

Figura 3.3. Confronto tra la funzione $f(x) = 1/x$ (in linea piena) ed il suo polinomio di Taylor di grado 10 riferito al punto $x_0 = 1$ (in tratteggio). L'espressione del polinomio di Taylor è riportata in figura

l'andamento della funzione $f(x) = 1/x$ con quello del suo polinomio di Taylor di grado 10 costruito attorno al punto $x_0 = 1$. Questa figura mostra anche l'interfaccia grafica del programma MATLAB `taylor` che consente di calcolare il polinomio di Taylor di grado arbitrario di una data funzione f . Come si vede più ci si allontana da x_0 più il polinomio di Taylor si discosta dalla funzione. Per altre funzioni ciò fortunatamente non si verifica; è il caso ad esempio della funzione esponenziale per la quale il polinomio di Taylor relativo al punto $x_0 = 0$ rappresenta una buona approssimazione per ogni valore di $x \in \mathbb{R}$ purché il grado n sia sufficientemente grande.

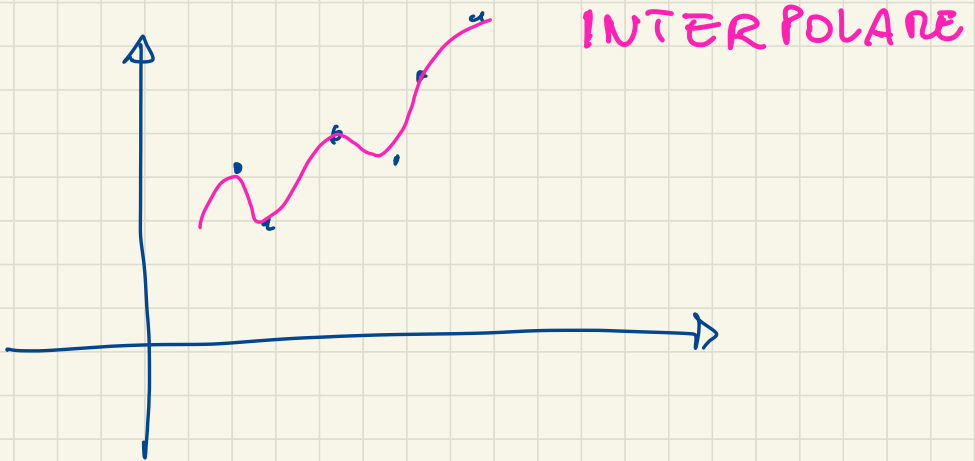
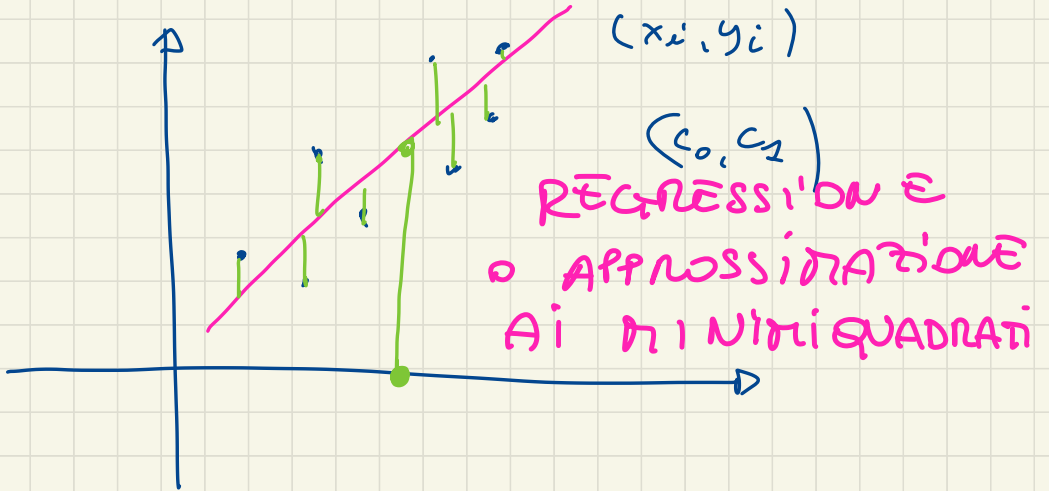
Servono quindi in generale dei metodi di approssimazione alternativi che illustreremo nei prossimi paragrafi.

