

# Università degli Studi di Cagliari

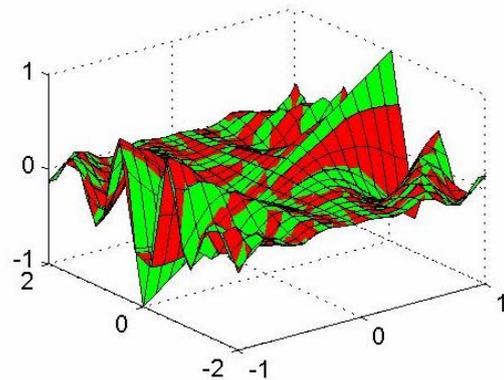
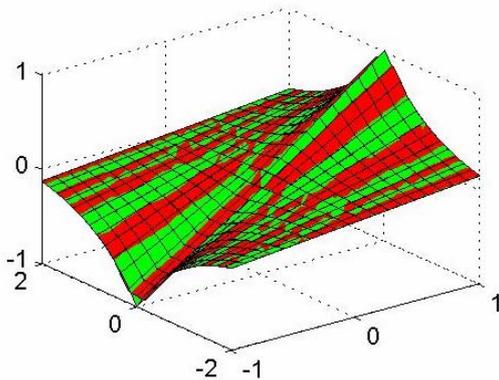
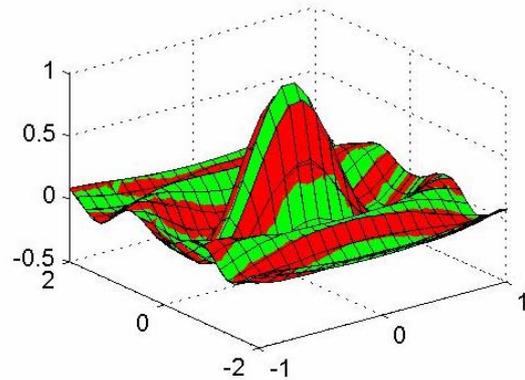
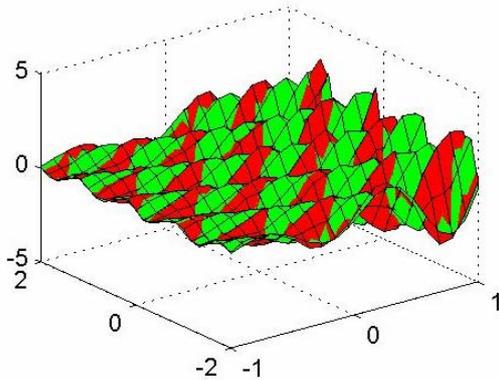
Facoltà di Ingegneria

Corso di Laurea Specialistica in Ingegneria per l'Ambiente ed il Territorio

---

## Approssimazione di Funzioni con l'ausilio di MATLAB.

---



*Elaborato di Calcolo Numerico*

Docente: **Prof. Giuseppe RODRIGUEZ**

Studente: **Alessandro MAZZELLA**

Anno Accademico 2007-2008

# Indice

<b>1. Introduzione</b> .....	3
<b>2. Interpolazione di una funzione</b> .....	4
<b>3. Interpolazione Polinomiale</b> .....	5
3.1 Polinomio Interpolante nella Forma di Lagrange .....	6
3.1.1 Implementazione MATLAB .....	7
<i>Esempio</i> .....	8
3.2 Errore di Interpolazione .....	10
3.2.1 Implementazione MATLAB .....	12
<i>Esempio</i> .....	13
<b>4. Approssimazione nel Senso dei Minimi Quadrati</b> .....	15
<i>Esempio</i> .....	17
<i>Esempio</i> .....	18

# 1. Introduzione

Molto spesso in campo ingegneristico ci si può trovare a dover risolvere problemi di **approssimazione di funzione**, riconducibili alle seguenti tipologie principali:

1. Della funzione  $f(x)$  si conoscono solamente i valori in un insieme discreto di punti  $P_i(x_i, y_i)$  con  $i = 0, 1, 2, \dots, n$  tali che  $y_i = f(x_i)$  provenienti, ad esempio, da una serie di analisi sperimentali. In questo frangente può essere necessario disporre di una approssimazione della funzione  $f(x)$  in maniera tale da poter stimare il suo valore in un punto di coordinate qualsiasi  $(X, Y)$  con  $X \neq x_i$ , oppure i valori di una sua derivata di ordine  $k$   $f^{(k)}(x)$ . Si possono presentare vari casi:
  - a) il numero di dati disponibili è piccolo, come nello studio di alcuni fenomeni fisici o biologici;
  - b) il numero di dati disponibili è molto elevato, come di solito accade nelle analisi statistiche;
  - c) i valori  $y_i = f(x_i)$  possono essere affetti da errori di misura limitati e con una distribuzione abbastanza regolare, come accade di solito per i dati rilevati in laboratorio;
  - d) i valori  $y_i = f(x_i)$  possono essere affetti da errori di misura molto elevati e senza alcuna distribuzione regolare, come può accadere per esempio per un non corretto funzionamento degli strumenti di laboratorio o quando, in una indagine economica o demografica, i dati vengono rilevati in modo errato.
2. La funzione  $f(x)$  possiede alcune proprietà di continuità e derivabilità che permettono di scriverne lo sviluppo di Taylor, oppure possiede delle proprietà più specifiche, come ad esempio periodicità o simmetria, che possono essere opportunamente sfruttate per costruire una funzione  $g(x)$  più facilmente calcolabile della stessa  $f(x)$ .

Le tecniche per la risoluzione di questi tipi di problemi sono principalmente due:

- **l'interpolazione**: si determina quella funzione, tra una fissata famiglia di funzioni, che in opportuni punti, detti "nodi", assuma gli stessi valori assunti dalla funzione data;
- **l'approssimazione**: si determina quella funzione, tra una fissata famiglia di funzioni, la cui distanza dalla funzione da approssimare risulti la minima possibile.

