

# APPROSSIMAZIONE DI FUNZIONI E DI DATI

Letizia SCUDERI

Dipartimento di Scienze Matematiche, Politecnico di Torino

[letizia.scuderi@polito.it](mailto:letizia.scuderi@polito.it)

A.A. 2022/2023

L'approssimazione riveste un ruolo fondamentale nel Calcolo Numerico.

**Approssimare una funzione**  $f$ , nota in forma analitica, significa sostituirla con una funzione  $\tilde{f}$  che le sia “vicina” in qualche senso e abbia una forma più semplice (per esempio, polinomiale) su cui si possa facilmente operare.

L'approssimazione di una funzione è utilizzata per esempio per la valutazione della funzione stessa su un calcolatore, oppure per rendere possibili operazioni quali la derivazione o l'integrazione.

**Approssimare un insieme di dati**  $(x_i, y_i)$  (ove, per esempio,  $y_i = f(x_i)$  e  $f$  non è nota esplicitamente) significa determinare una funzione  $\tilde{f}$  che abbia un andamento analogo a quello dei dati (o della funzione che ha generato i suddetti dati).

Un'applicazione di tale operazione si ha, per esempio, nella statistica, nella finanza, nella fisica e ogniqualvolta si vuole costruire un modello che descriva il fenomeno in questione e ci permetta di fare previsioni attendibili in punti  $x$  diversi da  $x_i$ .

Siano assegnati  $n + 1$  dati  $(x_i, y_i)$ ,  $i = 1, \dots, n + 1$ , ove  $y_i$  è il valore assunto da una certa funzione  $f(x)$  in  $x_i$  oppure la misura di un dato sperimentale in corrispondenza di  $x_i$ .

## Definizione

Il **criterio dell'interpolazione** consiste nello scegliere, come approssimante dei dati  $(x_i, y_i)$ ,  $i = 1, \dots, n + 1$ , una funzione  $\tilde{f}(x)$  soddisfacente le seguenti condizioni:

$$\tilde{f}(x_i) = y_i \quad i = 1, \dots, n + 1,$$

che impongono il passaggio di  $\tilde{f}$  per i punti  $(x_i, y_i)$ .

Le suddette equazioni sono dette **condizioni di interpolazione** e i punti  $x_i$  sono detti **nodi di interpolazione**; inoltre, si dice che  $\tilde{f}$  **interpola i dati**  $(x_i, y_i)$  oppure, se  $y_i = f(x_i)$ , **interpola la funzione**  $f$  nei punti  $x_i$ .

In seguito verranno considerate:

- funzioni interpolanti  $\tilde{f}(x)$  **polinomiali**, generalmente utilizzate per lo sviluppo di algoritmi nella differenziazione, nell'integrazione e nella soluzione di problemi differenziali;
- funzioni interpolanti  $\tilde{f}(x)$  **polinomiali a tratti**, largamente utilizzate nella grafica e nella soluzione di problemi differenziali.

# Interpolazione mediante polinomi

## Teorema

Assegnati  $n + 1$  punti  $(x_i, y_i)$ , con le ascisse distinte tra loro ( $x_i \neq x_j$  per  $i \neq j$ ), esiste uno e un sol polinomio  $p_n(x)$  di grado minore o uguale a  $n$  interpolante i dati assegnati, ovvero soddisfacente le condizioni di interpolazione:

$$p_n(x_i) = y_i, \quad i = 1, \dots, n + 1$$

## Dimostrazione

Consideriamo il generico polinomio di grado  $n$

$$p_n(x) = c_1 x^n + c_2 x^{n-1} + \dots + c_n x + c_{n+1},$$

definito dagli  $n + 1$  coefficienti  $c_1, \dots, c_{n+1}$  e imponiamo le condizioni di interpolazione:

$$c_1 x_i^n + c_2 x_i^{n-1} + \dots + c_n x_i + c_{n+1} = y_i, \quad i = 1, \dots, n + 1.$$

## ... continua dimostrazione

Otteniamo così il sistema lineare

$$\begin{pmatrix} x_1^n & \dots & x_1 & 1 \\ x_2^n & \dots & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n+1}^n & \dots & x_{n+1} & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n+1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n+1} \end{pmatrix} \iff \mathbf{V}_n \mathbf{c} = \mathbf{y}$$

la cui matrice dei coefficienti  $\mathbf{V}_n$  è la matrice di Vandermonde associata al vettore  $\mathbf{x} = (x_1, x_2, \dots, x_{n+1})^T$ . Poiché

$$\det(\mathbf{V}_n) = \prod_{i>j} (x_i - x_j)$$

e i nodi  $x_j$  sono distinti, la matrice  $\mathbf{V}_n$  è non singolare e dunque

$$\exists! \mathbf{c}^* = (c_1^*, \dots, c_{n+1}^*)^T : \mathbf{V}_n \mathbf{c}^* = \mathbf{y}.$$

# Rappresentazione monomiale

Il polinomio interpolante, pur essendo unico, ammette più di una rappresentazione.

## Definizione

Si definisce **rappresentazione monomiale** del polinomio interpolante  $p_n(x)$  l'espressione:

$$p_n(x) = c_1x^n + c_2x^{n-1} + \dots + c_nx + c_{n+1},$$

ove i coefficienti  $c_i$ ,  $i = 1, \dots, n + 1$ , sono determinati imponendo che  $p_n(x)$  soddisfi le condizioni di interpolazione.

I seguenti comandi `MATLAB` consentono di ottenere i coefficienti della rappresentazione monomiale e di valutare il polinomio interpolante in qualsiasi punto o vettore di punti assegnato.



## Comandi MATLAB

- `c = polyfit(x,y,n)` calcola e memorizza in `c`, i coefficienti della rappresentazione monomiale del polinomio di grado `n`,

$$p_n(x) = c(1)x^n + c(2)x^{n-1} + \dots + c(n)x + c(n+1),$$

**interpolante** i dati  $(x_i, y_i)$ ,  $i = 1, \dots, n+1$ , le cui ascisse  $x_i$  e ordinate  $y_i$  sono memorizzate nei vettori `x` e `y`, rispettivamente.

- `p = polyval(c,z)` calcola e memorizza nel vettore `p`, i valori che un **qualsiasi** polinomio

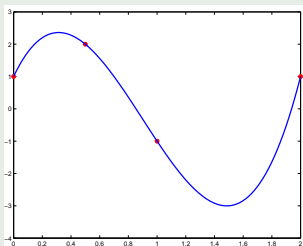
$$p_n(x) = c(1)x^n + c(2)x^{n-1} + \dots + c(n)x + c(n+1),$$

i cui coefficienti sono memorizzati in `c`, assume nelle componenti del vettore `z`. I coefficienti devono essere ordinati dal termine di grado massimo al termine di grado zero.

## Esempio 1

Le seguenti istruzioni MATLAB consentono di calcolare e rappresentare graficamente il polinomio interpolante i dati  $(0, 1)$ ,  $(1, -1)$ ,  $(2, 1)$ ,  $(1/2, 2)$ .

```
x = [0 1 2 1/2];  
y = [1 -1 1 2];  
c = polyfit(x,y,length(x)-1);  
z = linspace(min(x),max(x));  
p = polyval(c,z);  
plot(x,y,'ro',z,p,'b')
```

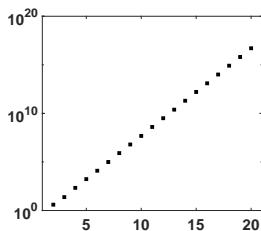


Tuttavia la determinazione del polinomio interpolante mediante la sua rappresentazione monomiale comporta la risoluzione di un problema mal condizionato, ovvero la risoluzione di un sistema lineare  $\mathbf{V}_n \mathbf{c} = \mathbf{y}$  la cui matrice dei coefficienti è una matrice di Vandermonde.

In figura è riportato il numero di condizionamento  $K_\infty$  (in norma infinito) della matrice di Vandermonde generata dal vettore  $\mathbf{x}$  di  $n = 2, \dots, 20$  elementi equispaziati in  $[0, 1]$ .

Si nota che il condizionamento della matrice peggiora al crescere dell'ordine  $n$  e, per questo motivo, non è possibile utilizzare la rappresentazione monomiale del polinomio interpolante quando  $n$  è grande.

Figura:  $K_\infty(\mathbf{V}_n)$  per  $n = 2, 3, \dots, 20$ .



Pertanto, quando  $n$  è grande, è necessario calcolare il polinomio interpolante attraverso rappresentazioni alternative.

