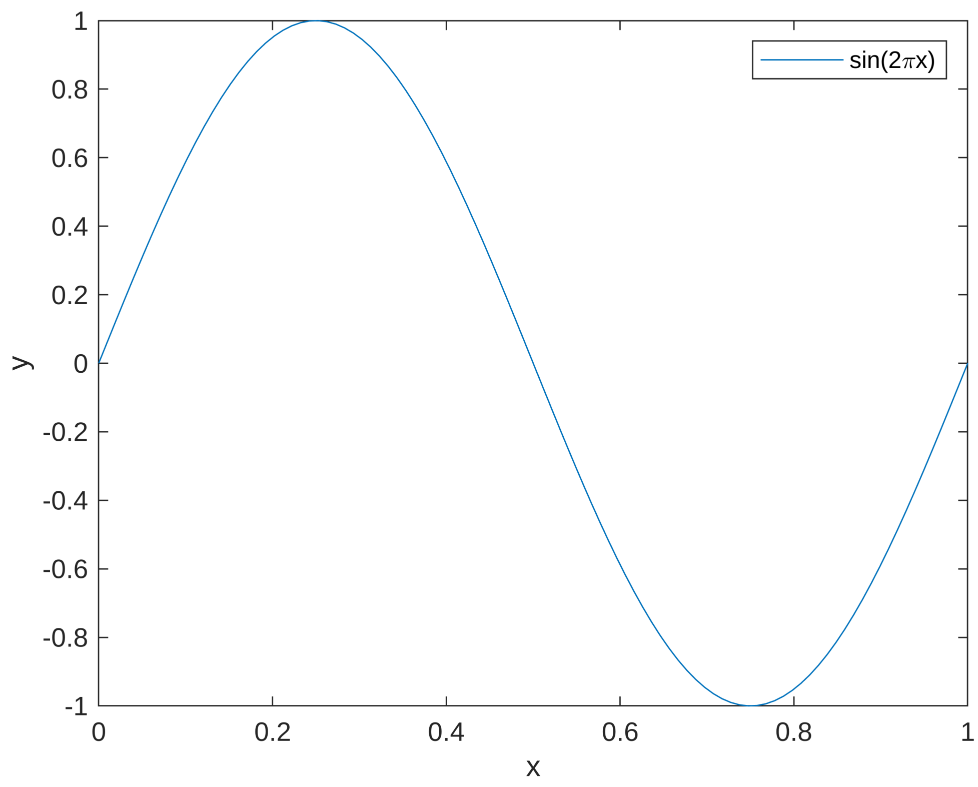
Polynomial interpolation

```
% cleaning
clc
clear

% funzione seno
sen = @(x) sin(2*pi*x);

% genero vettori
x = linspace(0,1,100);
y = sen(x);

% plotto funzione seno
figure;
plot(x,y)
xlabel("x")
ylabel("y")
legend("sin(2\pix)")
```



```
% genero set di learning
n_lrn = 10;
x_lrn = linspace(0,1,n_lrn);
```

```
x_lrn = 1×10
```

```
         0    0.1111    0.2222    0.3333    0.4444    0.5556    0.6667    0.7778 ···
```

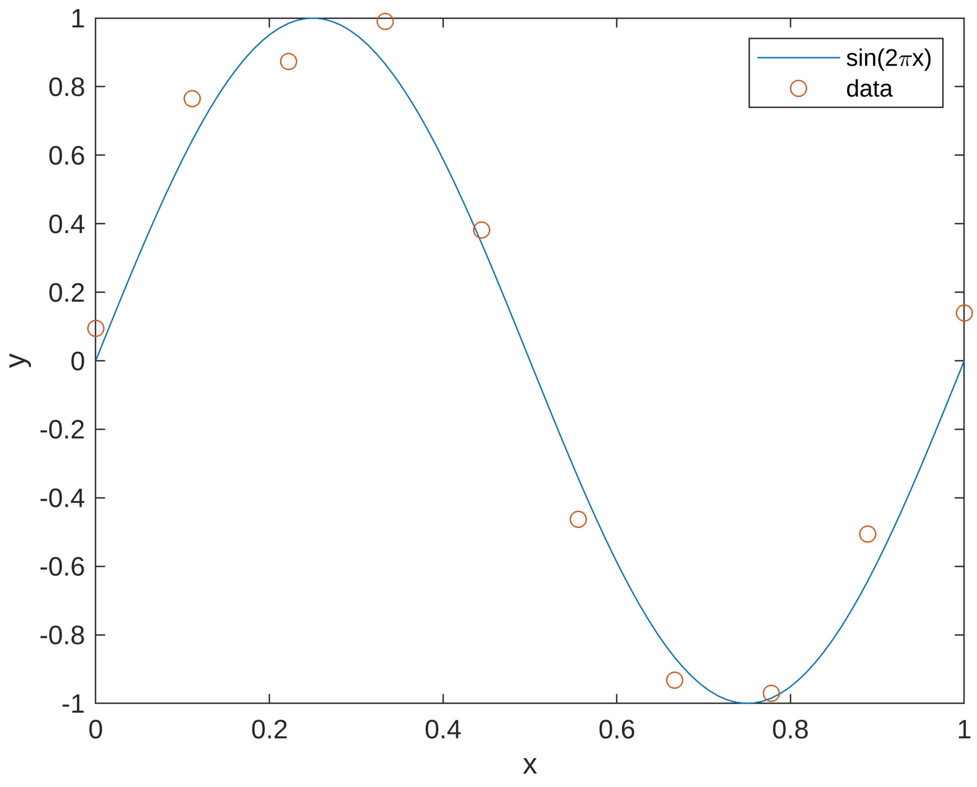
```
eps = 0.15;
y_lrn = sin(2*pi*x_lrn);
```

```
y_lrn = 1×10
         0    0.6428    0.9848    0.8660    0.3420   -0.3420   -0.8660   -0.9848 ···
```

```
y_lrn = y_lrn + (-eps + (2.*eps).*rand(n_lrn,1))';
