# Esercitazione 7: Aggiornamento minimi quadrati.

## Esercizio 1

Per calcolare il piano di equazione $p(x) = a + b \cdot xc \cdot y$ che minimizza gli scarti quadratici relativi alle misure $(x_i, y_i, z_i)$ è necessario risolvere il seguente problema di minimo

$$\min_{x=[a,b,c]} ||Ax - z||_2^2$$

dove $A$ è la seguente matrice di Vandermonde

$$A = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix}$$

Risolvere il problema sul seguente insieme di punti

| $x_i$ | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
|---|---|---|---|---|---|---|---|---|---|
| $y_i$ | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 |
| $z_i$ | 1 | 0 | 1 | 2 | 3 | 2 | 3 | 0 | 4 |

## Risultati

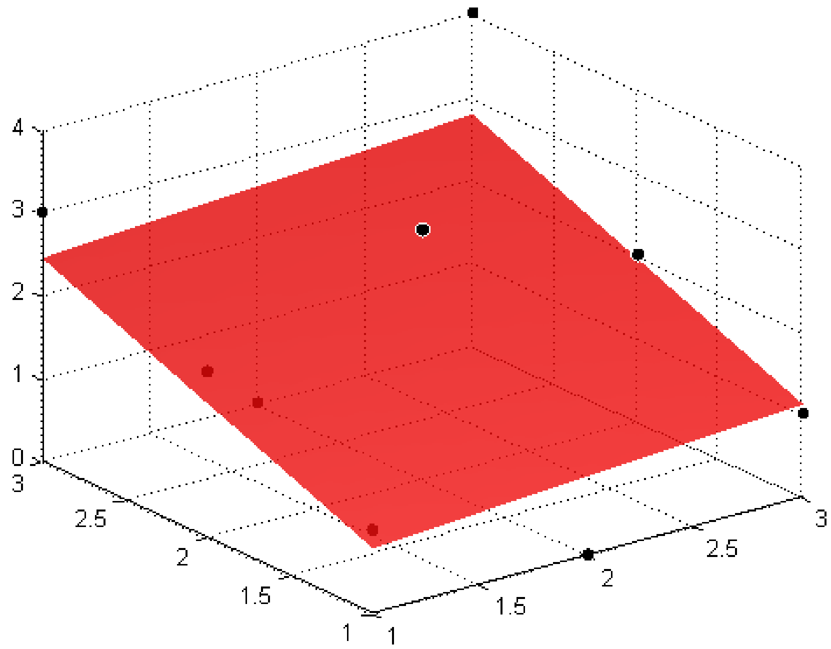


Figura 1: Andamento del piano dei minimi quadrati

## Esercizio 2

Siano dati le seguenti coppie di valori.

| $x_i$ | -5 | -3 | 1 | 3 | 4 | 6 | 8 |
|---|---|---|---|---|---|---|---|
| $y_i = f(x_i)$ | 18 | 7 | 0 | 7 | 16 | 50 | 67 |

Si vuole calcolare il polinomio della forma $p(x) = a + b \cdot x^3$ che meglio approssima (nel senso dei minimi quadrati) l'insieme dei dati del problema.
In questo caso la matrice di Vandermonde assume la forma

$$A = \begin{bmatrix} 1 & x_1^3 \\ 1 & x_2^3 \\ \vdots & \vdots \\ 1 & x_n^3 \end{bmatrix}$$

# Risultati

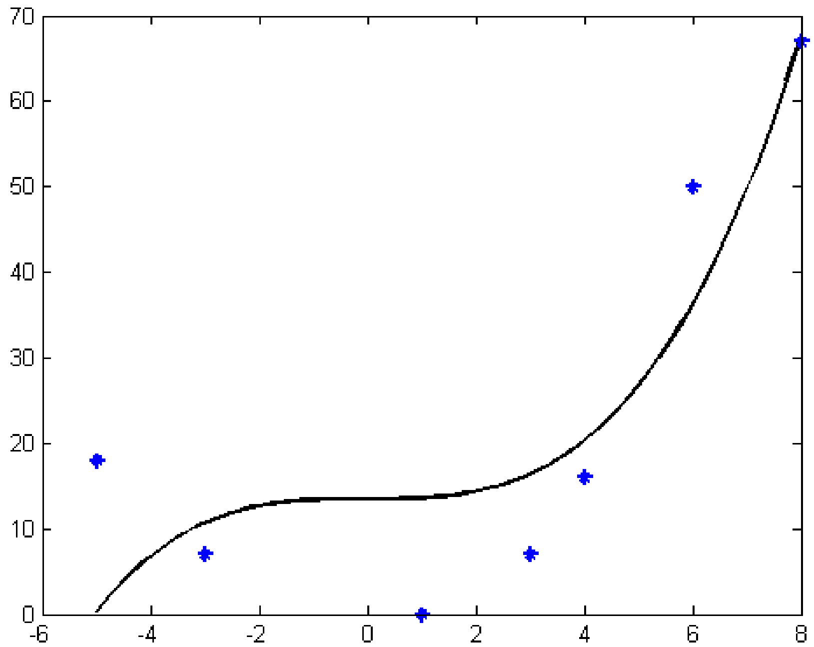


Figura 2: Andamento del polinomio $p(x) = a + b \cdot x^3$ che meglio approssima l'insieme dei dati.

# Esercitazione 8:
# Interpolazione polinomiale.

## Richiami di Teoria: Interpolazione polinomiale mediante forma di Lagrange

Sia data una funzione reale $f(x)$ definita in $I = [a, b]$ e si supponga nota in $n + 1$ punti (detti anche nodi) distinti di $I$, $x_0$, $x_1$, $\ldots$, $x_n$, con $x_i \neq x_j$ per $i \neq j$.
Il problema dell'interpolazione polinomiale consiste nel trovare il polinomio $P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$ tale che

$$P_n(x_i) = f(x_i), \quad 0 \leq i \leq n\,.$$

Esistono vari metodi per costruire il polinomio di interpolazione (unico). In questa sezione considereremo la forma di Lagrange.
Si chiama i–esimo polinomio di Lagrange $L_i^n(x)$ di grado $n$

$$L_i^n(x) = \frac{\prod_{k=0,k\neq i}^{n}(x - x_k)}{\prod_{k=0,k\neq i}^{n}(x_i - x_k)}\,.$$

