VI esercitazione di
Laboratorio Computazionale Numerico

1. I seguenti dati fanno riferimento alla portata di un canale misurata mensilmente in m^3/sec

mese	G	F	M	A	M	G	L	A	S	O	N	D
portata	12.51	13.05	11.7	9.26	8.3	6.25	5.34	4.59	5.14	6.36	10.31	13.88

Costruire, in un piano cartesiano con assi opportuni (utilizzando le opzioni della pagina grafica di Matlab) un grafico dei dati raccolti, della polinomiale lineare a tratti interpolante, del polinomio di interpolazione di grado ≤ 11 e della spline cubica naturale interpolante, ottenuti mediante comandi Matlab.

2. Si consideri la funzione $f(x) = -\dfrac{(x-5)\log x}{x^2}$ e l'insieme di nodi $xx = 1 : 0.5 : 5$. Dopo aver determinato le immagini tramite f di tali nodi, si utilizzino i comandi *interp1* e *spline* per interpolare tali punti determinando la migliore tra le due approssimazioni.

3. Si consideri il seguente insieme di punti

$$\{(-1, 0.5), (-0.5, 0.8), (0, 1), (0.5, 0.8), (1, 0.5)\}$$

Si utilizzi il comando *csape* per determinare la spline interpolante tali punti

- vincolata con derivate agli estremi rispettivamente $\dfrac{1}{2}$ e $-\dfrac{1}{2}$
- del tipo not-a-knot

Si confrontino infine i due grafici.

4. Dopo aver campionato la funzione $f(x) = \sin x$ su una decomposizione uniforme dell'intervallo $[0, 2\pi]$, si calcolino le spline, periodica e naturale, interpolanti tali punti determinando la migliore tra le due approssimazioni.

5. Si introducano i polinomi

$$p_1(x) = 2x^4 - 3x^3 + 4x - 7\,,\; p_2(x) = 3x^2 - 2x - 5$$

Si effettui somma, differenza, prodotto e divisione tra polinomi.

6. Si consideri la funzione $f(x) = e^{-x^2}$ nell'intervallo $[-3, 3]$. Si realizzi un vettore contenente 9 nodi equidistanti e un vettore contenente le relative immagini; si costruisca il polinomio di grado minimo interpolante tali punti. Si ripetano le stesse operazioni su una decomposizione di 9 nodi di Chebychev[1]. Infine si disegnino i grafici della funzione f e dei due polinomi interpolanti e si determini la migliore tra le due approssimazioni.

---

[1] I nodi di Chebychev sull'intervallo generico [a,b] sono definiti dalla formula

$$\frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\left(\frac{2i-1}{2n}\pi\right) \qquad 1 \leq i \leq n.$$

