Interpolazione polinomiale in Matlab per Ingegneria dell'Energia
Esercizi risolti. [1]

A. Sommariva[2]

Abstract

Interpolazione polinomiale, esempi.

Ultima revisione: 12 aprile 2019

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
$$s_k = -5 + (k-1)h, \quad k = 1, \ldots, 10001, h = \frac{10}{10000}$$
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)|
```

1.1. Svolgimento

Il codice necessita solo di poche modifiche suggerite dal testo dell'esercizio. Abbiamo quindi

```
function [ee,ec]=test_runge(n)

% 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)|

% NODI TEST.
s=-5:10/10000:5;
% NODI EQSP.: ASCISSE/ORDINATE + INTP.TEST.
x=-5:10/n:5; y=runge(x);
t=interpol(x,y,s);
% NODI GCL.: ASCISSE/ORDINATE+INTP.TEST.
xgcl=gcl(-5,5,n+1); ygcl=runge(xgcl);
tt =interpol(xgcl, ygcl,s);
```

2. 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 semi-logaritmica 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;

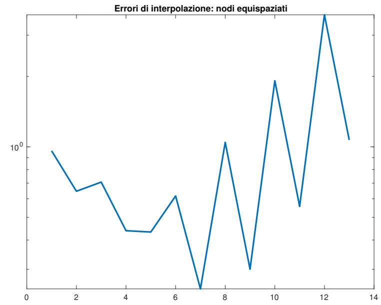
6. salvi su un file `errori_interpolazione.txt`, i valori di n utilizzati, gli errori `eev`, `ecv`, cosicchè la tabella risultante abbia alla k-sima riga,
    - l'indice k con 2 cifre prima della virgola e nessuna dopo la virgola, in notazione decimale,
    - l'errore `eev(k)`, ovvero la k-sima componente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

2.1. Svolgimento

Seppure le modifiche siano maggiori di quelle precedenti, basta seguire passo passo quanto richiesto. Quindi:

```
function demo_runge1

% test della interpolazione della funzione di Runge, in ...
    nodi equispaziati e
% di Chebyshev.

for n=1:13
    % test di runge, grado "n"
    [ee,ec]=test_runge(n);
    % immagazzinamento dati in "eev", "ecv".
    eev(n)=ee; ecv(n)=ec;
end

% NB: i vettori indici, eev, eec, sono di tipo riga,
%     di uguale lunghezza.

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

% prima figura
figure(1); % prima figura (nodi equi.)
semilogy(1:13,eev,'LineWidth',2);
title('Errori di interpolazione: nodi equispaziati');

% seconda figura
figure(2); % seconda figura
semilogy(1:13,ecv,'LineWidth',2);
title('Errori di interpolazione: nodi GCL');

% ---------- salvataggio risultati su file ----------

% creazione del file con facolta' di scrittura.
fid=fopen('errori_interpolazione.txt','w');
% dati immagazzinati nella matrice A (si immagazzinino ...
    come vettori riga,
% e bisogna ricordare che "eev", "ecv" sono riga.
A=[1:13; eev; ecv];
% scrittura dei dati su file.
fprintf(fid,'\n %3.0f  %1.15e %1.15e',A);
% chiusura file
fclose(fid);
```

La parte interessante riguarda l'esperimento, eseguito dalla command-window.



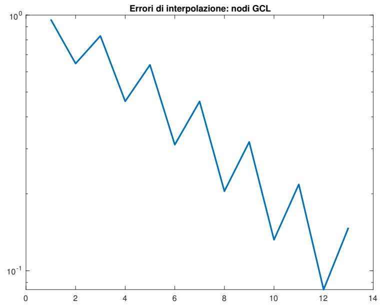


Figura 1: Grafici dei massimi errori assoluti tra funzione di Runge e le interpolanti di grado 12, in nodi equispaziati e di Gauss-Chebyshev-Lobatto. Dai grafici si capisce la povera performance dell'approssimazione fornita da $p_n^{(E)}$ e la discreta qualità dell'interpolante $p_n^{(GCL)}$ nei nodi di Gauss-Chebyshev-Lobatto.