Pertanto la forma di Lagrange diventa

$$P_n(x) = \sum_{i=0}^{n} L_i^n(x) f(x_i)$$

## Esercizio 1

**Premessa 1:** In questo esercizio vogliamo verificare numericamente la seguente proprietà dei polinomi di Lagrange

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

dove $\delta_{ij}$ è la delta di Kronecker che vale $1$ se $i = j$ e $0$ altrimenti.
**Premessa 2:** Si vuole inoltre verificare che

$$\sum_{i=0}^{n} L_i^n(x_j) = 1$$

cioè formano una partizione dell'unità.
Rappresentare graficamente i polinomi di Lagrange per $n = 1, \ldots, 4$ e la relativa somma. Come punti di interpolazione si utilizzi l'intervallo $[0, 1]$ equispaziato.
Il seguente codice mostra il caso $n = 2$. Si può utilizzare il comando MatLab `subplot` per suddividere una finestra di disegno in altre sottofinestre.

```
% Esempio polinomi di Lagrange di ordine 2.

ngrad = 2;                          % Grado del polinomio di interpolazione
xi = linspace(0,1,ngrad+1)';        % Punti definizione polinomio interpolazione
np = 60;                            % Numero di punti di valutazione dei polinomi
xval = linspace(0,1,np)';           % Coordinate su cui valutare i polinomi
L = zeros(np,ngrad+1);              % La colonna i-esima contiene L_i(xval)
for i=1:ngrad+1
    for p=1:np,
        x = xval(p);
        Lval = 1;
        for j=1:ngrad+1
            if(j ≠ i)
                Lval = Lval*(x-xi(j))/(xi(i)-xi(j));
            end
        end
        L(p,i) = Lval;
    end
end

figure(1);
h = plot(xval,L);
set(h,'LineWidth',2);

figure(2);
plot(xval,sum(L,2))
```

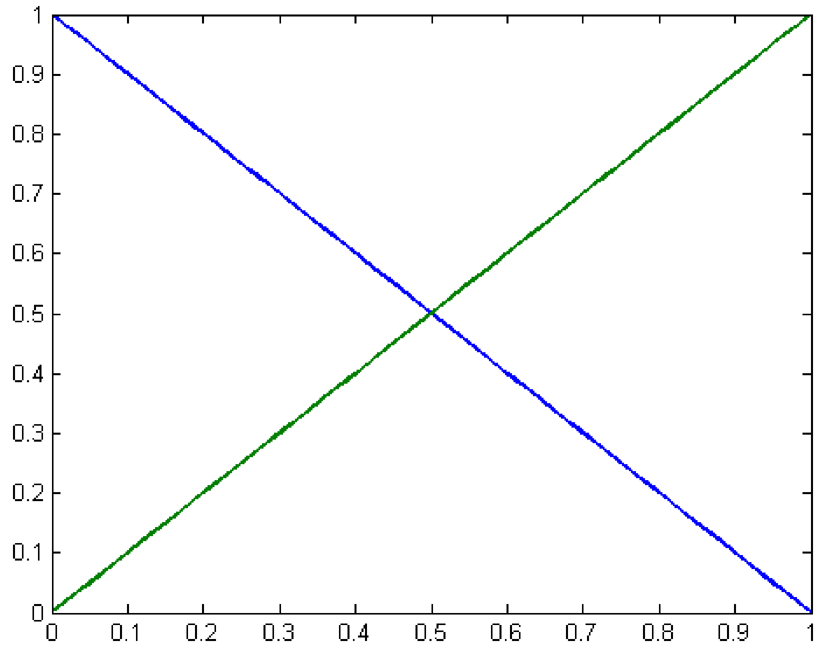


Figura 3: Polinomi di interpolazione di grado 1.

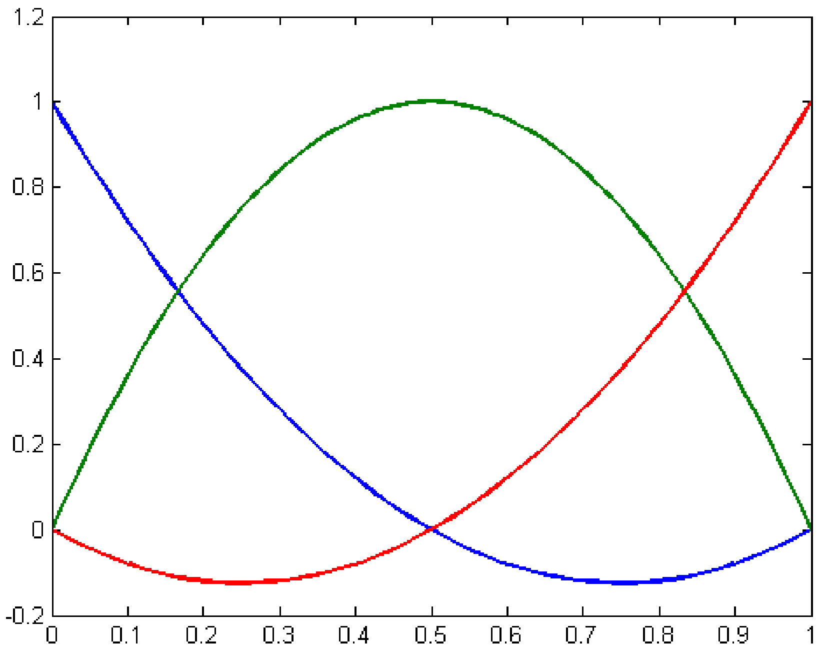


Figura 4: Polinomi di interpolazione di grado 2.

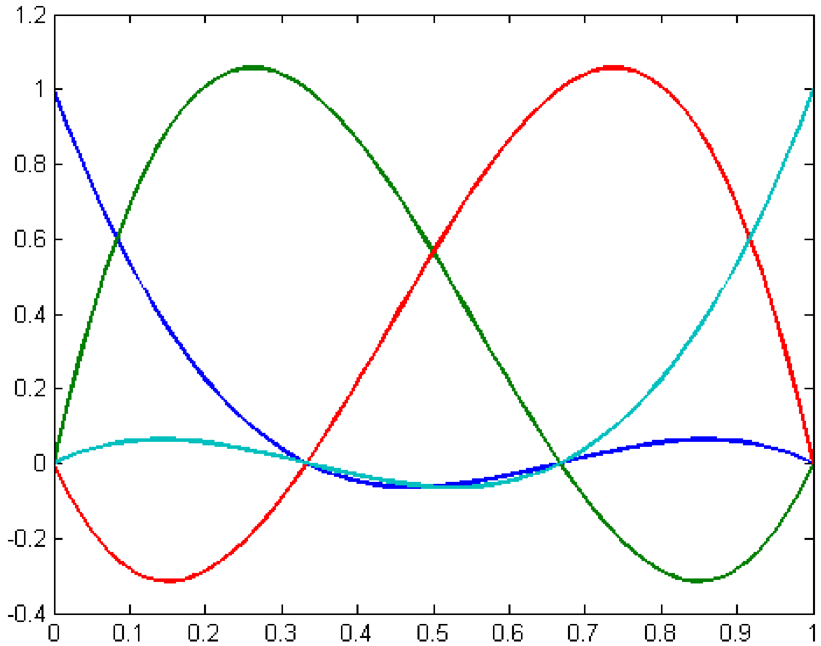


Figura 5: Polinomi di interpolazione di grado 3.