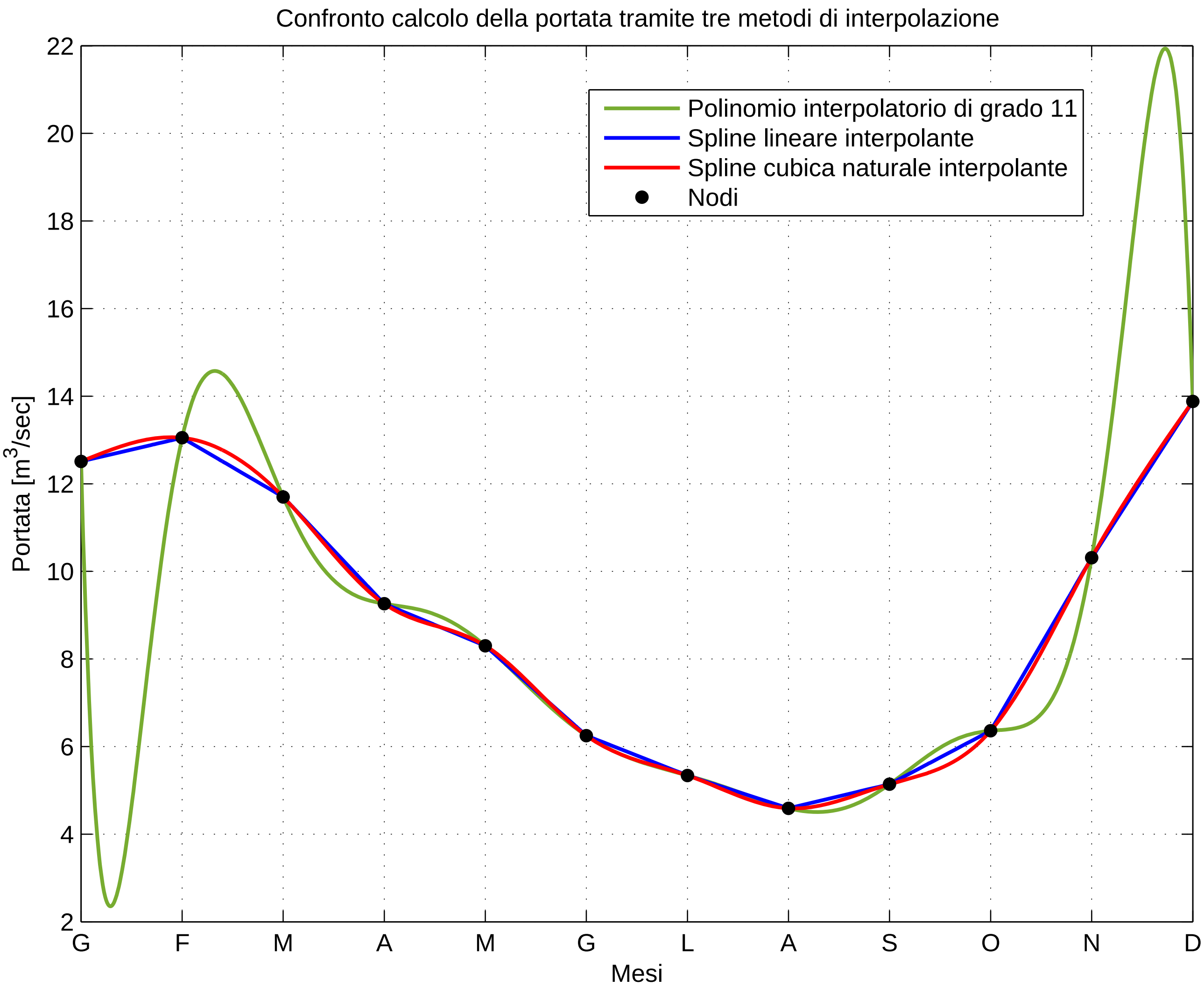
7. Si consideri la funzione $f(x) = \sin x + \log x - 1$ sull'intervallo $[1, 9]$. Si consideri il vettore dei nodi $xx = [1.5, 4, 5, 5.5, 8]$ e se ne calcoli l'immagine tramite f. Utilizzando il comando Matlab *polyfit* si calcoli il polinomio p interpolante i punti (xx_i, yy_i), con $i = 1, \dots, 5$. Si realizzi poi una figura nella quale tracciare i grafici di f e di p; si marchino anche i punti interpolati.

8. Data la funzione di Runge

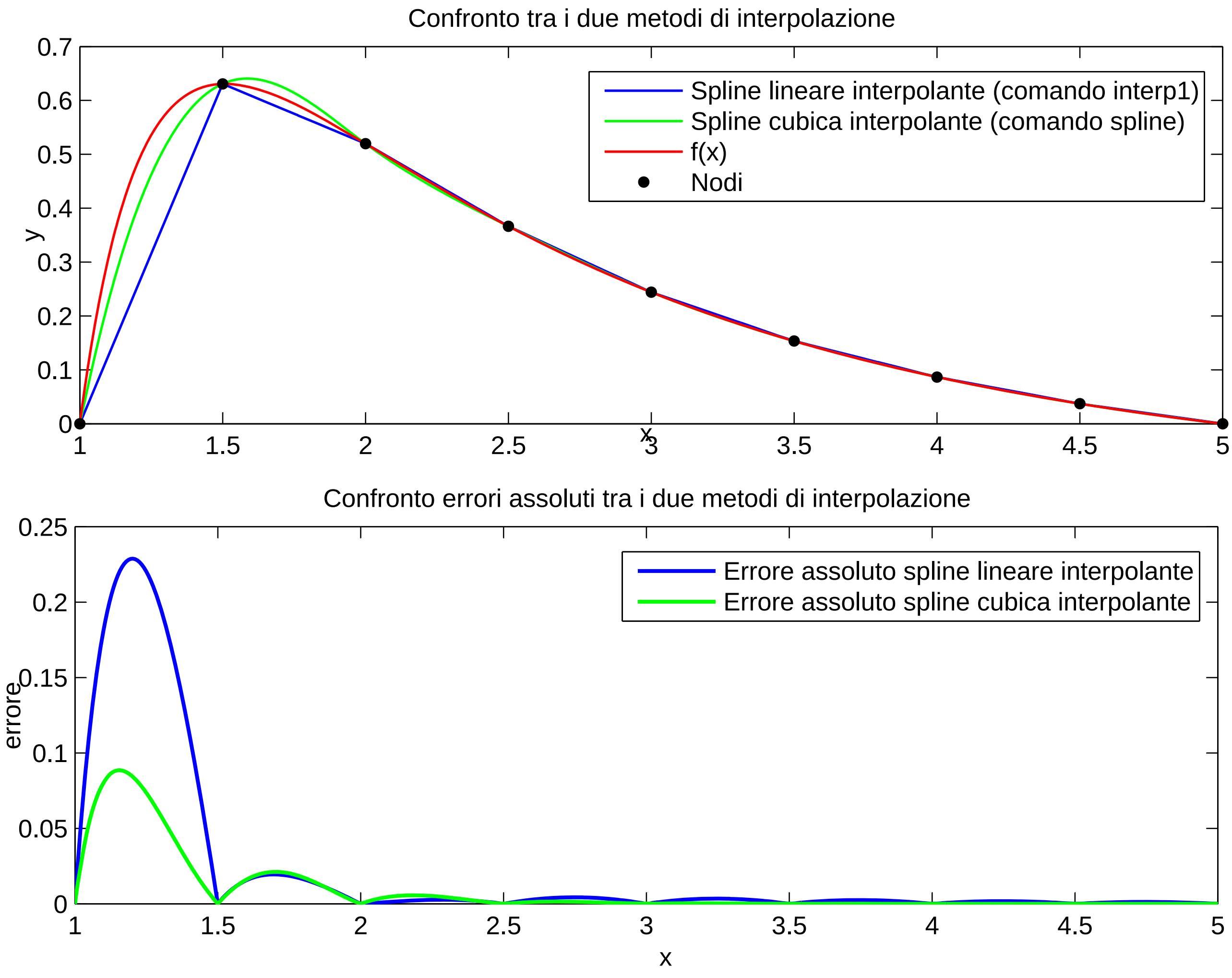
$$f(x) = \frac{1}{1 + 25x^2} \qquad x \in [-1, 1]\,.$$

