# Analisi numerica Appunti del corso

Appunti scritti dagli studenti degli anni precedenti

versione corretta per l'anno accademico 2017/2018

## 1 Premessa

Questi appunti del corso di Analisi numerica sono stati scritti negli ultimi anni con il contributo di diversi studenti. Pertanto, essi vanno presi per quello che sono: appunti del corso. Per una completa preparazione all'esame devono essere integrati, oltre che dai commenti fatti a lezione dal docente, dalla lettura di testi che affrontano in modo piu' approfondito e piu' rigoroso gli argomenti qui trattati. Tali testi sono indicati dal docente argomento per argomento durante il corso (o a ricevimento se necessario). Si raccomanda di segnalare errori ed imprecisioni presenti negli appunti al fine di eliminarne la maggiore quantita' nelle prossime revisioni.

## 2 Introduzione

In molte applicazioni puo' essere importante approssimare una funzione $f$, eventualmente nota solo in alcuni punti, per scopi diversi che vanno dall'approssimazione dell'integrale, all'approssimazione derivate, alla modellizzazione geometrica. Per costruire l'approssimazione $\tilde{f}$ di una funzione $f$, essenzialmente si deve:

1. stabilire la classe $\mathbb{F}$ di funzioni in cui si vuole costruire l'approssimante;
2. stabilire un criterio che ci permetta di individuare una specifica funzione approssimante $\tilde{f} \in \mathbb{F}$.

Generalmente la classe $\mathbb{F}$ e' uno spazio funzionale lineare generato data da una base opportuna. Esempi di possibili $\mathbb{F}$ sono di seguito elencati.

- $\mathbb{P}_n$ spazio dei polinomi di grado $n$.

$$\mathbb{P}_n = \left\{ \sum_{i=0}^{n} a_i x^i, a_i \in \mathbb{R} \right\}$$

  Questo spazio di funzioni e' vantaggioso per il fatto che i polinomi sono funzioni facilmente derivabili, integrabili, e sono di classe $C^\infty$. Tuttavia

tale classe di funzioni non permette di approssimare bene funzioni che presentano singolarita' (ad esempio delle cuspidi) o di approssimare delle funzioni periodiche.

- $\mathbb{T}_n$ classe dei polinomi trigonometrici.

$$\mathbb{T}_n(x) = \left\{ \sum_{k=0}^{n} a_k \cos(kx) + b_k \sin(kx), \quad a_k, b_k \in \mathbb{R} \right\}$$

  Questa classe e' adatta a rappresentare funzioni periodiche $C^\infty$.

- $\mathbb{R}_n$ spazio delle funzioni razionali

$$\mathbb{R}_{n,m} = \left\{ \frac{P_n(x)}{P_m(x)}, P_i \in \mathbb{P}_i, i = m, n \right\}$$

  Adatto a rappresentare funzioni che presentano singolarita'.

- $\mathbb{E}_n$ spazio delle funzioni esponenziali

$$\mathbb{E}_n = \left\{ \sum_{k=0}^{n} a_k e^{b_k x}, a_k, b_k \in \mathbb{R} \right\}$$

  Adatto per rappresentare fenomeni fisici, biologici.

- $S_{n+1}$ spazio delle funzioni spline

$$S_{n+1}\left([a,b], \Delta\right)$$

  polinomi di grado $n$ in ogni sottointervallo di $\Delta$ con continuita'globale $n-1$, dove $\Delta$ e' una partizione dell'intervallo $[a,b]$. Tale spazio di funzioni e' adatto per rappresentare profili di oggetti e molti altri tipi di funzioni.

Dopo aver elencato alcune possibili classi di funzioni, vediamo di seguito alcuni dei criteri che possono essere utilizzati per scegliere una specifica funzione approssimante $\tilde{f}$ nello spazio $\mathbb{F}$.

1a. **Interpolazione.** Interpolazione di Lagrange, abbiamo $(x_i, f_i)$ per $i = 0, \ldots, n$ e vogliamo che

$$\tilde{f}(x_i) = f_i, \qquad i = 0, \ldots, n.$$

1b. **Interpolazione osculatoria.** Abbiamo a disposizione $(x_i, f(x_i))$ per $i = 0, \ldots, n$ e le derivate $j$-esime, $(x_i, f^{(j)}(x_i))$ in qualche punto per $i \in \{0, \ldots, n\}$. A titolo di esempio possiamo pensare considerare il polinomio di Taylor centrato in un particolare punto $x_0$:

$$P^T_{n,x_0}(x) = \sum_{i=0}^{n} \frac{f^{(i)}(x_0)(x - x_0)^i}{i!}.$$

E' noto che $f(x) = P^T_{n,x_0}(x) + R^n(x_0)$ con $\lim_{n\to\infty} R_{n+1}(x) = 0$ in un opportuno intorno $I_{x_0}$ di $x_0$.

2. **Miglior approssimazione.** Si cerca di minimizzare l'errore "complessivo" fra la funzione $f$ e la sua approssimante $\tilde{f}$. Per esempio di minimizzare con la tecnica dei minimi quadrati, la quantita'

$$||F - \tilde{F}||_2\,,$$

dove $F$ e $\tilde{F}$ sono vettori costruiti tabulando $f$ e $\tilde{f}$, rispettivamente.

3. **Quasi interpolazione.** Andiamo a costruire $\tilde{f}$ come

$$\sum_{j=0}^{n} \lambda_j(f)\phi_j(x), \qquad \lambda_j(f) \in \mathbb{R}, \quad j = 0, \cdots, n$$

dove $\phi_j$ sono opportune funzioni in $\mathbb{F}$ con supporto compatto e forma a campana e $\lambda_j(f)$ sono coefficienti dipendenti da $f$. Questo tipo di approssimazione "segue" l'andamento dei dati senza necessariamente interpolarli.

# 3 Interpolazione di Lagrange

Assegnati i punti $(x_i, f_i)$ per $i = 0, \ldots, n$ (ricordiamo che $x_i,\ i = 0, \ldots, n$ sono detti *nodi fondamentali dell'interpolazione* o piu' semplicemente *nodi* e che non devono essere necessariamente ordinati sebbene in queste pagine noi assumeremo che lo siano) costruiamo il polinomio interpolante nella forma di Lagrange

$$L_n(x) = \sum_{j=0}^{n} f_j \ell_j(x)\,, \qquad \ell_j(x) := \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i}.$$

dove $\ell_j$ sono le basi di Lagrange che soddisfano le condizioni di cardinalita'

$$\ell_j(x_k) = \delta_{jk} = \begin{cases} 1 & j = k, \\ 0 & \text{altrimenti.} \end{cases} \tag{1}$$

Definiamo quindi il polinomio monico (cioe' il cui coefficiente del termine di grado massimo e' uguale ad 1) di grado $n + 1$

$$\omega_{n+1}(x) := (x - x_0)(x - x_1)\cdots(x - x_j)(x - x_{j+1})\cdots(x - x_n) \tag{2}$$

e la sua derivata prima

$$\begin{aligned} \omega'_{n+1}(x) &= (x - x_1)\cdots(x - x_n) + (x - x_0)(x - x_2)\cdots(x - x_n) + \cdots \\ &\cdots + (x - x_0)(x - x_1)\cdots(x - x_{n-1}). \end{aligned}$$

Quando calcoliamo $\omega'_{n+1}(x_j)$ otteniamo

$$\omega'_{n+1}(x_j) = (x_j - x_0)\cdots(x_j - x_{j-1})(x_j - x_{j+1})\cdots(x_j - x_n).$$

Questo ci permette di riscrivere la base $j$-esima di Lagrange come

$$\ell_j(x) = \frac{\omega_{n+1}(x)}{(x - x_j)\omega'_{n+1}(x_j)}$$

e di riscrivere il polinomio interpolante di Lagrange come

$$L_n(x) = \sum_{j=0}^{n} f_j \frac{\omega_{n+1}(x)}{(x - x_j)\omega'_{n+1}(x_j)}.$$

Per le basi di Lagrange vale il seguente risultato.

**Proposizione 1.** *Le basi di Lagrange formano una partizione dell' unita' e cioe' la loro somma e' identicamente uguale ad uno nell' intervallo dei nodi:*

$$\sum_{j=0}^{n} \ell_j(x) = 1, \quad x \in [x_0, x_n].$$

*Dimostrazione.* Per i nodi $x_0, \cdots, x_n$, costruiamo il polinomio interpolante la funzione costante uguale ad 1 fissando $f_j = 1$ per ogni $j = 0, \ldots, n$. Per l ' unicita' del polinomio interpolante, l' unico polinomio di grado al piu' $n$ che interpola i dati e' il polinomio costante stesso che puo' artificiosamente essere espresso dalla somma della basi di Lagrange per 1, cioe'

$$\sum_{j=0}^{n} 1 \cdot \ell_j(x) = \sum_{j=0}^{n} \ell_j(x) = 1, \quad x \in [x_0, x_n].$$

□

## 3.1 Stima dell'errore di interpolazione

Ci proponiamo adesso di studiare l' errore nelle formule di interpolazione. Cominciamo definendo $E_n(x) := f(x) - L_n(x)$ , l'errore commesso dal polinomio interpolante nei confronti della funzione $f$ che vogliamo approssimare. Risulta

**Teorema 1.** *Sia $f \in C^{n+1}[a = x_0, b = x_n]$ ed $L_n(x)$ il polinomio interpolante i dati $(x_i, f(x_i))$, $i = 0, \cdots, n$. Allora esiste $\xi \in [a, b]$, $\xi(x, \{x_i\}_{i=0}^n)$, tale che, per $\omega_{n+1}$ definito in (2), risulta*

$$E_n(x) = \frac{\omega_{n+1}(x) f^{(n+1)}(\xi)}{(n+1)!}; \tag{3}$$

*Dimostrazione.* Ovviamente

$$E_n(x_i) = 0, \qquad i = 0, \ldots, n,$$

e pertanto

$$E_n(x) = (x - x_0) \cdots (x - x_n) R_n(x) = \omega_{n+1}(x) R_n(x)$$

dove $R_n(x)$ e' una funzione incognita. Una sua espressione ci permetterebbe di conoscere l'errore di interpolazione in un qualunque punto fissato $x$ nell' intervallo dei nodi. Per ottenere un' espressione di $R_n(x)$ utilizziamo la seguente tecnica. Sia $x$ un punto specifico in $[x_0, x_n]$ e consideriamo la funzione ausiliaria

$$G(t) = f(t) - L_n(t) - \omega_{n+1}(t) R_{n+1}(x)$$

che ha le seguenti proprita' :

- $G(x_i) = 0$ per $i = 0, \ldots, n$ (per l'interpolazione)
- $G(x) = 0$ per costruzione.

Questo implica che la $G(t)$ si annulla negli $n + 2$ punti $x_0, \ldots, x_n, x$. Assumendo che $f \in C^{n+1}[a, b]$, allora per il teorema di Rolle possiamo concludere che esistono almeno $n + 1$ punti in cui si annulla la derivata prima; nuovamente, per il teorema di Rolle, esistono $n$ punti in cui si annulla la derivata seconda. Iterando il ragionamento $n + 1$ volte otteniamo l'esistenza di un punto $\xi(x, \{x_i\}_{i=0}^n)$ (che dipende da $x$ e dai nodi $x_0, \cdots, x_n$) in cui si annulla la derivata $(n + 1)$-esima di $G$:

$$G^{(n+1)}(\xi) = 0.$$

D' altra parte derivando $G$ otteniamo

$$G^{(n+1)}(t) = f^{(n+1)}(t) - L_n^{(n+1)}(t) - \omega_{n+1}^{(n+1)}(t) R(x).$$

Nella precedente uguaglianza, possiamo notare che

- essendo $L_n$ un polinomio di grado $n$, ha derivata $(n+1)$-esima e' nulla;
- essendo $\omega_{n+1}$ un polinomio monico di grado $n + 1$ risulta $\omega_{n+1}^{(n+1)}(t) = (n + 1)!$

e quindi possiamo riscrivere

$$0 = G^{(n+1)}(\xi) = f^{(n+1)}(\xi) - (n + 1)! R_{n+1}(x)$$

che implica

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!}.$$

In conclusione abbiamo dimostrato che

$$E_n(x) = \frac{\omega_{n+1}(x) f^{(n+1)}(\xi)}{(n+1)!}; \tag{4}$$

con $\xi \in [a = x_0, b = x_n]$, di cui peró non conosciamo l' esatto valore. □

E' importante ricordare che $\xi$ dipende da $x$ oltre che dai nodi $x_0, \cdots, x_n$. Dalla precedente espressione dell' errore possiamo concludere che $E_n(x)$, dipende da:

- la distribuzione dei nodi
- il comportamento/caratteristiche di $f$ e delle sue derivate
- il grado $n$ del polinomio interpolante

Notiamo che al crescere di $n$, $1/(n+1)! \to 0$ , ma, a causa del comportamento della derivata di $f$ nel punto $\xi(x, \{x_i\}_{i=0}^n)$ non abbiamo garanzia che anche l' errore tenda a zero quando $n$ tende all'infinito. Esistono infatti classi di funzione per cui al crescere di $n$, il polinomio interpolante non converge alla funzione. Per specifiche classi di funzioni e/o scelte dei nodi la convergenza risultata tuttavia garantita come vedremo in seguito.

Un utile e semplice risultato di convergenza del polinomio interpolante alla funzione per $n \to \infty$ e' il seguente la cui dimostrazione e' ovvia.

**Teorema 2.** *Sia $f \in C^{n+1}[a, b]$. Se $\|f^{(n+1)}\|_\infty \leq M_{n+1}$ $\forall n$ e se*

$$\lim_{n\to\infty} \frac{|b-a|^{n+1} M_{n+1}}{(n+1)!} = 0$$

*allora*

$$\lim_{n\to\infty} \|L_n - f\|_\infty = \lim_{n\to\infty} \sup_{x\in[a,b]} |L_n(x) - f(x)| = 0.$$

### 3.1.1 Studio del condizionamento del problema dell' interpolazione

Analizziamo adesso il condizionamento del problema dell' interpolazione polinomiale rispetto a errori nelle misure $f_0, f_1, \cdots, f_n$. Siano $\widetilde{f}_i = f_i(1 + \epsilon_i)$ per $i = 0, 1, \cdots, n$ i valori perturbati di $f_i$, $i = 0, 1, \cdots, n$ e sia

$$\epsilon = \max_{i=0,\cdots,n} |\epsilon_i|.$$

Vogliamo studiare la ripercussione di tali errori sui valori che il polinomio interpolante assume. A tale scopo definiamo la *funzione di Lebesgue* e la *costante di Lebesgue* rispettivamente come

$$\lambda_n(x) = \sum_{i=0}^{n} |\ell_i(x)|, \qquad \Lambda_n = \max_{x\in[a,b]} \lambda_n(x), \quad x \in [x_0, x_n].$$

Tale funzione e' in $[a,b]$ sempre maggiore o uguale ad 1. Infatti, per qualunque scelta dei nodi risulta,

$$\lambda_n(x) = \sum_{i=0}^{n} |\ell_i(x)| \geq \sum_{i=0}^{n} \ell_i(x) = 1 \quad \rightarrow \quad \Lambda_n \geq 1.$$

Consideriamo quindi il polinomio interpolante i dati affetti da errore

$$\widetilde{L_n}(x) = \sum_{i=0}^{n} \widetilde{f}_i \ell_i(x)$$

e l'errore assoluto $|\tilde{L}_n(x) - L_n(x)|$ che risulta

$$|\widetilde{L_n}(x) - L_n(x)| \leq \sum_{i=0}^{n} |\widetilde{f}_i - f_i||\ell_i(x)| \leq \epsilon \max_{i=0,\ldots,n} |f_i| \, \lambda_n(x) \leq \epsilon \max_{i=0,\ldots,n} |f_i| \, \Lambda_n \, .$$

Inoltre, poiche' $f_j = L_n(x_j),\ j = 0, \cdots, n$ per ogni $j$, si ha

$$max_{0\leq j\leq n} |f_j| = max_{0\leq j\leq n} |L_n(x_j)| \leq \max_{x\in[a,b]} |L_n(x)|$$

da cui consegue

$$\left|\widetilde{L}_n(x) - L_n(x)\right| \leq \epsilon \, max_{x\in[a,b]} |L_n(x)| \, \Lambda_n$$

e quindi la stima relativa

$$\frac{\max_{x\in[a,b]} \left|\widetilde{L}_n(x) - L_n(x)\right|}{\max_{x\in[a,b]} |L_n(x)|} \leq \epsilon \, \Lambda_n.$$

Si nota che:

- se $\Lambda_n$ e' un numero piccolo, a piccole perturbazioni dei dati corrispondono piccole perturbazioni dei risultati;
- se $\Lambda_n$ e' grande, a piccole perturbazioni dei dati possono corrispondere grandi perturbazioni dei risultati.

In conclusione, $\Lambda_n$ gioca il ruolo del numero di condizionamento per il problema dell' interpolazione.

## 3.2 Scelta dei nodi

In questo paragrafo studiamo come scegliere opportunamente i nodi fondamentali dell' interpolazione in modo da ridurre l' errore espresso con la formula (3), e quando possibile la costante di Lebesgue $\Lambda_n$. In particolare, siamo interessati a capire se esiste una scelta ottimale.

