end
    c = (a+b)/2;
    fc = feval(f,c);
    if nargout==2
        k = k+1;
    end
end
alfa = c;
return
```

## 2.4 Metodi di Iterazione Funzionale

Il metodo di bisezione può essere applicato ad una vastissima classe di funzioni, in quanto per poter essere applicato si richiede solo la continuità della funzione. Tuttavia ha lo svantaggio di risultare piuttosto lento, infatti ad ogni passo si guadagna in precisione una cifra binaria. Per ridurre l'errore di un decimo sono mediamente necessarie 3.3 iterazioni. Inoltre la velocità di convergenza non dipende dalla funzione $f(x)$ poichè il metodo utilizza esclusivamente il segno assunto dalla funzione in determinati punti e non il suo valore. Il metodo delle bisezioni può essere comunque utilizzato con profitto per determinare delle buone approssimazioni della radice $\alpha$ che possono essere utilizzate dai metodi iterativi che stiamo per descrivere.
Infatti richiedendo alla $f$ supplementari condizioni di regolarità è possibile individuare una vasta classe di metodi che forniscono le stesse approssimazioni del metodo di bisezione utilizzando però un numero di iterate molto minore. In generale questi metodi sono del tipo:

$$x_{k+1} = g(x_k) \qquad k = 0, 1, 2, \ldots \tag{2.5}$$

dove $x_0$ è un'assegnato valore iniziale e forniscono un'approssimazione delle soluzioni dell'equazione

$$x = g(x). \tag{2.6}$$

Ogni punto $\alpha$ tale che $\alpha = g(\alpha)$ si dice punto fisso o punto unito di $g$.
Per poter applicare uno schema del tipo (2.5) all'equazione $f(x) = 0$, bisogna prima trasformare questa nella forma (2.6). Ad esempio se $[a, b]$ è l'intervallo di definizione di $f$ ed $h(x)$ è una qualunque funzione tale che $h(x) \neq 0$, per ogni $x \in [a, b]$, si può porre:

$$g(x) = x - \frac{f(x)}{h(x)}. \tag{2.7}$$

Ovviamente ogni punto fisso di $g$ è uno zero di $f$ e viceversa.
Nel seguente teorema dimostriamo che se una successione è definita dalla relazione (2.5) risulta convergente il suo limite coincide con il punto fisso della funzione $g(x)$ (che coincide con la radice $\alpha$ della funzione $f(x)$.

**Teorema 2.4.1** *Sia $g \in \mathcal{C}([a, b])$ e assumiamo che la successione $\{x_k\}$ generata da (2.5) sia contenuta in $[a, b]$. Allora se tale successione converge, il limite è il punto fisso di $g$.*

Dimostrazione.

$$\alpha = \lim_{k\to+\infty} x_{k+1} = \lim_{k\to+\infty} g(x_k) = g\left(\lim_{k\to+\infty} x_k\right) = g(\alpha). \ \square$$

Il seguente teorema fornisce una condizione sufficiente per la convergenza della successione definita dalla relazione (2.5). Questo risultato, unitamente al Teorema 2.4.1, garantisce, sotto le ipotesi del Teorema 2.4.2, la convergenza della successione $x_k$ alla radice della funzione $f(x)$.

**Teorema 2.4.2** *Sia $\alpha$ punto fisso di $g$ e $g \in \mathcal{C}^1([\alpha-\rho, \alpha+\rho])$, per qualche $\rho > 0$, se si suppone che*

$$|g'(x)| < 1, \qquad \text{per ogni } x \in [\alpha-\rho, \alpha+\rho]$$

*allora valgono le seguenti asserzioni:*

1. *se $x_0 \in [\alpha-\rho, \alpha+\rho]$ allora anche $x_k \in [\alpha-\rho, \alpha+\rho]$ per ogni $k$;*
2. *la successione $\{x_k\}$ converge ad $\alpha$;*
3. *$\alpha$ è l'unico punto fisso di $g(x)$ nell'intervallo $[\alpha-\rho, \alpha+\rho]$.*

Dimostrazione. Sia

$$\lambda = \max_{|x-\alpha|\le\rho} |g'(x)| < 1.$$

Innanzitutto dimostriamo per induzione che tutti gli elementi della successione $\{x_k\}$ sono contenuti nell'intervallo di centro $\alpha$ e ampiezza $2\rho$. Per $k = 0$ si ha banalmente $x_0 \in [\alpha-\rho, \alpha+\rho]$. Assumiamo che $|x_k - \alpha| \le \rho$ e dimostriamolo per $k+1$.

$$|x_{k+1} - \alpha| = |g(x_k) - g(\alpha)| = |g'(\xi_k)||x_k - \alpha|$$

dove $|\xi_k - \alpha| < |x_k - \alpha| \le \rho$ e l'ultima uguaglianza segue dall'applicazione del teorema di Lagrange. Pertanto

$$|x_{k+1} - \alpha| \le \lambda|x_k - \alpha| < |x_k - \alpha| \le \rho.$$

Proviamo ora che:

$$\lim_{k\to+\infty} x_k = \alpha.$$

Da $|x_{k+1} - \alpha| \leq \lambda |x_k - \alpha|$ segue

$$|x_{k+1} - \alpha| \leq \lambda^{k+1} |x_0 - \alpha|.$$

Conseguentemente qualunque sia $x_0$ si ha:

$$\lim_{k \to +\infty} |x_k - \alpha| = 0 \Leftrightarrow \lim_{k \to +\infty} x_k = \alpha.$$

Per dimostrare l'unicità del punto ragioniamo per assurdo che supponiamo ch i punti fissi sono due, $\alpha, \beta \in [\alpha - \rho, \alpha + \rho]$. Allora

$$|\alpha - \beta| = |g(\alpha) - g(\beta)| = |g'(\xi)||\alpha - \beta|$$

con $\xi \in [\alpha - \rho, \alpha + \rho]$. Poichè $|g'(\xi)| < 1$ si ha

$$|\alpha - \beta| < |\alpha - \beta|$$

e ciò è assurdo. □

Nelle figure 2.2 e 2.1 è rappresentata l'interpretazione geometrica di un metodo di iterazione funzionale in ipotesi di convergenza.

**Definizione 2.4.1** *Un metodo iterativo del tipo (2.5) si dice localmente convergente ad una soluzione $\alpha$ del problema $f(x) = 0$ se esiste un intervallo $[a, b]$ contenente $\alpha$ tale che, per ogni $x_0 \in [a, b]$, la successione generata da (2.5) converge a $\alpha$.*

Come abbiamo già visto nel caso del metodo delle bisezioni anche per metodi di iterazione funzionale è necessario definire dei criteri di arresto per il calcolo delle iterazioni. Teoricamente, una volta stabilita la precisione voluta, $\varepsilon$, si dovrebbe arrestare il processo iterativo quando l'errore al passo $k$

$$e_k = |\alpha - x_k|$$

risulta minore della tolleranza prefissata $\varepsilon$. In pratica l'errore non può essere noto quindi è necessario utilizzare qualche stima. Per esempio si potrebbe considerare la differenza tra due iterate consecutive e fermare il calcolo degli elementi della successione quando

$$|x_{k+1} - x_k| \leq \varepsilon,$$

oppure

$$\frac{|x_{k+1} - x_k|}{\min(|x_{k+1}|, |x_k|)} \leq \varepsilon \qquad |x_{k+1}|, |x_k| \neq 0$$

se i valori hanno un ordine di grandezza particolarmente elevato. Una stima alternativa valuta il residuo della funzione rispetto al valore in $\alpha$, cioè

$$|f(x_k)| \leq \varepsilon.$$

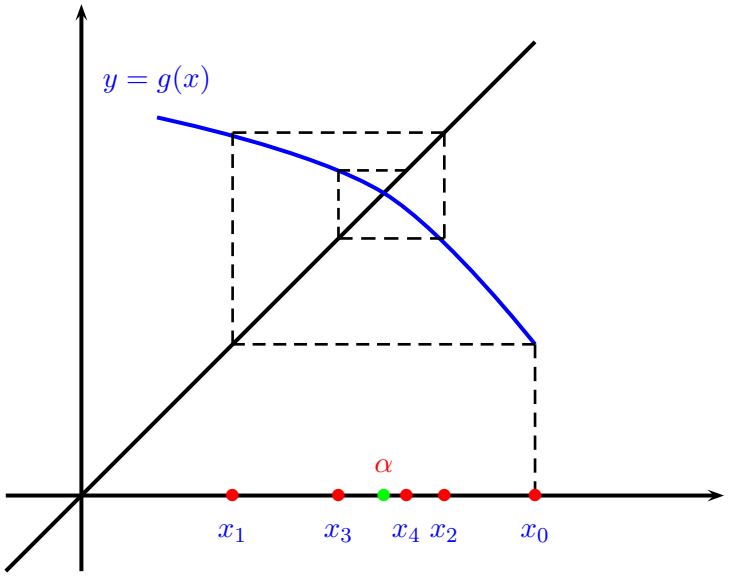



Figura 2.1: Interpretazione geometrica del processo $x_{k+1} = g(x_k)$, se $-1 < g'(\alpha) \leq 0$.

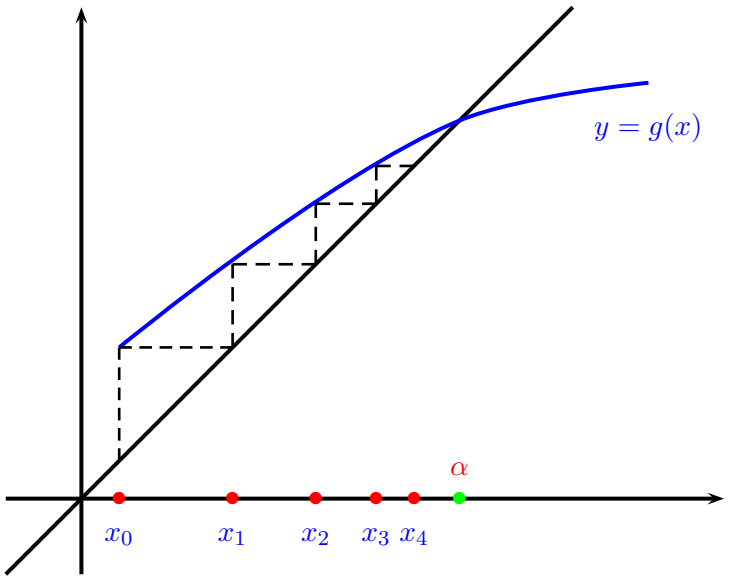



Figura 2.2: Interpretazione geometrica del processo $x_{k+1} = g(x_k)$, se $0 \leq g'(\alpha) < 1$.

### 2.4.1 Ordine di Convergenza

Per confrontare differenti metodi iterativi che approssimano la stessa radice $\alpha$ di $f(x) = 0$, si può considerare la velocità con cui tali successioni convergono verso $\alpha$. Lo studio della velocità di convergenza passa attraverso il concetto di ordine del metodo.

**Definizione 2.4.2** *Sia $\{x_k\}_{k=0}^{\infty}$ una successione convergente ad $\alpha$ e tale che $x_k \neq \alpha$, per ogni $k$. Se esiste un numero reale $p \geq 1$ tale che*

$$\lim_{k\to+\infty} \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} = \gamma \qquad con \quad \begin{cases} 0 < \gamma \leq 1 & se\ p = 1 \\ \gamma > 0 & se\ p > 1 \end{cases} \tag{2.8}$$

*allora si dice che la successione ha ordine di convergenza $p$. La costante $\gamma$ prende il nome di costante asintotica di convergenza.*

In particolare se $p = 1$ e $0 < \gamma < 1$ allora la convergenza si dice lineare, mentre se $p > 1$ allora la convergenza si dice genericamente superlineare, per esempio se $p = 2$ la convergenza si dice quadratica, se $p = 3$ cubica e così via.

*Osservazione.* La relazione (2.8) implica che esiste una costante positiva $\beta$ ($\beta \simeq \gamma$) tale che, per $k$ sufficientemente grande:

$$|x_{k+1} - \alpha| \leq \beta |x_k - \alpha|^p \tag{2.9}$$

ed anche

$$\frac{|x_{k+1} - \alpha|}{|\alpha|} \leq \beta |\alpha|^{p-1} \left| \frac{x_k - \alpha}{\alpha} \right|^p . \tag{2.10}$$

Le (2.9) e (2.10) indicano che la riduzione di errore (assoluto o relativo) ad ogni passo è tanto maggiore quanto più alto è l'ordine di convergenza e, a parità di ordine, quanto più piccola è la costante asintotica di convergenza.
In generale l'ordine di convergenza è un numero reale maggiore o uguale a 1. Tuttavia per i metodi di iterazione funzionale di tipo (2.5) è un numero intero per il quale vale il seguente teorema.

**Teorema 2.4.3** *Sia $\{x_k\}_{k=0}^{\infty}$ una successione generata dallo schema (2.5) convergente ad $\alpha$, punto fisso di $g(x)$, funzione sufficientemente derivabile in un intorno di $\alpha$. La successione ha ordine di convergenza $p \geq 1$ se e solo se*

$$g'(\alpha) = g''(\alpha) = \cdots = g^{(p-1)}(\alpha) = 0, \qquad g^{(p)}(\alpha) \neq 0. \tag{2.11}$$

*Dimostrazione.* Scriviamo lo sviluppo in serie di Taylor della funzione $g(x)$ in $x_k$ prendendo come punto iniziale $\alpha$:

$$\begin{aligned} g(x_k) &= g(\alpha) + g'(\alpha)(x_k - \alpha) + \frac{g''(\alpha)}{2!}(x_k - \alpha)^2 + \dots \\ &\quad \dots + \frac{g^{(p-1)}(\alpha)}{(p-1)!}(x_k - \alpha)^{p-1} + \frac{g^{(p)}(\xi_k)}{p!}(x_k - \alpha)^p. \end{aligned}$$

Sostituendo a $g(x_k)$ il valore $x_{k+1}$ e sfruttando l'ipotesi che $\alpha$ è punto fisso di $g(x)$ risulta

$$\begin{aligned} x_{k+1} - \alpha &= g'(\alpha)(x_k - \alpha) + \frac{g''(\alpha)}{2!}(x_k - \alpha)^2 + \dots \\ &\quad \dots + \frac{g^{(p-1)}(\alpha)}{(p-1)!}(x_k - \alpha)^{p-1} + \frac{g^{(p)}(\xi_k)}{p!}(x_k - \alpha)^p \end{aligned}$$

dove $\xi$ è compreso tra $x_k$ e $\alpha$. Quindi se vale l'ipotesi (2.11) e passando ai moduli risulta

$$\frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} = \frac{|g^{(p)}(\xi_k)|}{p!}$$

e quindi

$$\lim_{k\to\infty} \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} = \frac{|g^{(p)}(\alpha)|}{p!}.$$

Viceversa supponiamo per ipotesi che la successione ha ordine di convergenza $p$ e dimostriamo che

$$g'(\alpha) = g''(\alpha) = \dots = g^{(p-1)}(\alpha) = 0.$$

Ipotizziamo, per assurdo, che esista una derivata di ordine $i$, $i < p$, diversa da zero, ovvero

$$g^{(i)}(\alpha) \neq 0.$$

Scriviamo lo sviluppo in serie di Taylor di $x_{k+1} = g(x_k)$:

$$x_{k+1} = g(x_k) = g(\alpha) + \frac{g^{(i)}(\xi_k)}{i!}(x_k - \alpha)^i$$

da cui

$$x_{k+1} - \alpha = \frac{g^{(i)}(\xi_k)}{i!}(x_k - \alpha)^i.$$

Passando ai moduli e calcolando il limite della successione si ottiene:

$$\lim_{k\to+\infty} \frac{|x_{k+1}-\alpha|}{|x_k-\alpha|^i} = \frac{|g^{(i)}(\alpha)|}{i!} \neq 0$$

da cui segue che la successione ha ordine $i < p$ in contrasto con l'ipotesi fatta. □

*Osservazione.* L'ordine di convergenza $p$ può essere anche un numero non intero. In questo caso, posto $q = [p]$, se $g \in \mathcal{C}^q([a,b])$ si ha anche

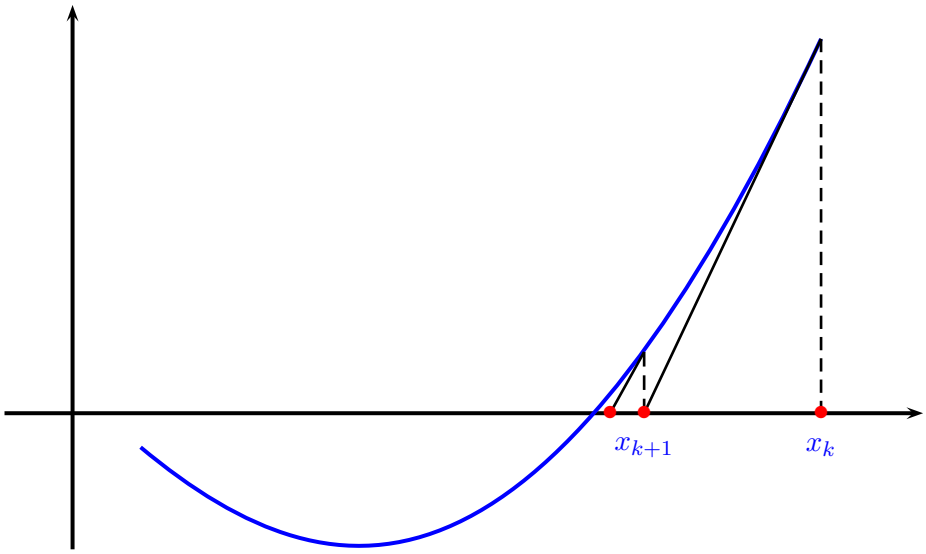
$$g'(\alpha) = g''(\alpha) = \cdots = g^{(q)}(\alpha) = 0,$$

e che $g$ non ha derivata di ordine $q+1$ altrimenti per il precedente teorema tutte le successioni ottenute da (2.5) a partire da $x_0 \in [\alpha-\rho, \alpha+\rho]$ avrebbero ordine almeno $q+1$.

**Definizione 2.4.3** *Un metodo iterativo convergente ad $\alpha$ si dice di ordine $p$ (di ordine almeno $p$) se tutte le successioni ottenute al variare del punto iniziale in un opportuno intorno di $\alpha$ convergono con ordine di convergenza $p$ (almeno $p$).*

### 2.4.2 Metodo di Newton-Raphson

Nell'ipotesi che $f$ sia derivabile ed ammetta derivata prima continua allora un altro procedimento per l'approssimazione dello zero della funzione $f(x)$ è il metodo di Newton-Raphson, noto anche come metodo delle tangenti. Nella figura seguente è riportata l'interpretazione geometrica di tale metodo. A partire dall'approssimazione $x_0$ si considera la retta tangente la funzione $f$ passante per il punto $P_0$ di coordinate $(x_0, f(x_0))$. Si calcola l'ascissa $x_1$ del punto di intersezione tra tale retta tangente e l'asse delle $x$ e si ripete il procedimento a partire dal punto $P_1$ di coordinate $(x_1, f(x_1))$. Nella seguente figura è rappresentato graficamente il metodo di Newton-Raphson.




Per ricavare la funzione iteratrice del metodo consideriamo l'equazione della retta tangente la funzione $y = f(x)$ nel punto di coordinate $(x_k, f(x_k))$

$$y - f(x_k) = f'(x_k)(x - x_k).$$

Posto $y = 0$ ricaviamo l'espressione di $x$ che diventa il nuovo elemento della successione $x_{k+1}$:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \qquad k = 0, 1, 2, \ldots \tag{2.12}$$

che equivale, scegliendo in (2.7) $h(x) = f'(x)$, al metodo di iterazione funzionale in cui la funzione $g(x)$ è

$$g(x) = x - \frac{f(x)}{f'(x)}. \tag{2.13}$$

Per la convergenza e l'ordine del metodo di Newton-Raphson vale il seguente teorema.

**Teorema 2.4.4** *Sia $f \in \mathcal{C}^3([a, b])$, tale che $f'(x) \neq 0$, per $x \in [a, b]$, dove $[a, b]$ è un opportuno intervallo contenente $\alpha$, allora valgono le seguenti proposizioni:*

1. *esiste un intervallo $[\alpha - \rho, \alpha + \rho]$, tale che, scelto $x_0$ appartenente a tale intervallo, la successione definita dal metodo di Newton-Raphson è convergente ad $\alpha$;*

*2. la convergenza è di ordine $p \geq 2$.*

*Dimostrazione.* Per valutare la convergenza del metodo calcoliamo la derivata prima della funzione iteratrice:

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

Poichè $f'(\alpha) \neq 0$ risulta:

$$g'(\alpha) = \frac{f(\alpha)f''(\alpha)}{[f'(\alpha)]^2} = 0$$

quindi, fissato un numero positivo $\kappa < 1$, esiste $\rho > 0$ tale che per ogni $x \in [\alpha - \rho, \alpha + \rho]$ si ha $|g'(x)| < \kappa$ e quindi vale il teorema di convergenza 2.4.2.
Per dimostrare la seconda parte del teorema si deve calcolare la derivata seconda di $g(x)$:

$$g''(x) = \frac{[f'(x)f''(x) + f(x)f'''(x)][f'(x)]^2 - 2f(x)f'(x)[f''(x)]^2}{[f'(x)]^4}.$$

Calcolando la derivata seconda in $x = \alpha$ risulta

$$g''(\alpha) = \frac{f''(\alpha)}{f'(\alpha)} \tag{2.14}$$

ne segue che se $f''(\alpha) \neq 0$ allora anche $g''(\alpha) \neq 0$ e quindi, applicando il Teorema 2.4.3, l'ordine $p = 2$. Se invece $f''(\alpha) = 0$ allora l'ordine è almeno pari a 3. Dalla relazione 2.14 segue inoltre che la costante asintotica di convergenza vale

$$\gamma = \frac{1}{2}\left|\frac{f''(\alpha)}{f'(\alpha)}\right|. \ \square$$

Il Teorema 2.4.4 vale nell'ipotesi in cui $f'(\alpha) \neq 0$, cioè se $\alpha$ è una radice semplice di $f(x)$. Se invece la radice $\alpha$ ha molteplicità $r > 1$ l'ordine di convergenza del metodo non è più 2. In questo caso infatti si può porre

$$f(x) = q(x)(x - \alpha)^r, \qquad q(\alpha) \neq 0,$$

quindi riscrivendo la funzione iteratrice del metodo di Newton-Raphson risulta

$$g(x) = x - \frac{q(x)(x-\alpha)}{rq(x) + q'(x)(x-\alpha)},$$

da cui, dopo una serie di calcoli, risulta

$$g'(\alpha) = 1 - \frac{1}{r}. \tag{2.15}$$

Pertanto, poichè $r > 1$ risulta $|g'(x)| < 1$ e quindi per il Teorema 2.4.2 il metodo è ancora convergente ma, applicando il Teorema 2.4.3 l'ordine di convergenza è 1.
Se si conosce la molteplicità della radice si può modificare il metodo di Newton-Raphson ottenendo uno schema numerico con ordine 2. Ponendo

$$x_{k+1} = x_k - r\frac{f(x_k)}{f'(x_k)} \qquad k = 0, 1, 2, \ldots$$

si definisce un metodo con la seguente funzione iteratrice

$$g(x) = x - r\frac{f(x)}{f'(x)}$$

da cui segue, tenendo conto della (2.15), che

$$g'(\alpha) = 0.$$

Riportiamo nel seguito l'implementazione MatLab del metodo di Newton-Raphson.

```
function [alfa,k]=newton(f,f1,x0,tol,Nmax)
%
% La funzione calcolo un'approssimazione
% della radice con il metodo di Newton-Raphson
%
% Parametri di input
% f = funzione della quale calcolare la radice
% f1 = derivata prima della funzione f
% x0 = approssimazione iniziale della radice
% tol = precisione fissata
```