```

```
y_lrn = 1×10
    0.0944    0.7645    0.8729    0.9900    0.3817   -0.4628   -0.9325   -0.9707 ···
```

```

figure
plot(x,y)
hold on
plot(x_lrn,y_lrn,"o")
legend("sin(2\pix)","data")
xlabel("x")
ylabel("y")
hold off
```



```
% genero matrice di Vandermonde
V = fliplr(vander(x_lrn))
```

```
V = 10×10
```

```
    1.0000         0         0         0         0         0         0         0···
    1.0000    0.1111    0.0123    0.0014    0.0002    0.0000    0.0000    0.0000
    1.0000    0.2222    0.0494    0.0110    0.0024    0.0005    0.0001    0.0000
    1.0000    0.3333    0.1111    0.0370    0.0123    0.0041    0.0014    0.0005
    1.0000    0.4444    0.1975    0.0878    0.0390    0.0173    0.0077    0.0034
    1.0000    0.5556    0.3086    0.1715    0.0953    0.0529    0.0294    0.0163
    1.0000    0.6667    0.4444    0.2963    0.1975    0.1317    0.0878    0.0585
    1.0000    0.7778    0.6049    0.4705    0.3660    0.2846    0.2214    0.1722
    1.0000    0.8889    0.7901    0.7023    0.6243    0.5549    0.4933    0.4385
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
```

Risolvo il sistema $y = \alpha V$ dove α sono i coefficienti del polinomio cercato: $y = \alpha_1 + \alpha_2 x_1 + \alpha_3 x_2^2 + ...$

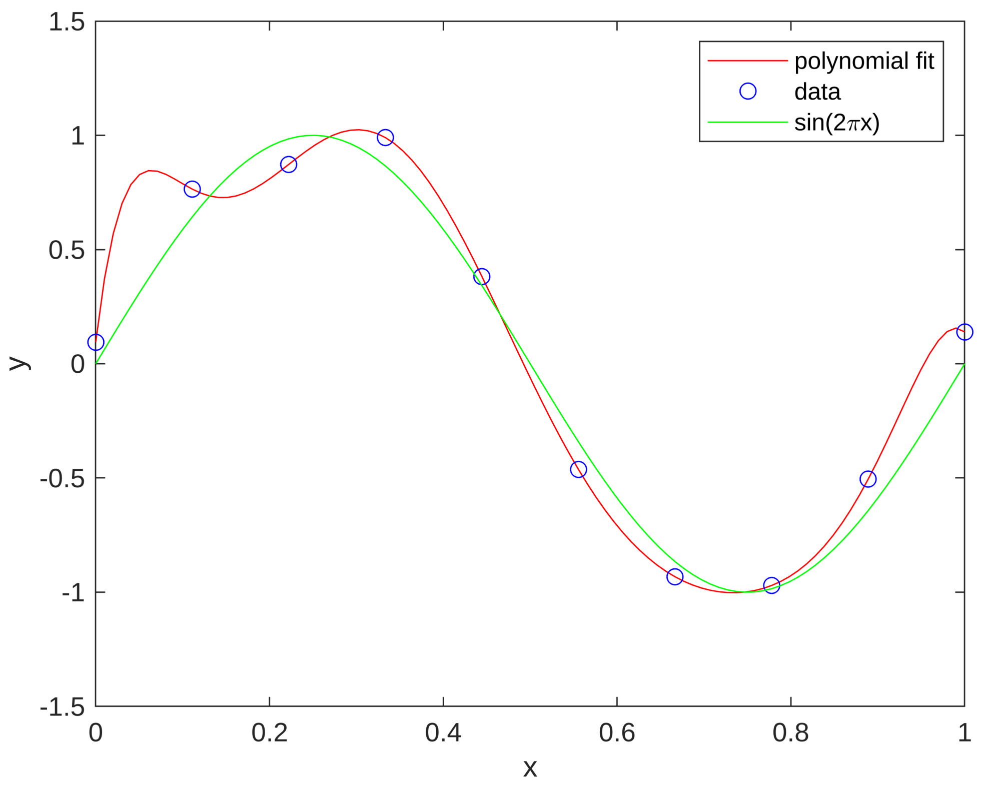
Alla luce della forma matriciale, è possibile determinare i coefficienti α eseguendo il prodotto righe per colonna tra l'inversa della matrice di Vandermonde e il vettore colonna y