### 3.2.1 Caso nodi uniformi

La scelta piu' semplice ma anche la meno vantaggiosa (come vedremo) e'quella dei nodi uniformi. In questo caso i nodi vengono scelti equidistanti tra di loro: a partire dal primo nodo $x_0$ si ha :

$$x_i = x_0 + ih \quad i = 0, 1, ..., n, \quad h = \frac{x_n - x_0}{n}$$

dove $h$ e' la distanza uniforme tra i nodi. Con il cambiamento di variabile :

$$x = x_0 + th \quad t \in [0, n],$$

possiamo riscrivere la funzione $\omega_{n+1}$ in funzione di t :

$$\omega_{n+1}(t) = (x_0 + th - x_0)(x_0 + th - x_0 - h)...(x_0 + th - x_0 - nh) =$$
$$= (th)(t-1)h \cdots (t-n)h =$$
$$= h^{n+1}t(t-1)(t-2)\cdots(t-n)$$
$$= h^{n+1}\prod_{i=0}^{n}(t-i).$$

Dalla precendente uguaglianza si evince che la funzione $\omega_{n+1}$ ha le seguenti proprieta' :

- ha come zeri $0, 1, \cdots, n$: che risulta ovvia;
- $\frac{n}{2}$ e' il punto di simmetria essendo $|\omega_{n+1}(\frac{n}{2} + t)| = |\omega_{n+1}(\frac{n}{2} - t)|$: Infatti

  $$|\omega_{n+1}(\frac{n}{2} + t)| = |h^{n+1}\prod_{i=0}^{n}(\frac{n}{2} + t - i)| = |h^{n+1}\prod_{i=0}^{n}(i - t - \frac{n}{2})|$$

  da cui sommando e sottraendo $\frac{n}{2}$, e ponendo $j = n - i$ si ottiene

  $$= |h^{n+1}\prod_{i=0}^{n}(i - n + \frac{n}{2} - t)| = |h^{n+1}\prod_{j=0}^{n}(\frac{n}{2} - t - j)| =$$
  $$= |\omega_{n+1}(\frac{n}{2} - t)|;$$

- per $t$ non intero, $t \leq \frac{n}{2}$, risulta $|\omega_{n+1}(t-1)| > |\omega_{n+1}(t)|$: Infatti

  $$|\omega_{n+1}(t-1)| = |\prod_{i=0}^{n}(t-1-i)| = |\prod_{j=1}^{n+1}(t-j)| = |\omega_{n+1}(t)| \cdot |\frac{t-n-1}{t}|$$

  ed essendo $t \leq \frac{n}{2}$ risulta $|t - n - 1| = |n + 1 - t| > \frac{n}{2}$, e quindi

  $$|\frac{t-n-1}{t}| > 1;$$

  da cui la tesi.

- a causa della simmetria e della proprieta' precedente, si ha che per $t$ non intero, $t \geq \frac{n}{2}$, risulta $|\omega_{n+1}(t)| < |\omega_{n+1}(t+1)|$.
- i massimi relativi di $|\omega_{n+1}(t)|$ in ogni intervallo $(i, i+1)$, crescono quando ci si allontana dal centro dell'intervallo $[0, n]$ verso gli estremi: deriva dalle due proprieta' precedenti;
- e' massima agli estremi: infatti

$$|\omega_{n+1}(t-1)| < |\omega_{n+1}(t)| < |\omega_{n+1}(t+1)|$$

- al crescere di $n$ il massimo di $|\omega_{n+1}(t)|$ in $[0, n]$, che viene assunto nel primo e nell'ultimo sottointervallo, cresce rapidamente con $n$: Infatti

$$\max_{t\in[0,n]}|\omega_{n+1}(t)| = \max_{t\in[0,1]}|\omega_{n+1}(t)| \geq |\omega_{n+1}(\frac{1}{2})| = |\prod_{j=0}^{n}(\frac{1}{2}-j)|$$

$$= \tfrac{1}{2^{n+1}}[1\cdot 3\cdot 5\cdots(2n-1)] = \tfrac{(2n-1)!}{2^{n+1}[2\cdot 4\cdots(2n-2)]}$$

$$= \frac{(2n-1)!}{2^{2n}(n-1)!}.$$

Per quanto riguarda la costante di Lebesgue $\Lambda_n = \max_{x\in[a,b]} \lambda_n(x)$, e quindi degli errori di propagazione, si puó dimostrare che nel caso di nodi uniformi si ha

$$\Lambda_n \simeq \frac{2^{n+1}}{en\log n} \quad \text{ed in particolare} \quad \Lambda_n \geq e^{\frac{n}{2}} \quad \text{per } n \text{ grande.}$$

La formula precedente mette in luce che, nel caso dei nodi uniformi all' aumentare di $n$ la costante di Lebesgue tende ad aumentare piuttosto che a diminuire mentre, al contrario, nel caso di altre scelte dei nodi l'andamento di $\Lambda_n$ risulta migliore.

La scelta dei nodi uniformi fa si che agli estremi dell' intervallo $[a, b]$ il comportamento dell' errore sia anomalo e che l'aumento del numero dei nodi non garantisca la convergenza del polinomio interpolante alla funzione. Una possibile soluzione al problema consiste nel prendere piu' nodi nella zona regolare, e meno in quelle irregolari. In ogni caso i nodi uniformi non rappresentano una buona scelta, anche in relazione al malcondizionamento del problema (vedi paragrafo precedente)

### 3.2.2 Zeri di Chebyshev

I nodi di Chebyshev sono gli zeri degli omonimi polinomi. L' idea alla base della loro definizione e' che, dato l' andamento di $\omega_{n+1}$, agli estremi dell'

intervallo occorrono piu' nodi in modo da fornire piu' informazioni proprio dove l' errore e' massimo. Il meccanismo adottato per costruirli e' la distribuzione uniforme di punti lungo la circonferenza unitaria e la loro proiezione sull' asse delle ascisse.

Proseguiamo con la definizione formale. Dato l' intervallo $[-1,1]$ gli $n+1$ nodi (o zeri) di Chebyshev sono definiti dalla formula

$$x_i = \cos(\frac{(2i+1)}{(n+1)}\frac{\pi}{2}) \quad i = 0, ..., n. \tag{5}$$

Per capire il significato del loro nome (*zeri di Chebyshev*) preso $\theta \in (0, 2\pi)$ e definito $x = \cos\theta \in [-1,1]$ (ovviamente $\theta = \arccos x$), si definisce

$$\{T_n(x) = \cos n(\arccos x)\}_{n\geq 0}$$

la successione di polinomi di Chebyshev, i cui zeri sono proprio i nodi in (5). Infatti gli zeri di $T_{n+1}$ sono gli zeri del coseno ($\cos(n+1)\theta$ ) e pertanto del tipo

$$(n+1)\theta_i = \frac{\pi}{2} + 2i\frac{\pi}{2} \quad i = 0, ..., n,$$

da cui si evince

$$\theta_i = \frac{(2i+1)}{(n+1)}\frac{\pi}{2}, \quad i = 0, ..., n.$$

Quindi, poiche' $x = \cos\theta$, gli zeri di Chebyshev sono

$$x_i = \cos(\theta_i) = \cos\frac{(2i+1)}{(n+1)}\frac{\pi}{2}, \quad i = 0, ..., n,$$

che si addensano all'origine, diversamente dai nodi uniformi. E' importante osservare che il primo e l' ultimo nodo di Chebyshev non coincidono necessariamente gli estremi dell' intervallo $[-1,1]$ cosa che in effetti dipende dal valore di $n$. É facile verificare che

$$T_0(x) = \cos(0) = 1, \quad T_1(x) = \cos(\arccos x) = x,$$

e che

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

e cioe' che i polinomi di Chebyshev possono essere definiti in modo ricorsivo mediante una relazione a tre termini tipica dei polinomi ortogonali (quali in effetti sono).
Scegliendo gli zeri di Chebyshev come nodi dell' interpolazione, si ha che anche la costante di Lebesgue risulta molto piu' piccola. Infatti si ha

$$\Lambda_n \simeq \frac{2}{\pi}\log n.$$

Inoltre vale il seguente risultato teorico di convergenza del polinomio interpolante alla funzione:

**Teorema 3.** *Sia $f$ una funzione Lipschitz-continua in $[a,b]$. Sia $L_n$ il polinomio interpolante costruito sui nodi di Chebyshev, allora risulta*

$$\lim_{n\to\infty} \sup_{x\in[a,b]} |f(x) - L_n(x)| = 0.$$

Sempre grazie ai nodi di Chebyshev, si riesce a rendere minimo il massimo valore assunto dal polinomio monico $\omega_{n+1}$ e quindi a minimizzare l' errore di interpolazione. Vale infatti il seguente risultato (per la cui dimostrazione si rimanda alla letteratura).

**Teorema 4.** *Sia $\tilde{T}_{n+1} = \prod_{i=0}^{n}(x - x_i)$, il polinomio monico costruito con i nodi di Chebyshev $x_i$, quadi $= 0 \cdots n$ definiti in* (5)*. Allora*

$$\max_{x\in[-1,1]} |\widetilde{T}_{n+1}(x)| \leq \max_{x\in[-1,1]} |\widetilde{P}_{n+1}(x)|$$

*dove $\tilde{P}_{n+1}(x)$ e' un qualunque altro polinomio monico di grado $n+1$.*

*Dimostrazione.* Il polinomio $\tilde{T}_{n+1}$ si ottiene dalla seguente espressione:

$$\tilde{T}_{n+1}(x) = \frac{1}{2^n} T_{n+1}(x), \quad \forall x \in [-1,1]. \tag{6}$$

Infatti, riprendendo la formula di ricorsione dei polinomi di Chebyshev

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

si dimostra facilmente per induzione che

$$T_{n+1}(x) = 2^n x^{n+1} + \text{termini di grado inferiore}$$

e quindi

$$\tilde{T}_{n+1}(x) = x^{n+1} + \text{termini di grado inferiore} \tag{7}$$

Da (6) e (7) si conclude facilmente che $\tilde{T}_{n+1}$ e' un polinomio monico di grado $n+1$ con gli stessi zeri di $T_{n+1}(x)$ in $[-1,1]$, ovvero:

$$\tilde{T}_{n+1}(x) = \frac{1}{2^n} T_{n+1}(x) = \prod_{i=0}^{n}(x - x_i), \quad x_i = \cos\theta_i,\ i = 0,\ldots,n$$

Utilizzando la definizione di $T_{n+1}(x)$

$$T_{n+1}(x) = \cos((n+1)\arccos x),$$

ricaviamo che $|T_{n+1}(x)| \leq 1$, e quindi che

$$|\tilde{T}_{n+1}(x)| \leq \frac{1}{2^n}.$$

I punti di massimo e di minimo di $\tilde{T}_{n+1}(x)$ sono gli stessi di $T_{n+1}(x)$ in $[-1,1]$, ovvero gli $n+2$ punti $x'_i, i = 0,\ldots,n+1$, tali che:

$$(n+1)\arccos x = i\pi \;\Leftrightarrow\; x'_i = \cos(\frac{i\pi}{n+1}), \quad i = 0,\ldots,n+1. \tag{8}$$

Ne segue che $|\tilde{T}_{n+1}(x)|$ assume $n+2$ volte in $[-1,1]$ il suo valore massimo $1/2^n$. Supponiamo ora per assurdo che esista un polinomio monico $p(x)$ di grado $n+1$ tale che:

$$\max_{x\in[-1,1]} |p(x)| < \max_{x\in[-1,1]} |\tilde{T}_{n+1}(x)| = \frac{1}{2^n} \tag{9}$$

Il polinomio $Q(x) := \tilde{T}_{n+1}(x) - p(x)$ risulterebbe di grado $\leq n$ in quanto differenza di due polinomi monici di grado $n+1$. Esso assumerebbe valore positivo nei punti in cui $\tilde{T}_{n+1}$ ha massimo e valore negativo in cui $\tilde{T}_{n+1}$ ha minimo, ovvero assumerebbe nei punti $x'_i$ definiti in (8) il segno di $\tilde{T}_{n+1}$. Pertanto $Q(x)$ cambierebbe di segno $(n+2)$ volte e si dovrebbe dunque annullare in almeno $n+1$ punti, ma essendo un polinomio di grado $n$ deve per forza risultare $Q(x) \equiv 0$. Dunque, $\tilde{T}_{n+1}(x) \equiv p(x)$ e si ha:

$$\frac{1}{2^n} = \max_{x\in[-1,1]} |\tilde{T}_{n+1}(x)| = \max_{x\in[-1,1]} |p(x)| < \frac{1}{2^n}$$

che e' un assurdo. Cio' conclude la dimostrazione. □

La funzione $E_n(x)$, che esprime l'errore nell' interpolazione della funzione $f$ con $n+1$ nodi, mostra che l' errore puó essere visto come formato da due componenti : la prima, $R(x)$, dipendente dalla natura della $f(x)$, e la seconda, $\omega_{n+1}(x)$, strettamente collegata con la distribuzione degli $n$ nodi. Il Teorema 3 ci garantisce che, se i nodi sono zeri di Chebyshev, la quantita'

$$\max_{x\in[-1,1]} |\omega_{n+1}(x)|$$

e' la minima possibile, e cioe' la migliore. Il risultato si estende al caso di intervalli generici $[a,b]$ applicando la trasformazione lineare:

$$y_i = \frac{a+b}{2} + \frac{(b-a)}{2} x_i$$

dove gli $x_i$ sono gli zeri di Chebyshev di $T_{n+1}(x) \quad \in [-1,1]$. La scelta dei nuovi nodi $y_i$ permette di minimizzare $max|\omega_{n+1}(x)|$ su $[a,b]$ ottenendo:

$$\max_{x\in[a,b]} |\omega_{n+1}(x)| \leq 2\Big[\frac{b-a}{4}\Big]^{n+1}. \tag{10}$$

In conclusione l' utilizzo dei nodi di Chebyshev permette di avere un controllo maggiore sul condizionamento del problema, di minimizzare il massimo

valore di $\omega_{n+1}$; in piú, di garantire la convergenza del polinomio interpolante alla funzione (qualora questa sia Lipschitz continua). La (10) evidenzia pero' il fatto che se l'intervallo $[a, b]$ non e' sufficientemente piccolo, il contributo dell'errore dovuto ai punti fondamentali puo' diventare rilevante. In questo caso puo' essere utile adottare soluzioni diverse, come la suddivisioni dell'intervallo $[a, b]$ in sottointervalli e la conseguente costruzione di una approssimante polinomiale a tratti, avente in $[a, b]$ opportune caratteristiche di regolaritá (come le Splines studiate oltre).

# 4 Polinomi osculatori

In questo paragrafo ci proponiamo di studiare altri tipi di polinomi interpolanti quali i polinomi osculatori. In generale, assegnata una funzione derivabile $f$ (o alcuni sui valori e valori delle sue derivate) ed i nodi dell' interpolazione $x_i,\ i = 0, \cdots, n$, un polinomio osculatore soddisfa le seguenti condizioni:

$$P^{(k)}(x_i) = f^{(k)}(x_i), \quad k \in I, \quad i = 0, \cdots, n,$$

dove $I$ e' un sottoinsieme di $\{0, ..., l\}$ con $l \in \mathbb{N}$. É evidente che per $l = 0$, si ha l' interpolazione dei soli valori di $f$ detta di *Lagrange*, mentre per $l = 1$ si ha l' interpolazione di valori e derivate detta di *Hermite*.

## 4.1 Interpolazione di Hermite

A partire da un insieme di dati del tipo $(x_j, f_j, f_j^{'}),\ j = 0, \ldots, n$, costruiamo un polinomio di grado $2n + 1$, $P_{2n+1}^H$, imponendo le $2(n + 1)$ condizioni di interpolazione

$$P_{2n+1}^H(x_j) = f_j, \quad (P_{2n+1}^H)^{'}(x_j) = f_j^{'}, \quad j = 0, \cdots, n.$$

Tale polinomio e' detto polinomio di *Hermite*. E' facile verificare che il polinomio di Hermite puó essere rappresentato nella seguente forma

$$P_{2n+1}^H(x) = \sum_{j=0}^{n} f_j u_j(x) + \sum_{j=0}^{n} f_j^{'} v_j(x),$$

dove le funzioni $u_j$ e $v_j$ (dette basi di Hermite), costruite come combinazione delle basi di Lagrange, hanno la seguente forma analitica

$$u_j(x) = \left(1 - 2\ell_j^{'}(x_j)(x - x_j)\right) \ell_j^2(x), \quad j = 0, \cdots, n$$

$$v_j(x) = (x - x_j)\ell_j^2(x), \quad j = 0, \cdots, n$$

e sono polinomi di grado $2n + 1$ che verificano le seguenti condizioni :

- $u_j(x_k) = \delta_{kj}\ , \quad u_j^{'}(x_k) = 0$ per $k = 0, \ldots, n$ e $j = 0, \cdots, n$;

- $v_j(x_k) = 0\ , \quad v_j^{'}(x) = \delta_{kj}$ per $k = 0, \ldots, n$ e $j = 0, \cdots, n$.

Per quanto riguarda l' errore, vale il seguente risultato.

**Teorema 5.** *Sia* $f \in C^{2n+2}[a,b]$, $\exists \xi(x, \{x_i\}_{i=0}^n) \in [a,b]$ *tale che :*

$$E_n^H(x) := f(x) - P_{2n+1}^H(x) = \frac{\omega_{n+1}^2(x) f^{2n+2}(\xi(x))}{(2n+2)!}.$$

*Dimostrazione.* La dimostrazione utilizza la stessa idea del caso di Lagrange e si basa sulla osservazione preliminare che, per le condizioni di interpolazione all' Hermite, l' errore assume la forma

$$E_n^H(x) := f(x) - P_{2n+1}^H(x) = \omega_{n+1}^2(x) R^H(x)$$

essendo zero, insieme alla sua derivata, nei nodi $x_i,\ i = 0, \ldots, n$. Si procede quindi considerando la funzione ausiliaria

$$G(t) = f(t) - P_{2n+1}^H(t) - \omega_{n+1}^2(t) R^H(x),$$

e la sua derivata.

$$G^{'}(t) = f^{'}(t) - (P_{2n+1}^H)^{'}(t) - 2\omega_{n+1}(t)\omega_{n+1}^{'}(t) R^H(x).$$

La funzione $G(t)$ si annulla negli $n+2$ punti $x, x_i,\ i = 0, \cdots, n$, e pertanto la sua derivata, $G'(t)$, si annulla in $2n+2$ punti dati da:

- $n+1$ punti $z_0, ..., z_n$ dentro agli intervalli, per il Teorema di Rolle applicato a $G$
- $n+1$ punti $x_0, ..., x_n$ distinti dai precedenti, per le condizione di interpolazione sulla derivata prima.

E quindi, come nel caso della formula di Lagrange ragionando su $G^{'}(t)$ (piuttosto che su $G(t)$), si conclude che la sua derivata di grado $2n+1$ si annullera' in un solo punto $\xi(x, \{x_i\}_{i=0}^n)$. Pertanto

$$G^{2n+2}(\xi) = 0 = f^{2n+2}(\xi(x, \{x_i\}_{i=0}^n)) - (2n+2)! R^H(x).$$

Da cio' segue la seguente espressione dell' errore

$$E_{2n+1}^H(x) = \frac{\omega_{n+1}^2(x) f^{2n+2}(\xi(x, \{x_i\}_{i=0}^n)}{(2n+2)!}.$$

□

## 4.2 Polinomi di Bernstein e Teorema di Weierstrass

Lo studio dell' errore nei polinomi di interpolazione di Hermite mette in evidenza il fatto che anche l' interpolazione polinomiale osculatoria soffre delle stesse problematiche di quella di Lagrange. In alcuni casi, pertanto, per approssimare bene una funzione utlizzando lo spazio dei polinomi, puo' essere conveniente abbandonare l' interpolazione optando per altre forme di "approssimazione" seppur di natura polinomiale. Questa idea e' confermata dal seguente importante risultato che afferma che ogni funzione continua puo'essere approssimata bene quanto si vuole da un polinomio (di cui tuttavia non si sa ne' il grado ne' se e' di tipo interpolatorio.) In altre parole, il risultato afferma che l'insieme dei polinomi e' denso in $C[a,b]$.

