V esercitazione di
Laboratorio Computazionale Numerico

1. a) Rappresentare graficamente su un intervallo $[a,b]$ la funzione

$$f(x) = 2\sin(8x) - \ln(x^2+1)$$

usando una griglia di punti equispaziati dell'intervallo $[a,b]$.

b) Rappresentare la funzione $f(x)$ definita da:

$$f(x) = \begin{cases} x^2 & x \leq 3 \\ 9 & x > 3 \end{cases}$$

sull'intervallo $[2,4]$.

2. Disegnare sull'intervallo $[-1,1]$ il grafico della funzione $|\omega_n(x)| = |\prod_{i=0}^{n}(x-x_i)|$ per alcuni valori di n considerando nodi equispaziati.
Ripetere l'esercizio considerando i nodi Chebyshev:

$$x_i = cos[(2i+1)\pi/(2n+2)],\ i = 0, \cdots, n\,.$$

Confrontare sullo stesso sistema di riferimento i due grafici.

3. Realizzare uno script Matlab che disegni i grafci delle seguenti funzioni:

$$f(x) = \begin{cases} \left(m - \dfrac{x^2}{m}\right)^m & x \in [-m, 0] \\ \left(\dfrac{x^2}{m} + m\right)^m & x \in [0, m] \end{cases}$$

per $m = 1, \ldots, 6$ sovrapposti nella stessa finestra e successivamente tutti in una stessa finestra utilizzando sei sottofinestre della stessa finestra grafica.

4. Realizzare un programma Matlab per la risoluzione di sistemi con matrice tridiagonale.

5. Data la matrice di ordine n:

$$A = \begin{bmatrix} 1 & 2 & & & 0 \\ & 1 & 2 & & \\ & & \ddots & \ddots & \\ & & & & 2 \\ 0 & & & & 1 \end{bmatrix}$$

Costruire un programma Matlab per calcolare A^{-1}.

6. Supponiamo di aver effettuato i seguenti rilevamenti:

t	0	1	2	3	4	5
y	0.5	0.8	0.7	0.3	0.1	0.4

Determinare i parametri a_0, a_1, a_2 del modello

$$y(t) = a_0 + a_1 \sin\frac{2\pi t}{6} + a_2 \cos\frac{2\pi t}{6}$$

7. Data una matrice simmetrica definita positiva di forma tridiagonale:

$$A = \begin{bmatrix} a_1 & c_1 & & & 0 \\ c_1 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & a_{n-1} & c_{n-1} \\ 0 & & & c_{n-1} & a_n \end{bmatrix}$$

Costruire la matrice:

$$B = \begin{bmatrix} p_1 & & & & 0 \\ q_1 & p_2 & & & \\ & \ddots & \ddots & \ddots & \\ & & q_{n-2} & p_{n-1} & \\ 0 & & & q_{n-1} & p_n \end{bmatrix} \qquad \text{dove} \qquad \begin{array}{l} p_1 = \sqrt{a_1} \\ i = 2, \cdots, n \\ q_{i-1} = c_{i-1}/p_{i-1} \\ p_i = \sqrt{a_i - q_{i-1}^2} \end{array}$$