```
% determino i coefficienti
a = pinv(V)*(y_lrn')
```

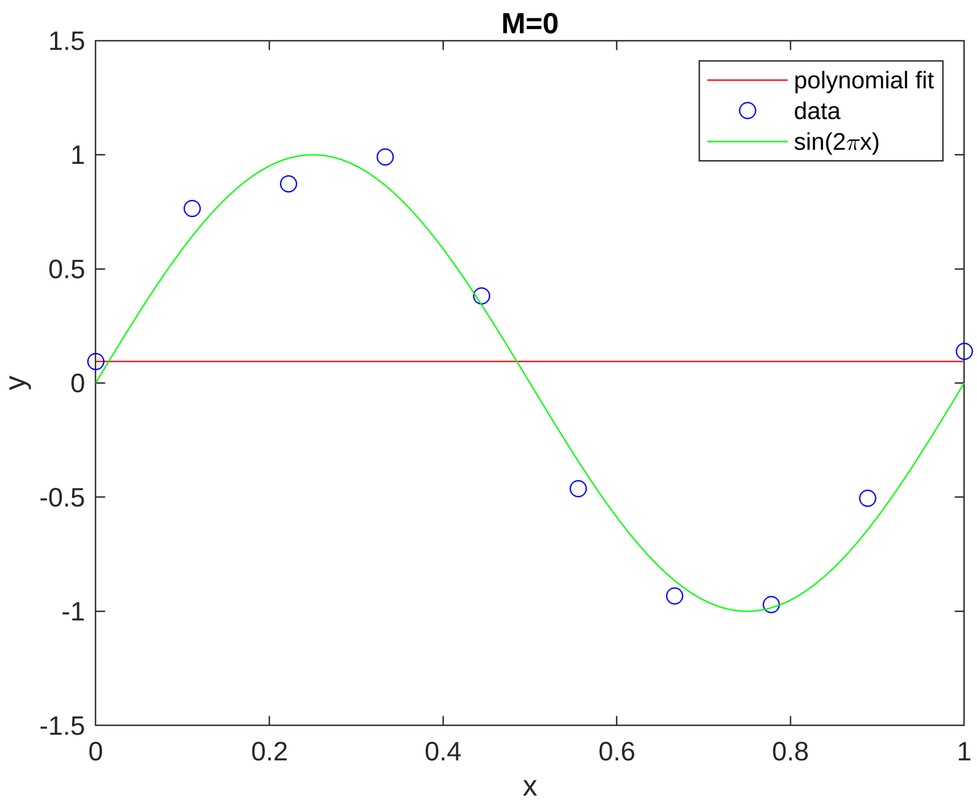
```
a = 10×1
10^4  ×
    0.0000
    0.0032
   -0.0496
    0.3438
   -1.2227
    2.4009
   -2.6659
    1.5967
   -0.4267
    0.0203
```

```

% ottengo il polinomio funzione degli scalari x e m (grado)
poly = @(x,m) (x.^(0:m))*(a(1:m+1));

% over-fitting
z = zeros(1,100);
for i=1:100
    z(i) = poly(x(i),n_lrn-1);
end
```

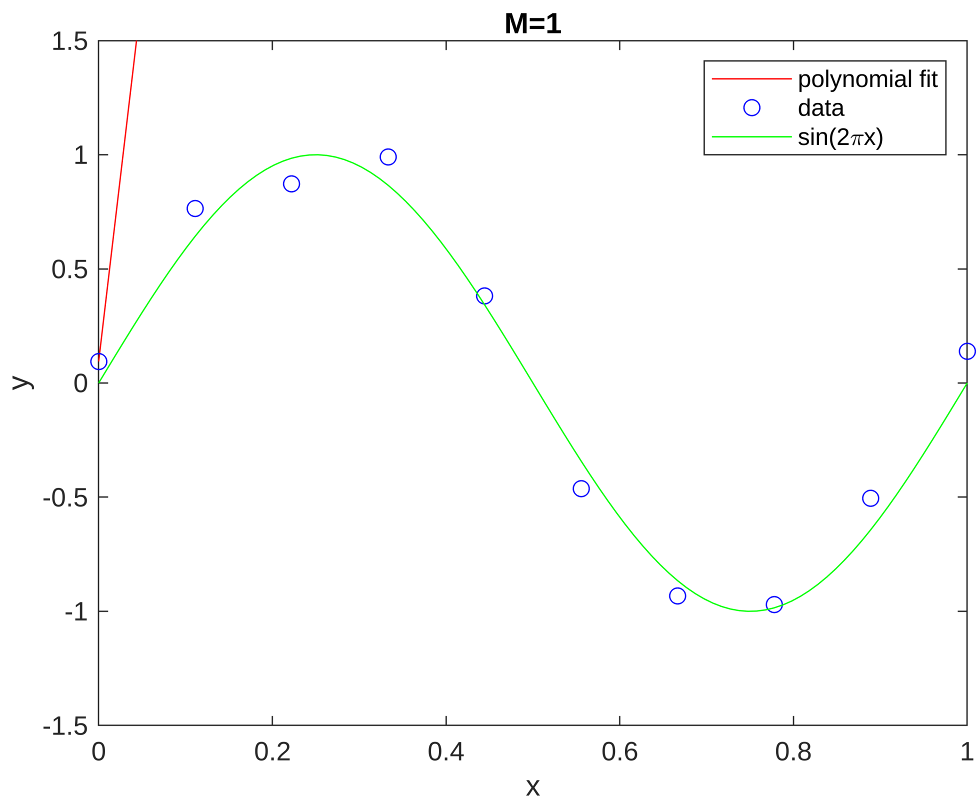
```
figure;
plot(x,z,"r")
hold on
plot(x_lrn,y_lrn,'ob')
plot(x,y,"g")
hold off
legend("polynomial fit", "data", "sin(2\pix)")
xlabel("x")
ylabel("y")
ylim([-1.5 1.5])
xlim([0 1])
```



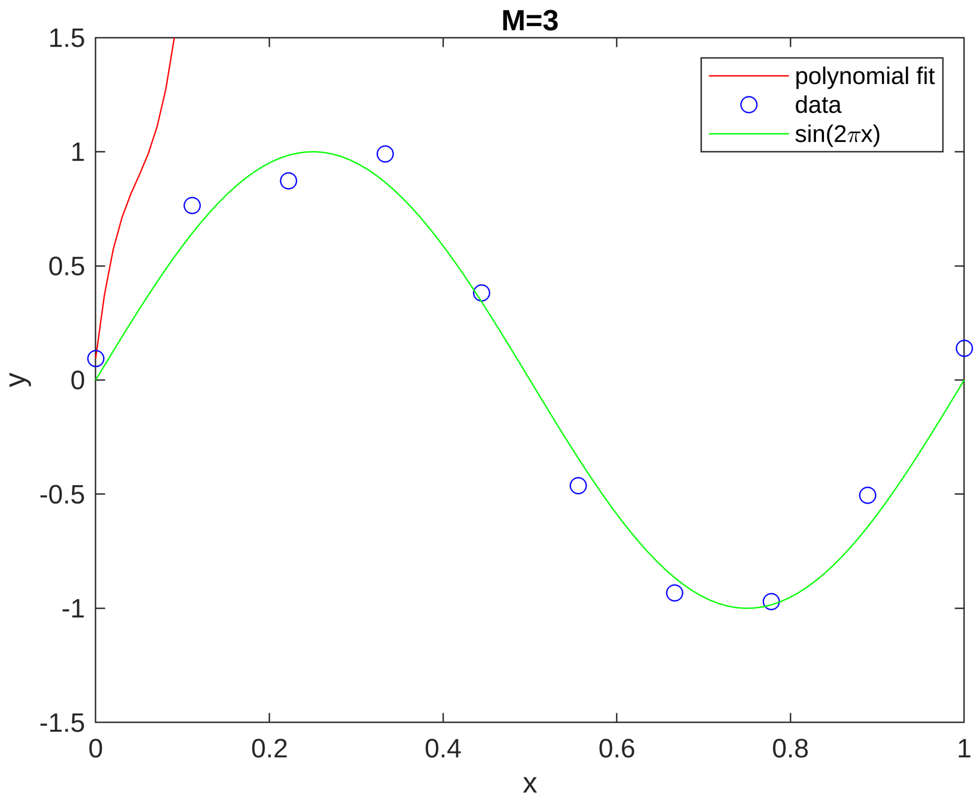