```
>> demo_runge1
Warning: Polynomial is badly conditioned. Add points ...
    with distinct
X values, reduce the degree of the polynomial, or try ...
    centering and
scaling as described in HELP POLYFIT.
> In polyfit (line 79)
  In interpol (line 11)
  In test_runge (line 23)
  In demo_runge1 (line 9)
>>
```

Il warning dice che il programma ha problemi a risolvere il problema nel caso di nodi equispaziati (è il grado 13), ma lo risolve comunque.

Quale grafici abbiamo

Nota. 2.1 (Per utenti più esperti). *Il warning è dovuto al fatto che si risolve un certo sistema lineare $Va = \gamma$ con V matrice di Vandermonde, che a grado 13 è molto malcondizionata.*

Sul file di testo `errori_interpolazione.txt` viene salvato

```
  1  9.615384615384616e-01 9.615384615384616e-01
  2  6.462292681836165e-01 6.462292681836164e-01
  3  7.070135746606334e-01 8.289124668435013e-01
  4  4.383571411190436e-01 4.599809461571263e-01
  5  4.326923076923077e-01 6.386413798741108e-01
  6  6.169479686633391e-01 3.111950486491427e-01
  7  2.473586065593149e-01 4.596053247740365e-01
  8  1.045176657221059e+00 2.046825430368999e-01
  9  3.002979350388099e-01 3.190952547347740e-01
 10  1.915658917643507e+00 1.321974272333205e-01
```

```
11  5.567751127350897e-01 2.177056172348010e-01
12  3.663394045882839e+00 8.439663553025600e-02
13  1.070105625417639e+00 1.473232595238319e-01
```

che mostra numericamente i problemi dell'interpolare la funzione di Runge in nodi equispaziati e la lenta convergenza nel caso di quelli di Chebyshev.

3. 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

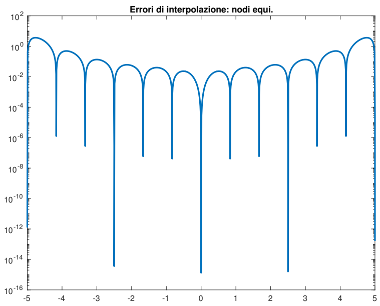
  `Errori di interpolazione: nodi equi.`
- la seconda figura abbia titolo

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

3.1. Commento

Come nel caso precedente, si tratta di piccole modifiche del file `esperimento`. Abbiamo

```
function demo_runge2

% oggetto:
% Esempio di Runge per grado fissato "n", in nodi ...
    equispaziati e di
% Chebyshev-Lobatto. E r r o r i puntuali.

% grado interpolante.
n=12;
% nodi test
s=-5:1/1000000:5;

% ---- 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 ----
```



[!ht]

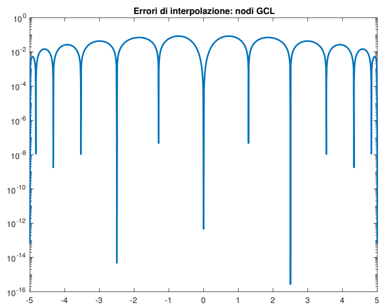


Figura 2: Grafici dei massimi errori assoluti tra funzione di Runge e le interpolanti di grado 12, in nodi equispaziati e di Gauss-Chebyshev-Lobatto. Dai grafici si capisce la conferma povera performance dell'approssimazione fornita da $p_{12}^{(E)}$ e la discreta qualità dell'interpolante $p_{12}^{(GCL)}$ nei nodi di Gauss-Chebyshev-Lobatto. Si notino i 13 picchi verticali, con errori quasi nulli, dovuti al campionamento del polinomio interpolante vicino a punti di interpolazione dove l'errore in effetti è nullo. Nel caso dei nodi equispaziati, si noti che intorno a 0 gli errori non sono eccessivi, mentre lo sono agli estremi.

```
fs=runge(s);

% prima figura
clf;
figure(1); % prima figura (nodi equi.)
semilogy(s,abs(fs-t),'LineWidth',2);
hold on;
title('Errori di interpolazione: nodi equi.');
hold off;

% seconda figura
figure(2); % seconda figura (nodi GCL)
semilogy(s,abs(fs-tt),'LineWidth',2);
hold on;
title('Errori di interpolazione: nodi GCL');
hold off;
```

Lanciato il programma da command-window abbiamo

```
>> demo_runge2
>>
```

e il plot di due figure.

Nel grafico relativo ai nodi equispaziati e di Gauss-Chebyshev-Lobatto, abbiamo 13 picchi verticali, in quanto i punti sono vicini ai nodi di interpolazione in cui gli errori tra la funzione e l'interpolante sono nulli.

Nel caso dell'interpolazione nei nodi equispaziati, si vede la cattiva performance vicino a $-5, 5$, dovuti alle oscillazioni già viste nell'esperimento della routine `esperimento`. Attorno all'origine $x = 0$ invece la performance è abbastanza buona.

Nel caso dell'interpolazione nei nodi di Gauss-Chebyshev-Lobatto, si vede in generale una discreta performance un po' ovunque.