Octave 3.1 `taylor` non è disponibile in Octave. ■

3.3 Interpolazione

Come abbiamo potuto notare dai problemi 3.1, 3.2 e 3.3, in molte applicazioni concrete si conosce una funzione solo attraverso i suoi valori in determinati punti. Supponiamo pertanto di conoscere $n + 1$ coppie di valori $\{x_i, y_i\}$, $i = 0, \dots, n$, dove i punti x_i , tutti distinti, sono detti *nodi*.

Ad esempio, con riferimento alla Tabella 3.1, n è uguale a 12, i nodi x_i sono i valori della latitudine riportati nella prima colonna, mentre



Dati :

$(x_i, y_i) \quad i = 0 \dots n$

$n+1$ punti

gli y_i sono i valori corrispondenti (della variazione di temperatura) che troviamo in una qualunque delle restanti colonne.

In tal caso, può apparire naturale richiedere che la funzione approssimante \tilde{f} soddisfi le seguenti uguaglianze

$$\tilde{f}(x_i) = y_i, \quad i = 0, 1, \dots, n \quad (3.1)$$

CONDIZIONI
di
INTERPOLAZIONE

Una tale funzione \tilde{f} è detta *interpolatore* dell'insieme di dati $\{y_i\}$ e le equazioni (3.1) sono le condizioni di interpolazione.

Si possono immaginare vari tipi di interpolatori, ad esempio:

- l'*interpolatore polinomiale*:

$$\tilde{f}(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n;$$

- l'*interpolatore trigonometrico*:

$$\tilde{f}(x) = a_{-M}e^{-iMx} + \dots + a_0 + \dots + a_Me^{iMx},$$

dove M è un intero pari a $n/2$ se n è pari, $(n+1)/2$ se n è dispari, e i è l'unità immaginaria;

- l'*interpolatore razionale*:

$$\tilde{f}(x) = \frac{a_0 + a_1x + \dots + a_kx^k}{a_{k+1} + a_{k+2}x + \dots + a_{k+n+1}x^n}.$$

Per semplicità considereremo soltanto quegli interpolatori che dipendono linearmente dai coefficienti incogniti a_i . Ad esempio, l'interpolazione polinomiale e quella trigonometrica rientrano in questa categoria, mentre quella razionale no.

3.3.1 Interpolazione polinomiale di Lagrange

Concentriamo la nostra attenzione sull'interpolazione polinomiale. Vale il seguente risultato:

Proposizione 3.1 Per ogni insieme di coppie $\{x_i, y_i\}$, $i = 0, \dots, n$, con i nodi x_i distinti fra loro, esiste un unico polinomio di grado minore od uguale a n , che indichiamo con Π_n e chiamiamo polinomio interpolatore dei valori y_i nei nodi x_i , tale che

$$\Pi_n(x_i) = y_i, \quad i = 0, \dots, n \quad (3.2)$$

Quando i valori $\{y_i, i = 0, \dots, n\}$, rappresentano i valori assunti da una funzione continua f (ovvero $y_i = f(x_i)$), Π_n è detto polinomio interpolatore di f (in breve, interpolatore di f) e viene indicato con $\Pi_n f$.

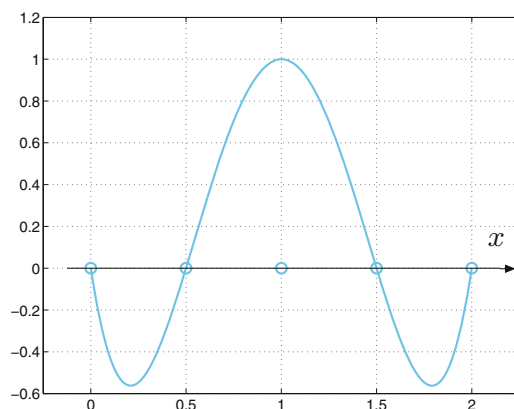


Figura 3.4. Il polinomio $\varphi_2 \in \mathbb{P}_4$ associato ai 5 nodi equispaziati in $[0, 2]$

Per verificare l'unicità procediamo per assurdo supponendo che esistano due polinomi distinti di grado n , Π_n e Π_n^* , che soddisfino entrambe le relazioni nodali (3.2). La loro differenza, $\Pi_n - \Pi_n^*$, sarà ancora un polinomio di grado n che si annulla in $n + 1$ punti distinti. Per un noto teorema dell'Algebra, esso deve essere identicamente nullo e, quindi, Π_n^* coincide con Π_n , da cui l'assurdo.