**Teorema 6** (Weierstrass)**.** *Sia $f \in C[a,b]$, $\forall \epsilon > 0$, $\exists \overline{n}$ tale che $\forall n > \overline{n}$*

$$\sup_{x\in[a,b]} |f(x) - P_n(x)| < \epsilon,$$

*o, equivalentemente, per $f \in C[a,b]$, esiste una successione di polinomi $\{P_n\}_{n\geq 0}$ tale che*

$$\lim_{n\to\infty} \|f - P_n\|_\infty = 0.$$

Con lo scopo di dimostrare il precedente Teorema, il matematico russo Sergei Natanovich Bernstein propose la costruzione di una successione di polinomi approssimante una assegnata funzione continua $f \in C[a,b]$. Tale costruzione utilizza una base diversa da quella canonica dello spazio dei polinomi. Per il momento, assuminamo $[a,b] = [0,1]$ e consideriamo $x_i = \frac{i}{n}$, con $i = 0, ..., n$, punti equidistanti in $[0,1]$. In polinomio di Berstein approssimante la funzione $f$ e' definito dalla formula:

$$P_{B_n}(x) = \sum_{i=0}^{n} f(x_i) B_i^n(x)$$

dove

$$B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i}, \quad x \in [0,1], \quad i = 0, \ldots, , n$$

sono i cosi' dette polinomi di Bernstein, e cioe' $n+1$ polinomi di grado $n$ soddisfacenti le seguente proprieta' :

$$B_i^n(x) \geq 0, \quad x \in [0,1], \quad \sum_{i=0}^{n} B_i^n(x) = 1, \ x \in [0,1].$$

La dimostrazione delle precedenti proprieta' e' alquanto banale se si considera che $x, (1-x)$ sono entrambi positivi per $x \in [0,1]$ e che

$$\sum_{i=0}^{n} B_i^n(x) = \sum_{i=0}^{n} \binom{n}{i} x^i (1-x)^{n-i} = (x + (1-x))^n = 1.$$

E' altrettanto semplice verificare direttamente che

$$P_{B_n}(0) = f(0) \quad P_{B_n}(1) = f(1),$$

e quindi che il polinomio approssimante $P_{B_n}$ e' di fatto interpolante la $f$ nel primo e nell' ultimo punto dell' intervallo (e quindi in 0 ed 1). Le basi di Bernstein ed il relativo polinomio interpolante sono facilmente definibili su $[a, b]$ utilizzando la trasformazione lineare $\mathcal{L} : [a, b] \longrightarrow [0, 1]$

$$\mathcal{L}(x) = \frac{a - x}{a - b}, \quad x \in [a, b],$$

ottenendo l' espressione del polinomio in $[a, b]$ data da

$$B_i^n(x) = \binom{n}{i} \frac{(a - x)^i (x - b)^{n-i}}{(a - b)^n}, \quad x \in [a, b] \quad i = 0, \ldots, n.$$

E' possibile utilizzare i polinomi di Bernstein per ottenere la dimostrazione costruttiva del Teorema di Weirstrass. Infatti si ha:

**Teorema 7.** *Sia $f \in C[a, b]$, $x_0, \ldots, x_n \in [a, b]$ $n + 1$ punti uniformi con $x_0 = a$, $x_n = b$. Risulta,*

$$\lim_{n \to \infty} \sup_{x \in [a,b]} |P_{B_n}(x) - f(x)| = 0.$$

# 5 Splines

Una classe di funzioni (alternativa ai polinomi) molto utilizzata sia per interpolare che approssimare una funzione $f : [a, b] \to \mathbb{R}$ e' la classe delle funzioni *splines* che, essenzialmente, consiste in polinomi a tratti con specifica regolarita'. L' idea e'quella di suddividere l' intervallo di definizione di $f$, quindi $[a, b]$, in sottointervalli e definire in ciascuno di essi un polinomio di grado fissato specificando la regolarita' complessiva che la funzione deve avere in $[a, b]$ (di fatto il problema della regolarita' si pone solo nei nodi data la natura polinomiale dei tratti). Proseguiamo introducendo formalmente la classe delle funzioni splines:

Dato un intervallo $[a, b]$, fissiamo $m$ punti ad esso interni $y_1 < y_2 < ... < y_m$ e poniamo quindi per comodita' $a = y_0$ e $b = y_{m+1}$. Cosí facendo possiamo vedere l'intervallo $[a, b]$ suddiviso in $m+1$ sottointervalli disgiunti $I_0, I_1, ..., I_m$, definiti da

$$I_i = [y_i, y_{i+1}) \quad \text{per } i = 0, ..., m - 1, \quad I_m = [y_m, y_{m+1}]$$

Si definisce *spline di grado $p$ e* nodi $y = (y_0, ..., y_{m+1})$ (attenzione: pur avendo lo stesso nome i nodi di una splines non sono i nodi fondamentali dell'

interpolazione!) una funzione polinomiale a tratti tale che su ogni sottointervallo $I_i$ coincide con un polinomio di grado $p$, e tale per cui i tratti di polinomio si raccordano nei nodi in modo che la funzione sia continua su tutto $[a, b]$ insieme alle sue derivate fino a quella di ordine $p - 1$. In altre parole, indicata con il simbolo $S_{p,y}$, la spline risulta:

$$S_{p,y}(x) \in \mathbb{P}_p \quad \text{per } x \in I_i, \quad i = 0, ..., m \qquad S_{p,y} \in C^{p-1}[a, b],$$

dove $\mathbb{P}_p$ indica l'insieme dei polinomi di grado $p$.

Per esempio, una spline quadratica ($p = 2$) e' costituita da archi di parabola che si raccordano nei nodi interni $y_i$, $i = 1, \cdots, m$ in modo tale che la funzione e la sua derivata prima siano continue (per esempio senza cuspidi!); una spline cubica ($p = 3$), e' data da $m$ tratti polinomiali di terzo grado (cubici) che si raccordano l'uno con l'altro in modo tale da dare una funzione continua su tutto $[a, b]$, insieme alla sua derivata prima e alla sua derivata seconda.

Chiediamoci quanti gradi di liberta' ci sono per individuare univocamente una spline, fissati $p$ ed $m$. Data la definizione di spline, in ognuno dei nodi interni $y_1, ..., y_m$ devono valere le condizione di raccordo per la funzione e le sue derivate

$$\lim_{x \to y_i^-} S_{p,y}^{(j)}(x) = \lim_{x \to y_i^+} S_{p,y}^{(j)}(x), \quad \text{per } j = 0, ..., p - 1,$$

per un totale di $m \times p$ condizioni. D' altra parte in ognuno degli $(m + 1)$ sottointervalli $I_0, I_1, ..., I_m$, la spline e' rappresentata da un polinomio di grado $p$ ed e' quindi definita mediante $(p+1)$ coefficienti. Allora $S_{p,y}$ dipende complessivamente da $(m+1) \times (p+1)$ coefficienti. La differenza tra il numero di coefficienti e il numero di condizioni di raccordo

$$(m + 1) \times (p + 1) - m \times p = m + p + 1,$$

fornisce il numero di gradi di liberta'. In pratica, una volta fissati i nodi ed il grado della splines, per individuare una particolare spline si devono fornire $m + p + 1$ condizioni.

Chiediamoci ora quale puo' essere una base per le funzioni splines. Partiamo con un esempio: supponiamo di avere quattro nodi

$$a = y_0 < y_1 < y_2 < y_3 = b,$$

e costruiamo le splines da sinistra a destra cominciando dall' intervallo $[y_0, y_1)$. In questo intervallo la spline e' rappresentata da un polinomio di grado $p$ che indichiamo con $P_p(x)$; si ha quindi

$$S_{p,y}(x) = P_p^0(x), \quad \text{per } x \in [y_0, y_1).$$

Passando all'intervallo successivo $[y_1, y_2)$, puo' essere conveniente rappresentare la spline in $[y_0, y_2)$ come

$$S_{p,y}(x) = P_p^0(x) + c_1 P_p^1(x),$$

dove $c_1$ e' un opportuno coefficiente e $P_p^1(x)$ e' una funzione che vale 0 per $x \in [y_0, y_1)$ e si identifica con un polinomio di grado $p$ su $[y_1, y_2)$. Per esempio

$$P_p^1(x) = (x - y_1)_+^p = \begin{cases} 0 & x < y_1 \\ (x - y_1)^p & x \geq y_1. \end{cases}$$

Analogamente, considerando l'ultimo sotto-intervallo $[y_2, y_3]$ possiamo pensare che la spline abbia in $[a, b] = [y_0, y_3]$ la forma

$$S_{p,y}(x) = P_p(x) + c_1 P_p^1(x) + c_2 P_p^2(x),$$

dove $c_2$ e' un altro coefficiente e $P_p^2(x)$ vale 0 in $[y_0, y_2)$ e coincide con un polinomio di grado $p$ in $[y_2, y_3]$; di nuovo possiamo considerare

$$P_p^2(x) = (x - y_2)_+^p = \begin{cases} 0 & x < y_2 \\ (x - y_2)^p & x \geq y_2. \end{cases}$$

In questo modo la nostra spline avra' in $[a, b] = [y_0, y_3]$ la generica espressione

$$S_{p,y}(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_p x^p + c_1 (x - y_1)_+^p + c_2 (x - y_2)_+^p,$$

che in effetti dipende da $(p+1)+2$ parametri liberi e che e', di fatto, definita localmente intervallo per intervallo.

Generalizzando al caso di $m$ nodi interni possiamo scrivere una generica splines di grado $p$ e nodi interni $y_1, \cdots, y_m$ come

$$S_{p,y}(x) = \sum_{i=0}^{p} a_i x^i + \sum_{i=1}^{m} c_i (x - y_i)_+^p,$$

dove, per $i = 1, \cdots, m$

$$(x - y_i)_+^p = \begin{cases} 0 & x < y_i \\ (x - y_i)^p & x \geq y_i, \end{cases}$$

prende il nome di *potenza troncata* di grado $p$ relativa al nodo $y_i$.

## 5.1 Potenze troncate e loro proprieta'

La potenza troncata $(x-l)_+^p$ non e' un polinomio, perche' se lo fosse dovrebbe essere derivabile infinite volte (cioe' essere $C^\infty$), ma e' una funzione derivabile con continuita' $p-1$ volte (e'quindi $C^{p-1}$); infatti, per esempio, per $p = 2$

$$((x - l)_+^2)' = \begin{cases} 0 & x < l \\ 2(x - l) & x \geq l \end{cases}$$

$$((x-l)_+^2)'' = \begin{cases} 0 & x < l \\ 2 & x \geq l \end{cases}$$

Nel caso di $p = 2$, la funzione e' derivabile una sola volta. In generale, si verifica facilmente che per $j \leq p-1$,

$$D^{(j)}(x-l)_+^p|_{x \in \{l^-,\ l^+\}} = 0,$$

e quindi possiamo verificare che $(x-l)_+^p$ e' di classe $C^{p-1}$ essendo

$$((x-l)_+^2)^p = \begin{cases} 0 & x < l \\ p! & x \geq l \end{cases}$$

Dall'analisi delle proprieta' delle potenze troncate, quali per esempio la loro lineare indipendenza, possiamo concludere che

$$< 1, x, x^2, ..., x^p, (x-y_1)_+^p, (x-y_2)_+^p, ..., (x-y_m)_+^p > \tag{11}$$

e' una base per lo spazio delle funzioni splines.

## 5.2 Funzioni spline interpolanti

Le funzioni spline introdotte sono una nuova classe di funzioni costruita a partire da un insieme di *nodi* che rappresentano soltanto i punti in cui suddividere l' intervallo di lavoro $[a, b]$. I nodi della spline non sono da confondere con i nodi dell' interpolazione e, in generale, una spline non e' ne interpolante ne approssimante ma solo una particolare funzione. Tuttavia, come tutte le classi di funzioni, le splines possono senz'altro essere utilizzate sia per interpolare che per approssimare. Ma i dati del problema di interpolazione/approssimazione ed i nodi della splines sono due cose distinte.

Proseguiamo analizzando l'utlizzo delle funzioni splines per risolvere un problema di interpolazione. Dato un intervallo $[a, b]$ nel quale siano fissati $m+2$ nodi, $a = y_0 < y_1 < ... < y_m < y_{m+1} = b$, e dati $n+1$ punti $(x_i, f_i)$ con $i = 0, 1, ..., n$, si cerca una funzione spline interpolante e cioe' che soddisfi le condizioni

$$S_{p,y}(x_i) = f_i, \quad \text{per} \quad i = 0, 1, ..., n.$$

Poiche' il numero di condizioni di interpolazione $(n+1)$ non puo' superare il numero di gradi di liberta' $(m+p+1)$, (questo perche' si potrebbe verificare la situazione in cui sia impossibile soddisfare i vincoli) se vogliamo essere certi dell' esistenza della spline dobbiamo supporre che $m, p, n$ soddisfino la relazione

$$n + 1 \leq m + p + 1,$$

quindi

$$n \leq m + p.$$

L'insieme dei punti di interpolazione e l'insieme dei nodi possono essere totalmente distinti tra loro, parzialmente o totalmente coincidenti. Consideriamo il caso di una spline lineare: se in un intervallo $[y_i, y_{i+1})$ cadono due punti di interpolazione, in tale intervallo viene definita univocamente una retta interpolante; supponiamo ora che nell'intervallo contiguo $[y_{i+1}, y_{i+2})$ ci siano altri due punti di interpolazione e quindi, di nuovo, un tratto di retta interpolante univocamente determinato. A questo punto, non c'e' modo di imporre le condizioni di raccordo tra i due tratti di retta adiacenti. Lo studio delle relazioni fra nodi di una spline e punti fondamentali dell'interpolazione esula dallo scopo di queste pagine ma puo' essere facilmente trovato in letteratura. Se, come spesso accade, i nodi e punti fondamentali dell'interpolazione coincidono (allora risulta $n = m + 1$) allora la condizione $n \leq m + p$ sara' senz'altro verificata ma ci sono piu' parametri liberi (nella definizione della spline) che vincoli (quelli di interpolazione). Infatti restano $(m + p + 1) - (m + 2)$ parametri liberi.

Le spline cubiche interpolanti in $m + 1$ nodi (che sono le piu' usate) sono definite a meno di 2 gradi di liberta' che vanno in qualche modo fissati. Classiche modalita' per fissare i due rimanenti gradi di liberta' sono

- se si hanno delle informazioni aggiuntive sul comportamento della derivata agli estremi di definisce univocamente una spline (detta *spline completa*) fissando
$$S^{'}(a) = f^{'}(a) \quad \text{e} \quad S^{'}(b) = f^{'}(b)$$
- se si hanno delle informazioni aggiuntive sul comportamento della derivata seconda agli estremi di definisce univocamente una spline fissando
$$S^{''}(a) = f^{''}(a) \quad \text{e} \quad S^{''}(b) = f^{''}(b);$$
- se si fissa a zero le derivate seconde agli estremi e cioe'
$$S^{''}(a) = S^{''}(b) = 0;$$
in tal caso la spline e' detta *spline naturale*
- se non sono note le derivate della funzione $f$, si puo' usare la condizione di *not-a-knot*: si chiede che la spline abbia derivata terza continua nel primo e nell' ultimo nodo interno.

Sempre per le spline cubiche sono noti i seguenti risultati di approssimazione:

**Teorema 8.** *Sia $f \in C^4[a, b]$ e sia $S$ una spline cubica completa o naturale interpolante nei nodi, allora:*

$$\max |f^{(k)}(x) - S^{(k)}_{3,y}(x)| \leq C_k \Delta^{3-k} \max_{x \in [a,b]} |f^{(k)}(x)|, \quad per \quad k = 0, 1, 2, 3$$

*dove*

$$\Delta = \max |y_{i+1} - y_i|, \quad per \quad i = 0, ..., m.$$

Poiche' $\Delta = \max_{i=0,\cdots,m} |y_{i+1} - y_i|$ tende a zero all'aumentare del numero dei nodi, il precedente teorema ci mostra che si ha la convergenza della spline alla funzione (e anche delle sue derivate); la convergenza e' meno veloce man mano che il grado della derivata sale.
Le spline cubiche sono quelle che per cui e' minimo l'integrale della derivata seconda, infatti si ha

**Teorema 9.** *Per ogni funzione $f \in C^2[a,b]$ passante per i punti $(x_i, f_i)$, $i = 0, ..., n$, sia $S_{3,y}$ la spline cubica naturale interpolante nei nodi . Si ha*

$$\int_a^b |S''_{3,y}(x)|^2 dx \leq \int_a^b |f''(x)|^2 dx$$

*Dimostrazione.* Per semplicita' denotiamo $S = S_{3,y}$. Si ha

$$0 \leq \int_a^b [f''(x) - S''(x)]^2 \, dx = \int_a^b [f''(x)]^2 - 2\int_a^b [f''(x) - S''(x)]S''(x) \, dx - \int_a^b [S''(x)]^2 \, dx. \tag{12}$$

Per ogni sotto-intervallo $[y_i, y_{i+1}]$ individuato dai nodi, si ottiene, integrando due volte per parti

$$\begin{aligned} \textstyle\int_{y_i}^{y_{i+1}} [f''(x) - S''(x)]S''(x) \, dx &= \Big[[f'(x) - S'(x)]S''(x)\Big]_{y_i}^{y_{i+1}} \\ &\quad - \Big[[f(x) - S(x)]S'''(x)\Big]_{y_i}^{y_{i+1}} \\ &\quad + \textstyle\int_{y_i}^{y_{i+1}} [f(x) - S(x)]S^4(x) \, dx. \end{aligned}$$

Poiche' $S$ nell' intervallo $[y_i, y_{i+1}]$ coincide con un polinomio di grado al piu' 3 e'ovvio che in tale intervallo $S^4(x) = 0$ ed inoltre, per l' interpolazione, $S(x_i) = f(x_i)$ e $S(x_{i+1}) = f(x_{i+1})$. Pertanto

$$\int_a^b [f''(x) - S''(x)]S''(x) \, dx = \sum_{i=0}^{n-1} \int_{y_i}^{y_{i+1}} [f''(x) - S''(x)]S''(x) = 0$$

poiche' la spline e'naturale e $S''(a) = S''(y_0) = 0$ e $S''(b) = S''(y_m) = 0$. In conclusione, da (12) concludiamo

$$0 \leq \int_a^b [f''(x)]^2 - \int_a^b [S''(x)]^2 \, dx.$$

e percio'

$$\int_a^b [S''(x)]^2 \leq \int_a^b [f''(x)]^2 \, dx.$$

□

## 5.3 B-splines

La costruzione delle spline mediante potenze troncate e' molto semplice, ma presenta alcuni svantaggi: nel caso in cui si voglia valutare la spline in un certo punto nell' intervallo $[a, b]$, essa richiede un costo computazionale che cresce man mano che il punto si sposta da $a$ verso $b$. Cio' dipende proprio dalla definizione di potenza troncata che diventa diversa da zero a partire dal nodo a cui si riferisce. Per questo motivo la definizione delle spline mediante potenze troncate non viene abitualmente usata, e si preferisce una formulazione alternativa, mediante le cosiddette *B-splines*. La base B-spline e' costituita da splines a supporto compatto con forma a "campana" e contiene, ovviamente, tanti elementi quanti la base delle potenze troncate (11). Quindi, la base B-splines e' del tipo

$$< B_{p,-p}(x), \cdots, B_{p,m}(x) >$$

ed i suoi elementi sono splines in $S_{p,y}$, positive, a supporto compatto, con forma a campana e,come vedremo in seguito, sono calcolabili in modo efficiente utilizzando una tecnica ricorsiva.

