Interpolazione polinomiale in Matlab per Ingegneria dell'Energia Laboratorio [1]

A. Sommariva[2]

Abstract

Interpolazione polinomiale, esempi.

Ultima revisione: 14 dicembre 2018

1. Interpolazione polinomiale

Sia $\mathbb{P}_n$ lo spazio dei polinomi algebrici di grado al più n, ovvero del tipo

$$q_n(x) = a_1 x^n + a_2 x^{n-1} + \ldots + a_n x + a_{n+1}.$$

Il problema dell'interpolazione polinomiale consiste nel determinare il polinomio algebrico $p_n \in \mathbb{P}_n$, tale che assegnate le $n+1$ coppie $(x_0,\, y_0)$, $(x_1,\, y_1)$,. . ., $(x_n,\, y_n)$, con $x_j \neq x_k$ se $j \neq k$, si abbia

$$p_n(x_j) = y_j, \;\; j = 0, \ldots, n.$$

Si dimostra che tale polinomio algebrico p_n esiste ed è unico. In particolare

$$p_n(x) = \sum_{i=0}^{n} y_i L_i(x)$$

dove L_i sono i polinomi di Lagrange

$$L_i(x) = \frac{\prod_{j=0, j\neq i}^{n} (x - x_j)}{\prod_{j=0, j\neq i}^{n} (x_i - x_j)}$$

Supponiamo di dover interpolare le coppie (x_k, y_k), $k = 0, \ldots, n$ e che sia

$$x = [x_0, \ldots, x_n], y = [y_0, \ldots, y_n];$$

I coefficienti del polinomio interpolatore sono ottenibili dal comando `polyfit`. A tal proposito l'help di Matlab suggerisce:

```
>> help polyfit
 polyfit Fit polynomial to data.
    P = polyfit(X,Y,N) finds the coefficients of a ...
     polynomial P(X) of
    degree N that fits the data Y best in a least-...
     squares sense. P is a
    row vector of length N+1 containing the polynomial ...
     coefficients in
    descending powers, P(1)*X^N + P(2)*X^(N-1) +...+ P(N...
     )*X + P(N+1).

   ... ...

    Reference page for polyfit
    Other functions named polyfit
```

Per capire qualcosa in più eseguiamo il seguente codice

```
>> x=[-2 1 3];
>> y=[-2 11 17];
>> a=polyfit(x,y,2)
a =
    -0.2667     4.0667     7.2000
>>
```

In effetti, calcolando manualmente il polinomio interpolatore si ha, semplificando quando ottenuto coi polinomi di Lagrange che è

$$\begin{aligned} p_2(x) &= (-4/15)\cdot x^2 + (61/15)\cdot x + (36/5) \\ &\approx 0.2\bar{6}x^2 + 4.0\bar{6}x + 7.2. \end{aligned}$$

Quindi, se $a = (a_k)_{k=1,\ldots,3}$, abbiamo $p_2(x) = a_1\, x^2 + a_2\, x + a_3$ e più in generale, se p_n è il polinomio interpolatore di grado n, e $a = (a_k)$ è il vettore ottenuto utilizzando `polyfit`, allora

$$p_n(x) = a_1\, x^n + a_2\, x^{n-1} + \ldots + a_{n+1}.$$

Per valutare in un vettore di ascisse $\mathbf{X} = [X_k]_{k=1,\ldots,m}$ un polinomio

$$p_n(x) = a_1\, x^n + a_2\, x^{n-1} + \ldots + a_{n+1}$$

i cui coefficienti sono quelli di in un vettore $\mathbf{P} = [a_k]_{k=1,\ldots,n+1}$ usiamo il comando `polyval`. Dall'help:

```
>> help polyval
To get started, select "MATLAB Help" from the
Help menu.
POLYVAL Evaluate polynomial.
Y = POLYVAL(P,X), when P is a vector of length
N+1 whose elements are the coefficients of a
polynomial, is the value of the polynomial
evaluated at X.
...
Y=P(1)*X^N+P(2)*X^(N-1)+...+P(N)*X+P(N+1)
>>
```

Dati i vettori

$$\mathbf{x} = [x_k]_{k=1,\ldots,m}, \;\; \mathbf{y} = [y_k]_{k=1,\ldots,m}$$

sia $p_{m-1}(x_k) = y_k$, per $k = 1, \ldots, m,$.