```
% Nmax = numero massimo di iterazioni fissate
%
% Parametri di output
% alfa = approssimazione della radice
% k = numero di iterazioni
%
if nargin==3
    tol=1e-8;
    Nmax=1000;
end
k=0;
x1=x0-feval(f,x0)/feval(f1,x0);
fx1 = feval(f,x1);
while abs(x1-x0)>tol | abs(fx1)>tol
    x0 = x1;
    x1 = x0-feval(f,x0)/feval(f1,x0);
    fx1 = feval(f,x1);
    k=k+1;
    if k>Nmax
        disp('Il metodo non converge');
        alfa = inf;
        break
    end
end
alfa=x1;
return
```

**Esempio 2.4.1** *Approssimare il numero* $\alpha = \sqrt[m]{c}$ *con* $m \in \mathbb{R}$, $m \geq 2$, $c > 0$.

Il numero $\alpha$ cercato è lo zero della funzione

$$f(x) = x^m - c.$$

Poichè per $x > 0$ la funzione risulta essere monotona allora è sufficiente scegliere un qualsiasi $x_0 > 0$ per ottenere una successione convergente alla radice $m-$esima di $c$. Il metodo di Newton-Raphson fornisce la formula

$$x_{k+1} = x_k - \frac{x_k^m - c}{m x_k^{m-1}} = \frac{1}{m}\left[(m-1)x_k + c x_k^{1-m}\right], \quad k = 0, 1, 2, \ldots.$$

Per $m = 2$ lo schema diviene

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{c}{x_k}\right),$$

che è la cosiddetta formula di Erone per il calcolo della radice quadrata, nota già agli antichi Greci.
Considerando come esempio $m = 4$ e $c = 3$, poichè $f(0) < 0$ e $f(3) > 0$ allora si può applicare il metodo di bisezione ottenendo la seguente sucessione di intervalli:

| Intervallo | Punto medio | Valore di $f$ nel punto medio |
|---|---|---|
| $[0, 3]$ | $c = 1.5$ | $f(c) = 2.0625$ |
| $[0, 1.5]$ | $c = 0.75$ | $f(c) = -2.6836$ |
| $[0.75, 1.5]$ | $c = 1.125$ | $f(c) = -1.3982$ |
| $[1.125, 1.5]$ | $c = 1.3125$ | $f(c) = -0.0325$ |
| $\vdots$ | $\vdots$ | $\vdots$ |

Dopo 10 iterazioni $c = 1.3154$ mentre $\alpha = 1.3161$, e l'errore è pari circa a $6.4433 \cdot 10^{-4}$.
Applicando il metodo di Newton-Raphson, si ottiene il processo iterativo

$$x_{k+1} = x_k - \frac{1}{3}\left(2x_k + 3x_k^{-3}\right).$$

Poichè per $x > 0$ la funzione è monotona crescente allora si può scegliere $x_0 = 3$ come approssimazione iniziale, ottenendo la seguente successione:

| | |
|---|---|
| $x_0 = 3$ | $f(x_0) = 78$ |
| $x_1 = 2.2778$ | $f(x_1) = 23.9182$ |
| $x_2 = 1.7718$ | $f(x_2) = 6.8550$ |
| $x_3 = 1.4637$ | $f(x_3) = 1.5898$ |
| $x_4 = 1.3369$ | $f(x_4) = 0.1948$ |
| $x_5 = 1.3166$ | $f(x_5) = 0.0044$ |
| $\vdots$ | $\vdots$ |

Dopo 10 iterazioni l'approssimazione è esatta con un errore dell'ordine di $10^{-16}$.

### 2.4.3 Il metodo della direzione costante

Se applicando ripetutamente la formula di Newton-Raphson accade che la derivata prima della funzione $f(x)$ si mantiene sensibilmente costante allora si può porre

$$M = f'(x)$$

e applicare la formula

$$x_{k+1} = x_k - \frac{f(x_k)}{M} \tag{2.16}$$

anzichè la (2.12). La (2.16) definisce un metodo che viene detto metodo di Newton semplificato oppure metodo della direzione costante in quanto geometricamente equivale all'applicazione del metodo di Newton in cui anzichè prendere la retta tangente la curva $f$ si considera la retta avente coefficiente angolare uguale a $M$. La funzione iteratrice del metodo è

$$g(x) = x - \frac{f(x)}{M}$$

ed il metodo è convergente se

$$|g'(x)| = \left|1 - \frac{f'(x)}{M}\right| < 1$$

da cui si deduce che è necessario che $f'(x)$ ed $M$ abbiano lo stesso segno.

### 2.4.4 Il Metodo della Secante

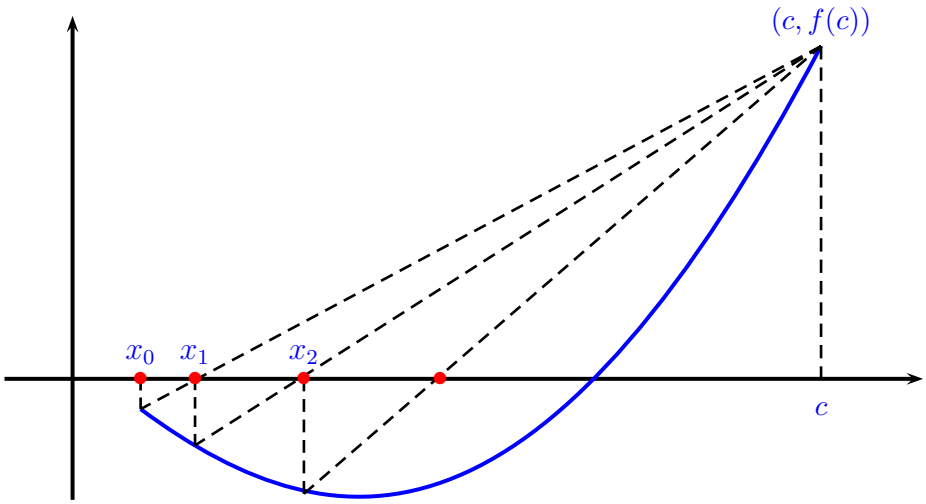
Il metodo della secante è definito dalla relazione

$$x_{k+1} = x_k - f(x_k)\frac{x_k - c}{f(x_k) - f(c)}$$

dove $c \in [a, b]$. Il significato geometrico di tale metodo è il seguente: ad un generico passo $k$ si considera la retta congiungente i punti di coordinate $(x_k, f(x_k))$ e $(c, f(c))$ e si pone $x_{k+1}$ pari all'ascissa del punto di intersezione di tale retta con l'asse $x$. Dalla formula si evince che la funzione iteratrice del metodo è

$$g(x) = x - f(x)\frac{x - c}{f(x) - f(c)}.$$

Il metodo è rappresentato graficamente nella seguente figura.




In base alla teoria vista nei paragrafi precedenti il metodo ha ordine di convergenza 1 se $g'(\alpha) \neq 0$. Può avere ordine di convergenza almeno 1 se $g'(\alpha) = 0$. Tale eventualità si verifica se la tangente alla curva in $\alpha$ ha lo stesso coefficiente angolare della retta congiungente i punti $(\alpha, 0)$ e $(c, f(c))$.
Poichè il metodo delle secanti ha lo svantaggio di avere, solitamente, convergenza lineare mentre il metodo di Newton-Raphson, pur avendo convergenza quadratica, ha lo svantaggio di richiedere, ad ogni passo, due valutazioni di funzioni: $f(x_k)$ ed $f'(x_k)$, quindi se il costo computazionale di $f'(x_k)$ è molto più elevato rispetto a quello di $f(x_k)$ può essere più conveniente l'uso di metodi che necessitano solo del calcolo del valore della funzione $f(x)$.

## 2.5 Sistemi di Equazioni non Lineari

Supponiamo che sia $\Omega$ un sottoinsieme di $\mathbb{R}^n$ e che siano assegnate le $n$ funzioni

$$f_i : \Omega \to \mathbb{R}, \qquad i = 1, \ldots, n.$$

Ogni vettore $\boldsymbol{x} \in \mathbb{R}^n$, soluzione del sistema non lineare di $n$ equazioni in $n$ incognite

$$\begin{aligned} &f_1(x_1, x_2, \ldots, x_n) = 0 \\ &f_2(x_1, x_2, \ldots, x_n) = 0 \\ &\vdots \\ &f_n(x_1, x_2, \ldots, x_n) = 0 \end{aligned}$$

prende il nome di radice dell'equazione vettoriale

$$F(\boldsymbol{x}) = 0$$

oppure di zero della funzione vettoriale

$$F : \Omega \to \mathbb{R}^n$$

dove il vettore $F(\boldsymbol{x})$ è definito da:

$$F(\boldsymbol{x}) = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) \end{bmatrix}.$$

Tutti i metodi per la risoluzione del sistema non lineare $F(\boldsymbol{x}) = 0$ partono dalle seguenti due ipotesi:

1. la funzione $F(\boldsymbol{x})$ è calcolabile in ogni punto del dominio $\Omega$;
2. la funzione $F(\boldsymbol{x})$ è continua in un opportuno intorno della radice.

Come nel caso scalare l'equazione $F(\boldsymbol{x}) = 0$ viene trasformata in un problema del tipo

$$\boldsymbol{x} = \Phi(\boldsymbol{x}) \tag{2.17}$$

ovvero

$$x_i = \Phi_i(x_1, x_2, \ldots, x_n), \qquad i = 1, 2, \ldots, n$$

con $\Phi(\boldsymbol{x})$ funzione definita in $\Omega$ e scelta in modo tale che le proprietà richiesta ad $F(\boldsymbol{x})$ si trasferiscano su $\Phi$, cioè anch'essa deve essere continua in un opportuno intorno della radice e calcolabile nell'insieme di definizione. Il motivo di tali richieste è che la funzione $\Phi(\boldsymbol{x})$ viene utilizzata per definire una successione di vettori nel seguente modo. Sia $\boldsymbol{x}^{(0)}$ un vettore iniziale appartenente a $\Omega$ e definiamo la seguente successione

$$\boldsymbol{x}^{(k+1)} = \Phi(\boldsymbol{x}^{(k)}), \qquad k = 0, 1, 2, 3, \ldots$$

ovvero

$$x_i^{(k+1)} = \Phi_i(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}), \qquad i = 1, 2, \ldots, n.$$

La funzione $\Phi(\boldsymbol{x})$ prende il nome di funzione iteratrice dell'equazione non lineare $F(\boldsymbol{x}) = 0$. Ricordiamo che un vettore $\boldsymbol{\alpha}$ che soddisfa la (2.17) viene

detto punto fisso di $\Phi(\boldsymbol{x})$ (oppure punto unito). La successione dei vettori $\boldsymbol{x}^{(k)}$ definisce il metodo delle approssimazioni successive per il calcolo appunto di tale punto fisso. Quello che si richiede a tale successione è che essa converga al vettore $\boldsymbol{\alpha}$, soluzione del sistema non lineare. In questo caso per convergenza si intende che

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \boldsymbol{\alpha}$$

cioè, in termini di componenti,

$$\lim_{k\to\infty} \boldsymbol{x}_i^{(k)} = \boldsymbol{\alpha}_i.$$

Per la convergenza del metodo delle approssimazioni successive vale quindi il seguente teorema.

**Teorema 2.5.1** *Se la funzione $\Phi(\boldsymbol{x})$ è differenziabile con continuità in un intorno del punto fisso $\boldsymbol{\alpha}$, e risulta*

$$\rho(J_\Phi(\boldsymbol{\alpha})) < 1$$

*allora, scelto $\boldsymbol{x}^{(0)}$ appartenente a tale intorno, la successione costruita con il metodo delle approssimazioni successive è convergente a $\boldsymbol{\alpha}$.*

Chiaramente il risultato appena enunciato ha un'importanza teorica in quanto generalmente è molto complesso (o non è possibile) conoscere gli autovalori della matrice Jacobiana nella soluzione del sistema non lineare.

### 2.5.1 Il Metodo di Newton per Sistemi non Lineari

Se si conosce abbastanza bene l'approssimazione iniziale della soluzione del sistema di equazioni

$$F(\boldsymbol{x}) = 0 \tag{2.18}$$

il metodo di Newton risulta molto efficace. Il Metodo di Newton per risolvere il sistema (2.18) può essere derivato in modo semplice come segue. Sia $\boldsymbol{x}^{(k)}$ una buona approssimazione a $\boldsymbol{\alpha}$, soluzione di $F(\boldsymbol{x}) = 0$, possiamo allora scrivere lo sviluppo in serie della funzione $F$ valutata nella soluzione del sistema non lineare prendendo come punto iniziale proprio il vettore $\boldsymbol{x}^{(k)}$ :

$$0 = F(\boldsymbol{\alpha}) = F(\boldsymbol{x}^{(k)}) + J_F(\boldsymbol{\delta}_k)(\boldsymbol{\alpha} - \boldsymbol{x}^{(k)})$$

dove $\boldsymbol{\delta}_k$ è un vettore appartenente al segmento congiungente $\boldsymbol{\alpha}$ e $\boldsymbol{x}^{(k)}$ e $J_F(\boldsymbol{x})$ indica la matrice Jacobiana i cui elementi sono le derivate prime delle funzioni componenti di $F(\boldsymbol{x})$ :

$$J_F(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\boldsymbol{x}) & \frac{\partial f_1}{\partial x_2}(\boldsymbol{x}) & \dots & \frac{\partial f_1}{\partial x_n}(\boldsymbol{x}) \\ \frac{\partial f_2}{\partial x_1}(\boldsymbol{x}) & \frac{\partial f_1}{\partial x_2}(\boldsymbol{x}) & \dots & \frac{\partial f_1}{\partial x_n}(\boldsymbol{x}) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\boldsymbol{x}) & \frac{\partial f_n}{\partial x_2}(\boldsymbol{x}) & \dots & \frac{\partial f_n}{\partial x_n}(\boldsymbol{x}) \end{bmatrix}.$$

Suppondendo ora che la matrice Jacobiana sia invertibile possiamo scrivere,

$$\boldsymbol{\alpha} - \boldsymbol{x}^{(k)} = -J_F^{-1}(\boldsymbol{\delta}_k)F(\boldsymbol{x}^{(k)}) \Rightarrow \boldsymbol{\alpha} = \boldsymbol{x}^{(k)} - J_F^{-1}(\boldsymbol{\delta}_k)F(\boldsymbol{x}^{(k)}). \tag{2.19}$$

Se $\boldsymbol{x}^{(k)}$ è sufficietemente vicino a $\boldsymbol{\alpha}$ allora possiamo confondere $\boldsymbol{x}^{(k)}$ con $\boldsymbol{\delta}_k$: in tal modo però (2.19) non fornirà esattamente $\boldsymbol{\alpha}$ ma una sua ulteriore approssimazione, che indichiamo con $\boldsymbol{x}^{(k+1)}$. In questo modo abbiamo definito il seguente processo iterativo

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - J_F^{-1}(\boldsymbol{x}^{(k)})F(\boldsymbol{x}^{(k)}). \tag{2.20}$$

che definisce, appunto il metodo di Newton.
Può essere interessante soffermarsi su alcuni dettagli di implementazione del metodo (2.20). Poniamo infatti

$$\boldsymbol{z}^{(k)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$$

e osserviamo che, moltiplicando per la matrice $J_F(\boldsymbol{x}^{(k)})$ l'espressione del metodo di Newton diventa

$$J_F(\boldsymbol{x}^{(k)})\boldsymbol{z}^{(k)} = -F(\boldsymbol{x}^{(k)})$$

da cui, risolvendo il sistema lineare che ha $J_F(\boldsymbol{x}^{(k)})$ come matrice dei coefficienti e $-F(\boldsymbol{x}^{(k)})$ come vettore dei termini noti si può ricavare il vettore $z^{(k)}$ e ottenere il vettore al passo $k+1$:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{z}^{(k)}.$$

L'algoritmo, ad un generico passo $k$, può essere così riassunto:

1. Calcolare la matrice $J_F(\boldsymbol{x}^{(k)})$ e il vettore $-F(\boldsymbol{x}^{(k)})$;
2. Risolvere il sistema lineare $J_F(\boldsymbol{x}^{(k)})\boldsymbol{z}^{(k)} = -F(\boldsymbol{x}^{(k)})$;
3. Calcolare il vettore $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{z}^{(k)}$;
4. Valutare la convergenza: fissata una tolleranza $\varepsilon$, se risulta
$$\|\boldsymbol{z}^{(k)}\| \leq \varepsilon$$
allora $\boldsymbol{x}^{(k+1)}$ è una buona approssimazione della soluzione, altrimenti si ritorna al passo 1.

Consideriamo come esempio la funzione vettoriale composta da due componenti
$$f_1(x, y) = x^3 + y - 1, \qquad f_2(x, y) = y^3 - x + 1.$$
Il sistema non lineare
$$F(\boldsymbol{x}) = 0 = \begin{bmatrix} x^3 + y - 1 \\ y^3 - x + 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
ammette come soluzione $x = 1$ e $y = 0$. La matrice Jacobiana di $F(\boldsymbol{x})$ è la seguente
$$J_F(x, y) = \begin{bmatrix} 3x^2 & 1 \\ -1 & 3y^2 \end{bmatrix}$$
pertanto il metodo di Newton è definito dal seguente schema:
$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \begin{bmatrix} 3x_k^2 & 1 \\ -1 & 3y_k^2 \end{bmatrix}^{-1} \begin{bmatrix} x_k^3 + y_k - 1 \\ y_k^3 - x_k + 1 \end{bmatrix}.$$

# Capitolo 3

# Metodi numerici per sistemi lineari

## 3.1 Introduzione

Siano assegnati una matrice non singolare $A \in \mathbb{R}^{n\times n}$ ed un vettore $\boldsymbol{b} \in \mathbb{R}^n$. Risolvere un sistema lineare avente $A$ come matrice dei coefficienti e $\boldsymbol{b}$ come vettore dei termini noti significa trovare un vettore $\boldsymbol{x} \in \mathbb{R}^n$ tale che

$$A\boldsymbol{x} = \boldsymbol{b}. \tag{3.1}$$

Esplicitare la relazione (3.1) significa imporre le uguaglianze tra le componenti dei vettori a primo e secondo membro:

$$\begin{array}{llll} a_{11}x_1+ & a_{12}x_2+ & \cdots+ & a_{1n}x_n = \quad b_1 \\ a_{21}x_1+ & a_{22}x_2+ & \cdots+ & a_{2n}x_n = \quad b_2 \\ \vdots & & & \vdots \\ a_{n1}x_1+ & a_{n2}x_2+ & \cdots+ & a_{nn}x_n = \quad b_n. \end{array} \tag{3.2}$$

Le (3.2) definiscono un sistema di $n$ equazioni algebriche lineari nelle $n$ incognite $x_1, x_2, \ldots, x_n$. Il vettore $\boldsymbol{x}$ viene detto vettore soluzione. Prima di affrontare il problema della risoluzione numerica di sistemi lineari richiamiamo alcuni importanti concetti di algebra lineare.

**Definizione 3.1.1** *Se $A \in \mathbb{R}^{n\times n}$ è una matrice di ordine 1, si definisce determinante di $A$ il numero*

$$\det A = a_{11}.$$

*Se la matrice $A$ è quadrata di ordine $n$ allora fissata una qualsiasi riga (colonna) di $A$, diciamo la $i$-esima ($j$-esima) allora applicando la cosiddetta regola di Laplace il determinante di $A$ è:*

$$\det A = \sum_{j=1}^{n} a_{ij}(-1)^{i+j} \det A_{ij}$$

*dove $A_{ij}$ è la matrice che si ottiene da $A$ cancellando la $i$-esima riga e la $j$-esima colonna.*

Il determinante è pure uguale a

$$\det A = \sum_{i=1}^{n} a_{ij}(-1)^{i+j} \det A_{ij},$$

cioè il determinante è indipendente dall'indice di riga (o di colonna) fissato. Se $A$ è la matrice di ordine 2

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

allora

$$\det A = a_{11}a_{22} - a_{21}a_{12}.$$

Il determinante ha le seguenti proprietà:
1. Se $A$ è una matrice triangolare o diagonale allora

$$\det A = \prod_{i=1}^{n} a_{ii};$$

2. $\det I = 1$;
3. $\det A^T = \det A$;
4. $\det AB = \det A \det B$ (Regola di Binet);
5. se $\alpha \in \mathbb{R}$ allora $\det \alpha A = \alpha^n \det A$;
6. $\det A = 0$ se una riga (o una colonna) è nulla, oppure una riga (o una colonna) è proporzionale ad un'altra riga (o colonna) oppure è combinazione lineare di due (o più) righe (o colonne) di $A$.
7. Se $A$ è una matrice triangolare a blocchi

$$A = \begin{bmatrix} B & C \\ O & D \end{bmatrix}$$

con $B$ e $D$ matrici quadrate, allora

$$\det A = \det B \det D. \tag{3.3}$$

Una matrice $A$ di ordine $n$ si dice non singolare se il suo determinante è diverso da zero, in caso contrario viene detta *singolare*. Si definisce inversa di $A$ la matrice $A^{-1}$ tale che:

$$AA^{-1} = A^{-1}A = I_n$$

Per quello che riguarda il determinante della matrice inversa vale la seguente proprietà:

$$\det A^{-1} = \frac{1}{\det A}.$$

Un metodo universalmente noto per risolvere il problema (3.1) è l'applicazione della cosiddetta Regola di Cramer la quale fornisce:

$$x_i = \frac{\det A_i}{\det A} \qquad i = 1, \ldots, n, \tag{3.4}$$

dove $A_i$ è la matrice ottenuta da $A$ sostituendo la sua $i$-esima colonna con il termine noto $\boldsymbol{b}$. Dalla (3.4) è evidente che per ottenere tutte le componenti del vettore soluzione è necessario il calcolo di $n+1$ determinanti di ordine $n$. Calcoliamo ora il numero di operazioni aritmetiche necessario per calcolare une determinante con la regola di Laplace. Indichiamo con $f(n)$ il numero di operazioni aritmetiche su numeri reali necessario per calcolare un determinante di ordine $n$, ricordando che $f(2) = 3$. La regola di Laplace richiede il calcolo di $n$ determinanti di matrici di ordine $n-1$ (il cui costo computazionale in termini di operazioni è $nf(n-1)$) inoltre $n$ prodotti ed $n-1$ somme algebriche, ovvero

$$f(n) = nf(n-1) + 2n - 1.$$

Per semplicità tralasciamo gli ultimi addendi ottenendo il valore approssimato

$$f(n) \simeq nf(n-1)$$

Applicando lo stesso ragionamento al numero $f(n-1) \simeq (n-1)f(n-2)$ e in modo iterativo si ottiene

$$f(n) \simeq n(n-1)(n-2)\ldots 3f(2) = \frac{3}{2}\, n!.$$

Se $n = 100$ si ha $100! \simeq 10^{157}$. Anche ipotizzando di poter risolvere il problema con un elaboratore in grado di eseguire miliardi di operazioni al secondo sarebbero necessari diversi anni di tempo per calcolare un singolo determinante. Questo esempio rende chiara la necessità di trovare metodi alternativi per risolvere sistemi lineari, in particolare quando le dimensioni sono particolarmente elevate.

## 3.2 Risoluzione di sistemi triangolari

Prima di affrontare la soluzione algoritmica di un sistema lineare vediamo qualche particolare sistema che può essere agevolmente risolto. Assumiamo che il sistema da risolvere abbia la seguente forma:

$$\begin{array}{ccccccccc} a_{11}x_1 & +a_{12}x_2 & \dots & +a_{1i}x_i & \dots & +a_{1n}x_n & = b_1 \\ & a_{22}x_2 & \dots & +a_{2i}x_i & \dots & +a_{2n}x_n & = b_2 \\ & & \ddots & \vdots & & \vdots & \vdots \\ & & & a_{ii}x_i & \dots & +a_{in}x_n & = b_i \\ & & & & \ddots & \vdots & \vdots \\ & & & & & a_{nn}x_n & = b_n \end{array} \tag{3.5}$$

In questo caso la matrice $A$ è detta triangolare superiore. Il determinante di una matrice di questo tipo è uguale al prodotto degli elementi diagonali pertanto la matrice è non singolare se risulta $a_{ii} \neq 0$ per ogni $i$. In questo caso, la soluzione è facilmente calcolabile infatti è sufficiente osservare che nell'ultima equazione compare solo un'incognita che può essere calcolata e che procedendo a ritroso da ogni equazione può essere ricavata un'incognita poichè le successive sono già state calcolate. Il metodo può essere riassunto nelle seguenti formule:

$$\begin{cases} x_n = \dfrac{b_n}{a_{nn}} \\ \\ x_i = \dfrac{b_i - \displaystyle\sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}} \qquad i = n-1, \dots, 1. \end{cases} \tag{3.6}$$

Il metodo (3.6) prende il nome di metodo di sostituzione all'indietro, poichè il vettore $\boldsymbol{x}$ viene calcolato partendo dall'ultima componente.

Anche per il seguente sistema il vettore soluzione è calcolabile in modo analogo.

$$\begin{array}{lllllll} a_{11}x_1 & & & & & & = b_1 \\ a_{21}x_1 & +a_{22}x_2 & & & & & = b_2 \\ \vdots & \vdots & \ddots & & & & \vdots \\ a_{i1}x_1 & +a_{i2}x_2 & \dots & +a_{ii}x_i & & & = b_i \\ \vdots & \vdots & & & \ddots & & \vdots \\ a_{n1}x_1 & +a_{n2}x_2 & \dots & +a_{ni}x_i & \dots & +a_{nn}x_n & = b_n \end{array} \tag{3.7}$$

In questo caso la matrice dei coefficienti è triangolare inferiore e la soluzione viene calcolata con il metodo di sostituzione in avanti:

$$\begin{cases} x_1 = \dfrac{b_1}{a_{11}} \\ \\ x_i = \dfrac{b_i - \displaystyle\sum_{j=1}^{i-1} a_{ij}x_j}{a_{ii}} \qquad i = 2, \dots, n. \end{cases}$$

Concludiamo questo paragrafo facendo alcune considerazioni sul costo computazionale dei metodi di sostituzione. Per costo computazionale di un algoritmo si intende il numero di operazioni che esso richiede per fornire la soluzione di un determinato problema. la misura del costo computazionale di un algoritmo fornisce una stima (seppur grossolana) del tempo che esso richiede per fornire la soluzione approssimata di un determinato problema indipendentemente dall'elaboratore che viene utilizzato e dal linguaggio di programmazione in cui esso è stato codificato. nel caso di algoritmi numerici le operazioni che si contano sono quelle aritmetiche su dati reali. considerando per esempio il metodo di sostituzione in avanti. per calcolare $x_1$ è necessaria una sola operazione (una divisione), per calcolare $x_2$ le operazioni sono tre (un prodotto, una somma algebrica e una divisione), mentre il generico $x_i$ richiede $2i-1$ operazioni ($i-1$ prodotti, $i-1$ somme algebriche e una divisione), indicato con $c(n)$ il numero totale di operazioni necessarie è:

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

sfruttando la proprietà che

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

Il costo computazionale viene sempre valutato in funzione di un determinato parametro (il numero assoluto in sè non avrebbe alcun significato) che, in questo caso è la dimensione del sistema. In questo modo è possibile prevedere il tempo necessario per calcolare la soluzione del problema.

## 3.3 Metodo di Eliminazione di Gauss

L'idea di base del metodo di Gauss è appunto quella di operare delle opportune trasformazioni sul sistema originale $A\boldsymbol{x} = \boldsymbol{b}$, che non costino eccessivamente, in modo da ottenere un sistema equivalente[1] avente come matrice dei coefficienti una matrice triangolare superiore.
Supponiamo di dover risolvere il sistema:

$$\begin{array}{rrrrcl} 2x_1 & +x_2 & +x_3 & & = & -1 \\ -6x_1 & -4x_2 & -5x_3 & +x_4 & = & 1 \\ -4x_1 & -6x_2 & -3x_3 & -x_4 & = & 2 \\ 2x_1 & -3x_2 & +7x_3 & -3x_4 & = & 0. \end{array}$$

Il vettore soluzione di un sistema lineare non cambia se ad un'equazione viene sommata la combinazione lineare di un'altra equazione del sistema. L'idea alla base del metodo di Gauss è quella di ottenere un sistema lineare con matrice dei coefficienti triangolare superiore effettuando opportune combinazioni lineari tra le equazioni. Poniamo

$$A^{(1)} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ -6 & -4 & -5 & 1 \\ -4 & -6 & -3 & -1 \\ 2 & -3 & 7 & -3 \end{bmatrix}, \qquad \boldsymbol{b}^{(1)} = \begin{bmatrix} -1 \\ 1 \\ 2 \\ 0 \end{bmatrix}$$

[1]Due sistemi si dicono equivalenti se ammettono lo stesso insieme di soluzioni, quindi nel nostro caso la stessa soluzione. Osserviamo che se $\boldsymbol{x}^*$ è un vettore tale che $A\boldsymbol{x}^* = \boldsymbol{b}$ e $B$ è una matrice non singolare allora $BA\boldsymbol{x}^* = B\boldsymbol{b}$; viceversa se $BA\boldsymbol{x}^* = B\boldsymbol{b}$ e $B$ è non singolare allora $B^{-1}BA\boldsymbol{x}^* = B^{-1}B\boldsymbol{b}$ e quindi $A\boldsymbol{x}^* = \boldsymbol{b}$. Dunque se $B$ è non singolare i sistemi $A\boldsymbol{x} = \boldsymbol{b}$ e $BA\boldsymbol{x} = B\boldsymbol{b}$ sono equivalenti.

rispettivamente la matrice dei coefficienti e il vettore dei termini noti del sistema di partenza. Calcoliamo un sistema lineare equivalente a quello iniziale ma che abbia gli elementi sottodiagonali della prima colonna uguali a zero. Azzeriamo ora l'elemento $a_{21}^{(1)}$. Lasciamo inalterata la prima equazione. Poniamo

$$l_{21} = -\frac{a_{21}}{a_{11}} = -\frac{-6}{2} = 3$$

e moltiplichiamo la prima equazione per $l_{21}$ ottenendo:

$$6x_1 + 3x_2 + 3x_3 = -3.$$

La nuova seconda equazione sarà la somma tra la seconda equazione e la prima moltiplicata per $l_{21}$:

$$\begin{array}{rrrrll} -6x_1 & -4x_2 & -5x_3 & +x_4 & = 1 & \\ 6x_1 & +3x_2 & +3x_3 & & = -3 & \\ \hline & -x_2 & -2x_3 & +x_4 & = -2 & [\texttt{Nuova seconda equazione}]. \end{array}$$

Prcediamo nello stesso modo per azzerare gli altri elementi della prima colonna. Poniamo

$$l_{31} = -\frac{a_{31}^{(1)}}{a_{11}^{(1)}} = -\frac{-4}{2} = 2$$

e moltiplichiamo la prima equazione per $l_{31}$ ottenendo:

$$4x_1 + 2x_2 + 2x_3 = -2.$$

La nuova terza equazione sarà la somma tra la terza equazione e la prima moltiplicata per $l_{31}$:

$$\begin{array}{rrrrll} -4x_1 & -6x_2 & -3x_3 & -x_4 & = 2 & \\ 4x_1 & +2x_2 & +2x_3 & & = -2 & \\ \hline & -4x_2 & -x_3 & -x_4 & = 0 & [\texttt{Nuova terza equazione}]. \end{array}$$

Poniamo ora

$$l_{41} = -\frac{a_{41}^{(1)}}{a_{11}^{(1)}} = -\frac{2}{2} = -1$$

e moltiplichiamo la prima equazione per $l_{41}$ ottenendo:

$$-2x_1 - x_2 - x_3 = 1.$$

La nuova quarta equazione sarà la somma tra la quarta equazione e la prima moltiplicata per $l_{41}$:

$$\begin{array}{rrrrl} 2x_1 & -3x_2 & +7x_3 & -3x_4 & = 0 \\ -2x_1 & -x_2 & -x_3 & & = 1 \\ \hline & -4x_2 & +6x_3 & -3x_4 & = 1 \end{array} \quad [\texttt{Nuova quarta equazione}].$$

I numeri $l_{21}, l_{3,1}, \ldots$ sono detti moltiplicatori.
Al secondo passo il sistema lineare è diventato:

$$\begin{array}{rrrrl} 2x_1 & +x_2 & +x_3 & & = -1 \\ & -x_2 & -2x_3 & +x_4 & = -2 \\ & -4x_2 & -x_3 & -x_4 & = 0 \\ & -4x_2 & +6x_3 & -3x_4 & = 1. \end{array}$$

La matrice dei coefficienti e il vettore dei termini noti sono diventati:

$$A^{(2)} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & -1 & -2 & 1 \\ 0 & -4 & -1 & -1 \\ 0 & -4 & 6 & -3 \end{bmatrix}, \qquad \boldsymbol{b}^{(2)} = \begin{bmatrix} -1 \\ -2 \\ 0 \\ 1 \end{bmatrix}.$$

Cerchiamo ora di azzerare gli elementi sottodiagonali della seconda colonna, a partire da $a_{32}$, usando una tecnica simile. Innanzitutto osserviamo che non conviene prendere in considerazione una combinazione lineare che coinvolga la prima equazione perchè avendo questa un elemento in prima posizione diverso da zero quando sommata alla terza equazione cancellerà l'elemento uguale a zero in prima posizione. Lasciamo inalterate le prime due equazioni del sistema e prendiamo come equazione di riferimento la seconda. Poichè $a_{22}^{(2)} \neq 0$ poniamo