Mostrare l'andamento di tale funzione insieme al grafico dei polinomi interpolatori di grado N = 9; 10; 11; 12 con nodi equispaziati.

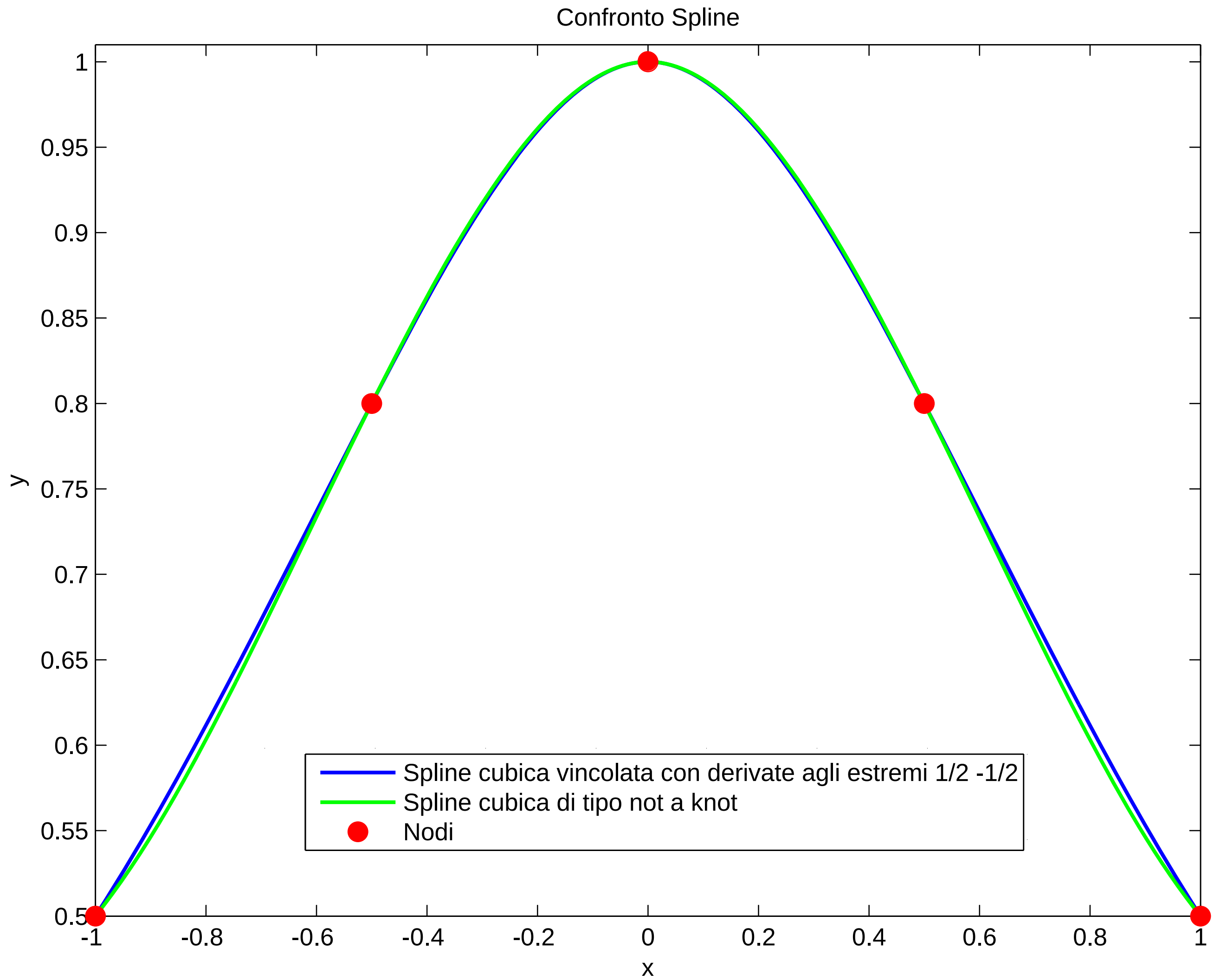
```
% Esercizio 1 - Esercitazione 6
clear all
close all
clc
% mesi (nodi del polinomio)
m=1:12;
% rilevamenti portata del canale
p=[12.51 13.05 11.7 9.26 8.3 6.25 5.34 4.59 5.14 6.36 10.31 13.88];
% vettore in cui valutare la funzione
x=linspace(1,12,1000);

% valori della spline lineare(primo grado) interpolante
y_sp_lin=interp1(m,p,x);

% valori del polinomio di interpolazione di grado <= 11
% coefficienti polinomio
coeff_11=polyfit(m,p,11);
% valori polinomio
y_pol11=polyval(coeff_11,x);
% calcoliamo la spline cubica con la condizione di naturalita
sp_cub_nat=csape(m,p,'variational');
% valori polinomio
y_sp_cub_nat=ppval(sp_cub_nat,x);
% Grafico
figure
hold on
box on
plot(x,y_pol11,x,y_sp_lin,x,y_sp_cub_nat,m,p);
```



```
% Esercizio 2 - Esercitazione 6
clear all
close all
clc
% nodi
xx=1:0.5:5;
% vettore in cui valutare la funzione
x=linspace(1,5,1000);
% immagine tramite f dei nodi xx
y=-(((xx-5).*log(xx))./xx.^2);

% valori della spline lineare
y_spl_lin=interp1(xx,y,x);
% valori della spline cubica
y_spl_cub=spline(xx,y,x);
% valore della funzione
y_esa=-(((x-5).*log(x))./x.^2);

% errori assoluti
err_ass_spl_lin=abs(y_spl_lin-y_esa);
err_ass_spl_cub=abs(y_spl_cub-y_esa);

% Grafico
figure
subplot(2,1,1)
hold on
box on
plot(x,y_spl_lin,x,y_spl_cub,x,y_esa,xx,y)
subplot(2,1,2)
plot(x,err_ass_spl_lin,x,err_ass_spl_cub)
% Dal grafico di confronto degli errori assoluti tra i due metodi di
% interpolazione si può affermare che il miglior metodo di approssimazione
% è la spline cubica interpolante (comando spline).
```