# Esercizio 2

Dati i seguenti nodi ed i relativi valori di una certa funzione $f(x)$
valutare (graficamente) il polinomio di interpolazione di Lagrange di $P_3(x)$ nell'intervallo

| $x$ | 5 | -7 | -6 | 0 |
|---|---|---|---|---|
| $f(x)$ | 1 | -23 | -54 | -954 |

$[-7, 5]$.
Per far questo bisogna utilizzare la formula

$$P_n(x) = \sum_{i=0}^{n} L_i^n(x) f(x_i)$$

che può essere vista come il prodotto matrice/vettore tra la matrice $L$ delle base polinomiali e il vettore dei nodi di inerpolazione $y = f(x_i)$.

# Risultati

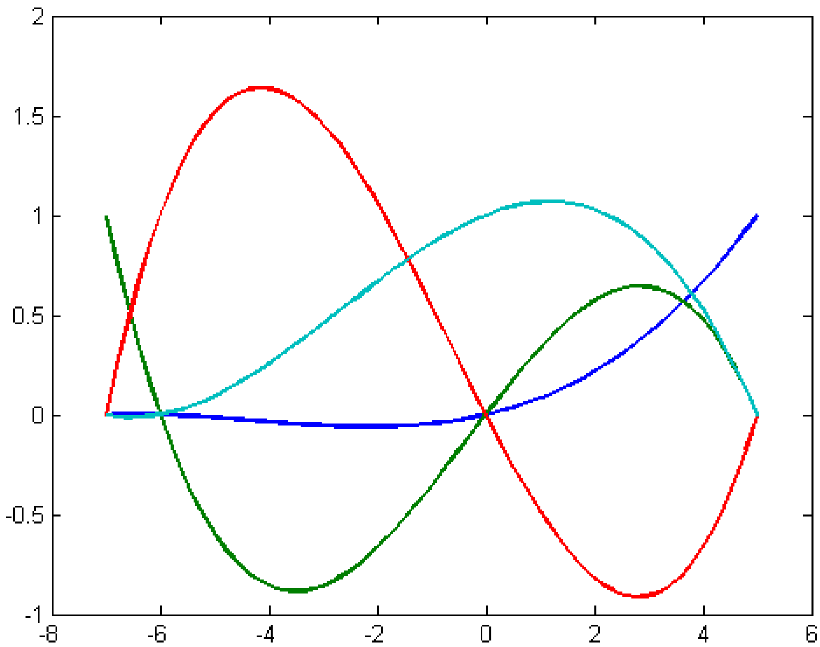


Figura 6: Base polinomiale del problema.

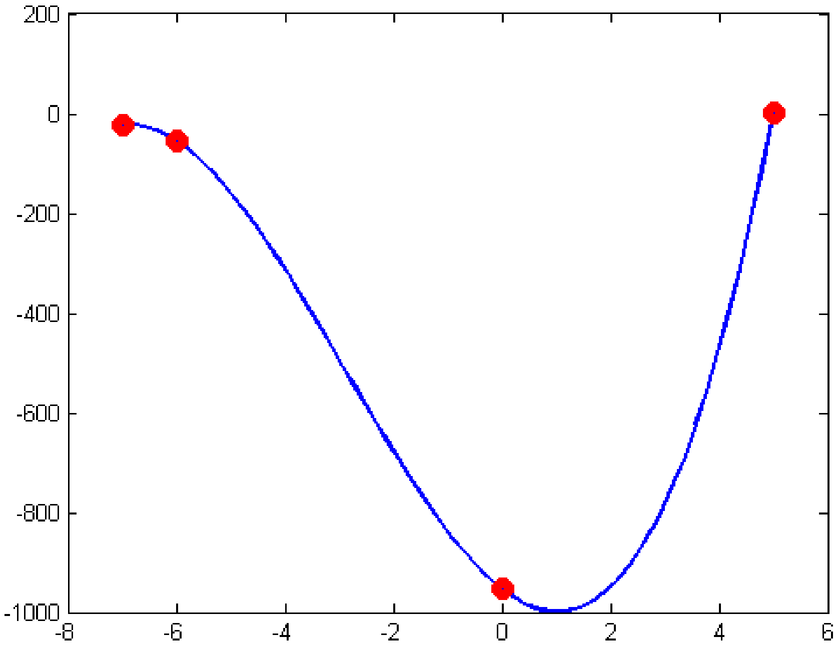


Figura 7: Polinomio di interpolazione.

## Richiami di Teoria: Interpolazione polinomiale mediante funciton MatLab `polyfit`

La function MatLab che implementa l'interpolazione polinomiale è `polyfit` mediante la sintassi `p=polyfit(x,y,n)` dove `n` è il grado del polinomio con cui vogliamo interpolare i dati e restituisce in uscita un vettore dei coefficienti del polinomio. Nel nostro caso sarà `n=length(x)-1`. I coefficienti del polinomio `p` sono da interpretarsi in ordine descrescente il cui primo coefficiente è relativo al monomio di grado massimo, mentre l'ultimo è relativo al termine costante. Per valutare un polinomio su un insieme di dati si utilizza la routine `polyval`.
La function `polyfit` risolve il problema del calcolo del polimonio di interpolazione mediante fattorizzione QR applicata alla minimizzazione di $||Ax - b||_2^2$ con $A$ matrice di Vandermonde che in questo caso risulta quadrata e di rango massimo e $b$ il vettore dei punti di interpolazione.

## Esercizio 3

Ripetere l'esercizio 2 utilizzando le funciton MatLab `polyfit`.

# Richiami di Teoria: Fenomeno di Runge

Se si ha una funzione $f(x)$ continua in un intervallo $[a, b]$ ed in tale intervallo si calcolano i poliniomi di interpolazione di grado via via maggiore sembrerebbe naturale aspettarsi che la successione di tali polinomi converga uniformemente ad $f(x)$ in $[a, b]$, ovvero $\lim_{n\to\infty} ||f - P_n||_\infty = 0$.
Nella realtà, per la maggior parte delle funzioni continue, ciò non è vero. Un esempio è fornito dalla funzione di Runge.