$$l_{32} = -\frac{a_{32}^{(2)}}{a_{22}^{(2)}} = -\frac{-4}{-1} = -4$$

e moltiplichiamo la seconda equazione per $l_{32}$ ottenendo:

$$4x_2 + 8x_3 - 4x_4 = 8.$$

La nuova terza equazione sarà la somma tra la terza equazione e la seconda appena modificata:

$$\begin{array}{rrrl} -4x_2 & -x_3 & -x_4 & = 0 \\ 4x_2 & +8x_3 & -4x_4 & = 8 \\ \hline & 7x_3 & -5x_4 & = 8 \end{array} \quad [\texttt{Nuova terza equazione}].$$

Poniamo

$$l_{42} = -\frac{a_{42}^{(2)}}{a_{22}^{(2)}} = -\frac{-4}{-1} = -4$$

e moltiplichiamo la seconda equazione per $l_{42}$ ottenendo:

$$4x_2 + 8x_3 - 4x_4 = 8.$$

La nuova quarta equazione sarà la somma tra la quarta equazione e la seconda appena modificata:

$$\begin{array}{rrrll} -4x_2 & +6x_3 & -3x_4 & = 1 & \\ 4x_2 & +8x_3 & -4x_4 & = 8 & \\ \hline & 14x_3 & -7x_4 & = 9 & [\texttt{Nuova quarta equazione}]. \end{array}$$

Al terzo passo il sistema lineare è diventato:

$$\begin{array}{rrrrl} 2x_1 & +x_2 & +x_3 & & = -1 \\ & -x_2 & -2x_3 & +x_4 & = -2 \\ & & 7x_3 & -5x_4 & = 8 \\ & & 14x_3 & -7x_4 & = 9. \end{array}$$

La matrice dei coefficienti e il vettore dei termini noti sono quindi

$$A^{(3)} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & -1 & -2 & 1 \\ 0 & 0 & 7 & -5 \\ 0 & 0 & 14 & -7 \end{bmatrix}, \qquad \boldsymbol{b}^{(2)} = \begin{bmatrix} -1 \\ -2 \\ 8 \\ 9 \end{bmatrix}.$$

Resta da azzerare l'unico elemento sottodiagonali della terza colonna. Lasciamo inalterate le prime tre equazioni del sistema. Poniamo

$$l_{43} = -\frac{a_{43}^{(3)}}{a_{33}^{(3)}} = -\frac{14}{7} = -2$$

e moltiplichiamo la terza equazione per $l_{43}$ ottenendo:

$$-14x_3 + 10x_4 = -16.$$

La nuova quarta equazione sarà la somma tra la quarta equazione e la terza appena modificata:

$$\begin{array}{rrrll} 14x_3 & -7x_4 & = -16 & & \\ -14x_3 & +10x_4 & = 9 & & \\ \hline & 3x_4 & = -7 & [\texttt{Nuova quarta equazione}]. \end{array}$$

Abbiamo ottenuto un sistema triangolare superiore:

$$\begin{array}{rrrrl} 2x_1 & +x_2 & +x_3 & & = -1 \\ & -x_2 & -2x_3 & +x_4 & = 4 \\ & & 7x_3 & -5x_4 & = 8 \\ & & & 3x_4 & = -7. \end{array}$$

La matrice dei coefficienti e il vettore dei termini noti sono diventati:

$$A^{(4)} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 0 & -1 & -2 & 1 \\ 0 & 0 & 7 & -5 \\ 0 & 0 & 0 & 3 \end{bmatrix}, \qquad \boldsymbol{b}^{(4)} = \begin{bmatrix} -1 \\ 4 \\ 8 \\ -7 \end{bmatrix}.$$

Cerchiamo ora di ricavare le formule di trasformazione del metodo di eliminazione di Gauss per rendere un generico sistema di ordine $n$ in forma triangolare superiore.
Consideriamo il sistema di equazioni nella sua forma scalare (3.2):

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \qquad i = 1, \ldots, n. \tag{3.8}$$

Poichè il procedimento richiede un certo numero di passi indichiamo con $a_{ij}^{(1)}$ e $b_i^{(1)}$ gli elementi della matrice dei coefficienti e del vettore dei termini noti del sistema di partenza. Isoliamo in ogni equazione la componente $x_1$. Abbiamo:

$$a_{11}^{(1)} x_1 + \sum_{j=2}^{n} a_{1j}^{(1)} x_j = b_1^{(1)} \tag{3.9}$$

$$a_{i1}^{(1)} x_1 + \sum_{j=2}^{n} a_{ij}^{(1)} x_j = b_i^{(1)}, \qquad i = 2, \ldots, n. \tag{3.10}$$

Moltiplicando l'equazione (3.9) per $-a_{i1}^{(1)}/a_{11}^{(1)}, i = 2, \ldots, n$, si ottengono le seguenti $n-1$ equazioni:

$$-a_{i1}^{(1)} x_1 + \sum_{j=2}^{n} \left( -\frac{a_{i1}^{(1)}}{a_{11}^{(1)}} a_{1j}^{(1)} \right) x_j = -\frac{a_{i1}^{(1)}}{a_{11}^{(1)}} b_i^{(1)}, \qquad i = 2, \ldots, n. \tag{3.11}$$

Sommando alle equazioni (3.10) le (3.11) si ricavano $n-1$ nuove equazioni:

$$\sum_{j=2}^{n}\left(a_{ij}^{(1)}-\frac{a_{i1}^{(1)}}{a_{11}^{(1)}}a_{1j}^{(1)}\right)x_j=b_i^{(1)}-\frac{a_{i1}^{(1)}}{a_{11}^{(1)}}b_1^{(1)},\qquad i=2,\ldots,n. \tag{3.12}$$

L'equazione (3.9) insieme alle (3.12) formano un nuovo sistema di equazioni, equivalente a quello originario, che possiamo scrivere nel seguente modo:

$$\begin{cases} a_{11}^{(1)}x_1+\displaystyle\sum_{j=2}^{n}a_{1j}^{(1)}x_j=b_1^{(1)} \\ \\ \displaystyle\sum_{j=2}^{n}a_{ij}^{(2)}x_j=b_i^{(2)} \qquad i=2,\ldots,n \end{cases} \tag{3.13}$$

dove

$$\begin{cases} a_{ij}^{(2)}= \quad a_{ij}^{(1)}-\dfrac{a_{i1}^{(1)}}{a_{11}^{(1)}}a_{1j}^{(1)} \qquad i,j=2,\ldots,n \\ \\ b_i^{(2)}= \quad b_i^{(1)}-\dfrac{a_{i1}^{(1)}}{a_{11}^{(1)}}b_1^{(1)} \qquad i=2,\ldots,n. \end{cases} \tag{3.14}$$

Osserviamo che la matrice dei coefficienti del sistema (3.13) è la seguente

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

Ora a partire dal sistema di equazioni

$$\sum_{j=2}^{n}a_{ij}^{(2)}x_j=b_i^{(2)}\qquad i=2,\ldots,n,$$

ripetiamo i passi fatti precedentemente:

$$a_{22}^{(2)}x_2+\sum_{j=3}^{n}a_{2j}^{(2)}x_j=b_2^{(2)} \tag{3.15}$$

$$a_{i2}^{(2)} x_2 + \sum_{j=3}^{n} a_{ij}^{(2)} x_j = b_i^{(2)}, \qquad i = 3, \ldots, n. \tag{3.16}$$

Moltiplicando l'equazione (3.15) per $-a_{i2}^{(2)}/a_{22}^{(2)}$, per $i = 3, \ldots, n$, si ottiene

$$a_{i2}^{(2)} x_2 + \sum_{j=3}^{n} \left( -\frac{a_{i2}^{(2)}}{a_{22}^{(2)}} a_{2j}^{(2)} \right) x_j = -\frac{a_{i2}^{(2)}}{a_{22}^{(2)}} b_2^{(2)}, \qquad i = 3, \ldots, n. \tag{3.17}$$

Sommando le equazioni (3.17) alle (3.16) si ottengono $n-2$ nuove equazioni:

$$\sum_{j=3}^{n} \left( a_{ij}^{(2)} - \frac{a_{i2}^{(2)}}{a_{22}^{(2)}} a_{2j}^{(2)} \right) x_j = b_i^{(2)} - \frac{a_{i2}^{(2)}}{a_{22}^{(2)}} b_2^{(2)}, \qquad i = 3, \ldots, n \tag{3.18}$$

che possiamo scrivere in forma più compatta:

$$\sum_{j=3}^{n} a_{ij}^{(3)} x_j = b_i^{(3)} \qquad i = 3, \ldots, n$$

dove

$$\begin{cases} a_{ij}^{(3)} = a_{ij}^{(2)} - \dfrac{a_{i2}^{(2)}}{a_{22}^{(2)}} a_{2j}^{(2)} & i, j = 3, \ldots, n \\ \\ b_i^{(3)} = b_i^{(2)} - \dfrac{a_{i2}^{(2)}}{a_{22}^{(2)}} b_2^{(2)} & i = 3, \ldots, n. \end{cases}$$

Abbiamo il nuovo sistema equivalente:

$$\begin{cases} \displaystyle\sum_{j=1}^{n} a_{1j}^{(1)} x_j = b_1^{(1)} \\ \\ \displaystyle\sum_{j=2}^{n} a_{2j}^{(2)} x_j = b_2^{(2)} \\ \\ \displaystyle\sum_{j=3}^{n} a_{ij}^{(3)} x_j = b_i^{(3)} \qquad i = 3, \ldots, n. \end{cases}$$

Osserviamo che in questo caso la matrice dei coefficienti è

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

È evidente ora che dopo $n-1$ passi di questo tipo arriveremo ad un sistema equivalente a quello di partenza avente la forma:

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & \dots & a_{2n}^{(2)} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & a_{n-1\ n-1}^{(n-1)} & a_{n-1\ n}^{(n-1)} \\ 0 & 0 & \dots & 0 & a_{nn}^{(n)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_{n-1}^{(n-1)} \\ b_n^{(n)} \end{bmatrix}$$

la cui soluzione, come abbiamo visto, si ottiene facilmente, e dove le formule di trasformazione al passo $k$ sono:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)} \qquad i,j = k+1,\dots,n \tag{3.19}$$

e

$$b_i^{(k+1)} = b_i^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} b_k^{(k)} \qquad i = k+1,\dots,n. \tag{3.20}$$

Soffermiamoci ora un momento sul primo passo del procedimento. Osserviamo che per ottenere il $1^o$ sistema equivalente abbiamo operato le seguenti fasi:

1. moltiplicazione della prima riga della matrice dei coefficienti (e del corrispondente elemento del termine noto) per un opportuno scalare;

2. sottrazione dalla riga $i$-esima di $A$ della prima riga modificata dopo il passo 1.

Il valore di $k$ varia da 1 (matrice dei coefficienti e vettori dei termini noti iniziali) fino a $n-1$, infatti la matrice $A^{(n)}$ avrà gli elementi sottodiagonali

delle prime $n-1$ colonne uguali a zero.
Si può osservare che il metodo di eliminazione di Gauss ha successo se tutti gli elementi $a_{kk}^{(k)}$ sono diversi da zero, che sono detti elementi pivotali.
Un proprietà importante delle matrici $A^{(k)}$ è il fatto che le operazioni effettuate non alterano il determinante della matrice, quindi

$$\det A^{(k)} = \det A,$$

per ogni $k$. Poichè la matrice $A^{(n)}$ è triangolare superiore allora il suo determinante può essere calcolato esplicitamente

$$\det A^{(k)} = \prod_{k=1}^{n} a_{kk}^{(k)}.$$

Quello appena descritto è un modo, alternativo alla regola di Laplace per calcolare il determinante della matrice $A$.

**Esempio 3.3.1** *Calcolare il determinante della matrice*

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

*utlizzando il metodo di eliminazione di Gauss.*

Posto $A^{(1)} = A$, calcoliamo i tre moltiplicatori

$$l_{2,1} = -1, \qquad l_{3,1} = 0, \qquad l_{4,1} = -\frac{1}{3}.$$

Calcoliamo la seconda riga:

| | | | | | |
|---|---|---|---|---|---|
| $[2^a$ `riga di` $A^{(1)}$ `+` $]$ | 3 | 2 | 6 | $-1$ | $+$ |
| $[(-1)\times 1^a$ `riga di` $A^{(1)}]$ | $-3$ | $-3$ | $-5$ | 0 | $=$ |
| $[2^a$ `riga di` $A^{(2)}]$ | 0 | $-1$ | 1 | $-1$ | |

La terza riga non cambia perchè il moltiplicatore è nullo, mentre la quarta riga è

| | | | | | |
|---|---|---|---|---|---|
| $[4^a$ `riga di` $A^{(1)}$ `+` $]$ | 1 | 3 | 0 | 4 | $+$ |
| $[(-1/3)\times 1^a$ `riga di` $A^{(1)}]$ | $-1$ | $-1$ | $-5/3$ | 0 | $=$ |
| $[4^a$ `riga di` $A^{(2)}]$ | 0 | 2 | $-5/3$ | 4 | |

Abbiamo ottenuto la seguente matrice al passo 2:

$$A^{(2)} = \begin{bmatrix} 3 & 3 & 5 & 0 \\ 0 & -1 & 1 & -1 \\ 0 & 2 & 0 & 4 \\ 0 & 2 & -5/3 & 4 \end{bmatrix}.$$

Calcoliamo i due moltiplicatori

$$l_{3,2} = 2, \qquad l_{4,2} = 2.$$

Calcoliamo la terza riga:

| | | | | | |
|---|---|---|---|---|---|
| $[3^a$ `riga di` $A^{(2)}$ `+` $]$ | $0$ | $2$ | $0$ | $4$ | $+$ |
| $[(2)\times$ $2^a$ `riga di` $A^{(2)}]$ | $0$ | $-2$ | $2$ | $-2$ | $=$ |
| $[3^a$ `riga di` $A^{(3)}]$ | $0$ | $0$ | $2$ | $2$ | |

La quarta riga è

| | | | | | |
|---|---|---|---|---|---|
| $[4^a$ `riga di` $A^{(2)}$ `+` $]$ | $0$ | $2$ | $-5/3$ | $4$ | $+$ |
| $[(2)\times$ $2^a$ `riga di` $A^{(2)}]$ | $0$ | $-2$ | $2$ | $-2$ | $=$ |
| $[4^a$ `riga di` $A^{(3)}]$ | $0$ | $0$ | $1/3$ | $2$ | |

Abbiamo ottenuto la seguente matrice al passo 3:

$$A^{(3)} = \begin{bmatrix} 3 & 3 & 5 & 0 \\ 0 & -1 & 1 & -1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 1/3 & 2 \end{bmatrix}.$$

Calcoliamo l'unico moltiplicatore del terzo passo:

$$l_{4,3} = -\frac{1}{6}.$$

La quarta riga è

| | | | | | |
|---|---|---|---|---|---|
| $[4^a$ `riga di` $A^{(3)}$ `+` $]$ | $0$ | $0$ | $1/3$ | $2$ | $+$ |
| $[(-1/6)\times$ $3^a$ `riga di` $A^{(3)}]$ | $0$ | $0$ | $-1/3$ | $-1/3$ | $=$ |
| $[4^a$ `riga di` $A^{(4)}]$ | $0$ | $0$ | $0$ | $5/3$ | |

La matrice triagolarizzata è

$$A^{(4)} = \begin{bmatrix} 3 & 3 & 5 & 0 \\ 0 & -1 & 1 & -1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 5/3 \end{bmatrix}.$$

Il determinante della matrice è uguale al prodotto degli elementi diagonali della matrice triangolare, ovvero

$$\det A = -10.$$

**Esempio 3.3.2** *Calcolare l'inversa della matrice*

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

*utlizzando il metodo di eliminazione di Gauss.*

L'inversa di $A$ è la matrice $X$ tale che

$$AX = I$$

ovvero, detta $\boldsymbol{x}_i$ la $i-$esima colonna di $X$, questo è soluzione del sistema lineare

$$A\boldsymbol{x}_i = \boldsymbol{e}_i \tag{3.21}$$

dove $\boldsymbol{e}_i$ è l'$i-$esimo versore della base canonica di $\mathbb{R}^n$. Posto $i = 1$ risolvendo il sistema

$$A\boldsymbol{x}_1 = \boldsymbol{e}_1, \qquad \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 1 & 2 & 0 \\ -1 & 0 & 3 & 1 \\ 1 & 1 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

si ottengono gli elementi della prima colonna di $A^{-1}$. Posto $A^{(1)} = A$ gli elementi della matrice al passo 2 sono calcolati applicando le formule

$$a_{ij}^{(2)} = a_{ij}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} a_{1j}^{(1)}, \qquad i, j = 2, 3, 4.$$

Tralasciando il dettaglio delle operazioni risulta

$$A^{(2)} = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 0 & 1/2 & 2 & -1/2 \\ 0 & 1/2 & 3 & 3/2 \\ 0 & 1/2 & 2 & 3/2 \end{bmatrix}, \qquad \boldsymbol{e}_1^{(2)} = \begin{bmatrix} 1 \\ -1/2 \\ 1/2 \\ -1/2 \end{bmatrix}$$

Applicando le formula

$$a_{ij}^{(3)} = a_{ij}^{(2)} - \frac{a_{i2}^{(2)}}{a_{22}^{(2)}} a_{2j}^{(2)}, \qquad i, j = 3, 4.$$

si ottiene il sistema al terzo passo

$$A^{(3)} = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 0 & 1/2 & 2 & -1/2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \qquad \boldsymbol{e}_1^{(3)} = \begin{bmatrix} 1 \\ -1/2 \\ 1 \\ 0 \end{bmatrix}.$$

In questo caso non è necessario applicare l'ultimo passo del metodo in quanto la matrice è già triangolare superiore e pertanto si può risolvere il sistema triangolare superiore ottenendo:

$$x_4 = 0, \quad x_3 = 1, \quad x_2 = -5, \quad x_1 = 3.$$

Cambiando i termini noti del sistema (3.21), ponendo $i = 2, 3, 4$ si ottengono le altre tre colonne della matrice inversa.

### 3.3.1 Costo Computazionale del Metodo di Eliminazione di Gauss

Cerchiamo ora di determinare il costo computazionale (cioè il numero di operazioni aritmetiche) richiesto dal metodo di eliminazione di Gauss per risolvere un sistema lineare di ordine $n$. Il calcolo del costo computazionale richiede quattro fasi:

1. Numero di operazioni aritmetiche necessarie per modificare un singolo elemento della matrice dei coefficienti e del vettore dei termini noti;

2. Numero di operazioni aritmetiche necessarie per calcolare la matrice $A^{(k+1)}$ ed il vettore $\boldsymbol{b}^{(k+1)}$ partendo da $A^{(k)}$ e $\boldsymbol{b}^{(k)}$, con $k$ valore generico;

3. Numero di operazioni aritmetiche richiesto per effettuare tutte gli $n-1$ passi del metodo;

4. Numero di operazioni aritmetiche richiesto dalla risoluzione del sistema triangolare superiore.

Di tali fasi solo per l'ultima sappiamo che esso è pari a $n^2$.
Per la prima fase dalle relazioni

$$b_i^{(k+1)} = b_i^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} b_k^{(k)}, \qquad i = k+1, \ldots, n,$$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)}, \qquad i, j = k+1, \ldots, n$$

è evidente che servono 3 operazioni aritmetiche per calcolare $b_i^{(k+1)}$ (noti $a_{ij}^{(k)}$ e $b_i^{(k)}$) mentre sono necessarie che solo 2 operazioni per calcolare $a_{ij}^{(k+1)}$ (noti $a_{ij}^{(k)}$ e $b_i^{(k)}$), infatti il moltiplicatore viene calcolato solo una volta.
Per determinare il numero richiesto dalla seconda fase esso è pari a:
$3\times$ **elementi del vettore calcolati** $+2\times$ **elementi della matrice calcolati**.
Il numero di elementi del vettore dei termini noti che vengono modificati è pari ad $n-k$ mentre gli elementi della matrice cambiati sono $(n-k)^2$ quindi complessivamente il numero di operazioni per calcolare gli elementi al passo $k+1$ è:

$$2(n-k)^2 + 3(n-k). \tag{3.22}$$

Osserviamo che nel computo del numero di elementi della matrice che vengono calcolati non si tiene conto degli elementi che sono stati azzerati, in quanto è noto che sono uguali a zero non cè alcuna necessità di calcolarli.
Per trasformare $A$ in $A^{(n)}$ e $\boldsymbol{b}$ in $\boldsymbol{b}^{(n)}$ è necessario un numero di operazioni pari alla somma, rispetto a $k$, di (3.22), ovvero

$$f(n) = 2\sum_{k=1}^{n-1}(n-k)^2 + 3\sum_{k=1}^{n-1}(n-k).$$

Sapendo che

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

ed effettuando un opportuno cambio di indice nelle sommatorie risulta

$$f(n) = 2\left[\frac{n(n-1)(2n-1)}{6}\right] + 3\frac{n(n-1)}{2} = \frac{2}{3}n^3 + \frac{n^2}{2} - \frac{7}{6}n.$$

Nel calcolo del costo computazionale di un algoritmo si tende a considerare solo la componente più grande tralasciando quelle che contribuiscono meno a tale valore, pertanto si ha

$$f(n) \simeq \frac{2}{3}n^3.$$

A questo valore bisognerebbe aggiungere le $n^2$ operazioni aritmetiche necessarie per risolvere il sistema triangolare superiore ma tale valore non altera l'ordine di grandezza della funzione che è un valore molto inferiore rispetto alle $n!$ operazioni richieste dalla regola di Cramer, applicata insieme alla regola di Laplace.
Nel calcolo delle operazioni aritmetiche sono state conseiderate tutte le 4 operazioni aritmetiche, ipotizzando implicitamente che esse richiedano lo stesso tempo di esecuzione da parte dall'elebaoratore (ottenendo una stima del tempo di risoluzione richiesto dal metodo). Nella realtà non è così in quanto le somme algebriche richiedono un tempo inferiore rispetto al prodotto ed al quoziente e pertanto il numero di tali operazioni andrebbe contato a parte. Facendo questo tipo di calcolo si scoprirebbe che il numero di moltiplicazioni/divisioni richiesto dal metodo è circa la metà di quello trovato:

$$f_1(n) \simeq \frac{n^3}{3}.$$

### 3.3.2 Strategie di Pivoting per il metodo di Gauss

Nell'eseguire il metodo di Gauss si è fatta l'implicita ipotesi (vedi formule (3.19) e (3.20)) che gli elementi pivotali $a_{kk}^{(k)}$ siano non nulli per ogni $k$. Tale situazione si verifica quando i minori principali di testa di ordine di $A$ sono diversi da zero. Infatti vale il seguente risultato.

**Teorema 3.3.1** *Se $A \in \mathbb{R}^{n\times n}$ , indicata con $A_k$ la matrice principale di testa di ordine $k$, risulta*

$$a_{kk}^{(k)} = \frac{\det A_k}{\det A_{k-1}}, \qquad k = 1, \ldots, n$$

*avendo posto per convenzione* $\det A_0 = 1$.

In pratica questa non è un'ipotesi limitante in quanto la non singolarità di $A$ permette, con un opportuno scambio di righe in $A^{(k)}$, di ricondursi a questo caso. Infatti scambiare due righe in $A^{(k)}$ significa sostanzialmente scambiare due equazioni nel sistema $A^{(k)}\boldsymbol{x} = \boldsymbol{b}^{(k)}$ e ciò non altera la natura del sistema stesso.
Consideriamo la matrice $A^{(k)}$ e supponiamo $a_{kk}^{(k)} = 0$. In questo caso possiamo scegliere un elemento sottodiagonale appartenente alla $k-$esima colonna diverso da zero, supponiamo $a_{ik}^{(k)}$, scambiare le equazioni di indice $i$ e $k$ e continuare il procedimento perchè in questo modo l'elemento pivotale è diverso da zero. In ipotesi di non singolarità della matrice $A$ possiamo dimostrare tale elemento diverso da zero esiste sicuramente. Infatti supponendo che, oltra all'elemento pivotale, siano nulli tutti gli $a_{ik}^{(k)}$ per $i = k+1, \dots, n$, allora $A^{(k)}$ ha la seguente struttura:

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & \dots & a_{1,k-1}^{(1)} & a_{1k}^{(1)} & a_{1,k+1}^{(1)} & \dots & a_{1n}^{(1)} \\ & \ddots & \vdots & \vdots & \vdots & & \vdots \\ & & a_{k-1,k-1}^{(k-1)} & a_{k-1,k}^{(k-1)} & a_{k-1,k+1}^{(k-1)} & \dots & a_{k-1,n}^{(k-1)} \\ & & & 0 & a_{k,k+1}^{(k)} & & a_{kn}^{(k)} \\ & 0 & & \vdots & \vdots & & \vdots \\ & & & 0 & a_{n,k+1}^{(k)} & \dots & a_{nn}^{(k)}. \end{bmatrix}$$

Se partizioniamo $A^{(k)}$ nel seguente modo

$$A^{(k)} = \begin{bmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{bmatrix}$$

con $A_{11}^{(k)} \in \mathbb{R}^{(k-1)\times(k-1)}$ allora il determinante di $A^{(k)}$ è

$$\det A^{(k)} = \det A_{11}^{(k)} \det A_{22}^{(k)} = 0$$

perchè la matrice $A_{22}^{(k)}$ ha una colonna nulla. Poichè tutte la matrici $A^{(k)}$ hanno lo stesso determinante di $A$, dovrebbe essere $\det A = 0$ e questo contrasta con l'ipotesi fatta. Possiamo concludere che se $a_{kk}^{(k)} = 0$ e $\det A \neq 0$ deve necessariamente esistere un elemento $a_{ik}^{(k)} \neq 0$, con $i \in \{k+1, k+2, \dots, n\}$. Per evitare che un elemento pivotale possa essere uguale a zero si applica una delle cosiddette strategie di pivoting. La strategia di Pivoting parziale prevede che prima di fare ciò si ricerchi l'elemento di massimo modulo tra

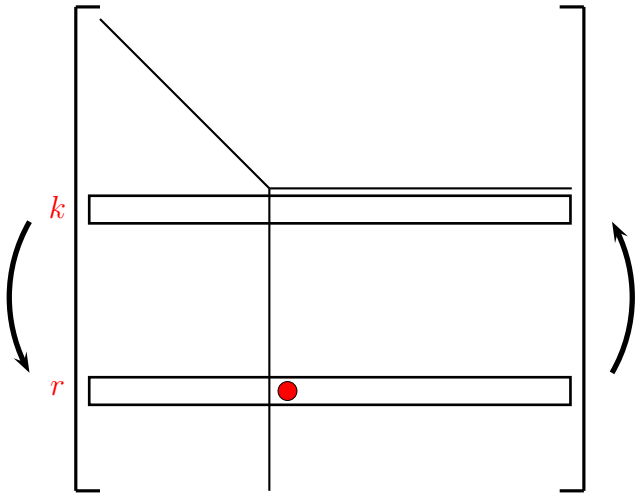



Figura 3.1: Strategia di pivoting parziale.

gli elementi $a_{kk}^{(k)}, a_{k+1,k}^{(k)}, \ldots, a_{nk}^{(k)}$ e si scambi l'equazione in cui si trova questo elemento con la $k$-esima qualora esso sia diverso da $a_{kk}^{(k)}$. In altri termini il pivoting parziale richiede le seguenti operazioni:

1. determinare l'elemento $a_{rk}^{(k)}$ tale che
$$|a_{rk}^{(k)}| = \max_{k \le i \le n} |a_{ik}^{(k)}|;$$
2. effettuare lo scambio tra le equazioni del sistema di indice $r$ e $k$.

in alternativa si può adottare la strategia di pivoting totale che è la seguente:

1. determinare gli indici $r, s$ tali che
$$|a_{rs}^{(k)}| = \max_{k \le i,j \le n} |a_{ij}^{(k)}|;$$
2. effettuare lo scambio tra le equazioni del sistema di indice $r$ e $k$.
3. effettuare lo scambio tra le colonne di indice $s$ e $k$ della matrice dei coefficienti.

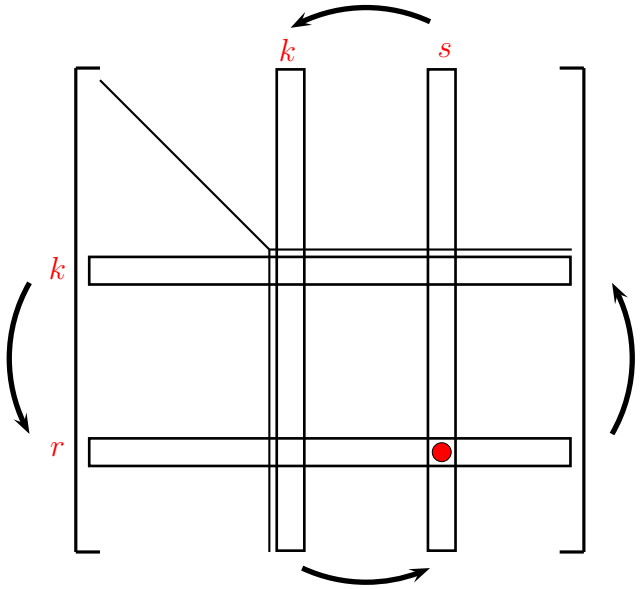



Figura 3.2: strategia di pivoting totale.

La strategia di pivoting totale è senz'altro migliore perchè garantisce maggiormente che un elemento pivotale non sia un numero piccolo (in questa eventualità potrebbe accadere che un moltiplicatore sia un numero molto grande) ma richiede che tutti gli eventuali scambi tra le colonne della matrice siano memorizzati. Infatti scambiare due colonne significa scambiare due incognite del vettore soluzione pertanto dopo la risoluzione del sistema triangolare per ottenere il vettore soluzione del sistema di partenza è opportuno permutare le componenti che sono state scambiate.

**Esempio 3.3.3** *Risolvere il sistema lineare* $A\boldsymbol{x} = \boldsymbol{b}$ *dove*

$$A = \begin{bmatrix} 1 & 2 & -1 & 0 \\ 2 & -1 & -1 & 1 \\ 3 & 0 & -1 & 1 \\ 1 & -3 & 1 & 1 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 2 \\ 1 \\ 4 \\ 2 \end{bmatrix}$$

*utlizzando il metodo di eliminazione di Gauss con strategia di pivoting parziale.*

Posto $A^{(1)} = A$, osserviamo che l'elemento pivotale della prima colonna si trova sulla terza riga allora scambiamo per equazioni 1 e 3:

$$A^{(1)} = \begin{bmatrix} 3 & 0 & -1 & 1 \\ 2 & -1 & -1 & 1 \\ 1 & 2 & -1 & 0 \\ 1 & -3 & 1 & 1 \end{bmatrix}, \qquad \boldsymbol{b}^{(1)} = \begin{bmatrix} 4 \\ 1 \\ 2 \\ 2 \end{bmatrix}$$

calcoliamo i tre moltiplicatori

$$l_{2,1} = -\frac{2}{3}, \qquad l_{3,1} = -\frac{1}{3}, \qquad l_{4,1} = -\frac{1}{3}.$$

Calcoliamo la seconda riga:

| | | | | | | |
|---|---|---|---|---|---|---|
| $[2^a$ `riga di` $A^{(1)}$ `+` $]$ | 2 | $-1$ | $-1$ | 1 | 1 | $+$ |
| $[(-2/3)\times 1^a$ `riga di` $A^{(1)}]$ | $-2$ | 0 | $2/3$ | $-2/3$ | $-8/3$ | $=$ |
| $[2^a$ `riga di` $A^{(2)}]$ | 0 | $-1$ | $-1/3$ | $1/3$ | $-5/3$ | |

La terza riga è la seguente:

| | | | | | | |
|---|---|---|---|---|---|---|
| $[3^a$ `riga di` $A^{(1)}$ `+` $]$ | 1 | 2 | $-1$ | 0 | 2 | $+$ |
| $[(-1/3)\times 1^a$ `riga di` $A^{(1)}]$ | $-1$ | 0 | $1/3$ | $-1/3$ | $-4/3$ | $=$ |
| $[3^a$ `riga di` $A^{(2)}]$ | 0 | 2 | $-2/3$ | $-1/3$ | $2/3$ | |

mentre la quarta riga è

| | | | | | | |
|---|---|---|---|---|---|---|
| $[4^a$ `riga di` $A^{(1)}$ `+` $]$ | 1 | $-3$ | 1 | 1 | 2 | $+$ |
| $[(-1/3)\times 1^a$ `riga di` $A^{(1)}]$ | $-1$ | 0 | $1/3$ | $-1/3$ | $-4/3$ | $=$ |
| $[4^a$ `riga di` $A^{(2)}]$ | 0 | $-3$ | $4/3$ | $2/3$ | $2/3$ | |

Abbiamo ottenuto la matrice ed il vettore al passo 2:

$$A^{(2)} = \begin{bmatrix} 3 & 0 & -1 & 1 \\ 0 & -1 & -1/3 & 1/3 \\ 0 & 2 & -2/3 & -1/3 \\ 0 & -3 & 4/3 & 2/3 \end{bmatrix}, \qquad \boldsymbol{b}^{(2)} = \begin{bmatrix} 4 \\ -5/3 \\ 2/3 \\ 2/3 \end{bmatrix}.$$

L'elemento pivotale della seconda colonna si trova sulla quarta riga quindi scambiamo le equazioni 2 e 4:

$$A^{(2)} = \begin{bmatrix} 3 & 0 & -1 & 1 \\ 0 & -3 & 4/3 & 2/3 \\ 0 & 2 & -2/3 & -1/3 \\ 0 & -1 & -1/3 & 1/3 \end{bmatrix}, \qquad \boldsymbol{b}^{(2)} = \begin{bmatrix} 4 \\ 2/3 \\ 2/3 \\ -5/3 \end{bmatrix}.$$

Calcoliamo i due moltiplicatori

$$l_{3,2} = \frac{2}{3}, \qquad l_{4,2} = -\frac{1}{3}.$$

La terza riga è la seguente:

| | | | | | | |
|---|---|---|---|---|---|---|
| $[3^a$ `riga di` $A^{(2)}$ `+` $]$ | $0$ | $2$ | $-2/3$ | $-1/3$ | $2/3$ | $+$ |
| $[(2/3)\times 2^a$ `riga di` $A^{(2)}]$ | $0$ | $-2$ | $8/9$ | $4/9$ | $4/9$ | $=$ |
| $[3^a$ `riga di` $A^{(3)}]$ | $0$ | $0$ | $2/9$ | $1/9$ | $10/9$ | |

mentre la quarta riga è

| | | | | | | |
|---|---|---|---|---|---|---|
| $[4^a$ `riga di` $A^{(2)}$ `+` $]$ | $0$ | $-1$ | $-1/3$ | $1/3$ | $-5/3$ | $+$ |
| $[(-1/3)\times 2^a$ `riga di` $A^{(2)}]$ | $0$ | $1$ | $-4/9$ | $-2/9$ | $-2/9$ | $=$ |
| $[4^a$ `riga di` $A^{(3)}]$ | $0$ | $0$ | $-7/9$ | $1/9$ | $-17/9$ | |

Abbiamo ottenuto la matrice ed il vettore al passo 3:

$$A^{(3)} = \begin{bmatrix} 3 & 0 & -1 & 1 \\ 0 & -3 & 4/3 & 2/3 \\ 0 & 0 & 2/9 & 1/9 \\ 0 & 0 & -7/9 & 1/9 \end{bmatrix}, \qquad \boldsymbol{b}^{(3)} = \begin{bmatrix} 4 \\ 2/3 \\ 10/9 \\ -17/9 \end{bmatrix}.$$

L'elemento pivotale della terza colonna si trova sulla quarta riga quindi scambiamo le equazioni 3 e 4:

$$A^{(3)} = \begin{bmatrix} 3 & 0 & -1 & 1 \\ 0 & -3 & 4/3 & 2/3 \\ 0 & 0 & -7/9 & 1/9 \\ 0 & 0 & 2/9 & 1/9 \end{bmatrix}, \qquad \boldsymbol{b}^{(3)} = \begin{bmatrix} 4 \\ 2/3 \\ -17/9 \\ 10/9 \end{bmatrix}.$$

Calcoliamo l'unico moltiplicatore del terzo passo:

$$l_{4,3} = \frac{2}{7}.$$

La quarta riga è

| | | | | | | |
|---|---|---|---|---|---|---|
| $[4^a$ `riga di` $A^{(3)}$ `+` $]$ | $0$ | $0$ | $2/9$ | $1/9$ | $10/9$ | $+$ |
| $[(2/7)\times 3^a$ `riga di` $A^{(3)}]$ | $0$ | $0$ | $-2/9$ | $2/63$ | $-34/63$ | $=$ |
| $[4^a$ `riga di` $A^{(4)}]$ | $0$ | $0$ | $0$ | $1/7$ | $4/7$ | |

Il sistema triagolare superiore equivalente a quello iniziale ha come matrice dei coefficienti e come termine noto:

$$A^{(3)} = \begin{bmatrix} 3 & 0 & -1 & 1 \\ 0 & -3 & 4/3 & 2/3 \\ 0 & 0 & -7/9 & 1/9 \\ 0 & 0 & 0 & 1/7 \end{bmatrix}, \qquad \boldsymbol{b}^{(3)} = \begin{bmatrix} 4 \\ 2/3 \\ -17/9 \\ 4/7 \end{bmatrix}.$$

Risolvendo tale sistema triangolare superiore si ricava il vettore:

$$x_4 = 4, \qquad x_3 = 3, \qquad x_2 = 2, \qquad x_1 = 1.$$

Nelle pagine seguenti sono riportati i codici MatLab che implementano il metodo di Gauss con entrambe le strategie di pivoting descritte.

