

Interpolazione

Lucia Gastaldi

DICATAM - Sez. di Matematica,
<http://www.ing.unibs.it/gastaldi/>

Indice

- 1 Interpolazione
- 2 Interpolazione polinomiale
 - Polinomi
 - Valutazione di un polinomio
 - Algoritmo di Horner–Ruffini
 - Errore di approssimazione
 - Nodi di Chebyshev
- 3 Interpolazione a tratti
 - Interpolazione a tratti
 - Spline
 - Le funzioni MATLAB per l'interpolazione

Interpolazione

Problema Dati $n + 1$ punti $(x_i, y_i = f(x_i))$ per $i = 0, 1, \dots, n$ si cerca una **funzione approssimante** $\tilde{f} : \mathbb{R} \rightarrow \mathbb{R}$ tale che

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

La funzione \tilde{f} è detta **interpolatore** di f e le condizioni (1) sono dette **condizioni di interpolazione**.

Interpolazione polinomiale

$$\tilde{f}(x) = p(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x + a_n$$

Interpolazione razionale

$$\tilde{f}(x) = \frac{a_0x^m + a_1x^{m-1} + a_2x^{m-2} + \dots + a_{m-1}x + a_m}{b_0x^k + b_1x^{k-1} + b_2x^{k-2} + \dots + b_{k-1}x + b_k}$$

Interpolazione trigonometrica

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

dove i è l'unità immaginaria.

Polinomi

Un polinomio di grado n , con n intero non negativo, è una funzione del tipo

$$p(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x + a_n = \sum_{j=0}^n a_jx^{n-j}$$

dove $a_j \in \mathbb{R}$, per $j = 0, 1, 2, \dots, n$, sono i coefficienti del polinomio. Il polinomio è individuato dai coefficienti che devono essere memorizzati in un vettore.

Nota bene

In MATLAB i coefficienti devono essere **ordinati** a partire da quello corrispondente al termine di grado **più elevato** fino a quello di grado zero.

I coefficienti nulli vanno esplicitati.

Ad esempio al polinomio $p(x) = 1 - 2x + 4x^3$ si associa il vettore $p = [4 \ 0 \ -2 \ 1]$.

Algoritmo di Horner–Ruffini

L'algoritmo di Horner–Ruffini permette di calcolare il valore di un polinomio in un punto ad un costo computazionale inferiore rispetto all'uso della formula

$$p(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \cdots + a_{n-1}x + a_n = \sum_{i=0}^n a_i x^{n-i}$$

Consideriamo il polinomio

$$p(x) = 1 - 2x + 5x^2 + 4x^3;$$

questo si può scrivere anche nella forma seguente:

$$p(x) = ((4x + 5)x - 2)x + 1.$$

Contiamo le operazioni

- Nel primo caso: 6 (1+2+3) moltiplicazioni + 3 somme per ciascuna componente di x
- Nel secondo caso: 3 moltiplicazioni + 3 somme per ciascuna componente di x

Algoritmo di Horner–Ruffini

In generale il polinomio

$$p(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x + a_n$$

può essere scritto nella forma di Horner:

$$p(x) = (((((a_0x + a_1)x + a_2) \dots)x + a_{n-1})x + a_n.$$

Numero di operazioni

- Nel primo caso: $n(n+1)/2$ ($1 + 2 + \dots + n$) moltiplicazioni + n somme per ciascuna componente di x

Totale $\frac{n}{2}(n+3)N$ se N è il numero delle componenti di x

- Nel secondo caso: n moltiplicazioni + n somme per ciascuna componente di x

Totale $2nN$ se N è il numero delle componenti di x

polyval

La function `polyval` valuta il valore di un polinomio in una griglia di punti usando l'algorithmo di [Horner Ruffini](#).

$$y = \text{polyval}(p, z)$$

restituisce il vettore y contenente i valori di un polinomio di grado n calcolati nei punti z . Il vettore p di $n + 1$ componenti deve contenere i coefficienti del polinomio corrispondenti alle potenze in ordine decrescente.

