Interpolazione polinomiale

1 Interpolazione polinomiale

Siano dati $N+1$ punti $x_0,\dots,x_N$ a due a due distinti e in ordine crescente (cioé $x_i < x_{i+1}$), e i valori $y_0,\dots,y_N$ ivi assunti da una funzione $y = f(x)$.

Il problema dell'*interpolazione polinomiale* consiste nel calcolare il polinomio p_N di grado N tale che

$$p_N(x_i) = f_i,\ i = 0,\dots,N. \tag{1}$$

Perché questo problema abbia senso, necessita garantire l'esistenza ed unicitá di tale polinomio p_N, questione che puó essere provata in vari modi (ad esempio in termini della non singolaritá della matrice di Vandermonde).

Nel caso $N = 1$, il problema diventa relativamente semplice. Non è altro che il calcolo della retta che passa per due punti assegnati $P_0 = (x_0, y_0)$ e $P_1 = (x_1, y_1)$ cioè da

$$\frac{y-y_0}{y_1-y_0} = \frac{x-x_0}{x_1-x_0} \tag{2}$$

abbiamo facilmente

$$y = y_0\frac{x-x_1}{x_0-x_1} + y_1\frac{x-x_0}{x_1-x_0}. \tag{3}$$

Notiamo subito che i polinomi

$$L_0(x) := \frac{x-x_1}{x_0-x_1}$$

$$L_1(x) := \frac{x-x_0}{x_1-x_0}$$

hanno la particolarità che

$$L_i(x_j) = \delta_{ij}$$

dove al solito $\delta_{i,j}$ è il *delta di Kronecker*, cioè

$$\delta_{i,i} = 1, \delta_{i,j} = 0 \text{ per } i \neq j$$

L'idea dei polinomi di Lagrange è di estendere questa proprietà al caso $N > 1$. Lagrange osservò che il polinomio di grado N

$$L_k(x) := \prod_{j=0, j\neq k}^{N} \frac{x-x_j}{x_k-x_j} = \frac{(x-x_0)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_N)}{(x_k-x_0)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_N)} \tag{4}$$

è tale che $L_k(x_j) = \delta_{ij}$. Quindi si vede facilmente che

$$p_N(x) = \sum_{k=0}^{N} f_k L_k(x). \tag{5}$$

Esempio Calcoliamo il polinomio di grado 2 che assume nei nodi $x_0 = -2$, $x_1 = 1$, $x_2 = 3$ rispettivamente i valori $f_0 = 7$, $f_1 = 11$, $f_2 = 17$. Come si vede dalla definizione, i polinomi di Lagrange sono

$$\begin{aligned}
L_0(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-1)(x-3)}{(-2-1)(-2-3)} = \frac{(x-1)(x-3)}{15}\\
L_1(x) &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x-(-2))(x-3)}{(1-(-2))(1-3)} = \frac{(x+2)(x-3)}{-6}\\
L_2(x) &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x-(-2))(x-1)}{(3-(-2))(3-1)} = \frac{(x+2)(x-1)}{10}
\end{aligned}$$

Si vede subito che i polinomi L_0, L_1, L_2 sono di secondo grado (e quindi tali le loro combinazioni lineari), che

$$L_0(x_0) = L_0(-2) = 1, L_0(x_1) = L_0(1) = 0, L_0(x_2) = L_0(3) = 0$$

$$L_1(x_1) = L_1(1) = 1, L_1(x_0) = L_1(-2) = 0, L_1(x_2) = L_1(3) = 0$$

$$L_2(x_2) = L_2(3) = 1, L_2(x_0) = L_2(-2) = 0, L_2(x_1) = L_2(1) = 0$$

e che il polinomio definito in (5)

$$p_2(x) = -2\,L_0(x) + 11\,L_1(x) + 17\,L_2(x)$$

è tale che

$$p(x_0) = -2\,L_0(x_0) + 11\,L_1(x_0) + 17\,L_2(x_0) = -2\cdot 1 + 11\cdot 0 + 17\cdot 0 = -2,$$

$$p(x_1) = -2\,L_0(x_1) + 11\,L_1(x_1) + 17\,L_2(x_1) = -2\cdot 0 + 11\cdot 1 + 17\cdot 0 = 11,$$

$$p(x_2) = -2\,L_0(x_2) + 11\,L_1(x_2) + 17\,L_2(x_2) = -2\cdot 0 + 11\cdot 0 + 17\cdot 1 = 17,$$

e quindi è proprio il polinomio interpolante cercato.

Dal punto di vista pratico, (5) ha alcuni problemi. Se dopo aver calcolato il polinomio p_N interpolante in $N+1$ punti (x_0, y_0), ..., (x_N, y_N), desideriamo ottenere il polinomio p_{N+1} interpolante in $N+2$ punti (x_0, y_0), ..., (x_N, y_N), la formula (5) è inefficiente poichè bisogna ricalcolare tutti i polinomi di Lagrange. Fortunatamente, esistono altre maniere di esprimere il polinomio interpolatore, come quella di Newton, che non soffrono di questo problema.