```
function x=Gauss(A,b)
%
% Metodo di eliminazione di Gauss
%
% Parametri di input:
% A = Matrice dei coefficienti del sistema
% b = Vettore dei termini noti del sistema
%
% Parametri di input:
% x = Vettore soluzione del sistema lineare
%
n = length(b);
x = zeros(n,1);
for k=1:n-1
  if abs(A(k,k))<eps
    error('Elemento pivotale nullo ')
  end
  for i=k+1:n
    A(i,k) = A(i,k)/A(k,k);
    b(i) = b(i)-A(i,k)*b(k);
    for j=k+1:n
      A(i,j) = A(i,j)-A(i,k)*A(k,j);
    end
 end
```

```
end
x(n) = b(n)/A(n,n);
for i=n-1:-1:1
  x(i) = (b(i)-A(i,i+1:n)*x(i+1:n))/A(i,i);
end
return

function x=Gauss_pp(A,b)
%
% Metodo di Gauss con pivot parziale
%
% Parametri di input:
% A = Matrice dei coefficienti del sistema
% b = Vettore dei termini noti del sistema
%
% Parametri di input:
% x = Vettore soluzione del sistema lineare
%
n = length(b);
x = zeros(n,1);
for k=1:n-1
  [a,i] = max(abs(A(k:n,k)));
  i = i+k-1;
  if i~=k
    A([i k],:) = A([k i],:);
    b([i k]) = b([k i]);
  end
  for i=k+1:n
    A(i,k) = A(i,k)/A(k,k);
    b(i) = b(i)-A(i,k)*b(k);
    for j=k+1:n
      A(i,j) = A(i,j)-A(i,k)*A(k,j);
    end
 end
end
x(n) = b(n)/A(n,n);
for i=n-1:-1:1
  x(i) = (b(i)-A(i,i+1:n)*x(i+1:n))/A(i,i);
```

```
end
return

function x=Gauss_pt(A,b)
%
% Metodo di Gauss con pivot totale
%
% Parametri di input:
% A = Matrice dei coefficienti del sistema
% b = Vettore dei termini noti del sistema
%
% Parametri di input:
% x = Vettore soluzione del sistema lineare
%
n = length(b);
x = zeros(n,1);
x1 = x;
indice = [1:n];
for k=1:n-1
  [a,riga] = max(abs(A(k:n,k:n)));
  [mass,col] = max(a);
  j = col+k-1;
  i = riga(col)+k-1;
  if i~=k
  A([i k],:) = A([k i],:);
  b([i k]) = b([k i]);
  end
  if j~=k
  A(:,[j k]) = A(:,[k j]);
    indice([j k]) = indice([k j]);
  end
  for i=k+1:n
    A(i,k) = A(i,k)/A(k,k);
    b(i) = b(i)-A(i,k)*b(k);
    for j=k+1:n
      A(i,j) = A(i,j)-A(i,k)*A(k,j);
    end
  end
```

```
end
%
% Risoluzione del sistema triangolare superiore
%
x1(n) = b(n)/A(n,n);
for i=n-1:-1:1
  x1(i) = (b(i)-A(i,i+1:n)*x1(i+1:n))/A(i,i);
end
%
% Ripermutazione del vettore
%
for i=1:n
  x(indice(i))=x1(i);
end
return
```

### 3.3.3 La Fattorizzazione *LU*

#### Introduzione

Supponiamo di dover risolvere un problema che richieda, ad un determinato passo, la risoluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ e di utilizzare il metodo di Gauss. La matrice viene resa triangolare superiore e viene risolto il sistema triangolare

$$A^{(n)}\boldsymbol{x} = \boldsymbol{b}^{(n)}. \tag{3.23}$$

Ipotizziamo che, nell'ambito dello stesso problema, dopo un certo tempo sia necessario risolvere il sistema

$$A\boldsymbol{x} = \boldsymbol{c}$$

i cui la matrice dei coefficienti è la stessa mentre è cambiato il termine noto. Appare chiaro che non è possibile sfruttare i calcoli gia fatti in quanto il calcolo del vettore dei termini noti al passo $n$ dipende dalle matrici ai passi precedenti all'ultimo, quindi la conoscenza della matrice $A^{(n)}$ è del tutto inutile. È necessario pertanto applicare nuovamente il metodo di Gauss e risolvere il sistema triangolare

$$A^{(n)}\boldsymbol{x} = \boldsymbol{c}^{(n)}. \tag{3.24}$$

L'algoritmo che sarà descritto in questo paragrafo consentirà di evitare l'eventualità di dover rifare tutti i calcoli (o una parte di questi).

### Calcolo diretto della fattorizzazione $LU$

La Fattorizzazione $LU$ di una matrice stabilisce, sotto determinate ipotesi, l'esistenza di una matrice $L$ triangolare inferiore con elementi diagonali uguali a 1 e di una matrice triangolare superiore $U$ tali che $A = LU$.
Vediamo ora di determinare le formule esplicite per gli elementi delle due matrici. Fissata la matrice $A$, quadrata di ordine $n$, imponiamo quindi che risulti

$$A = LU.$$

Una volta note tali matrici il sistema di partenza $A\boldsymbol{x} = \boldsymbol{b}$ viene scritto come

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

e, posto $U\boldsymbol{x} = \boldsymbol{y}$, il vettore $\boldsymbol{x}$ viene trovato prima risolvendo il sistema triangolare inferiore

$$L\boldsymbol{y} = \boldsymbol{b}$$

e poi quello triangolare superiore

$$U\boldsymbol{x} = \boldsymbol{y}.$$

Imponiamo quindi che la matrice $A$ ammetta fattorizzazione $LU$:

$$\begin{bmatrix} a_{11} & \dots & a_{1j} & \dots & a_{1n} \\ \vdots & & \vdots & & \vdots \\ a_{i1} & \dots & a_{ij} & \dots & a_{in} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & \dots & a_{nj} & \dots & a_{nn} \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ l_{21} & 1 & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & 0 & & \vdots \\ l_{i1} & \dots & l_{i,i-1} & 1 & \ddots & \vdots \\ \vdots & & \vdots & & \ddots & 0 \\ l_{n1} & \dots & l_{n,i-1} & l_{n,i} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & \dots & \dots & u_{1j} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2j} & \dots & u_{2n} \\ \vdots & \ddots & \ddots & \vdots & & \vdots \\ \vdots & & \ddots & u_{jj} & \dots & u_{jn} \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & u_{nn} \end{bmatrix}.$$

Deve essere

$$a_{ij} = \sum_{k=1}^{n} l_{ik}u_{kj} = \sum_{k=1}^{\min(i,j)} l_{ik}u_{kj} \qquad i,j = 1,\dots,n. \tag{3.25}$$

Considerando prima il caso $i \leq j$, uguagliando quindi la parte triangolare superiore delle matrici abbiamo

$$a_{ij} = \sum_{k=1}^{i} l_{ik} u_{kj} \qquad j \geq i \tag{3.26}$$

ovvero

$$a_{ij} = \sum_{k=1}^{i-1} l_{ik} u_{kj} + l_{ii} u_{ij} = \sum_{k=1}^{i-1} l_{ik} u_{kj} + u_{ij} \qquad j \geq i$$

infine risulta

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \qquad j \geq i \tag{3.27}$$

e ovviamente $u_{1j} = a_{1j}$, per $j = 1, \ldots, n$. Considerando ora il caso $j < i$, uguagliando cioè le parti strettamente triangolari inferiori delle matrici risulta:

$$a_{ij} = \sum_{k=1}^{j} l_{ik} u_{kj} \qquad i > j \tag{3.28}$$

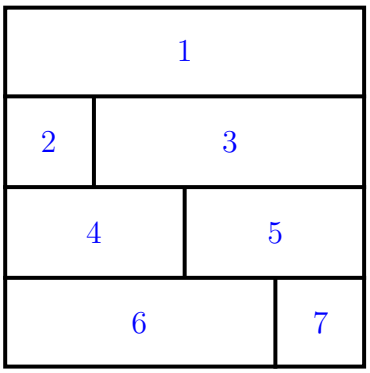
ovvero

$$a_{ij} = \sum_{k=1}^{j-1} l_{ik} u_{kj} + l_{ij} u_{jj} \qquad i > j$$

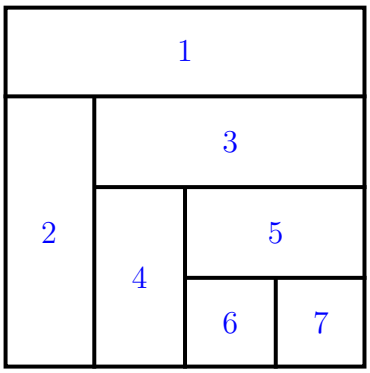
da cui

$$l_{ij} = \frac{1}{u_{jj}} \left( a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right) \qquad i > j. \tag{3.29}$$

Si osservi che le formule (3.27) e (3.29) vanno implementate secondo uno degli schemi riportati nella seguente figura.




**Tecnica di Crout**




**Tecnica di Doolittle**

Ogni schema rappresenta in modo schematico una matrice la cui parte triangolare superiore indica la matrice $U$ mentre quella triangolare inferiore la matrice $L$ mentre i numeri indicano l'ordine con cui gli elementi saranno calcolati. Per esempio applicando la tecnica di Crout si segue il seguente ordine:

- $1^o$ Passo: Calcolo della prima riga di $U$;
- $2^o$ Passo: Calcolo della seconda riga di $L$;
- $3^o$ Passo: Calcolo della seconda riga di $U$;
- $4^o$ Passo: Calcolo della terza riga di $L$;
- $5^o$ Passo: Calcolo della terza riga di $U$;
- $6^o$ Passo: Calcolo della quarta riga di $L$;
- $7^o$ Passo: Calcolo della quarta riga di $U$;

e così via procedendo per righe in modo alternato. Nel caso della tecnica di Doolittle si seguono i seguenti passi:

- $1^o$ Passo: Calcolo della prima riga di $U$;
- $2^o$ Passo: Calcolo della prima colonna di $L$;
- $3^o$ Passo: Calcolo della seconda riga di $U$;

- $4^o$ Passo: Calcolo della seconda colonna di $L$;
- $5^o$ Passo: Calcolo della terza riga di $U$;
- $6^o$ Passo: Calcolo della terza colonna di $L$;
- $7^o$ Passo: Calcolo della quarta riga di $U$.

La fattorizzazione $LU$ è un metodo sostanzialmente equivalente al metodo di Gauss, infatti la matrice $U$ che viene calcolata coincide con la matrice $A^{(n)}$. Lo svantaggio del metodo di fattorizzazione diretto risiede essenzialmente nella maggiore difficoltà, rispetto al metodo di Gauss, di poter programmare una strategia di pivot. Infatti se un elemento diagonale della matrice $U$ è uguale a zero non è possibile applicare l'algoritmo.

```
function [L,U]=crout(A);
%
% La funzione calcola la fattorizzazione LU della
% matrice A applicando la tecnica di Crout
%
% L = matrice triang. inferiore con elementi diagonali
%     uguali a 1
% U = matrice triangolare superiore
%
[m n] = size(A);
U = zeros(n);
L = eye(n);
U(1,:) = A(1,:);
for i=2:n
    for j=1:i-1
        L(i,j) = (A(i,j) - L(i,1:j-1)*U(1:j-1,j))/U(j,j);
    end
    for j=i:n
        U(i,j) = A(i,j) - L(i,1:i-1)*U(1:i-1,j);
    end
end
return