Quindi per calcolare il valore del polinomio $p(x) = 1 + 2x - 4x^3$ nei punti z distribuiti in maniera equispaziata nell'intervallo $[a, b]$ si può usare la seguente sequenza di comandi:

```
>> z=linspace(-1,1,101);  
>> p=[-4 0 2 1];  
>> y=polyval(p,z);
```

Esercizi

Esercizio 1

Riportare in una stessa figura il grafico dei seguenti due polinomi:

$$p_1(x) = 1 - 3x - 4x^2 + 2x^5 \quad x \in [-3/2, 3/2]$$

$$p_2(x) = 2 + 3x - 2x^3 - 3x^4 \quad x \in [-3/2, 3/2]$$

Esercizio 2

Sia x il vettore che contiene i punti dell'intervallo $[0.995, 1.005]$ equispaziati a distanza 10^{-4} (usare `x=.995:1.e-4:1.005`). Fare il grafico del polinomio:

$$p(x) = x^6 - 6x^5 + 15x^4 - 20x^3 + 15x^2 - 6x + 1.$$

Confrontare il grafico ottenuto con quello della funzione $f(x) = (x - 1)^6$ nello stesso intervallo.

Esistenza ed unicità del polinomio interpolatore

Teorema

Per ogni insieme di punti (x_i, y_i) per $i = 0, 1, \dots, n$, con gli x_i distinti tra loro, esiste un unico polinomio di grado n , che indicheremo con Π_n , tale che

$$\Pi_n(x_i) = y_i \quad \text{per } i = 0, 1, \dots, n.$$

Esso viene detto **polinomio interpolatore** dei valori y_i nei **nodi** x_i .

Se per una opportuna funzione si ha $y_i = f(x_i)$ allora indichiamo con $\Pi_n f$ il polinomio interpolatore che **approssima** la funzione f .

polyfit

La function **polyfit** fornisce i coefficienti del polinomio interpolatore.

La sintassi di **polyfit** è:

```
p=polyfit(x,y,n)
```

dove x contiene i nodi x_i , y contiene i valori della funzione y_i e n è il **grado** del polinomio interpolatore.

Ad esempio:

```
>> x=0:4;  
>> y=[2 0 -1 -2 1];  
>> p=polyfit(x,y,4)
```

```
p = 0.2083    -1.4167     3.2917    -4.0833     2.0000
```

Per fare il grafico del polinomio interpolatore:

```
>> z=linspace(0,4,51);  
>> yp=polyval(p,z);  
>> plot((z,yp,x,y,'or'))
```

Esercizio 3

Calcolo dei coefficienti del polinomio interpolatore e sua rappresentazione

Si consideri la funzione $f(x) = (1 - x^2) \arctan(x) + e^x$ nell'intervallo $[-4, 4]$. Dati i nodi $x = [-3 \ -1 \ 2 \ 3]$; , usare il comando `polyfit` per trovare i coefficienti del polinomio interpolatore.

Riportare in una stessa figura il grafico del polinomio interpolatore e della funzione f .

Suggerimento

Per ottenere il grafico valutare il polinomio (usare il comando `polyval`) e la funzione in un numero appropriato di punti equispaziati nell'intervallo dato (usare il comando `linspace`).

Errore di approssimazione

Teorema: stima dell'errore di interpolazione

Dati $n + 1$ nodi di interpolazione x_i per $i = 0, 1, \dots, n$. Sia f una funzione derivabile con continuità $n + 1$ volte in un intervallo I contenente tutti i nodi di interpolazione e sia Π_n il polinomio interpolatore nei nodi x_i , allora per ogni $x \in I$, esiste un punto $\xi \in I$ tale che

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

Funzione di Runge

Si consideri la **funzione di Runge** $f(x) = \frac{1}{1+x^2}$, $x \in [-5, 5]$.

Esercizio 4

- Interpolare con polinomi di grado $n=2:2:12$, la funzione data, usando $n + 1$ punti equispaziati nell'intervallo $[-5, 5]$.
- Riportare il grafico di ciascun polinomio interpolatore insieme con quello della funzione data.
- Calcolare per ciascun valore di n l'errore commesso ossia

$$E_n = \max_{a \leq x \leq b} |f(x) - \Pi_n(x)|$$

Costruire un vettore contenente gli errori ottenuti per ciascun valore di n e riportare gli errori in un grafico in scala semilogaritmica `semilogy(n,E)`.