## Definizione

Si definisce **rappresentazione di Lagrange** del polinomio  $p_n$  interpolante i dati  $(x_i, y_i)$ ,  $i = 1, \dots, n+1$  con  $x_i \neq x_j$  per  $i \neq j$ , l'espressione

$$p_n(x) = \sum_{j=1}^{n+1} y_j \ell_j(x)$$

I polinomi

$$\ell_j(x) = \frac{(x - x_1) \cdots (x - x_{j-1})(x - x_{j+1}) \cdots (x - x_{n+1})}{(x_j - x_1) \cdots (x_j - x_{j-1})(x_j - x_{j+1}) \cdots (x_j - x_{n+1})}$$

$j = 1, \dots, n+1$ , di grado  $n$  sono detti **polinomi fondamentali di Lagrange** associati ai nodi  $x_j$ .

Poiché per  $i = 1, \dots, n + 1$  si ha

$$\ell_j(x_i) = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases},$$

il polinomio  $\ell_j(x)$  interpola i dati:

$$(x_1, 0), \dots, (x_{j-1}, 0), (x_j, 1), (x_{j+1}, 0), \dots, (x_{n+1}, 0)$$

Il polinomio

$$p_n(x) = \sum_{j=1}^{n+1} y_j \ell_j(x),$$

invece, interpola i dati  $(x_i, y_i)$ ,  $i = 1, \dots, n + 1$ :

$$p_n(x_i) = \sum_{j=1}^{n+1} y_j \ell_j(x_i) = y_i \ell_i(x_i) = y_i, \quad i = 1, \dots, n + 1$$

## Esempio

Scriviamo la rappresentazione di Lagrange del polinomio  $p_3(x)$  interpolante i dati  $(0, 1)$ ,  $(1, -1)$ ,  $(2, 1)$ ,  $(1/2, 2)$ , precedentemente considerati.

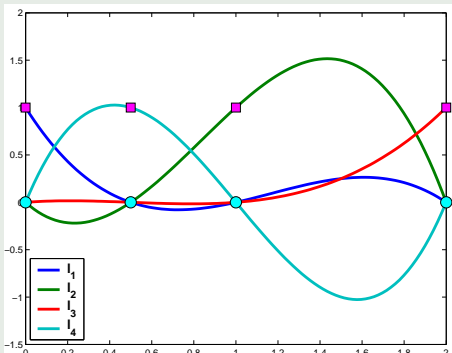
Per i polinomi  $\ell_j(x)$ ,  $j = 1, 2, 3, 4$ , si ha:

$$\ell_1(x) = \frac{(x-1)(x-2)(x-1/2)}{(0-1)(0-2)(0-1/2)}$$

$$\ell_2(x) = \frac{(x-0)(x-2)(x-1/2)}{(1-0)(1-2)(1-1/2)}$$

$$\ell_3(x) = \frac{(x-0)(x-1)(x-1/2)}{(2-0)(2-1)(2-1/2)}$$

$$\ell_4(x) = \frac{(x-0)(x-1)(x-2)}{(1/2-0)(1/2-1)(1/2-2)}$$



Quindi la rappresentazione di Lagrange del polinomio  $p_3(x)$  interpolante è

$$p_3(x) = 1 \cdot \ell_1(x) - 1 \cdot \ell_2(x) + 1 \cdot \ell_3(x) + 2 \cdot \ell_4(x)$$

## Osservazioni

- La rappresentazione di Lagrange del polinomio di interpolazione è interessante perché rappresenta la base per la costruzione di formule di integrazione numerica e di metodi per l'approssimazione della soluzione di problemi differenziali.
- Dal punto di vista del calcolo numerico del polinomio, la rappresentazione di Lagrange del polinomio interpolante è di interesse pratico solo per un numero piccolo di dati di interpolazione oppure quando i nodi  $x_i$  sono fissati e si calcolano i polinomi di interpolazione corrispondenti a diversi insiemi di valori  $\{y_i\}$ . Infatti, poiché ciascun polinomio  $\ell_j(x)$  dipende da tutti i nodi  $x_1, \dots, x_{n+1}$ , se si desidera aggiungere un dato di interpolazione  $(x_{n+2}, y_{n+2})$  all'insieme dei dati  $\{(x_i, y_i)\}_{i=1, \dots, n+1}$ , occorre ricalcolare tutti i polinomi  $\ell_j(x)$ !
- Per il calcolo effettivo del polinomio interpolante si utilizza una rappresentazione, alternativa a quella monomiale e di Lagrange, nota sotto il nome di rappresentazione di Newton.



# Rappresentazione di Newton

Per definire la rappresentazione di Newton del polinomio di interpolazione, introduciamo la seguente definizione. Siano assegnati  $n + 1$  dati  $(x_i, y_i)$ ,  $i = 1, \dots, n + 1$ , con  $x_i \neq x_j$  per  $i \neq j$  e, eventualmente,  $y_i = f(x_i)$ .

## Definizione

Si definisce **differenza divisa di ordine 1** di  $f(x)$  associata ai nodi  $x_1$  e  $x_2$ , la quantità

$$f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

**differenza divisa di ordine 2** di  $f(x)$  associata ai nodi  $x_1, x_2$  e  $x_3$ ,

$$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}$$

e, più in generale, **differenza divisa di ordine  $n$**  di  $f(x)$  associata ai nodi  $x_1, \dots, x_{n+1}$ ,

$$f[x_1, x_2, \dots, x_{n+1}] = \frac{f[x_2, \dots, x_{n+1}] - f[x_1, \dots, x_n]}{x_{n+1} - x_1}$$

## Proprietà

Le differenze divise sono invarianti rispetto a permutazioni degli argomenti, cioè

$$f[x_1, x_2] = f[x_2, x_1], \quad f[x_1, x_2, x_3] = f[x_2, x_1, x_3] = f[x_3, x_2, x_1], \dots$$

## Osservazione

Quando due argomenti risultano coincidenti si può ugualmente dare significato alla differenza divisa di ordine 1, purchè  $f'(x)$  esista in quel punto:

$$f[x_1, x_1] = \lim_{x \rightarrow x_1} f[x_1, x] = \lim_{x \rightarrow x_1} \frac{f(x) - f(x_1)}{x - x_1} = f'(x_1)$$

Più in generale, se esiste  $f^{(k)}(x_1)$  si definisce

$$f[\underbrace{x_1, x_1, \dots, x_1}_{k+1}] = \frac{f^{(k)}(x_1)}{k!}$$

È facile verificare che il seguente polinomio di grado 0

$$p_0(x) = f(x_1)$$

interpola  $f(x)$  in  $x_1$  e il polinomio di grado 1

$$p_1(x) = \underbrace{f(x_1)}_{p_0(x)} + f[x_1, x_2](x - x_1)$$

interpola  $f(x)$  in  $x_1$  e in  $x_2$ . Così proseguendo, denotato con  $p_{n-1}(x)$  il polinomio di grado  $n - 1$  interpolante  $f(x)$  in  $x_1, \dots, x_n$ , il polinomio di grado  $n$

$$p_n(x) = p_{n-1}(x) + f[x_1, x_2, \dots, x_{n+1}](x - x_1) \dots (x - x_n)$$

interpola  $f(x)$  in  $x_1, \dots, x_{n+1}$ .

Dall'unicità del polinomio, di grado  $n$  e interpolante  $n + 1$  punti, segue allora che quest'ultima espressione del polinomio  $p_n$  è una terza rappresentazione del polinomio interpolante (già, precedentemente definito, nelle rappresentazioni monomiale e di Lagrange).

## Definizione

Si definisce **rappresentazione di Newton** del polinomio interpolante i dati  $(x_i, y_i)$  con  $y_i = f(x_i)$ , la seguente espressione

$$p_n(x) = f(x_1) + f[x_1, x_2](x - x_1) + f[x_1, x_2, x_3](x - x_1)(x - x_2) + \dots + f[x_1, x_2, \dots, x_{n+1}](x - x_1) \dots (x - x_n)$$

I polinomi  $1, x - x_1, \dots, (x - x_1) \dots (x - x_n)$  sono detti **polinomi fondamentali di Newton**.

## Osservazioni

- Se si vuole aggiungere all'insieme dei nodi  $x_1, \dots, x_{n+1}$  un ulteriore nodo  $x_{n+2}$ , poiché l'espressione del polinomio interpolante  $f(x)$  in  $x_i$ ,  $i = 1, \dots, n+2$ , è data da

$$p_{n+1}(x) = p_n(x) + f[x_1, x_2, \dots, x_{n+2}](x - x_1) \dots (x - x_{n+1}),$$

noto  $p_n$ , occorre calcolare soltanto l'ultimo addendo.

- Se  $f$  è sufficientemente regolare, il polinomio di Newton rappresenta una generalizzazione del polinomio di Taylor della funzione  $f$  nell'intorno di  $x_1$ : infatti, se tutti i punti  $x_i$  coincidono con  $x_1$ , in virtù della precedente osservazione sulle differenze divise ad argomenti coincidenti, l'espressione del polinomio di Newton coincide con quella del polinomio di Taylor della funzione  $f$ , di grado  $n$  e centrato in  $x_1$ .

Organizzando in maniera opportuna i calcoli, è possibile calcolare facilmente le differenze divise presenti nella rappresentazione di Newton del polinomio interpolante.