Per ottenere un'espressione di Π_n , iniziamo da una funzione molto speciale per la quale tutti gli y_i sono nulli fuorché quello per $i = k$ (per un dato k) per il quale $y_k = 1$. Posto allora $\varphi_k(x) = \Pi_n(x)$, si dovrà avere (si veda la Figura 3.4)

$$\varphi_k \in \mathbb{P}_n, \varphi_k(x_j) = \delta_{jk} = \begin{cases} 1 & \text{se } j = k, \\ 0 & \text{altrimenti,} \end{cases}$$

dove δ_{jk} è il simbolo di Kronecker.

Le funzioni φ_k possono essere scritte come

$$\varphi_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^n \frac{x - x_j}{x_k - x_j}, \quad k = 0, \dots, n. \quad (3.3)$$

Mettiamoci ora nel caso generale in cui $\{y_i, i = 0, \dots, n\}$ sia un insieme di valori arbitrari. Per il principio di sovrapposizione degli effetti abbiamo

$$\Pi_n(x) = \sum_{k=0}^n y_k \varphi_k(x) \quad (3.4)$$

In effetti, questo polinomio soddisfa le condizioni di interpolazione (3.2) in quanto

$$\Pi_n(x_i) = \sum_{k=0}^n y_k \varphi_k(x_i) = \sum_{k=0}^n y_k \delta_{ik} = y_i, \quad i = 0, \dots, n.$$

Funzioni di Lagrange

$$\varphi_k : \mathbb{R} \rightarrow \mathbb{R} \quad k = 0 \dots n$$

$$\varphi_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^n \frac{(x - x_j)}{(x_k - x_j)} \quad \leftarrow$$

φ_k polinomio di grado n

$$\varphi_k(x_i) = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases}$$

$$\begin{aligned} i=k & \frac{(x_k - x_0)(x_k - x_1) \dots (x_k - x_n)}{(x_k - x_0)(x_k - x_1) \dots (x_k - x_n)} = 1 \end{aligned}$$

$$\frac{(x_i - x_i)}{(x_i - x_0) \dots} = 0$$

$$\Pi_m(x_i) = y_i$$

$$\Pi_m(x) = c_0 \varphi_0(x) + c_1 \varphi_1(x) + \dots + c_n \varphi_n(x)$$

$$x = x_0$$

\Downarrow

polinomio di grado n

$$\rightarrow \Pi_m(x_0) = y_0 = c_0 \varphi_0(x_0) + c_1 \varphi_1(x_0) + \dots + c_n \varphi_n(x_0)$$

$$\Pi_m(x_1) = y_1 = c_0 \varphi_0(x_1) + c_1 \varphi_1(x_1) + \dots + c_n \varphi_n(x_1)$$

\vdots

$$\Pi_m(x_n) = y_n = c_0 \varphi_0(x_n) + c_1 \varphi_1(x_n) + \dots + c_n \varphi_n(x_n)$$

$$\Rightarrow c_0 = y_0$$

$$c_1 = y_1$$

\vdots

$$c_n = y_n$$

Polinomio di interpolazione

$$\begin{aligned} \Pi_n(x) &= y_0 \varphi_0(x) + y_1 \varphi_1(x) + \dots \\ &\quad \dots y_n \varphi_n(x) \end{aligned}$$

polinomio di grado n
unico

Non ci sono polinomi di
grado $> n$ che interpolano
 $n+1$ punti

Per il loro ruolo peculiare, le funzioni φ_k sono dette *polinomi caratteristici di Lagrange* e la (3.4) è nota come *forma di Lagrange* del polinomio interpolatore.

In MATLAB possiamo memorizzare le $n+1$ coppie $\{(x_i, y_i)\}$ in due vettori, ad esempio \mathbf{x} e \mathbf{y} , e con l'istruzione `c=polyfit(x,y,n)` possiamo generare i coefficienti del polinomio interpolatore. In particolare, $c(1)$ conterrà il coefficiente di x^n , $c(2)$ quello di x^{n-1} , ... e $c(n+1)$ il valore di $\Pi_n(0)$. (Maggiori dettagli su questo comando sono contenuti nel paragrafo 3.6.)

Come abbiamo visto nel Capitolo 1, noti i coefficienti, attraverso l'istruzione `p=polyval(c,z)` è poi possibile calcolare i valori $p(j)$ del polinomio interpolatore in m punti arbitrari $z(j)$, $j=1, \dots, m$.

Nel caso in cui $y_i = f(x_i)$ e l'espressione della funzione f sia nota in forma esplicita, possiamo memorizzare i nodi x_i nel vettore \mathbf{x} e costruire il vettore \mathbf{y} mediante l'istruzione `y=eval(f)`.