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

nell'intervallo $[-5, 5]$.
Come si può verificare (Esercizio 4) il polinomio di interpolazione ha un minimo nel punto medio dell'intervallo e delle ampie oscillazioni attorno gli estermi. Pertanto non è consigliabile usare punti equidistanti per interpolare la $f(x)$ in tutto $I$ con polinomi di grado elevato. Altre, scelte di punti, per esempio i nodi di Chebyshev, sono quasi ottimali nel senso che si ha convergenza uniforme per tutte le funzioni di classe $\mathcal{C}^1(I)$ se la successione $\{P_n(x)\}$ di polinomi è costruita su questi nodi.
I nodi di Chebyshev-Gauss-Lobatto (scalati) sono definiti nel seguente modo: fissato $N$, i punti $x_k$ $(N + 1)$ sono dati da

$$x_k = \frac{a+b}{2} - \frac{b-a}{2}\cos\left(\frac{k\pi}{N}\right) \quad \text{per } k = 0, \ldots, N$$

con $a$ e $b$ estremi dell'intervallo.

# Esercizio 4

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 (utilizzare `polyfit`).
3. Si testi il codice, producendo dei grafici relativo alla funzionei di Runge nell'intervallo $[-5, 5]$, nel caso di $n$ punti equispaziati ($n$ varia da 6 punti equispaziati a $11$ punti equispaziati) e $n$ nodi di Chebyshev;
4. si calcoli gli errori in norma infinito tra la funzione e le interpolanti.

# Risultati

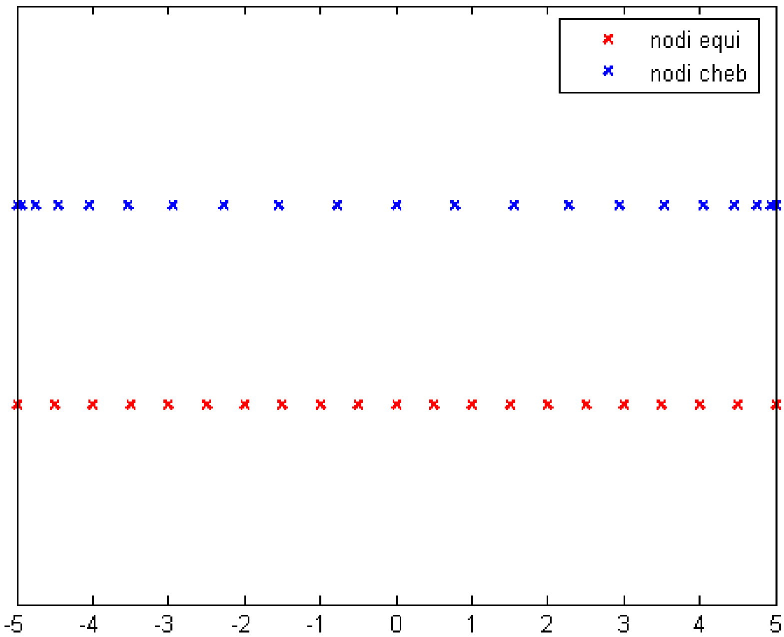


Figura 8: Punti di Chebichev e punti equispaziati (21 punti).

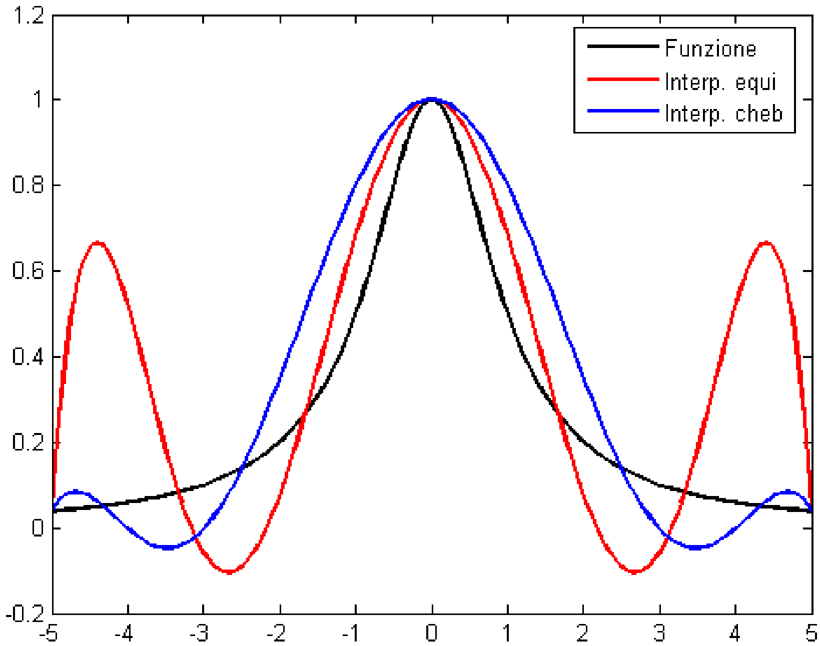


Figura 9: Interpolazione con 7 punti.

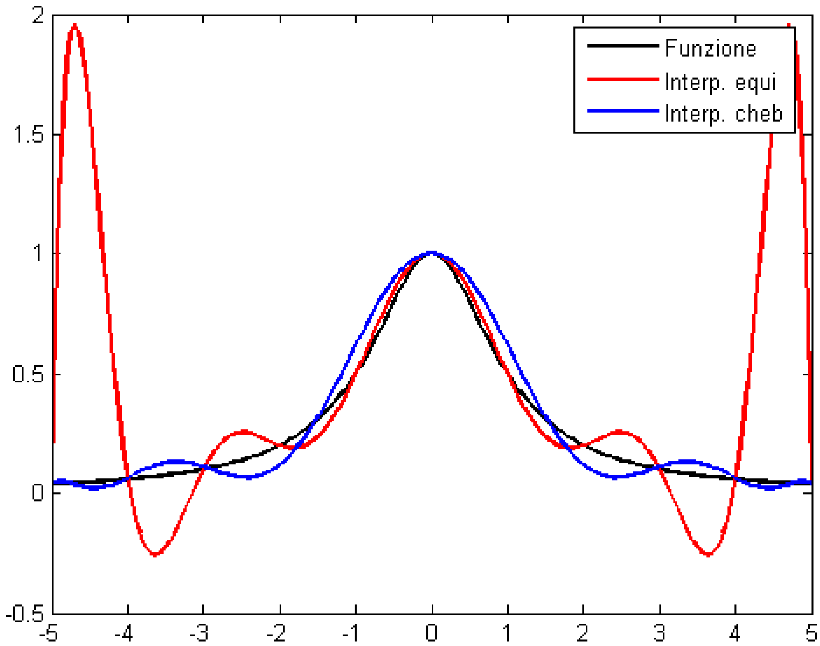


Figura 10: Interpolazione con 11 punti.