Traccia per la risoluzione dell'esercizio

1. Assegnare un vettore che contiene i valori di n .
2. Costruire il vettore z dei punti per valutare tutti i polinomi.
3. Valutare la funzione in z (risultato yy).
4. Per ogni valore di n (`for i=1:length(n)`) eseguire la seguente sequenza:
 - Costruire il vettore x dei nodi con il comando `x=linspace(a,b,n(i)+1)`.
 - Valutare la funzione nei nodi $y=f(x)$.
 - Trovare i coefficienti del polinomio con il comando `polyfit`.
 - Valutare il polinomio nei punti z con il comando `polyval` (risultato py).
 - Plottare la funzione e il polinomio di grado n (inserire una pausa `pause`).
 - Calcolare l'errore

$$E(i)=\text{norm}(yy-py,\text{inf})$$

5. Plottare l'errore con il comando `semilogy(n,E)`.

Esercizio 5

Si consideri la funzione $f(x) = e^x$ per $x \in [-5, 5]$.

- Costruire il polinomio interpolatore $\Pi_n f$ per $n = 2 : 2 : 12$ usando $n + 1$ punti equispaziati nell'intervallo $[-5, 5]$.
- Confrontare il grafico di ciascun polinomio interpolatore con quello della funzione data.
- Calcolare per ciascun valore di n l'errore commesso E_n e riportare l'errore su un grafico in scala semilogaritmica.

Interpolazione di Chebyshev

Il fenomeno di Runge può essere evitato utilizzando **opportune distribuzioni** di nodi.

Nell'intervallo $[a, b]$ consideriamo i nodi x_i dati da:

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} \hat{x}_i \quad \text{con } \hat{x}_i = -\cos\left(\frac{\pi i}{n}\right), \quad i = 0, \dots, n.$$

I punti $\hat{x}_i \in [-1, 1]$ si dicono **nod**i di Chebyshev.

Teorema di Bernstein

Sia $f : [a, b] \rightarrow \mathbb{R}$ una funzione di classe \mathbf{C}^1 . Sia Π_n il polinomio interpolatore di grado n costruito usando i nodi di Chebyshev.

Allora

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

Calcolo del polinomio con nodi di Chebyshev

Esercizio 6

Calcolare il polinomio di grado 7 che interpola la seguente funzione usando 8 nodi di Chebyshev

$$f(x) = x^2 + \frac{10}{\sin(x) + 1.2} \quad x \in [-2, 8].$$

Usare la seguente istruzione per costruire i nodi di Chebyshev:

```
xc=(a+b)/2-(b-a)/2*cos(pi*(0:n)/n)
```

Funzione di Runge

Esercizio 7

Per $n=2:2:12$, eseguire le seguenti operazioni:

- Interpolare la funzione di Runge nell'intervallo $[-5, 5]$ con i polinomi di grado n , costruiti usando $n + 1$ nodi di Chebyshev e $n + 1$ punti equidistanti nell'intervallo $[-5, 5]$.
- Usando il comando `subplot` riportare 4 grafici contenenti rispettivamente:
 - la funzione;
 - la funzione e il polinomio interpolatore con nodi equispaziati;
 - la funzione e il polinomio interpolatore con nodi di Chebyshev;
 - la funzione e i due polinomi interpolatori.

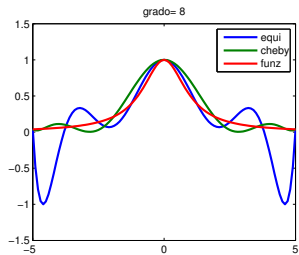
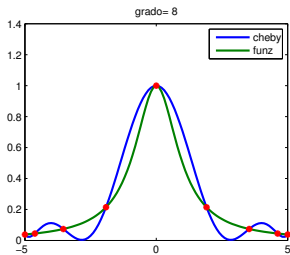
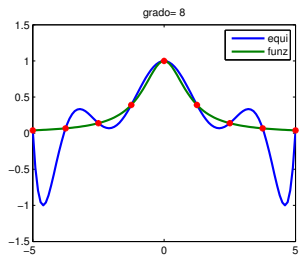
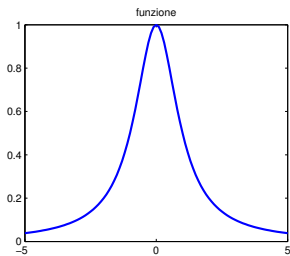
Si veda la figura nella pagina seguente.

- Calcolare per ciascun valore di n l'errore commesso.

Riportare gli errori per i due polinomi in uno stesso grafico in scala semilogaritmica `semilogy(n,E1,n,E2)`.