Nel presente studio sono stati sviluppati e confrontati alcuni dei metodi di interpolazione polinomiale e di approssimazione nel senso dei minimi quadrati attraverso l'ausilio del software *MATLAB*.

## 2. Interpolazione di una funzione

Consideriamo  $n+1$  punti  $P_i$  di coordinate  $(x_i, y_i)$  con  $i = 0, 1, 2, \dots, n$ . Vogliamo costruire una **funzione interpolante**  $\Phi(x)$  **passante per tutti gli  $i$  punti  $P_i$** , cioè tale che

$$\Phi(x_i) = y_i \text{ per } i = 0, 1, 2, \dots, n$$

Ipotezzando che i dati provengano da una funzione continua è possibile scegliere  $n+1$  funzioni continue  $\Phi_0, \Phi_1, \Phi_2, \dots, \Phi_n$  di base e approssimare la funzione  $f(x)$  data mediante una loro opportuna combinazione lineare

$$\Phi(x) = \sum_{j=0}^n a_j \Phi_j(x)$$

Per calcolare i coefficienti della combinazione lineare di cui sopra è sufficiente imporre le  $n+1$  **condizioni di interpolazione**

$$\sum_{j=0}^n a_j \Phi_j(x_i) = y_i \text{ per } i = 0, 1, 2, \dots, n$$

ovvero risolvere il sistema lineare  $\Phi \underline{a} = \underline{y}$ . Tale sistema lineare ammetterà una e una sola soluzione se e solo se la matrice  $\Phi$  è **non singolare**. Pertanto: **la funzione che interpola i dati esiste ed è unica se e solo se il determinante della matrice  $\Phi$  (detto determinante di Haar) è diverso da zero.**

$$\Phi = \begin{bmatrix} \Phi_0(x_0) & \Phi_1(x_0) & \dots & \Phi_n(x_0) \\ \Phi_0(x_1) & \Phi_1(x_1) & \dots & \Phi_n(x_1) \\ \dots & \dots & \dots & \dots \\ \Phi_0(x_n) & \Phi_1(x_n) & \dots & \Phi_n(x_n) \end{bmatrix}$$

### 3. Interpolazione Polinomiale

Le funzioni utilizzate più di frequente come **base** per l'interpolazione sono i **polinomi**. Tale scelta è giustificata dalla loro attitudine ad approssimare eccellentemente comportamenti regolari.

Il problema dell'**approssimazione polinomiale** viene affrontato costruendo, a partire dagli  $n+1$  punti  $P_i(x_i, y_i)$  con  $i = 0, 1, 2, \dots, n$ , il polinomio  $P_n$  di grado  $n$  espresso nella forma

$$P_n(x) = \sum_{j=0}^n a_j x^j$$

Per quanto scritto precedentemente, il sistema lineare che consente di determinare i coefficienti del polinomio interpolante sarà del tipo

$$X \underline{a} = \underline{y}$$

dove:

$\underline{y} = (y_0, y_1, y_2, \dots, y_n)$  è il vettore delle ordinate

$$X = \begin{bmatrix} 1 & x_0^1 & x_0^2 & \dots & x_0^n \\ 1 & x_1^1 & x_1^2 & \dots & x_1^n \\ 1 & x_2^1 & x_2^2 & \dots & x_2^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n^1 & x_n^2 & \dots & x_n^n \end{bmatrix} \text{ è la } \mathbf{\text{matrice di Vandermonde}} \text{ costruita a partire dai nodi } \{x_i\}$$

Poiché la condizione di unisolvenza si traduce in  $\det(X) = \prod_{i>j} (x_i - x_j) \neq 0$ , questo implica che il **polinomio interpolante esiste ed è unico se e solo se i nodi di interpolazione sono tutti distinti**.

Tuttavia questo approccio non è numericamente conveniente per le seguenti motivazioni:

1. la matrice  $X$  tende ad essere estremamente **mal condizionata quando i nodi  $\{x_i\}$  sono vicini**;
2. **il costo computazionale per la risoluzione dell'interpolazione è molto elevato  $o(n^3)$** ;
3. **la rappresentazione canonica è instabile**, cioè piccole perturbazioni nei coefficienti

$a_j$  possono produrre grandi variazioni sui valori di  $P_n(x)$ .

Per ovviare a questi inconvenienti, esistono degli algoritmi che permettono di rappresentare in altra forma il polinomio  $P_n(x)$ .

### 3.1 Polinomio Interpolante nella Forma di Lagrange

La forma di Lagrange è una rappresentazione dell'unico polinomio interpolante  $P_n(x)$  in una base, diversa da quella canonica  $\{1, x, x^2, \dots, x^n\}$  (numericamente non conveniente per le motivazioni esposte precedentemente) e costituita dai **Polinomi caratteristici di Lagrange** così definiti:

$$\begin{aligned}
 L_0(x) &= \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)} \\
 L_1(x) &= \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)} \\
 &\vdots \\
 L_n(x) &= \frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1})}
 \end{aligned}$$

in generale:

$$L_j(x) = \prod_{\substack{k=0 \\ k \neq j}}^n \frac{x-x_k}{x_j-x_k}$$

È possibile osservare che ciascun polinomio caratteristico di Lagrange  $L_j(x)$  è un polinomio di grado  $n$ , con le ascisse di interpolazione distinte  $x_i \neq x_j$ , che verifica la condizione di interpolazione:

$$L_j(x_i) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

cioè il valore del generico polinomio di Lagrange  $L_j(x)$  calcolato in un punto  $x_i$  vale 1 se  $x_i \equiv x_j$ , mentre vale 0 se viene calcolato in un punto  $x_i \neq x_j$ .