Per quanto concerne l'errore, se la funzione è di classe C^{N+1}, si puó vedere dal *teorema del resto* che

$$f(x) - p_N(x) = f^{(N+1)}(\xi)\frac{\prod_{i=0}^{N}(x-x_i)}{(N+1)!} \tag{6}$$

dove $\xi \in I$ con I il più piccolo intervallo aperto contente $x_0, \ldots, x_N$.

Ma come scegliere (avendone la possibilità) i punti in cui interpolare la funzione $f : [a, b] \to \mathbb{R}$? Mostriamo due casi notevoli:

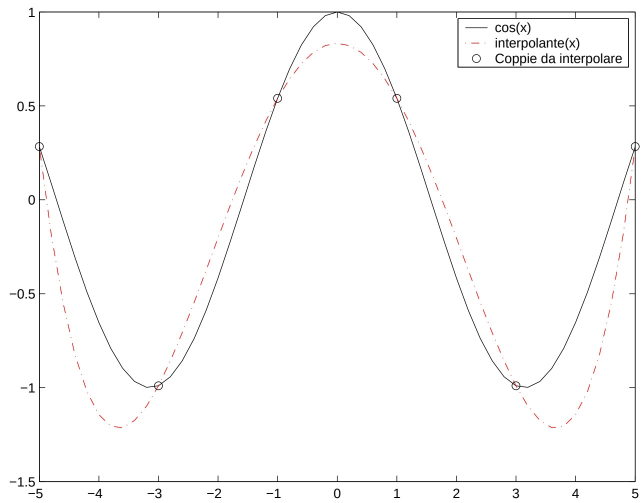

Figure 1: Grafico che illustra l'interpolazione su nodi equispaziati (la funzione ha la linea continua, il polinomio interpolatore è tratteggiato, i nodi di interpolazioni coi cerchietti).

1. *nodi equispaziati*: fissato N, i punti sono

$$x_k = a + k\frac{(b-a)}{N},\ k = 0,\dots,N; \tag{7}$$

2. *nodi di Chebyshev (scalati)*: fissato N, i punti sono

$$x_k = \frac{(a+b)}{2} + \frac{(b-a)}{2}\,t_k,\ k = 0,\dots,N \tag{8}$$

con

$$t_k = \cos(\frac{2k+1}{2N+2}\pi),\ k = 0,\dots,N; \tag{9}$$

In seguito useremo i nodi di Chebyshev-Lobatto, che a differenza dei nodi di Chebyshev includono gli estremi dell'intervallo. Si consiglia di effettuare gli esercizi successivi sostituendo ai nodi di Chebyshev-Lobatto (scalati), quelli di Chebyshev.

Viene da credere che aumentando N, nel caso dei nodi equispaziati, si abbia che $p_N(x) \to f(x)$ per $x \in [a,b]$. Come provato da Runge ciò non è vero. Infatti, per

$$f(x) = \frac{1}{1+x^2},\ x \in [-5,5] \tag{10}$$

si ha che p_N non converge (puntualmente) a f. Fortunatamente ciò non succede per i nodi di Chebyshev, per cui comunque (per un teorema dovuto a Faber) esistono funzioni continue f ma non C^1 tali che l'interpolante p_N non converge puntualmente a f.

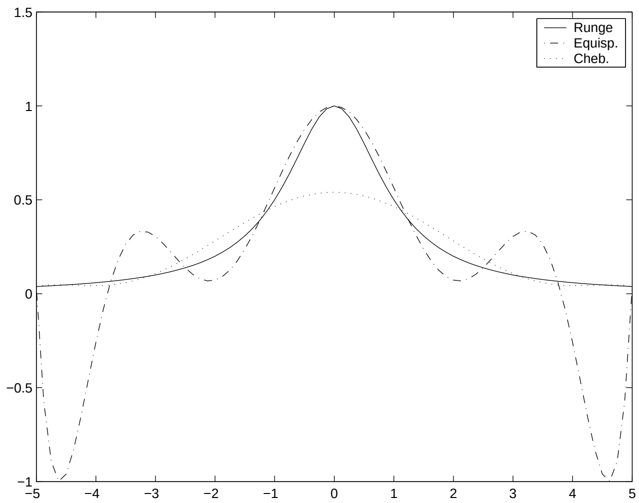

Figure 2: Esempio di Runge.