Si consideri, per esempio, il polinomio di grado 3 interpolante i dati  $(x_i, f(x_i))$ ,  $i = 1, 2, 3, 4$ :

$$p_3(x) = f(x_1) + f[x_1, x_2](x - x_1) + f[x_1, x_2, x_3](x - x_1)(x - x_2) + f[x_1, x_2, x_3, x_4](x - x_1)(x - x_2)(x - x_3)$$

Per determinare  $p_3(x)$  può essere utilizzata la seguente tabella, sulla cui diagonale si trovano le differenze divise presenti nella rappresentazione di Newton del polinomio stesso.

		ordine 1	ordine 2	ordine 3
$x_1$	$f(x_1)$			
$x_2$	$f(x_2)$	$f[x_1, x_2]$		
$x_3$	$f(x_3)$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	
$x_4$	$f(x_4)$	$f[x_3, x_4]$	$f[x_2, x_3, x_4]$	$f[x_1, x_2, x_3, x_4]$

## Esempio 1

Scriviamo la rappresentazione di Newton del polinomio  $p_3(x)$  interpolante i dati  $(0, 1)$ ,  $(1, -1)$ ,  $(2, 1)$ ,  $(1/2, 2)$ , precedentemente considerati.

Calcoliamo le differenze divise:

			ordine 1		ordine 2		ordine 3
0	1	$\searrow$					
1	-1	$\rightarrow$	$\frac{-1-1}{1-0} = -2$	$\searrow$			
2	1	$\rightarrow$	$\frac{1-(-1)}{2-1} = 2$	$\rightarrow$	$\frac{2-(-2)}{2-0} = 2$	$\searrow$	
1/2	2	$\rightarrow$	$\frac{2-1}{1/2-2} = -\frac{2}{3}$	$\rightarrow$	$\frac{-2/3-2}{1/2-1} = \frac{16}{3}$	$\rightarrow$	$\frac{16/3-2}{1/2-0} = \frac{20}{3}$

Quindi la rappresentazione di Newton del polinomio  $p_3(x)$ , interpolante i dati assegnati, è

$$p_3(x) = 1 - 2(x - 0) + 2(x - 0)(x - 1) + \frac{20}{3}(x - 0)(x - 1)(x - 2)$$

## Esempio 2

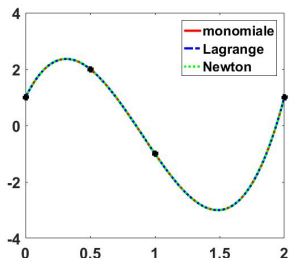
L'unicità del polinomio interpolante garantisce che le tre rappresentazioni (monomiale, di Lagrange e di Newton) che abbiamo determinato, individuano lo stesso polinomio. Verifichiamo quanto affermato, implementando il calcolo delle tre rappresentazioni del polinomio interpolante i dati  $(0, 1)$ ,  $(1, -1)$ ,  $(2, 1)$ ,  $(1/2, 2)$ , considerati negli esempi precedenti.



## ... continua esempio

```
x = [0 1 2 1/2];
y = [1 -1 1 2];
z = linspace(0,2,100);
% rappresentazione monomiale
c = polyfit(x,y,3);
p1 = polyval(c,z);
% rappresentazione di Lagrange
elle1 = ((z-1).*(z-2).*(z-1/2))/((0-1)*(0-2)*(0-1/2));
elle2 = ((z-0).*(z-2).*(z-1/2))/((1-0)*(1-2)*(1-1/2));
elle3 = ((z-0).*(z-1).*(z-1/2))/((2-0)*(2-1)*(2-1/2));
elle4 = ((z-0).*(z-1).*(z-2))/((1/2-0)*(1/2-1)*(1/2-2));
p2 = elle1-elle2+elle3+2*elle4;
% rappresentazione di Newton
p3 = 1-2*(z-0)+2*(z-0).*(z-1)+20/3*(z-0).*(z-1).*(z-2);
plot(z,p1,'r-',z,p2,'b-.',z,p3,'g:', 'linewidth',3)
legend('monomiale','Lagrange','Newton')
hold on
plot(x,y,'ko')
err1 = max(abs(p3-p1))
err2 = max(abs(p3-p2))
err1 =
    1.6875e-14
ans =
    8.8818e-16
```

## ... continua esempio



Si osservi che, prendendo come polinomio di riferimento il polinomio di Newton, il massimo errore assoluto in 100 punti associato al polinomio determinato con `polyfit` è dell'ordine di  $10^{-14}$ , quello associato al polinomio di Lagrange è dell'ordine della precisione di macchina. Nel primo caso l'errore è maggiore a causa del cattivo condizionamento precedentemente menzionato.

Per il calcolo degli elementi diagonali della tabella delle differenze divise si procede nel seguente modo: nel vettore  $y$  vengono inizialmente memorizzati i valori  $f(x_i)$ ,  $i = 1, \dots, n + 1$ . Successivamente, vengono determinati in modo ricorsivo, colonna dopo colonna, gli elementi di ciascuna colonna, partendo dall'ultimo elemento sino ad ottenere quello diagonale e sovrascrivendo i nuovi elementi su quelli precedenti. Al termine del processo il vettore  $y$  conterrà tutti gli elementi diagonali.

### Function MATLAB

```
function a = difdiv(x,y)
n = length(x)-1
for i = 1:n
    for j = n+1:-1:i+1
        y(j) = (y(j)-y(j-1))/(x(j)-x(j-i));
    end
end
a = y;
```

## Osservazione

Il costo dell'algoritmo per la costruzione della tabella triangolare è pari a  $n^2$  somme o sottrazioni e  $n^2/2$  divisioni

Descriviamo ora l'algoritmo che consente di valutare in un punto  $z$  oppure in un vettore di punti il polinomio interpolante. A tal scopo consideriamo la rappresentazione di Newton del polinomio di grado 3 interpolante i punti  $(x_1, f(x_1))$ ,  $(x_2, f(x_2))$ ,  $(x_3, f(x_3))$ ,  $(x_4, f(x_4))$  e, indicando con  $a_i$  le differenze divise in esso presenti, scriviamo

$$\begin{aligned} p_3(z) &= a_1 + a_2(z - x_1) + a_3(z - x_1)(z - x_2) \\ &\quad + a_4(z - x_1)(z - x_2)(z - x_3) \\ &= [a_4(z - x_3) + a_3](z - x_1)(z - x_2) + a_2(z - x_1) + a_1 \\ &= \{ \underbrace{[ \underbrace{a_4(z - x_3) + a_3}_p ]}_{p} (z - x_2) + a_2 \}_{p} (z - x_1) + a_1 \end{aligned}$$

Ne deduciamo il seguente schema di calcolo:

$$\left\{ \begin{array}{l} p = a_4 \\ p = p(z - x_3) + a_3 \\ p = p(z - x_2) + a_2 \\ p = p(z - x_1) + a_1 \end{array} \right. \Rightarrow \left\{ \begin{array}{l} p = a_{n+1} \\ p = p(z - x_k) + a_k, \quad k = n, \dots, 1 \end{array} \right.$$

Tale algoritmo noto sotto il nome di **algoritmo di Horner** è l'algoritmo più stabile e meno costoso in termini di operazioni aritmetiche per la valutazione di un polinomio. Esso infatti richiede  $n$  flops, mentre la rappresentazione monomiale richiederebbe  $n$  flops e  $n - 1$  moltiplicazioni.

## Function MATLAB

```
function p = interp(x,a,z)
n = length(x)-1;
p = a(n+1)*ones(1,length(z));
for k = n:-1:1
    p = p.*(z-x(k))+a(k);
end
```

# Condizionamento del polinomio di interpolazione

Per studiare il condizionamento del problema di interpolazione polinomiale introduciamo la seguente norma di funzione.

## Definizione

Sia  $g(x)$  una funzione continua in  $[a, b]$ . Si definisce **norma uniforme** o **norma del massimo** della funzione  $g(x)$  la quantità

$$\|g\|_{\infty} = \max_{x \in [a, b]} |g(x)|$$

Esaminiamo quindi la reazione del polinomio interpolante  $p_n(x)$  alla presenza di perturbazioni nelle ordinate  $f(x_j)$ ,  $j = 1, \dots, n+1$ .