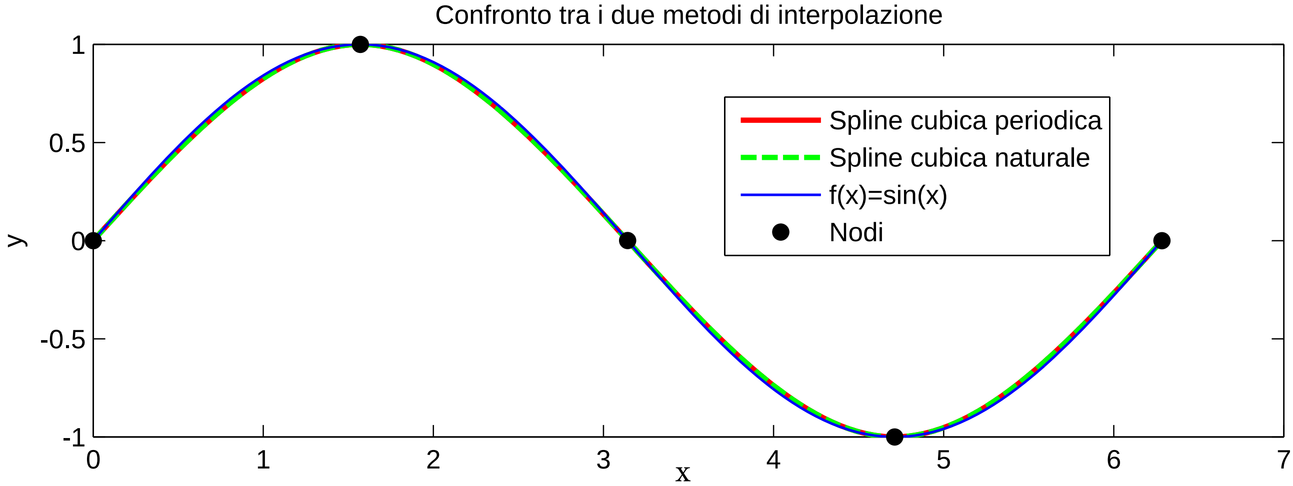


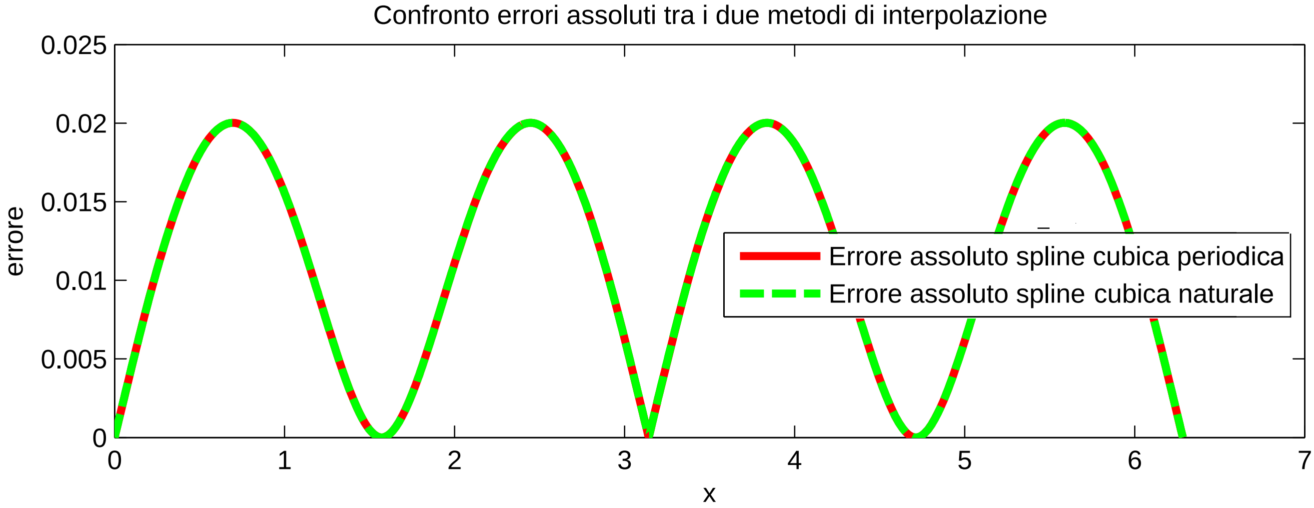
```
% Esercizio 3 - Esercitazione 6
clear all
close all
clc
% punti di interpolazione
x=[-1 -0.5 0 0.5 1];
% valori nei punti di interpolazione
y=[0.5 0.8 1 0.8 0.5];

% vettore di punti per valutare la spline
xx=linspace(-1,1,1000);

% spline cubica vincolata con derivate agli estremi 1/2 -1/2
spl_cub_vin=csape(x,y,[1/2 -1/2]);
% spline cubica di tipo not a knot
spl_cub_nak=csape(x,y,'not-a-knot');

% valori della spline cubica spl_cub_vin nei punti xx
y_spl_cub_vin=ppval(spl_cub_vin,xx);
% valori della spline cubica spl_cub_nak nei punti xx
y_spl_cub_nak=ppval(spl_cub_nak,xx);

figure
hold on
box on
% Grafico
plot(xx,y_spl_cub_vin,xx,y_spl_cub_nak,x,y)
```



```
% Esercizio 4 - Esercitazione 6
clear all
close all
clc
% nodi decomposizione uniforme
x=linspace(0,2*pi,5);
% immagine della funzione nei nodi x
y=sin(x);
% vettore di punti per valutare la spline
xx=linspace(0,2*pi,1000);
% funzione esatta
y_esa=sin(xx);

% spline cubica periodica
spl_cub_per=csape(x,y,'periodic');
% spline cubica naturale
spl_cub_nat=csape(x,y,'varitional');
% valori della spline cubica periodica nei punti xx
y_spl_cub_per=ppval(spl_cub_per,xx);
% valori della spline cubica naturale nei punti xx
y_spl_cub_nat=ppval(spl_cub_nat,xx);

% calcolo errore assoluto della spline cubica periodica
err_ass_spl_per=abs(y_esa-y_spl_cub_per);
% calcolo errore assoluto della spline cubica naturale
err_ass_spl_nat=abs(y_esa-y_spl_cub_nat);


% Creazione del Grafico
figure
subplot(2,1,1)
hold on
box on
plot(xx,y_spl_cub_per,xx,y_spl_cub_nat,xx,y_esa,x,y)
subplot(2,1,2)
plot(xx,err_ass_spl_per,xx,err_ass_spl_nat)
% esaminando il grafico di confronto delle spline possiamo affermare che le
% spline prese in esame approssimano in modo equivalente la funzione data.
```