### 5.3.1 Un esempio semplice: il caso lineare

Nel caso lineare la base B-splines per lo spazio $S_{1,y}$ con $m$ nodi interni in $[a, b]$ e' costituita da $m + 2$ elementi:

$$< B_{1,-1}, B_{1,0}, ..., B_{1,m} > .$$

Ogni spline lineare sara' quindi combinazione lineare di queste funzioni di base e quindi del tipo

$$S_{1,y}(x) = \sum_{i=-1}^{m} b_i B_{1,i}(x),$$

dove $B_{1,i}(x)$ rappresenta la $i$-esima B-spline lineare e $b_i$ l' $i$-esimo coefficiente. Per costruire $B_{1,i}$ consideriamo una spline lineare con forma a campana e a supporto $[y_i, y_{i+2}]$ del tipo

$$B_{1,i}(x) = a(x - y_i)_+ + b(x - y_{i+1})_+ + c(x - y_{i+2})_+ .$$

Imponendo le condizioni $B_{1,i}(y_i) = 0$, $B_{1,i}(y_{i+1}) = 1$, $B_{1,i}(y_{i+2}) = 0$ otteniamo i valori dei coefficienti $a, b, c$.
Appare evidente che mentre $B_{1,i}(x),\ i = 0, \cdots, m$ possono essere costruite con gli $m + 2$ nodi a disposizione $y_i,\ i = 0, \cdots, m + 1$, per costruire $B_{1,-1}$ e $B_{1,m}$ e' necessario aggiungere dei nodi "fittizzi" e cioe' $y_{-1},\ y_{m+2}$.

### 5.3.2 Caso generale e formula di De-Boore

Discutiamo ora la costruzione di una base B-spline di grado $p$ e cioe'

$$< B_{p,-p}(x), \cdots, B_{p,m}(x) >$$

in modo tale che una generica spline $S_{p,y}$ possa essere riscritta come

$$S_{p,y}(x) = \sum_{i=-p}^{m} d_i B_{p,i}(x),$$

dove $d_i,\ i = -p, \cdots, m$ siano opportuni coefficienti.
Ciascuna funzione appartenente alla base, quindi ciascuna B-splines $B_{p,i}(x)$, e' ovviamente una spline $S_{p,y}$ costruita soltanto sui nodi dall' $i$-esimo al $i+p+1$-esimo che ne definisco il supporto; Altre importanti proprieta' delle funzioni B-splines sono:

- Supporto compatto: $[y_i, y_{i+p+1}]$
- Positivita' sul supporto: $B_{p,i}(x) \geq 0,\ x \in [y_i, y_{i+p+1}]$
- Partizione dell' unita' sull' intervallo: $\sum_{i=-p}^{m} B_{p,i}(x) = 1,\ x \in [y_0, y_{m+1}]$;

Per il calcolo delle B-spline si utlizza la formula di *De-Boor*: essa e' ricorsiva, che risulta un grosso vantaggio in termina di efficienza computazionale. Il punto di partenza e' la B-spline di grado 0 definita come

$$B_{0,i} = \begin{cases} 1 & y_i \leq x \leq y_{i+1} \\ 0 & altrimenti \end{cases}, \quad i = 0, \cdots, m$$

quindi quella di grado $p$ viene calcolata come

$$B_{p,i}(x) = \frac{(x - y_i)}{(y_{i+p} - y_i)} B_{p-1,i}(x) + \frac{(y_{i+p+1} - x)}{(y_{i+p+1} - y_{i+1})} B_{p-1,i+1}(x), \quad i = -p, \cdots, m,$$

per cui servono $p+1$ nodi ausiliari. In particolare, nel caso lineare, otteniamo la seguente espressione

$$B_{1,i}(x) = \frac{(x - y_i)}{(y_{i+1} - y_i)} B_{0,i}(x) + \frac{y_{i+2} - x}{y_{i+2} - y_{i+1}} B_{0,i+1}(x), \quad i = -1, \cdots, m,$$

che ci consente di ottenere la forma analitica della $i$-esima B-spline di primo grado che risulta

$$B_{1,i}(x) = \begin{cases} r_1(x) & y_i < x \leq y_{i+1} \\ r_2(x) & y_{i+1} \leq x < y_{i+2} \\ 0 & \text{per } \ x \leq y_i \ \text{ ex} \geq y_{i+2} \end{cases}$$

dove

$$r_1(x) = \frac{(x - y_i)}{(y_{i+1} - y_i)}, \quad \text{e} \quad r_2(x) = \frac{(x - y_{i+2})}{(y_{i+1} - y_{i+2})}.$$

# 6 Interpolazione parametrica

Scopo di questo paragrafo e'studiare il problema dell' interpolazione parametrica utile nel caso in cui i punti da interpolare non originino da una funzione ma da una curva. Ovviamente una curva puo' essere rappresentata in forma implicita o in forma parametrica. Ad esempio un cerchio di raggio $a$ e centro l'origine puo' essere rappresentato in forma implicita

$$x^2 + y^2 - a^2 = 0,$$

oppure in forma parametrica

$$C(t) = \begin{cases} x(t) = a\cos(2\pi t) \\ y(t) = a\sin(2\pi t) \end{cases} \quad t \in [0,1].$$

Un'altro esempio e'dato dalla *lumaca di Pascal* che in forma implicita ha equazione

$$(x^2 + y^2 - bx)^2 - a^2(x^2 + y^2) = 0$$

mentre in forma parametrica equazione

$$C(t) = \begin{cases} x(t) = \frac{b}{2} + a\cos(2\pi t) + \frac{b}{2}\cos(2\pi t) \\ y(t) = a\sin(2\pi t) + \frac{b}{2}\sin(2\pi t) \end{cases} \quad t \in [0,1] \quad 0 \le a < b.$$

Il cerchio e la lumaca di Pascal sono esempi di curve in $\mathbb{R}^2$ mentre un esempio di curva in $\mathbb{R}^3$ e'dato dall' elica di equazione parametrica

$$C(t) = \begin{cases} x(t) = a\cos(2\pi t) \\ y(t) = b\sin(2\pi t) \\ z(t) = bt \end{cases} \quad t \in [0,1].$$

E' importante ricordare che una curva parametrica nel piano o nello spazio e' una applicazione definita nel seguente modo

$$C : [a,b] \to \mathbb{R}^s, \quad s \in \{2,3\}$$

e che la rappresentazione parametrica permette di esprimere l'equazione di una curva in $\mathbb{R}^2$ o in $\mathbb{R}^3$ o piu' in generale in $\mathbb{R}^n$ sempre in modo analogo. Infine ricordiamo che la rappresentazione parametrica di una curva non e' unica.

Tornando al problema dell' interpolazione dati provevieniti da curve, consideriamo il caso piano, perche' quello nello spazio puo' essere trattato in modo del tutto analogo. Assegnati $n+1$ punti nel piano, $P_0, P_1, ..., P_n$ dove $P_i = (x_i, y_i)$, $i = 0, \cdots, n$, si vuole costruire una curva parametrica

$C : [a, b] \to \mathbb{R}^2$ interpolante tali punti e cioe' tale che, per alcuni valori del parametro $t \in [a, b]$, diciamo $t_0, t_1, \cdots, t_n$ con $t_i \in [a, b]$ risulti

$$C(t_i) = P_i \quad \leftrightarrow \quad \begin{cases} x_1(t_i) = x_i \\ x_2(t_i) = y_i \end{cases} \quad , \quad i = 0, \cdots, n.$$

Rispetto al caso "funzionale" il nuovo problema che si pone adesso e' quello della scelta dei valori del parametro $t$ cioe' dei valori $t_0, t_1, \cdots, t_n$, in cui imporre le condizioni di interpolazione. E' importante notare che tali valori non fanno parte dei dati ma possono essere scelti in modo arbitrario e quindi risultano "parametri" liberi. Una possibile scelta e' quella uniforme (in tal caso si parla di *parametrizzazione* uniforme) in cui i $t_i$ vengono scelti uniformemente nell'intervallo $[a, b]$

$$t_i = a + \frac{(b-a)i}{n} \qquad i = 0, ..., n.$$

Altre paramettrizzazioni sono piu' geometriche, e dipendono dalla posizione dei punti nello spazio come la parametrizzazione *arco della curva*. A partire da un insieme di punti $P_0, P_1, ..., P_n$ in $\mathbb{R}^2$ si calcolano le distanze tra due punti consecutivi, ovvero

$$d_i = \overline{P_{i-1}P_i} = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}, \quad i = 1, \ldots, n.$$

Quindi i valori $t_0, t_1, \cdots, t_n$ del parametro $t$ in cui interpolare sono scelti nel seguente modo

$$t_0 = 0, \quad d = \sum_{i=1}^{i} d_n, \quad t_i = \sum_{i=1}^{i} d_i/d \qquad i = 1, ..., n$$

in cui $t_i$ rappresenta un valore del parametro proporzionale alla lunghezza cumulata della spezzata che, a partire dal punto $P_0$, collega tutti i punti fino a $P_i$.

Una volta risolto il problema della scelta dei parametri, l' interpolazione si realizza utilizzando un qualunque metodo applicato componente per componente. Per esempio, nel caso dell'interpolazione di Lagrange si producono due polinomi interpolanti di Lagrange: Il primo, $L_n^x(t)$ associato ai dati $(t_i, x_i),\ i = 0, \cdots, n$ ed il secondo $L_n^y(t)$ associato ai dati $(t_i, y_i),\ i = 0, \cdots, n$. Poiche' soddisfano

$$L_n^x(t_i) = x_i \quad L_n^y(t_i) = y_i \quad i = 0, \cdots, n,$$

in conclusione, $C(t) = (L_n^x(t), L_n^y(t)), t \in [t_0, t_n]$ e' l'interpolante parametrico polinomiale cercato che soddisfa

$$C(t_i) = (L_n^x(t_i), L_n^y(t_i) = (x_i, y_i) = P_i,\ i = 0, \cdots, n.$$

# 7 Sistemi lineari rettangolari: il problema lineare dei minimi quadrati

In questo capitolo ci proponiamo di studiare come risolvere un sistema lineare *rettangolare* cioe' in cui la matrice dei coefficienti non sia quadrata come abitualmente accadde. Sostanzialmente siamo interessati a risovere il problema

$$Ax = b, \quad A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m, \quad m \neq n. \tag{13}$$

Come vedremo piu' avanti un problema di questo tipo e' connesso con la determinazione di funzioni *approssimanti* nel senso dei *minimi quadrati*.

## 7.1 Esistenza ed unicita' della soluzione

Cominciamo con l' osservare che un sistema lineare del tipo (13) puo' non avere soluzione. Pertanto, fissata una norma vettoriale, $\|\cdot\|$, si risolve un problema ad esso legato cercando i vettori che minimizzano la quantita' $\min_{y\in\mathbb{R}^n} \|Ay - b\|$. Se la norma indotta e'la norma due (denotata con $\|.\|_2$ e definita come $\|y\|_2 = (y^T y)^{\frac{1}{2}}$) il precedente problema di minimo viene chiamato *problema dei minimi quadrati* e risulta

$$\text{determinare} \quad x \in \mathbb{R}^n \quad \text{tale che} \quad \|Ax - b\|_2 = \min_{y\in\mathbb{R}^n} \|Ay - b\|_2. \tag{14}$$

Osserviamo che se esiste un vettore tale che $Ax = b$ allora $\|Ax - b\|_2 = 0$ e percio' il vettore soluzione del sistema lineare (13), se esiste, e' senz'altro soluzione del *problema dei minimi quadrati* (14). Tuttavia il viceversa non e' vero.

Quello che a noi interessa e' studiare la risoluzione di un sistema lineare *sovradeterminato* ($m > n$, e piu' in generale $m >> n$) ovvero un sistema che ha piu' equazioni che incognite. Questo e' il caso che considereremo da ora in poi.

## 7.2 Risoluzione mediante il sistema delle equazioni normali

**Teorema 10.** *Il problema di minimo* $\min_{y\in\mathbb{R}^n} \|Ay - b\|_2 = \|Ax - b\|_2$ *ha soluzione se e solo se*

$$A^\top Ax = A^\top b.$$

*Dimostrazione.* Partendo dalla definizione, abbiamo $\|Ay - b\|_2^2 = (Ay - b)^\top (Ay - b) = (y^\top A^\top - b^\top)(Ay - b) = y^\top A^\top Ay - y^\top A^\top b - b^\top Ay + b^\top b$. Quindi derivando rispetto ad $y$ e imponendo che le derivate siano zero (sostanzialmente che il gradiente della funzione $F(x) = Ax - b$), non e' difficile verificare che $x$ e' soluzione se e solo se $A^\top Ax = A^\top b$ □

Per quanto riguarda la caratterizzazione dello spazio $X$ delle soluzioni vale il seguente risultato:

- $X \neq \emptyset$
- $\exists x^* \in X : \|x^*\|_2 = \min_{x \in \mathbb{X}} \|x\|_2$
- $x \in X \iff A^\top Ax = A^\top b$
  $(A^\top Ax = A^\top b)$ sono dette *Equazioni Normali*

L'equazione $A^\top Ax = A^\top b$ si chiama *sistema delle equazioni normali.* Ovviamente, ci sono due possibilita':

**i)** A ha rango massimo

**ii)** A non ha rango massimo

dove per *rango* si intende il massimo numero di colonne (o righe) linearmente indipendenti in A. Vale la pena distinguere i due casi:

**A con rango massimo:** Sotto tale ipotesi, la matrice quadrata $A^\top A$ e' non singolare e pertanto, in linea teorica, si puo' procedere alla risoluzione del sistema delle equazioni normali utilizzando un metodo qualunque. Tuttavia se $A$ e' mal condizionata (piccole perturbazioni negli elementi di $A$, o piccole variazioni del vettore $b$, possono produrre grandi variazioni nelle soluzioni) il condizionamento di $A^\top A$ puo' essere ancora peggiore (all'incirca il quadrato del condizionamento di $A$). Questo fatto si puo' vedere andando a stimare l'*indice di condizionamento* di $A^\top A$. Si ha

$$K(A^\top A) = \|A^\top A\|\|(A^\top A)^{-1}\| \leq \|A^\top\|\|A\|\|A^{-1}\|\|(A^\top)^{-1}\| \simeq (K(A))^2 \tag{15}$$

La questione legata al condizionamento della matrice $A^\top A$, scoraggia la soluzione del sistema delle equazioni normali e motiva la risoluzione del problema (14) direttamente utilizzando tecniche basate sulla fattorizzazione $QR$ di $A$ o sulla decomposizione $SVD$ di $A$;

**A non a rango massimo:** Se $A$ non e' a rango massimo la questione e' piu' complicata perche' $A^\top A$ e' una matrice singolare e la soluzione del sistema delle equazioni normali non sara' unica. Anche in questo caso si cerca una soluzione del problema (14) direttamente via fattorizzazione $QR$ e via $SVD$.

Prima di studiare la soluzione del problema (14) direttamente via fattorizzazione $QR$, vediamo ora due applicazioni in cui e' utile la risoluzione del problema (14).