Introdotti i valori perturbati  $\bar{f}_j = f(x_j) + \epsilon_j$ ,  $j = 1, \dots, n+1$ , denotiamo con

$$\bar{p}_n(x) = \sum_{j=1}^{n+1} \bar{f}_j \ell_j(x), \quad x \in [a, b]$$

il polinomio interpolante i dati perturbati. Si ha allora

$$p_n(x) - \bar{p}_n(x) = \sum_{j=1}^{n+1} (f(x_j) - \bar{f}_j) \ell_j(x)$$

da cui

$$|p_n(x) - \bar{p}_n(x)| \leq \sum_{j=1}^{n+1} |f(x_j) - \bar{f}_j| |\ell_j(x)| \leq \max_{1 \leq j \leq n+1} |\epsilon_j| \sum_{j=1}^{n+1} |\ell_j(x)|$$



Denotando con

$$\Lambda_n = \left\| \sum_{j=1}^{n+1} |\ell_j| \right\|_{\infty}$$

e con  $\epsilon = (\epsilon_1, \epsilon_2, \dots, \epsilon_{n+1})^T$  si ottiene

$$|p_n(x) - \bar{p}_n(x)| \leq \Lambda_n \|\epsilon\|_{\infty} \quad \forall x \in [a, b] \quad \Rightarrow \quad \|p_n(x) - \bar{p}_n(x)\|_{\infty} \leq \Lambda_n \|\epsilon\|_{\infty}$$

Poiché

$$\|p_n\|_{\infty} = \max_{x \in [a, b]} |p_n(x)| \geq \max_{1 \leq j \leq n+1} |p_n(x_j)| = \max_{1 \leq j \leq n+1} |f(x_j)| = \|\mathbf{f}\|_{\infty}$$

con  $\mathbf{f} = (f(x_1), f(x_2), \dots, f(x_{n+1}))^T$ , possiamo scrivere

$$\frac{\|p_n - \bar{p}_n\|_{\infty}}{\|p_n\|_{\infty}} \leq \Lambda_n \frac{\|\epsilon\|_{\infty}}{\|p_n\|_{\infty}} \quad \Rightarrow \quad \frac{\|p_n - \bar{p}_n\|_{\infty}}{\|p_n\|_{\infty}} \leq \Lambda_n \frac{\|\epsilon\|_{\infty}}{\|\mathbf{f}\|_{\infty}}$$

## Definizione

Il numero  $\Lambda_n$  è detto **costante di Lebesgue** associata ai nodi  $x_1, x_2, \dots, x_{n+1}$  e all'intervallo  $[a, b]$ .

Osserviamo che la costante di Lebesgue rappresenta una maggiorazione del coefficiente di amplificazione degli errori  $\epsilon_j$ ,  $j = 1, \dots, n + 1$ , nei dati sul risultato.

Si dimostra che, **qualunque sia la scelta dei nodi di interpolazione**,

$$\lim_{n \rightarrow \infty} \Lambda_n = +\infty$$

Nel caso “ottimale” l'ordine di divergenza è di tipo logaritmico.

In particolare per le seguenti scelte di nodi, tutte appartenenti al generico intervallo  $[a, b]$ , si hanno le seguenti costanti di Lebesgue.

Per i **nodì di Chebyshev**

$$x_i = \frac{b-a}{2} \cos\left(\frac{(2i-1)\pi}{2(n+1)}\right) + \frac{b+a}{2}, \quad i = 1, \dots, n+1$$

e per i **nodì di Chebyshev-Lobatto**

$$x_i = \frac{b-a}{2} \cos\left(\frac{(i-1)\pi}{n}\right) + \frac{b+a}{2}, \quad i = 1, \dots, n+1$$

si ha

$$\Lambda_n = O(\log n) \quad \text{per } n \rightarrow \infty$$

Per i **nodì equispaziati**:

$$x_i = a + (i-1) \frac{b-a}{n}, \quad i = 1, \dots, n+1$$

si ha

$$\Lambda_n = O(e^n) \quad \text{per } n \rightarrow \infty$$

## Definizioni

Denotato con  $p_n(x)$  il polinomio interpolante una funzione continua  $f(x)$  nei punti  $x_1, \dots, x_{n+1}$ , si definisce **errore di interpolazione** la funzione

$$E_n(x) = f(x) - p_n(x)$$

Si dice che la successione di polinomi interpolanti  $\{p_n\}_{n \in \mathbb{N}}$  **converge uniformemente** a  $f$  se, e solo se,

$$\lim_{n \rightarrow \infty} \|E_n\|_{\infty} = 0$$

Nell'ipotesi  $f(x) \in C^{n+1}([a, b])$  è possibile ricavare un'espressione per l'errore di interpolazione  $E_n(x)$ . Vale infatti il seguente teorema.

## Teorema

Assegnati  $n + 1$  punti  $(x_i, y_i)$ , con le ascisse distinte tra loro e tutte appartenenti ad  $[a, b]$  e  $y_i = f(x_i)$  per  $i = 1, \dots, n + 1$ , se  $f(x) \in C^{n+1}([a, b])$  allora per ogni fissato  $x$  esiste  $\xi \in (a, b)$  tale che

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

ove

$$\omega_{n+1}(x) = (x - x_1) \cdot (x - x_2) \cdot \dots \cdot (x - x_{n+1})$$

## Dimostrazione

Ricordando che  $p_n(x_i) = f(x_i)$ ,  $i = 1, \dots, n + 1$ , possiamo scrivere

$$E_n(x) = f(x) - p_n(x) = R_n(x) \omega_{n+1}(x)$$

## ... continua dimostrazione

Per definire  $R_n(x)$ , introduciamo la funzione ausiliaria:

$$G(t) = f(t) - p_n(t) - R_n(x)\omega_{n+1}(t), \quad t \in [a, b]$$

Poiché  $G(t) \in C^{n+1}([a, b])$ ,  $G(x_i) = 0$ , per  $i = 1, \dots, n+1$ , e  $G(x) = 0$ , possiamo applicare il teorema di Rolle e affermare che esistono  $n+1$  punti distinti  $\xi_1^{(1)}, \xi_2^{(1)}, \dots, \xi_{n+1}^{(1)}$  in  $(a, b)$ , nei quali  $G^{(1)}(\xi_i^{(1)}) = 0$ ,  $i = 1, \dots, n+1$ .

Applicando nuovamente il teorema di Rolle alla funzione  $G^{(1)}(x) \in C^n([a, b])$ , possiamo affermare che esistono  $n$  punti distinti  $\xi_1^{(2)}, \xi_2^{(2)}, \dots, \xi_n^{(2)}$  in  $(a, b)$ , nei quali  $G^{(2)}(\xi_i^{(2)}) = 0$ ,  $i = 1, \dots, n$ .

Proseguendo allo stesso modo con le derivate successive, sino alla  $(n+1)$ -esima, possiamo affermare che esiste  $\xi := \xi_1^{(n+1)}$  in  $(a, b)$  per il quale si ha  $G^{(n+1)}(\xi) = 0$ . Tenendo infine conto che  $\omega_{n+1}^{(n+1)}(x) = (n+1)!$  e che  $p_n^{(n+1)}(x) = 0$ , possiamo scrivere

$$0 = G^{(n+1)}(\xi) = f^{(n+1)}(\xi) - (n+1)!R_n(x) \quad \Rightarrow \quad R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

e quindi la tesi, ricordando che  $E_n(x) = R_n(x)\omega_{n+1}(x)$ .

- La rappresentazione dell'errore  $E_n(x)$  che abbiamo costruito riveste un'importanza soprattutto teorica. Infatti, il suo utilizzo richiede non solo la determinazione della derivata  $f^{(n+1)}$ , ma anche la valutazione di quest'ultima in un punto  $\xi = \xi_1^{(n+1)}$  (dipendente da  $x$ ) non noto.
- Quando la funzione  $f(x)$  è un polinomio di grado non superiore a  $n$ , la derivata  $f^{(n+1)}$  è identicamente nulla e dalla relazione precedente deduciamo che  $f(x) \equiv p_n(x)$ .
- Supponiamo che gli  $n + 1$  punti distinti appartengano all'intervallo  $I = [x, x + h]$  e che si voglia approssimare in  $I$  una funzione  $f(x) \in C^{n+1}(I)$  con un polinomio interpolante di grado  $n$  fissato. Denotato con  $M_{n+1} = \max_{x \in I} f^{(n+1)}(x)$  e tenendo conto che  $|\omega_{n+1}(x)| \leq h^{n+1}$ , si ha

$$\max_{x \in I} |f(x) - p_n(x)| \leq \frac{h^{n+1}}{(n+1)!} M_{n+1} \Rightarrow \|f - p_n\|_{\infty} = O(h^{n+1}), \quad h \rightarrow 0$$

Quest'ultima relazione giustifica l'interesse per la costruzione di funzioni polinomiali a tratti.

Determiniamo ora un'altra espressione dell'errore di interpolazione a partire dalla rappresentazione di Newton del polinomio interpolante.

## Teorema

Assegnati  $n + 1$  punti  $(x_i, y_i)$ , con le ascisse distinte tra loro e tutte appartenenti ad  $[a, b]$  e  $y_i = f(x_i)$ , allora

$$E_n(x) = f[x, x_1, \dots, x_{n+1}] \omega_{n+1}(x)$$

## Dimostrazione

Riprendiamo la definizione di differenza divisa e scriviamo

$$f[x, x_1] = \frac{f(x_1) - f(x)}{x_1 - x}$$

da cui

$$f(x) = f(x_1) + f[x, x_1](x - x_1)$$

Il termine  $p_0(x) = f(x_1)$  rappresenta il polinomio interpolante la funzione  $f(x)$  nel nodo  $x = x_1$ , mentre  $E_0(x) = (x - x_1)f[x, x_1]$  denota l'errore  $f(x) - p_0(x)$ .