Esempio 3.1 (Climatologia) Calcoliamo il polinomio interpolatore di grado 4 per i dati del Problema 3.1 relativi ad una concentrazione X di acido carbonico pari a 0.67 (prima colonna), utilizzando i valori della temperatura corrispondenti alle sole latitudini 65, 35, 5, -25, -55. Possiamo utilizzare le seguenti istruzioni MATLAB

```
x=[-55 -25 5 35 65]; y=[-3.25 -3.2 -3.02 -3.32 -3.1];
format short e; c=polyfit(x,y,4)

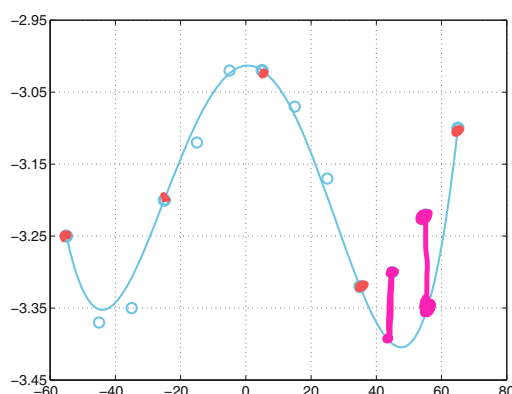
c =
    8.2819e-08   -4.5267e-07   -3.4684e-04    3.7757e-04   -3.0132e+00
```

Il grafico del polinomio interpolatore può allora essere generato come segue

```
p=polyval(c,z);
plot(z,p,x,y,'o'); grid on;
```

Si noti che al solo scopo di ottenere una rappresentazione “liscia” il polinomio è stato valutato in 101 punti equispaziati nell'intervallo $[-55, 65]$ (in effetti, quando MATLAB disegna una curva si limita a congiungere due punti consecutivi con un segmento). L'istruzione `x(end)` consente di accedere direttamente all'ultima componente del vettore \mathbf{x} , senza bisogno di conoscerne la lunghezza. In Figura 3.5 i cerchietti pieni corrispondono ai dati utilizzati per costruire il polinomio di interpolazione, mentre quelli vuoti corrispondono ai dati che non sono stati utilizzati. Si può apprezzare il buon accordo a livello qualitativo fra il polinomio interpolatore e la distribuzione dei dati. ■

Grazie al risultato seguente possiamo quantificare l'errore che si commette sostituendo ad una funzione f il suo polinomio interpolatore $\Pi_n f$.



$$(x_i, y_i) = f(x_i)$$

Figura 3.5. Il polinomio interpolatore di grado 4 introdotto nell'Esempio 3.1

Proposizione 3.2 Sia I un intervallo limitato, e si considerino $n + 1$ nodi di interpolazione distinti $\{x_i, i = 0, \dots, n\}$ in I . Sia $f \in C^{n+1}$ derivabile con continuità fino all'ordine $n + 1$ in I . Allora $\forall x \in I$ $\exists \xi \in I$ tale che

$$E_n f(x) = f(x) - \Pi_n f(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i) \quad (3.5)$$

Ovviamente, $E_n f(x_i) = 0$, per $i = 0, \dots, n$.

Il risultato (3.5) può essere meglio specificato nel caso di una distribuzione uniforme di nodi, ovvero quando $x_i = x_{i-1} + h$ per $i = 1, \dots, n$, per un dato $h > 0$ ed un dato x_0 . In tal caso, si veda l'Esercizio 3.1, $\forall x \in (x_0, x_n)$ si può verificare che

$$\left| \prod_{i=0}^n (x - x_i) \right| \leq n! \frac{h^{n+1}}{4}, \quad (3.6)$$

e quindi

$$\max_{x \in I} |E_n f(x)| \leq \frac{\max_{x \in I} |f^{(n+1)}(x)|}{4(n+1)!} h^{n+1}. \quad (3.7)$$

Purtroppo non si può dedurre dalla (3.7) che l'errore tende a 0 per $n \rightarrow \infty$, nonostante $h^{n+1}/[4(n+1)!]$ tenda a 0. Infatti, come mostra l'Esempio 3.2, esistono funzioni f per le quali addirittura tale limite può essere infinito, ovvero

$$\lim_{n \rightarrow \infty} \max_{x \in I} |E_n f(x)| = \infty.$$

Questo risultato indica che ad un aumento del grado n del polinomio interpolatore non corrisponde necessariamente un miglioramento nella ricostruzione di una funzione f . Ad esempio, se interpolassimo tutti i dati della seconda colonna della Tabella 3.1, troveremmo il polinomio Π_{12} , rappresentato in Figura 3.6 (a sinistra), il cui comportamento, nelle vicinanze dell'estremo sinistro dell'intervallo è assai meno soddisfacente di quello mostrato in Figura 3.5 utilizzando un numero inferiore di nodi. Si può riscontrare un comportamento ancor più insoddisfacente per particolari funzioni, come risulta dall'esempio seguente.