function [L,U]=doolittle(A);
```

```
%
% La funzione calcola la fattorizzazione LU della
% matrice A applicando la tecnica di Doolittle
%
% L = matrice triang. inferiore con elementi diagonali
%     uguali a 1
% U = matrice triangolare superiore
%
[m n] = size(A);
L = eye(n);
U = zeros(n);
U(1,:) = A(1,:);
for i=1:n-1
    for riga=i+1:n
        L(riga,i)=(A(riga,i)-L(riga,1:i-1)*U(1:i-1,i))/U(i,i);
    end
    for col=i+1:n
        U(i+1,col) = A(i+1,col)-L(i+1,1:i)*U(1:i,col);
    end
end
return
```

### Equivalenza tra metodo di Gauss e fattorizzazione $LU$

In questo paragrafo esplicitiamo la relazione di equivalenza che lega il metodo di eliminazione di Gauss (senza alcuna strategia di pivoting) e la fattorizzazione $LU$.
Supponiamo di dover risolvere il sistema

$$A\boldsymbol{x} = \boldsymbol{b} \qquad \Leftrightarrow \qquad A^{(1)}\boldsymbol{x} = \boldsymbol{b}^{(1)}$$

con $A \in \mathbb{R}^{n\times n}$ e tale che tutti i suoi minori principali siano diversi da zero, e $\boldsymbol{b} \in \mathbb{R}^n$. Definiamo ora la seguente matrice $L^{(1)}$, quadrata di ordine $n$, detta matrice elementare di Gauss:

$$L^{(1)} = \begin{bmatrix} 1 & & & & \\ m_{21} & 1 & & 0 & \\ m_{31} & 0 & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ m_{n1} & 0 & \dots & 0 & 1 \end{bmatrix}, \quad m_{i1} \in \mathbb{R} \quad i = 2, \dots, n. \tag{3.30}$$

i cui elementi $m_{i1}$ sono i moltiplicatori definiti al primo passo del metodo di Gauss. È facile verificare che

$$A^{(2)} = L^{(1)}A^{(1)}, \qquad \boldsymbol{b}^{(2)} = L^{(1)}\boldsymbol{b}^{(1)}$$

pertanto il sistema al secondo passo si ottiene moltiplicando (a sinistra) il sistema di partenza per la matrice (3.30). La matrice $L^{(1)}$ ha determinante unitario pertanto le matrici $A^{(1)}$ e $A^{(2)}$ hanno lo stesso determinante (come abbiamo già osservato in precedenza). Si può verificare che ad un generico passo $k$, definita la $k-$esima matrice elementare di Gauss

$$L^{(k)} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{k+1,k} & & \\ & & \vdots & \ddots & \\ & & m_{n,k} & & 1 \end{bmatrix},$$

in cui i numeri $m_{ik}$ sono i moltiplicatori al passo $k$, si ottiene

$$A^{(k+1)} = L^{(k)}A^{(k)}, \qquad \boldsymbol{b}^{(k+1)} = L^{(k)}\boldsymbol{b}^{(k)}. \tag{3.31}$$

Arrivando all'ultimo passo si ottiene

$$A^{(n)} = L^{(n-1)}A^{(n-1)}, \qquad \boldsymbol{b}^{(n)} = L^{(n-1)}\boldsymbol{b}^{(n-1)}$$

e, applicando ripetutamente la (3.31) è possibile mettere in relazione la matrice triangolare $A^{(n)}$ con la matrice dei coefficienti del sistema iniziale:

$$A^{(n)} = L^{(n-1)}L^{(n-2)}\dots L^{(2)}L^{(1)}A^{(1)} = L^{(n-1)}L^{(n-2)}\dots L^{(2)}L^{(1)}A. \tag{3.32}$$

A questo punto enunciamo, senza dimostrare, le seguenti proprietà:
• I proprietà: l'inversa di una matrice elementare di Gauss si ottiene cambiando il segno dei moltiplicatori:

$$\left(L^{(k)}\right)^{-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -m_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & -m_{n,k} & & 1 \end{bmatrix}.$$

• II proprietà: per ogni $k = 1, \ldots, n-1$ risulta

$$\left(L^{(1)}\right)^{-1} \ldots \left(L^{(k-1)}\right)^{-1} = \begin{bmatrix} 1 & & & & & \\ -m_{21} & \ddots & & & & \\ -m_{31} & \ddots & 1 & & & \\ \vdots & & -m_{k+1,k} & 1 & & \\ \vdots & & \vdots & & \ddots & \\ -m_{n1} & \ldots & -m_{n,k} & & & 1 \end{bmatrix}.$$

La relazione (3.32) può essere riscritta come

$$\left(L^{(n-1)} L^{(n-2)} \ldots L^{(2)} L^{(1)}\right)^{-1} A^{(n)} = A$$

da cui, sfruttando la proprietà della matrice inversa di un prodotto di matrici

$$\left(L^{(1)}\right)^{-1} \left(L^{(2)}\right)^{-1} \ldots \left(L^{(n-1)}\right)^{-1} A^{(n)} = A \tag{3.33}$$

Applicando la II proprietà si deduce che il prodotto delle inverse delle matrici elementari di Gauss è una matrice triangolare inferiore con elementi diagonali uguali a 1, pertando ponendo

$$L = \left(L^{(1)}\right)^{-1} \left(L^{(2)}\right)^{-1} \ldots \left(L^{(n-1)}\right)^{-1},$$

e

$$U = A^{(n)}$$

da (3.33) segue

$$A = LU.$$

## 3.4 Condizionamento di sistemi lineari

Nel Capitolo 1 è stato introdotto il concetto di rappresentazione in base ed è stata motivata la sostanziale inaffidabilità dei risultati dovuti ad elaborazioni numeriche, a causa dell'artimetica finita dell'elaboratore. Appare chiaro come la bassa precisione nel calcolo potrebbe fornire dei risultati numerici molto lontani da quelli reali. In alcuni casi tale proprietà è insita nel problema. Consideriamo il sistema lineare

$$\begin{array}{rcrcr} x & + & y & = & 2 \\ 1000x & + & 1001y & = & 2001 \end{array} \tag{3.34}$$

la cui soluzione è $x = y = 1$. Perturbiamo ora dell'1% il coefficiente di $x$ nella prima equazione e consideriamo pertanto il seguente sistema

$$\begin{array}{rcrcl} (1+0.01)x & + & y & = & 2 \\ 1000x & + & 1001y & = & 2001. \end{array}$$

Sarebbe naturale attendersi che la soluzione del sistema non sia molto lontana da quella del sistema (3.34), invece la soluzione è $\tilde{x} = -1/9$ e $\tilde{x} = 1901/900$, il che porta ad una differenza pari a

$$\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| = 1.57.$$

Se consideriamo inoltre il sistema

$$A\boldsymbol{x} = \boldsymbol{b} \tag{3.35}$$

dove $A \in \mathbb{R}^{n\times n}$ è la cosiddetta matrice di Hilbert, i cui elementi sono

$$a_{ij} = \frac{1}{i+j-1}, \qquad i,j = 1,\ldots,n$$

ovvero, se $n = 5$ :

$$A = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 & 1/5 \\ 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \end{bmatrix}$$

mentre il vettore $\boldsymbol{b}$ è scelto in modo tale che il vettore soluzione abbia tutte componenti uguali a 1, cosicchè si possa conoscere con esattezza l'errore commesso nel suo calcolo. Risolvendo il sistema di ordine 20 con il metodo di Gauss senza pivoting si osserva che la soluzione è, in realtà, molto lontana da quella teorica (l'errore relativo è pari circa a 23.5). Questa situazione peggiora prendendo matrici di dimensioni crescenti.

**Definizione 3.4.1** *Un sistema lineare per cui a piccoli errori dei dati corrispondono grandi errori nella soluzione si definisce mal condizionato o mal posto.*

L'importanza dello studio del condizionamento dei problemi dipende dal fatti che bisogna ricordare che, a causa degli errori legati alla rappresentazione dei numeri reali, il sistema che l'elaboratore risolve non coincide con quello teorico, poichè alla matrice $A$ ed al vettore $\boldsymbol{b}$ e necessario aggiungere la matrice $\delta A$ ed il vettore $\delta\boldsymbol{b}$ (che contengono le perturbazioni legate a tali errori), e che la soluzione ovviamente non è la stessa, pertanto la indichiamo con $\boldsymbol{x} + \delta\boldsymbol{x}$:

$$(A + \delta A)(\boldsymbol{x} + \delta\boldsymbol{x}) = \boldsymbol{b} + \delta\boldsymbol{b}. \tag{3.36}$$

Si può dimostrare che l'ordine di grandezza della perturbazione sulla soluzione è

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \|A\|\|A^{-1}\| \left( \frac{\|\delta A\|}{\|A\|} + \frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|} \right).$$

Il numero $K(A) = \|A\|\|A^{-1}\|$, detto indice di condizionamento del sistema, misura le amplificazioni degli errori sui dati del problema (ovvero la misura di quanto aumentano gli errori sulla soluzione). Il caso della matrice di Hilbert è appunto uno di quelli per cui l'indice di condizionamento assume valori molto grandi (di ordine esponenziale) all'aumentare della dimensione, si parla infatti di matrici malcondizionate. Quando ciò non accade si parla invece di matrici bencondizionate. Tra i metodi numerici che si possono applicare per la risoluzione di un problema un metodo risulta più stabile di un altro se è meno sensibile agli errori indotti dai calcoli. Lo studio della stabilità di un metodo numerico può perdere di significato quando il problema è fortemente mal condizionato, poichè in questo caso l'errore inerente (legato alla rappresentazione dei dati) prevale sull'errore algoritmico (introdotto nelle operazioni macchina).

# Capitolo 4

# Interpolazione di dati e Funzioni

## 4.1 Introduzione

Nel campo del Calcolo Numerico si possono incontrare diversi casi nei quali è richiesta l'approssimazione di una funzione (o di una grandezza incognita):
1) non è nota l'espressione analitica della funzione $f(x)$ ma si conosce il valore che assume in un insieme finito di punti $x_1, x_2, \ldots, x_n$. Si potrebbe pensare anche che tali valori siano delle misure di una grandezza fisica incognita valutate in differenti istanti di tempo.
2) Si conosce l'espressione analitica della funzione $f(x)$ ma è così complicata dal punto di vista computazionale che è più conveniente cercare un'espressione semplice partendo dal valore che essa assume in un insieme finito di punti. In questo capitolo analizzeremo un particolare tipo di approssimazione di funzioni cioè la cosiddetta interpolazione che richiede che la funzione approssimante assume in determinate ascisse esattamente lo stesso valore di $f(x)$. In entrambi i casi appena citati è noto, date certe informazioni supplementari, che la funzione approssimante va ricercata della forma:

$$f(x) \simeq g(x; a_0, a_1, \ldots, a_n). \tag{4.1}$$

Se i parametri $a_0, a_1, \ldots, a_n$ sono definiti dalla condizione di coincidenza di $f$ e $g$ nei punti $x_0, x_1, \ldots, x_n$, allora tale procedimento di approssimazione si chiama appunto Interpolazione. Invece se $x \notin [\min_i x_i, \max_i x_i]$ allora si parla di *Estrapolazione*. Un problema simile è invece quello in cui i valori

della funzione $f$ che sono noti sono affetti da errore e quindi si cerca una funzione approssimante che passi vicino ai valori assegnati ma che non sia perfettamente coincidente con essi. Il problema in questo caso prende il nome di Approssimazione. Tra i procedimenti di interpolazione il più usato è quello in cui si cerca la funzione $g$ in (4.1) nella forma

$$g(x; a_0, a_1, \ldots, a_n) = \sum_{i=0}^{n} a_i \Phi_i(x)$$

dove $\Phi_i(x)$, per $i = 0, \ldots, n$, sono funzioni fissate e i valori di $a_i$, $i = 0, \ldots, n$, sono determinati in base alle condizioni di coincidenza di $f$ con la funzione approssimante nei punti di interpolazione (detti anche nodi), $x_j$, cioè si pone

$$f(x_j) = \sum_{i=0}^{n} a_i \Phi_i(x_j) \qquad j = 0, \ldots, n. \tag{4.2}$$

Il processo di determinazione degli $a_i$ attraverso la risoluzione del sistema (4.2) si chiama metodo dei coefficienti indeterminati. Il caso più studiato è quello dell'interpolazione polinomiale, in cui si pone:

$$\Phi_i(x) = x^i \qquad i = 0, \ldots, n$$

e perciò la funzione approssimante $g$ assume la forma

$$\sum_{i=0}^{n} a_i x^i,$$

mentre le condizioni di coincidenza diventano

$$\begin{array}{ccccccccc} a_0 & +a_1 x_0 & +a_2 x_0^2 & +\ldots & +a_{n-1} x_0^{n-1} & +a_n x_0^n & = & f(x_0) \\ a_0 & +a_1 x_1 & +a_2 x_1^2 & +\ldots & +a_{n-1} x_1^{n-1} & +a_n x_1^n & = & f(x_1) \\ \vdots & \vdots & \vdots & & & & & \vdots \\ a_0 & +a_1 x_n & +a_2 x_n^2 & +\ldots & +a_{n-1} x_n^{n-1} & +a_n x_n^n & = & f(x_n) \end{array} \tag{4.3}$$

Le equazioni (4.3) costituiscono un sistema di $n+1$ equazioni nelle $n+1$ incognite $a_i$, $i = 0, \ldots, n$ :

$$V\boldsymbol{a} = \boldsymbol{y}$$

dove la matrice dei coefficienti è

$$V = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^{n-1} & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^{n-1} & x_1^n \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} & x_n^n \end{bmatrix},$$

i vettori dei termini noti e delle incognite sono, rispettivamente,

$$\boldsymbol{y} = [f(x_0), f(x_1), \dots, f(x_n)]^T$$

e $\boldsymbol{a} = [a_0, a_1, \dots, a_n]^T$.
Se i nodi $x_j$ sono a due a due distinti allora la matrice dei coefficienti del sistema (4.3), detta matrice di Vandermonde, è non singolare e pertanto il problema dell'interpolazione ammette sempre un'unica soluzione. Il metodo dei coefficienti indeterminati consente di trovare la soluzione del problema solo risolvendo un sistema lineare che potrebbe avere grandi dimensioni, essere malcondizionato (soprattutto se due nodi sono molto vicini) e comunque non in grado di fornire un'espressione in forma chiusa del polinomio. Per questi motivi descriviamo un modo alternativo per risolvere il problema di interpolazione in grado di fornire l'espressione esplicita del polinomio cercato.

## 4.2 Il Polinomio Interpolante di Lagrange

Al fine di dare una forma esplicita al polinomio interpolante, scriviamo il candidato polinomio nella seguente forma:

$$L_n(x) = \sum_{k=0}^{n} l_{nk}(x) f(x_k) \tag{4.4}$$

dove gli $l_{nk}(x)$ sono per il momento generici polinomi di grado $n$. Imponendo le condizioni di interpolazione

$$L_n(x_i) = f(x_i) \qquad i = 0, \dots, n$$

deve essere, per ogni $i$:

$$L_n(x_i) = \sum_{k=0}^{n} l_{nk}(x_i) f(x_k) = f(x_i)$$

ed è evidente che se

$$l_{nk}(x_i) = \begin{cases} 0 & \text{se } k \neq i \\ \\ 1 & \text{se } k = i \end{cases} \tag{4.5}$$

allora esse sono soddisfatte. Infatti calcolando il polinomio (4.4) in un generico nodo $x_i$ risulta

$$\begin{aligned} L_n(x_i) &= \sum_{k=0}^{n} l_{nk}(x_i) f(x_k) \\ &= \underbrace{\sum_{k=0}^{i-1} l_{nk}(x_i) f(x_k)}_{=0} + \underbrace{l_{ni}(x_i)}_{=1} f(x_i) + \underbrace{\sum_{k=i+1}^{n} l_{nk}(x_i) f(x_k)}_{=0} = f(x_i). \end{aligned}$$

Per determinare l'espressione del generico polinomio $l_{nk}(x)$ osserviamo che la prima condizione di (4.5) indica che esso si annulla negli $n$ nodi $x_0, x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n$ pertanto deve essere

$$l_{nk}(x) = c_k \prod_{i=0, i\neq k}^{n} (x - x_i)$$

mentre impondendo la seconda condizione di (4.5)

$$l_{nk}(x_k) = c_k \prod_{i=0, i\neq k}^{n} (x_k - x_i) = 1$$

si trova immediatamente:

$$c_k = \frac{1}{\prod_{i=0, i\neq k}^{n} (x_k - x_i)}.$$

In definitiva il polinomio interpolante ha la seguente forma:

$$L_n(x) = \sum_{k=0}^{n} \left( \prod_{i=0, i\neq k}^{n} \frac{x - x_i}{x_k - x_i} \right) f(x_k). \tag{4.6}$$

Il polinomio (4.6) prende il nome di Polinomio di Lagrange mentre i polinomi:

$$l_{nk}(x) = \prod_{i=0, i\neq k}^{n} \frac{x - x_i}{x_k - x_i}; \qquad k = 0, 1, \ldots, n$$

si chiamano Polinomi Fondamentali di Lagrange.

### 4.2.1 Il Resto del Polinomio di Lagrange

Assumiamo che la funzione interpolata $f(x)$ sia di classe $\mathcal{C}^{n+1}([a, b])$ e valutiamo l'errore che si commette nel sostituire $f(x)$ con $L_n(x)$ in un punto $x \neq x_i$. Supponiamo che l'intervallo $[a, b]$ sia tale da contenere sia i nodi $x_i$ che l'ulteriore punto $x$. Sia dunque

$$e(x) = f(x) - L_n(x)$$

l'errore (o resto) commesso nell'interpolazione della funzione $f(x)$. Poichè

$$e(x_i) = f(x_i) - L_n(x_i) = 0 \qquad i = 0, \ldots, n$$

è facile congetturare per $e(x)$ la seguente espressione:

$$e(x) = c(x)\omega_{n+1}(x)$$

dove

$$\omega_{n+1}(x) = \prod_{i=0}^{n} (x - x_i)$$

è il cosiddetto polinomio nodale mentre $c(x)$ è una funzione da determinare. Definiamo ora la funzione

$$\Phi(t; x) = f(t) - L_n(t) - c(x)\omega_{n+1}(t)$$

dove $t$ è una variabile ed $x$ è un valore fissato. Calcoliamo la funzione $\Phi(t; x)$ nei nodi $x_i$:

$$\Phi(x_i; x) = f(x_i) - L_n(x_i) - c(x)\omega_{n+1}(x_i) = 0$$

e anche nel punto $x$:

$$\Phi(x;x) = f(x) - L_n(x) - c(x)\omega_{n+1}(x) = e(x) - c(x)\omega_{n+1}(x) = 0$$

pertanto la funzione $\Phi(t;x)$ ammette almeno $n+2$ zeri distinti. Osserviamo inoltre che è $\Phi(t;x)$ è derivabile con continuità $n+1$ volte poichè, per ipotesi, $f(x)$ è di classe $\mathcal{C}^{n+1}$. Applicando il teorema di Rolle segue che $\Phi'(t;x)$ ammette almeno $n+1$ zeri distinti. Riapplicando lo stesso teorema segue che $\Phi''(t;x)$ ammette almeno $n$ zeri distinti. Così proseguendo segue che

$$\exists \xi_x \in [a,b] \;\; \ni' \;\; \Phi^{(n+1)}(\xi_x;x) = 0.$$

Calcoliamo ora la derivata di ordine $n+1$ della funzione $\Phi(t;x)$, osservando innanzitutto che la derivata di tale ordine del polinomio $L_n(x)$ è identicamente nulla. Pertanto

$$\Phi^{(n+1)}(t;x) = f^{(n+1)}(t) - c(x)\frac{d^{n+1}}{dt^{n+1}}\omega_{n+1}(t).$$

Calcoliamo la derivata di ordine $n+1$ del polinomio nodale. Osserviamo innanzitutto che

$$\omega_{n+1}(t) = \prod_{i=0}^{n}(t - x_i) = t^{n+1} + p_n(t)$$

dove $p_n(t)$ è un polinomio di grado al più $n$. Quindi

$$\frac{d^{n+1}}{dt^{n+1}}\omega_{n+1}(t) = \frac{d^{n+1}}{dt^{n+1}}t^{n+1}.$$

Poichè

$$\frac{d}{dt}t^{n+1} = (n+1)t^n$$

e

$$\frac{d^2}{dt^2}t^{n+1} = (n+1)nt^{n-1}$$

è facile dedurre che

$$\frac{d^{n+1}}{dt^{n+1}}t^{n+1} = \frac{d^{n+1}}{dt^{n+1}}\omega_{n+1}(t) = (n+1)!.$$

Pertanto

$$\Phi^{(n+1)}(t;x) = f^{(n+1)}(t) - c(x)(n+1)!$$

e

$$\Phi^{(n+1)}(\xi_x; x) = f^{(n+1)}(\xi_x) - c(x)(n+1)! = 0$$

cioè

$$c(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}$$

e in definitiva

$$e(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega_{n+1}(x). \tag{4.7}$$

**Esempio 4.2.1** *Supponiamo di voler calcolare il polinomio interpolante di Lagrange passante per i punti* $(-1,-1)$, $(0,1)$, $(1,-1)$, $(3,2)$ *e* $(5,6)$. *Il grado di tale polinomio è* 4, *quindi definiamo i nodi*

$$x_0 = -1, \quad x_1 = 0, \quad x_2 = 1, \quad x_3 = 3, \quad x_4 = 5,$$

*cui corrispondono le ordinate che indichiamo con* $y_i$, $i = 0, \ldots, 4$*:*

$$y_0 = -1, \quad y_1 = 1, \quad y_2 = -1, \quad y_3 = 2, \quad y_4 = 6.$$

*Scriviamo ora l'espressione del polinomio* $L_4(x)$*:*

$$L_4(x) = l_{4,0}(x)y_0 + l_{4,1}(x)y_1 + l_{4,2}(x)y_2 + l_{4,3}(x)y_3 + l_{4,4}(x)y_4 \tag{4.8}$$

*e calcoliamo i* 5 *polinomi fondamentali di Lagrange:*

$$\begin{aligned} l_{4,0}(x) &= \frac{(x-0)(x-1)(x-3)(x-5)}{(-1-0)(-1-1)(-1-3)(-1-5)} \\ &= \frac{1}{48}\, x(x-1)(x-3)(x-5) \\ l_{4,1}(x) &= \frac{(x+1)(x-1)(x-3)(x-5)}{(0+1)(0-1)(0-3)(0-5)} \\ &= -\frac{1}{15}(x+1)(x-1)(x-3)(x-5) \\ l_{4,2}(x) &= \frac{(x+1)(x-0)(x-3)(x-5)}{(1+1)(1-0)(1-3)(1-5)} \\ &= \frac{1}{16}x(x+1)(x-3)(x-5) \end{aligned}$$

$$
\begin{aligned}
l_{4,3}(x) &= \frac{(x+1)(x-0)(x-1)(x-5)}{(3+1)(3-0)(3-1)(3-5)} \\
&= -\frac{1}{48}x(x+1)(x-1)(x-5) \\
l_{4,4}(x) &= \frac{(x+1)(x-0)(x-1)(x-3)}{(5+1)(5-0)(5-1)(5-3)} \\
&= \frac{1}{240}x(x+1)(x-1)(x-3)
\end{aligned}
$$

*Sostituendo in (4.8) il valore della funzione nei nodi si ottiene l'espressione finale del polinomio interpolante:*

$$L_4(x) = -l_{4,0}(x) + l_{4,1}(x) - l_{4,2}(x) + 2l_{4,3}(x) + 6l_{4,4}(x).$$

*Se vogliamo calcolare il valore approssimato della funzione $f(x)$ in un'ascissa diversa dai nodi, per esempio $x = 2$ allora dobbiamo calcolare il valore del polinomio interpolante $L_4(2)$.*
*Nelle figure 4.1-4.5 sono riportati i grafici dei cinque polinomi fondamentali di Lagrange: gli asterischi evidenziano il valore assunto da tali polinomi nei nodi di interpolazione. Nella figura 4.6 è tracciato il grafico del polinomio interpolante di Lagrange, i cerchi evidenziano ancora una volta i punti di interpolazione.*

### 4.2.2 Il fenomeno di Runge

Nell'espressione dell'errore è presente, al denominatore, il fattore $(n+1)!$, che potrebbe indurre a ritenere che, utilizzando un elevato numero di nodi, l'errore tenda a zero ed il polinomio interpolante converga alla funzione $f(x)$. Questa ipotesi è confutata se si costruisce il polinomio che interpola la funzione

$$f(x) = \frac{1}{1+x^2}$$

nell'intervallo $[-5, 5]$ e prendendo 11 nodi equidistanti $-5, -4, -3, \ldots, 3, 4, 5$. Nella successiva figura viene appunto visualizzata la funzione (in blu) ed il relativo polinomio interpolante (in rosso).
Il polinomio interpolante presenta infatti notevoli oscillazioni, soprattutto verso gli estremi dell'intervallo di interpolazione, che diventano ancora più

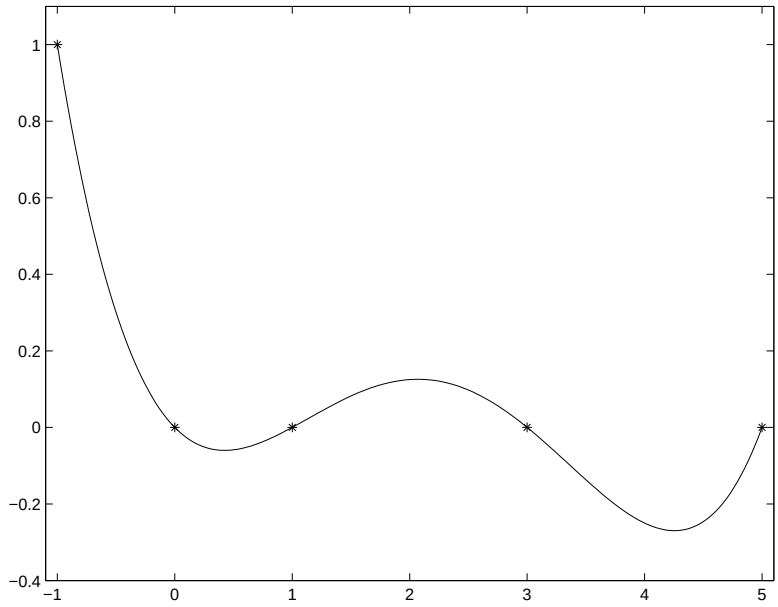



Figura 4.1: Grafico del polinomio $l_{40}(x)$.

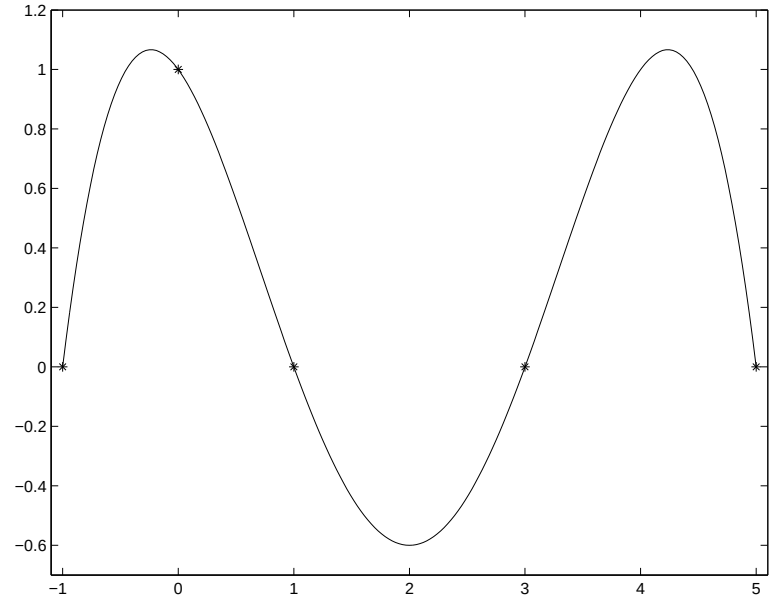



Figura 4.2: Grafico del polinomio $l_{41}(x)$.

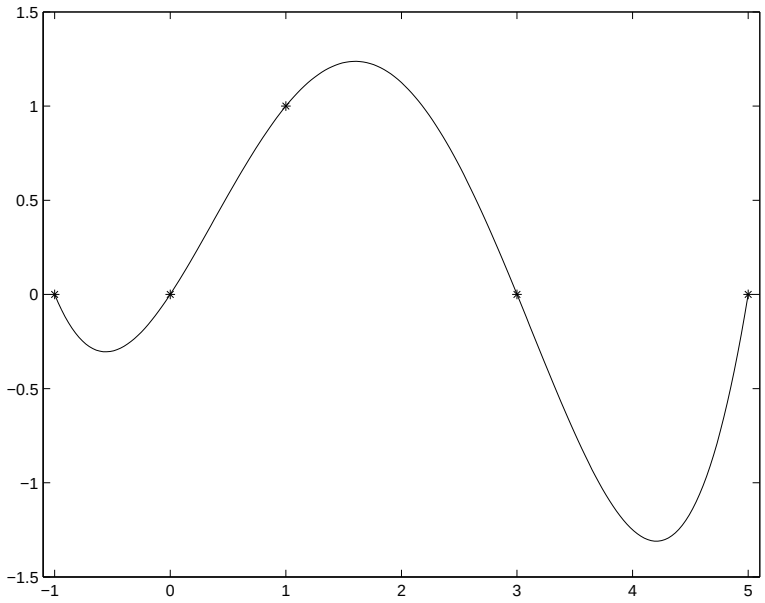



Figura 4.3: Grafico del polinomio $l_{42}(x)$.

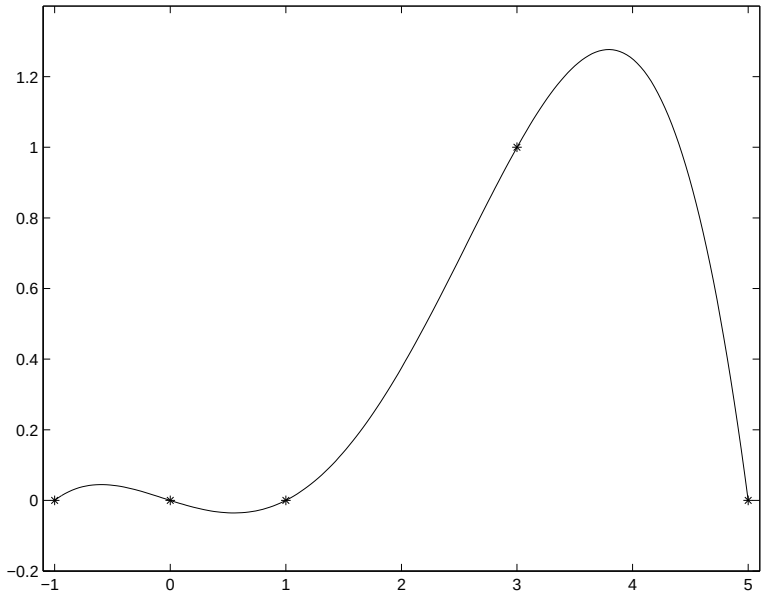



Figura 4.4: Grafico del polinomio $l_{43}(x)$.

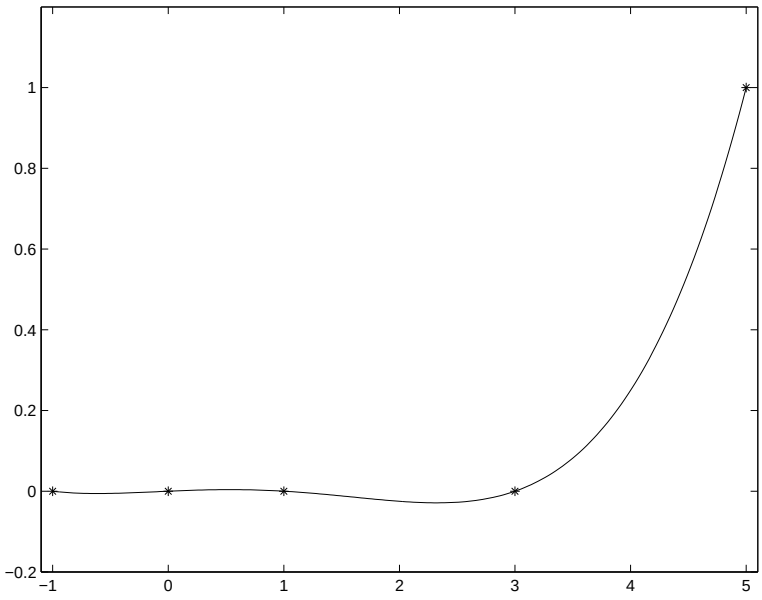



Figura 4.5: Grafico del polinomio $l_{44}(x)$.

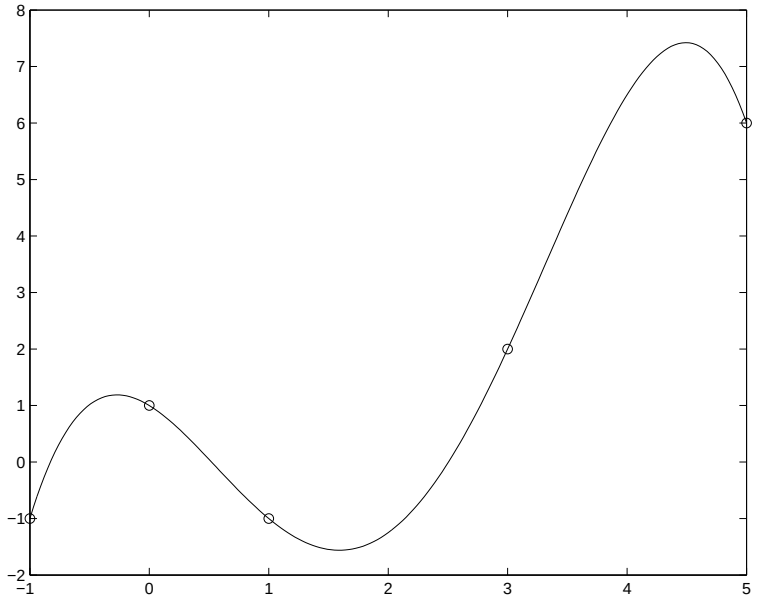



Figura 4.6: Grafico del polinomio interpolante di Lagrange $L_4(x)$.

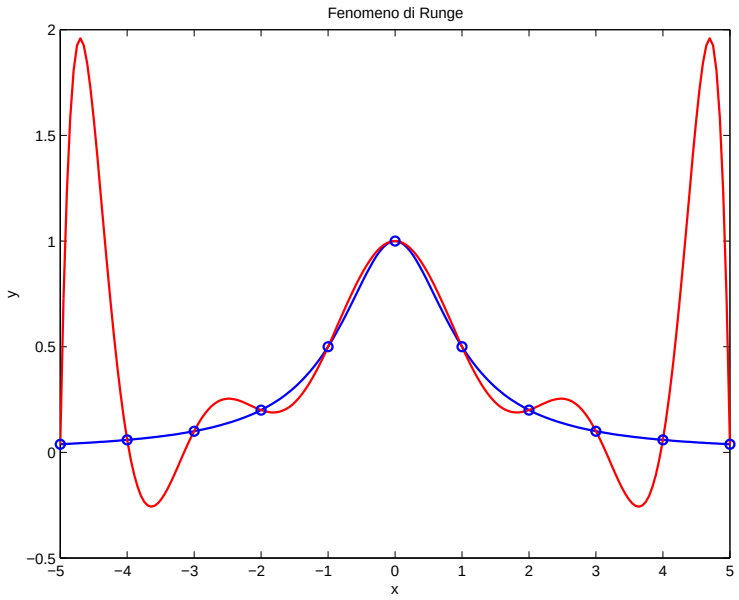



Figura 4.7: Il fenomeno di Runge.

evidenti all'aumentare di $n$. Tale fenomeno, detto appunto fenomeno di Runge, è dovuto ad una serie di situazioni concomitanti:

1. il polinomio nodale, al crescere di $n$, assume un'andamento fortemente oscillante, soprattutto quando i nodi sono equidistanti;

2. alcune funzioni hanno le derivate il cui valore tende a crescere con un ordine di grandezza talmente elevato da neutralizzare di fatto la presenza del fattoriale al denominatore dell'espressione dell'errore.

Per ovviare al fenomeno di Runge si possono utilizzare insiemi di nodi non equidistanti oppure utilizzare funzioni interpolanti polinomiali a tratti (interpolando di fatto su intervalli più piccoli e imponendo le condizioni di continuità fino ad un ordine opportuno).

```
function  yy=lagrange(x,y,xx);
%
% La funzione calcola il polinomio interpolante di Lagrange
% in un vettore assegnato di ascisse
%
```