## ... continua dimostrazione

Successivamente consideriamo

$$f[x, x_1, x_2] = \frac{f[x_1, x_2] - f[x, x_1]}{x_2 - x_1}$$

ovvero

$$f[x, x_1] = f[x_1, x_2] + f[x, x_1, x_2](x - x_2)$$

e sostituiamo quest'ultima nell'espressione precedentemente determinata per  $f(x)$ :

$$f(x) = f(x_1) + f[x_1, x_2](x - x_1) + f[x, x_1, x_2](x - x_1)(x - x_2)$$

Il polinomio  $p_1(x) = f(x_1) + f[x_1, x_2](x - x_1)$ , di grado 1, è il polinomio interpolante la funzione  $f(x)$  in  $x_1$  e in  $x_2$ , mentre

$E_1(x) = f[x, x_1, x_2](x - x_1)(x - x_2)$  denota l'errore di interpolazione  $f(x) - p_1(x)$ . Così proseguendo, sino a includere tutti i nodi  $x_1, \dots, x_{n+1}$ , si perviene alla rappresentazione:

$$f(x) = p_n(x) + f[x, x_1, x_2, \dots, x_{n+1}](x - x_1)(x - x_2) \cdot \dots \cdot (x - x_{n+1})$$

da cui segue immediatamente la tesi.

Sfortunatamente a un aumento del grado  $n$  del polinomio interpolante, non corrisponde necessariamente un miglioramento nell'approssimazione di una funzione  $f$ !

Enunciamo un risultato di NON convergenza:

### Teorema

Data una **qualunque successione di nodi distinti**, tutti situati in  $[a, b]$ , esiste sempre una **funzione continua**  $f(x)$  in  $[a, b]$  che, interpolata su quei nodi, genera una successione di polinomi di interpolazione  $\{p_n(x)\}$  **non uniformemente convergente** a  $f(x)$  in  $[a, b]$ .

## Esempio

Se si considerano i **nodi equispaziati** in  $[a, b] = [-5, 5]$ ,

$$x_i = a + (i - 1)h, \quad i = 1, \dots, n + 1, \quad h = \frac{b - a}{n}$$

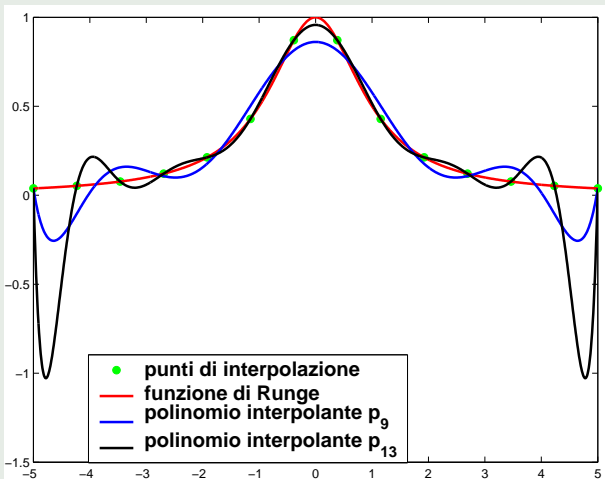
come nodi di interpolazione, l'esempio che segue mostra che la successione dei polinomi interpolanti la funzione continua (infinitamente derivabile)

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

detta **funzione di Runge**, non converge uniformemente a  $f(x)$  in  $[a, b]$ , al tendere del numero dei nodi all'infinito.

... continua

Il limite della norma uniforme dell'errore di interpolazione è addirittura infinito!



Enunciamo un risultato di convergenza.

## Teorema

Sia

$$\begin{array}{cccc} x_1^{(0)} & & & \\ x_1^{(1)} & x_2^{(1)} & & \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \\ \dots & \dots & \dots & \dots \\ x_1^{(n)} & x_2^{(n)} & \dots & x_{n+1}^{(n)} \\ \dots & \dots & \dots & \dots \end{array}$$

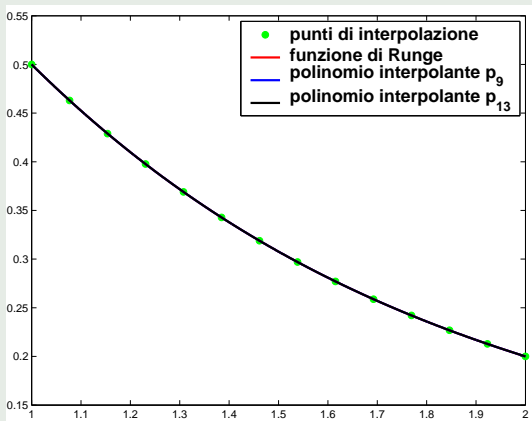
una successione di nodi distinti, tutti in  $[a, b]$  e sia  $f(z)$  una funzione analitica in una regione  $D$  contenente  $[a, b]$ . Si denoti con  $p_n(x)$  il polinomio di grado  $n$  interpolante  $f$  nei nodi  $x_1^{(n)}, x_2^{(n)}, \dots, x_{n+1}^{(n)}$ . Se la distanza delle singolarità di  $f(z)$  da  $[a, b]$  è maggiore di  $b - a$ , allora

$$\lim_{n \rightarrow \infty} \|f - p_n\|_{\infty} = 0.$$

## Esempio

Denotato con  $p_n$  il polinomio interpolante  $f(x) = \frac{1}{1+x^2}$  nei nodi  $x_i$ ,  $i = 1, \dots, n+1$ , equispaziati in  $[1, 2]$ , in virtù del precedente teorema possiamo affermare che

$$\lim_{n \rightarrow \infty} \|E_n\|_{\infty} = 0$$



Denotiamo con  $\mathbb{P}_n$  lo spazio lineare dei polinomi algebrici di grado minore o uguale a  $n$ . La costante di Lebesgue riveste un ruolo cruciale nella stima dell'errore di interpolazione. Per dimostrare quanto affermato, introduciamo la seguente definizione.

### Definizione

Sia  $f \in C^0([a, b])$ . Si definisce **polinomio di migliore approssimazione uniforme algebrica** per la funzione  $f$ , il polinomio  $q_n^*$  per il quale si ha

$$\|f - q_n^*\|_\infty = \min_{q_n \in \mathbb{P}_n} \|f - q_n\|_\infty$$

La quantità

$$E_n^*(x) = f(x) - q_n^*(x)$$

è detta **errore di migliore approssimazione uniforme algebrica**.

Si può dimostrare (teoremi di Jackson) che:

- Se  $f \in C^0([a, b])$ ,

$$\lim_{n \rightarrow \infty} \|E_n^*\|_\infty = 0$$

- Se  $f$  è hölderiana di ordine  $\mu > 0$  in  $[a, b]$ <sup>1</sup>, allora

$$\|E_n^*\|_\infty = O\left(\frac{1}{n^\mu}\right) \text{ per } n \rightarrow \infty$$

- Se  $f \in C^{k,\mu}([a, b])$  (ovvero  $f^{(k)}$  è hölderiana di ordine  $\mu$ ),  $k \geq 1$  e  $0 \leq \mu \leq 1$ , allora

$$\|E_n^*\|_\infty = O\left(\frac{1}{n^{k+\mu}}\right) \text{ per } n \rightarrow \infty$$

---

<sup>1</sup>Una funzione di variabile reale  $f : [a, b] \rightarrow \mathbb{R}$  soddisfa la condizione di Hölder di ordine  $\mu$ , con  $0 < \mu \leq 1$ , se esiste una costante  $C > 0$  tale che: per ogni  $x, y \in [a, b]$  si ha  $|f(x) - f(y)| \leq C|x - y|^\mu$ . Il numero  $\mu$  si dice esponente di Hölder, mentre  $f$  si dice Hölder-continua o hölderiana. Se  $\mu = 1$  allora  $f$  è lipschitziana.



Ciò premesso, possiamo dare il seguente risultato di convergenza.