Il polinomio di Lagrange assume, quindi, la forma

$$P_n(x_i) = \sum_{j=0}^n y_j L_j(x_i) = \sum_{j=0}^n y_j \delta_{ij} = y_i \text{ per } i = 0, 1, 2, \dots, n$$

È evidente che anche in questo caso, il calcolo del polinomio interpolante di Lagrange  $L_j(x)$  può essere instabile quando due nodi sono molto vicini tra loro poiché, in tal caso, almeno uno dei fattori a denominatore risulterebbe prossimo allo zero.

### 3.1.1 Implementazione MATLAB

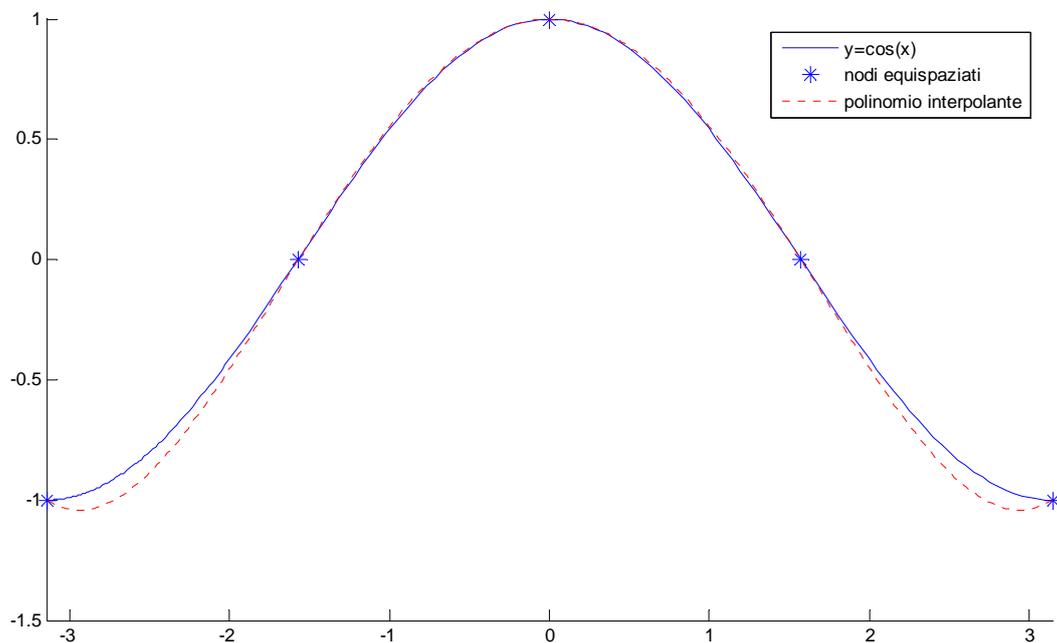
La valutazione del Polinomio Interpolante di Lagrange può essere ricavata in MATLAB mediante poche e semplici righe di comando: la funzione *ac\_intlagr.m* implementa tale algoritmo, prendendo in ingresso i punti di interpolazione contenuti nei vettori **xi** e **yi** e i nodi di valutazione del polinomio contenuti in **x**.

```
0 % ac_intlagr.m
1 n = length(xi);
2 m = length(x);
3 y = zeros(m,1);
4 for j = 1:m
5     y(j) = 0;
6     for k = 1:n
7         space = [1:k-1, k+1:n];
8         L = prod(x(j) - xi(space))/prod(xi(k) - xi(space));
9         y(j) = y(j) + yi(k) * L;
10    end
11 end
```

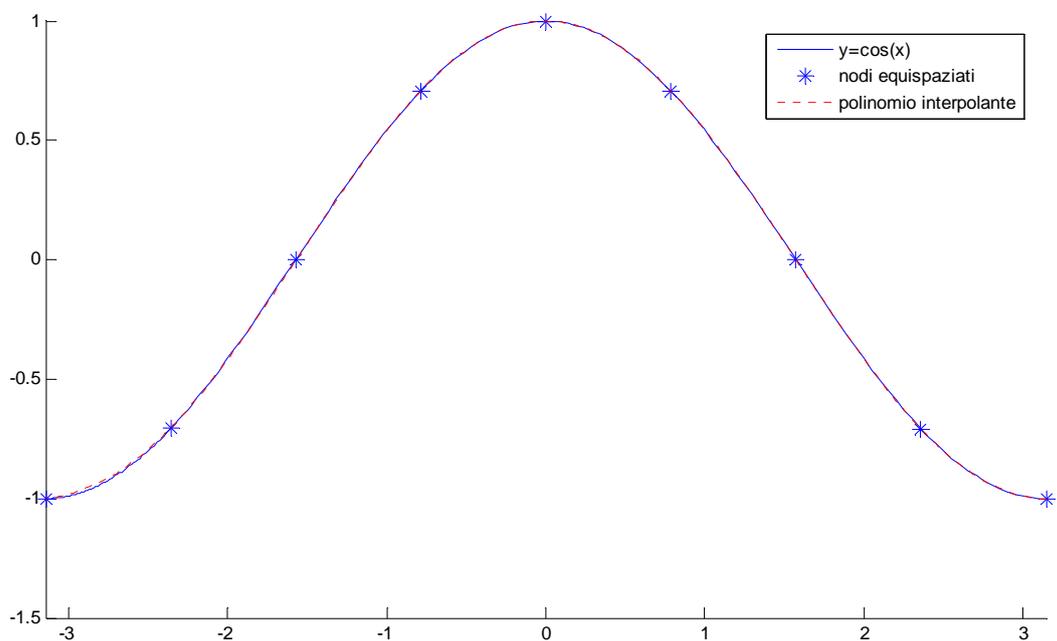
In riferimento al listato precedente:

- ❑ le righe (1) e (2) inizializzano i contatori m ed n necessari per l'esecuzione della sommatoria e della produttoria per il calcolo dei valori numerici dei polinomi caratteristici di Lagrange;
- ❑ la riga (3) inizializza il vettore y, nel quale verranno memorizzati le ordinate interpolate nei nodi x;
- ❑ la riga (8) calcola, step dopo step, il valore del Polinomio Caratteristico di Lagrange necessario alla riga successiva per il calcolo della j-esima ordinata interpolata;
- ❑ la riga (9) calcola, step dopo step, il valore dell'ordinata interpolata nel j-esimo nodo di interpolazione memorizzato nel vettore x e lo memorizza nella posizione j-esima del vettore y.