Ripetere l'esercizio usando la funzione $f(x) = \sin(x)$ per $x \in [0, 2\pi]$.

Figura



Interpolazione a tratti

Dato un intervallo $I = [a, b]$, si introduce una **partizione** mediante un **numero finito** di punti

$$a = x_0 < x_1 < \cdots < x_m = b;$$

$I_k = [x_{k-1}, x_k]$, $k = 1, \dots, m$ indica il k -esimo sottointervallo.

Definizione

Si definisce **polinomio a tratti** una funzione $g : [a, b] \rightarrow \mathbb{R}$ tale che

$$g(x) = p_n(x) \quad \forall x \in I_k,$$

essendo $p_n(x)$ un polinomio di grado n .

Interpolazione lineare a tratti

Sia $f : [a, b] \rightarrow \mathbb{R}$ una funzione sufficientemente regolare.

Problema

costruire un polinomio lineare a tratti che interpoli la funzione f nei nodi x_i , $i = 0, \dots, n$.

Consideriamo la partizione dell'intervallo $[x_0, x_n]$ data dai nodi x_i . Quindi su ciascun intervallino $[x_{i-1}, x_i]$ $i = 1, \dots, n$ il polinomio interpolatore a tratti è

$$g(x) = f(x_{i-1}) \frac{x - x_i}{x_{i-1} - x_i} + f(x_i) \frac{x - x_{i-1}}{x_i - x_{i-1}}$$

Stima dell'errore di approssimazione

Sia $H = \max_{1 \leq i \leq n} (x_i - x_{i-1})$. Sia f una funzione continua insieme alle sue derivate prima e seconda. Sia g il polinomio lineare a tratti definito prima.

Per ogni $i = 1, \dots, n$ esiste un punto $\eta_i \in [x_{i-1}, x_i]$ tale che

$$f(x) - g(x) = \frac{f''(\eta_i)}{2} (x - x_{i-1})(x - x_i) \quad \text{per } x \in [x_{i-1}, x_i],$$

da cui segue la seguente maggiorazione:

$$\begin{aligned} & \max_{1 \leq i \leq n} \max_{x_{i-1} \leq x \leq x_i} |f(x) - g(x)| \\ & \leq \max_{1 \leq i \leq n} \frac{(x_i - x_{i-1})^2}{2} \max_{x_{i-1} \leq x \leq x_i} |f''(x)| \leq \frac{H^2}{8} \max_{a \leq x \leq b} |f''(x)|. \end{aligned}$$

Spline

Siano x_i , per $i = 0, \dots, n, n+1$ nodi distinti e ordinati sull'intervallo $[a, b]$, tali che $a = x_0 < x_1 < \dots < x_n = b$.

Definizione

La funzione $s_m : [a, b] \rightarrow \mathbb{R}$ è una **funzione spline** di grado m relativa ai nodi x_i se

- $s_m(x)$ per $x \in [x_{i-1}, x_i]$, $i = 1, \dots, n$, è un polinomio di grado m .
- La funzione s_m è continua sull'intervallo $[a, b]$ insieme alle sue derivate fino all'ordine $m - 1$.

La spline più usata è la **spline cubica** s_3 .

Stime dell'errore

$$\max_{x_0 \leq x \leq x_n} |f^{(r)}(x) - s_3^{(r)}(x)| \leq C_r H^{4-r} \max_{x_0 \leq x \leq x_n} |f^{(4)}(x)| \quad r = 0, 1, 2$$

$$\max_{x_0 \leq x \leq x_n, x \neq \{x_0, \dots, x_n\}} |f^{(3)}(x) - s_3^{(3)}(x)| \leq C_3 H \max_{x_0 \leq x \leq x_n} |f^{(4)}(x)|$$

Le funzioni MATLAB per l'interpolazione

Funzione	Significato
interp1	Interpolazione 1D.
interp2	Interpolazione 2D.
interp3	Interpolazione 3D.
spline	Spline cubica interpolante.
pchip	Interpolazione cubica "shape preserving"
interpft	Interpolazione mediante il metodo FFT.

interp1 e spline

```
yi=interp1(x,y,z,metodo)
```

`x`, `y` specificano le coordinate dei punti di interpolazione.
`z` sono i punti in cui si vuole valutare il valore interpolato.
`metodo` è una stringa di caratteri che specifica il metodo da utilizzare:

- `metodo='nearest'` si sceglie il valore nel nodo di interpolazione più vicino;
- `metodo='linear'` interpolazione lineare a tratti;
- `metodo='spline'` interpolazione con spline cubica;
- `metodo='pchip'` o `metodo='cubic'` interpolazione di Hermite cubica a tratti shape preserving.

```
s=spline(x,y,z)
```

valuta nei punti `z`, la spline cubica che passa per i punti di ascissa `x` e ordinata `y`.