### Teorema

Denotato con  $p_n(x)$  il polinomio interpolante una funzione continua  $f$  nei nodi distinti  $x_1, x_2, \dots, x_{n+1}$  e con

$$E_n^*(x) = f(x) - q_n^*(x)$$

l'errore di migliore approssimazione uniforme algebrica, risulta

$$\|f - p_n\|_\infty \leq (1 + \Lambda_n) \|E_n^*\|_\infty$$

## Dimostrazione

Per l'unicità del polinomio interpolante, possiamo scrivere

$$q_n^*(x) = \sum_{j=1}^{n+1} \ell_j(x) q_n^*(x_j)$$

Pertanto, si ha

$$\begin{aligned} f(x) - p_n(x) &= f(x) - q_n^*(x) + q_n^*(x) - p_n(x) \\ &= f(x) - q_n^*(x) + \sum_{j=1}^{n+1} \ell_j(x) (q_n^*(x_j) - f(x_j)) \end{aligned}$$

La tesi discende allora dalla seguente disuguaglianza:

$$\begin{aligned} |f(x) - p_n(x)| &\leq |f(x) - q_n^*(x)| + \sum_{j=1}^{n+1} |\ell_j(x)| |q_n^*(x_j) - f(x_j)| \\ &\leq \|f - q_n^*\|_\infty + \|f - q_n^*\|_\infty \sum_{j=1}^{n+1} |\ell_j(x)| \leq \|f - q_n^*\|_\infty (1 + \Lambda_n) \end{aligned}$$

Tenendo conto delle stime di convergenza per l'errore di migliore approssimazione uniforme algebrica e del comportamento della costante di Lebesgue associata ai nodi di Chebyshev e di Chebyshev-Lobatto, è immediato dimostrare il seguente teorema.

### Teorema

Sia  $\{p_n(x)\}$  la successione dei polinomi interpolanti  $f(x)$  nei nodi  $x_i \in [a, b]$  di Chebyshev oppure di Chebyshev-Lobatto.

Se  $f$  è hölderiana di ordine  $\mu > 0$  in  $[a, b]$

$$\|f - p_n\|_{\infty} = O\left(\frac{\log n}{n^{\mu}}\right), \quad n \rightarrow \infty$$

Se  $f \in C^{k,\mu}([a, b])$ ,  $k \geq 1$  e  $0 \leq \mu \leq 1$ , allora

$$\|f - p_n\|_{\infty} = O\left(\frac{\log n}{n^{k+\mu}}\right), \quad n \rightarrow \infty$$

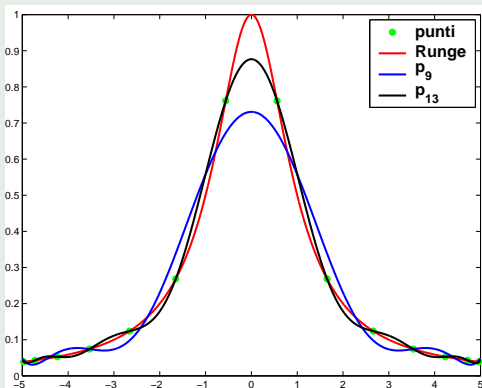
## Osservazioni

- Dal precedente teorema si deduce che la convergenza uniforme della successione dei polinomi interpolanti alla funzione interpolata  $f$ , è tanto più rapida quanto più è regolare la funzione  $f$ .
- Si precisa che la convergenza uniforme NON è garantita se la funzione  $f$  è solo continua, è invece garantita se la funzione  $f$  è almeno hölderiana.

## Esempio

Denotato con  $p_n$  il polinomio interpolante  $f(x) = \frac{1}{1+x^2}$  nei nodi di Chebyshev  $x_i$ ,  $i = 1, \dots, n+1$ ,  $n = 9, 13$  e relativi all'intervallo  $[-5, 5]$ , si ha

$$\lim_{n \rightarrow \infty} \|f - p_n\|_{\infty} = 0$$



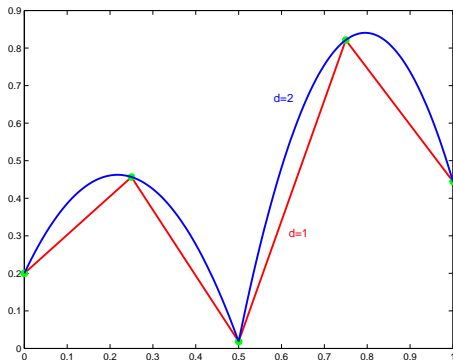
I precedenti risultati di convergenza mostrano dunque che:

- a un aumento del grado  $n$  del polinomio interpolante, non corrisponde necessariamente un miglioramento nell'approssimazione di una funzione  $f$ ;
- la convergenza uniforme del polinomio interpolante alla funzione interpolata dipende dalla scelta dei nodi e dalla regolarità della funzione  $f$ ;
- la scelta dei nodi equispaziati non garantisce la convergenza uniforme, neppure per funzioni dotate di derivata continua di qualunque ordine;
- se  $f$  è hölderiana, la scelta dei nodi di Chebyshev oppure di Chebyshev-Lobatto, invece, garantisce la convergenza uniforme.

## Definizione

Si definisce **funzione polinomiale a tratti** di grado  $d$  associata a una partizione dell'intervallo  $[a, b]$ , una funzione continua in  $[a, b]$  e definita da un'unione di tratti contigui di polinomi algebrici diversi, ciascuno di grado  $d$ .

In generale il grado  $d$  viene scelto piccolo ( $d = 1, 2, 3$ ).



Si osservi che le funzioni polinomiali a tratti sono generalmente continue ma non sono derivabili nei punti di raccordo.



Data una partizione  $a \equiv x_1 < \dots < x_{n+1} \equiv b$  di  $[a, b]$  e, in ciascuno intervallo  $[x_i, x_{i+1}]$  assegnate  $d + 1$  ascisse

$$x_i \equiv x_{i,1} < \dots < x_{i,d+1} \equiv x_{i+1},$$

supponiamo di conoscere le corrispondenti ordinate  $y_{i,j} = f(x_{i,j})$ ,  $j = 1, \dots, d + 1$ . A partire dai dati  $(x_{i,j}, y_{i,j})$  è possibile costruire la funzione polinomiale a tratti

$$f_n(x) = p_{d,i}(x), \quad x \in [x_i, x_{i+1}],$$

con

$$p_{d,i}(x) = \sum_{j=1}^{d+1} \ell_j(x) f(x_{i,j}), \quad p_{d,i}(x_{i,j}) = f(x_{i,j}), \quad i = 1, \dots, n, \quad j = 1, \dots, d + 1$$

Osserviamo che

- se  $x_{i,j} \in (x_i, x_{i+1})$ , per ogni  $j$ , allora  $f_n(x)$  è discontinua in  $x_i$  e in  $x_{i+1}$ ;
- se  $x_{i,1} = x_i$  e  $x_{i,d+1} = x_{i+1}$ , per ogni  $i$ , allora  $f_n(x) \in C([a, b])$ .

Se  $f(x) \in C^{d+1}([a, b])$ , utilizzando un precedente risultato di convergenza dell'interpolazione polinomiale e denotando con

$$M_{d+1} = \|f^{(d+1)}\|_{\infty}, \quad h_i = x_{i+1} - x_i, \quad h = \max_{1 \leq i \leq n} h_i,$$

si dimostra facilmente che

$$\|f - f_n\|_{\infty} = O(h^{d+1})$$

Questo risultato è infatti una conseguenza immediata del fatto che

$$\|f - f_n\|_{\infty} = \max_{1 \leq i \leq n} \|f - f_n\|_{\infty, [x_i, x_{i+1}]} = \max_{1 \leq i \leq n} \max_{x \in [x_i, x_{i+1}]} |f(x) - p_{i,d}(x)|$$

e della seguente stima

$$|f(x) - p_{i,d}(x)| \leq \frac{M_{d+1}}{(d+1)!} h_i^{d+1} \leq \frac{M_{d+1}}{(d+1)!} h^{d+1}, \quad \forall x \in [x_i, x_{i+1}], \quad \forall i$$

Osserviamo che, anche quando  $f \in C^m([a, b])$  con  $m > d + 1$ , si ha sempre  $\|f - f_n\|_{\infty} = O(h^{d+1})$ .

# Spline interpolanti

Le funzioni **spline** sono particolari funzioni polinomiali a tratti che soddisfano opportune proprietà di regolarità nei punti di raccordo. Se denotiamo con  $S_d(x)$  la generica spline di grado locale  $d$  definita in  $[a, b]$ , allora  $S_d(x) \in C^{d-1}([a, b])$ . Di seguito la definizione di spline interpolante.

## Definizione

Sia  $a \equiv x_1 < x_2 < \dots < x_{n+1} \equiv b$ , una partizione dell'intervallo  $[a, b]$  e  $y_i = f(x_i)$ . Si definisce **spline di ordine  $d$  interpolante**  $f$  nei nodi assegnati  $x_i$ , una funzione  $S_d(x)$  soddisfacente le seguenti condizioni:

- i)  $S_d(x)$  è un polinomio di grado (al più)  $d$  per  $x \in [x_i, x_{i+1}]$ ,  $i = 1, \dots, n$ ;
- ii) la derivata  $S_d^{(k)}(x)$  di ordine  $k$  di  $S_d(x)$  è una funzione continua in  $[a, b]$  per  $k = 0, 1, \dots, d - 1$ ;
- iii)  $S_d(x_i) = y_i$ ,  $i = 1, \dots, n + 1$ .