## Esempio



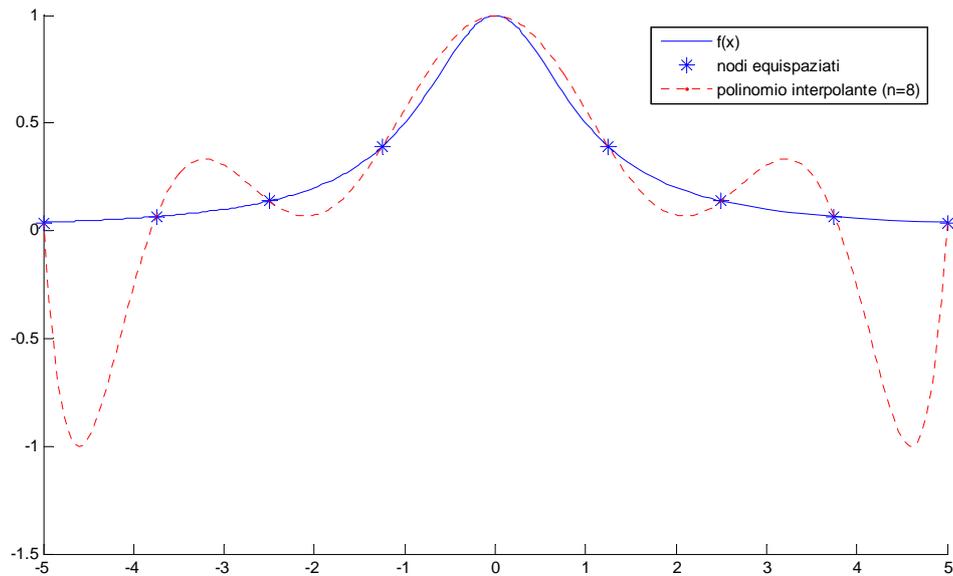
**Figura 1 - Interpolazione con Polinomio di Lagrange di  $y = \cos(x)$  [4 nodi di interpolazione].**



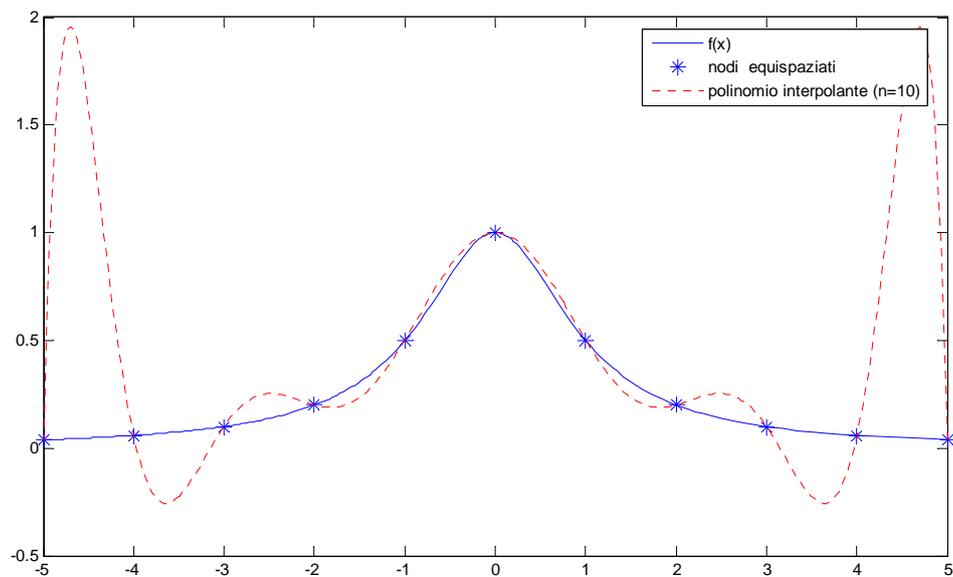
**Figura 2 - Interpolazione con Polinomio di Lagrange di  $y = \cos(x)$  [8 nodi di interpolazione]**

Erroneamente si potrebbe concludere che al crescere del grado  $n$ , il polinomio  $P_n(x)$  si avvicini indefinitamente alla funzione  $f(x)$ . Tuttavia questo non è vero in generale. Esiste, infatti, il classico

controesempio di Runge (Carle Runge, 1856-1927): per la funzione  $f(x) = \frac{1}{1+x^2}$  il polinomio interpolatore, all'aumentare del grado  $n$  (e quindi dei punti in comune con  $f(x)$ ), presenta un comportamento patologico, e anziché avvicinarsi a  $f(x)$  se ne allontana indefinitamente, come mostrato dai grafici seguenti.



**Figura 3 - Interpolazione con Polinomio di Lagrange di  $y = \frac{1}{1+x^2}$  [8 nodi di interpolazione]**



**Figura 4 - Interpolazione con Polinomio di Lagrange di  $y = \frac{1}{1+x^2}$  [10 nodi di interpolazione]**

## 3.2 Errore di Interpolazione

Alla luce di quanto emerso nei precedenti paragrafi, risulta evidente l'importanza di valutare l'errore di interpolazione ovvero la distanza tra la funzione  $f(x)$  interpolanda e il polinomio interpolatore  $P_n(x)$  nei punti  $x \neq x_i$ .