```
% plotting at different M (polynomial order)

% plot M = 0
figure;
plot(x,repelem(a(1),length(x)),"r")
hold on
plot(x_lrn,y_lrn,'ob')
plot(x,y,"g")
hold off
legend("polynomial fit", "data", "sin(2\pix)")
xlabel("x")
ylabel("y")
ylim([-1.5 1.5])
xlim([0 1])
title("M=0")
```



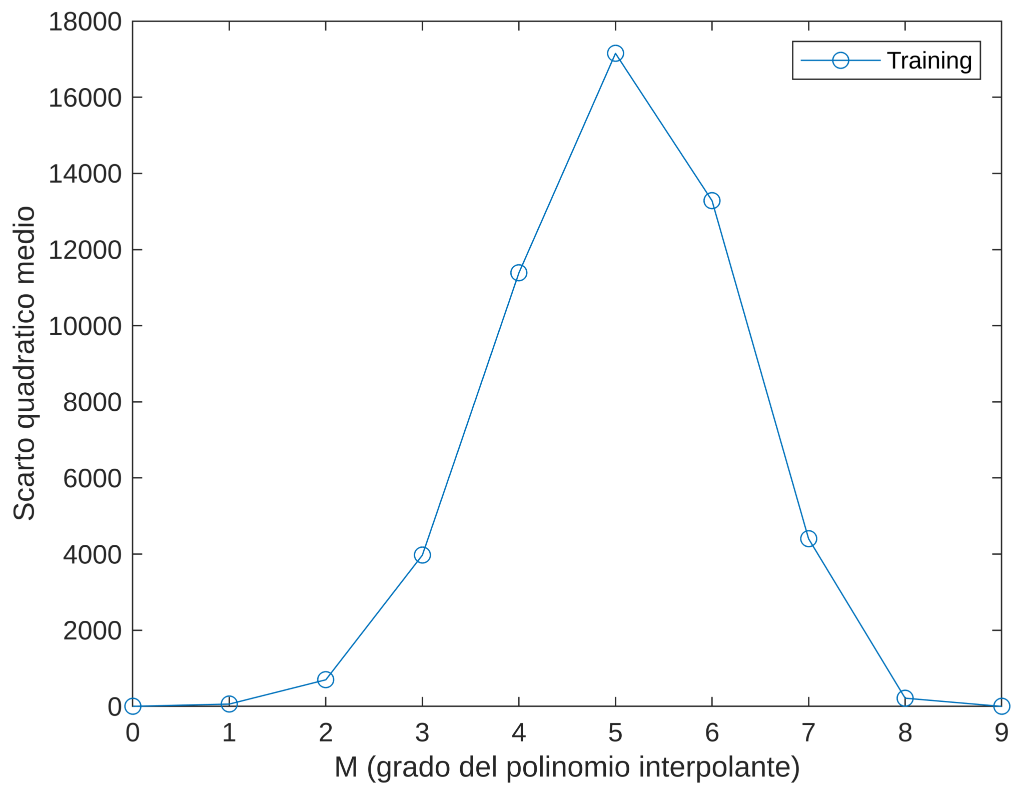
```
% plot M = 1
figure;
plot(x,(a(1)+a(2).*x),"r")
hold on
plot(x_lrn,y_lrn,'ob')
plot(x,y,"g")
hold off
legend("polynomial fit", "data", "sin(2\pix)")
xlabel("x")
ylabel("y")
ylim([-1.5 1.5])
xlim([0 1])
title("M=1")
```



```
% plot M = 3
figure;
plot(x,(a(1) + (a(2).*x) + a(3).*x.^2 + a(4).*x.^3),"r")
hold on
plot(x_lrn,y_lrn,'ob')
plot(x,y,"g")
hold off
legend("polynomial fit", "data", "sin(2\pix)")
xlabel("x")
ylabel("y")
ylim([-1.5 1.5])
xlim([0 1])
title("M=3")
```



```
% learning error

% initializing vectors
learning_error = zeros(1,n_lrn);
y_fit = learning_error;

for j = 1:n_lrn

    % estimate y points
    for i = 1:n_lrn
        y_fit(i) = poly(x_lrn(i),j-1);
    end

    % calculating learning error
    learning_error(j) = sqrt(sum((y_fit-y_lrn).^2));
end

% plotting learning error
plot(0:n_lrn-1,learning_error,"-o")
xlabel("M (grado del polinomio interpolante)")
ylabel("Scarto quadratico medio")
legend("Training")
```



Training
Scarto quadratico medio
M (grado del polinomio interpolante)
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
0
1
2
3
4
5
6
7
8
9