Grafico Esercizio 4 - Esercitazione 6





```
% Esercizio 5 - Esercitaione 6
diary esercizio_5.txt
clear all
close all
clc
% vettore dei coefficienti del polinomio 1
p1= [2 -3 0 +4 -7];
% vettore dei coefficienti del polinomio 2
p2= [0 0 3 -2 -5];

% somma tra polinomi
som=p1+p2
% differenza tra polinomi
diff=p1-p2
% prodotto tra polinomi tramite funzione matlab conv
prod=conv(p1,p2)
% divisione tra polinomi tramite funzione matlab deconv
format rat
[q r]=deconv(p1,[3 -2 -5])
format short
```

FUNZIONE RUNGE

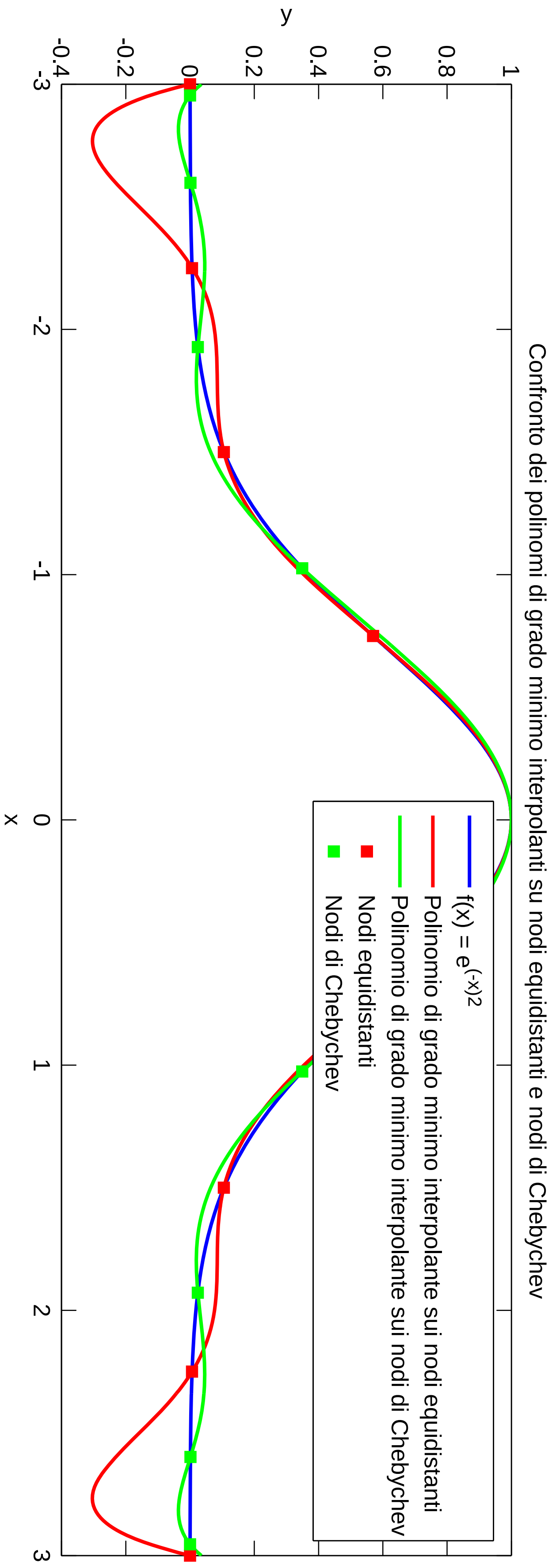
```
function [ y ] = Runge( x )
% Runge
y=1./(1+25.*x.^2);
end
```