Sia $\mathbf{s} = [s_k]_{k=1,\ldots,M}$ un vettore di ascisse e desideriamo calcolare $\mathbf{t} = [t_k]_{k=1,\ldots,M}$ per cui $t_k = p_{n-1}(s_k)$, per ogni k.

Un tal proposito può essere raggiunto mediante la funzione:

```
function t=interpol(x,y,s)

m=length(x)-1;
coeff=polyfit(x,y,m);
t=polyval(coeff,s);
```

Quindi, siamo in grado tramite `polyfit` di determinare i coefficienti del polinomio interpolatore e tramite `interpol` di valutarlo in ascisse arbitrarie.

2. L'esempio di Runge

Interpoliamo la funzione di Runge

$$f(x) = \frac{1}{1+x^2}, \quad x \in [-5, 5]$$

sia su nodi equispaziati che di tipo Gauss-Chebyshev-Lobatto, scalati nell'intervallo ovvero

$$x_k = \frac{(a+b)}{2} + \frac{(b-a)}{2} t_k, \quad k = 0, \ldots, n \qquad (1)$$

con $a = -5$, $b = 5$ Non è difficile verificare che $f \in C^\infty([-5, 5])$. A tal proposito definiamo la funzione di Runge

```
function [fx]=runge(x)

% input
% x: vettore di ascisse

% output
% y: vettore la cui "k"-sima componente e' il
%   valore assunto dalla funzione di Runge "f(s)=1/(1+s...
    ^2)";
%   nella "k"-sima componente di "x".

fx=1./(x.^2+1);
```

e una funzione `gcl` che genera n nodi di Gauss-Chebyshev-Lobatto nell'intervallo $[a, b]$:

```
function xc=gcl(a,b,n)

% input:
% a,b: estremi dell'intervallo
% n  : numero dei nodi di Chebyshev-Lobatto.
%
% output:
% xc  : vettore di nodi di Chebyshev-Lobatto.

m=1:1:n;
xc=(a+b)/2-((b-a)/2)*cos(pi*(m-1)/(n-1));
```

Si osservi che essendo m un vettore di lunghezza n, necessariamente `xc` mye pure un vettore di lunghezza n. Quindi scriviamo il file `esperimento.m`

```
function esperimento

% oggetto:
% esempio di Runge per grado fissato "n", in nodi ...
    equispaziati e di
% Chebyshev-Lobatto.

% grado interpolante.
n=12;
% nodi test
s=-5:10/1000:5;

% ---- interpolazione nodi equispaziati ----
x=-5:10/n:5; y=runge(x);
t=interpol(x,y,s);

% ---- interpolazione nodi GCL ----
xgcl=gcl(-5,5,n+1); ygcl=runge(xgcl);
tt =interpol(xgcl, ygcl,s);

% ---- plot runge vs interpolanti ----

% prima figura
fs=runge(s);
clf;
figure(1); % prima figura (nodi equi.)
plot(s,fs,s,t,'LineWidth',2);
hold on;
title('Errori di interpolazione');
legend('funzione runge','intp. nodi eqsp.');
hold off;

% seconda figura
figure(2); % seconda figura (nodi GCL)
plot(s,fs,s,tt,'LineWidth',2);
hold on;
legend('funzione runge','intp. nodi GCL');
title('Errori di interpolazione');
hold off;

% valutazione errori assoluti
fs=runge(s); % valutazione funzione di Runge, punti test
ee=max(abs(fs-t)); % nodi equispaziati
ec=max(abs(fs-tt));  % nodi GCL
fprintf('\n \t[Errori interpolazione][E]:%1.2e [GCL...
    ]:%1.2e',ee,ec);
fprintf('\n \n');
```

3. Commento a `esperimento`

Nell'esperimento, detta f la funzione di Runge, abbiamo

- determinato un vettore s di 1001 ascisse equispaziate in $[-5, 5]$, ovvero
$$s_k = -5 + (k-1)h, \quad k = 1, \ldots, 1001, \quad h = 1/1000;$$
- determinato un vettore x di 13 ascisse equispaziate in $[-5, 5]$, ovvero
$$x_k = -5 + (k-1)h, \quad k = 1, \ldots, 13, \quad h = 1/12;$$
- determinato un vettore y di 13 ordinate in cui $y_k = f(x_k)$, $k = 1, \ldots, 13$;
- valutato nei punti s_i, $i = 1, \ldots, 1001$ il polinomio $p_{12}^{(E)} \in \mathbb{P}_12$ che interpola le coppie (x_k, y_k), $k = 1, \ldots, 13$ e posto $t_i = p_1^{(E)}2(s_i)$, $i = 1, \ldots, 1001$
- determinato un vettore `xgcl` di 13 ascisse, diciamo $x_k^{(GCL)}$ corrispondenti ai nodi di Gauss-Chebysheb-Lobatto;