Data una funzione  $f(x)$  derivabile parecchie volte (cioè  $f \in C^{n+1}[a,b]$ ) e dato il polinomio  $P_n(x)$  che interpola  $f(x)$  sulle ascisse  $\{x_0, x_1, \dots, x_n\}$ , è possibile dimostrare che per ogni  $x$  esiste un punto appartenente al più piccolo intervallo che contiene i punti  $\{x_0, x_1, \dots, x_n\}$ , tale che la **funzione errore** è definita come:

$$E_n(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi_k)}{(n+1)!} \omega_n(x)$$

Tale teorema implica necessariamente due aspetti fondamentali:

- la quantità  $f^{(n+1)}(\xi_k)$  dipende dalla funzione  $f(x)$  ed è certamente maggiorabile con una costante, essendo la funzione di classe  $C^{n+1}$ ;
- la presenza del termine  $\omega_n(x)$  implica che la disposizione dei nodi influenza l'entità dell'errore di interpolazione.

Richiamiamo a questo punto tre risultati fondamentali della teoria dell'approssimazione:

- qualsiasi sia la distribuzione dei nodi esiste sempre almeno una funzione continua per cui l'errore tende all'infinito (**Teorema di Faber**). Perciò approssimando una funzione con un polinomio può accadere che, pur aumentando i punti di interpolazione (e quindi il grado del polinomio), la distanza tra quest'ultimo e la funzione non vada a zero;
- scelta una funzione continua esiste sempre una distribuzione di nodi tale che mandi l'errore di interpolazione a zero;
- se la funzione oltre ad essere continua è derivabile almeno una volta e vengono scelti come nodi gli **zeri del polinomio di Chebychev**:

$$x_k = \cos\left(\frac{2k+1}{n+1} \cdot \frac{\pi}{2}\right) \text{ con } k = 0, 1, 2, \dots, n$$

allora l'errore di interpolazione tende a zero (**Teorema di Bernstein**)..

Poiché i nodi di Chebychev forniscono la soluzione del problema di minimizzazione dell'errore nell'intervallo  $[-1,+1]$ , per ottenere la disposizione di tali nodi su di un generico intervallo  $[a,b]$  è necessario applicare la trasformazione:

$$t_k = \left(\frac{b-a}{2}\right) \cdot x_k - \left(\frac{b+a}{2}\right)$$

## 3.2.1 Implementazione MATLAB

La valutazione dei Nodi di Chebychev può essere effettuata in MATLAB mediante la funzione ***ac\_chebychev.m***.

```
0      % ac_chebychev.m
1      k=length(xi);
2      x=zeros(1,k);
3      % calcolo degli zeri del polinomio di Chebychev di grado k
4      for i=1:k
5          x(i)=cos((((2*i)+1)/(k))*(pi/2));
6      end
7      % calcolo i nodi nell'intervallo [a,b]
8      for i=1:k
9          t(i)=((b-a)/2)*x(i)-((a+b)/2);
10     end
```

In riferimento al listato precedente:

- ❑ la riga (1) inizializza il contatore k necessario per il calcolo degli zeri del polinomio di Chebychev;
- ❑ la riga (2) inizializza il vettore x nel quale verranno memorizzate le ascisse dei nodi di Chebychev;
- ❑ le righe (4), (5) e (6) sono un loop per il calcolo dell'i-esimo zero del polinomio di Chebychev;
- ❑ le righe (8), (9) e (10) sono un loop per il calcolo del valore dell'ascissa dell'i-esimo nodo di Chebychev, tenuto conto che non sono disposti nell'intervallo [-1,1] bensì in un generico intervallo [a,b].

## Esempio

Consideriamo nuovamente la funzione  $f(x) = \frac{1}{1+x^2}$  e valutiamo l'andamento dell'errore di interpolazione

nel caso di **nodi equispaziati** e nel caso di **nodi di Chebychev**.

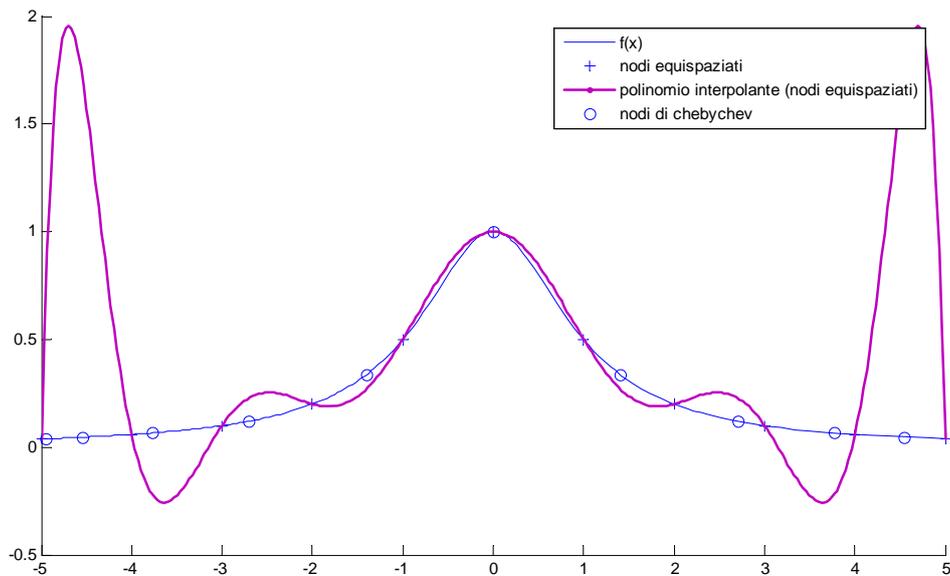


Figura 4 – Interpolazione di  $y = \frac{1}{1+x^2}$  [10 nodi equispaziati]