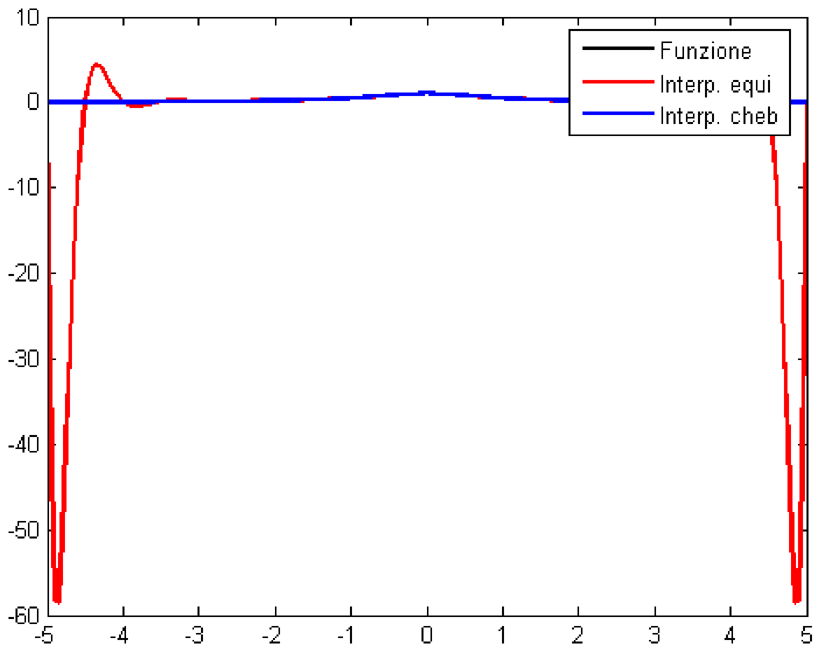


Figura 11: Interpolazione con 21 punti.

# Esercizio 5

Ripetere l'esercizio 4 nel caso della funzione di Runge nell'intervallo $[-1, 1]$.

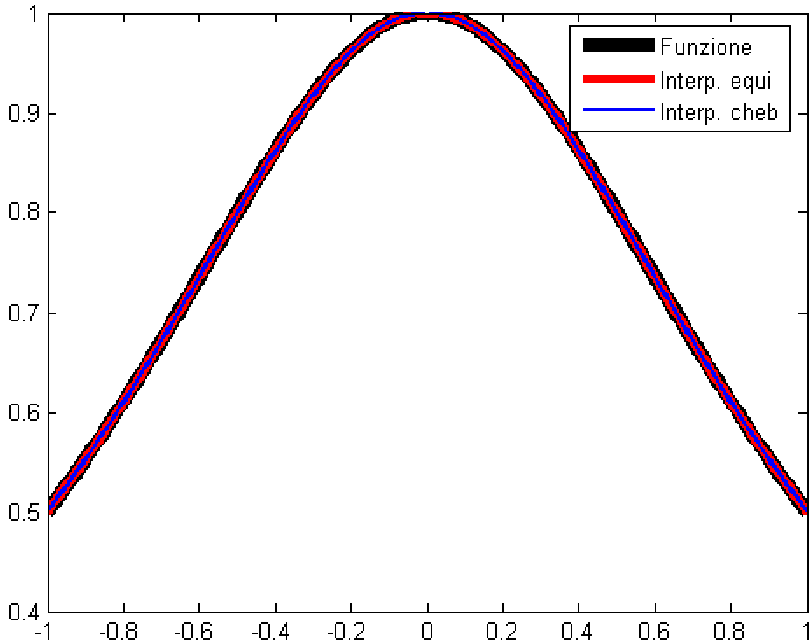


Figura 12: Interpolazione con 21 punti.

# Richiami di Teoria: Spline

Si è visto che nel caso dell'interpolazione polinomiale, dati $N+1$ punti $a = x_0 < \ldots < x_N = b$, e i valori $y_0, \ldots, y_N$ assunti da una funzione $y = f(x)$, esiste uno ed un solo polinomio $p_N$ di grado $N$ tale che

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

Nel caso di nodi equispaziati

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

al crescere di $N$, non si può garantire (in generale) che $||f(x) - p_n(x)||_\infty$ tenda a $0$ (si ricordi il fenomeno di Runge!).
Sorge spontaneo chiedersi se qualora si possegga un gran numero di punti, anche equispaziati, sia possibile calcolare un'approssimante di tipo polinomiale per cui al crescere di $N$ si abbia $p_N \to f$.
Una risposta è stata data nel 1946 da Schoenberg, lo scopritore delle splines.
Il primo caso è quello delle splines di grado $1$, cioè funzioni che in ogni intervallo $[x_i, x_{i+1}]$ (per $i = 0, \ldots, N-1$) sono polinomi di grado $m = 1$ e globalmente funzioni di classe $C^{m-1} = C^0$, cioè continue.
Il caso generale di splines di grado $m$ risulta più complicato. Un esempio notevole è quello delle splines cubiche $s_3$, cioè funzioni che in ogni intervallo $[x_i, x_{i+1}]$ (per $i = 0, \ldots, N-1$) siano polinomi di grado $m = 3$ e globalmente funzioni di classe $C^{m-1} = C^2$.
Osserviamo infatti che in ogni intervallo $[x_i, x_{i+1}]$ le spline si possano rappresentare come

$$s_3(x) = c_{1,i} + c_{2,i}(x - x_i) + c_{3,i}(x - x_i)^2 + c_{4,i}(x - x_i)^3, \ i = 0, \ldots, N-1$$

e quindi per determinare $s_3$ in $\{x_i\}_{i=0,\ldots,N}$ servano $4N$ valori $c_{i,j}$. Da ragionamenti sulle proprietà della regolarità della spline interpolante si vede che sono disponibili solo $4N-2$ condizioni (di cui $N+1$ dal fatto che $s_3(x_i) = f_i$). Si procede richiedendo quindi una delle seguenti proprietà aggiuntive a $s_3$:

- *Spline naturale*: $s_3^{(2)}(a) = s_3^{(2)}(b) = 0$.
- *Spline periodica*: $s_3^{(1)}(a) = s_3^{(1)}(b)$, $s_3^{(2)}(a) = s_3^{(2)}(b)$.
- *Spline vincolata*: $s_3^{(1)}(a) = f^{(1)}(a), \ s_3^{(1)}(b) = f^{(1)}(b)$.