Esempio 3.2 (Runge) Se interpoliamo la funzione $f(x) = 1/(1+x^2)$ (detta di Runge) su un insieme di nodi equispaziati nell'intervallo $I = [-5, 5]$, l'errore $\max_{x \in I} |E_n f(x)|$ tende all'infinito quando $n \rightarrow \infty$. Questo è dovuto al fatto che per $n \rightarrow \infty$ l'ordine di infinito di $\max_{x \in I} |f^{(n+1)}(x)|$ supera quello di infinitesimo di $h^{n+1}/[4(n+1)]$. Possiamo verificare questa conclusione calcolando il massimo delle derivate di f fino all'ordine 21 con le seguenti istruzioni MATLAB:

```
syms x; n=20; f=1/(1+x^2); df=diff(f,1); cdf=char(df);
for i = 1:n+1, df = diff(df,1); cdfn = char(df);
    x = fzero(cdfn,0); M(i) = abs(eval(cdf)); cdf = cdfn;
end
```

I massimi dei valori assoluti delle funzioni $f^{(n)}$, $n = 1, \dots, 21$, sono stati memorizzati nel vettore M. Si noti che il comando `char` converte la variabile simbolica `df` in una stringa che possa poi essere valutata dalla funzione `fzero`. In particolare, i valori assoluti di $f^{(n)}$ per $n = 3, 9, 15, 21$ sono

```
format short e; M([3,9,15,21])
ans =
    4.6686e+00    3.2426e+05    1.2160e+12    4.8421e+19
```

mentre i corrispondenti valori assoluti di $\prod_{i=0}^n (x - x_i)/(n+1)!$ sono

```
z = linspace(-5,5,10000);
for n=0:20; h=10/(n+1); x=[-5:h:5];
    c=poly(x); r(n+1)=max(polyval(c,z));
    r(n+1)=r(n+1)/prod([1:n+1]);
end
r([3,9,15,21])
ans =
    1.1574e+01    5.1814e-02    1.3739e-05    4.7247e-10
```

dove `c=poly(x)` è un vettore i cui elementi sono i coefficienti del polinomio che ha come radici proprio gli elementi del vettore `x`. Ne consegue che $\max_{x \in I} |E_n f(x)|$ assume i seguenti valori

```
5.4034e+01    1.6801e+04    1.6706e+07    2.2877e+10
```

rispettivamente per $n = 3, 9, 15, 21$. La mancanza di convergenza si manifesta nelle forti oscillazioni, presenti nel grafico del polinomio interpolatore rispetto a quello di f , che tendono ad amplificarsi in prossimità degli estremi dell'intervallo (si veda la Figura 3.6 a destra). Questo comportamento è noto come *fenomeno di Runge*. ■

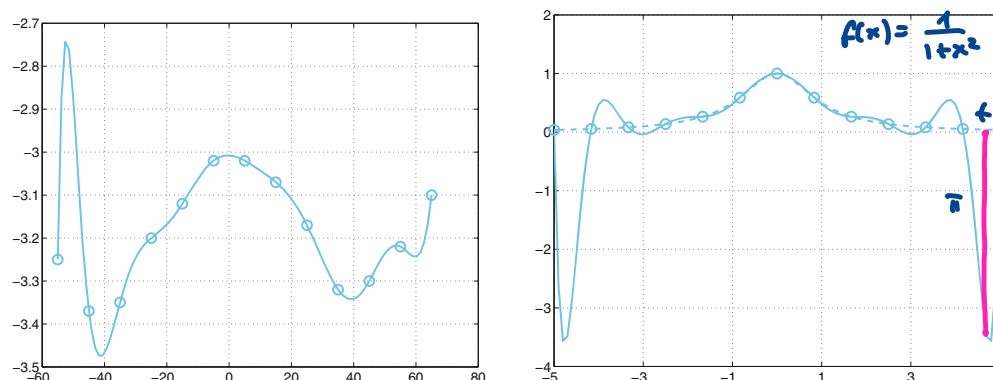


Figura 3.6. Due esemplificazioni del fenomeno di Runge: a sinistra, $\Pi_{12}f$ calcolato per l'insieme di dati della Tabella 3.1, colonna $K = 0.67$; a destra, $\Pi_{12}f$ (in linea continua) calcolato su 13 nodi equispaziati nel caso della funzione di Runge $f(x) = 1/(1+x^2)$ (in linea tratteggiata)