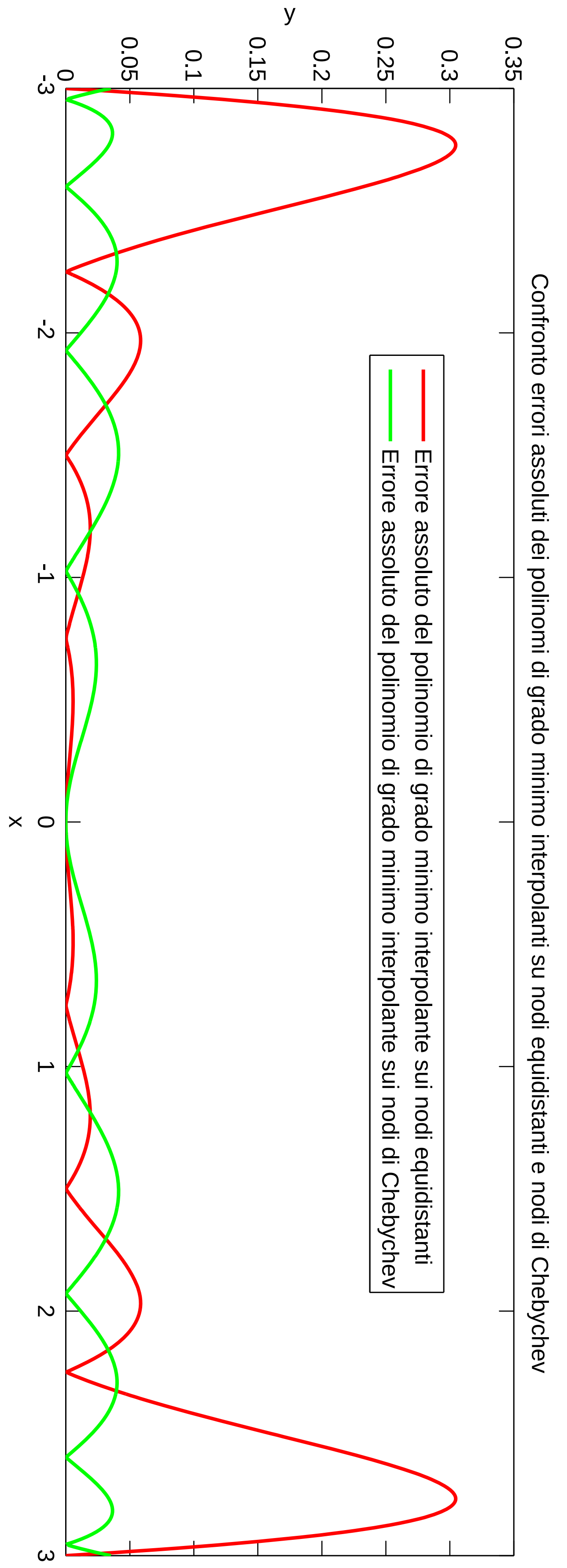
COMMAND WINDOW

```
som =

     2    -3     3     2   -12


diff =

     2    -3    -3     6    -2


prod =

     0     0     6   -13    -4    27   -29    -6    35


q =

       2/3           -5/9           20/27


r =

       0              0              0             73/27         -89/27
```

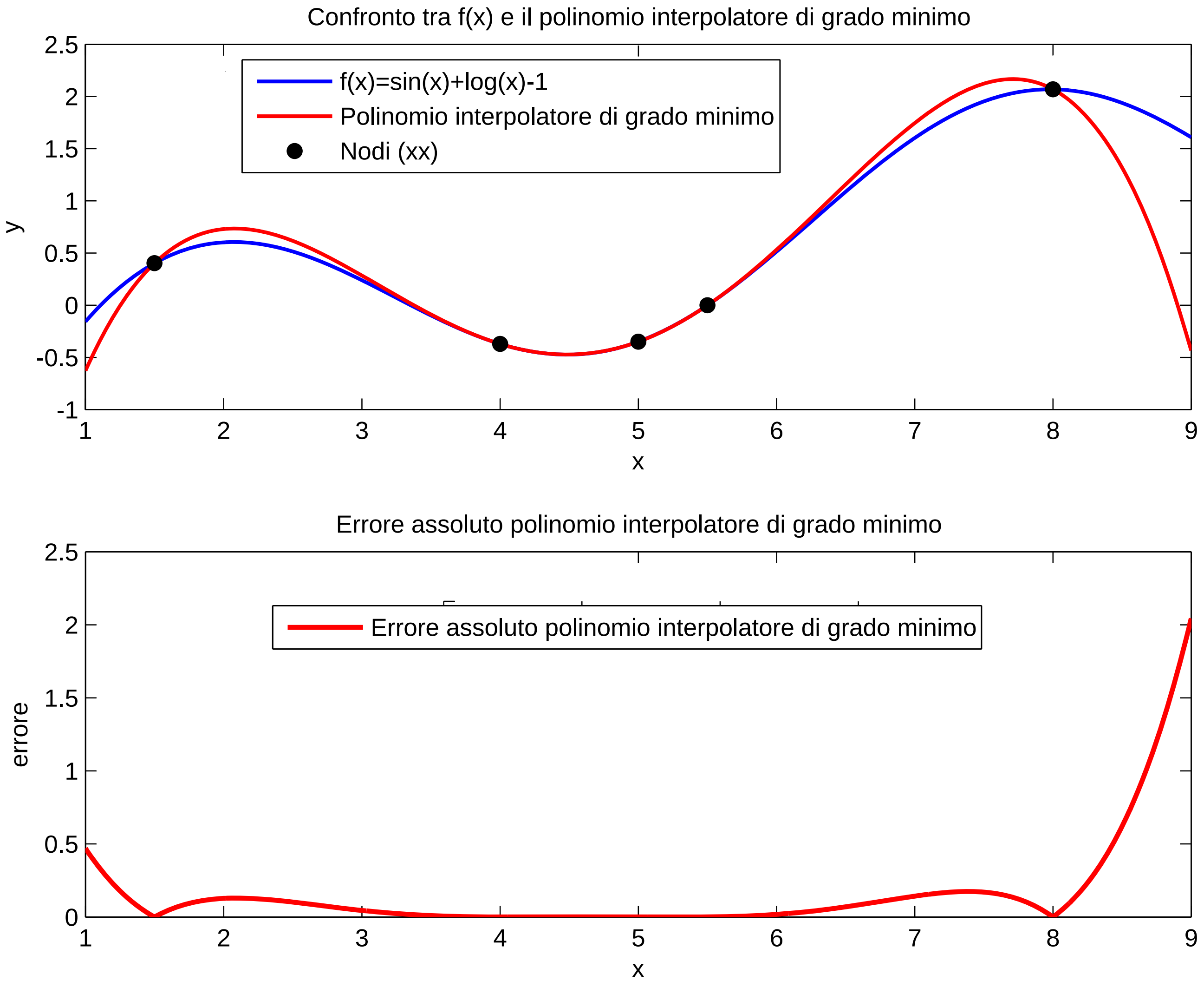
```
% Esercizio 6 - Esercitazione 6
clear all
close all
clc
% vettore di punti per valutare la funzione e i polinomi interpolatori
xx=linspace(-3,3,1000);
% vettore di 9 nodi equidistanti nell'intervallo [-3,3]
x=linspace(-3,3,9);
% vettore contente le immagini della funzione f(x) sui 9 punti nodali
y=exp(-x.^2);
% valori esatti della funzione nei punti xx
y_esa=exp(-xx.^2);

% polinomio di grado minimo interpolante sui nodi equidistanti
pol=polyfit(x,y,8);
% valori del polinomio di grado minimo interpolante nei punti xx
y_pol=polyval(pol,xx);

% ripetiamo le stesse operazioni su una decomposizione di 9 nodi di