## 7.3 Migliore approssimazione ai m.q. polinomiale.

Assegnato l'insieme dei dati $(x_i, f_i)$ con $i = 0, ....., n$, fissato $m$ tale che $n >> m$ si cerca il polinomio $p_m(x) = a_0 + a_1 x + ... + a_m x^m$ che meglio approssima i dati nel seguente senso: definito il *vettore errore* di componenti $e_i = p_m(x_i) - f_i,\ i = 0, \cdots, n$ si cerca il polinomio (e quindi la $m+1$-pla $a_0, ..., a_m$) che rende minimo tale errore rispetto alla norma due. In sostanza si risolve il seguente problema di minimo

$$\min_{a_0,...,a_m} \|e\|_2$$

o piu' specificatamente

$$\min_{a_0,...,a_m} \left\| \begin{bmatrix} a_0 + a_1 x_0 + a_2 x_0^2 + ... + a_n x_0^m - f_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + ... + a_n x_1^m - f_1 \\ \vdots \\ a_0 + a_1 x_n + a_2 x_n^2 + ... + a_n x_n^m - f_n \end{bmatrix} \right\|_2 .$$

Il vettore $e$ puo' essere convenientemente scritto come $e = Ay - b$ dove

$$A = \begin{bmatrix} 1 & x_0 & x_0^2 & . & . & x_0^m \\ 1 & x_1 & x_1^2 & . & . & x_1^m \\ & & & . & & \\ & & & . & & \\ & & & . & & \\ 1 & x_n & x_n^2 & . & . & x_n^m \end{bmatrix}, = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} \text{ e } b = \begin{bmatrix} f_0 \\ f_1 \\ . \\ . \\ . \\ f_n \end{bmatrix}$$

Pertanto il problema di approssimazione si riduce proprio ad un problema del tipo (14) in cui $n >> m$:

$$\min_{a_0,...,a_n} \|e\|_2 = \min_{y=(a_0,...,a_n)} \|Ax-b\|_2, \quad A \in \mathbb{R}^{(n+1)\times(m+1)},\ b \in \mathbb{R}^{(n+1)}, \quad m \neq n.$$

E' da notare che $A$ e' la matrice di Vandermonde associata ai nodi $x_0, \cdots, x_n$ e quindi *mal condizionata*. Inoltre, se $m = n$ la matrice $A$ e' quadrata e la soluzione e' data dal polinomio interpolante che genera il vettore errore $e = [0, \cdots, 0]^T$. Per il Teormema 10 la risoluzione del problema e' equivalente alla risoluzione del sistema lineare $A^\top A = A^\top b$ e dove:

$$A^\top A = \begin{bmatrix} n+1 & \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 & \cdots & \sum_{i=0}^n x_i^n \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 & \sum_{i=0}^n x_i^4 & \cdots & \\ & & \vdots & & & \\ \sum_{i=0}^n x_i^n & & \cdots & & & \sum_{i=0}^n x_i^{2n} \end{bmatrix} \quad (16)$$

che e' simmetrica e definita positiva e

$$A^\top b = \begin{bmatrix} \sum_{i=0}^n f_i \\ \sum_{i=0}^n x_i f_i \\ \sum_{i=0}^n x_i^2 f_i \\ \vdots \\ \sum_{i=0}^n x_i^n f_i \end{bmatrix}. \tag{17}$$

**Spline di migliore approssimazione:** Assegnato l'insieme dei dati $(x_i, f_i)$ con $i = 0, ....., n$, fissati $mp$ tale che $n >> m + p + 1$ si cerca una spline $S_{y,p}(x) = \sum_{i=0}^p a_i x^i + \sum_{i=1}^m c_i (x - y_i)_+^p$ che meglio approssima i dati nel senso dei minimi quadrati. Come nel caso polinomiale, consideriamo il vettore errore $e_i = S_{y,p}(x_i) - f_i$ con $i = 0, ....., n$. Ragionando come nel caso precedente e' facile vedere che il problema si riduce alla determinazione del vettore di dimensione $m + p + 1$, $y = [a_0, a_1, \cdots, a_p, c_1, \cdots, c_m]^T$ soluzione del problema di minimo $\min_{y \in \mathbb{R}^{m+p+1}} \|Ay - b\|_2$ dove

$$A = \begin{bmatrix} 1 & x_1 & \cdots & x_1^p & (x_1 - y_1)_+^p & \cdots & (x_1 - y_m)_+^p \\ & & \vdots & & & & \\ 1 & x_l & \cdots & x_l^p & (x_l - y_1)_+^p & \cdots & (x_l - y_m)_+^p \end{bmatrix} \tag{18}$$

e', ancora una volta, una matrice mal condizionata e $b$ e'il vettore di $n + 1$ componenti $b = [f_0, \cdots, f_n]^T$.

## 7.4 Matrici ortogonali: le matrici di Householder

Studiamo quindi la soluzione del problema (14) direttamente via fattorizzazione $QR$. Cominciamo col ricordare la definizione di matrice ortogonale.

**Definizione 1.** Una matrice $Q \in \mathbb{R}^{n \times n}$ e' una matrice ortognonale se

$$Q^T Q = Q Q^T = I \tag{19}$$

ovvero se la trasposta di Q e' uguale alla sua inversa.

Partizionando la matrice $Q$ in base alle colonne nella seguente forma

$$Q = \left[ \; Q_1 \; \middle| \; Q_2 \; \middle| \; \ldots \; \middle| \; Q_n \; \right] \tag{20}$$

si osserva l' ortogonalita' delle stesse:

$$Q_i^T Q_j = \delta_{ij} = \begin{cases} 0, & \text{se } i \neq j \\ 1, & \text{se } i = j. \end{cases} \tag{21}$$

Proseguiamo con la definizione di *matrice di Householder*.

**Definizione 2.** Dato un vettore $u \in \mathbb{R}^n$, la matrice di Hauseholder ad esso associata, $H_u \in \mathbb{R}^{n\times n}$, e' la matrice di rango 1 correzione dell' identita'

$$H_u = I - \alpha uu^\top, \quad \alpha = \frac{2}{u^T u} \in \mathbb{R}. \tag{22}$$

E' facile dimostrare le seguente risultato.

**Proposizione 2.** *Una matrice $H_u$ di Hauseholder soddisfa:*

- *$H_u$ e' simmetrica: $H_u^\top = H_u$*
- *$H_u$ e' ortogonale: $H_u^\top H_u = I$*
- *$H_u$ e' involutoria: $H_u^2 = I$ (segue dalle due precedenti)*

*Osservazione* 1. E' utile ricorda che la norma due e' invariante per trasformazioni ortogonali e cioe' se $Q \in \mathbb{R}^{n\times n}$ e' una matrice ortogonale, allora dato $x \in \mathbb{R}^n$:

$$||Qx||_2 = ||x||_2. \tag{23}$$

Infatti:

$$||Qx||_2^2 = (Qx)^T Qx = x^T Q^T Qx = x^T x = ||x||_2^2. \tag{24}$$

Un' altra utile osservazione: Assegnato $u \in \mathbb{R}^2$, Introduciamo un vettore $w$ ortogonale a $u$ (ovvero tale che $w^\top u = 0$), definendo la coppia $\langle u, w\rangle$ come base. Per questo motivo e' possibile scrivere ogni vettore $x \in \mathbb{R}^2$ come :

$$x = au + bw, \quad a, b \in \mathbb{R} \tag{25}$$

Se costruiamo ma matrice di Hauseholder $H_u$ associata ad $u$ e l'applichiamo ad $x$ costruendo $y = H_u x$ risulta

$$H_u x = (I - \alpha uu^\top)x = x - \alpha u(u^\top x) \tag{26}$$

con

$$u^\top x = u^\top(au + bw) = au^\top u. \tag{27}$$

Per la (25):

$$H_u x = x - \alpha u\,(u^\top x) = au + bw - (\frac{2}{u^\top u} au^\top u)u = au + bw - 2au = -au + bw \tag{28}$$

In conclusione, dal punto di vista geometrico, in $\mathbb{R}^2$, una trasformazione di Hauseholder associata ad un vettore $u$ opera una riflessione del vettore $x$ a cui e' applicata rispetto al vettore $w$ ortogonale ad $u$. Per questo motivo la matrice $H_u$ viene detta *riflettore di $x$* ed il vettore $y = H_u x$ viene detto *riflesso di $x$.*

**Teorema 11.** *Sia $A \in \mathbb{R}^{m\times n}$, allora $\exists\ Q \in \mathbb{R}^{m\times m}$ matrice ortogonale ed $\exists R \in \mathbb{R}^{m\times n}$ triangolare superiore tale che $A = QR$.*

*Dimostrazione.* Questa dimostrazione e' costruttiva nel senso che l'esistenza della fattorizzazione QR e' evidenziata nella costruzione. Cominciamo mostrando che, dato $x$ possiamo trovare $u$ tale che:

$$H_u x = \alpha e_1 \tag{29}$$

In tal caso deve essere vero anche che :

$$||x||_2 = ||H_u x||_2 = ||\alpha e_1||_2 = |\alpha| \tag{30}$$

e quindi $\alpha = \pm||x||_2$. Scegliamo

$$u = x \pm ||x||_2 e_1 \tag{31}$$

Con tale scelta non e' difficile verificare che

$$\begin{aligned} u^\top u = (x \pm ||x||_2 e_1)^\top (x \pm ||x||_2 e_1) = (\pm||x||_2 e_1^\top + x^\top)(x \pm ||x||_2 e_1) = \\ = \pm 2||x||_2 x_1 + 2||x||_2^2 = 2||x||_2(||x||_2 \pm x_1) \end{aligned} \tag{32}$$

$$u^\top x = ||x||^2(||x||_2 \pm x_1) \tag{33}$$

e quindi

$$H_u x = x - \frac{2||x||_2(||x||_2 \pm x_1)}{2||x||_2(||x||_2 \pm x_1)}(x \pm ||x||_2 e_1) = \pm||x||_2 e_1 \tag{34}$$

Si noti che, vista questa ambiguita' sui segni, la fattorizzazione QR non sara' mai unica. Dovendo costruire u in pratica si puo' scegliere:

$$u = x + sgn(x_1)||x||_2 e_1 \tag{35}$$

Piu' generale, dato $x \in \mathbb{R}^m$, se vogliamo determinare $u$ tale che $H_u x = [\underbrace{*...*}_{k}, 0...0]^\top$ (con $m-k$ elementi nulli), possiamo usare la strategia precedente come segue. Costruiamo il sotto-vettore $\tilde{x} = [x_k, ..., x_m]^\top \in R^{m-k+1}$ ed il corrispondente $\tilde{u} = \tilde{x} \pm ||x||_2 e_1$. Definiamo $H_{\tilde{u}} \in R^{(m-k+1)\times(m-k+1)}$ e quindi

$$H_u = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_{\tilde{u}} \end{bmatrix} \in R^{m\times m}. \tag{36}$$

Non e' difficile verificare che

$$H_u x = [x_1, ...x_{k-1}, \pm||\tilde{x}||_2, 0...0]^\top$$

e che la matrice $H_u$ e' effettivamente la matrice di Householder associata al vettore $u^\top = [\underbrace{0, ..., 0}_{k}, \tilde{u}^\top]$. Si noti anche che l'applicazione di $H_u$ definita in (36) non modifica le prime $k-1$ componenti di un generico vettore. Con le matrici di Householder precedentemente descritte, la fattorizzazione $QR$ e' ottenuta iterativamente: al primo passo si costruisce una matrice di Householder $H_1$ tale che:

$$H_1 A = \left[\begin{array}{c|c} \alpha_{11} & \\ 0 & \vdots \\ \vdots & \vdots \\ 0 & \end{array}\right] \tag{37}$$

dove $\alpha_{11}$ e' un numero reale. Al secondo passo si costruisce una matrice di Householder $H_2$, tale che:

$$H_2 H_1 A = \left[\begin{array}{cc|c} \alpha_{11} & \alpha_{12} & \\ 0 & \alpha_{22} & \\ 0 & 0 & \vdots \\ \vdots & \vdots & \vdots \\ 0 & 0 & \end{array}\right] \tag{38}$$

con $\alpha_{i2},\ i = 1, 2$ numeri reali. Al terzo passo si costruisce una matrice di Householder $H_3$, tale che:

$$H_3 H_2 H_1 A = \left[\begin{array}{ccc|c} \alpha_{11} & \alpha_{12} & \alpha_{13} & \\ 0 & \alpha_{22} & \alpha_{23} & \\ 0 & 0 & \alpha_{33} & \vdots \\ 0 & 0 & 0 & \vdots \\ \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & \end{array}\right] \tag{39}$$

con $\alpha_{i3},\ i = 1, 2, 3$ numeri reali. Continuando in modo analogo, si arriva all'ultimo passo $n$ costruendo una matrice $H_n$ tale che:

$$H_n = \left[\begin{array}{ccc|c} & & & 0 \\ & I_{n-1} & & \vdots \\ & & & 0 \\ \hline 0 & \dots & 0 & q_{nn} \end{array}\right] \tag{40}$$

Si ottiene percio':

$$H_n H_{n-1} \dots H_1 A = R \tag{41}$$

dove $R$ e' una matrice triangolare superiore. Pertanto, la matrice ortogonale $Q$ tale che $A = QR$ e' la matrice

$$Q = (H_n H_{n-1} \dots H_1)^\top = H_1^\top H_2^\top \dots H_n^\top. \tag{42}$$

□

*Osservazione* 2. La matrice $Q$ pur essendo ortogonale non e' simmetrica, per cui NON puo' essere una matrice di Householder.

La tecnica iterativa sopra descritta e'meglio sintetizzata nel seguente algoritmo che utilizza una sintassi Matlab.

**Algoritmo per la fattorizzazione QR** :

**Data**: A
**Result**: Q, R
[m,n] = size(A);
M = n;
**if** *m<=n* **then**
  M = m-1;
**end**
Q = eye(m);
**for** *k = 1:M* **do**
  v = A(k:m,k);
  u = v +norm(v)*sign(v).*($e_1$$^{\mathrm{T}}$);
  $H_k$ = eye(m);
  $H_k$(k:m,k:m) = eye(m-k+1) - (2/(u$^{\mathrm{T}}$*u))*(u*u$^{\mathrm{T}}$);
  A = $H_k$*A;
  Q = Q*$H_k$;
**end**
R = A.

## 7.5 Risoluzione del problema lineare dei minimi quadrati utilizzando la fattorizzazione QR

Mediante l'utilizzo delle matrici di Hauseholder e' possibile risolvere il problema dei minimi quadrati.
Dati $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ si vuole risolvere il seguente problema:

$$\min_{x\in\mathbb{R}^n} ||Ax - b||_2^2. \tag{43}$$

Come descritto nella sezione precedente e' possibile fattorizzare la matrice $A$ nella forma $A = QR$ dove $Q \in \mathbb{R}^{m\times m}$ e' una matrice ortogonale ed $R \in \mathbb{R}^{m\times n}$ e' una matrice triangolare superiore non singolare:

$$||Ax - b||_2^2 = ||QRx - b||_2^2 = ||Q(Rx - Q^T b)||_2^2 = ||Rx - Q^T b||_2^2 \tag{44}$$

ponendo $c = Q^T b$ segue che:

$$\min_{x\in\mathbb{R}^n} ||Ax - b||_2^2 = \min_{x\in\mathbb{R}^n} ||Rx - c||_2^2. \tag{45}$$

In virtu' delle rispettive proprieta', $R$ e $c$ possono essere partizionate nelle seguenti forme:

$$R = \left[\begin{array}{c} U \\ \hline 0 \end{array}\right] \qquad c = \left[\begin{array}{c} c_1 \\ \hline c_2\ , \end{array}\right] \tag{46}$$

ottenendo una matrice $U \in \mathbb{R}^{n\times n}$ triangolare superiore ed una coppia di vettori $c_1 \in \mathbb{R}^n$ e $c_2 \in \mathbb{R}^{m-n}$.
Da quest'analisi segue che:

$$Rx - c = \left[\begin{array}{c} U \\ \hline 0 \end{array}\right] \left[x\right] - \left[\begin{array}{c} c_1 \\ \hline c_2 \end{array}\right] = \left[\begin{array}{c} Ux - c_1 \\ \hline -c_2 \end{array}\right] \tag{47}$$

e conseguentemente:

$$||Rx - c||_2^2 = ||Ux - c_1||_2^2 + ||c_2||_2^2. \tag{48}$$

Poiche'

$$\min_{x\in\mathbb{R}^n} ||Rx - c||_2^2 = \min_{x\in\mathbb{R}^n} ||Ux - c_1||_2^2 + \min_{x\in\mathbb{R}^n} ||c_2||_2^2 = \min_{x\in\mathbb{R}^n} ||Ux - c_1||_2^2 + ||c_2||_2^2 \tag{49}$$

si arriva alla seguente rifornumalzione del problema di minimo

$$\min_{x\in\mathbb{R}^n} ||Ax - b||_2^2 = \min_{x\in\mathbb{R}^n} ||Ux - c_1||_2^2 + ||c_2||_2^2\,. \tag{50}$$

Per la sua risoluzione si hanno essenzialmente due possibilita'

- Se il rango della matrice $A$ e' massimo $U$ e' non singolare e quindi l'equazione lineare $Ux = c_1$ ammette una ed una sola soluzione $x = U^{-1}c_1$ e:
$$\min_{x\in\mathbb{R}^n} ||Ax - b||_2^2 = ||c_2||_2^2 \tag{51}$$
- Altrimenti, se il rango di $A$ non e' massimo, non si ha un'unica soluzione al problema dei minimi quadrati. Per ottenerle si puo' ricorrere ad altre tecniche (non trattate in questo corso).

## 7.6 La migliore approssimazione ai minimi quadrati trigonometrica ed il caso particolare dell'interpolazione

Il problema dell'approssimazione di una funzione $f(x)$ *periodica* di periodo $2\pi$, viene generalmente affrontato utilizzando, come classe $\mathbb{F}$ di funzioni approssimanti, l'insieme dei polinomi trigonometrici. Si pone percio' per $x \in [0, 2\pi)$,

$$\widetilde{f}_m(x) = \frac{a_0}{2} + \sum_{k=1}^{m}(a_k \cos kx + b_k \sin kx) \tag{52}$$

e si determinano i $2m+1$ coefficienti $\{a_k\}$,$\{b_k\}$ in base ad un dato criterio di approssimazione. Ad esempio, tali coefficienti possono essere quelli dello sviluppo in *serie di Fourier* e cioe'

$$a_k = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos kx dx \qquad k = 0, 1, 2, .... \tag{53}$$

$$b_k = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin kx dx \qquad k = 1, 2, .... \tag{54}$$

Con questi coefficienti, e' noto che sotto opportune ipotesi (per esempio se $f$ e' continua in $[0, 2\pi)$) l'approssimazione (52) converge alla funzione $f$ al crescere di $n$:

$$\lim_{m\to\infty} \widetilde{f}_m(x) = f(x). \tag{55}$$

Un'altro approccio e' quello ai minimi quadrati, dal quale si perviene anche all'interpolazione trigonometrica. Assegnati i dati $(x_i, f(x_i)),\ i = 0, \cdots, n$, $n \geq 2m$ si vuole costruire una funzione $\widetilde{f}_m$ di tipo (52). L'idea e', come gia' visto in precedenza, quella di costruire un vettore errore $e_i = f(x_i) - \widetilde{f}_m(x_i)$ e di determinare i coefficienti della funzione trigonometrica $a_0, \cdots, a_m, b_1, \cdots, b_m$ tali da minimizzare la norma due del vettore errore.

$$\min_{a_0,\cdots,a_n,b_1,\cdots,b_n} \|e\|_2 = \min_{y=(a_0,\cdots,a_n,b_1,\cdots,b_n)} \|Ay - b\|_2, .$$

dove

$$A = \begin{bmatrix} 1 & \cos x_1 & \sin x_1 & \cos 2x_1 & \sin 2x_1 & \cdots & \cdots \\ 1 & \cos x_2 & \sin x_2 & \cos 2x_2 & \sin 2x_2 & \cdots & \cdots \\ & & \vdots & & & & \\ & & \vdots & & & & \\ & & \vdots & & & & \\ & & \vdots & & & & \\ & & \vdots & & & & \\ 1 & \cos x_n & \sin x_n & \cdots & \cdots & \cos mx_n & \sin mx_n \end{bmatrix}, \tag{56}$$

$$y = \begin{bmatrix} \frac{a_0}{2} \\ a_1 \\ b_1 \\ \vdots \\ a_m \\ b_m \end{bmatrix} \quad \text{e} \quad b = \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_n) \end{bmatrix}. \tag{57}$$

Questo problema puo' ovviamente essere risolto con la fattorizzazione $QR$ precedentemente descritta. Tuttavia, se consideriamo il caso in cui gli $n+1$ nodi $x_i,\ i = 0, \cdots, n$ siano equispaziati in $[0, 2\pi)$ ed $n \geq 2m$, e' facile vedere la soluzione puo' essere ottenuta analiticamente senza risolvere nessun sistema lineare passando dalle equazioni normali $A^\top Ax = A^\top b$. Infatti, per $x_i = \frac{2\pi\, i}{n}, \quad i = 0, ..., n$ e per le proprieta' delle funzioni seno e coseno, risulta

$$A^\top A = \begin{bmatrix} n & 0 & 0 & 0 & \cdots & 0 \\ 0 & \frac{n}{2} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & & & \\ 0 & 0 & & \cdots & \frac{n}{2} & 0 \\ 0 & 0 & & \cdots & 0 & n \end{bmatrix} \quad A^\top b = \begin{bmatrix} \sum_{i=0}^n f(x_i) \\ \sum_{i=0}^n f(x_i)\cos x_i \\ \sum_{i=0}^n f(x_i)\sin x_i \\ \vdots \end{bmatrix} \tag{58}$$

ragion per cui pertanto siamo in grado di ricavare direttamente i coefficienti $\{a_k\}$ e $\{b_k\}$ che risultano

$$\begin{cases} a_k = \frac{2}{n}\sum_{i=0}^n f(x_i)\cos kx_i \quad k = 0, 1, 2, ..., m \\ \\ b_k = \frac{2}{n}\sum_{i=0}^n f(x_i)\sin kx_i \quad k = 1, 2, ..., m. \end{cases} \tag{59}$$

Possiamo quindi esplicitare la funzione approssimante $\widetilde{f}_m$ che risulta

$$\widetilde{f}_m(x) = \frac{1}{n}\sum_{i=0}^n f(x_i) + \sum_{k=1}^m (\frac{2}{n}\sum_{i=0}^n f(x_i)\cos kx_i)\cos kx + (\frac{2}{n}\sum_{i=0}^n f(x_i)\sin kx_i)\sin kx \tag{60}$$

Nel caso particolare $n = 2m$ (il numero dei vincoli coincide con la dimensione dello spazio trigonometrico) e quindi $\widetilde{f}_m(x)$ e' il *polinomio interpolante trigonometrico* .