Le spline più utilizzate sono quelle di ordine  $d = 1, 2, 3$ .

## Osservazioni

- La derivata di una spline di ordine  $d$  è un'altra spline di ordine  $d - 1$ , mentre la primitiva di una spline di ordine  $d$  è una spline di ordine  $d + 1$ .
- La spline  $S_d$  dipende da  $(d + 1)n$  parametri  $\{c_{i,1}, c_{i,2}, \dots, c_{i,d+1}\}$ ,  $i = 1, \dots, n$ ; imponendo le  $n - 1$  condizioni ii) per  $k = 0, 1, 2, \dots, d - 1$ , e le  $n + 1$  condizioni di interpolazione iii), si ottiene un sistema lineare di  $d(n - 1) + (n + 1) = (d + 1)n - (d - 1)$  equazioni nelle  $(d + 1)n$  incognite  $\{c_{i,1}, c_{i,2}, \dots, c_{i,d+1}\}$ . Pertanto, per definire univocamente una spline occorre imporre  $d - 1$  ulteriori condizioni.
- Per le spline  $S_1(x)$  di ordine  $d = 1$  non occorre imporre nessuna condizione aggiuntiva ( $d - 1 = 0$ ).
- Per le spline  $S_2(x)$  di ordine  $d = 2$  occorre imporre una sola condizione aggiuntiva ( $d - 1 = 1$ ). Tipicamente si impone  $S'_2(a)$  oppure  $S'_2(b)$  uguale a un certo valore, oppure nel caso di funzioni  $f$  periodiche con periodo  $b - a$  si impone  $S'_2(a) = S'_2(b)$ .

La spline  $S_1(x)$  di ordine 1 e interpolante  $f$  nei punti  $x_i$ ,  $i = 1, \dots, n + 1$  ha la seguente espressione:

$$S_1(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i), \quad x \in [x_i, x_{i+1}]$$

Per calcolare e valutare la spline  $S_1(x)$  si può utilizzare il seguente comando MATLAB .

## Comando MATLAB

`s = interp1(x,y,z)` calcola e memorizza in `s` i valori che la spline lineare interpolante i dati  $(x_i, y_i)$ ,  $i = 1, \dots, n + 1$ , assume in `z`; i valori  $x_i$  e  $y_i$  sono memorizzati in `x` e in `y`, rispettivamente.

Per le spline lineare vale il seguente teorema di convergenza.

### Teorema

Data una partizione  $a \equiv x_1 < x_2 < \dots < x_{n+1} \equiv b$ , sia  $S_1$  la spline interpolante una funzione  $f \in C^2([a, b])$  nei nodi della partizione  $x_i$ ,  $i = 1, \dots, n+1$ . Denotato con  $h = \max_{1 \leq i \leq n} (x_{i+1} - x_i)$ , allora

$$\|f - S_1\|_{\infty} = O(h^2), \quad h \rightarrow 0$$

### Osservazione

La convergenza uniforme è comunque garantita qualunque sia la funzione continua  $f$  e qualunque sia la scelta dei nodi  $x_i$  per  $h \rightarrow 0$ .

# Spline cubiche

Per l'unicità di una spline cubica è necessario imporre due ulteriori condizioni aggiuntive ( $d - 1 = 2$ ). Fra tutte le possibili scelte, tipiche condizioni aggiuntive sono:

- 1)  $S_3^{(2)}(x_1) = 0, S_3^{(2)}(x_{n+1}) = 0$  **spline naturale**;
- 2)  $S_3^{(2)}(x_1) = f^{(2)}(x_1), S_3^{(2)}(x_{n+1}) = f^{(2)}(x_{n+1})$ ;
- 3)  $S_3^{(3)}(x_2^-) = S_3^{(3)}(x_2^+), S_3^{(3)}(x_n^-) = S_3^{(3)}(x_n^+)$  **spline “not-a-knot”**<sup>1</sup>;
- 4)  $S_3^{(1)}(x_1) = f^{(1)}(x_1), S_3^{(1)}(x_{n+1}) = f^{(1)}(x_{n+1})$  **spline vincolata**;
- 5)  $S_3^{(1)}(a) = S_3^{(1)}(b), S_3^{(2)}(a) = S_3^{(2)}(b)$  **spline periodica** (tali condizioni si impongono nel caso di fenomeni periodici di periodo  $b - a$  per cui le condizioni di interpolazione implicano  $S_3(a) = S_3(b)$ ).

---

<sup>1</sup>Imponendo tali condizioni, i due tratti (contigui) di polinomi (di terzo grado) che definiscono la spline cubica nell'intervallo  $(x_1, x_3)$  e  $(x_{n-1}, x_{n+1})$  appartengono a uno stesso polinomio di terzo grado. In questo caso, i nodi  $x_2$  e  $x_n$  sono nodi di interpolazione ma non più di partizione, nel senso che non separano più tratti di polinomi diversi.

Imponendo le condizioni 1), oppure 2), oppure 3), oppure 4), oppure 5), la spline corrispondente esiste ed è unica. Per determinare una spline cubica non si ricorre alla risoluzione di un sistema lineare nelle  $4n$  incognite  $\{c_{i,1}, c_{i,2}, c_{i,3}, c_{i,4}\}$ , ma alla risoluzione di un sistema lineare nelle  $n + 1$  incognite  $M_i = S_3^{(2)}(x_i)$ ,  $i = 1, \dots, n + 1$ , che gode di proprietà particolari.

In particolare, se si impone una delle condizioni aggiuntive 1)-4), il sistema lineare nelle  $n + 1$  (al massimo) incognite  $M_i$  risulta simmetrico, a diagonale dominante e tridiagonale, per la cui risoluzione è possibile utilizzare l'algoritmo di Thomas.

Per la scelta 5) il sistema lineare nelle  $n + 1$  incognite  $M_i$  non è più esattamente tridiagonale, ma è ancora simmetrico e a diagonale dominante.



Per quanto riguarda la convergenza uniforme delle spline cubiche interpolanti alla funzione interpolata, si può dimostrare il seguente teorema.

### Teorema

Sia  $S_3(x)$  la spline cubica interpolante i dati  $(x_i, f(x_i))$  con  $a \leq x_i \leq b$  e soddisfacente una delle condizioni aggiuntive 1)-5).

Denotato con  $h_i = x_{i+1} - x_i$  e  $h = \max_{1 \leq i \leq n} h_i$ , se  $f \in C^4([a, b])$  e se  $h/h_i \leq \gamma < \infty$  per  $h \rightarrow 0$ , si ha

$$\|f^{(p)} - S_3^{(p)}\|_{\infty} = O(h^{4-p}), \quad h \rightarrow 0, \quad p = 0, 1, 2, 3.$$

## Osservazioni

- La convergenza uniforme delle spline cubiche interpolanti, a differenza dei polinomi interpolanti, è comunque garantita qualunque sia la funzione continua  $f$  e qualunque sia la scelta dei nodi  $x_i$  per  $h \rightarrow 0$ .
- Il precedente teorema assicura l'approssimazione uniforme in  $[a, b]$  anche delle stesse derivate di  $f(x)$  (simultanea approssimazione).

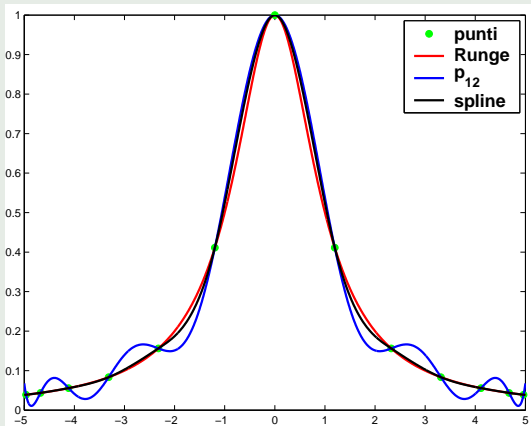
Per calcolare e valutare le spline “not-a-knot” o le spline vincolate si possono utilizzare i seguenti comandi.

### Comandi MATLAB

- `s = spline(x,y,z)` calcola e memorizza in `s` i valori che la spline cubica interpolante i dati  $(x_i, y_i)$ ,  $i = 1, \dots, n + 1$  e soddisfacente la condizione 3) “not-a-knot”, assume in `z`; i valori  $x_i$  e  $y_i$  sono memorizzati in `x` e in `y`, rispettivamente.
- `s = spline(x,[yd1 y ydn],z)` calcola e memorizza in `s` i valori che la spline cubica interpolante i dati  $(x_i, y_i)$ ,  $i = 1, \dots, n + 1$  e soddisfacente la condizione 4) (`yd1` =  $f'(x_1)$ , `ydn` =  $f'(x_{n+1})$ ), assume in `z`; i valori  $x_i$  e  $y_i$  sono memorizzati in `x` e in `y`, rispettivamente.

