**IV esercitazione di**
**Laboratorio Computazionale Numerico**

1. Determinare i coefficienti $a_i$ dei polinomi di grado 10, 15, 20 interpolanti la funzione $f(x) = e^x + 1$ nei nodi equispaziati dell'intervallo $[-1, 1]$ (sugg. Usare la matrice di Vandermonde `V=vander(x)` di elementi $V(i, j) = x(i)^{(n-j)}$ dove `n=length(x)`... e poi risolvere il sistema lineare). Successivamente considerare i nuovi dati perturbati $\hat{f}(x_i) = f(x_i) + \varepsilon_i$ con $\varepsilon_i = (-1)^i 10^{-5}$ e calcolare i coefficienti $\hat{a}_i$ del polinomio perturbato $\hat{p}(x)$ interpolante i dati $(x_i, \hat{f}(x_i))$. Confrontare i grafici dei polinomi $p(x)$ e $\hat{p}(x)$.
   Calcolare:
   $$\max \mid a_i - \hat{a}_i \mid \qquad \text{e} \qquad \max \mid p(t) - \hat{p}(t) \mid$$
   dove $t$ è un vettore formato da 101 punti equispaziati in $[-1, 1]$.

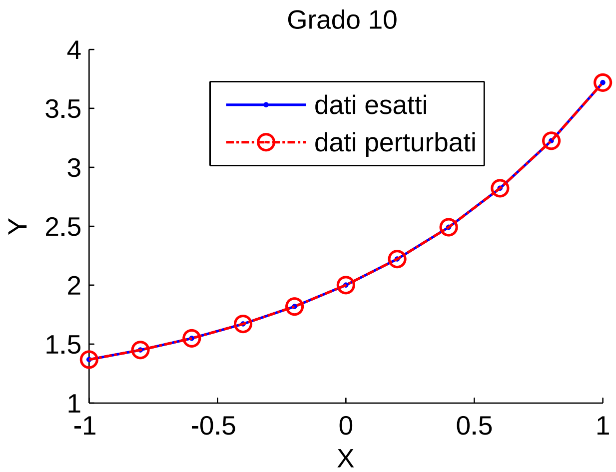
2. Siano A=hilb(1000) e B=rand(1000).
   a) Costruire il vettore b in modo che il sistema Ax = b sia risolto da x=ones(1000,1), e il vettore c in modo che il sistema By = c sia risolto da y=ones(1000,1);
   b) Calcolare i vettori x e y come soluzione dei sistemi lineari Ax = b, By = c utilizzando il comando \ (`mldivide`) di Matlab;
   c) Calcolare gli errori relativi rapportandoli al numero di condizionamento delle corrispondenti matrici. Cosa si osserva?

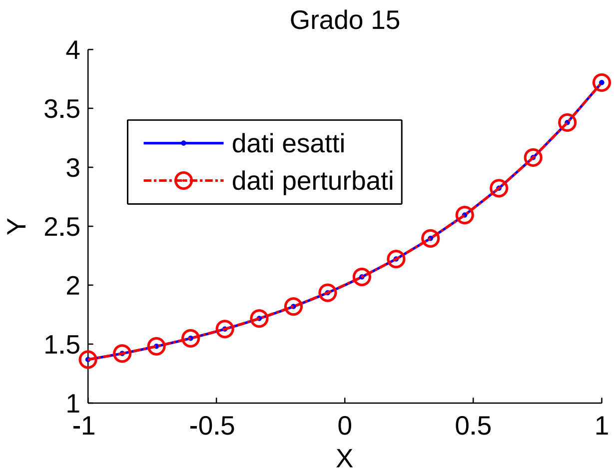
```
% Esercizio 1 - Esercitazione 4
clear all
close all
clc
% ciclo for per i gradi del polinomio uguali a 10, 15, 20
for n=10:5:20

    % intervallo [-1,1] , con n+1 punti
    x=linspace(-1,1,n+1);
    % la funzione interpolata è f(x)=exp(x)+1
    y=exp(x)+1;
    % Usiamo la matrice di Vandermonde
    V=vander(x);
    % Vx=y quindi per risolvere il sistema lineare e trovare i coefficienti
    coeff=V\y';

    % consideriamo i dati perturbati, aggiungendo la perturbazione ad ogni
    % punto di y
    for i = 0:n
    epsi=((-1)^(i))*(10^-5);
    y_p(i+1)=y(i+1)+epsi;
    end


    % calcolo dei cofficienti del polinomio perturbato, che interpola i dati
    %(x, y_p)
    coeff_p=V\y_p';
    % Confrontiamo i grafici del polinomio e del polinomio perturbato
    % impostiamo le pagina del plot con 2 righe, 2 colonne.
    subplot(2,2,(n-5)/5);
    hold on;
    plot(x,y);
    plot(x,y_p,'-o');
    % Calcoliamo il massimo errore assoluto del coefficiente e del
    % polinomio
    t=linspace(-1,1,101);
    err_coeff=max(abs(coeff-coeff_p));
    err_polin=max(abs(polyval(coeff,t)-polyval(coeff_p,t)));
    vet_err_coeff(:,(n-5)/5)=err_coeff;
    vet_err_polin(:,(n-5)/5)=err_polin;
end
err_coeff=max(vet_err_coeff)
err_polin=max(vet_err_polin)
```

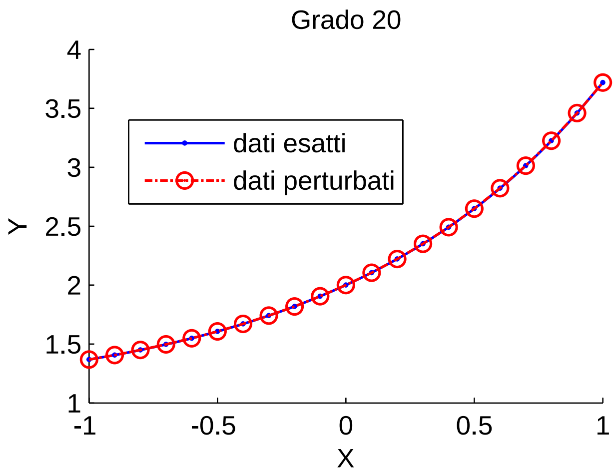
COMMAND WINDOW

```
  err_coeff =

     2.5685e+03


  err_polin =

      0.1069
```

Grafico Esercizio 1 - Esercitazione 4



```
% Esercizio 2 - Esercitazione 4
clear all
close all
clc

A=hilb(1000);
B=rand(1000);
%a)
x=ones(1000,1);
y=x;
b=A*x;
c=B*y;
%b)
x=A\x;
y=B\y;
%c)
err_rel_x=abs(x-ones(1000,1))./ones(1000,1);
err_rel_y=abs(y-ones(1000,1))./ones(1000,1);
% Valutiamo il condizionamento della matrice per capire quanto si propaghi
% l'errore
condiz_A=cond(A)
condiz_B=cond(B)
hold on;
plot(1:1000,err_rel_x)
plot(1:1000,err_rel_y)
% la matrice di Hilbert ha un condizionamento maggiore rispetto a una
% matrice con valori casuali, infatti grafico si può notare che l'errore
% relativo sul vettore delle incognite x(inerente alla matrice di Hilbert)
% è maggiore rispetto all'errore relativo di y.
```

COMMAND WINDOW

```
condiz_A =

  6.2323e+21

condiz_B =

  8.5596e+04
```