Oltre alla (3.7) si può anche dimostrare che vale la seguente disuguaglianza

$$\max_{x \in I} |f'(x) - (\Pi_n f)'(x)| \leq Ch^n \max_{x \in I} |f^{(n+1)}(x)|,$$

dove C è una costante indipendente da h . Quindi se approssimiamo la derivata prima di f con la derivata prima di $\Pi_n f$, dobbiamo aspettarci di perdere un ordine di convergenza rispetto a h . In MATLAB $(\Pi_n f)'$ può essere calcolato tramite il comando MATLAB `[d]=polyder(c)`, dove il parametro `c` di *input* è il vettore che memorizza i coefficienti del polinomio interpolatore, mentre `d` è il vettore dei coefficienti della sua derivata (si veda il paragrafo 1.5.2).

3.3.2 Stabilità dell'interpolazione polinomiale

Cosa succede al polinomio di interpolazione se, anziché partire da dati esatti $f(x_i)$ relativi ai nodi x_i , con $i = 0, \dots, n$, in un intervallo I , si considera una loro approssimazione, diciamo $\tilde{f}(x_i)$? La perturbazione $f(x_i) - \tilde{f}(x_i)$ potrebbe essere dovuta ad esempio all'effetto degli errori di arrotondamento oppure essere causata da un errore nella misurazione dei dati stessi.

Sia $\Pi_n \tilde{f}$ il polinomio interpolatore corrispondente ai valori $\tilde{f}(x_i)$. Indicando con \mathbf{x} il vettore le cui componenti sono i nodi di interpolazione $\{x_i\}$, si ha

$$\begin{aligned} \max_{x \in I} |\Pi_n f(x) - \Pi_n \tilde{f}(x)| &= \max_{x \in I} \left| \sum_{i=0}^n (f(x_i) - \tilde{f}(x_i)) \varphi_i(x) \right| \\ &\leq \Lambda_n(\mathbf{x}) \max_{0 \leq i \leq n} |f(x_i) - \tilde{f}(x_i)| \end{aligned} \quad (3.8)$$

dove

$$A_n(\mathbf{x}) = \max_{x \in I} \left| \sum_{i=0}^n \varphi_i(x) \right| \quad (3.9)$$

indica la cosiddetta *costante di Lebesgue* che dipende dai nodi di interpolazione. Di conseguenza, a piccole perturbazioni sui dati corrisponderanno piccole variazioni sul polinomio interpolatore purché la costante di Lebesgue sia piccola. Quest'ultima assume il significato di *numero di condizionamento* del problema dell'interpolazione. Nel caso dell'interpolazione polinomiale di Lagrange su nodi equispaziati, si trova

$$A_n(\mathbf{x}) \simeq \frac{2^{n+1}}{en(\log n + \gamma)},$$

dove $e \simeq 2.71834$ è il numero di Nepero e $\gamma \simeq 0.547721$ rappresenta la costante di Eulero (si veda [Hes98] e [Nat65]).

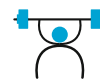
Ciò comporta che per n grande questo tipo di interpolazione potrebbe essere instabile, come si evince dal seguente esempio. (Si veda anche l'Esercizio 3.8.)

Esempio 3.3 Sull'intervallo $[-1, 1]$ interpoliamo la funzione $f(x) = \sin(2\pi x)$ su 22 nodi equispaziati x_i . Generiamo un insieme di valori $\tilde{f}(x_i)$ ottenuti perturbando in maniera casuale i valori $f(x_i)$, in modo che

$$\max_{i=0, \dots, 21} |f(x_i) - \tilde{f}(x_i)| \simeq 9.5 \cdot 10^{-4}.$$

In Figura 3.7 vengono confrontati i due polinomi di interpolazione $\Pi_{21}f$ e $\Pi_{21}\tilde{f}$: come si vede agli estremi dell'intervallo di interpolazione la differenza è molto più grande della perturbazione operata, essendo $\max_{x \in I} |\Pi_{21}f(x) - \Pi_{21}\tilde{f}(x)| \simeq 3.1342$. Si noti che in questo caso la costante di Lebesgue è molto grande, essendo $A_{21}(\mathbf{x}) \simeq 20574$. ■

Si vedano gli Esercizi 3.1-3.4.