## Esempio

In figura sono rappresentati graficamente il polinomio e la spline cubica “not-a-knot” interpolanti la funzione di Runge in 13 nodi di Chebyshev definiti nell'intervallo  $[-5, 5]$ .



Le spline cubiche sono suscettibili di una rappresentazione analoga a quella di Lagrange per i polinomi. Per esempio, è possibile riscrivere la generica spline interpolante i dati  $(x_i, y_i)$  con  $y_i = f(x_i)$ ,  $i = 1, \dots, n+1$ , e vincolata ( $S'_3(x_1) = f'(x_1)$  e  $S'_3(x_{n+1}) = f'(x_{n+1})$ ) nella forma:

$$S_3(x) = \sum_{j=1}^{n+1} f(x_j) S_{3,j}(x) + f'(x_1) S_{3,0}(x) + f'(x_{n+1}) S_{3,n+2}(x)$$

dove le  $n+3$  spline cubiche

$$S_{3,0}(x), S_{3,1}(x), \dots, S_{3,n+1}(x), S_{3,n+2}(x)$$

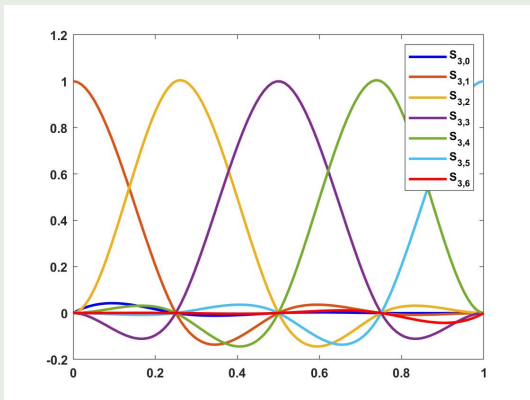
sono così definite per  $j = 0, 1, \dots, n+1, n+2$ :

$$\begin{cases} S_{3,0}(x_i) = 0 \\ S'_{3,0}(x_1) = 1 \\ S'_{3,0}(x_{n+1}) = 0 \end{cases} \quad \begin{cases} S_{3,j}(x_i) = \begin{cases} 1 & \text{se } j = i \\ 0 & \text{se } j \neq i \end{cases} \\ S'_{3,j}(x_1) = 0 \\ S'_{3,j}(x_{n+1}) = 0 \end{cases} \quad \begin{cases} S_{3,n+2}(x_i) = 0 \\ S'_{3,n+2}(x_1) = 0 \\ S'_{3,n+2}(x_{n+1}) = 1 \end{cases}$$

e rappresentano una base per lo spazio lineare delle spline cubiche e vincolate, associate alla partizione  $x_1 < \dots < x_{n+1}$ . Le spline  $S_{3,j}(x)$  dipendono unicamente dai nodi  $x_i$  e non dai particolari valori  $y_i$  assegnati.

## Esempio

In figura sono rappresentate graficamente le spline  $S_{3,j}(x)$ ,  $j = 0, 1, \dots, 6$ , associate alla partizione dell'intervallo  $[0, 1]$  in  $n = 4$  sottointervalli di uguale ampiezza.



Inoltre, è possibile riscrivere la generica spline interpolante i dati  $(x_i, y_i)$  con  $y_i = f(x_i)$ ,  $i = 1, \dots, n + 1$ , e naturale nella forma:

$$S_3(x) = \sum_{j=1}^{n+1} f(x_j) S_{3,j}(x)$$

dove le  $n + 1$  spline cubiche

$$S_{3,1}(x), \dots, S_{3,n+1}(x)$$

sono naturali e sono così definite per  $j = 1, \dots, n + 1$ :

$$\begin{cases} S_{3,j}(x_i) = \begin{cases} 1 & \text{se } j = i \\ 0 & \text{se } j \neq i \end{cases} \\ S''_{3,j}(x_1) = 0 \\ S''_{3,j}(x_{n+1}) = 0 \end{cases}$$

# Funzioni di base con supporto locale

Per le spline è possibile definire funzioni di base con supporto locale<sup>1</sup>.

Supponiamo  $D \equiv [a, b]$  e suddividiamo tale dominio in  $M$  parti  $a \equiv x_0 < x_1 < \dots < x_M \equiv b$ . Sia  $X_M$  l'insieme di tutte le poligonal  $S_1$  associate alla suddetta partizione.

Un modo alternativo per rappresentare una poligonale (spline lineare), di interesse nelle applicazioni, consiste nel rappresentare la generica funzione lineare a tratti e interpolante i dati  $(x_i, y_i)$ ,  $i = 0, \dots, M$ , nella seguente forma:

$$S_1(x) = \sum_{j=0}^M y_j N_j(x)$$

---

<sup>1</sup>Il supporto di una funzione  $f(x)$  definita in  $D$  è la chiusura dell'insieme  $\{x \in D : f(x) \neq 0\}$



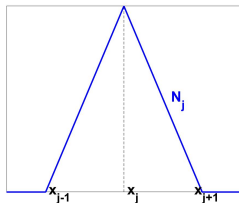
Le funzioni  $N_j(x)$  soddisfano la condizione

$$N_j(x_i) = \begin{cases} 1 & \text{se } j = i \\ 0 & \text{se } j \neq i \end{cases},$$

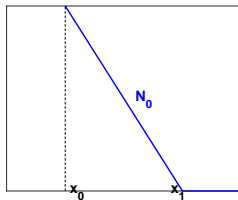
e rappresentano una base con supporto locale per lo spazio  $X_M$ .

Le funzioni  $\{N_j(x), j = 0, 1, \dots, M\}$  sono nulle quasi ovunque tranne in piccoli sottoinsiemi del dominio di definizione. Esse sono così definite:

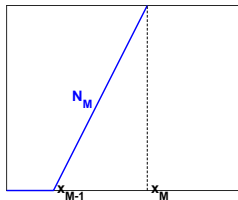
$$N_j(x) = \begin{cases} \frac{x - x_{j-1}}{x_j - x_{j-1}} & x_{j-1} \leq x \leq x_j \\ \frac{x_{j+1} - x}{x_{j+1} - x_j} & x_j \leq x \leq x_{j+1} \\ 0 & \text{altrimenti} \end{cases}$$



$$N_0(x) = \begin{cases} \frac{x_1 - x}{x_1 - x_0} & x_0 \leq x \leq x_1 \\ 0 & \text{altrimenti} \end{cases}$$

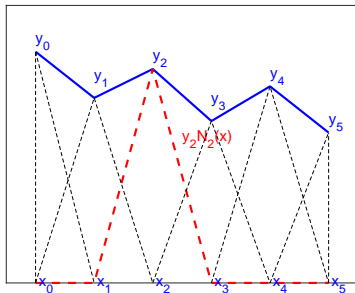


$$N_M(x) = \begin{cases} \frac{x - x_{M-1}}{x_M - x_{M-1}} & x_{M-1} \leq x \leq x_M \\ 0 & \text{altrimenti} \end{cases}$$



## Esempio

Il grafico della spline  $S_1(x)$  interpolante i dati  $(x_i, y_i)$ ,  $i = 0, \dots, 5$ , è il seguente



Analogamente è possibile riscrivere la generica spline cubica interpolante i dati  $(x_i, y_i)$  con  $y_i = f(x_i)$ ,  $i = 0, \dots, M$  come combinazione lineare di funzioni di base con supporto locale.

A tal scopo, introduciamo 6 nodi ausiliari:

$$x_{-3} < x_{-2} < x_{-1} < x_0 < x_1 < \dots < x_M < x_{M+1} < x_{M+2} < x_{M+3}$$

e definiamo  $n + 3$  funzioni spline cubiche  $B_{3,j}$  con  $j = -1, 0, \dots, M + 1$  che soddisfano le seguenti condizioni

$$\begin{cases} B_{3,j}(x) = 0, & \text{per } x \leq x_{j-2} \text{ e } x \geq x_{j+2} \\ B_{3,j}(x_j) = 1 \\ B'_{3,j}(x) = 0, \quad B''_{3,j}(x) = 0 & \text{per } x = x_{j-2} \text{ e } x = x_{j+2} \end{cases}$$

Tali funzioni esistono e sono univocamente determinate dalle condizioni sopra scritte, e rappresentano una base per lo spazio lineare delle spline cubiche associate ai nodi  $x_0 < x_1 < \dots < x_M$ . Tali funzioni vengono denominate **B-spline cubiche**, ovvero basis-spline o bell-spline per la loro caratteristica forma a campana.

## Esempio

In figura è rappresentata graficamente la B-spline  $B_{3,2}(x)$  associata alla partizione dell'intervallo  $[0, 1]$  in  $M = 4$  sottointervalli di uguale ampiezza. La B-spline  $B_{3,2}(x)$  è costituita da 4 tratti contigui di polinomi di grado 3, denominati  $p_{3,i}$ ,  $i = 1, \dots, 4$ .