```
% Parametri di input
% x = vettore dei nodi
% y = vettore delle ordinate nei nodi
% xx = vettore delle ascisse in cui calcolare il polinomio
% Parametri di output
% yy = vettore delle ordinate del polinomio
%
n = length(x);
m = length(xx);
yy = zeros(size(xx));
for i=1:m
    yy(i)=0;
    for k=1:n
        yy(i)=yy(i)+prod((xx(i)-x([1:k-1,k+1:n]))./...
            (x(k)-x([1:k-1,k+1:n])))*y(k);
    end
end
return
```

## 4.3 Minimizzazione del Resto nel Problema di Interpolazione

Supponiamo che la funzione $f(x)$ sia approssimata su $[a, b]$ dal polinomio interpolante $L_n(x)$ e siano $x_0, x_1, \ldots, x_n$ i nodi di interpolazione. Come già sappiamo se $x \in [a, b]$ risulta

$$e(x) = f(x) - L_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega_{n+1}(x) \qquad \xi_x \in [a, b]$$

e dove

$$\omega_{n+1}(x) = \prod_{i=0}^{n}(x - x_i).$$

Si noti che variando i nodi $x_i$, $i = 0, \ldots, n$, cambia il polinomio $\omega_{n+1}(x)$ e di conseguenze cambia l'errore. Ha senso allora porsi il seguente problema: indicato con $\mathcal{P}_{n+1}$ l'insieme di tutti i polinomi di grado al più $n+1$ cerchiamo il polinomio $\tilde{p} \in \mathcal{P}_{n+1}$ tale che:

$$\max_{x\in[a,b]} |\tilde{p}(x)| = \min_{p\in\mathcal{P}_{n+1}} \max_{x\in[a,b]} |p(x)|. \tag{4.9}$$

Per dare una risposta a questo problema è essenziale introdurre i Polinomi di Chebyshev di $1^a$ Specie.

### 4.3.1 Polinomi di Chebyshev

I polinomi di Chebyshev $T_n(x)$, $n \geq 0$, sono così definiti:

$$T_n(x) = \cos(n \arccos x) \tag{4.10}$$

per $x \in [-1, 1]$. Per esempio:

$$\begin{aligned} T_0(x) &= \cos(0 \arccos x) = \cos 0 = 1 \\ T_1(x) &= \cos(1 \arccos x) = x \end{aligned}$$

e così via. È possibile ricavare una relazione di ricorrenza sui polinomi di Chebyshev che permette un più agevole calcolo. Infatti, posto

$$\arccos x = \theta \qquad (\text{ovvero } x = \cos\theta)$$

risulta

$$T_n(x) = \cos n\theta(x).$$

Considerando le relazioni

$$T_{n+1}(x) = \cos(n+1)\theta = \cos n\theta \ \cos\theta - \sin n\theta \ \sin\theta$$

$$T_{n-1}(x) = \cos(n-1)\theta = \cos n\theta \ \cos\theta + \sin n\theta \ \sin\theta$$

e sommandole membro a membro,

$$T_{n+1}(x) + T_{n-1}(x) = 2\cos\theta\cos n\theta = 2xT_n(x)$$

si ricava la seguente relazione di ricorrenza

$$T_{n+1}(x) = 2x \ T_n(x) - T_{n-1}(x), \qquad n \geq 1 \tag{4.11}$$

che, insieme all'espressione dei primi due polinomi,

$$T_0(x) = 1, \qquad T_1(x) = x.$$

consente di calcolare tutti i polinomi di Chebyshev.
L'espressione dei primi polinomi è la seguente

$$\begin{aligned} T_2(x) &= 2xT_1(x) - T_0(x) = 2x^2 - 1 \\ T_3(x) &= 2xT_2(x) - T_1(x) = 4x^3 - 3x \\ T_4(x) &= 2xT_3(x) - T_2(x) = 8x^4 - 8x^2 + 1 \\ T_5(x) &= 2xT_4(x) - T_3(x) = 16x^5 - 20x^3 + 5x \end{aligned}$$

Le seguenti proprietà dei polinomi di Chebyshev sono di facile dimostrazione:

1. $\max_{x\in[-1,1]} |T_n(x)| = 1$
2. $T_{2k}(-x) = T_{2k}(x)$ ovvero i polinomio di grado pari sono funzioni pari, quindi tutti i coefficienti delle potenze dispari di $x$ sono nulli;
3. $T_{2k+1}(-x) = -T_{2k+1}(x)$ ovvero i polinomio di grado dispari sono funzioni dispari, quindi tutti i coefficienti delle potenze pari di $x$ sono nulli;
4. $T_n(x) = 2^{n-1}x^n + \ldots\ldots$
5. $T_n(x)$ assume complessivamente $n+1$ volte il valore $+1$ e $-1$ nei punti:

$$\begin{aligned} x_k &= \cos\frac{k\pi}{n} \qquad k = 0, \ldots, n; \\ T_n(x_k) &= (-1)^k \qquad k = 0, \ldots, n; \end{aligned}$$

6. $T_n(x)$ ha $n$ zeri distinti nell'intervallo $]-1, 1[$ dati da

$$x_k = \cos\frac{(2k+1)\pi}{2n} \qquad k = 0, \ldots, n-1.$$

Infatti è sufficiente porre

$$\cos n\theta = 0$$

da cui risulta

$$n\theta = \frac{\pi}{2} + k\pi = \frac{(2k+1)\pi}{2}, \qquad k = 0, \ldots, n-1.$$

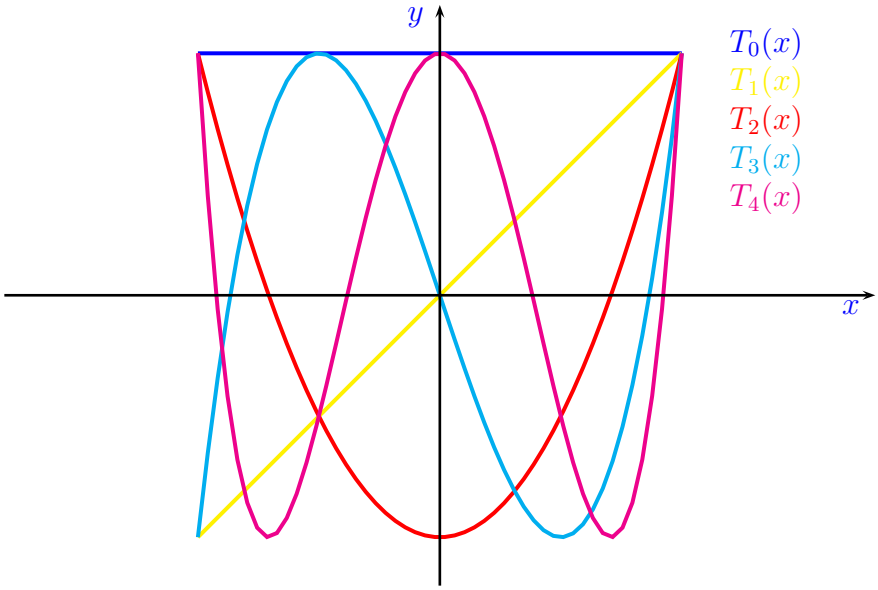



Figura 4.8: Grafico dei primi cinque polinomi di Chebyshev

Nella Figura 4.8 sono tracciati i grafici dei primi cinque polinomi di Chebyshev nell'intervallo $[-1, 1]$. Ovviamente per calcolare il valore del polinomio $T_n(x)$ in un punto $x$ fissato si usa la formula di ricorrenza (4.11), in quanto tale espressione è valida per ogni $x \in \mathbb{R}$.
Sia

$$\tilde{T}_n(x) = \frac{1}{2^{n-1}} T_n(x)$$

il polinomio di Chebyshev normalizzato in modo da risultare monico (ricordiamo che un polinomio di grado $n$ è monico se il coefficiente del termine di grado massimo è 1). Vale allora la seguente proprietà di minimax.

**Teorema 4.3.1** *(Proprietà di minimax) Se $p_n(x)$ è un qualunque polinomio monico di grado $n$ si ha:*

$$\frac{1}{2^{n-1}} = \max_{x\in[-1,1]} |\tilde{T}_n(x)| \leq \max_{x\in[-1,1]} |p_n(x)|.$$

*Dimostrazione.* Assumiamo per assurdo che sia

$$\max_{x\in[-1,1]} |p_n(x)| < \frac{1}{2^{n-1}}$$

e consideriamo il polinomio $d(x) = \tilde{T}_n(x) - p_n(x)$. Osserviamo subito che essendo sia $\tilde{T}_n(x)$ che $p_n(x)$ monici, $d(x)$ è un polinomio di grado al più $n-1$. Siano $t_0, t_1, \ldots, t_n$ i punti in cui $T_n$ assume valore $-1$ e $+1$. Allora:

$$segn(d(t_k)) \quad = segn(\tilde{T}_n(t_k) - p_n(t_k)) = segn(\tilde{T}_n(t_k)).$$

Poichè $\tilde{T}_n(x)$ cambia segno $n$ volte anche $d(x)$ cambia segno $n$ volte e pertanto ammetterà $n$ zeri, in contraddizione con il fatto che $d(x)$ è un polinomio di grado al più $n-1$. □

Osservazione. In verità vale un'affermazione più forte di quella del teorema, cioè se $p(x)$ è un polinomio monico di grado $n$ diverso da $\tilde{T}_n(x)$ allora:

$$\max_{x\in[-1,1]} |p(x)| > \frac{1}{2^{n-1}}.$$

Il teorema di minimax stabilisce che, tra tutti i polinomi di grado $n$ definiti nell'intervallo $[-1,1]$, il polinomio di Chebyshev monico è quello che ha il massimo più piccolo. Supponendo che l'intervallo di interpolazione della funzione $f(x)$ sia appunto $[-1,1]$ e scegliendo come nodi gi zeri del polinomio di Chebyshev risulta

$$\omega_{n+1}(x) = \tilde{T}_{n+1}(x)$$

pertanto

$$e(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\tilde{T}_{n+1}(x)$$

e, massimizzando tale errore, risulta

$$\begin{aligned}\max_{x\in[-1,1]} |e(x)| \quad &\leq \max_{x\in[-1,1]} \left|\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\right| \max_{x\in[-1,1]} |\omega_{n+1}(x)| \\ &= \frac{1}{2^n(n+1)!} \max_{x\in[-1,1]} |f^{(n+1)}(\xi_x)|.\end{aligned}$$

La crescita dell'errore può dipendere solo dalla derivata di ordine $n+1$ della funzione $f(x)$.
Se l'intervallo di interpolazione è $[a,b] \neq [-1,1]$ allora il discorso può essere ripetuto egualmente effettuando una trasformazione lineare tra i due intervalli, nel modo riportato in Figura 4.9. Calcolando la retta nel piano $(x,t)$

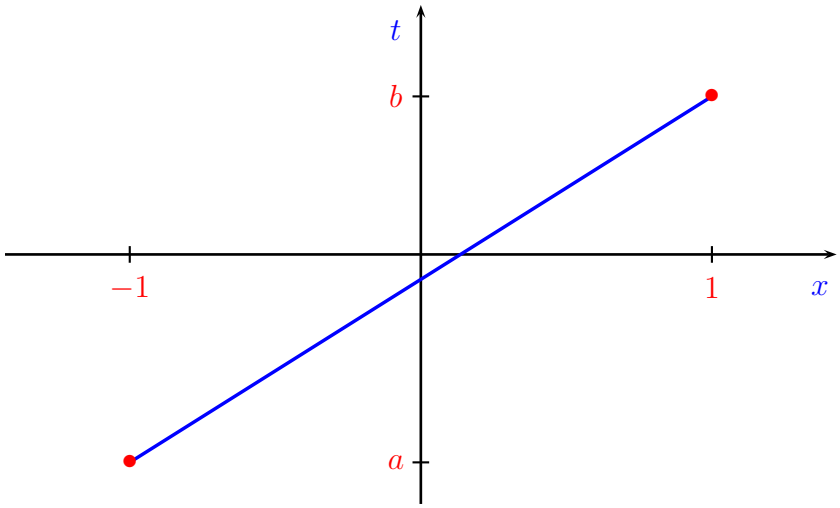



Figura 4.9: Trasformazione lineare tra gli intervalli $[-1, 1]$ e $[a, b]$.

passante per i punti $(-1, a)$ e $(1, b)$:

$$t = \frac{b-a}{2}x + \frac{a+b}{2} \tag{4.12}$$

detti $x_k$ gli zeri del polinomio di Chebyshev $T_{n+1}(x)$ allora si possono usare come nodi i valori

$$\tau_k = \frac{b-a}{2}x_k + \frac{a+b}{2}, \qquad k = 0, 1, \ldots, n,$$

ovvero

$$\tau_k = \frac{b-a}{2}\cos\frac{(2k+1)\pi}{2(n+1)} + \frac{a+b}{2} \qquad k = 0, 1, \ldots, n. \tag{4.13}$$

Per determinare l'espressione del polinomio di Chebyshev traslato nell'intervallo $[a, b]$, si deve utilizzare la trasformazione lineare che fornisce $x \in [a, b]$ a partire da $t \in [-1, 1]$:

$$x = \frac{2t - (b+a)}{b-a}$$

ovvero

$$T_{n+1}^{[a,b]}(x) = T_{n+1}\left(\frac{2t-(b+a)}{b-a}\right),$$

il cui coefficiente di grado massimo vale

$$2^n \frac{2^{n+1}}{(b-a)^{n+1}} = \frac{2^{2n+1}}{(b-a)^{n+1}}.$$

Se come nodi di interpolazione scegliamo i punti $t_k$ dati da (4.13), cioè gli $n+1$ zeri del polinomio $\tilde{T}_{n+1}^{[a,b]}(x)$, allora abbiamo il polinomio monico è

$$\tilde{T}_{n+1}^{[a,b]}(x) = \frac{(b-a)^{n+1}}{2^{2n+1}} T_{n+1}\left(\frac{2t-(b+a)}{b-a}\right),$$

considerato che la trasformazione lineare inversa della (4.12) è

$$x = \frac{2t-(b+a)}{b-a}, \qquad t \in [a,b] \to x \in [-1,1]$$

quindi per l'errore dell'interpolazione vale la seguente maggiorazione:

$$\begin{aligned} \max_{x\in[a,b]} |e(x)| &\leq \max_{x\in[a,b]} \left|\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\right| \max_{x\in[a,b]} |\tilde{T}_{n+1}^{[a,b]}(x)| \\ &= \max_{x\in[a,b]} \left|\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\right| \frac{(b-a)^{n+1}}{2^{2n+1}}. \end{aligned}$$

Nella Figura 4.10 sono raffigurati la funzione di Runge ed il polinomio interpolante di Lagrange di grado 10 calcolato prendendo come nodi gli zeri del polinomio di Chebyshev di grado 11. Si può osservare la differenza con la Figura 4.7. Di seguito viene riportato il codice per tracciare il grafico del polinomio interpolante la funzione di Runge nei nodi di Chebyshev in un intervallo scelto dall'utente.

```
clear
format long e
a = input('Inserire estremo sinistro ');
b = input('Inserire estremo destro ');
n = input('Inserire il numero di nodi ');
%
% Calcolo del vettore dei nodi di Chebyshev
%
x = (a+b)/2+(b-a)/2*cos((2*[0:n-1]+1)*pi./(2*n));
```

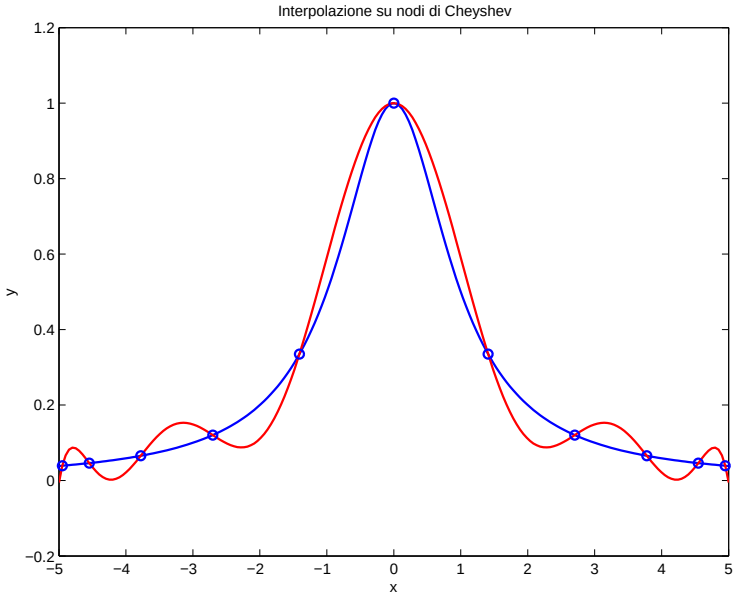



Figura 4.10: Interpolazione su nodi di Chebyshev.