% Chebychev.
% vettore dei 9 nodi di Chebychev
for i=1:9
   x_c(1,i)=1/2*(-3+3)+1/2*(3+3).*(cos(((2.*i-1)./18).*pi));
end
% vettore contente le immagini della funzione f(x) sui nodi di Chebychev
y_c=exp(-x_c.^2);
% polinomio di grado minimo interpolante sui nodi di Chebychev
pol_c=polyfit(x_c,y_c,8);
% valori del polinomio di grado minimo interpolante pol_c nei punti xx
y_pol_c=polyval(pol_c,xx);



% errore assoluto del polinomio di grado minimo interpolante sui nodi
% equidistanti
err_ass_pol=abs(y_esa-y_pol);
% errore assoluto del polinomio di grado minimo interpolante sui nodi di
% Chebychev
err_ass_pol_c=abs(y_esa-y_pol_c);

%Grafico
figure
subplot(2,1,1)
hold on
box on
plot(xx,y_esa,xx,y_pol,xx,y_pol_c,x,y,x_c,y_c)
subplot(2,1,2)
plot(xx,err_ass_pol,xx,err_ass_pol_c)
% Dal confronto degli errori assoluti dei polinomi di grado minimo
% interpolanti su nodi equidistanti e nodi di Chebychev possiamo affermare
% che la migliore tra le due approssimazioni è quella sui nodi di
% Chebychev.
```