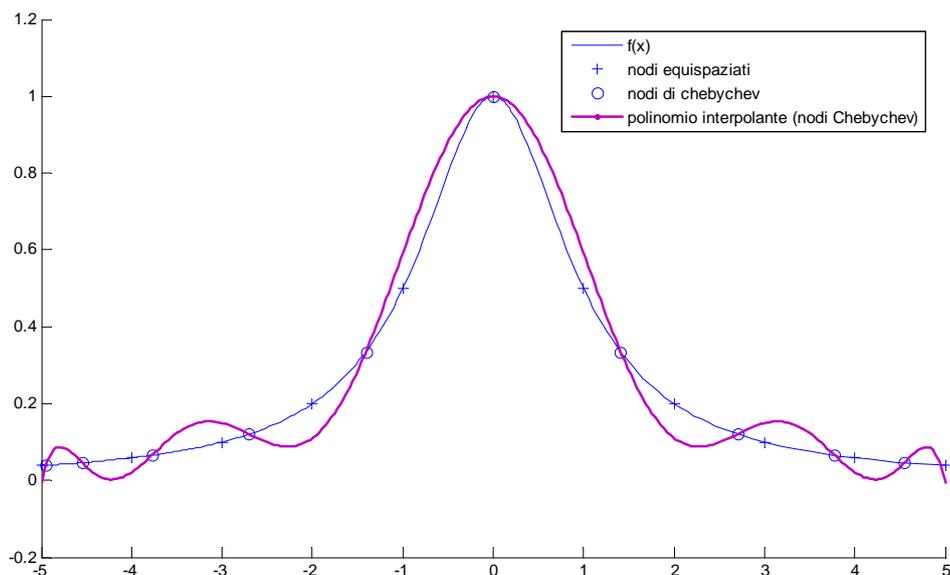
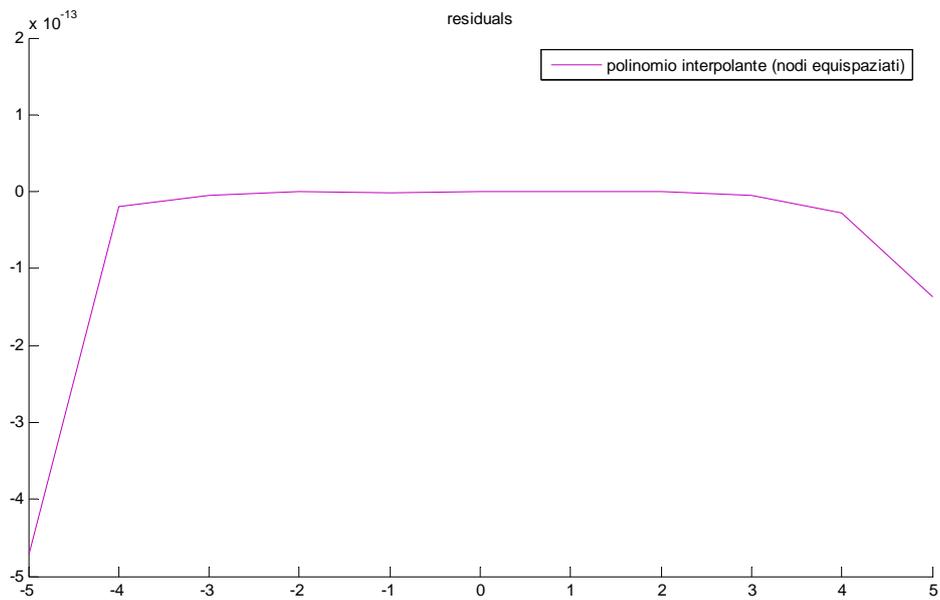
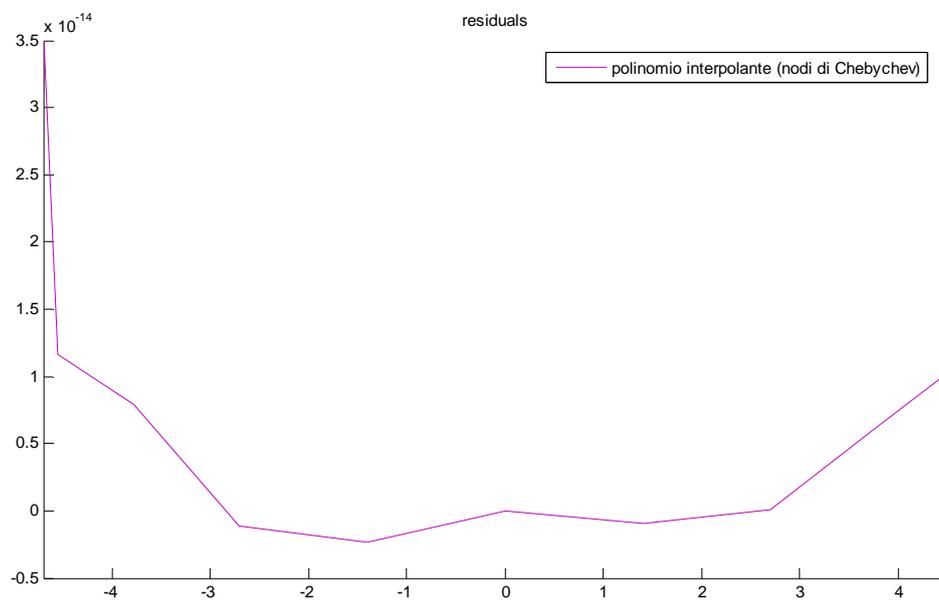


Figura 5 - Interpolazione di  $y = \frac{1}{1+x^2}$  [10 nodi di Chebychev]