Per altri tipi di splines come le *not a knot* o le *Hermite* si veda la dispensa "Appunti di Calcolo Numerico" di S. De Marchi, pagg. 142 e seguenti.

# Esercizio 6

Fissato $N = \{11, 101\}$ si calcoli l'interpolante spline lineare $s_1$ e cubica $s_3$ della funzione di Runge nei $N$ nodi equispaziati dell'intervallo $[-5, 5]$.

Confrontare il valore dell'interpolante spline (calcolo dell'errore assoluto) con quello della funzione di Runge utilizzando un maggior numero di nodi diciamo $401$, questo perchè ricordiamo che l'errore di interpolazione è nullo nei nodi di interpolazione.
Utilizzare i seguenti comandi MatLab `interpr1` e `spline` rispettivamente per l'interpolazione linare e cubica.

## Risultati

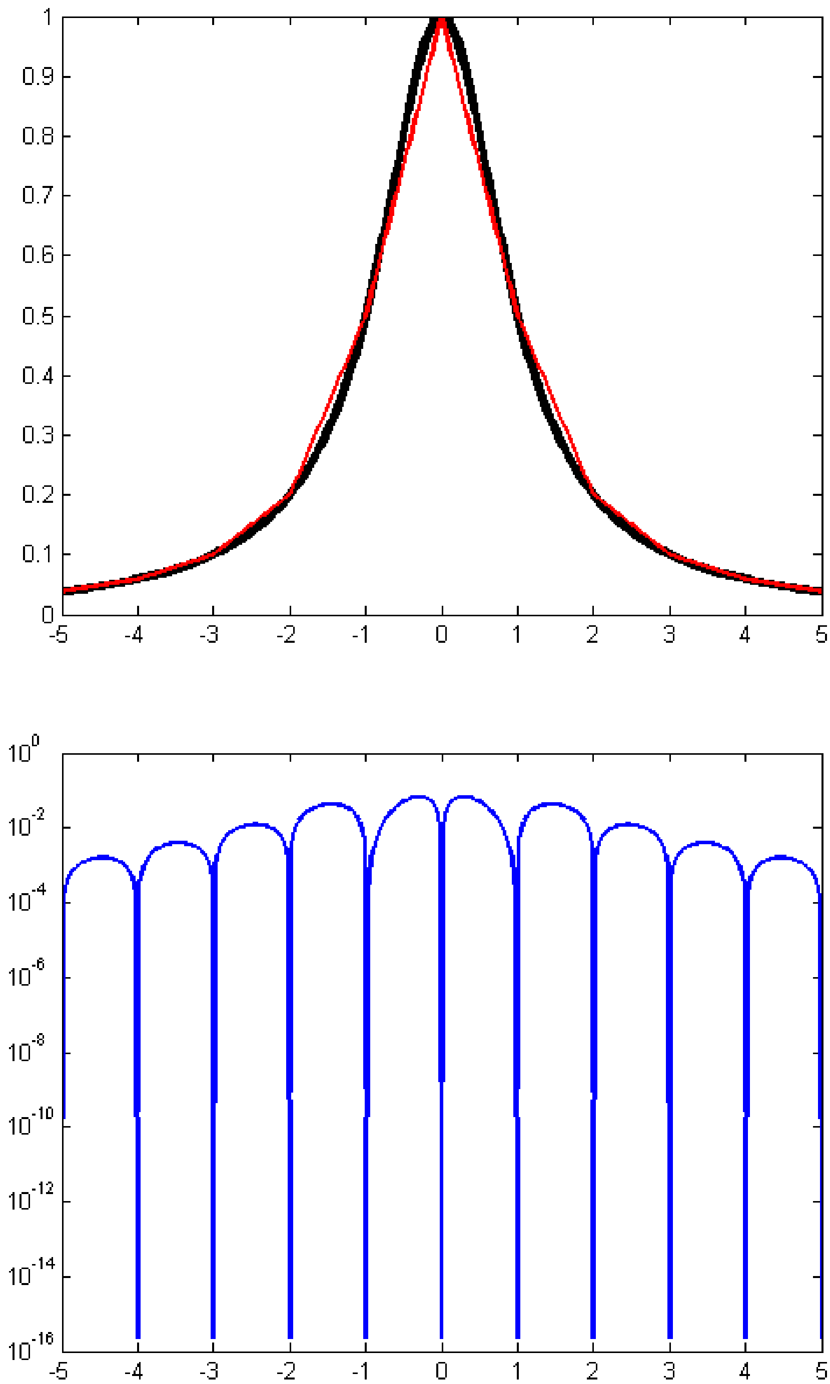


Figura 13: Interpolazione lineare con 11 punti equispaziati e relativo errore.

