Interpolazione polinomiale in Matlab

Alvise Sommariva

Università degli Studi di Padova
Dipartimento di Matematica Pura e Applicata

25 aprile 2023

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

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

Problema. (interpolazione polinomiale)

Il problema dell'interpolazione polinomiale consiste nel determinare il polinomio algebrico $p_n \in \mathbb{P}_n$, tale che assegnate le $n+1$ coppie

$$(x_1, y_1), (x_2, y_2), \ldots, (x_{n+1}, y_{n+1}), \quad x_j \neq x_k \text{ se } j \neq k$$

si abbia

$$p_n(x_j) = y_j, \quad j = 1, \ldots, n+1.$$

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

$$L_i(x) = \frac{\prod_{j=1, j\neq i}^{n+1} (x - x_j)}{\prod_{j=1, j\neq i}^{n+1} (x_i - x_j)}.$$

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

$$\mathtt{x} = (x_1, \dots, x_{n+1}), \quad \mathtt{y} = (y_1, \dots, y_{n+1})$$

I coefficienti del polinomio interpolatore sono ottenibili dal comando `polyfit`.

A tal proposito l'help di Matlab (versione R2019a) 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
$$\begin{aligned} p_2(x) &= (-4/15) \cdot x^2 + (61/15) \cdot x + (36/5) \\ &\approx 0.2\overline{6}x^2 + 4.0\overline{6}x + 7.2. \end{aligned}$$
- Quindi, se $a = (a_1, a_2, a_3)$, abbiamo
$$p_2(x) = a_1\, x^2 + a_2\, x + a_3.$$
- In generale, se $a = (a_1, \ldots, a_{n+1})$ è il vettore ottenuto utilizzando `polyfit`, allora il polinomio interpolatore di grado n risulta
$$p_n(x) = a_1\, x^n + a_2\, x^{n-1} + \ldots + a_{n+1}.$$

Per valutare il 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 $\texttt{P} = (a_1, \ldots, a_{n+1})$, in un vettore di ascisse $\texttt{X} = (X_1, \ldots, X_m)$, 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)
>>
```

Siamo pronti per definire una routine `interpol` che

- dati i vettori
$$\mathtt{x} = (x_1, \ldots, x_m),\ \ \mathtt{y} = (y_1, \ldots, y_m)$$
calcoli i coefficienti `coeff` che definiscono il polinomio p_{m-1}, interpolante le coppie (x_k, y_k) per $k = 1, \ldots, m$;
- definito il vettore $\mathtt{s} = (s_1, \ldots, s_M)$, definisca un vettore $\mathtt{t} = (t_1, \ldots, t_M)$ in cui $t_k = p_{m-1}(s_k)$.

Un tal proposito può essere raggiunto mediante la funzione interpol:

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

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

Quindi,

- tramite `polyfit` siamo in grado di determinare i coefficienti del polinomio interpolatore;
- tramite `interpol` di valutarlo in ascisse arbitrarie.

Interpoliamo la funzione di Runge

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

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

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

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

```
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);
```

Nota.

Abbiamo valutato la funzione di Runge, in un vettore `s` *mediante una routine. Alternativamente si poteva definire la funzione di Runge e poi valutarla in un generico vettore* `s`*, ovvero*

`f=@(x) 1./(1+x.^2); fs=feval(f,s);`

Definiamo gcl.m che genera n nodi di Gauss-Chebyshev- Lobatto in $[a, b]$:

```
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));
```

Nota.

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;
title('Errori di interpolazione');
legend('funzione runge','intp. nodi GCL');
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');
```

Nota.

Si osservi che questo file, essendo senza variabili di input e output, non avendo in calce introdotte altre subroutines, puo' essere scritta evitando la prima riga `function 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 un vettore `y` di 13 ordinate in cui $y_k = f(x_k)$, $k = 1, \ldots, 13$;
- detto $p_{12}^{(E)} \in \mathbb{P}_{12}$ il polinomio che interpola le coppie

$$(x_k, y_k), \quad k = 1, \ldots, 13,$$

  abbiamo definito il vettore `t` di 1001 componenti posto

$$t_i = p_{12}^{(E)}(s_i), \quad i = 1, \ldots, 1001.$$

- determinato un vettore `xgcl` di 13 ascisse, diciamo $x_k^{(GCL)}$, corrispondenti ai nodi di Gauss-Chebysheb-Lobatto;
- determinato un vettore `ygcl` di 13 ordinate, diciamo $y_k^{(GCL)}$, in cui