E' da osservare che in questo caso il numero dei punti fondamentali dell' interpolazione e' necessariamente dispari (essendo $n$ pari). Se, tuttavia, i punti da interpolare sono in numero pari (cioe' $n$ e' dispari) si ricerca una funzione trigonometrica approssimante di una forma un po' diversa

$$\widetilde{f}_m(x) = \frac{a_0}{2} + \sum_{k=1}^{m-1}(a_k\cos kx + b_k\sin kx) + \frac{a_m}{2}\cos nx \tag{61}$$

i cui coefficienti $\{a_k\}$ e $\{b_k\}$( ancora espressi dalla (59)) possono essere ottenuti con lo stesso procedimento di prima, ma utilizzando la matrice del sistema delle equazioni normali $\widetilde{A^\top A}$ e' ottenuta da $A^\top A$ sopprimendo le ultime riga e colonna:

$$\widetilde{A^\top A} = \begin{bmatrix} n & 0 & 0 & 0 & \cdots & 0 \\ 0 & \frac{n}{2} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & & & \\ 0 & 0 & & & \cdots & \frac{n}{2} \end{bmatrix} \tag{62}$$

E' possibile dimostrare che i coefficienti $\{a_k\}$ e $\{b_k\}$ sono ancora espressi dalla (59).

# 8 Derivazione numerica

In molte applicazioni pratiche capita di dover calcolare la derivata, fino ad un certo ordine, di una funzione di cui conosciamo il valore solo in certi punti. Può anche succedere che l'espressione analitica di questa funzione renda non immediata o troppo onerosa la sua derivazione e quindi ci sia bisogno di ricorrere alla *derivazione numerica o approssimata*. Una possibilità è quella di utilizzare una *approssimazione* della $f$ e di procedere derivando quella. Dati una funzione $f$ e le coppie di $n+1$ valori $\{x_i, f_i\}$ per $i = 0, 1, \ldots, n$, con $f_i = f(x_i)$, in sostanza il metodo risulta nell'approssimare la derivata di un certo ordine di $f$ nell'$i$-esimo punto $x_i$ con una formula algebrica che richiede di valutare la funzione $f$ in un numero finito di punti.

Un semplice approccio al problema della derivazione numerica di una funzione $f$ è utilizzare la *formula di interpolazione di Lagrange*, utilizzando come nodi i punti $x_i$ e come valori $f_i$ conoscendo o meno l'espressione analitica di $f$.

## 8.1 Derivazione tramite interpolazione di Lagrange

Consideriamo il caso della derivata prima. Come già visto per l'interpolazione di Lagrange, possiamo esprimere una funzione sufficientemente regolare come:

$$f(x) = P_n(x) + E_n(x) = P_n(x) + \frac{w_{n+1}(x) f^{(n+1)}(\xi(x))}{(n+1)!} \tag{63}$$

dove $P_n(x)$ è il polinomio interpolante, $E_n(x)$ è l'errore commesso $\xi(x)$ e'un punto opportuno nell' intervallo dei nodi e $w_{n+1}(x) = (x-x_0)\cdot \ldots \cdot (x-x_n)$. Si può dimostrare che se $f \in C^{n+2}[a,b]$, allora $\exists\ \eta(x) \in [a,b]$ tale che

$$f'(x) = P_n'(x) + \frac{w_{n+1}'(x)}{(n+1)!} f^{(n+1)}(\xi(x)) + \frac{w_{n+1}(x)}{(n+1)!} f^{(n+2)}(\eta(x)) \tag{64}$$

dove $\xi(x), \eta(x) \in [x_0, x_n]$. Tale formula risulta tuttavia poco utile perchè $\xi(x)$ e $\eta(x)$ non si conoscono ed $f$ può non essere così regolare da avere

la derivata $(n+2)$-esima. Se però consideriamo le derivata prima nei nodi fondamentali dell'interpolazione $x = x_j$ otteniamo una espressione piu' semplice:

$$f'(x_j) = P'_n(x_j) + \frac{w'_{n+1}(x_j)}{(n+1)!} f^{(n+1)}(\xi(x_j))\,. \tag{65}$$

### 8.1.1 Caso nodi uniformi

Se i nodi sono assunti uniformi, possiamo effettuare il cambio di variabile

$$x = x_0 + hs \qquad (x_j = x_0 + jh,\ s = j) \tag{66}$$

e sostituendo in $w_{n+1}(x)$ ottieniamo

$$w_{n+1}(x) = (x - x_0) \cdot \ldots \cdot (x - x_n) \quad \Rightarrow \quad w_{n+1}(x_0 + sh) = h^{n+1} \prod_{k=0}^{n} (s - k)\,. \tag{67}$$

D'altra parte, essendo $P'_n(x_j) = \frac{d}{dx} P_n(x)\big|_{x=x_j}$, si ha

$$P'_n(x_j) = \frac{d}{dx} \sum_{i=0}^{n} f_i l_i(x_j) = \frac{d}{ds} \sum_{i=0}^{n} f_i l_i(x_0 + hs) \frac{ds}{dx} = \frac{1}{h} \sum_{i=0}^{n} f_i \frac{d}{ds} l_i(x_0 + sh) \tag{68}$$

e per $\frac{d}{dx} w_n(x)\big|_{x=x_j}$

$$\begin{aligned} \frac{d}{dx} \left(w_{n+1}(x_0 + sh)\right)\Big|_{s=j} &= \frac{d}{ds} \left(w_{n+1}(x_0 + sh)\right)\Big|_{s=j} \frac{ds}{dx} \\ &= \frac{1}{h} h^{n+1} \left.\frac{d}{ds} \left( \prod_{k=0}^{n} (s - k) \right)\right|_{s=j} . \end{aligned} \tag{69}$$

In conclusione possiamo scrivere:

$$f'(x_j) = \frac{1}{h} \sum_{i=0}^{n} f_i \left.\frac{d}{ds} \left(l_i(x_0 + sh)\right)\right|_{s=j} + \frac{f^{n+1}(\xi(x_j))}{(n+1)!} h^n \left.\frac{d}{ds} \left( \prod_{k=0}^{n} (s - k) \right)\right|_{s=j} . \tag{70}$$

Notiamo che essendo $h$ la distanza tra i nodi quando aumentiamo i nodi $h$ tende a 0 e ci possono essere situazioni in cui l'errore potrebbe andare a 0 con $h$. Tuttavia, in generale, non possiamo affermare che la derivata prima del polinomio interpolante converga alla derivata prima della funzione.
In ogni caso, la discussione appena fatta ci motiva a considerare formule di approssimazione della derivata nei nodi di interpolazione della forma

$$f'(x_j) = \frac{1}{h} \sum_{i=0}^{n} f_i C_i \tag{71}$$

dove $C_i$ sono costanti numeriche. Tuttavia se abbiamo dati affetti da errore, cioe' del tipo $\tilde{f}_i = f(x_i) + \epsilon_i$, si verifica facilmente che

$$\frac{1}{h}\sum_{i=0}^{n} \tilde{f}_i C_i = \frac{1}{h}\sum_{i=0}^{n} f_i C_i + \frac{1}{h}\sum_{i=0}^{n} \epsilon_i C_i \tag{72}$$

dove $\frac{1}{h}\sum \epsilon_i C_i$ puo' divergere per $h \to 0$. Per questo motivo non si usano, di fatto, polinomi interpolanti costruiti a partire da troppi punti, perche' si rischia di far esplodere l'errore a causa di $h$ molto piccolo. Le formule piu' utilizzate per approssimare le derivate sono quelle che usano il minor numero possibile di punti.

## 8.2 Derivate approssimate: alcuni semplici esempi

### 8.2.1 Derivata prima

Dato il punto $x_i$ nel quale si vuole calcolare la derivata prima è possibile sfruttare tre tecniche differenti di approssimazione basate sul calcolo del coefficiente angolare della retta passante per due punti.

**Differenza in avanti** Dati i punti $x_i$, $x_{i+1}$ che possiamo riscrivere rispettivamente come $x_i$, $x_i + h$, con $h =\mid x_{i+1} - x_i \mid$, il polinomio interpolante è della forma

$$P_1(x) = f(x_i) + \frac{f(x_i + h) - f(x_i)}{h}(x - x_i), \tag{73}$$

da cui segue

$$f'(x_i) \simeq P_1'(x) = \frac{f(x_i + h) - f(x_i)}{h}. \tag{74}$$

**Differenza all'indietro** In modo simile è possibile definire quella in dietro sui punti $x_i - h$, $x_i$, in questo caso si ha

$$f'(x_i) \simeq \frac{f(x_i) - f(x_i - h)}{h}. \tag{75}$$

**Differenza centrata** Dato il punto centrale $x_i$ la differenze centrata viene costruita a partire dal valore della funzione nel punto precedente e successivo, $x_{i-1}$ e $x_{i-1}$, che diviene $x_i - h$, $x_i + h$ nel caso di campionamento uniforme

$$f'(x_i) \simeq \frac{f(x_i + h) - f(x_i - h)}{2h}. \tag{76}$$

Le formule in avanti e in dietro non sono bilanciate dispetto al punto centrale, ma sono senz'altro vantaggiose per approssimare la derivata prima in un punto ai "bordi" di un intervallo.

### 8.2.2 Derivata prima costruita su piu' punti

**Ordine 2** In questo caso si considerano tre punti $x_i$, $x_{i+1}$, $x_{i+2}$ uniformi a distanza $h$, la formula di derivazione numerica diventa

$$f'(x_i) \simeq \frac{-3f_i + 4f_{i+1} - f_{i+2}}{2h}. \tag{77}$$

**Ordine 3** Dati $x_{i-2}$, $x_{i-1}$, $x_i$, $x_{i+1}$, uniformi a distanza $h$, si ha

$$f'(x_i) \simeq \frac{f_{i-2} - 6f_{i-1} + 3f_i + 2f_{i+1}}{6h}. \tag{78}$$

### 8.2.3 Derivata seconda

Nel caso di nodi uniformi per approssimare la derivata seconda è sufficiente osservare che

$$f''(x_i) = \frac{d}{dx} f'(x)\Big|_{x=x_i} \simeq \frac{f'(x_i + h) - f'(x_i)}{h} \tag{79}$$

ricavabile dalla formula (76). Sostituendo nell'equazione (79) $f'(x_i + h)$ e $f'(x_i)$ con le loro approssimazioni in avanti nei rispettivi punti, si ottiene

$$\begin{aligned} f''(x_i) &\simeq \frac{f'(x_i + h) - f'(x_i)}{h} \\ &\simeq \frac{\frac{f(x_i+2h)-f(x_i+h)}{h} - \frac{f(x_i+h)-f(x_i)}{h}}{h} \\ &= \frac{f(x_i + 2h) - 2f(x_i + h) + f(x_i)}{h^2} \end{aligned} \tag{80}$$

Quest'ultima formula non è centrata rispetto al punto $x_i$, ma si puòo ottenere una versione centrata usando sia la formula in avanti che quella all'indietro per l'approssimazione della derivata prima. Prese le approssimazioni

$$f'(x_i + h) \simeq \frac{f(x_i + h) - f(x_i)}{h}, \qquad f'(x_i) \simeq \frac{f(x_i) - f(x_i - h)}{h}$$

e sostituendo in equazione (79), è possibile ottenere una approssimazione centrata della derivata seconda:

$$\begin{aligned} f''(x_i) &\simeq \frac{f'(x_i + h) - f'(x_i)}{h} \\ &\simeq \frac{\frac{f(x_i+h)-f(x_i)}{h} - \frac{f(x_i)-f(x_i-h)}{h}}{h} \\ &= \frac{f(x_i - h) - 2f(x_i) + f(x_i + h)}{h^2} \end{aligned} \tag{81}$$

In modo simile possiamo approssimare la derivata prima e quella seconda attraverso le differenze centrate

$$f'(x_i+h) \simeq \frac{f(x_i+2h)-f(x_i)}{2h}, \qquad f'(x_i-h) \simeq \frac{f(x_i)-f(x_i-2h)}{2h}$$

e sostituendo nuovamente in (79) si ottiene

$$\begin{aligned} f''(x_i) &\simeq \frac{f'(x_i+h)-f'(x_i-h)}{2h} \\ &\simeq \frac{\frac{f(x_i+2h)-f(x_i)}{2h} - \frac{f(x_i)-f(x_i-2h)}{2h}}{2h} \\ &= \frac{f(x_i-2h)-2f(x_i)+f(x_i+2h)}{4h^2} \end{aligned} \tag{82}$$

### 8.2.4 Derivata terza

Nel caso di nodi uniformi per approssimare la derivata terza è sufficiente osservare che

$$f'''(x_i) = \frac{d}{dx}f''(x)\bigg|_{x=x_i} \simeq \frac{f''(x_i+h)-f'(x_i)}{h} \tag{83}$$

ricavabile dalla formula (76). Sostituendo nell'equazione (83) $f''(x_i+h)$ e $f''(x_i)$ con le loro approssimazioni del tipo (81) otteniamo

$$f''(x_i) \simeq \frac{f(x_i+2h)-3f(x_i+h)+3f(x_i)-f(x_i-2h)}{h}. \tag{84}$$

### 8.2.5 Derivata quarta

Come visto per la derivata seconda, nel caso di nodi uniformi, e' possibile approssimare la derivata quarta attraverso l'applicazione ricorsiva di una formula di grado inferiore, in questo caso useremo l'equazione (81)

$$\begin{aligned} &f^{iv}(x_i) \simeq \\ &\simeq \frac{\frac{f(x_i-2h)-2f(x_i-h)+f(x_i)}{h^2} - 2\frac{f(x_i-h)-2f(x_i)+f(x_i+h)}{h^2} + \frac{f(x_i)-2f(x_i+h)-f(x_i+2h)}{h^2}}{h^2} \\ &\simeq \frac{f(x_i-2h)-4f(x_i-h)+6f(x_i)-4f(x_i+h)+f(x_i+2h)}{h^4} \end{aligned} \tag{85}$$

### 8.2.6 Derivata di grado n

Non e'difficile verificare che le formule precedenti sono esatte se $f$ è un polinomio di grado opportuno. Possiamo definire una formulazione generale per la derivata approssimata di grado $n$ del tipo

$$f^{(k)}(x_j) \simeq \sum_{i=0}^{n} w_i f(x_i), \quad k \leq n \tag{86}$$

dove gli $n+1$ coefficienti $w_i$ sono detti pesi della formula e devono essere determinati. Un modo per fissarli è fare ricorso al *metodo dei coefficienti indeterminati*, imponendo che la formula (147) sia esatta per i polinomi di grado $\leq n$.
Per la linearità delle derivata è sufficiente imporre che la (147) sia esatta per funzioni del tipo

$$f(x) = (x - x_j)^r \quad \text{per } r = 0, ..., n \tag{87}$$

e poichè vale che

$$f^{(k)}(x_j) = \begin{cases} 0 & \text{se } r \neq k \\ k! & \text{se } r = k \end{cases} \tag{88}$$

segue che gli $w_i$, $i = 0, ..., n$ devono soddisfare il sistema lineare

$$\sum_{i=0}^{n} w_i (x_i - x_j)^r = \begin{cases} 0 & \text{se } r \neq k \\ k! & \text{se } r = k \end{cases} \qquad r = 0, ..., n \tag{89}$$

Essendo la matrice del sistema (89) una matrice di Vandermonde ed essendo i punti $x_i$ distinti, il sistema è non singolare e la sua risoluzione consente di ricavare i $w_i$ $i = 0, ..., n$. Dalla prima equazione del sistema (89), essendo $k \geq 1$, risulta $\sum_{i=0}^{n} w_i = 0$, segue che la somma dei coefficienti di una formula di derivazione approssimata è nulla.