**Figura 6 – Errore di interpolazione [nodi equispaziati]**



**Figura 7 – Errore di interpolazione [nodi di Chebychev]**

## 4. Approssimazione nel Senso dei Minimi Quadrati

Un problema di approssimazione coinvolge generalmente casi in cui si posseggono molti punti più o meno affetti da errori. In tal caso è sconsigliabile affrontare un'interpolazione dei dati dal momento che è nota la presenza di forti errori sperimentali e la curva di interpolazione oscillerebbe per passare in tutti i punti. Inoltre calcolare il polinomio di migliore approssimazione consisterebbe nel determinare quel polinomio  $P_n(x)$  di grado  $n$  che minimizzi una norma dell'errore, cioè

$$\min_{P_n \in \Pi_n} \|P_n - f\|$$

Poiché siamo in uno spazio a dimensione infinita, le norme non sono tutte equivalenti, pertanto una scelta razionale è quella di considerare la norma 2. Per le funzioni essa è definita come

$$\|f - P_n\|_2 = \left( \int_a^b (f(x) - P_n(x))^2 \cdot dx \right)^{1/2}$$

Siano  $\{x_0, x_1, \dots, x_n\}$  le ascisse degli  $m-1$  punti per i quali sono noti i valori  $\{y_0, y_1, \dots, y_n\}$  della funzione  $f(x)$ . Fissato un numero naturale  $n \leq m$ , per ogni  $P_n \in \Pi_n$  si considera la norma 2 discreta definita come:

$$\|f - P_n\|_2^2 = \left( \sum_{i=0}^m |P_n(x_i) - f(x_i)|^2 \right)^{1/2}$$

Se  $n = m$  la soluzione coincide con la soluzione del polinomio interpolante se invece  $m > n$  si ottiene la migliore approssimazione nel senso dei minimi quadrati rispetto alla norma scelta.

Utilizzando la base canonica  $\{1, x, x^2, \dots, x^n\}$  si ottiene:

$$P_n(x_i) = \sum_{j=0}^n a_j x_i^j = \begin{pmatrix} X & a \end{pmatrix}_i \text{ con } i = 0, 1, 2, \dots, n$$

dove:

$$X = \begin{bmatrix} 1 & x_0^1 & x_0^2 & \dots & x_0^n \\ 1 & x_1^1 & x_1^2 & \dots & x_1^n \\ 1 & x_2^1 & x_2^2 & \dots & x_2^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_m^1 & x_m^2 & \dots & x_m^n \end{bmatrix} \text{ è la } \mathbf{\text{matrice di Vandermonde}} \text{ di dimensione } (m+1) \times (n+1)$$

Di conseguenza si ha:

$$\|f - P_n\|_2^2 = \sum_{i=0}^m [(Xa)_i - y_i]^2 = \|X \underline{a} - y\|_2^2$$

ovvero:

$$\min_{P_n \in \Pi_n} \|f - P_n\|_2^2 = \min_{\underline{a} \in R^{n+1}} \|X \underline{a} - y\|_2^2$$

Il problema risulta essere equivalente alla risoluzione, nel senso dei minimi quadrati, del sistema sovradeterminato  $X \underline{a} = y$  che può essere risolto con uno dei metodi numerici per la risoluzione dei sistemi lineari.

## Esempio

Per la sperimentazione numerica si è considerata la funzione trigonometrica  $f(x) = \sin(x)$ . Si è ipotizzato un vettore  $y$  di dati affetti da errore con distribuzione gaussiana:

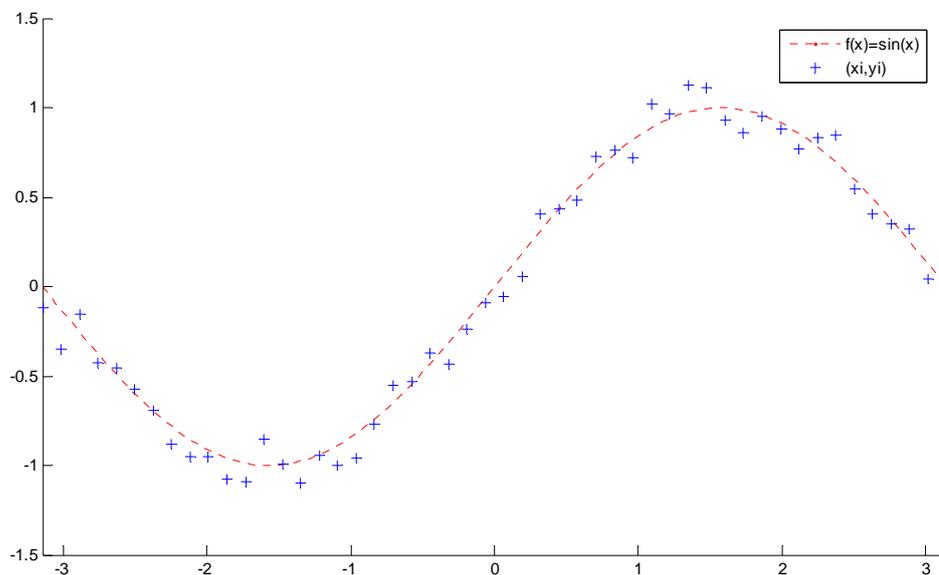


Figura 8 – Grafico della funzione  $y = \sin(x)$   $[-\pi, +\pi]$

Attraverso alcuni semplici passaggi si è proceduto alla costruzione della matrice di Vandermonde, alla risoluzione del sistema lineare sovradeterminato  $Xa = y$  ed alla definizione del polinomio interpolante.