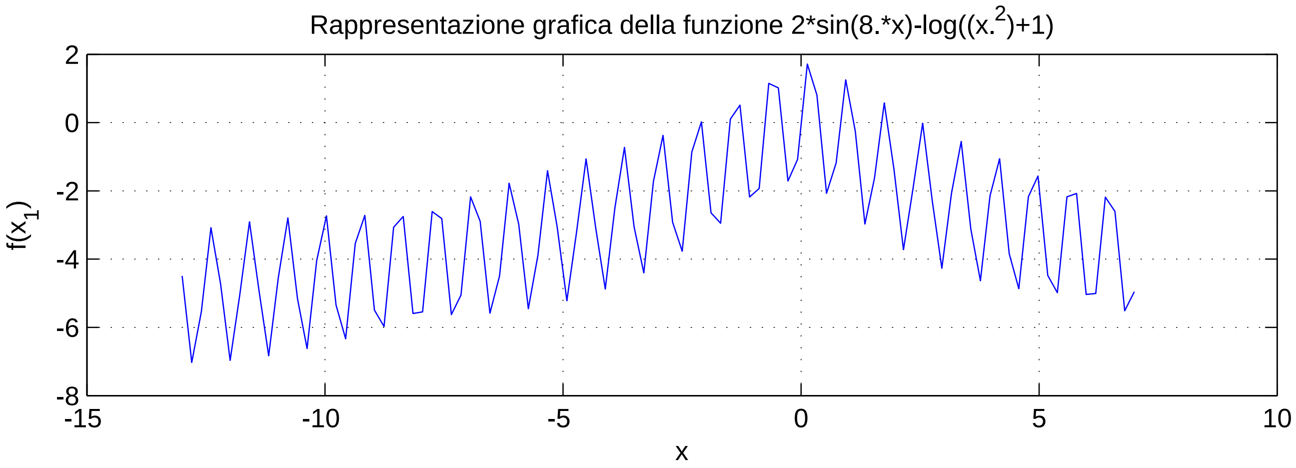
Verificare poi che $A = BB^t$.

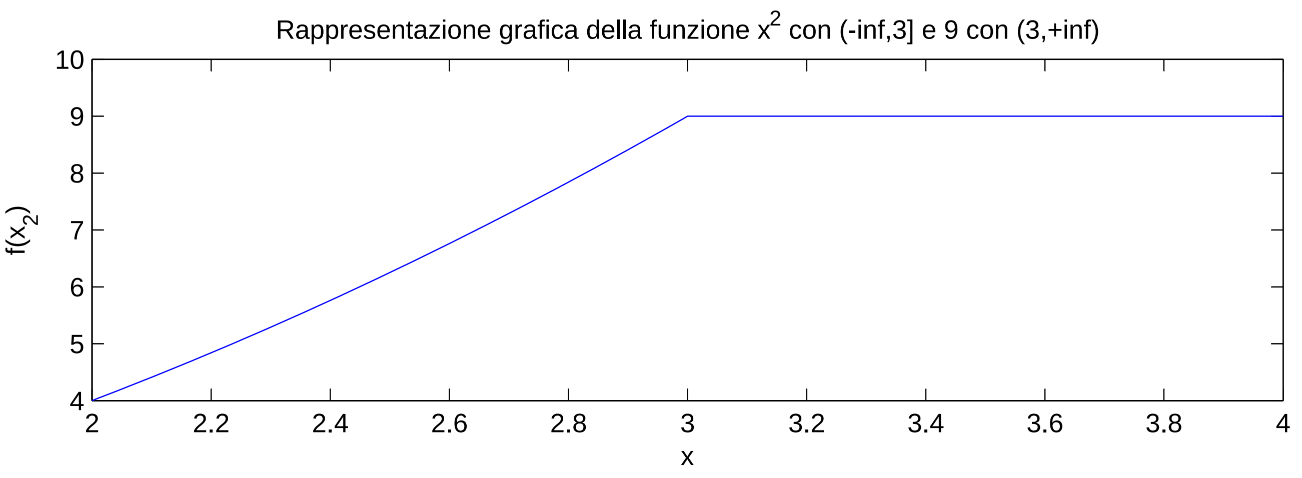
8. Per il calcolo di π si può utilizzare una troncata della serie:

$$\pi = \sum_{n=0}^{\infty} 16^{-n}\left(\frac{4}{8n+1} - \frac{2}{8n+4} - \frac{1}{8n+5} - \frac{1}{8n+6}\right)$$

Utilizzando diversi valori di n valutare una approssimazione di π e confrontarla con il valore memorizzato nella variabile `pi`.