## 8.3 Esempio

Vogliamo approssimare la derivata prima ($k = 1$) di una funzione nel punto $x_0$ con $n = 1$ e cioè con una formula del tipo

$$\sum_{i=0}^{1} w_i f(x_i) = w_0 f(x_0) + w_1 f(x_1)\,. \tag{90}$$

Applicando il metodo dei coefficienti indeterminati poichè

$$D(x - x_j)^r|_{x=x_j} = \begin{cases} 0 & \text{se } r \neq 1 \\ 1! & \text{se } r = 1 \end{cases} \tag{91}$$

dovremo risolvere quindi il sistema lineare associato alle equazioni

$$r = 0 \qquad \sum_{i=0}^{1} w_i = 0 \tag{92}$$

$$r = 1 \qquad \sum_{i=0}^{1} w_i (x_i - x_0) = 1 \tag{93}$$

ovvero

$$\begin{pmatrix} 1 & 1 \\ 0 & (x_1 - x_0) \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{94}$$

le cui soluzioni sono

$$w_1 = \frac{1}{x_1 - x_0} \qquad w_0 + w_1 = 0 \qquad w_0 = -\frac{1}{x_1 - x_0}\,. \tag{95}$$

Sostituendo i valori ottenuti nell'equazione (90) si ottiene la formula di quadratura numerica:

$$f'(x_0) \simeq -\frac{1}{x_1 - x_0} f(x_0) + \frac{1}{x_1 - x_0} f(x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}\,. \tag{96}$$

# 9 Formule di Quadratura

Sia $f$ una funzione definita su un intervallo $[a, b]$, il problema dell'integrazione numerica e' quello di calcolare

$$I(f) = \int_a^b f(x)\,\mathrm{d}x, \tag{97}$$

utilizzando soltanto il valore della funzione in una serie di punti chiamati *nodi*. Il calcolo esatto dell'integrale richiede la conoscenza della funzione primitiva $F(x)$, per la quale, in virtu' del teorema fondamentale del calcolo integrale risulta:

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a). \tag{98}$$

L'utilizzo di una *formula di quadratura* al posto dell'integrale e' utile quando non si sa risolvere l'integrale o quando il calcolo e' troppo costoso oppure quando non si conosce la funzione primitiva o nel caso in cui la funzione integranda sia effettivamente nota solo per punti.
L'integrazione numerica si basa sull'approssimazione del valore dell'integrale mediante la determinazione di una funzione approssimante scelta in una specifica classe. Nel nostro caso studieremo l'utilizzo della classe dei polinomi per la loro semplicita', ma nulla impedirebbe di utilizzare un'altra classe di funzioni.

Ovviamente esiste anche la possibilita' di risolvere il problema nel caso di integrali indefiniti, illimitati, multidimensionali, con singolarita' , etc. ma il loro studio esula dagli scopi di queste pagine.

**Definizione 3.** Le formule di *quadratura* o di *integrazione numerica* che consideriamo in queste pagine sono del tipo

$$S_{n+1}(f) := \sum_{i=0}^{n} w_i f(x_i) \tag{99}$$

dove $x_i$ con $i = 0, \ldots, n$ sono i nodi della formula, ovvero i punti delle ascisse dove la funzione e' conosciuta, $w_i$ con $i = 0, \ldots, n$ sono i pesi o coefficienti della formula. In genere si preferisce utilizzare $f_i$ al posto di $f(x_i)$ per semplificare la notazione. E' importante osservare che ci sono anche formule di tipo diverso ma non le vediamo in questo contesto. Per ogni formula di quadratura risulta

$$\int_a^b f(x)\,\mathrm{d}x = \underbrace{\sum_{i=0}^{n} w_i f(x_i)}_{S_{n+1}(f)} + R_{n+1}(f)\,, \tag{100}$$

e il termine $S_{n+1}(f) = \sum_{i=0}^{n} w_i f(x_i)$ e' detto *somma* o *parte approssimante* e $R_{n+1}(f)$ e' detto *resto* o *errore di troncamento*.

In generale l'operazione di integrazione non si compie da sola, ma fa parte di un problema piu' generale. Se i dati sono sporchi, cioe' affetti da errore, ogni operazione che si compie su di essi puo' propagarne l'errore (aumentandolo o diminuendolo). É quindi importante valutare in modo quantitativo l'influenza degli errori e possibilmente farli "convergere" a zero. Se i dati sono affetti da errore possiamo scrivere :

$$\tilde{f}_i = f(x_i) + \varepsilon \tag{101}$$

In questo caso la formula di quadratura diventa:

$$\sum_{i=0}^{n} w_i \tilde{f}_i = \sum_{i=0}^{n} w_i (f_i + \varepsilon_i) \tag{102}$$

E quindi risulta:

$$I(f) = S_{n+1}(f) + \underbrace{\sum_{i=0}^{n} w_i \varepsilon_i}_{R^*_{n+1}(f)} + R_{n+1}(f) \tag{103}$$

dove $R^*_{n+1}$ e' l'errore dovuto all' errore sui dati e non al metodo. E', ovviamente, nostro interesse studiare formule di quadratura tali per cui $R_{n+1}(f)$ converga a zero ma, anche se cio' si verifica, la presenza di $R^*_{n+1}(f)$ detto *errore di propagazione* puo' rappresentare un problema, soprattutto se si utilizza un computer dove la rappresentazione *floating point* genera errori sui dati.

Grazie al teorema di approssimazione di Weierstrass, sappiamo che ogni funzione reale continua definita in un intervallo chiuso e limitato puo' essere approssimata a piacere con un polinomio di grado opportuno. Partendo quindi dal presupposto che un polinomio e' adatto ad approssimare le funzioni, viene naturale lo studio del comportamento delle formule di quadratura rispetto alla classe dei polinomi.

**Definizione 9.1.** *Una formula di quadratura* $S_{n+1}(f) := \sum_{i=0}^{n} w_i f(x_i)$ *ha* ***grado di precisione*** $\nu$ *se risulta che*

$$R_{n+1}(x^r) = 0 \text{ se } r = 0, \ldots, \nu, \quad R_{n+1}(x^{\nu+1}) \neq 0 \tag{104}$$

Il grado di precisione e' un criterio arbitrario per confrontare tra loro formule di precisione. Ad esempio se una formula di quadratura ha grado 2 allora questa e' in grado di calcolare l'integrale esatto in un intervallo chiuso e limitato $[a, b]$ di una parabola o di una retta, ma non e' in grado di calcolarlo in modo esatto per un polinomio di terzo grado (ovvero commette un errore). Il grado di precisione dipende da molte cose, in particolare dalla scelta dei nodi e dal loro numero.

**Osservazione 9.1.** *Il grado di precisione di una formula dipende oltre che dal numero, dalla distribuzione dei nodi.*

Una formula di quadratura della forma indicata finora (costruita quindi su $n+1$ nodi) ha un limite sul grado di precisione raggiungibile che risulta $\nu \leq 2n+1$. Tuttavia, e' possibile ottenere il massimo grado di precisione soltanto con formule abbastanza complicate (dette Gaussiane ma non studiate in queste pagine). Questo risultato e'presentato nel seguente Teorema.

**Teorema 12.** *Una formula di quadratura del tipo*

$$S_{n+1}(f) = \sum_{i=0}^{n} w_i f_i \tag{105}$$

*ha grado di precisione* $\nu \leq 2n + 1$.

*Dimostrazione.* Basta far vedere che esiste un polinomio $\pi$ di grado $2n + 2$ su cui la formula non e' precisa ovvero che $R_{n+1}(\pi) \neq 0$ con $\pi \in \mathbb{P}_{2n+2}$. Consideriamo i nodi della formula, $x_0, \cdots, x_n$ e consideriamo

$$\pi(x) = w_{n+1}^2(x) = (x - x_0)^2 (x - x_1)^2 \cdots (x - x_n)^2 , \tag{106}$$

che e' un polinomio di grado $2n + 2$. La formula di quadratura applicata a $\pi$ produce

$$S_{n+1}(\pi) = \sum_{i=0}^{n} w_i \pi(x_i) = 0 \tag{107}$$

in quanto la funzione $\pi$ assume valore 0 in ogni nodo per costruzione. Si tratta quindi di un polinomio che rende il risultato dell'integrale sempre uguale a 0 per qualunque scelta dei nodi e dei pesi, e quindi, per qualunque formula di quadratura del tipo in questione. Tuttavia,

$$R_{n+1}(\pi) = \int_a^b \pi(x)\, \mathrm{d}x > 0 \neq S_{n+1}(\pi) \tag{108}$$

in quanto $\pi$ e' una funzione sempre positiva eccetto che nei nodi dove si annulla. Conlcudendo: esiste un polinomio di grado $2n+2$ sul quale la formula di quadratura non e' esatta e pertanto il suo grado di precisione sara' minore o uguale ad $2n+1$. Risulta quindi dimostrato il teorema. $\square$

Proseguiamo studiando un importante risultato di convergenza per formule di quadratura del tipo $S_{n+1}(f)$ al crescere del numero dei nodi.

**Teorema 13.** *Sia $f$ una funzione continua e limitata definita in $[a,b]$, $S_{n+1}(f)$ una formula di quadratura con grado di precisione almeno $n$ e supponiamo che i suoi pesi siano equilimitati, ovvero tali che*

$$\sum_{i=0}^{n} |w_i| < C \qquad \textit{per } C \in \mathbb{R}_+. \tag{109}$$

*Allora*

$$\lim_{n\to\infty} S_{n+1}(f) = \int_a^b f(x)\,\mathrm{d}x\,. \tag{110}$$

*Dimostrazione.* Notiamo che richiedere (110) e' la stessa cosa che richiedere che il resto vada a zero cioe'

$$\lim_{n\to\infty} R_{n+1}(f) = 0\,. \tag{111}$$

Consideriamo la formula di quadratura

$$S_{n+1}(f) = \sum_{i=0}^{n} w_i f(x_i)\,, \tag{112}$$

e prendiamo $p_n \in \mathbb{P}_n$. Poiche' la formula di quadratura ha grado di precisione almeno $n$ risulta

$$S_{n+1}(p_n) = \int_a^b p_n(x)\,\mathrm{d}x \tag{113}$$

Considerando questo, il resto sara' uguale a

$$\begin{aligned} R_n(f) &= \int_a^b f(x) - S_{n+1}(f)dx = \\ &= \Big[\int_a^b f(x) - S_{n+1}(f)\Big]dx + S_{n+1}(p_n) - \int_a^b p_n(x)\,\mathrm{d}x \end{aligned} \tag{114}$$

dove abbiamo aggiunto e sottratto la formula di quadratura applicata al polinomio. Passando al valore assoluto possiamo fare una maggiorazione

$$\begin{aligned} 0 \leq |R_{n+1}(f)| &\leq \int_a^b |f(x) - p_n(x)|\,\mathrm{d}x + \Big|\sum_{i=0}^{n} w_i(p_n(x_i) - f(x_i))\Big| \\ &\leq \int_a^b |f(x) - p_n(x)|\,\mathrm{d}x + \sum_{i=0}^{n} |w_i||p_n(x_i) - f(x_i)| \\ &\leq ||f - p_n||(b-a) + C||p_n - f|| \qquad (115) \\ &= (C + (b-a))||p_n - f|| \qquad (116) \end{aligned}$$

dove $||p_n - f|| = \max_{x\in[a,b]} |p_n(x) - f(x)|$. Ma se $f$ e' continua in $[a, b]$, per il teorema di Weierstrass esiste una successione di polinomi tali che $\tilde{p}_n \to f$ per $n \to \infty$, da cui segue che $||\tilde{p}_n - f|| \to 0$. Se prendiamo quindi come $p_n$ l' ennesimo polinomio delle successione convergente ad $f$ e cioe' $p_n = \tilde{p}_n$, il lato a destra della maggiorazione converge a zero e per il teorema dei due carabinieri segue che

$$lim_{n\to\infty}|R_{n+1}(f)| = 0 \qquad (117)$$

che dimostra la tesi del teorema. □

*Osservazione* 3. E' importante notare che il penultimo passaggio e' possibile soltanto perche abbiamo assunto che i pesi siano equilimitati e cioe' che la costante $C$ sia indipendente da $n$. In caso contrario non e' possibile concludere che per $n \to \infty$ il lato a destra della maggiorazione converge a zero. Infatti la successione di pesi $C_n$ (se non sono equilimitati) potrebbe far divergere il risultato, senza portare alla conclusione di convergenza a zero del resto.

**Proposizione 3.** *Se $S_{n+1}(f)$ ha grado di precisione almeno zero e $w_i > 0$ allora i suoi pesi sono equilimitati ovvero esiste $C$ tale che*

$$\sum_{i=0}^{n} |w_i| < C. \qquad (118)$$

*Dimostrazione.* Se $S_{n+1}(f)$ ha grado di precisione zero e' esatta per tutte le costanti. In tal caso

$$S_{n+1}(1) = \int_a^b 1\,\mathrm{d}x = b - a = \sum_{i=0}^{n} w_i = \sum_{i=0}^{n} |w_i| \qquad (119)$$

ovvero per il polinomio costante $p = 1$ la formula e' proprio $\sum_{i=0}^{n} |w_i|$. Ne consegue l'esistenza di una costante $C = b - a + 1$ tale che

$$\sum_{i=0}^{n} |w_i| = b - a \leq b - a + 1\,. \qquad (120)$$

□

Continuiamo osservando che l' equilimitatezza dei pesi garantisce anche la *stabilita'* della formula di quadratura.

**Proposizione 4.** *Se* $\sum_{i=0}^{n} |w_i| < C$ *allora anche gli errori sono limitati, ovvero*

$$\Big|\sum_{i=0}^{n} w_i\varepsilon_i\Big| \leq \varepsilon C \qquad \textit{dove } \varepsilon = \max_{i=0,\dots,n} |\varepsilon_i| \tag{121}$$

*Dimostrazione.* Partendo dalla (102) si ha

$$\Big|\sum_{i=0}^{n} w_i\varepsilon_i\Big| \leq \sum_{i=0}^{n} |w_i||\varepsilon_i| \leq C\varepsilon \tag{122}$$

□

*Osservazione* 4. In virtu' del precedente risultato, possiamo concludere che se i pesi sono equilimitati l 'errore si mantiene proporzionale all'errore dei dati e non "esplode.

## 9.1 Formule di quadratura interpolatorie

In virtu' delle considerazioni sui polinomi interpolatori e del fatto che si cercano formule con grado di precisione almeno $n$, si considerano generalmente *formule di quadratura interpolatorie* che sono cioe' ottenute dall'intergrazione del polinomio interpolante. Infatti, assegnati i dati $(x_i, f(x_i)),\ i = 0, \cdots, n$ si costruisce $p_n \in \mathbb{P}_n$ il corriposndente polinomio interpolante e la formula di quadratura $S_{n+1}(f)$ si definisce come segue:

$$S_{n+1}(f) = \int_a^b p_n(x)\,\mathrm{d}x = \int_a^b \sum_{i=0}^{n} l_i(x) f(x_i)\,\mathrm{d}x = \sum_{i=0}^{n} \underbrace{\Big(\int_a^b l_i(x)\,\mathrm{d}x\Big)}_{w_i} f(x_i) \tag{123}$$

dove $l_i(x)$ sono le basi di Lagrange:

$$l_i(x) = \frac{w_{n+1}(x)}{w'_{n+1}(x_i)(x - x_i)}, \quad i = 0, \cdots n. \tag{124}$$

I pesi $w_i$ di una formula di quadratura interpolatoria assumono quindi la forma:

$$w_i = \int_a^b l_i(x)\,\mathrm{d}x = \frac{1}{w'_{n+1}(x_i)} \int_a^b \frac{w_{n+1}(x)}{x - x_i}\,\mathrm{d}x = \int_a^b \frac{\prod_{\substack{k=0 \\ k\neq i}}^{n} (x - x_k)}{\prod_{\substack{k=0 \\ k\neq i}}^{n} (x_i - x_k)}\,\mathrm{d}x \tag{125}$$

Utilizzando l' errore di interpolazione e'facile vedere che il resto della formula di quadratura risulta:

$$R_{n+1}(f) = \int_a^b E_n(f)\,\mathrm{d}x = \int_a^b w_{n+1}(x) f[x, x_0, \ldots, x_n]\,\mathrm{d}x$$
$$= \int_a^b \frac{w_{n+1}(x)}{(n+1)!} f^{n+1}(\xi(x))\,\mathrm{d}x. \tag{126}$$

Appare quindi evidente che, nel caso in cui $f$ sia un polinomio di grado al piu' $n$, il valore della formula sia esatto (perche' il polinomio interpolante di grado $n$ di un polinomio di grado al piu' $n$ e'il polinomio stesso!) e quindi l' errore sia esattamente uguale a 0. Notiamo che i pesi sono proprio gli integrali delle basi di Lagrange e pertanto niente garantisce che siano positivi. Ricordiamo che le basi di Lagrange si annullano in tutti i nodi ed in genere possono assumere anche valori negativi rendendo possibile l'esistenza di pesi negativi.
Il prossimo Teorema studia il grado di precisione delle formule di quadratura.

**Teorema 14.** *Le formule di quadratura interpolatorie $S_{n+1}(f)$ costruite su $n+1$ nodi, hanno grado di precisione $\nu$ che soddisfa*

$$n \leq \nu \leq 2n+1.$$

*Dimostrazione.* Se $f \in \mathbb{P}_n$ (e quindi e' derivabile infinite volte) allora il resto della formula e'

$$R_{n+1}(f) = \int_a^b \frac{f^{n+1}(\xi(x)) w_{n+1}(x)}{(n+1)!}\,\mathrm{d}x = 0. \tag{127}$$

La parte $\nu \leq 2n+1$ e' dimostrata nel Teorema 12. □

Vale anche un altro interessante risultato "inverso".

**Teorema 15.** *Ogni formula di quadratura su $n+1$ nodi, $S_{n+1}(f)$, e grado di precisione almeno $n$ e' interpolatoria.*

*Dimostrazione.* Consideriamo la formula di quadratura

$$S_{n+1}(f) = \sum_{i=0}^{n} w_i f_i. \tag{128}$$

Se scegliamo $l_j \in \mathbb{P}_n$ elemento della base di Lagrange come funzione da integrare, la formula di quadratura risulta esatta e percio'

$$S_{n+1}(l_j) = \int_a^b l_j(x)\,\mathrm{d}x\,. \tag{129}$$

Osservando che $l_j(x_i) = \delta_{ij}$, dalla applicazione della formula di quadratura si ha anche che

$$S_{n+1}(l_j) = \sum_{i=0}^{n} w_i l_j(x_i) = \sum_{i=0}^{n} w_i \delta_{ij} = w_j \tag{130}$$

da cui consegue

$$w_j = \int_a^b l_j(x)\,\mathrm{d}x \tag{131}$$

e cioe' che la formula e' interpolatoria. □

Dalla precedente analisi consegue che l'unica cosa su cui possiamo agire per migliorare il grado di precisione di una formula di quadratura interpolatoria e' il metodo di selezione dei nodi.