Confronto fra i diversi metodi di interpolazione

Considerare i seguenti punti:

```
x=1:6;
```

```
y=[16 18 21 17 15 12];.
```

- Usare l'interpolazione polinomiale e tutti i metodi disponibili nella function `interp1` per interpolare i punti dati.
- Riportare separatamente i grafici delle funzioni ottenute insieme ai nodi marcati con un pallino. Usare il comando `subplot` per avere tutti i grafici in una stessa finestra.
- Riportare in una stessa figura il grafico ottenuto con le spline e con il metodo `pchip` (shape preserving piecewise cubic).

Esercizio

Esercizio 8

- Calcolare l'approssimazione spline e lineare a tratti relative alla funzione di Runge nell'intervallo $[-1, 1]$, usando $n + 1$ punti equidistribuiti nell'intervallo dato essendo `n=2:2:20`. Usare le function `spline` e `interp1`.
- Calcolare per ciascun valore di n l'errore commesso ossia

$$E_n = \max_{a \leq x \leq b} |f(x) - \Pi_n(x)|$$

Costruire un vettore contenente gli errori ottenuti per ciascun valore di n e riportare gli errori in un grafico in scala bilogaritmica `loglog(n,E)`.

Esercizio 9

- Ripetere l'esercizio 8 approssimando la funzione di Runge nell'intervallo $[-5, 5]$.
- Sempre sull'intervallo $[-5, 5]$ usare n punti invece che $n + 1$.

Shape preserving piecewise cubic

Le spline cubiche non conservano le proprietà di monotonia delle funzioni.

Esempio

Per approssimare la semicirconferenza di centro l'origine e raggio 1 costruire i punti di coordinate:

$$x_k = -\cos(k\pi/6), \quad y_k = \sin(k\pi/6) \quad k = 0, 6.$$

- Calcolare la spline che interpola tali punti, in un campionamento di punti sufficientemente grande dell'intervallo $[-1, 1]$ (usare il comando `spline`).
- Riportare su una stessa figura la spline e la semicirconferenza.
- Osservato che la spline è oscillante intorno alla circonferenza, usare il comando `pchip` per generare un interpolante che conserva le proprietà di monotonia della funzione.

La function spline

Sono dati i vettori x, y contenenti le coordinate dei punti.
`pp=spline(x,y)` fornisce la struttura `pp` da cui si possono estrarre le informazioni relative alla spline.

```
[breaks,coefs] = unmkpp(pp)
```

- `breaks` punti di suddivisione (vettore x di partenza);
- `coefs` coefficienti.

`pp = mkpp(breaks,coefs)` costruisce un polinomio a tratti.

`v = ppval(pp,z)` valuta il polinomio individuato dalla struttura `pp` nei punti z .

Approssimazione delle derivate

Si consideri la funzione di Runge $f(x) = 1/(1+x^2)$ per $x \in [-5, 5]$.
Suddividere l'intervallo in n parti e costruire la spline che l'approssima.

Rappresentare la funzione e la spline in uno stesso grafico.

Determinare i coefficienti corrispondenti alle derivate fino all'ordine 3 e rappresentarle insieme alla corrispondente derivata della funzione di partenza.

Grafico della mano

Fare il grafico della propria mano.

- Disegnare su un foglio il contorno della propria mano;
- Dare i seguenti comandi:

```
figure('position',get(0,'screensize'))  
axes('position',[0 0 1 1])  
[x,y]=ginput;
```

- Cliccare su una ventina di punti sul contorno della mano, alla fine dare **INVIO**;
- I vettori **x** e **y** sono i valori di due funzioni di una variabile indipendente **t**. Interpolare entrambe le funzioni mediante le spline:

```
n=length(x);  
s=(1:n)';  
t=(1:0.05:n);  
u=spline(s,x,t);  
v=spline(s,y,t);  
clf  
plot(x,y,'.',u,v,'-')
```