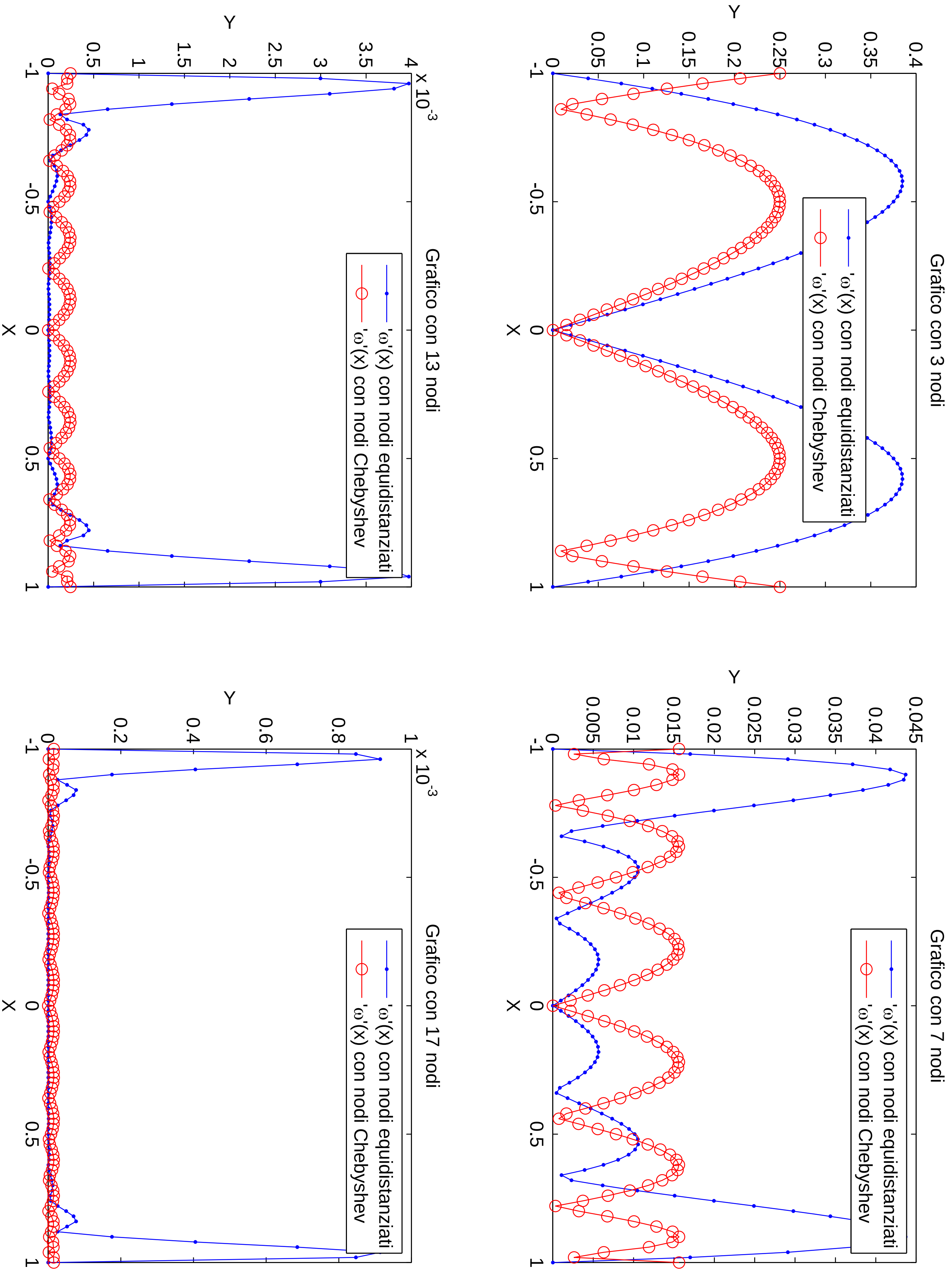
```
% Esercizio 1 - Esercitazione 5
clear all
close all
clc
%a)
a=-13;
b=7;
x=linspace(a,b,100);
y=2*sin(8.*x)-log((x.^2)+1);
subplot(2,1,1);
plot(x,y)
grid on
%b)
x_2_1=linspace(2,3,100);
x_2_2=linspace(3,4,100);
% togliamo nel secondo intervallo x=3, per ottenere x>3
x_2_2(1)=[];
y_2_1=x_2_1.^2;
y_2_2=9.*ones(1,length(x_2_2));
x_2= [x_2_1 x_2_2];
y_2= [y_2_1 y_2_2];
subplot(2,1,2);
plot(x_2,y_2)
% cambiamo il range dell'asse y
ylim([4,10])
```





```
% Esercizio 2 - Esercitazione 5
clear all
close all
clc
% contatore ausiliario per la costruzione del grafico
 ig=0;
% numero punti in cui il polinomio interpola la funzione
for n_nodi = [3 7 13 17]
    % punti in cui il polinomio annulla la funzione
    nodi=linspace(-1,1,n_nodi);
    % punti in cui valuti le funzioni (x di f(x))
    x=linspace(-1,1,101);
    % Calcoliamo il polinomio considerando i nodi equispaziati
    % inizializziamo omega_ne
    omega_ne=ones(1,101);
    for i = 1:n_nodi
        omega_ne=abs(omega_ne.*(x-nodi(i)));
    end
    hold on
    ig=ig+1;
    subplot(2,2,ig)
    plot(x,omega_ne)
    % Calcoliamo il polinomio considerando i nodi di Chebyshev
    % punti di Chebyshev in cui il polinomio annulla la funzione, utilizziamo
    % vett per calcolare gli n_nodi
    vett=0:n_nodi-1;
    nodi_C=cos(((2.*vett+1).*pi)./(2*(n_nodi-1)+2));
    % inizializziamo omega_C
    omega_C=ones(1,101);
    for i = 1:n_nodi
        omega_C=abs(omega_C.*(x-nodi_C(i)));
    end
    hold on
    plot(x,omega_C)
end
```