2 Esercizi sull'interpolazione polinomiale

1. Si implementi una function che calcola i nodi di Chebyshev per un intervallo qualsiasi.

2. Si implementi poi una function che interpoli polinomialmente una funzione che assume valori y su un vettore di nodi x che vengono forniti in input. Tale funzione Matlab deve inoltre calcolare i valori t che l'interpolante polinomiale p_N assume nei nodi test s. Si testi il codice, producendo dei grafici, sulle funzioni

   - $f_1(x) = x^{11} - 3x^7 + 6$ in $[-2, 2]$.
   - $f_2(x) = \frac{1}{1+x^2}$ in $[-5, 5]$.

   usando n nodi equispaziati (con alcuni valori di n tra 6 e 11) e n nodi di Chebyshev (con i medesimi valori di n), calcolando gli errori in norma infinito tra la funzione e le interpolanti. In particolare, si mostri (mediante grafici e/o valori dell'errore) che per l'esempio di Runge, l'aumento del numero di nodi equispaziati non migliora la ricostruzione della funzione da parte dell'interpolante.

2.1 Traccia di risoluzione

Si scrivano le functions `cheb.m`, `interpol.m`, `runge.m` definite rispettivamente come

```
function xc=cheb(a,b,n)
for m=1:1:n
```

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


function [t]=interpol(x,y,s)

%----------------------------------------------
% Interpolazione
%
% In input:
% x: nodi.
% y: valori nei nodi.
% s: nodi su cui calcolare l'interpolante.
%
% In output:
% t: valori dell'interpolante o approssimante.
%----------------------------------------------

m=max(size(x))-1;
coeff=polyfit(x,y,m);
t=polyval(coeff,s);


function [fx]=runge(x)
fx=1./(x.^2+1);
```

Eseguire poi da Matlab/Octave il programma principale `esperimento.m`

```
n=11;                % GRADO.
x=-5:10/n:5;         % NODI INTERP. (EQUISPAZ.).
y=runge(x);          % FUNZIONE NEI NODI EQUISP.
s=-5:10/(10*n):5;     % NODI TEST.
t=interpol(x,y,s);   % INTERPOLANTE NEI NODI TEST.

xcheb=cheb(-5,5,n); % NODI CHEB.
ycheb=runge(xcheb); % FUNZIONE NEI NODI CHEB.
tcheb=interpol(xcheb,ycheb,s);  % INTP. CHEB.
plot(s,runge(s),s,t,s,tcheb);   % PLOT INTP. VS RUNGE.
err_eqs=norm(runge(s)-t,inf);     % ABS. ERR. EQUISPAZ.
err_cheb=norm(runge(s)-tcheb,inf); % ABS. ERR. CHEB.
fprintf('\n \t [ABS.ERR.][EQS]: 2.2e [CHEB]: %2.2e',err_eqs,err_cheb);
```

Provando per diversi valori di n, e tralasciando i warnings di Matlab relativi a polyfit, otteniamo:

```
[N]:  2 [ABS.ERR.][EQS]: 6.46e-001 [CHEB]: 9.62e-001
[N]:  3 [ABS.ERR.][EQS]: 7.07e-001 [CHEB]: 6.46e-001
[N]:  4 [ABS.ERR.][EQS]: 4.38e-001 [CHEB]: 8.29e-001
[N]:  5 [ABS.ERR.][EQS]: 4.33e-001 [CHEB]: 4.58e-001
[N]:  6 [ABS.ERR.][EQS]: 6.09e-001 [CHEB]: 6.39e-001
```