## 9.2 Metodo dei coefficienti indeterminati

Il metodo dei coefficienti indeterminati e' un metodo generale che permette di trovare i pesi da utilizzare per realizzare una formula di quadratura con nodi assegnati. Fissati $x_0, \ldots, x_n$ nodi in qualsiasi modo si preferisce, si vuole ottenere una formula di quadratura interpolatoria della forma

$$S_{n+1}(f) = \sum_{i=0}^{n} w_i f_i. \tag{132}$$

L'idea alla base del metodo e' di determinare i pesi imponendo che la formula abbia grado di precisione $n$.
Si procede per gradi successivi: imporre che la formula abbia grado di precisione 0 fa si che, scegliendo $f = 1$, debba risultare

$$\sum_{i=0}^{n} w_i f_i = b - a \Rightarrow \sum_{i=0}^{n} w_i = b - a\,. \tag{133}$$

Imporre che la formula abbia grado di precisione 1 fa si che, per $f = x$, debba valere

$$\sum_{i=0}^{n} w_i f_i = \int_a^b x\,\mathrm{d}x \Rightarrow \sum_{i=0}^{n} w_i x_i = \int_a^b x\,\mathrm{d}x = \frac{x^2}{2}\bigg|_a^b = \frac{b^2 - a^2}{2}. \tag{134}$$

Imporre che la formula abbia grado di precisione 2 fa si che, per $f = x^2$, debba valere

$$\sum_{i=0}^{n} w_i f_i = \int_a^b x^2\,\mathrm{d}x \Rightarrow \sum_{i=0}^{n} w_i x_i^2 = \int_a^b x^2\,\mathrm{d}x = \frac{x^3}{3}\bigg|_a^b = \frac{b^3 - a^3}{3}\,. \tag{135}$$

Procedendo in questo modo fino al grado di precisione $n$ otteniamo

$$\sum_{i=0}^{n} w_i f_i = \int_a^b x^n \, \mathrm{d}x \Rightarrow \sum_{i=0}^{n} w_i x_i^n = \int_a^b x^n \, \mathrm{d}x = \frac{x^{n+1}}{n+1}\Big|_a^b = \frac{b^{n+1} - a^{n+1}}{n+1}. \tag{136}$$

In sostanza, per determinare i pesi della formula di quadratura, si deve risolvere un sistema lineare della forma

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_n \\ x_0^2 & x_1^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_0^n & x_1^n & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} b-a \\ \frac{b^2-a^2}{2} \\ \frac{b^3-a^3}{3} \\ \vdots \\ \frac{b^{n+1}-a^{n+1}}{n+1} \end{bmatrix}, \tag{137}$$

la cui matrice e' la trasposta della matrice di Vandermonde, $V(x_0, \cdots, x_n)$, che e' malcondizionata. Se i nodi sono distinti esiste una ed una sola soluzione al sistema lineare poiche' e' noto che, in tal caso, la matrice di Vandermonde ha rango massimo Purtroppo si tratta di un sistema mal condizionato con conseguenti problemi risolutivi. Pertanto questo non e' un metodo consigliabile in particolare se $n$ e' grande.

# 10 Formule di Newton-Cotes

Sono formule interpolatorie con nodi uniformi, ovvero, nell' intervallo $[a, b]$ con nodi $[x_0, \cdots, x_n]$ del tipo $x_i = a + i * h,\ i = 0, \cdots, n$, con $h = \frac{b-a}{n}$. si dividono in *formule chiuse* che utilizzano tutti i nodi e *formule aperte* dove non si considerano i nodi all'estremita' dell'intervallo e cioe' si escludono dai nodi che utilizza la formua $x_0 = a$ e $x_n = b$.

Il grado di precisione di queste formule cambia a seconda del numero di nodi utilizzati e del tipo di formula utilizzata(chiusa o aperta) e segue lo schema mostrato in tabella:

| tipo | n+1 | g.d.p. |
|---|---|---|
| chiusa | pari | n |
| chiusa | dispari | n+1 |
| aperta | pari | n-2 |
| aperta | dispari | n-1 |

## 10.1 Formule chiuse

In questo paragrafo considereremo le formule chiuse e vedremo quanto valgono i pesi della formula all'aumentare dei nodi. Le piu' utilizzate considerano al massimo $n = 7$ perche' per $n > 7$ i pesi iniziano a diventare negativi e

quindi non e' piu' assicurata la equi-limitatezza dei coefficienti. In queste pagine saranno considerate solo le formule per $n = 1, 2, 3, 4$.

### 10.1.1 Formula di Newton-Cotes chiusa costruita con 2 nodi

Per $n + 1 = 2$ la formula di Newton-Cotes prende il nome di "formula del trapezio" (o dei trapezi). I nodi utilizzati sono gli unici presenti che sono gli estremi dell'intervallo $[a, b]$. Essendo $n + 1$ pari il grado di precisione e' $\nu = n$, cioe' $\nu = 1$.

$$\int_a^b f(x)dx \cong \frac{h}{2}(f_0 + f_1), \quad h = \frac{b-a}{1}. \tag{138}$$

E' possibile dimostrare che: $R_2^C(f) = \frac{h^3}{12} f^{(2)}(\zeta)$ dove $\zeta \in [a, b]$.

### 10.1.2 Formula di Newton-Cotes chiusa costruita con 3 nodi

Per $n + 1 = 3$ la formula di Newton-Cotes prende il nome di "formula di Simpson-Cavalieri". Essendo $n+1$ dispari, il grado di precisione e' $\nu = n+1$ e quindi $\nu = 3$

$$\int_a^b f(x)dx \cong \frac{h}{3}(f_0 + 4f_1 + f_2), \quad h = \frac{b-a}{2}. \tag{139}$$

E' possibile dimostrare che: $R_3^C(f) = -\frac{h^5}{90} f^{(4)}(\zeta)$ dove $\zeta \in [a, b]$.

### 10.1.3 Formula di Newton-Cotes chiusa costruita con 4 nodi

Per $n + 1 = 4$ la formula di Newton-Cotes prende il nome di "formula dei 3 ottavi". Essendo $n + 1$ e' pari, il grado di precisione e' $\nu = n = 3$.

$$\int_a^b f(x)dx \cong \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3), \quad h = \frac{b-a}{3}, \tag{140}$$

E' possibile dimostrare che: $R_4^C(f) = -\frac{3h^5}{80} f^{(4)}(\zeta)$ dove $\zeta \in [a, b]$.

### 10.1.4 Formula di Newton-Cotes chiusa costruita con 5 nodi

Per $n + 1 = 5$ la formula di Newton-Cotes prende il nome di "formula di Milne-Boole". Essendo $n+1$ dispari, il grado di precisione e' $\nu = n+1 = 5$.

$$\int_a^b f(x)dx \cong \frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4), \quad h = \frac{b-a}{4}, \tag{141}$$

E' possibile dimostrare che: $R_5^C(f) = -\frac{8h^7}{945} f^{(6)}(\zeta)$ dove $\zeta \in [a, b]$.

## 10.2 Formule aperte

In questo paragrafo ci occuperemo delle formule aperte. In questo caso il numero totale di nodi che utilizza la formula e' $n-1$ ed e' riferito al numero di nodi "interni", cioe' i nodi che sono utilizzi per l'interpolazione che sta dietro al metodo. Tuttavia, per il calcolo del grado di precisione dalla tabella, vanno considerati anche i nodi estremi dell'intervallo $[a, b]$, quindi in totale i nodi sono $n+1$.

## 10.3 Formula di Newton-Cotes aperta costruita con 1 nodo

Per $n-1=1$ la formula di Newton-Cotes prende il nome di "formula del punto medio". Il grado di precisione e' $\nu = 1$.

$$\int_a^b f(x)dx \cong 2hf_1, \quad h = \frac{b-a}{2}. \tag{142}$$

E' possibile dimostrare che: $R_1^A(f) = \frac{h^3}{3} f^{(2)}(\zeta)$, dove $\zeta \in [a, b]$.

### 10.3.1 Formula di Newton-Cotes aperta costruita con 2 nodi

In questo caso consideriamo 2 punti interni ($n-1=2$), quindi il grado di precisione e' $\nu = 1$.

$$\int_a^b f(x)dx \cong \frac{3h}{2}(f_1 + f_2), \quad h = \frac{b-a}{3}. \tag{143}$$

E' possibile dimostrare che: $R_2^A(f) = \frac{3h^3}{4} f^{(2)}(\zeta)$, dove $\zeta \in [a, b]$.

### 10.3.2 Formula di Newton-Cotes aperta costruita con 3 nodi

In questo caso considero 3 punti interni ($n-1=3$), quindi il grado di precisione e' $\nu = 3$.

$$\int_a^b f(x)dx \cong \frac{4h}{3}(2f_1 - f_2 + 2f_3), \quad h = \frac{b-a}{4}. \tag{144}$$

E' possibile dimostrare che: $R_3^A(f) = \frac{14h^5}{45} f^{(4)}(\zeta)$, dove $\zeta \in [a, b]$.

Dall' analisi delle precedenti formule di quadratura di Newton Cotes, possiamo concludere che le formule di tipo chiuso, usando piu' informazioni, sono piu' precise (almeno rispetto ai polinomi). Tuttavia ci sono situazioni in cui e' importante escludere i nodi esterni (si pensi al caso in cui in un nodo e' presente una singolarita') e in questo caso risulta utile fare uso di formule aperte anche se con grado di precisione piu' basso.

## 10.4 Formule di Newton-Cotes generalizzate o composite

Per evitare l' uso di formule di grado elevato (ricordiamo che per $n > 7$ i pesi della formula di quadratura non sono piu' tutti positivi), si utilizzano le cosi' dette *Formule di Newton-Cotes generalizzate*. Sono basate sull' idea di dividere l' intervallo in sottointervalli in ciascuno dei quali si utilizzano formule di quadratura di grado basso, ad esempio la formule dei trapezi o di Simpson-Cavalieri.

Si parte dividendo l'intervallo [a,b] in $m$ sottointervalli uniformi gli estremi di ognuno dei quali si calcola quindi come:

$$X_j = a + j\frac{b-a}{m}, j = 0, \cdots, m\,. \tag{145}$$

In ogni sottointervallo si usano $n+1$ nodi uniformi di distanza $h = \frac{b-a}{mn}$. Ogni nodo dell'intervallo $[a, b]$ ha quindi la forma

$$x_r = a + rh, r = 0, \cdots, mn\,. \tag{146}$$

Per la linearita' dell' integrale rispetto agli estremi di integrazione, l'integrale in $[a, b]$ sara' dato dalla somma degli integrali calcolati nei sottointervalli:

$$I(f) = \sum_{j=0}^{m-1} I_j(f) \tag{147}$$

e corrispondentemente, la formula di quadratura sara' la somma delle formule di quadratura (con meno nodi) nei sottointervalli.

Nel caso si vogliano utilizzare 2 nodi per sottointervallo la (147) diventa:

$$I(f) = \frac{b-a}{2m}[f_0 + 2\sum_{j=1}^{m-1} f_j + f_m] - \frac{b-a}{12}h^2 f^{(2)}(\tau) \tag{148}$$

la cui corrispondente formula di quadrata prende il nome di *formula composita dei trapezi*. Nel caso si vogliano utilizzare 3 nodi per sottointervallo si ottiene

$$I(f) = \frac{b-a}{6m}[f_0 + 4\sum_{j=0}^{m-1} f_{2j+1} + 2\sum_{j=1}^{m-1} f_{2j} + f_{2m}] - \frac{b-a}{180}h^4 f^{(4)}(\tau)\,, \tag{149}$$

la cui corrispondente formula di quadrata prende il nome di *formula composita di Simpson*.

# 11 Criterio di Runge e formule adattative

Con il criterio di Runge, che andremo a descrivere, si stima l'errore che si commette utilizzando le formule di quadrature composite. Questa stima dell'errore verra' poi utilizzata sia per costruire nuove formule di quadratura (generalmente piu'esatte) che all'interno delle cosi' dette *formule adattative* che sono le piu' utilizzate nella pratica.

## 11.1 Stima dell'errore nella formula composita dei trapezi

Nel caso di formule di quadratura composite dei trapezi, dal paragrafo precedente sappiamo che per $h = \frac{b-a}{n}$

$$I(f) = S_{n+1}(f) + R_{n+1}(f) = S_{n+1}(f) - \frac{b-a}{12}h^2 f^{(2)}(\tau). \tag{150}$$

Se aggiungiamo un nodo in ogni intervallo (raddoppiando praticamente i nodi) otteniamo

$$I(f) = S_{2n+1}(f) + R_{2n+1}(f) = S_{2n+1}(f) - \frac{b-a}{12}\left(\frac{h}{2}\right)^2 f^{(2)}(\sigma). \tag{151}$$

Quindi, ipotizzando che $f^{(2)}(\sigma) \cong f^{(2)}(\tau)$, otteniamo $R_{n+1}(f) \cong 4R_{2n+1}(f)$ e quindi che

$$O = I(f) - I(f) = S_{n+1}(f) - S_{2n+1}(f) + 3R_{2n+1}(f) \tag{152}$$

da cui segue una stima di $R_{2n+1}(f)$

$$R_{2n+1}(f) \cong \frac{S_{2n+1}(f) - S_{n+1}(f)}{3}. \tag{153}$$

Usando tale stima dell' errore possiamo ottenere una nuova formula di quadratura per il calcolo approssimato dell'integrale data da:

$$I(f) \cong S_{2n+1}(f) + R_{2n+1}(f) = \frac{1}{3}[4S_{2n+1}(f) - S_{n+1}(f)]. \tag{154}$$

Il precedente metodo, che somma alla formula $S_{2n+1}(f)$ una stima dell'errore della stessa (se la stima fosse esatta si otterrebbe quindi l' integrale esatto) prende il nome di *estrapolazione di Richardson.*

## 11.2 Stima dell'errore utilizzando la formula di Simpson

Lo stesso procedimento si puo' utilizzare per stimare l'errore nel caso di formule di quadratura composite di Simpson-Cavalieri. Per $h = \frac{b-a}{n}$,

$$I(f) = S_{n+1}(f) + R_n(f) = S_{n+1}(f) - \frac{b-a}{180}h^4 f^{(4)}(\tau). \tag{155}$$

Se "raddoppiamo" i nodi otteniamo:

$$I(f) = S_{2n+1}(f) + R_{2n+1}(f) = S_{2n+1}(f) - \frac{b-a}{180}\left(\frac{h}{2}\right)^4 f^{(4)}(\sigma). \quad (156)$$

Quindi, come nel caso dei trapezi, ipotizzando che $f^{(4)}(\sigma) \cong f^{(4)}(\tau)$, risulta $R_{n+1}(f) \cong 16R_{2n+1}(f)$ e quindi

$$O = I(f) - I(f) = S_{n+1}(f) - S_{2n+1}(f) + 15R_{2n+1}(f), \quad (157)$$

da cui segue che

$$R_{2n+1}(f) \cong \frac{S_{2n+1}(f) - S_{n+1}(f)}{15}. \quad (158)$$

Usando poi il metodo di estrapolazione di Richardson, possiamo ottenere una nuova formula di quadratura per il calcolo approssimato dell'integrale data da:

$$I(f) \cong S_{2n+1}(f) + R_{2n+1}(f) = \frac{1}{15}[16S_{2n+1}(f) - S_{n+1}(f)]. \quad (159)$$

## 11.3 Formule di quadrature adattative

Supponiamo di voler integrare una funzione $f$, nell'intervallo $[a, b]$ e che il suo comportamento vari molto in una parte dell'intervallo e sia piu' regolare nell'altra. In questo caso, per ottenere una buona stima dell' integrale con una formula composita siamo costretti ad utilizzare molti punti in modo da "catturare" il comportamento variabile della funzione. Tuttavia, e' ragionevole pensare che potremmo ottenere lo stesso risultato utilizzando piu' nodi nella parte irregolare, in modo da catturare meglio le variazioni della $f$, e pochi nodi altrove. Cio' e' possibile utilizzando le *formule di quadratura adattative*. L'idea delle formule adattative e' quella di utilizzare una "coppia" di formule composite ($S_{n+1}(f)$, $S_{2n+1}(f)$) stimare l'errore nell'intervallo $[a, b]$ con il metodo di Runge descritto in precedenza, e qualora non risulti accettabile, aumentare si il numero dei nodi ma dividendo l'intervallo in 2 sottointervalli e applicando la stessa coppia di formule di quadratura composita in entrambi i sottointervalli. In tal modo il numero dei nodi che la formula utilizza risultera' "addensato" solo nei sottointervalli di $[a, b]$ in cui la stima dell' errore non soddisfa il controllo.

Quindi, riassumendo: il metodo fissa una coppia di formule di quadratura composite, $S_{n+1}(f)$, $S_{2n+1}(f)$ ed inizia applicandole all'intervallo $[a, b]$. Stabilita una tolleranza $\tau$, usa le due formule per stimare l' errore verificando

$$|S_{n+1}(f) - S_{2n+1}(f)| < \tau. \quad (160)$$

Se la condizione e' soddisfatta il metodo e' concluso e, in $[a, b]$, utilizza la formula di quadratura $S_{2n+1}(f)$ (che si presume essere piu' esatta di $S_{n+1}(f)$).

Altrimenti il metodo procede dividendo l'intervallo in due sottointervalli uguali in ciascuno dei quali utilizza la stessa coppia di formule di quadratura composite, $S_{n+1}(f)$, $S_{2n+1}(f)$. Utilizzando tali formule si verifica se la condizione (160) e' soddisfatta in entrambi i sottointervalli. Se lo e' come approssimazione dell' integrale si sceglie $S_{2n+1}(f|_{[a,\frac{a+b}{2}]}) + S_{2n+1}(f|_{[\frac{a+b}{2},b]})$. Altrimenti, negli intervalli in cui il criterio non e' soddisfatto (quindi in uno dei due sottointervalli o in entrambi), il metodo ripete gli stessi passi: utilizza la stessa coppia di formule di quadratura composite, $S_{n+1}(f)$, $S_{2n+1}(f)$ e verifica se la condizione (160) e' soddisfatta e se non lo e' ridivide l' intervallo in due sottointervalli fino a quando la condizione (160) non viene soddisfatta in ogni intervallo. Alla fine il valore approssimato dell'integrale su $[a, b]$ sara' la somma del risultato della formula di quadratura $S_{2n+1}(f)$ applicata nei sottointervalli (di misura diversa). In conclusione quindi, il numero totale di punti utilizzato dalla formula non e' noto a priori e dipendera' dalle caratteristiche della funzione, dalla scelta della coppia iniziale di formule composite e dalla tolleranza $\tau$.

Le formule di quadrature adattative sono le formule piu' utilizzate nella pratica.