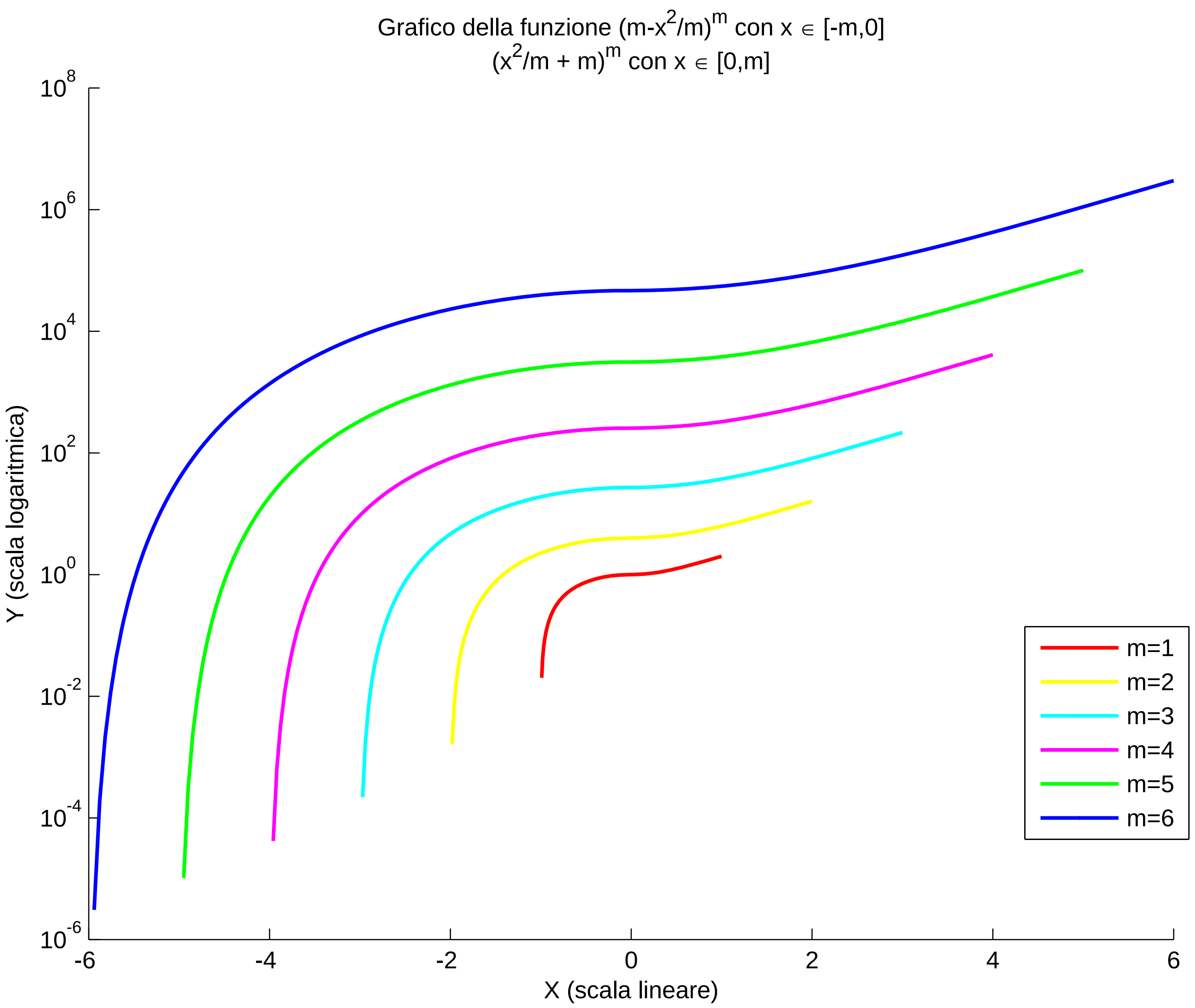
Grafico Esercizio 2 - Esercitazione 5



```
% Esercizio 3a - Esercitazione 5
clear all
close all
clc
% visualizzazione in un unica finestra, con sovrapposizione grafici
% Ciclo per valori da 1 a 6
for m=1:6;
   [x y]=Calcola_Grafico_es3(m);
   hold on
   plot(x,y)
end
```