3.3.3 Interpolazione rispetto ai nodi di Chebyshev

Il fenomeno di Runge può essere evitato utilizzando opportune distribuzioni di nodi. In particolare, su un arbitrario intervallo $[a, b]$ consideriamo i cosiddetti *nodì di Chebyshev-Gauss-Lobatto*

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} \hat{x}_i, \text{ dove } \hat{x}_i = -\cos(\pi i/n), \quad i = 0, \dots, n \quad (3.10)$$

Naturalmente $x_i = \hat{x}_i$, $i = 0, \dots, n$ quando $[a, b] = [-1, 1]$. Si può dimostrare che se f è una funzione continua e derivabile con continuità

In generale, se x_i sono scelti equidistanti in $[x_0, x_n]$

$$\max_{\Pi} E_n[f(x)] \xrightarrow{n \rightarrow \infty} \infty$$

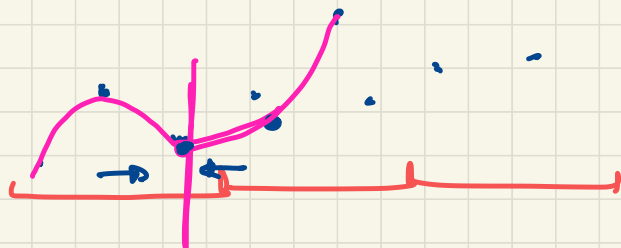
$$f(x) - \Pi_n(x)$$

. (funzione di Runge $\frac{1}{1+x^2}$
x $[-5, 5]$)

se i punti sono scelti come
zeri di Chebyshev, allora

$$E_n[f(x)] \xrightarrow{n \rightarrow \infty} 0$$

Interpolatione a tratti



$$p_s(x^*) = p_d(x^*)$$

$$p'_s(x^*) = p'_d(x^*)$$

$$p''_s(x^*) = p''_d(x^*)$$

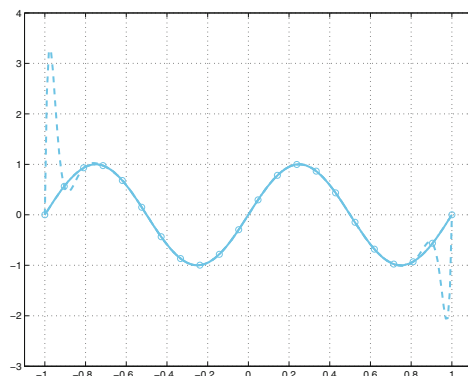


Figura 3.7. Effetti dell'instabilità nell'interpolazione semplice di Lagrange. In linea continua $\Pi_{21}f$, relativo ai dati imperturbati, in linea tratteggiata, $\Pi_{21}\tilde{f}$, relativo ai dati perturbati per l'Esempio 3.3

in $[a, b]$, il polinomio interpolatore $\Pi_n f$ associato a questa particolare distribuzione di nodi converge a f per $n \rightarrow \infty$, per ogni $x \in [a, b]$.

I nodi di Chebyshev-Gauss-Lobatto, che sono le ascisse di nodi equispaziati sulla semicirconferenza di raggio uno, appartengono all'intervallo $[a, b]$ e si addensano vicino agli estremi dell'intervallo (si veda la Figura 3.8 a destra).

Un'altra distribuzione di nodi sull'intervallo (a, b) , per la quale si hanno le stesse proprietà di convergenza, è data dai *nodi di Chebyshev-Gauss*

$$x_i = \frac{a+b}{2} - \frac{b-a}{2} \cos\left(\frac{2i+1}{n+1} \frac{\pi}{2}\right), \quad i = 0, \dots, n \quad (3.11)$$