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

- detto $p_{12}^{(GCL)} \in \mathbb{P}_{12}$ il polinomio che interpola le coppie

$$(x_k^{(GCL)}, y_k^{(GCL)}), \quad k = 1, \ldots, 13,$$

  abbiamo definito il vettore `tt` di 1001 componenti tale che

$$tt_i = p_{12}^{(GCL)}(s_i), \quad i = 1, \ldots, 1001.$$

- disegnato una prima figura, utilizzando il comando `figure(1)` con
  1. titolo `'Errori di interpolazione'`
  2. legenda `'funzione runge'`, `'intp.  nodi eqsp.'`,
  3. spessore della linea specificato da `'LineWidth',2`.

- disegnato una second figura, utilizzando il comando `figure(2)` con
  1. titolo `'Errori di interpolazione'`
  2. legenda `'funzione runge'`, `'intp.  nodi GCL'`,
  3. spessore della linea specificato da `'LineWidth',2`.
- assegnato alla variabile `ee` il valore

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

- assegnato alla variabile `ec` il valore

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

- 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 i grafici

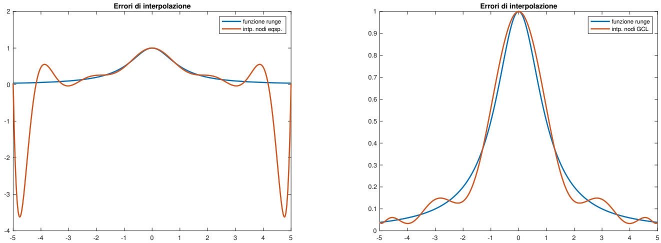

Figura: Grafici della funzione di Runge e delle interpolanti di grado 12, in nodi equispaziati e di Gauss-Chebyshev-Lobatto Dai grafici e dai risultati numerici si capisce la povera performance di $p_{12}^{(E)}$ in nodi equispaziati, soprattutto in virtù delle forti oscillazioni vicino agli estremi,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

Nota.

Al variare di n:

- *Notiamo che la scelta di "n" non può essere eccessiva. Per convincersene provare $n = 30$ (al posto di $n = 12$).*
- *Risulta evidente che non sussiste la convergenza 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 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.*

Esercizio (1)

Avendo ad esempio il file `esperimento`*, lo si modifichi nel file* `test_runge` *cosicchè :*

- *abbia come input la variabile n, grado dell'interpolante p_n che non sia necessariamente 12;*
- *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;*
- *esegua il test dell'interpolazione in*

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

- *non contenga grafici;*
- *non contenga statistiche fornite all'utente;*

- *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)|
```

Per la soluzione si veda il file test_runge.

Esercizio (2)

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

- *non abbia variabili di input, né di output;*
- *mediante ciclo-for, per* $n = 1, \ldots, 13$*, chiami per ogni* `n` *la funzione* `test_runge`*, avente in output* `ee`, `ec`*; di seguito ponga*
    - *la n-sima componente di* `eev` *pari al valore* `ee`*;*
    - *la n-sima componente di* `ecv` *pari al valore* `ec` *;*
- *esegua una prima figura (si utilizzi il comando* `figure(1)` *per rappresentarla in tale finestra) con i grafici in scala semilogaritmica delle coppie (n,*`eev`(n)*),* $n = 1, \ldots, 13$*; utilizzi quale titolo della prima figura la stringa*

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

  ed il plot abbia la preferenza `'LineWidth',2`*;*
- *esegua una seconda figura (si utilizzi il comando* `figure(2)` *per rappresentarla in tale finestra) con i grafici in scala semilogaritmica delle coppie (n,*`ecv`(n)*),* $n = 1, \ldots, 13$*;utilizzi quale titolo della seconda figura la stringa*

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

- *terminato il ciclo for, 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.*

Per la soluzione si veda il file demo_runge1.m.

Esercizio (3)

Avendo ad esempio il file `esperimento`*, lo si modifichi in* *demo_runge2.m* *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*

  $$|f(x) - p_{12}^{(E)}(x)|, \quad |f(x) - p_{12}^{(GCL)}(x)|$$

  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 equispaziati`
- *la seconda figura abbia titolo*

  `Errori di interpolazione: nodi GCL`
- *non si salvino risultati su testo.*

Per la soluzione si veda il file demo_runge2.m.