FUNZIONE Calcola_Grafico_es3

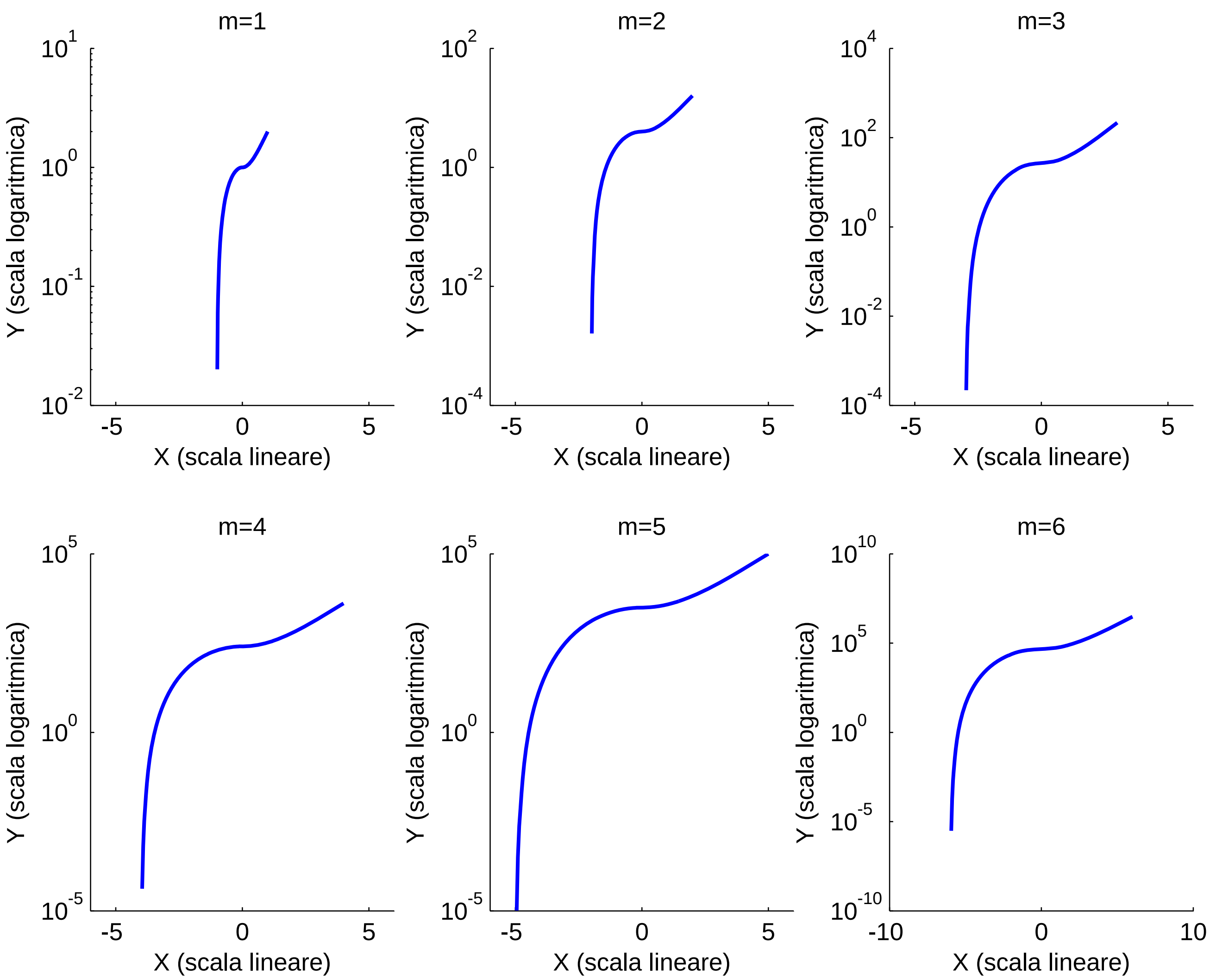
```
function [ x,y ] = Calcola_Grafico_es3( m )
% Calcola_Grafico_es3 restituisce le cordinate per costruire il grafico
% della funzione prendendo in input l'intervallo m
   x_1=linspace(-m,0,50);
   x_2=linspace(0,m,50);
   x=[x_1 x_2];
   y_1=(m-(x_1.^2)./m).^m;
   y_2=((x_2.^2)./m + m).^m;
   y=[y_1 y_2];
end
```



```
% Esercizio 3b - Esercitazione 5
clear all
close all
clc
% visualizzazione in m sottofinestre, quindi per ogni grafici
% Ciclo per valori da 1 a 6
% Vettore x
for m=1:6;
    [x y]=Calcola_Grafico_es3(m);
    subplot(2,3,m)
    hold on
    plot(x,y)
end
```

FUNZIONE Calcola_Grafico_es3

```
function [ x,y ] = Calcola_Grafico_es3( m )
% Calcola_Grafico_es3 restituisce le cordinate per costruire il grafico
% della funzione prendendo in input l'intervallo m
    x_1=linspace(-m,0,50);
    x_2=linspace(0,m,50);
    x=[x_1 x_2];
    y_1=(m-(x_1.^2)./m).^m;
    y_2=((x_2.^2)./m + m).^m;
    y=[y_1 y_2];
end
```



```
% Esercizio 4 - Esercitazione 5
diary esercizio_4.txt
clear all
close all
clc
format rat
% matrice tridiagonale
A=[2 1/2 0;1/2 2 1/2;0 1/2 2]
% vettore termini noti
b=[-6;6;-6]
x=RisolSisMatTrid(A,b)
 format short
```