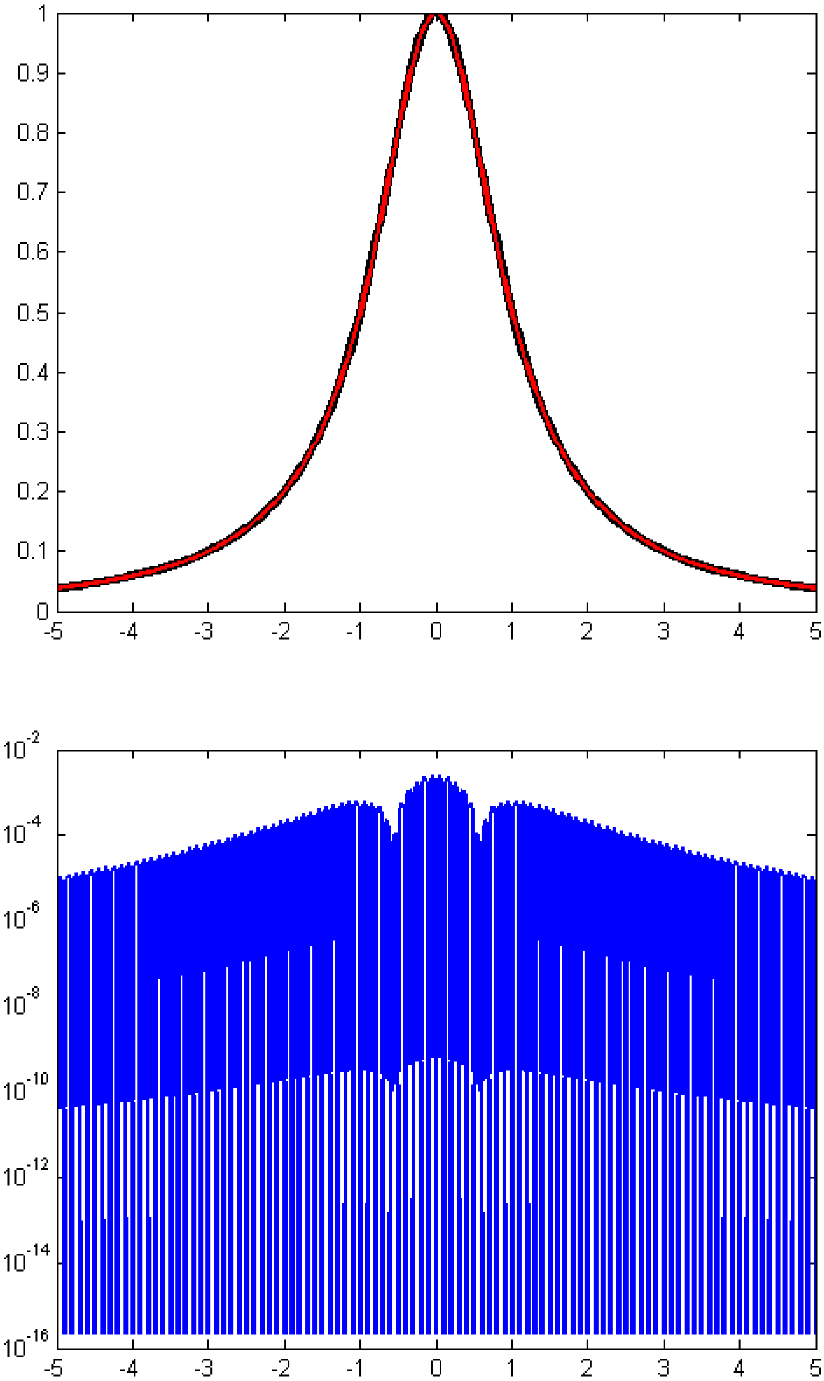


Figura 14: Interpolazione lineare con 101 punti equispaziati e relativo errore.

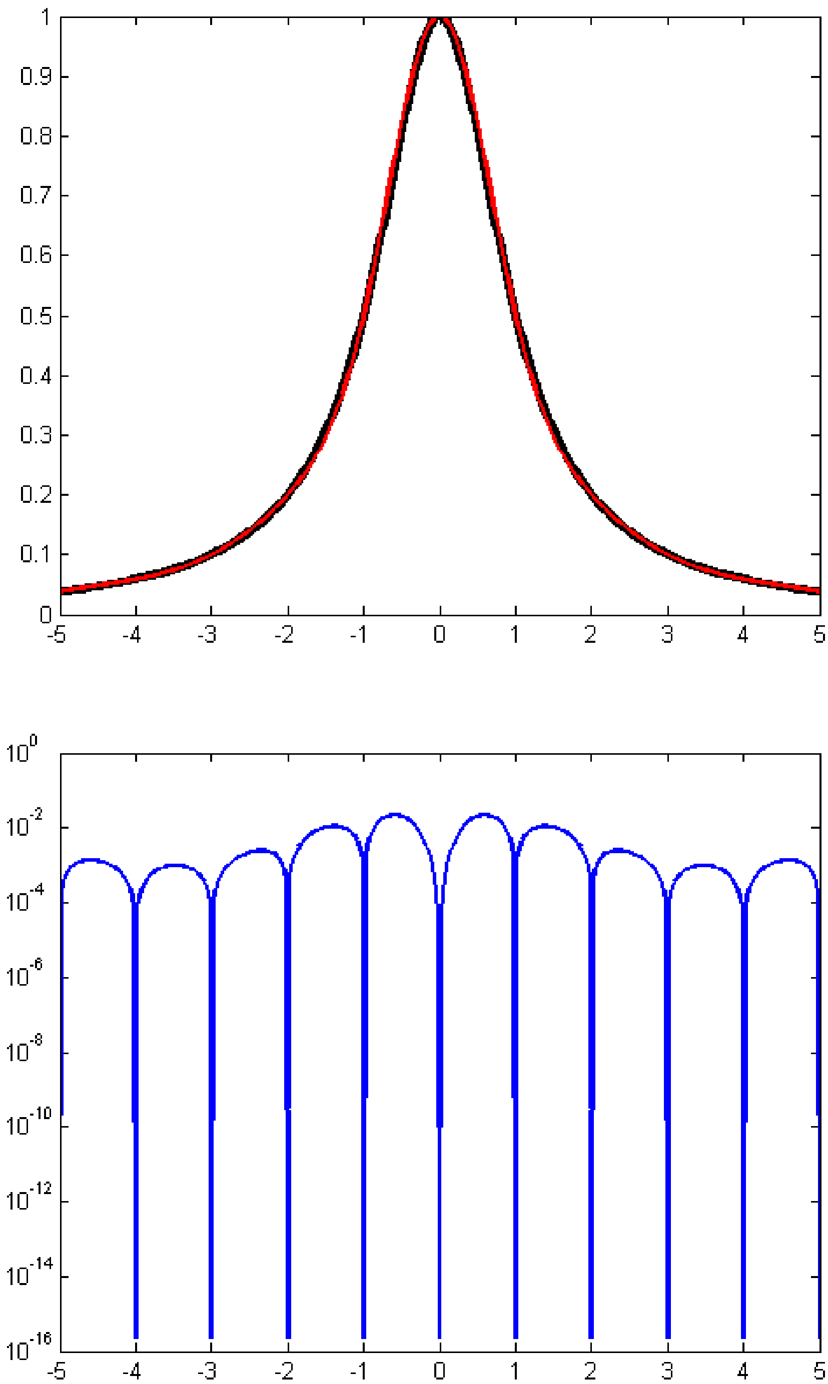


Figura 15: Interpolazione cubica con 11 punti equispaziati e relativo errore.