```
[N]:  7 [ABS.ERR.][EQS]: 2.47e-001 [CHEB]: 3.11e-001
[N]:  8 [ABS.ERR.][EQS]: 1.04e+000 [CHEB]: 4.60e-001
[N]:  9 [ABS.ERR.][EQS]: 2.99e-001 [CHEB]: 2.04e-001
[N]: 10 [ABS.ERR.][EQS]: 1.92e+000 [CHEB]: 3.19e-001
[N]: 11 [ABS.ERR.][EQS]: 5.57e-001 [CHEB]: 1.32e-001
[N]: 12 [ABS.ERR.][EQS]: 3.66e+000 [CHEB]: 2.18e-001
[N]: 13 [ABS.ERR.][EQS]: 1.07e+000 [CHEB]: 8.41e-002
[N]: 14 [ABS.ERR.][EQS]: 7.15e+000 [CHEB]: 1.47e-001
[N]: 15 [ABS.ERR.][EQS]: 2.10e+000 [CHEB]: 5.33e-002
```

Si nota subito che

- l'errore dell'interpolante della funzione di Runge nei nodi equispaziati non converge a 0;
- l'errore dell'interpolante della funzione di Runge nei nodi di Chebyshev converge a 0 (si osservi la particolare decrescita dei gradi pari e dispari).

3 Sulla formula del resto

Si consideri la funzione

$$f(x) := \log(x)$$

e si supponga di conoscere i valori per $x = 1$, $x = 1.1$ ed $x = 1.2$. Si calcoli il polinomio di secondo grado p_2 che interpola tali punti, e lo si valuti in $s = 1.09$. Quindi si utilizzi la formula dell'errore per funzioni di classe C^{N+1} (*teorema del resto*)

$$f(x) - p_N(x) = f^{(N+1)}(\xi) \frac{\prod_{i=0}^{N}(x - x_i)}{(N+1)!}, \xi \in I(x_0, \ldots, x_N) \tag{11}$$

per valutare

$$|\log(s) - p_2(s)|.$$

Risoluzione. Dalla formula del resto (per $f(x) = \log(x)$, $x = 1.09$, $x_0 = 1$, $x_1 = 1.1$, $x_2 = 1.2$ ed $N = 2$) otteniamo:

$$|\log(1.09) - p_2(1.09)| = |\frac{2}{\xi^3} \frac{(1.09-1)(1.09-1.1)(1.09-1.2)}{6}|,$$

in quanto la derivata terza di $\log(x)$ è $\frac{2}{x^3}$. Siccome $\xi \in (1, 1.2)$, dalla decrescenza di $\frac{2}{x^3}$ deduciamo che

$$1.1574 \approx \frac{2}{1.2^3} < \frac{2}{\xi^3} < \frac{2}{1^3} = 2.$$

Essendo

$$(1.09-1)(1.09-1.1)(1.09-1.2) \approx 9.9 \cdot 10^{-5}$$

ricaviamo quindi che l'errore dell'interpolante è

$$1.9097\cdot 10^{-5} \approx 1.1574\cdot 9.9\cdot 10^{-5}/6 \le |\log(s) - p_2(s)| \le 2\cdot 9.9\cdot 10^{-5}/6 = 3.3\cdot 10^{-5}.$$

Utilizzando la funzione *interpol.m* precedentemente introdotta, si verifica che

$$p_2(1.09) \approx 0.08615260795055$$

mentre

$$\log(1.09) \approx 0.08617769624105$$

e quindi

$$|\log(1.09) - p_2(1.09)| \approx 2.51\cdot 10^{-5}$$

perfettamente in linea con le stime fornite.

Esercizio. Effettuare un programma Matlab che calcoli il valore assunto in 1.09 dal polinomio interpolatore della funzione log relativamente ai punti 1, 1.1, 1.2. E' buona l'approssimazione fornita dal polinomio interpolatore? (Suggerimento: si noti che

1. il programma è una variante di quanto visto in `esperimento.m`, per una adeguata scelta dei nodi `x`, dei valori `y`, del punto `s`;
2. eliminare la parte relativa ai nodi di Chebyshev-Lobatto;
3. il plot non è molto indicativo;
4. siccome il comando `norm(v,inf)` calcola per $v = \{v_i\}$ la quantità

   $$\|v\|_\infty = \max_i |v_i|$$

   visto che si deve valutare l'errore su un solo punto, il comando `norm(...,inf)` in `esperimento.m` può essere sostituito da `abs(...)` (rifletterci sopra);
5. la funzione `log` è predefinita in Matlab/Octave e quindi non serve ridefinirla come funzione.

Esercizio. Aiutandosi con Matlab/Octave eseguire un programma che calcoli l'interpolante della funzione

$$f(x) := \exp(x)$$

relativamente alle ascisse $x = 1$, $x = 1.1$ ed $x = 1.2$. Si calcoli il polinomio di secondo grado p_2 che interpola tali punti, e lo si valuti in $s = 1.09$. Fornire una stima dell'errore e verificarne la bontà rispetto al risultato esatto.

Esercizio. Fissati $a = -1$, $b = 1$, $n = 15$, si plottino i nodi di Chebyshev. Si suggerisce di utilizzare la function `cheb.m` e per il plottaggio un comando del tipo

```
plot(x,y,'r-o')
```

Ricordiamo che

- l'opzione `r-o`
  1. disegna un cerchietto rosso per ognuno dei punti (x_1, y_1), ..., (x_i, y_i), ..., (x_n, y_n);
  2. per $i = 1, \ldots, n-1$, unisce i punti (x_i, y_i), (x_{i+1}, y_{i+1}) con un segmento rosso.
- essendo importante disegnare solo le ascisse (e non le ordinate), quali ordinate si può porre

```
y=zeros(size(x));
```

Una volta completato l'esercizio si osservi la particolare disposizione dei nodi di Chebyshev. Si accumulano verso gli estremi dell'intervallo?

References

[1] V. Comincioli, *Analisi Numerica, metodi modelli applicazioni*, Mc Graw-Hill, 1990.

[2] The MathWorks Inc., *Numerical Computing with Matlab*, `http://www.mathworks.com/moler`.

[3] A. Quarteroni e F. Saleri, *Introduzione al calcolo scientifico*, Springer Verlag, 2006.