Laboratorio di Calcolo Numerico: Interpolazione e approssimazione di funzioni

Giacomo Elefante

Laboratorio di calcolo numerico
16/05/23

Scopo di oggi

Analizzare algoritmi per l'interpolazione e approssimazione di dati e funzioni:

- Problema di interpolazione
- Interpolazione su nodi equispaziati
- Interpolazione su nodi di Chebyshev

Problema di interpolazione

Dato $\mathbb{P}_n$ lo spazio dei polinomi di gradi al più n, ovvero i polinomi del tipo

$$a_n x^n + a_{n-1} x^{n-1} + \dots a_1 x + a_0$$

con $a_i \in \mathbb{R}, i = 0, \dots n$ allora il problema di interpolazione, ovvero trovare una funzione $p(x)$ che dati $n+1$ coppie $(x_0, y_0), \dots (x_n, y_n)$ soddisfa alla relazione

$$p(x_i) = y_i, \qquad i = 0, \dots, n,$$

ha una soluzione con $p(x) \in \mathbb{P}_n$ ed è unica.

Problema di interpolazione

In particolare, il polinomio si può costruire utilizzando le basi di Lagrange $\ell_i(x)$, ovvero

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

definendo il polinomio come

$$p(x) = \sum_{i=0}^{n} \ell_i(x) y_i,$$

poiché i polinomi di Lagrange sono tali che

$$\ell_i(x_j) = \begin{cases} 1, & j = i \\ 0, & \text{altrimenti} \end{cases}$$

Problema di interpolazione

In particolare, il polinomio si può costruire utilizzando le basi di Lagrange $\ell_i(x)$, ovvero

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

definendo il polinomio come

$$p(x) = \sum_{i=0}^{n} \ell_i(x) y_i,$$

poiché i polinomi di Lagrange sono tali che

$$\ell_i(x_j) = \begin{cases} 1, & j = i \\ 0, & \text{otherwise} \end{cases}$$

Esercizio

Esercizio

Il polinomio di Lagrange può essere utile da generare, anche se in Matlab esistono comandi più immediati per generare il polinomio di interpolazione. A tal proposito si chiede di creare una function matlab che dati dei nodi, dei punti e l'indice i-esimo mi valuti il polinomio ℓ_i di Lagrange su tali punti. Per fare ciò si generi la funzione `lagrai_target` che prende i seguenti input

- `z`: vettore di nodi (riga) di interpolazione
- `x`: vettore di punti (colonna) di valutazione del polinomio
- `i`: indice indicante il polinomio da generare considerando però i nodi ordinati da 1 a $n + 1$ (e non da 0 a n).

e che restituisca come output il vettore (colonna) l delle valutazioni di ℓ_i sui punti `x`.
Per creare la function possono essere utili i comandi `repmat` e `prod` di Matlab.

Interpolazione su Matlab

Su Matlab, gli $n+1$ coefficienti a_i, possono essere trovati attraverso facilmente con il comando `polyfit`, infatti date delle coppie di dati $(x_0, y_0), \dots, (x_n, y_n)$, `polyfit` può generare i coefficienti del polinomio interpolatorio tra questi punti.
In seguito per valutare un polinomio su dei punti, dati i coefficienti, è utile il comando `polyval`.

```
function t=interpol(x,y,s)
% Input:
%         x,y  vettori di ascisse cui si associa il polinomio interpolatore "p_n"
%         s    vettore di ascisse in cui valutare il polinomio interpolatore
% Output:
%         t    vettore di valutazioni "p_n(s)"

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

Figure: Codice per generare le valutazioni del polinomio interpolante.

Nodi equispaziati

Consideriamo ora come nodi $n+1$ punti equispaziati in un intervallo $[a, b]$, generabili attraverso i comandi `linspace` oppure `a:h:b` definendo però il passo `h` come

$$h = \frac{b-a}{n}.$$

Essi potranno essere usati per interpolare una funzione f.

Esercizio

Esercizio

Data la funzione $f(x) = \frac{e^{-2x}}{3} + x^2 + 7x^3$, si crei uno script dove vengono definiti

- Una variabile `n` indicante il grado massimo del polinomio interpolante
- Un vettore contenente i nodi equispaziati `x`
- Un vettore `y` i cui elementi sono $f(x_i)$
- Un vettore `s` con 500 punti di valutazione.

Utilizzare quindi la routine `interpol` per valutare il polinomio interpolante in questi punti e creare un grafico dove vengono sovrascritti la funzione f in blu e il polinomio interpolante in rosso tratteggiato.
Calcolare quindi l'errore assoluto massimo tra il polinomio e la funzione e stamparne il valore a schermo. Come si comporta al variare di n?

In questo caso la convergenza dovrebbe essere assicurata per il tipo di funzione scelta ma al crescere di n si vede l'interpolante costruita sui nodi equispaziati perdere di efficacia e crescere il suo errore. Infatti, tale interpolante non è molto stabile poiché la sua costante di Lebesgue cresce esponenzialmente con n.
Per l'interpolazione infatti, si preferisce utilizzare delle distribuzioni di punti migliori più stabili come i punti di Chebyshev-Lobatto

$$x_i = -\cos\left(\frac{i\pi}{n}\right), \quad i = 0, \ldots, n,$$

la cui interpolante costruita su questi punti ha una costante di Lebesgue che cresce come $\log(n)$, quindi più lentamente.

Esercizio

Ripetere l'esercizio precedente ma utilizzando i punti di Chebyshev-Lobatto al posto di quelli equispaziati.

Esercizio

In alcuni casi invece la convergenza con i punti equispaziati può non esserci proprio, come può essere il caso della funzione di Runge in $I = [-1, 1]$, ovvero

$$f(x) = \frac{1}{1 + 25x^2}$$

Si costruisca, quindi, uno script dove al variare del grado del polinomio n da 1 a 25, si calcola il polinomio interpolante sia su nodi equispaziati sia su nodi di Chebyshev-Lobatto, valutandone poi il valore su 1000 punti equispaziati `s` in I.
Trovare quindi l'errore assoluto massimo dei polinomi con la funzione in `s` e fare in seguito la sovrapposizione dei grafici in scala semilogaritmica degli errori in funzione del grado del polinomio.

Esercizio (per casa)

Esercizio

Creare una funzione `interpol2` che prende in input/outup gli stessi input/output della funzione `interpol`, ovvero il vettore dei nodi `x`, il vettore dei valori della funzione nei nodi `y` e il vettore dei punti su cui valutare il polinomio interpolante `s` e come output il vettore `t` dei valori del polinomio interpolante sui punti `s`.
A differenza di ciò che è stato visto a lezione con la funzione `interpol`, per costruire il polinomio interpolante si utilizzi direttamente la definizione di polinomio con basi di Lagrange, ovvero descrivere p_n come

$$p_n(x) = \sum_{i=0}^{n} \ell_i(x) y_i = \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}.$$