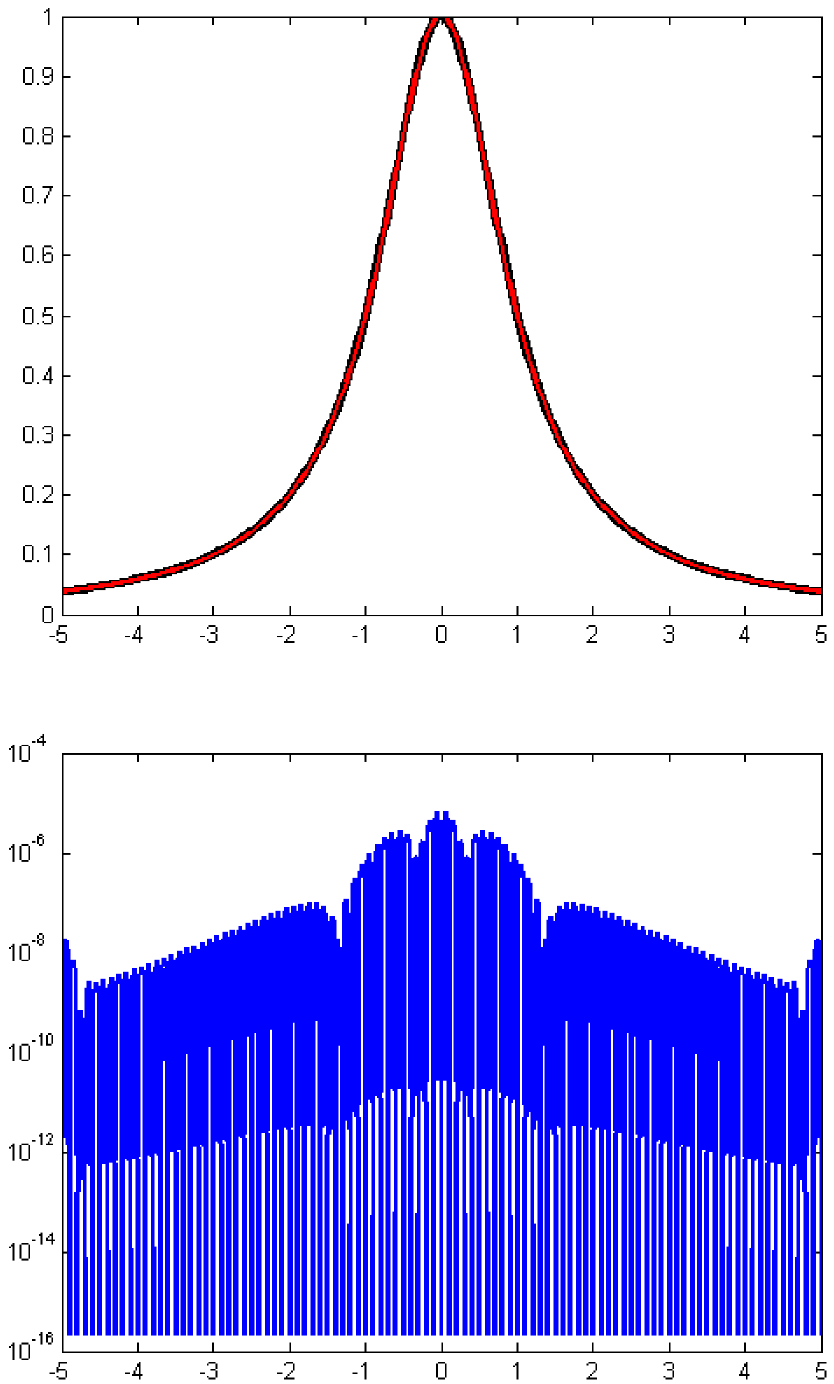


Figura 16: Interpolazione cubica con 101 punti equispaziati e relativo errore.

# Esercizio 7[1]

Al sito

`http://antoine.frostburg.edu/chem/senese/javascript/water-density.html`

viene proposto una pagina web con il calcolo della densità dell'acqua.
Se si va a vedere i sorgenti della pagina si vede che viene utilizzata l'interpolazione polinomia su un insieme di 5 punti centrati nella temperatura in cui si vuole calcolare il valore.
Dopo alcune prove si vede che la tabella delle densità varia da $-10$ [C] a $109$ [C] e pertanto il range di validità dell'interpolazione, utilizzando la tecnica precedente, va da $-8$ [C] a $107$ [C].
Si vuole confrontare l'interpolazione polinomiale basata su 5 punti e spline sul valore $1.4142$ [C].

# Risultati

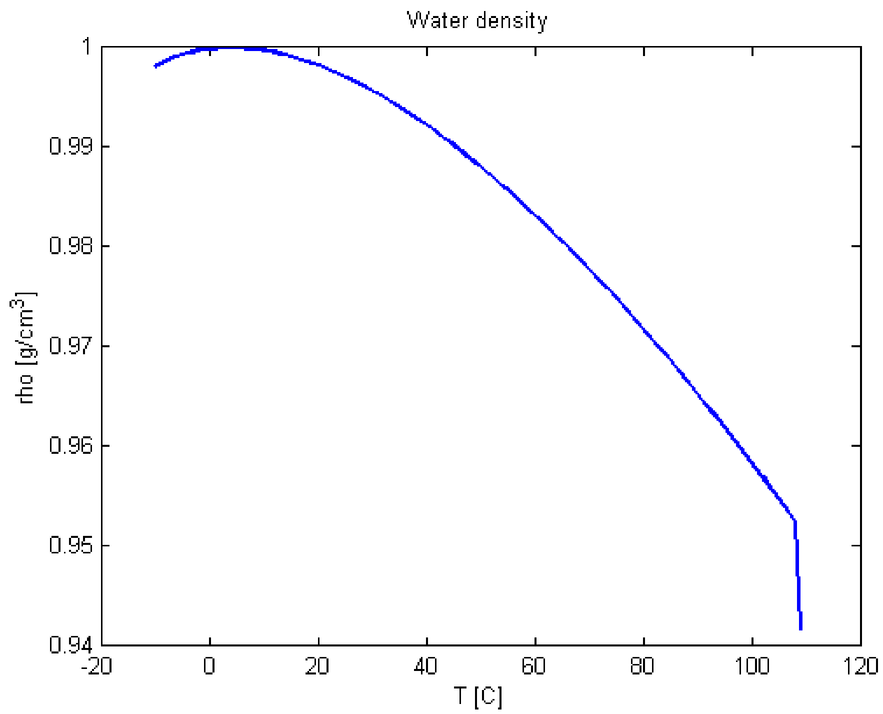


Figura 17: Andamento della densità

```
Inserisci temperatura da calcolare: 1.4142
rho   0.99992075 [g/cm^3] in T  1.41  [C] (interp. 5 punti)
rho   0.99992074 [g/cm^3] in T  1.41  [C] (spline)
```

[1]Facoltativo.