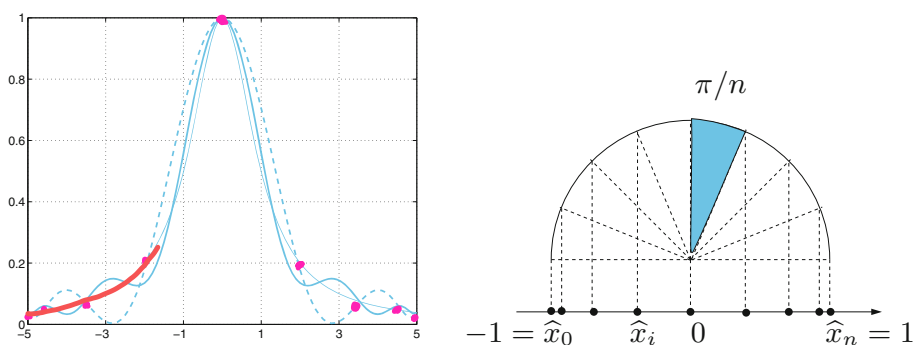


Figura 3.8. A sinistra, la funzione di Runge $f(x) = 1/(1+x^2)$ (in linea continua sottile) a confronto con i polinomi interpolatori sui nodi di Chebyshev-Gauss-Lobatto di grado 8 (linea tratteggiata) e 12 (linea continua più marcata). Si noti come ora, al crescere del grado, le oscillazioni si smorzino e l'approssimazione divenga sempre più accurata. A destra, riportiamo la distribuzione dei nodi di Chebyshev-Gauss-Lobatto nell'intervallo $[-1, 1]$

n	5	10	20	40
E_n	0.6386	0.1322	0.0177	0.0003

Tabella 3.3. L'errore di interpolazione per la funzione di Runge $f(x) = 1/(1+x^2)$ qualora si utilizzino i nodi di Chebyshev-Gauss-Lobatto (3.10)

Esempio 3.4 Riprendiamo la funzione di Runge ed interpoliamola nei nodi di Chebyshev-Gauss-Lobatto. Per generarli possiamo usare i seguenti comandi

```
xc = -cos(pi*[0:n]/n); x = (a+b)*0.5+(b-a)*xc*0.5;
```

dove $n+1$ è il numero di nodi, mentre a e b sono gli estremi dell'intervallo di interpolazione (nel nostro caso porremo $a=-5$ e $b=5$). Quindi, il polinomio interpolatore si genererà con le seguenti istruzioni

```
f = '1./(1+x.^2)'; y = eval(f); c = polyfit(x,y,n);
```

Valutiamo a questo punto il valore assoluto delle differenze fra f ed il suo polinomio interpolatore rispetto ai nodi di Chebyshev in 1000 punti equispaziati nell'intervallo $[-5, 5]$ e prendiamone il massimo

```
x = linspace(-5,5,1000); p=polyval(c,x);  
fx = eval(f); err = max(abs(p-fx));
```

Come si vede in Tabella 3.3, il massimo dell'errore decresce quando n cresce. ■

È interessante osservare che, qualora si considerino i nodi di Chebyshev-Gauss-Lobatto (3.10), la costante di Lebesgue si può maggiorare come segue ([Hes98])

$$A_n(\mathbf{x}) < \frac{2}{\pi} \left(\log n + \gamma + \log \frac{8}{\pi} \right) + \frac{\pi}{72 n^2}, \quad (3.12)$$

mentre qualora si considerino i nodi di Chebyshev-Gauss (3.11) si ha

$$A_n(\mathbf{x}) < \frac{2}{\pi} \left(\log(n+1) + \gamma + \log \frac{8}{\pi} \right) + \frac{\pi}{72(n+1)^2} \quad (3.13)$$

dove $\gamma \simeq 0.57721$ denota sempre la costante di Eulero.

Confrontando le maggiorazioni (3.12) e (3.13) con la stima (3.9) valida per nodi equispaziati, possiamo dedurre che l'interpolazione su nodi di Chebyshev è molto meno sensibile alla propagazione degli errori di arrotondamento di quanto non lo sia l'interpolazione su nodi equispaziati.

Esempio 3.5 Riprendiamo i dati dell'esempio 3.3 operando stavolta l'interpolazione sui nodi di Chebyshev (3.10) e (3.11). Partendo dalle stesse perturbazioni sui dati utilizzate per l'esempio 3.3 (inferiori a $9.5 \cdot 10^{-4}$), con $n=21$ otteniamo $\max_{x \in I} |\Pi_n f(x) - \Pi_n \tilde{f}(x)| \simeq 1.0977 \cdot 10^{-3}$ per i nodi (3.10) e $\max_{x \in I} |\Pi_n f(x) - \Pi_n \tilde{f}(x)| \simeq 1.1052 \cdot 10^{-3}$ per i nodi (3.11). Questo è in accordo con le stime (3.12) e (3.13) le quali, per $n=21$, fornirebbero rispettivamente $A_n(\mathbf{x}) \lesssim 2.9008$ e $A_n(\mathbf{x}) \lesssim 2.9304$. ■