- determinato un vettore `ygcl` di 13ordinate, diciamo $y_k^{(GCL)}$ in cui

$$y_k^{(GCL)} = f(x_k^{(GCL)}),\ \ k = 1,\dots,13$$

- valutato nei punti s_i, $i = 1,\dots,1001$, il polinomio $p_{12}^{(GCL)} \in \mathbb{P}_12$ che interpola le coppie $(x_k^{(GCL)}, y_k^{(GCL)})$, $k = 1,\dots,13$ e posto $tt_i = p_{12}^{(GCL)}(s_i)$, $i = 1,\dots,1001$,

- disegnato in un due figure, utilizzando rispettivamente i comandi `figure(1)` e `figure(2)`), e con legende basate rispettivamente sulle stringhe

  1. `'funzione runge','intp. nodi eqsp.',`
  2. `'funzione runge','intp. nodi GCL.',`

  e le figure abbiano spessore della linea specificato da

  `'LineWidth',2`

  e in entrambi i casi quale titolo

  `'Errori di interpolazione'`

- calcolato il valore

$$\max_{k=1,\dots,1001} |f(s_k) - p_{12}^{(E)}(s_k)|$$

  e assegnato alla variabile `ee`;

- calcolato il valore

$$\max_{k=1,\dots,1001} |f(s_k) - p_{12}^{(GCL)}(s_k)|$$

  e assegnato alla variabile `ec`;

- espresso in command-window i valori `ee` e `ec` con una cifra prima della virgola e 2 dopo, in notazione esponenziale.

- andato a capo due volte per abbellimento.

Se lo eseguiamo dalla command-window:

```
>> esperimento
  [ERR.][EQS]:3.66e+00 [GCL]:8.44e-02
>>
```

e il grafico
Al variare di n:

- Otteniamo la tabella degli errori vista in precedenza.

- Notiamo che la scelta di n non può essere eccessiva. Provare $n = 30$.

- Risulta evidente che non sussiste la convergenza puntuale al crescere di n, di $p_n^{(E}$ alla funzione di Runge $f(x) = 1/(1+x^2)$, qualora si utilizzino nodi equispaziati.

- Risulta evidente che sussiste la convergenza puntuale al crescere di n, di $p_n^{(GCL}$ alla funzione di Runge $f(x) = 1/(1+x^2)$, qualora si utilizzino nodi di Gauss-Chebyshev-Lobatto.

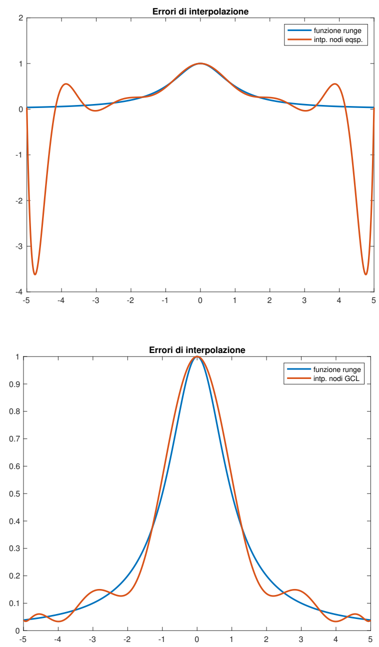

Figura 1: Grafici della funzione di Runge e delle interpolanti di grado 12, in nodi equispaziati e di Gauss-Chebyshev-Lobatto Dai grafici e dai risultati numerici si capisce la povera performance di $p_{12}^{(E)}$ in nodi equispaziati, soprattutto in virtù delle forti oscillazioni vicino agli estremi, e la buona qualità dell'interpolante $p_{12}^{(GCL)}$ nei nodi di Gauss-Chebyshev-Lobatto.