FUNZIONE RisolSisMatTrid

```
function [ x ] = RisolSisMatTrid( M,b )
% Risoluzione di sistemi con matrice tridiagonale utilizzando il metodo di
% fattorizzazione LU
% input: M: matrice tridiagonale associata al sistema
%        b: vettore termnini noti
% x: vettore soluzione del sistema
    n=length(M);
    % estraiamo la diagonale principale, inferiore e superiore dalla
    % matrice M
    ad=diag(M,0);
    bd=[0 diag(M,-1)'];
    cd=[diag(M,1)' 0];
    % per definizione ricaviamo alfa, beta e c(coicide con cd)
    alfa(1)=ad(1);
    for i=2:n
        beta(i)=bd(i)/alfa(i-1);
        alfa(i)=ad(i)-beta(i)*cd(i-1);
    end
    % togliamo l'elemento in piu in testa al vettore beta
    beta(1)=[];
    cd(end)=[];
    % L: matrice bidiagonale inferiore (specifica per tridiagonale)
    L=zeros(n)+diag(ones(1,n),0)+diag(beta,-1);
    % U: matrice bidiagonale superiore (specifica per tridiagonale)
    U=zeros(n)+diag(alfa,0)+diag(cd,1);

    %risolviamo il sistema L*z=b
    z=L\b;
    %risolviamo il sistema U*x=z
    x=U\z;

end
```

COMMAND WINDOW

```
   A =
        2          1/2        0
        1/2        2          1/2
        0          1/2        2
   b =
      -6
       6
      -6

   x =
      -30/7
       36/7
      -30/7
```

```
% Esercizio 5 - Esercitazione 5
clear all
close all
clc
% Programma pe ril calcolo di una matrice bidiagonale superiore
% n: ordine della matrice A
n=5;
A=zeros(n)+diag(ones(n,1),0)+2*diag(ones(n-1,1),1)
% Calcoliamo l'inversa di A tramite la definizione A*A^-1=id
A_id=eye(n);
A_inv=A\A_id
% In alternativa calcoliamo l'inversa di A iterando sulle diagonali
A_inv2=eye(n);
for i=i:n
    A_inv2=A_inv2+diag((-2)^i*ones(n-i,1),i);
end
A_inv2
```

COMMAND WINDOW

```
A =

     1     2     0     0     0
     0     1     2     0     0
     0     0     1     2     0
     0     0     0     1     2
     0     0     0     0     1

A_inv =

     1    -2     4    -8    16
     0     1    -2     4    -8
     0     0     1    -2     4
     0     0     0     1    -2
     0     0     0     0     1

A_inv2 =

     2    -2     4    -8    16
     0     2    -2     4    -8
     0     0     2    -2     4
     0     0     0     2    -2
     0     0     0     0     2
```

```
% Esercizio 6 - Esercitazione 5
diary esercizio_6.txt
clear all
close all
clc
% nodi
t=0:5;
% valori nei nodi
y=[0.5 0.8 0.7 0.3 0.1 0.4];
% calcoliamo i parametri nei vari nodi
for i=1:6
   % consideriamo y come vettore dei termini noti e M come la matrice
   % associata al sistema Mx=y
   A(i,:)= [1 sin((2*pi*t(i))/6) cos((2*pi*t(i))/6)];
end
% Utilizziamo la fattorizzazione QR con il metodo Householder per calcolare
% i coefficienti che descrivono una curva ai minimi quadrati.

% Q: Matrice Ortogonale, impostiamo il valore iniziale di Q con la matrice
% identita per poi moltiplicarla nel ciclo for in modo da ottenere
% Q = Q_1*...Q_t   , dove t è il numero totale di iterazioni
Q = eye(6);
% R: Matrice diagonale superiore, impostiamo inizialmente R = A per poi
% moltiplicarla nel ciclo for in modo da ottenere R= Q_1*...*A    .
R = A;
for i = 1:3
   % la dimensione del vettori x e e varia a seconda del valore di i.
   %x: Vettore colonna arbitrario
   x = R(:,i);
   %alfa: valore assuluto della Lunghezza di x
   alfa = abs(norm(x));
   %e: Vettore colonna (1,0,...,0)'
   e = [zeros(i-1,1); 1; zeros(6-i,1)];
   %u: Vettore colonna tale che u = x - alfa*e
   u = x - alfa*e;
   v = u/norm(u);
   I=eye(6);
   %Qi: Matrice di Householder Q=I-(2*v*v')
   Qi = I - (2*v*(v'));
   R = Qi*R;
   Q = Q*Qi;
end
% Q e R sono costruite in modo che Q*R = A.
% la decomposizione A=QR è una decomposizione QR di A.

% Risoluzione del sistema lineare per trovare i coefficienti
% Q essendo una matrice ortogonale si potrebbe risolvere il sitema anche
% con z= Q'*y'  , dato che Q'=Q^-1  .
% y' è il vettore colonna dei termini noti
z = Q\y';
x = R\z
diary off
```