```
xx = linspace(a,b,200);
y = 1./(x.^2+1);
yy = 1./(xx.^2+1);
%
% Calcolo del polinomio interpolante
%
zz = lagrange(x,y,xx);
figure(1)
plot(xx,yy)
hold on
pause
plot(x,y,'ok')
pause
plot(xx,zz,'r')
title('Grafico della funzione e del polinomio interpolante ')
hold off
figure(2)
plot(xx,abs(yy-zz))
title('Grafico dell'errore nell''interpolazione')
```

## 4.4 Interpolazione con Funzioni Polinomiali a Tratti

L'interpolazione polinomiale con un numero di nodi sufficientemente alto può dar luogo a polinomi interpolanti che mostrano un comportamento fortemente oscillatorio che può essere inaccettabile. In questo caso si preferisce usare una diversa strategia consistente nell'approssimare la funzione con polinomi di basso grado su sottointervalli dell'intervallo di definizione. Per esempio, supposto che l'intero $n$ sia un multiplo di 3, denotiamo con $P_{3,j}(x)$ il polinomio di interpolazione di terzo grado associato ai nodi $x_{3j-3}$, $x_{3j-2}$, $x_{3j-1}$, $x_{3j}$, $j = 1, 2, \ldots, n/3$. Come funzione interpolante prendiamo poi la funzione:

$$I_n(x) = P_{3,j}(x) \qquad \text{in } [x_{3j-3}, x_{3j}]$$

che prende il nome di Funzione di tipo polinomiale a tratti. La tecnica esposta non è l'unica, anzi la più popolare è forse quella basata sull'uso delle cosiddette Funzioni Spline.

### 4.4.1 Interpolazione con Funzioni Spline

Con il termine spline si indica in lingua inglese un sottile righello usato nella progettazione degli scafi dagli ingegneri navali, per raccordare su un piano un insieme di punti $(x_i, y_i)$, $i = 0, \ldots, n+1$.
Imponendo mediante opportune guide che il righello passi per i punti assegnati, si ottiene una curva che li interpola. Detta $y = f(x)$ l'equazione della curva definita dalla spline, sotto opportune condizioni $f(x)$ può essere approssimativamente descritta da pezzi di polinomi di terzo grado in modo che la funzione e le sue prime due derivate risultino continue nell'intervallo di interesse. La derivata terza può presentare discontinuità nei punti $x_i$. La spline può essere concettualmente rappresentata e generalizzata nel seguente modo.
Sia

$$\Delta =: a \equiv x_0 < x_1 < x_2 < \cdots < x_n < x_{n+1} \equiv b$$

una decomposizione dell'intervallo $[a, b]$.

**Definizione 4.4.1** *Si dice funzione Spline di grado $m \geq 1$ relativa alla decomposizione $\Delta$ una funzione $s(x)$ soddisfacente le seguenti proprietà:*

1. *$s(x)$ ristretta a ciascun intervallo $[x_i, x_{i+1}]$, $i = 0, \ldots, n$, è un polinomio di grado al più $m$;*

2. *la derivata $s^{(k)}(x)$ è una funzione continua su $[a, b]$ per $k = 0, 1, \ldots, m-1$.*

Si verifica facilmente che l'insieme delle spline di grado assegnato è uno spazio vettoriale. In generale le spline vengono utilizzate in tutte quelle situazioni dove l'approssimazione polinomiale sull'intero intervallo non è soddisfacente. Per $m = 1$ si hanno le cosiddette spline lineari, mentre per $m = 3$ si hanno le spline cubiche.

## 4.5 Approssimazione ai minimi quadrati

Come si è già accennato nell'introduzione di questo Capitolo quando i dati $(x_i, y_i)$, $i = 0, \ldots, n$, sono rilevati con scarsa precisione, non ha molto senso cercare un polinomio di grado $n$ (o, più in generale una funzione $\Psi(x)$) che interpoli i valori $y_i$ nei nodi $x_i$. In questo caso è più utile cercare una funzione che si avvicini il più possibile ai dati rilevati. Chiaramente i criteri che si possono scegliere per tradurre l'espressione "*si avvicini il più possibile*" in termini matematici sono molteplici. Nel seguito descriviamo uno dei più usati non senza aver richiamato alcune definizioni di algebra lineare. In particolare ricordiamo che si definisce norma 2 di un vettore (o norma euclidea) $\boldsymbol{x} \in \mathbb{R}^n$ la quantità

$$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

che, introducendo il prodotto scalare tra vettori $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$,

$$(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{y}^T \boldsymbol{x} = \sum_{i=1}^{n} x_i y_i.$$

può essere scritta nel seguente modo:

$$\|\boldsymbol{x}\|_2 = \sqrt{\boldsymbol{x}^T \boldsymbol{x}} = \sqrt{(\boldsymbol{x}, \boldsymbol{x})}.$$

Indichiamo con $\Phi(a_0, a_1, \ldots, a_m; x)$ la funzione (nella variabile $x$) che stiamo cercando e che dipende dagli $m + 1$ coefficienti $a_0, a_1, \ldots, a_m$, e sia $\varepsilon_i$ la

differenza tra il valore assunto da tale funzione nei nodi $x_i$ ed valore rilevato $y_i$:

$$\varepsilon_i = \Phi(a_0, a_1, \ldots, a_m; x_i) - y_i, \qquad i = 0, \ldots, n.$$

Si possono determinare i coefficienti $a_0, \ldots, a_m$ in modo tale che il vettore

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_0 & \varepsilon_2 & \ldots & \varepsilon_n \end{bmatrix}$$

abbia la minima norma euclidea al quadrato. Definita la funzione

$$Q(a_0, a_1, \ldots, a_m) = \|\boldsymbol{\varepsilon}\|_2^2 = \sum_{i=0}^{n} \varepsilon_i^2 = \sum_{i=0}^{n} \left(\Phi(a_0, a_1, \ldots, a_m; x_i) - y_i\right)^2$$

si deve risolvere il seguente problema di minimo

$$Q(a_0^*, a_1^*, \ldots, a_m^*) = \min_{a_0, \ldots, a_m \in \mathbb{R}^{m+1}} Q(a_0, a_1, \ldots, a_m). \tag{4.14}$$

Tale metodo prende il nome, appunto, di approssimazione ai minimi quadrati, poichè consiste nel minimizzare una somma di quadrati. Un caso particolare di tale metodo consiste nel cercare una funzione $\Phi(a_0, \ldots, a_m)$ di tipo lineare che risolve il problema di minimo appena definito. Tale metodo viene descritto nel successivo paragrafo.

### 4.5.1 La Retta di Regressione

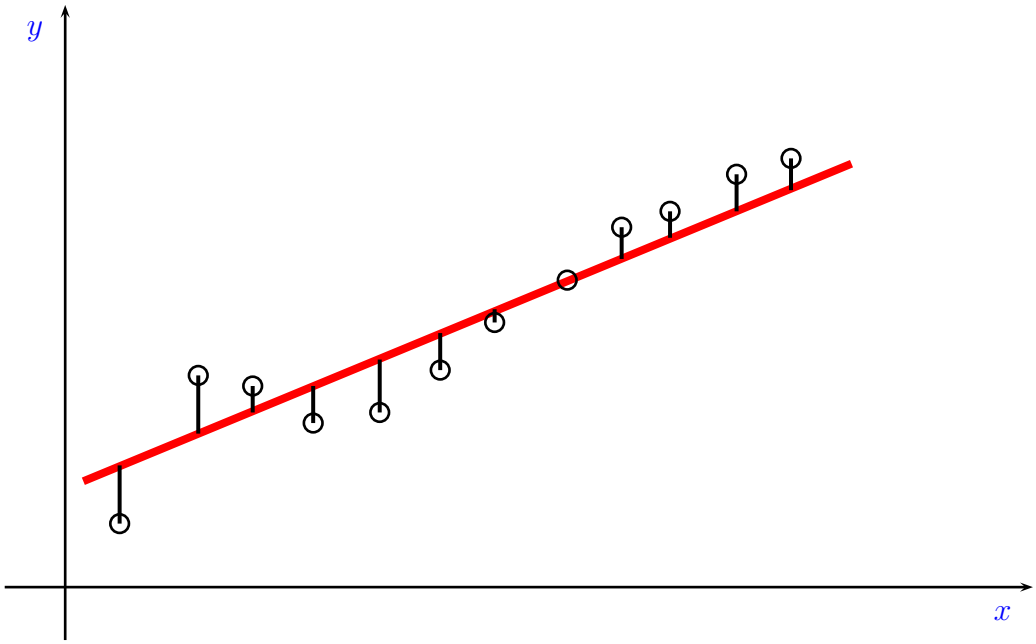
In questo caso si pone

$$\Phi(\alpha, \beta; x) = \alpha x + \beta, \qquad \alpha, \beta \in \mathbb{R} \tag{4.15}$$

e si cercano, tra tutte le possibili rette, i coefficienti $\alpha$ e $\beta$ che globalmente minimizzano la differenza

$$\Phi(\alpha, \beta; x_i) - y_i = \alpha x_i + \beta - y_i$$

La retta (4.15) che risolve tale problema viene detta Retta di regressione. Nella seguente figura sono evidenziate le quantità che devono essere globalmente minimizzate (i punti $(x_i, y_i)$ sono evidenziati con il simbolo $\circ$).




Un modo per minimizzare globalmente le distanze della retta dalle approssimazioni è quello di trovare i valori $\alpha, \beta$ che minimizzano la funzione:

$$\Psi(\alpha, \beta) = \sum_{i=0}^{n} (\alpha x_i + \beta - y_i)^2 .$$

Per questo si parla di problema ai minimi quadrati (si minimizza una somma di quantità elevate al quadrato).
Per determinare tali valori calcoliamo le derivate parziali rispetto alle incognite:

$$\frac{\partial \Psi}{\partial \alpha} = 2 \sum_{i=0}^{n} x_i (\alpha x_i + \beta - y_i)$$

$$\frac{\partial \Psi}{\partial \beta} = 2 \sum_{i=0}^{n} (\alpha x_i + \beta - y_i)$$

$$\begin{cases} \dfrac{\partial \Psi}{\partial \alpha} = 2 \displaystyle\sum_{i=0}^{n} x_i (\alpha x_i + \beta - y_i) = 0 \\ \dfrac{\partial \Psi}{\partial \beta} = 2 \displaystyle\sum_{i=0}^{n} (\alpha x_i + \beta - y_i) = 0 \end{cases}$$

$$\begin{cases} \sum_{i=0}^{n} x_i \left(\alpha x_i + \beta - y_i\right) = 0 \\ \\ \sum_{i=0}^{n} \left(\alpha x_i + \beta - y_i\right) = 0 \end{cases}$$

$$\begin{cases} \alpha \sum_{i=0}^{n} x_i^2 + \beta \sum_{i=0}^{n} x_i - \sum_{i=0}^{n} x_i y_i = 0 \\ \\ \alpha \sum_{i=0}^{n} x_i + (n+1)\beta - \sum_{i=0}^{n} y_i = 0. \end{cases}$$

Poniamo per semplicità

$$S_{xx} = \sum_{i=0}^{n} x_i^2 \qquad S_x = \sum_{i=0}^{n} x_i$$

$$S_{xy} = \sum_{i=0}^{n} x_i y_i \qquad S_y = \sum_{i=0}^{n} y_i.$$

Il sistema diventa

$$\begin{cases} S_{xx}\alpha + S_x\beta = S_{xy} \\ \\ S_x\alpha + (n+1)\beta = S_y \end{cases}$$

la cui soluzione è

$$\alpha = \frac{(n+1)S_{xy} - S_x S_y}{(n+1)S_{xx} - S_x^2}$$

$$\beta = \frac{S_y S_{xx} - S_x S_{xy}}{(n+1)S_{xx} - S_x^2}.$$

La tecnica della retta di regressione può essere applicata anche nel caso in cui la relazione tra le ascisse $x_i$ e le ordinate $y_i$ sia di tipo esponenziale, ovvero si può ipotizzare che la funzione che meglio approssima i dati sperimentali sia

$$\Phi(x) = Be^{Ax}, \qquad A, B \in \mathbb{R}, B > 0.$$

Ponendo

$$Y = \log \Phi(x)$$

risulta

$$Y = \log(Be^{Ax}) = Ax + \log B$$

ovvero

$$Y = \alpha x + \beta, \qquad \alpha = A,\ \beta = \log B$$

quindi si può applicare la tecnica della retta di regressione ai dati $(x_i, \log y_i)$ (osserviamo che affinchè il modello abbia senso i valori $y_i$ devono essere tutti strettamente positivi).

### 4.5.2 Approssimazione polinomiale ai minimi quadrati

Torniamo ora al problema di minimo (4.14). Poichè la funzione $f : \mathbb{R}^n \to \mathbb{R}$ condizione necessaria affinchè un punto sia di minimo è

$$\frac{\partial Q}{\partial a_k}(a_0, \ldots, a_m) = 0, \qquad k = 0, \ldots, m$$

e, poichè

$$Q(a_0, a_1, \ldots, a_m) = \sum_{i=0}^{n} \left(\Phi(a_0, a_1, \ldots, a_m; x_i) - y_i\right)^2$$

segue

$$\sum_{i=0}^{n} \left(\Phi(a_0, a_1, \ldots, a_m; x) - y_i\right) \frac{\partial \Phi}{\partial a_k}(a_0, \ldots, a_m; x) = 0$$

Si ottiene un sistema di $m+1$ equazioni (in generale non lineari) nelle $m+1$ incognite $a_0, \ldots, a_m$, detto sistema delle equazioni normali.
Vediamo ora come affrontare in generale tale problema. Consideriamo $m+1$ funzioni base $\varphi_0(x)$, $\varphi_1(x), \ldots,$ $\varphi_m(x)$ e supponiamo che la funzione $\Phi(x)$ abbia la seguente forma:

$$\Phi(a_0, \ldots, a_m; x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_m\varphi_m(x).$$

In questo caso la funzione $Q(a_0, \ldots, a_m)$ da minimizzare assume una forma particolare, infatti, osservato che

$$\begin{aligned} \Phi(a_0, \ldots, a_m; x) &= a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_m\varphi_m(x) \\ &= \begin{bmatrix} \varphi_0(x) & \ldots & \varphi_m(x) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_m \end{bmatrix} \end{aligned}$$

calcolando la funzione nei nodi $x_i$ :

$$\begin{bmatrix} \Phi(a_0,\ldots,a_m;x_0) \\ \Phi(a_0,\ldots,a_m;x_1) \\ \vdots \\ \Phi(a_0,\ldots,a_m;x_n) \end{bmatrix} = \underbrace{\begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \ldots & \varphi_m(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \ldots & \varphi_m(x_1) \\ \vdots & \vdots & & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \ldots & \varphi_m(x_n) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix}}_{\boldsymbol{\alpha}} = A\boldsymbol{\alpha}.$$

Ricaviamo ora l'espressione della funzione $Q(a_0,\ldots,a_m)$

$$\begin{aligned} Q(a_0,\ldots,a_m) &= \sum_{i=0}^{n} (\Phi(a_0,\ldots,a_m;x_i) - y_i)^2 \\ &= \left\| \begin{bmatrix} \Phi(a_0,\ldots,a_m;x_0) \\ \Phi(a_0,\ldots,a_m;x_1) \\ \vdots \\ \Phi(a_0,\ldots,a_m;x_n) \end{bmatrix} - \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \right\|_2^2 \\ &= \|A\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 \\ &= (A\boldsymbol{\alpha} - \boldsymbol{y})^T (A\boldsymbol{\alpha} - \boldsymbol{y}) \\ &= (\boldsymbol{\alpha}^T A^T - \boldsymbol{y}^T)(A\boldsymbol{\alpha} - \boldsymbol{y}) \\ &= \boldsymbol{\alpha}^T A^T A\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T A^T \boldsymbol{y} + \boldsymbol{y}^T \boldsymbol{y}. \end{aligned}$$

Calcolando le derivate parziali rispetto ad $a_i$ ed imponendo che siano uguali a zero risulta

$$\frac{\partial Q}{\partial a_i} = 0 \quad \Rightarrow \quad A^T A\boldsymbol{\alpha} - A^T \boldsymbol{y} = 0.$$

Il vettore dei coefficienti cercato è la soluzione del sistema di equazioni normali

$$A^T A\boldsymbol{\alpha} = A^T \boldsymbol{y} \tag{4.16}$$

che ammette un'unica soluzione se e solo se le colonne di $A$ sono linearmente indipendenti e che vale

$$\boldsymbol{\alpha} = (A^T A)^{-1} A^T \boldsymbol{y}.$$

Un caso particolare è il caso dell'approssimazione polinomiale ai minimi quadrati, in cui le funzioni base sono

$$\varphi_j(x) = x^j, \qquad j = 0, \ldots, m.$$

In tal caso

$$A = \begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^{n-1} & x_0^m \\ 1 & x_1 & x_1^2 & \ldots & x_1^{n-1} & x_1^m \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^{n-1} & x_n^m \end{bmatrix},$$

e il sistema ammette un'unica soluzione. Osserviamo infine che le dimensioni del sistema da risolvere dipendono solo dal numero di funzioni base scelte e non dal numero di dati a disposizione.
Per risolvere il sistema delle equazioni normali si può utilizzare un metodo alternativo alla fattorizzazione $LU$, ovvero la cosiddetta fattorizzazione di Cholesky

$$A = LL^T$$

dove $A$ indica la matrice dei coefficienti del sistema delle equazioni normali, $L$ è una matrice triangolare inferiore con elementi diagonali positivi. Le formule per il calcolo di $l_{ij}$ sono le seguenti:

$$l_{ij} = \frac{1}{l_{jj}} \left( a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk} \right) \qquad i = 1, \ldots, n, j < i$$

$$l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}$$

# Capitolo 5

# Quadratura e Derivazione Numerica

## 5.1 Formule di Quadratura di Tipo Interpolatorio

Siano assegnati due valori $a, b$, con $a < b$, ed una funzione $f$ integrabile sull'intervallo $(a, b)$. Il problema che ci poniamo è quello di costruire degli algoritmi numerici che ci permettano di valutare, con errore misurabile, il numero

$$I(f) = \int_a^b f(x)dx.$$

Diversi sono i motivi che possono portare alla richiesta di un algoritmo numerico per questi problemi.
Per esempio pur essendo in grado di calcolare una primitiva della funzione $f$, questa risulta così complicata da preferire un approccio di tipo numerico. Non è da trascurare poi il fatto che il coinvolgimento di funzioni, elementari e non, nella primitiva e la loro valutazione negli estremi $a$ e $b$ comporta comunque un'approssimazione dei risultati. Un'altra eventualità è che $f$ sia nota solo in un numero finito di punti o comunque può essere valutata in ogni valore dell'argomento solo attraverso una routine. In questi casi l'approccio analitico non è neanche da prendere in considerazione.
Supponiamo dunque di conoscere la funzione $f(x)$ nei punti distinti $x_0, x_1, \ldots,$

$x_n$ prefissati o scelti da noi, ed esaminiamo la costruzione di formule del tipo

$$\sum_{k=0}^{n} w_k f(x_k) \tag{5.1}$$

che approssimi realizzare $I(f)$.
Formule di tipo (5.1) si dicono di quadratura, i numeri reali $x_0, x_1, \ldots, x_n$ e $w_0, \ldots, w_n$ si chiamano rispettivamente nodi e pesi della formula di quadratura.
Il modo più semplice ed immediato per costruire formule di tipo (5.1) è quello di sostituire la funzione integranda $f(x)$ con il polinomio di Lagrange $L_n(x)$ interpolante $f(x)$ nei nodi $x_i$, $i = 0, \ldots, n$. Posto infatti

$$f(x) = L_n(x) + e(x)$$

dove $e(x)$ è la funzione errore, abbiamo:

$$\begin{aligned}\int_a^b f(x)dx &= \int_a^b [L_n(x) + e(x)]dx = \int_a^b L_n(x)dx + \int_a^b e(x)dx \\ &= \int_a^b \sum_{k=0}^{n} l_{nk}(x) f(x_k)dx + \int_a^b e(x)dx \\ &= \sum_{k=0}^{n} \left( \int_a^b l_{nk}(x)dx \right) f(x_k) + \int_a^b e(x)dx.\end{aligned}$$

Ponendo

$$w_k = \int_a^b l_{nk}(x)dx \qquad k = 0, 1, \ldots, n \tag{5.2}$$

e

$$R_{n+1}(f) = \int_a^b e(x)dx \tag{5.3}$$

otteniamo

$$I(f) \simeq \sum_{k=0}^{n} w_k f(x_k)$$

con un errore stabilito dalla relazione (5.3). Le formule di quadratura con pesi definiti dalle formule (5.2) si dicono interpolatorie. La quantità $R_{n+1}(f)$

prende il nome di Resto della formula di quadratura. Un utile concetto per misurare il grado di accuratezza con cui una formula di quadratura, interpolatoria o meno, approssima un integrale è il seguente.

**Definizione 5.1.1** *Una formula di quadratura ha grado di precisione $q$ se fornisce il valore esatto dell'integrale quando la funzione integranda è un qualunque polinomio di grado al più $q$ ed inoltre esiste un polinomio di grado $q+1$ tale che l'errore è diverso da zero.*

È evidente da questa definizione che ogni formula di tipo interpolatorio con nodi $x_0, x_1, \ldots, x_n$ ha grado di precisione almeno $n$. Infatti applicando una formula di quadratura costruita su $n+1$ nodi al polinomio $p_n(x)$, di grado $n$ si ottiene:

$$\int_a^b p_n(x)dx = \sum_{i=0}^{n} w_i p_n(x_i) + R_{n+1}(f)$$

e

$$R_{n+1}(f) = \int_a^b \omega_{n+1}(x)\frac{p_n^{(n+1)}(x)}{(n+1)!}dx \equiv 0$$

ovvero la formula fornisce il risultato esatto dell'integrale, quindi $q \geq n$.

## 5.2 Formule di Newton-Cotes

Suddividiamo l'intervallo $[a, b]$ in $n$ sottointervalli di ampiezza $h$, con

$$h = \frac{b-a}{n}$$

e definiamo i nodi

$$x_i = a + ih \qquad\qquad i = 0, 1, \ldots, n.$$

La formula di quadratura interpolatoria costruita su tali nodi, cioè

$$\int_a^b f(x)dx = \sum_{i=0}^{n} w_i f(x_i) + R_{n+1}(f)$$

è detta Formula di Newton-Cotes.
Una proprietà di cui godono i pesi delle formule di Newton-Cotes è la cosiddetta proprietà di simmetria. Infatti poichè i nodi sono a due a due simmetrici

rispetto al punto medio $c$ dell'intervallo $[a, b]$, cioè $c = (x_i + x_{n-i})/2$, per ogni $i$, tale proprietà si ripercuote sui pesi che infatti sono a due a due uguali, cioè $w_i = w_{n-i}$, per ogni $i$. Infatti

$$\begin{aligned} w_k &= \int_a^b \prod_{i=0, i\neq k}^{n} \frac{x - x_i}{x_k - x_i} dx \\ &= \int_a^b \prod_{i=0, i\neq k}^{n} \frac{x - 2c + x_{n-i}}{2c - x_{n-k} - 2c + x_{n-i}} dx \\ &= \int_a^b \prod_{i=0, i\neq k}^{n} \frac{x - 2c + x_{n-i}}{x_{n-i} - x_{n-k}} dx \\ &= \int_a^b \prod_{i=0, i\neq k}^{n} \frac{2c - x - x_{n-i}}{x_{n-k} - x_{n-i}} dx. \end{aligned}$$

Posto $t = 2c - x$ risulta

$$\begin{aligned} x = a \qquad &\Rightarrow \qquad t = 2c - a = b \\ x = b \qquad &\Rightarrow \qquad t = 2c - b = a \end{aligned}$$

quindi gli estremi di integrazione risultano invertiti, ma poichè $dt = -dx$ possiamo invertirli nuovamente, ottenendo

$$w_k = \int_a^b \prod_{i=0, i\neq k}^{n} \frac{t - x_{n-i}}{x_{n-k} - x_{n-i}} dt,$$

ponendo quindi nella produttoria $j = n - i$ risulta

$$w_k = \int_a^b \prod_{j=0, j\neq n-k}^{n} \frac{t - x_j}{x_{n-k} - x_j} dt = w_{n-k},$$

e la proprietà di simmetria dei pesi è dimostrata.
Descriviamo ora due esempi di formule di Newton-Cotes.

### 5.2.1 Formula dei Trapezi

Siano $x_0 = a$, $x_1 = b$ e $h = b - a$.

$$T_2 = w_0 f(x_0) + w_1 f(x_1)$$

$$w_0 = \int_a^b l_{1,0}(x)dx = \int_a^b \frac{x-x_1}{x_0-x_1}dx = \int_a^b \frac{x-b}{a-b}dx$$

$$= \frac{1}{a-b}\left[\frac{(x-b)^2}{2}\right]_{x=a}^{x=b} = \frac{h}{2}.$$

Poichè i nodi scelti sono simmetrici rispetto al punto medio $c = (a+b)/2$ è

$$w_1 = w_0 = \frac{h}{2}.$$

Otteniamo dunque la formula

$$T_2 = \frac{h}{2}\left[f(a) + f(b)\right].$$

che viene detta Formula dei Trapezi. Per quanto riguarda il resto abbiamo

$$R_2(f) = \frac{1}{2}\int_a^b (x-a)(x-b)f''(\xi_x)dx.$$

Prima di vedere come tale espressione può essere manipolata enunciamo il seguente teorema che è noto come teorema della media generalizzato.

**Teorema 5.2.1** *Siano $f, g : [a,b] \to \mathbb{R}$, funzioni continue con $g(x)$ a segno costante e $g(x) \neq 0$ per ogni $x \in ]a,b[$. Allora*

$$\int_a^b f(x)g(x)dx = f(\xi)\int_a^b g(x)dx, \qquad \xi \in [a,b]. \ \square$$

Poichè la funzione $(x-a)(x-b)$ è a segno costante segue:

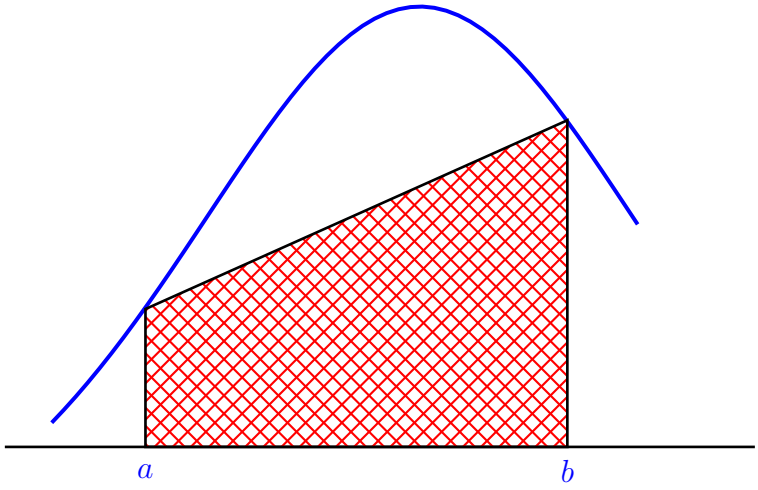
$$R_2(f) = \frac{1}{2}f''(\eta)\int_a^b (x-a)(x-b)dx$$

posto $x = a + ht$ otteniamo

$$R_2(f) = \frac{1}{2}f''(\eta)h^3\int_0^1 t(t-1)dt = -\frac{1}{12}h^3 f''(\eta).$$

L'errore della formula dipende dalla derivata seconda della funzione quindi il grado di precisione è pari a 1 in quanto solo se $f$ è un polinomio di grado

al più 1 essa fornisce il risultato esatto dell'integrale.
L'interpretazione geometrica della formula del trapezio è riassunta nella seguente figura, l'area tratteggiata (ovvero l'integrale della funzione viene approssimato attraverso l'area del trapezio che ha come basi i valori della funzione in $a$ e $b$ e come altezza l'intervallo $[a, b]$).




### 5.2.2 Formula di Simpson

Siano $x_0 = a$, $x_2 = b$ mentre poniamo $x_1 = c$, punto medio dell'intervallo $[a, b]$. Allora

$$S_3 = w_0 f(a) + w_1 f(c) + w_2 f(b).$$

Posto

$$h = \frac{b-a}{2}$$

abbiamo

$$w_0 = \int_a^b l_{2,0}(x)dx = \int_a^b \frac{(x-c)(x-b)}{(a-c)(a-b)}dx.$$

Effettuando il cambio di variabile $x = c + ht$ è facile calcolare quest'ultimo integrale, infatti

$$x = a \;\Rightarrow\; a = c + ht \;\Rightarrow\; a - c = ht \;\Rightarrow\; -h = ht \;\Rightarrow\; t = -1$$

e

$$x = b \;\Rightarrow\; b = c + ht \;\Rightarrow\; b - c = ht \;\Rightarrow\; h = ht \;\Rightarrow\; t = 1.$$

Inoltre $a-c=-h$ e $a-b=-2h$ mentre

$$x-c=c+ht-c=ht, \qquad x-b=c+ht-b=c-b+ht=-h+ht=h(t-1),$$

ed il differenziale $dx=hdt$ cosicchè

$$\begin{aligned} w_0 &= \int_a^b \frac{(x-c)(x-b)}{(a-c)(a-b)}dx = \int_{-1}^1 \frac{hth(t-1)}{(-h)(-2h)}hdt \\ &= \frac{h}{2}\int_{-1}^1 (t^2-t)dt = \frac{h}{2}\int_{-1}^1 t^2dt = \frac{h}{2}\left[\frac{t^3}{3}\right]_{-1}^1 = \frac{h}{3}. \end{aligned}$$

Per la proprietà di simmetria è anche

$$w_2 = w_0 = \frac{h}{3}$$

mentre possiamo calcolare $w_1$ senza ricorrere alla definizione. Infatti possiamo notare che la formula deve fornire il valore esatto dell'integrale quando la funzione è costante nell'intervallo $[a,b]$, quindi possiamo imporre che, prendendo $f(x)=1$ in $[a,b]$, sia

$$\int_a^b dx = b-a = \frac{h}{3}(f(a)+f(b)) + w_1 f(c) = \frac{2}{3}h + w_1$$

da cui segue

$$w_1 = b-a-\frac{2}{3}h = 2h-\frac{2}{3}h = \frac{4}{3}h.$$

Dunque

$$S_3 = \frac{h}{3}\left[f(a)+4f(c)+f(b)\right].$$

Questa formula prende il nome di Formula di Simpson. Per quanto riguarda l'errore si può dimostrare, e qui ne omettiamo la prova, che vale la seguente relazione

$$R_3(f) = -h^5\frac{f^{(4)}(\sigma)}{90} \qquad \sigma \in (a,b),$$

che assicura che la formula ha grado di precisione 3.

## 5.3 Formule di Quadratura Composte

Come abbiamo già avuto modo di vedere le formule di quadratura interpolatorie vengono costruite approssimando su tutto l'intervallo di integrazione la funzione integranda con un unico polinomio, quello interpolante la funzione sui nodi scelti. Per formule convergenti la precisione desiderata si ottiene prendendo $n$ sufficientemente grande. In tal modo comunque, per ogni fissato $n$, bisogna costruire la corrispondente formula di quadratura. Una strategia alternativa che ha il pregio di evitare la costruzione di una nuova formula di quadratura, e che spesso produce risultati più apprezzabili, è quella delle formule composte. Infatti scelta una formula di quadratura l'intervallo di integrazione $(a, b)$ viene suddiviso in $N$ sottointervalli di ampiezza $h$,

$$h = \frac{b-a}{N} \tag{5.4}$$

sicchè

$$\int_a^b f(x)dx = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x)dx$$

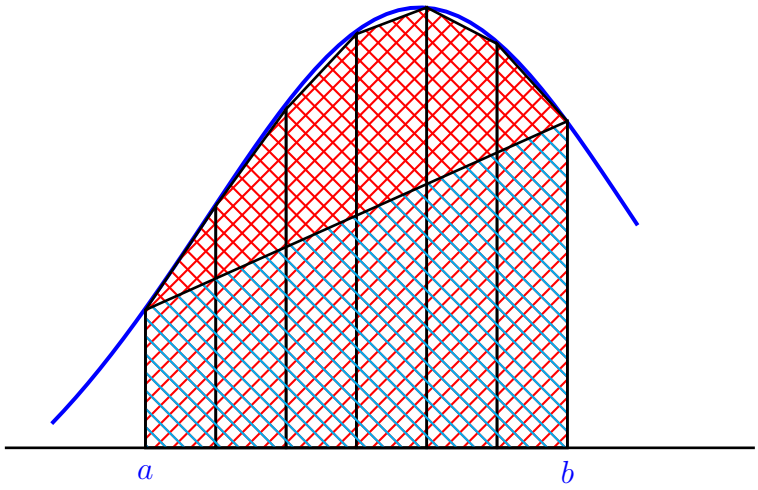
dove i punti $x_i$ sono:

$$x_i = a + ih \qquad i = 0, \ldots, N \tag{5.5}$$

quindi la formula di quadratura viene applicata ad ognuno degli intervalli $[x_i, x_{i+1}]$. Il grado di precisione della formula di quadratura composta coincide con il grado di precisione della formula da cui deriva. Descriviamo ora la Formula dei Trapezi Composta.

### 5.3.1 Formula dei Trapezi Composta

Per quanto visto in precedenza suddividiamo l'intervallo $[a, b]$ in $N$ sottointervalli, ognuno di ampiezza data da $h$, come in (5.4), e con i nodi $x_i$ definiti in (5.5). Applichiamo quindi in ciascuno degli $N$ intervalli $[x_i, x_{i+1}]$ la formula dei trapezi. Nella seguente figura sono evidenziate le aree che approssimano l'integrale utilizzando la formula dei trapezi semplice e quella composta.




Applicando la formula dei trapezi a ciascun sottointervallo si ottiene

$$\int_a^b f(x)dx = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x)dx = \sum_{i=0}^{N-1} \left[ \frac{h}{2} \left( f(x_i) + f(x_{i+1}) \right) - \frac{1}{12} h^3 f''(\eta_i) \right]$$

con $\eta_i \in (x_i, x_{i+1})$. Scrivendo diversamente la stessa espressione

$$\begin{aligned} \int_a^b f(x)dx &= \frac{h}{2} \left( f(x_0) + f(x_N) \right) + h \sum_{i=1}^{N-1} f(x_i) - \frac{1}{12} h^3 \sum_{i=0}^{N-1} f''(\eta_i) \\ &= \frac{h}{2} \left( f(x_0) + f(x_N) \right) + h \sum_{i=1}^{N-1} f(x_i) - \frac{1}{12} h^3 N f''(\eta) \end{aligned}$$

dove $\eta \in (a, b)$. L'esistenza di tale punto $\eta$ è garantito dal cosiddetto Teorema della media nel discreto applicato a $f''(x)$, che stabilisce che se $g(x)$ è una funzione continua in un intervallo $[a, b]$ e $\eta_i \in [a, b]$ $i = 1, N$, sono $N$ punti distinti, allora esiste un punto $\eta \in (a, b)$ tale che

$$\sum_{i=1}^{N} g(\eta_i) = N g(\eta).$$

Dunque la formula dei trapezi composta è data da:

$$T_C(h) = \frac{h}{2} \left( f(x_0) + f(x_N) \right) + h \sum_{i=1}^{N-1} f(x_i)$$

con resto

$$R_T = -\frac{1}{12}h^3 N f''(\eta) = -\frac{1}{12}\frac{(b-a)^3}{N^3}Nf''(\eta) = -\frac{1}{12}\frac{(b-a)^3}{N^2}f''(\eta).$$

Quest'ultima formula può essere utile per ottenere a priori una suddivisione dell'intervallo $[a, b]$ in un numero di intervalli che permetta un errore non superiore ad una prefissata tolleranza. Infatti

$$|R_T| \leq \frac{1}{12}\frac{(b-a)^3}{N^2}M, \qquad M = \max_{x\in[a,b]} |f''(x)|.$$

Imponendo che $|R_T| \leq \varepsilon$, precisione prefissata, segue

$$N_\varepsilon \geq \sqrt{\frac{(b-a)^3 M}{12\varepsilon}}. \tag{5.6}$$

Tuttavia questo numero spesso risulta una stima eccessiva a causa della maggiorazione della derivata seconda tramite $M$.

**Esempio 5.3.1** *Determinare il numero di intervalli cui suddividere l'intervallo di integrazione per approssimare*

$$\int_1^2 \log x \; dx$$

*con la formula dei trapezi composta con un errore inferiore a* $\varepsilon = 10^{-4}$.

La derivata seconda della funzione integranda è

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

quindi il valore di $M$ è 1. Dalla relazione (5.6) segue che

$$N_\varepsilon \geq \sqrt{\frac{1}{12\varepsilon}} = 29.$$

### 5.3.2 Formula di Simpson Composta

Per ottenere la formula di Simpson composta, si procede esattamente come per la formula dei trapezi composta. Suddividiamo $[a, b]$ in $N$ intervalli di ampiezza $h$, con $N$ numero pari. Allora

$$\begin{aligned}\int_a^b f(x)dx &= \sum_{i=0}^{\frac{N}{2}-1} \int_{x_{2i}}^{x_{2i+2}} f(x)dx \\ &= \sum_{i=0}^{\frac{N}{2}-1} \left[\frac{h}{3}\left(f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})\right) - \frac{h^5}{90} f^{(4)}(\eta_i)\right] \\ &= \frac{h}{3} \sum_{i=0}^{\frac{N}{2}-1} \left[f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})\right] - \frac{h^5 N}{180} f^{(4)}(\eta)\end{aligned}$$

dove $\eta_i \in (x_i, x_{i+1})$ e $\eta \in (a, b)$.
La formula di Simpson composta è dunque

$$\begin{aligned}S_C(h) &= \frac{h}{3} \sum_{i=0}^{\frac{n}{2}-1} \left[f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})\right] \\ &= \frac{h}{3} \left[f(x_0) + f(x_n) + 2\sum_{i=1}^{\frac{n}{2}-1} f(x_{2i}) + 4\sum_{i=0}^{\frac{n}{2}-1} f(x_{2i+1})\right]\end{aligned}$$

mentre la formula dell'errore è

$$R_S = -\frac{(b-a)^5}{180N^4} f^{(4)}(\eta)$$

Anche quest'ultima formula talvolta può essere utile per ottenere a priori una suddivisione dell'intervallo $[a, b]$ in un numero di intervalli che permetta un errore non superiore ad una prefissata tolleranza. Infatti

$$|R_S| \leq \frac{1}{180} \frac{(b-a)^5}{N^4} M, \qquad M = \max_{x \in [a,b]} |f^{(\mathrm{iv})}(x)|.$$

Imponendo che $|R_S| \leq \varepsilon$ segue

$$N_\varepsilon \geq \sqrt[4]{\frac{(b-a)^5 M}{180\varepsilon}}. \tag{5.7}$$

**Esempio 5.3.2** *Risolvere il problema descritto nell'esempio 5.3.1 applicando la formula di Simpson composta.*

La derivata quarta della funzione integranda è

$$f^{\text{iv}}(x) = -\frac{6}{x^4}$$

quindi è maggiorata da $M = 6$. Dalla relazione (5.7) segue che

$$N_\varepsilon \geq \sqrt[4]{\frac{6}{180\varepsilon}} > 4,$$

quindi $N_\varepsilon \geq 6$.

### 5.3.3 La formula del punto di mezzo

Sia $c$ il punto medio dell'intervallo $[a, b]$. Sviluppiamo $f(x)$ in serie di Taylor prendendo $c$ come punto iniziale:

$$f(x) = f(c) + f'(c)(x - c) + \frac{f''(\xi_x)}{2}(x - c)^2, \qquad \xi_x \in [a, b].$$

Integrando membro a membro

$$\begin{aligned}\int_a^b f(x)dx &= \int_a^b f(c)dx + f'(c)\int_a^b (x - c)dx + \int_a^b \frac{f''(\xi_x)}{2}(x - c)^2 dx \\ &= (b - a)f(c) + \int_a^b \frac{f''(\xi_x)}{2}(x - c)^2 dx.\end{aligned}$$

Poichè la funzione $x - c$ è dispari rispetto a $c$ il suo integrale nell'intervallo $[a, b]$ è nullo. La formula

$$\int_a^b f(x)dx \simeq (b - a)f(c)$$

prende appunto il nome di formula del punto di mezzo (o di midpoint).
Per quanto riguarda l'errore abbiamo

$$\begin{aligned}R(f) &= \int_a^b \frac{f''(\xi_x)}{2}(x - c)^2 dx \\ &= \frac{f''(\xi)}{2}\int_a^b (x - c)^2 dx.\end{aligned}$$

In questo caso la funzione $(x-c)^2$ è a segno costante quindi è stato possibile applicare il teorema 5.2.1. Calcoliamo ora l'integrale

$$\int_a^b (x-c)^2 dx = 2\int_c^b (x-c)^2 = \frac{2}{3}\left[(x-c)^3\right]_c^b = \frac{h^3}{12}$$

avendo posto $h = b-a$. L'espressione del resto di tale formula è quindi

$$R(f) = \frac{h^3}{24} f''(\xi).$$

Osserviamo che la formula ha grado di precisione 1, come quella dei trapezi, però richiede il calcolo della funzione solo nel punto medio dell'intervallo mentre la formula dei trapezi necessita di due valutazioni funzionali.

### 5.3.4 Formula del punto di mezzo composta

Anche in questo caso suddividiamo l'intervallo $[a,b]$ in $N$ intervallini di ampiezza $h$, con $N$ pari. Allora

$$\begin{aligned}
\int_a^b f(x)dx &= \sum_{i=0}^{\frac{N}{2}-1} \int_{x_{2i}}^{x_{2i+2}} f(x)dx \\
&= \sum_{i=0}^{\frac{N}{2}-1} \left[2hf(x_{2i+1}) + \frac{(2h)^3}{24} f''(\eta_i)\right] \\
&= 2h \sum_{i=0}^{\frac{N}{2}-1} f(x_{2i+1}) + \frac{Nh^3}{6} f''(\eta) \\
&= 2h \sum_{i=0}^{\frac{N}{2}-1} f(x_{2i+1}) + \frac{(b-a)^3}{6N^2} f''(\eta)
\end{aligned}$$

dove $\eta_i \in (x_{2i}, x_{2i+2})$ e $\eta \in (a,b)$. La formula del punto di mezzo composta è dunque

$$M_C(h) = 2h \sum_{i=0}^{\frac{N}{2}-1} f(x_{2i+1})$$

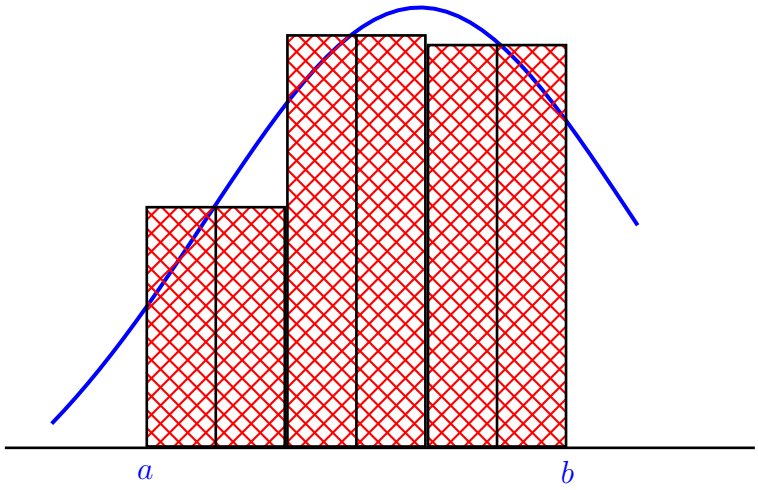



Figura 5.1: Formula del Punto di Mezzo Composta

mentre il resto è

$$R_M = \frac{(b-a)^3}{6N^2} f''(\eta). \tag{5.8}$$

Se $\varepsilon$ è la tolleranza fissata risulta

$$|R_M| \leq \frac{1}{6}\frac{(b-a)^3}{N^2} M, \qquad M = \max_{x\in[a,b]} |f''(x)|.$$

Imponendo che $|R_T| \leq \varepsilon$, precisione prefissata, segue

$$N_\varepsilon \geq \sqrt{\frac{(b-a)^3 M}{6\varepsilon}}. \tag{5.9}$$

Nella Figura 5.1 sono evidenziate le aree che approssimano l'integrale utilizzando la formula del punto di mezzo composta.

**Esempio 5.3.3** *Risolvere il problema descritto nell'esempio 5.3.1 applicando la formula di Simpson composta.*

La derivata seconda della funzione integranda è maggiorata da $M = 1$. Da (5.9) risulta

$$N_\varepsilon \geq \sqrt{\frac{1}{6\varepsilon}} > 40.$$

## 5.4 Derivazione numerica

Il problema della derivazione numerica consiste nell'approssimazione delle derivate di una funzione in un punto del dominio utilizzando opportune combinazioni lineari tra i valori assunti dalla funzione in un insieme discreto di punti. In questo paragrafo considereremo esclusivamente le formule per l'approssimazione della derivate prima e seconda di una funzione in una variabile, precisando che tali formule possono essere utilizzate anche per l'approssimazione discreta delle derivate parziali di una funzione in due variabili.
Supponiamo per ipotesi che $f \in \mathcal{C}^k([a,b])$ e suddividiamo l'intervallo di variabilità di $t$ in sottointervalli di ampiezza $h$. Consideriamo tre punti consecutivi appartenenti a tale reticolazione, rispettivamente $t_{n-1}$, $t_n$ e $t_{n+1}$ tali che

$$t_{n-1} = t_n - h, \qquad t_{n+1} = t_n + h.$$

Scriviamo lo sviluppo in serie di Taylor di $f(t_{n+1})$ prendendo come punto iniziale $t_n$:

$$f(t_{n+1}) = f(t_n) + hf'(t_n) + \frac{h^2}{2}f''(t_n) + \frac{h^3}{6}f'''(t_n) + \frac{h^4}{24}f^{\text{iv}}(\xi_n),\ \xi_n \in [t_n, t_{n+1}]$$

e procediamo in modo analogo per $f(t_{n-1})$:

$$f(t_{n-1}) = f(t_n) - hf'(t_n) + \frac{h^2}{2}f''(t_n) - \frac{h^3}{6}f'''(t_n) + \frac{h^4}{24}f^{\text{iv}}(\eta_n),\ \eta_n \in [t_{n-1}, t_n].$$

Sommiamo ora le due espressioni

$$f(t_{n+1}) + f(t_{n-1}) = 2f(t_n) + h^2 f''(t_n) + \frac{h^4}{24}\left[f^{\text{iv}}(\xi_n) + f^{\text{iv}}(\eta_n)\right]$$

ricavando

$$f''(t_n) = \frac{f(t_{n+1}) - 2f(t_n) + f(t_{n-1})}{h^2} - \frac{h^2}{24}\left[f^{\text{iv}}(\xi_n) + f^{\text{iv}}(\eta_n)\right]$$

e, trascurando l'ultimo termine, l'approssimazione della derivata seconda è:

$$f''(t_n) \simeq \frac{f(t_{n+1}) - 2f(t_n) + f(t_{n-1})}{h^2} \tag{5.10}$$

mentre si può provare che l'errore vale:

$$E(f''(t_n)) = -\frac{h^2}{12}f^{\text{iv}}(\xi), \qquad \xi \in [t_{n-1}, t_{n+1}].$$

Poniamoci il problema di approssimare derivata prima e procediamo nello stesso modo cioè scrivendo le serie di Taylor per $f(t_{n+1})$ e $f(t_{n-1})$ :

$$f(t_{n+1}) = f(t_n) + hf'(t_n) + \frac{h^2}{2}f''(t_n) + \frac{h^3}{6}f'''(\sigma_n),\ \sigma_n \in [t_n, t_{n+1}]$$

$$f(t_{n-1}) = f(t_n) - hf'(t_n) + \frac{h^2}{2}f''(t_n) - \frac{h^3}{6}f'''(\mu_n),\ \mu_n \in [t_{n-1}, t_n]$$

e questa volta sottraiamo la seconda dalla prima:

$$f(t_{n+1}) - f(t_{n-1}) = 2hf'(t_n) + \frac{h^3}{6}\left[f'''(\sigma_n) + f'''(\mu_n)\right]$$

ottenendo

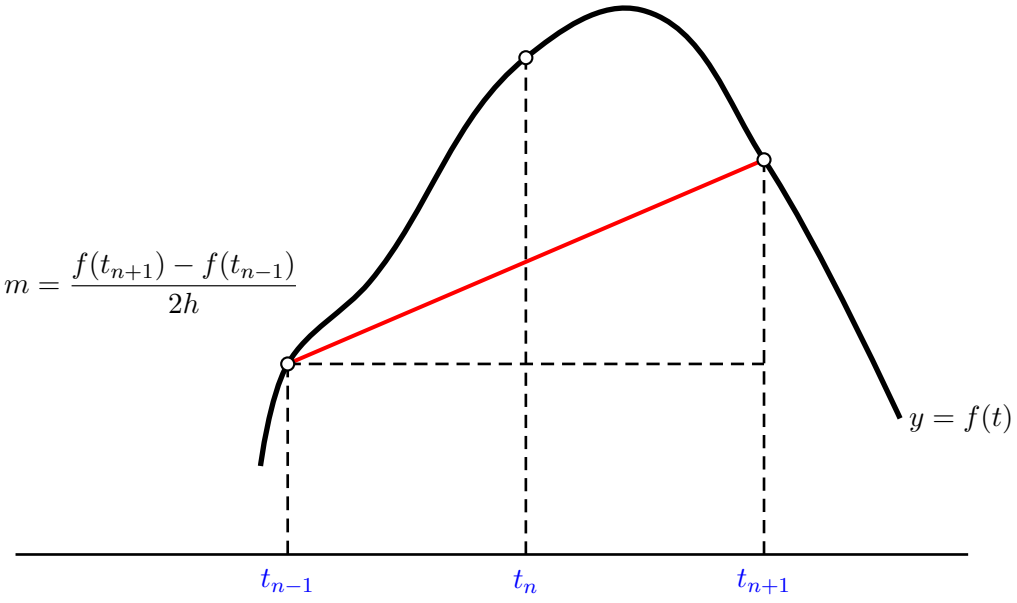
$$f'(t_n) = \frac{f(t_{n+1}) - f(t_{n-1})}{2h} - \frac{h^2}{12}\left[f'''(\sigma_n) + f'''(\mu_n)\right]$$

e, trascurando l'ultimo termine, l'approssimazione della derivata prima è:

$$f'(t_n) \simeq \frac{f(t_{n+1}) - f(t_{n-1})}{2h} \tag{5.11}$$

mentre si può provare che l'errore vale:

$$E(f'(t_n)) = -\frac{h^2}{6}f'''(\delta), \qquad \delta \in [t_{n-1}, t_{n+1}].$$



La formula (5.11) prende il nome di formula alle differenze centrali. Osserviamo che sia per questa che per l'approssimazione numerica per la derivata seconda l'errore dipende da $h^2$, sono formule cioè *del secondo ordine*. Vediamo ora altre due approssimazioni per la derivata prima. Infatti possiamo anche scrivere:

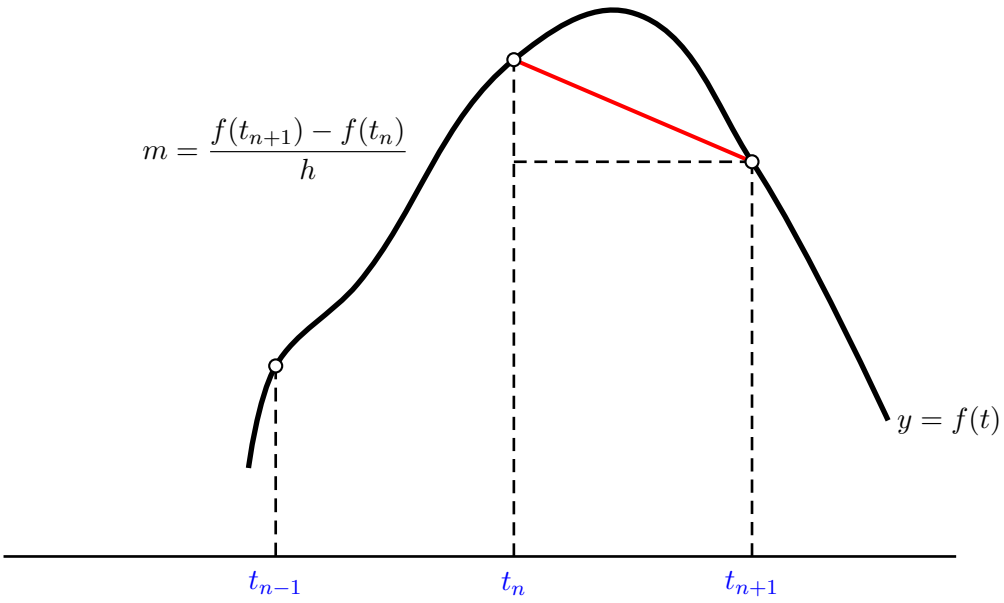
$$f(t_{n+1}) = f(t_n) + hf'(t_n) + \frac{h^2}{2}f''(\xi_n), \qquad \xi_n \in [t_n, t_{n+1}]$$

da cui si ricava immediatamente la formula alle differenze in avanti:

$$f'(t_n) \simeq \frac{f(t_{n+1}) - f(t_n)}{h} \tag{5.12}$$

con errore

$$E(f'(t_n)) = -\frac{h}{2}f''(\xi_n).$$




Analogamente si ricava

$$f(t_{n-1}) = f(t_n) - hf'(t_n) + \frac{h^2}{2}f''(\mu_n), \qquad \mu_n \in [t_{n-1}, t_n]$$

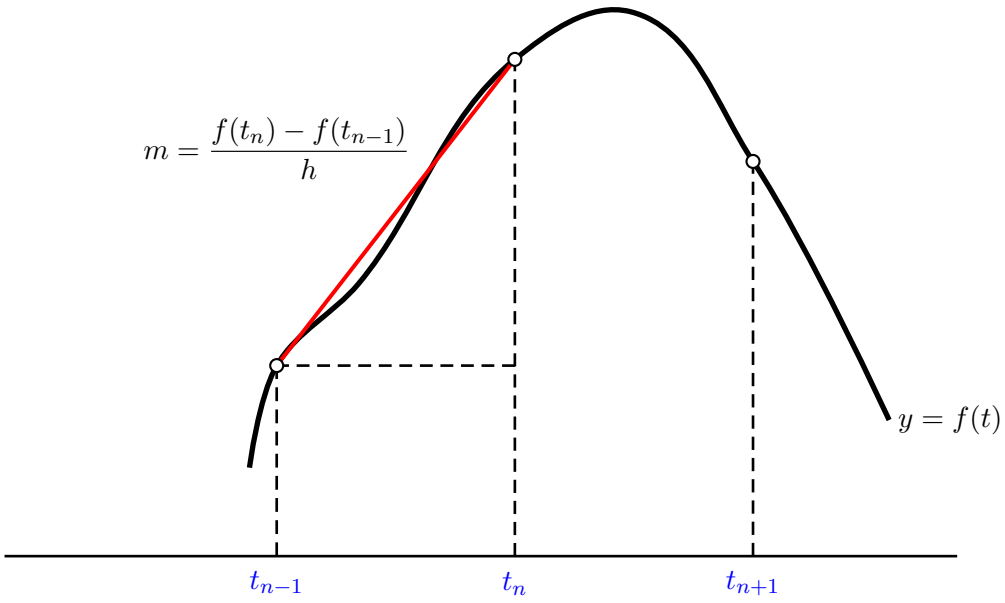
da cui si ricava immediatamente la formula alle differenze all'indietro:

$$f'(t_n) \simeq \frac{f(t_n) - f(t_{n-1})}{h} \tag{5.13}$$

con errore

$$E(f'(t_n)) = -\frac{h}{2} f''(\mu_n).$$

Queste due formule hanno ordine 1, quindi sono meno precise rispetto alla formula alle differenze centrali, tuttavia hanno il pregio di poter essere applicate quando la funzione è discontinua (oppure non è definita) a destra o a sinistra di $t_n$.



# Capitolo 6

# Esercitazioni di laboratorio MatLab

## Esercitazione 1

Argomento: **Introduzione al MATLAB**
Scopo: Eseguire alcune semplici istruzioni MatLab e imparare l'uso della grafica.

Scopo di questa prima esercitazione è quello di iniziare a conoscere l'ambiente MatLab ed in particolare le istruzioni per la manipolazione di matrici e vettori, le funzioni predefinite, le istruzioni per la grafica e quelle di iterazione e selezione.
Una volta lanciato il programma iniziare la sessione di lavoro assegnando alcuni vettori o matrici:

```
>> x = [1 4 5 3 -4 5]
>> y = [-1; 0; -5; 13; 4; -5]
>> length(x)
>> length(y)
>> z=x+y
>> z=x'+y
>> a=x*y
>> a=y*x
>> x=[x 10]
>> n=length(x)
>> x=x(n:-1:1)
```

Dal risultato delle operazioni precedenti si è potuto osservare che la somma dei due vettori non è consentita a meno che questi non abbiano esattamente le stesse dimensioni (cioè siano due vettori riga o colonna della stessa lunghezza). Anche per il prodotto le dimensioni devono essere compatibili. Un vettore riga (di dimensione $1 \times n$) può essere moltiplicato per un vettore colonna (di dimensione $n \times 1$) e dà come risultato un valore scalare. Un vettore colonna (di dimensione $n \times 1$) può essere moltiplicato per un vettore riga (di dimensione $1 \times n$) e produce come risultato una matrice quadrata di dimensione $n$. In ultimo osserviamo che l'ultima istruzione di questo blocco inverte gli elementi del vettore.

```
>> a=max(x)
>> [a i]=max(x)
>> a1=min(x)
>> [a k]=min(x)
>> sort(x)
```

In questo caso possiamo osservare come le funzioni predefinite `max` e `min` possano dare due tipi di output diversi, cioè possono fornire solo il valore del massimo (o del minimo) del vettore ma anche l'indice della componente massima (o minima).
Vediamo ora alcune istruzioni che riguardano le matrici.

```
>> A = [1 4 5 3; 0 1 -4 5; 3 4 5 6; -1 0 1 9 ...
; 0 7 6 -9]
>> A(1:3,4)
>> A(2,2:4)
>> A(:,4)
>> A(5,:)
>> A(1:3,2:4)
>> A([1 5],:)=A([5 1],:)
>> [m,n]=size(A)
>> x=max(A)
>> x=max(max(A))
>> B=cos(A)
```

La prima istruzione di questo blocco consiste nell'assegnazione di una matrice $5 \times 4$ alla variabile $A$. Si osservi la funzione dei tre punti che servono a

spezzare su più righe istruzioni troppo lunghe. Nelle altre possiamo osservare come la cosiddetta notazione due punti permetta di visualizzare in modo compatto porzioni di righe o di colonne, o intere sottomatrici. La sesta istruzione permette di poter scambiare simultaneamente due righe di una stessa matrice (istruzione analoga vale anche per le colonne) senza l'ausilio di vettori ausiliari. Va infine osservato cosa succede se si applica una funzione di tipo vettoriale (in questo caso `max`) ad una matrice: il risultato è un vettore, che (in questo caso) contiene i massimi delle colonne di $A$. Applicandolo due volte si ottiene come risultato il massimo elemento della matrice.
Vediamo ora di scrivere la seguente funzione che calcoli le radici del polinomio di secondo grado

$$ax^2 + bx + c$$

che indichiamo con `x1` e `x2`. Come noto, posto

$$\Delta = b^2 - 4ac$$

allora

$$x_1 = \frac{-b + \sqrt{\Delta}}{2a}, \qquad x_2 = \frac{-b - \sqrt{\Delta}}{2a}.$$

```
function [x1,x2]=radici(a,b,c)
%
% Sintassi [x1,x2]=radici(a,b,c)
%
% Calcola le radici di un polinomio di secondo grado
%
Delta = b^2-4*a*c;
x1 = (-b+sqrt(Delta))/(2*a);
x2 = (-b-sqrt(Delta))/(2*a);
return
```

Applichiamo ora la funzione al polinomio che ammette come radici i due numeri $x_1 = 10^7$ e $x_2 = 10^{-7}$. In questo caso i valori dei coefficienti $a$, $b$ e $c$ sono:

$$a = 1, \qquad b = -(10^7 + 10^{-7}), \qquad c = 1.$$

Scriviamo pertanto le seguenti istruzioni:

```
>> a = 1;
>> b = -(10^7+10^(-7));
>> c = 1;
>> [x1,x2] = radici(a,b,c);
```

Adesso provvediamo a modificare la funzione nel seguente modo:

```
function [x1,x2]=radici1(a,b,c)
%
% Sintassi [x1,x2]=radici1(a,b,c)
%
% Calcola le radici di un polinomio di secondo grado
%
Delta = b^2-4*a*c;
x1 = (-b-sign(b)*sqrt(Delta))/(2*a);
x2 = c/x1;
return
```

Scriviamo pertanto le seguenti istruzioni:

```
>> a = 1;
>> b = -(10^7+10^(-7));
>> c = 1;
>> [r1,r2] = radici1(a,b,c);
```

Osserviamo la differenza tra i valori calcolati.
Proviamo ora a tracciare il grafico di una funzione. In MatLab ciò può essere fatto in molti modi diversi, vediamone solo i più semplici. Innanzitutto scegliamo una funzione, per esempio:

$$f(x) = \sin^2(x)\cos(x) + (\sin(e^x))^2 + 1$$

e decidiamo di tracciarne il grafico nell'intervallo $[0, 2\pi]$. Come è noto un grafico in MatLab non è nient'altro se non una spezzata che congiunge un insieme discreto di punti del piano. Per prima cosa dobbiamo scegliere nell'intervallo un certo numero di punti equidistanti, per esempio 100 punti, utilizzando la seguente istruzione:

```
>> x=linspace(0,2*pi,100);
```

Adesso dobbiamo calcolare il valore della funzione $f(x)$ nel vettore delle ascisse appena assegnato. Il modo più semplice è quello di utilizzare una variabile di tipo stringa per memorizzare la funzione attraverso la funzione `inline`:

```
>> funz=inline('(sin(x).^2).*cos(x)+(sin(exp(x))).^2+1')
```

Osserviamo che quando alla variabile `funz` viene assegnata una funzione le operazioni che compaiono nella stringa devono essere considerate come se fossero applicate a vettori.
A questo punto per calcolare il valore della funzione nel vettore `x` si può utilizzare la funzione `feval`:

```
>> y=feval(funz,x);
```

A questo punto si può procedere a tracciare il grafico della funzione:

```
>> plot(x,y,'b-');
```

Il grafico è stato tracciato in blu a tratto continuo, ma possiamo anche variare il colore e il tipo di tratto, proviamo le seguenti istruzioni:

```
>> plot(x,y,'y--');
>> plot(x,y,'r:');
>> plot(x,y,'go');
```

Un secondo modo per tracciare il grafico è quello di utilizzare la funzione predefinita `fplot`. In questo caso il modo di procedere è lo stesso tranne per la definizione del vettore delle ascisse che non va assegnato:

```
>> fplot(funz,[0 2*pi]);
```

Infatti i parametri di tale funzione sono solo la stringa contenente la funzione e l'intervallo di variabilità delle ascisse.
Tracciando i diversi grafici si è potuto osservare che ogni volta che viene aperta una nuova figura la precedente viene cancellata. Per poter tracciare più grafici su una stessa figura va utilizzata l'opzione `hold on` nel seguente modo:

```
>> fplot(funz,[0 2*pi]);
>> hold on
>> g=inline('2+sin(x).*cos(x)');
>> y1=feval(g,x);
>> plot(x,y1);
```

Una volta che tale opzione è eseguita essa rimane attiva per tutta la sessione di lavoro. Questo vuol dire che tutti i grafici che saranno tracciati successivamente si andranno a sovrapporre sulla stessa figura. Per disattivare tale opzione è sufficiente l'istruzione

```
>> hold off
```

# Esercitazione 2

Argomento: **Sistemi triangolari**
Scopo: Implementare i metodi di sostituzione in avanti e all'indietro per sistemi triangolari inferiori e superiori.

```
function x=indietro(A,b)
%
% Sintassi x=indietro(A,b)
%
% Risolve un sistema triangolare superiore utilizzando
% il metodo di sostituzione all'indietro
%
% Parametri di input:
% A = Matrice triangolare superiore
% b = Vettore colonna
%
% Parametri di output:
% x = Vettore soluzione
%
n=length(b);
x=zeros(n,1);
if abs(A(n,n))<eps
     error('La matrice A e'' singolare ');
end
x(n)=b(n)/A(n,n);
for k=n-1:-1:1
     x(k)=b(k);
     for i=k+1:n
          x(k)=x(k)-A(k,i)*x(i);
     end
     if abs(A(k,k))<eps
          error('La matrice A e'' singolare ');
     else
          x(k)=x(k)/A(k,k);
     end
end
return
```

**Esempio di applicazione:** Vedere la routine `gauss.m` in una delle prossime esercitazioni.

**Possibili modifiche:**
La routine appena descritta risolve un sistema triangolare superiore. Osserviamo innanzitutto che se viene incontrato un elemento diagonale più piccolo, in modulo, della precisione di macchina allora l'algoritmo segnala un errore. Si può inoltre osservare che la routine potrebbe essere scritta in modo più compatto utilizzando la notazione : del MatLab. Infatti il ciclo descritto dalla variabile `i` si potrebbe sostituire con un'unica istruzione:

```
x(k)=b(k)-A(k,k+1:n)*x(k+1:n);
```

Per completezza vediamo anche l'implementazione del metodo di sostituzione in avanti per matrici triangolari inferiori.

```
function x=avanti(A,b)
%
% Sintassi x=avanti(A,b)
%
% Risolve un sistema triangolare inferiore utilizzando
% il metodo di sostituzione in avanti
%
% Parametri di input:
% A = Matrice triangolare inferiore
% b = Vettore colonna
%
% Parametri di output:
% x = Vettore soluzione
%
n=length(b);
x=zeros(n,1);
if abs(A(1,1))<eps
     error('La matrice A e'' singolare ');
end
x(1)=b(1)/A(1,1);
for k=2:n
     x(k)=b(k)-A(k,1:k-1)*x(1:k-1);
```

```
     if abs(A(k,k))<eps
          error('La matrice A e'' singolare ');
     else
     x(k)=x(k)/A(k,k);
     end
end
return
```

# Esercitazione 3

Argomento: **Il metodo di eliminazione di Gauss**
Scopo: Risoluzione di un sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ utilizzando il metodo di eliminazione di Gauss senza strategie di pivoting.

```
function x=gauss(A,b);
%
% Sintassi x=gauss(A,b)
%
% Risolve un sistema lineare utilizzando il
% metodo di eliminazione di Gauss
%
% Parametri di input:
% A = Matrice dei coefficienti
% b = Vettore dei termini noti
%
% Parametri di output:
% x = Vettore soluzione
%
[m,n]=size(A);
if m~=n
     error('Metodo non applicabile');
end
if length(b)~=n
     error('Metodo non applicabile');
end
for k=1:n
     if abs(A(k,k))<eps
          error('Elemento pivotale nullo ');
     end
     for i=k+1:n
          A(i,k)=A(i,k)/A(k,k);
          for j=k+1:n
               A(i,j)=A(i,j)-A(k,j)*A(i,k);
          end
          b(i)=b(i)-b(k)*A(i,k);
     end
```

```
end
x=indietro(A,b);
return
```

**Esempi di applicazione:** Per verificare il funzionamento dell'algoritmo si può applicare ad un sistema lineare avente una matrice dei coefficienti a predominanza diagonale per colonne.

```
>> A=[6 4 1 0;-1 8 1 1;3 0 6 -3;1 -2 1 7]
>> b=[1;2;3;4]
>> x=gauss(A,b)
```

Per verificare invece che il metodo di Gauss non funziona se la matrice dei coefficienti ammette un minore principale uguale a zero si può applicarlo in questa circostanza e verificare che la routine appena scritta segnala tale circostanza.

```
>> A=[1 1 2 1 0;2 1 3 1 -4;-1 -1 -2 3 0;4 2 -1 1 0;5 2 -2 1 7]
>> b=[1;2;3;4;5]
>> x=gauss(A,b)
```

Ci sono casi in cui il metodo di eliminazione di Gauss può fornire una soluzione del sistema molto diversa da quella teorica. Vediamo il seguente esempio: scegliamo come matrice dei coefficienti una cosiddetta matrice di Hilbert, definita nel seguente modo:

$$h_{ij} = \frac{1}{i+j-1} \qquad i,j = 1,\ldots,n.$$

Per esempio se $n = 4$ la matrice sarebbe

$$H = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{bmatrix}.$$

Proviamo ora ad applicare il metodo di Gauss ad un sistema di dimensione 15 avente come matrice dei coefficienti quella di Hilbert e come soluzione il vettore avente tutte le componenti uguali a 1 e confrontiamo la soluzione che ci fornisce il metodo di Gauss con quella teorica.

```
>> clear
>> format long e
>> n=15;
>> A=hilb(n);
>> x=ones(n,1);
>> b=A*x;
>> y=gauss(A,b)
>> norm(x-y,'inf')
```

Le prime due istruzioni servono rispettivamente a cancellare tutte le variabili presenti nell'area di lavoro del MatLab e a scrivere i valori delle variabili in formato esponenziale lungo, cioè con 15 cifre decimali. La funzione `hilb(n)` assegna ad una variabile la matrice di Hilbert della dimensione indicata. Il vettore `b` viene assegnato in modo tale che la soluzione del sistema, cioè il vettore colonna `x`, sia nota. Nella variabile `y` viene memorizzata la soluzione del sistema calcolata utilizzando il metodo di Gauss. L'ultima istruzione serve a dare una misura della differenza tra la soluzione teorica del sistema e quella calcolata utilizzando la funzione `norm` che, in questo caso, misura la norma infinito della differenza tra i due vettori, cioè il massimo valore assoluto del vettore differenza `x-y`.
Argomento: **Il metodo di eliminazione di Gauss con pivot parziale**
Scopo: Risoluzione di un sistema lineare utilizzando il metodo di eliminazione di Gauss con strategia di pivoting parziale.

```
function x=gausspiv(A,b);
%
% Sintassi x=gausspiv(A,b)
%
% Risolve un sistema lineare utilizzando il metodo
% di eliminazione di Gauss con pivoting parziale
%
% Parametri di input:
% A = Matrice dei coefficienti
% b = Vettore dei termini noti
%
% Parametri di output:
% x = Vettore soluzione
```

```
%
[m,n]=size(A);
if m~=n
     error('Metodo non applicabile');
end
if length(b)~=n
     error('Metodo non applicabile');
end
for k=1:n
     [pivot indice]=max(abs(A(k:n,k)));
     riga=indice+k-1;
     if riga~=k
          A([riga k],:)=A([k riga],:);
          b([riga k])=b([k riga]);
     end
     if abs(A(k,k))<eps
          error('Elemento pivotale nullo ');
     end
     for i=k+1:n
          A(i,k)=A(i,k)/A(k,k);
          for j=k+1:n
               A(i,j)=A(i,j)-A(k,j)*A(i,k);
          end
          b(i)=b(i)-b(k)*A(i,k);
     end
end
x=indietro(A,b);
```

**Esempi di applicazione:** Si può applicare la funzione ad un sistema lineare la cui matrice dei coefficienti ha un minore principale uguale a zero e verificare che in questo caso essa fornisce la soluzione del sistema.

```
>> A=[1 1 2 1 0;2 1 3 1 -4;-1 -1 -2 3 0;4 2 -1 1 0;5 2 -2 1 7]
>> b=[1;2;3;4;5]
>> x=gausspiv(A,b)
```

# Esercitazione 4

Argomento: **Il polinomio interpolante di Lagrange**
Scopo: Tracciare il grafico del polinomio di Lagrange che interpola un insieme discreto di dati.

```
function z=lagrange(x,y,x1)
%
% Sintassi z=lagrange(x,y,x1)
%
% Calcola il polinomio interpolante di Lagrange
% nei punti memorizzati nel vettore x1
%
% Parametri di input:
% x = vettore dei nodi
% y = vettore delle ordinate
% x1= vettore delle ascisse
%
% Parametri di output:
% z = vettore delle ordinate del polinomio interpolante
%
n=length(x);
m=length(x1);
z=zeros(1,m);
for i=1:m
     for j=1:n
          p=1;
          for k=1:n
               if j~=k
                    p=p*(x1(i)-x(k))/(x(j)-x(k));
               end
          end
          z(i)=z(i)+y(j)*p;
     end
end
return
```

**Esempio di applicazione:** Per eseguire l'algoritmo appena scritto è necessario assegnare i vettori `x` e `y`, cioè i nodi e le ordinate dei dati da interpolare. Vediamo un esempio.

```
>> x=[-4; -3; 0; 1; 4; 5];
>> y=[-1; 3; 4; 5; -3; 7];
>> a=min(x);
>> b=max(x);
>> x1=linspace(a,b,200);
>> y1=lagrange(x,y,x1);
>> plot(x1,y1,'r',x,y,'o')
```

**Esempio di applicazione:** Osserviamo che la funzione `lagrange` è utilizzata anche nella successiva esercitazione.

# Esercitazione 5

Argomento: **Il fenomeno di Runge**
Scopo: Visulizzazione grafica del fenomeno di Runge scegliendo come funzione da interpolare:

$$f(x) = \frac{1}{1+x^2}.$$

```
function runge(a,b,n)
%
% Sintassi runge(a,b,n)
%
% Visualizza il fenomeno di Runge
%
% Parametri di input:
% a = estremo sinistro dell'intervallo
% b = estremo destro dell'intervallo
% n = numero di nodi
%
% Grafico:
% Curva rossa = funzione interpolata
% Curva blu = polinomio interpolante
% Cerchi neri = nodi dell'interpolazione
%
f=inline('1./(x.^2+1)');
x=linspace(a,b,n);
y=feval(f,x);
x1=linspace(a,b,100);
y2=feval(f,x1);
y1=lagrange(x,y,x1);
plot(x,y,'ko',x1,y2,'r',x1,y1,'b')
return
```

**Esempio di applicazione:** Per eseguire l'algoritmo appena scritto è necessario assegnare solo gli estremi dell'intervallo e il numero dei nodi. È conveniente assegnare prima gli estremi ed eseguire la funzione con un numero crescente di nodi per osservare meglio le oscillazioni del polinomio interpolante verso gli estremi dell'intervallo.

```
>> a=-5;
>> b=5;
>> runge(a,b,5)
>> runge(a,b,10)
>> runge(a,b,20)
>> runge(a,b,30)
>> runge(a,b,40)
```

Si può modificare il codice scegliendo i nodi coincidenti con gli zeri del polinomio di Chebyshev di grado $n$, $T_n(x)$ ed interpolando la stessa funzione di Runge:

```
function rungeCheb(a,b,n)
%
% Sintassi rungeCheb(a,b,n)
%
% Traccia il grafico del polinomio interpolante la funzione di Runge
% utilizzando i nodi di Chebyshev
%
% Parametri di input:
% a = estremo sinistro dell'intervallo
% b = estremo destro dell'intervallo
% n = numero di nodi
%
% Grafico:
% Curva rossa = funzione interpolata
% Curva blu = polinomio interpolante
% Cerchi neri = nodi dell'interpolazione
%
f=inline('1./(x.^2+1)');
x=(a+b)/2+((b-a)/2)*cos((2*[0:n-1]*pi+1)/(2*n));
y=feval(f,x);
x1=linspace(a,b,100);
y2=feval(f,x1);
y1=lagrange(x,y,x1);
plot(x,y,'ko',x1,y2,'r',x1,y1,'b')
return
```

# Esercitazione 6

Argomento: **Il metodo delle successive bisezioni**
Riferimenti teorici: Capitolo 6, Paragrafo 6.2
Scopo: Implementare il metodo delle successive bisezioni per la soluzione di equazioni non lineari.

```
function [alfa,iter]=bisez(f,a,b,epsilon)
%
% Sintassi [alfa,iter]=bisez(f,a,b,epsilon)
%
% Calcola la radice della funzione f
% con il metodo delle bisezioni
%
% Parametri di input:
% f = stringa contenente il nome della funzione
% a = estremo sinistro dell'intervallo
% b = estremo destro dell'intervallo
% epsilon = tolleranza prefissata
%
% Parametri di output:
% alfa = approssimazione della radice
% iter = numero di iterate occorse
%
if feval(f,a)*feval(f,b)>0
     disp('Il metodo non converge')
     return
end
c=(a+b)/2;
iter=1;
fc=feval(f,c);
while abs(fc)>epsilon | (b-a)>epsilon
     if fc*feval(f,a)<0
          b=c;
     else
          a=c;
     end
```

```
    iter=iter+1;
    c=(a+b)/2;
    fc=feval(f,c);
end
alfa=c;
```

**Esempio di applicazione:** La funzione richiede in ingresso la variabile stringa dove è memorizzata la funzione, gli estremi dell'intervallo e la precisione voluta. Come esempio si può considerare la funzione

$$f(x) = x - e^{-x}$$

e prendere come intervallo iniziale $[0, 1]$ e fissare come precisione $\varepsilon = 10^{-8}$.

```
>> format long e
>> f=inline('x-exp(-x)')
>> a=0;
>> b=1;
>> epsilon=1e-8;
>> [alfa, iter]=bisez(f,a,b,epsilon)
```

**Possibili modifiche:**
La funzione appena descritta prevede come parametri di output un'approssimazione della radice e il numero di iterate, tuttavia quest'ultimo può essere calcolato a priori tenendo conto che, una volta nota la precisione richiesta il numero di iterate necessario per calcolare l'approssimazione è

$$k > \log_2 \left( \frac{b-a}{\varepsilon} \right).$$

Si potrebbe calcolare tale valore di $k$ e trasformare il ciclo `while` in un ciclo `for` e vedere se i risultati del metodo sono gli stessi nei due casi.

Argomento: **Il metodo di Newton-Raphson**
Riferimenti teorici: Capitolo 6, Paragrafo 6.6
Scopo: Implementare il metodo di Newton-Raphson per la soluzione di equazioni non lineari.

```
function [x1,iter]=newtraph(f,f1,x0,epsilon)
%
% Sintassi [x1,iter]=newtraph(f,f1,x0,epsilon)
%
% Calcola la radice della funzione f con il
% il metodo di Newton-Raphson
%
% Parametri di input:
% f = funzione della quale si vuole approssimare la radice
% f1 = derivata prima di f
% x0 = approssimazione iniziale
% epsilon = tolleranza prefissata
%
% Parametri di output:
% x1 = approssimazione della radice
% iter = numero di iterate
%
iter=0;
err=2*epsilon;
ff=feval(f,x0);
maxiter=100;
while err>epsilon | abs(ff)>epsilon
     x1=x0-ff/feval(f1,x0);
     err=abs(x1-x0);
     iter=iter+1;
     if iter>maxiter
          error('Il metodo non converge ')
     else
     x0=x1;
     ff=feval(f,x0);
     end
end
```

**Esempio di applicazione:** La funzione richiede in ingresso le variabili di tipo stringa dove sono memorizzate la funzione e la sua derivata prima, l'approssimazione iniziale $x_0$, la precisione voluta e il numero massimo di iterate. Per esempio si può considerare la funzione

$$f(x) = x^3 - 3x + 2 \qquad\qquad f'(x) = 3x^2 - 3$$

prendendo come approssimazione iniziale prima $x_0 = -2.5$ e poi $x_0 = 1.4$, e fissando come precisione $\varepsilon = 10^{-8}$.

```
>> format long e
>> f=inline('x.^3-3*x+2')
>> f1=inline('3*x.^2-3')
>> x0=-2.5;
>> epsilon=1e-8;
>> maxiter=100;
>> [alfa0,iter0]=newtraph(f,f1,x0,epsilon)
>> x0=1.4;
>> [alfa1,iter1]=newtraph(f,f1,x0,epsilon)
```

Dai risultati emerge il diverso comportamento del metodo per le due diverse radici: infatti la radice $-2$ è semplice e il metodo di Newton-Raphson converge con ordine 2 (quindi più rapidamente), mentre per la radice doppia 1 la convergenza è più lenta poichè l'ordine è 1.