4. Esercizi

4.1. Esercizio `test_runge`

Prendendo come esempio il file `esperimento`, lo si modifichi nel file `test_runge` cosicchè :

1. abbia come input la variabile n, grado dell'interpolante p_n che non sia necessariamente 12;
2. abbia come output le variabili `ee`, `ec`, ovvero approssimanti $\max_{x\in[-5,5]} |f(x) - p_n^{(E)}(x)|$, $\max_{x\in[-5,5]} |f(x) - p_n^{(GCL)}(x)|$, con $p_n^{(E)}$, $p_n^{(GCL)}$ le interpolanti polinomiali di grado n della funzione di Runge f rispettivamente in $n+1$ nodi equispaziati e di Gauss-Chebyshev-Lobatto;
3. esegua il test dell'interpolazione in

$$s_k = -5 + (k-1)h,\ \ k = 1,\dots,10001, h = \frac{1}{10000}$$

4. non contenga grafici;
5. non contenga statistiche fornite all'utente;
6. abbia la seguente intestazione

```
  % Oggetto:
% Sia "f" la funzione di Runge e con "p^(E)_n",
% "p^(GCL)_n" le interpolanti polinomiali della
% funzione di Runge "f" rispettivamente in "n+1"
% nodi equispaziati e di Gauss-Chebyshev-Lobatto.
% Si approssimano
% ee=max_{x in [-5,5]} |f(x)-p^(E)_n(x)|
% ec=max_{x in [-5,5]} |f(x)-p^(GCL)_n(x)|
%
% Input:
```

```
% n: grado delle interpolanti
%
% Output:
% ee: max_{x in [-5,5]} |f(x)-p^(E)_n(x)|
% ec: max_{x in [-5,5]} |f(x)-p^(GCL)_n(x)|
```

5. Esercizio `demo_runge1`

Utilizzando la funzione `test_runge`, si definisca una funzione `demo_runge1` che

1. non abbia variabili di input, né di output;
2. calcoli il valore assunto dalle variabili `ee` e `ec`, definendo i vettori `eev`, `ecv`, aventi lunghezza 13, tali che
   - la n sima componente di `eev` corrisponda al valore `ee` fornito tramite `test_runge` per tale n,
   - la n sima componente di `ecv` corrisponda al valore `ec` fornito tramite `test_runge` per tale n,
3. esegua in una due figure (prima del primo plot si utilizzi il comando `figure(1)` e prima del secondo plot si si utilizzi il comando `figure(2)`) i grafici in scala semilogaritmica sia delle coppie (n,`eev`(n)) che delle coppie (n,`ecv`(n)),
4. utilizzi quale titolo della prima figura la stringa

   ```
   'Errori di interpolazione:  nodi equispaziati'
   ```

   ed il plot abbia la preferenza `'LineWidth',2`;
5. utilizzi quale titolo della seconda figura la stringa

   ```
   'Errori di interpolazione:  nodi GCL'
   ```

   ed il plot abbia la preferenza `'LineWidth',2`;
6. salvi su un file `errori_interpolazione.txt`, i valori di n utilizzati, gli errori `eev`, `ecv`, cosicchè la tabella risultante abbia alla k-sima riga,
   - l'indice k con 2 cifre prima della virgola e nessuna dopo la virgola, in notazione decimale,
   - l'errore `eev(k)`, ovvero la k-sima componente del vettore `eev`, con 1 cifra prima della virgola, una dopo la virgola, in notazione esponenziale,
   - l'errore `ecv(k)`, ovvero la k-sima componente del vettore `ecv`, con 1 cifra prima della virgola, una dopo la virgola, in notazione esponenziale.

6. Esercizio `demo_runge2`

Prendendo come esempio il file `esperimento`, lo si modifichi nel file `demo_runge2` cosicchè :

- invece di eseguire il grafico della funzione e delle sue interpolanti polinomiali di grado 12, ovvero $p_{12}^{(E)}$, $p_{12}^{(GCL)}$, valuti le funzioni

  $$|f(x) - p_{12}^{(E)}(x)|$$

  $$|f(x) - p_{12}^{(GCL)}(x)|$$

  nei punti

  $$s_k = -5+(k-1)h, \;\; k = 1, \ldots, 1000001, h = \frac{1}{1000000}$$

  e ne disegni in due figure separate, in scala semilogaritmica.
- la prima figura abbia titolo

  ```
  Errori di interpolazione:  nodi equi.
  ```

- la seconda figura abbia titolo

  ```
  Errori di interpolazione:  nodi GCL
  ```

- non si salvino risultati su testo.