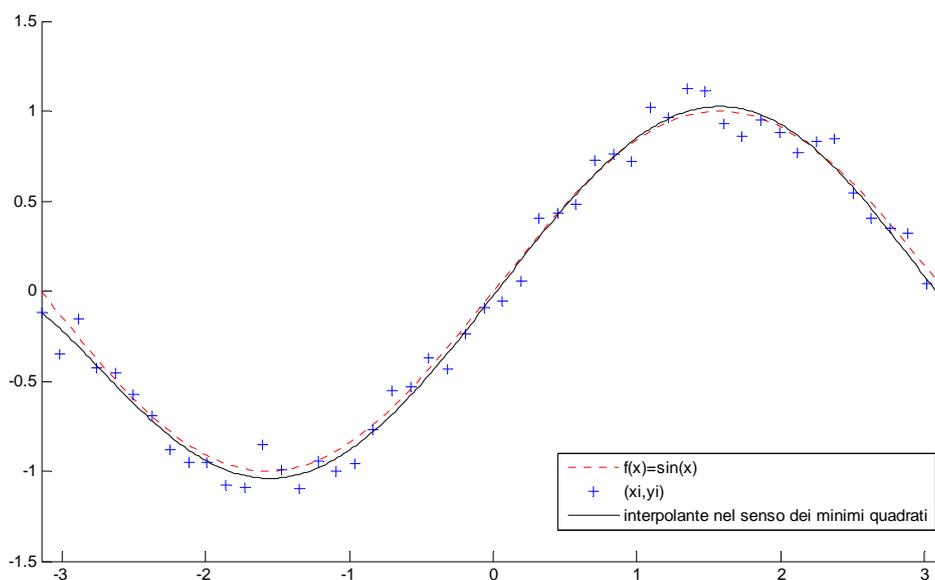
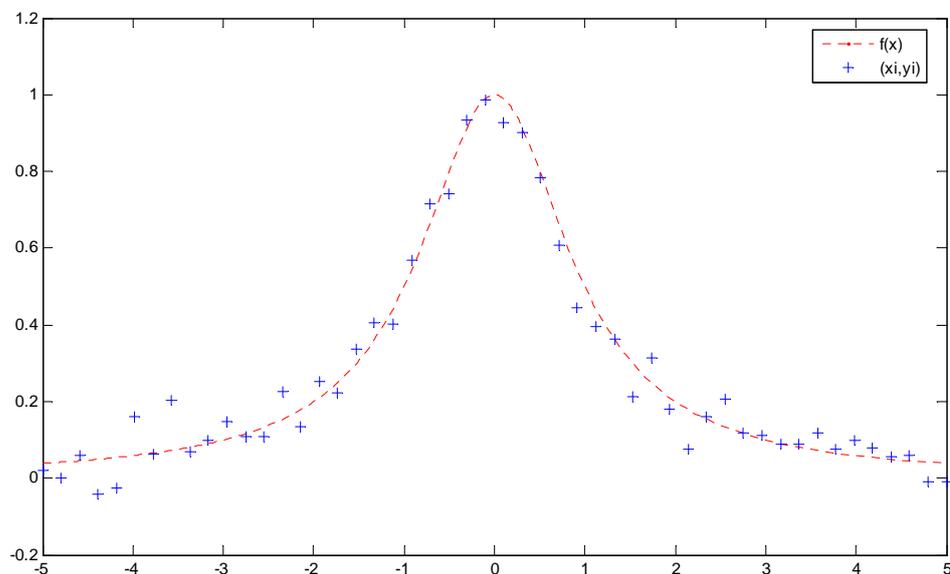


Figura 9 – Polinomio Interpolante ai Minimi Quadrati

## Esempio

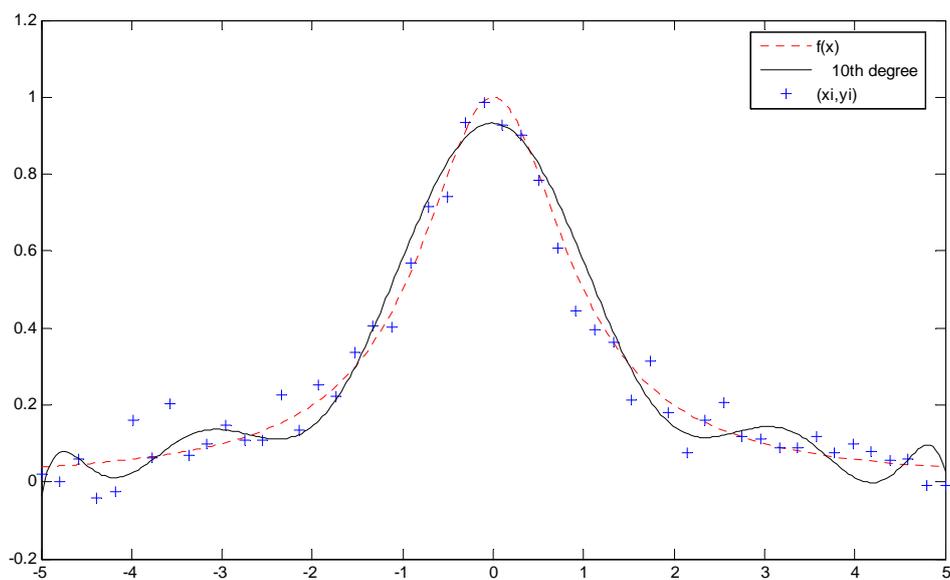
Nel secondo esempio numerico è stato considerato il caso della la funzione di Runge  $f(x) = \frac{1}{1+x^2}$ . Si è

ipotizzato un vettore  $y$  di dati affetti da errore gaussiano:



**Figura 10 – Grafico della funzione  $y = f(x) = \frac{1}{1+x^2}$   $[-5,+5]$**

I risultati della risoluzione del problema di approssimazione nel senso dei minimi quadrati sono mostrati nella figura seguente:



**Figura 11 – Polinomio Interpolante ai Minimi Quadrati**