COMMAND WINDOW

```
x =

    0.4667
    0.2887
    0.1333
```

```
% Esercizio 7 - Esercitazione 5
diary esercizio_7.txt
clear all
close all
clc
% dimensione matrice
n=3;
% matrice tridiagonale casuale
A=[1 4 0; 4 2 5; 0 5 3];
% estrazione diagonale principale e diagonale superiore(coincidente con
% diagonale inferiore)
a=diag(A,0);
c=diag(A,1);
% calcolo parametri per costruire di b
p(1)=sqrt(a(1));
for i=2:n
    q(i-1)=c(i-1)./p(i-1);
    p(i)=sqrt(a(i)-(q(i-1)).^2);
end
% costruzione matrice B
B=zeros(n)+diag(p,0)+diag(q,-1)
% verifichiamo che A=B*B'
A_1=B*B';
% errore assoluto dovuto all'approssimazione
err_ass=abs(A-A_1)
diary off
```

COMMAND WINDOW

```
B =

   1.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   4.0000 + 0.0000i   0.0000 + 3.7417i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 - 1.3363i   2.1876 + 0.0000i


err_ass =

         0         0         0
         0   28.0000   10.0000
         0   10.0000    3.5714
```

```
% Esercizio 8 - Esercitazione 5
diary esercizio_8.txt
clear all
close all
clc
format long
% numero ripetizioni
n=10;
% calcolo sommatoria con n=0
pi_appr=1*(4-2/4-1/5-1/6);
% errore relativo con n=0
err_rel=abs(pi-pi_appr)./pi;
% inserimento nella prima riga della tabella die valori con n=0
tab_ris(1,:)=[0 pi_appr err_rel];
for i=1:n
    pi_appr=pi_appr + 16^(-i)*((4/(8*i+1))-(2/(8*i+4))-(1/(8*i+5))-(1/(8*i+6)));
    % errore relativo
    err_rel=abs(pi-pi_appr)./pi;
    % costruzione tabella
    tab_ris(i,:)=[i pi_appr err_rel];
end
tab_ris
format short
diary off
```

COMMAND WINDOW

```
tab_ris =

   1.000000000000000   3.141422466422466   0.000054172257862
   2.000000000000000   3.141587390346582   0.000001675342348
   3.000000000000000   3.141592457567436   0.000000062395854
   4.000000000000000   3.141592645460337   0.000000002587686
   5.000000000000000   3.141592653228088   0.000000000115134
   6.000000000000000   3.141592653572881   0.000000000005383
   7.000000000000000   3.141592653588973   0.000000000000261
   8.000000000000000   3.141592653589752   0.000000000000013
   9.000000000000000   3.141592653589791   0.000000000000001
  10.000000000000000   3.141592653589793                   0
```