Grafico Esercizio 6 - Esercitazione 6





```
% Esercizio 7 - Esercitazione 6
clear all
close all
clc

% vettore dei punti per valutare la funzione e il polinomio
x=linspace(1,9,1000);
% vettore dei nodi
xx=[1.5 4 5 5.5 8];
% vettore delle immagini della funzione sui nodi
y=sin(xx)+log(xx)-1;
% valori della funzione esatti
y_esa=sin(x)+log(x)-1;
% polinomio interpolatorio di grado minimo sui nodi xx
pol=polyfit(xx,y,4);
% valori del polinomio interpolatorio sui punti x
y_pol=polyval(pol,x);
% errore assoluto del polinomio interpolatorio di grado minimo
err_ass_pol=abs(y_esa-y_pol);
% Grafico
figure
hold on
box on
subplot(2,1,1)
plot(x,y_esa,x,y_pol,xx,y)
subplot(2,1,2)
plot(x,err_ass_pol)
```



```
% Esercizio 8 - Esercitazione 6
clear all
close all
clc
% comandi per grafico
figure
hold on
box on
% vettore dei punti per valutare il polinomio e la funzione
xx=linspace(-1,1,1000);
% valori esatti della funzione
y_esa=Runge(xx);
% calcoliamo il polinomio interpolatore di grado minimo per i gradi
% 9,10,11,12 con nodi equidistanti
for i=9:12
    % vettore dei nodi equidistanti
    x=linspace(-1,1,i+1);
    % vettore immagini della funzione Runge
    y=Runge(x);
    % polinomio interpolatorio di grado minimo nei nodi x
    pol=polyfit(x,y,i);
    % valori del polinomio interpolatorio di grado minimo sui punti xx
    y_pol=polyval(pol,xx);
    % Grafico
    subplot(2,2,i-8)
    plot(xx,y_esa,xx,y_pol,x,y)
end
```

Grafico Esercizio 8 - Esercitazione 6