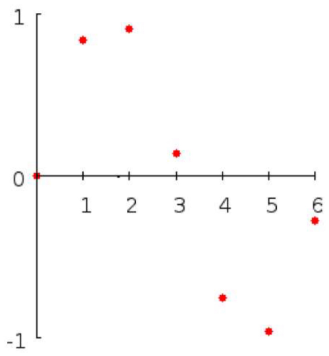
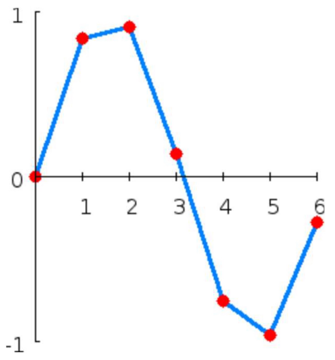
Interpolation :

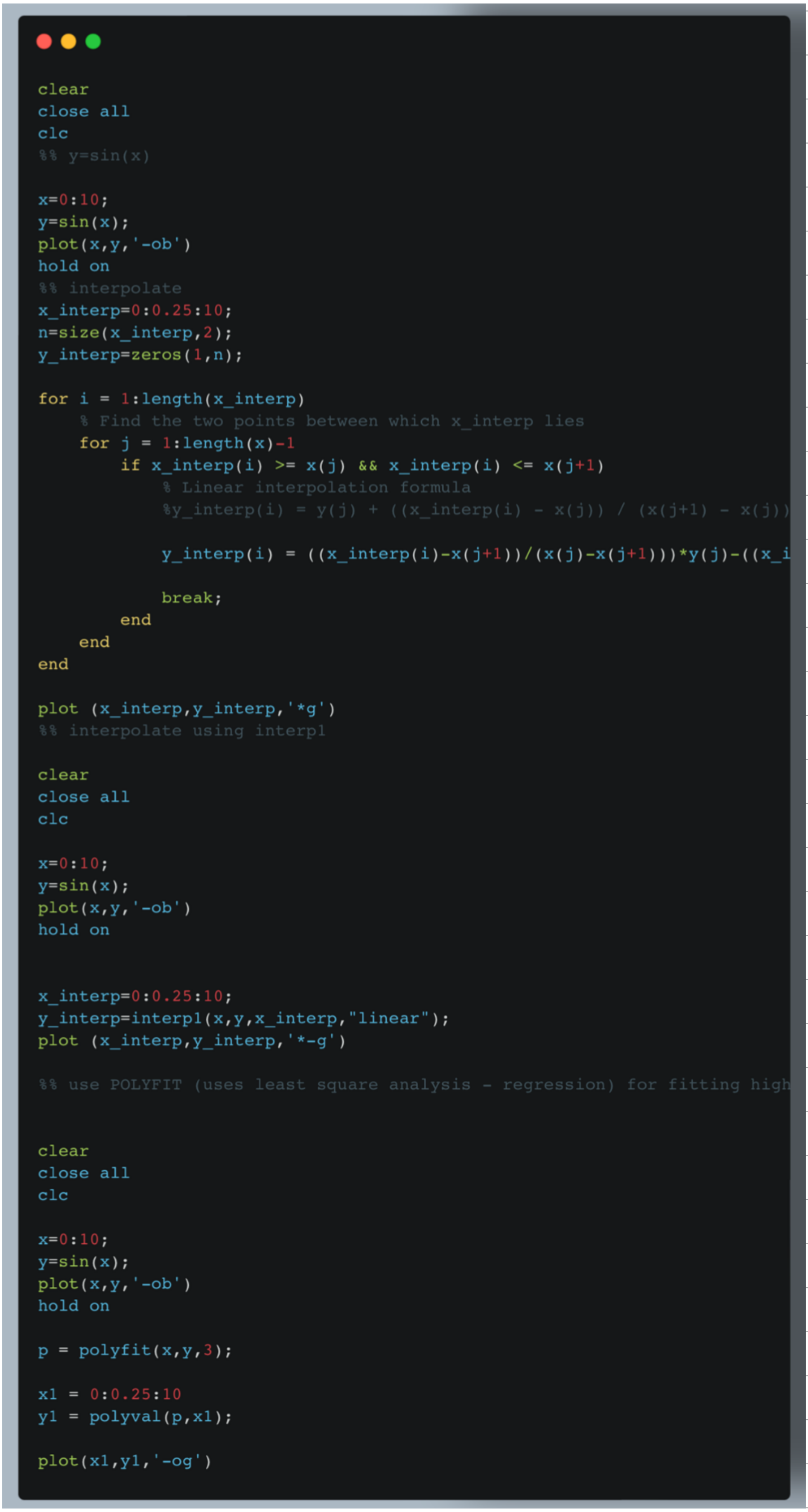
- Deriving a function from given discrete data points
- Passing through the points



linear interpolation



Exercise-4. *y* = sin(*x*); *x* = 0:10; *x*(*i*) = 0:0.25:10; Construct the interpolant *y* and plot.



```
clear
close all
clc
%% y=sin(x)

x=0:10;
y=sin(x);
plot(x,y,'-ob')
hold on
%% interpolate
x_interp=0:0.25:10;
n=size(x_interp,2);
y_interp=zeros(1,n);

for i = 1:length(x_interp)
    % Find the two points between which x_interp lies
    for j = 1:length(x)-1
        if x_interp(i) >= x(j) && x_interp(i) <= x(j+1)
            % Linear interpolation formula
            %y_interp(i) = y(j) + ((x_interp(i) - x(j)) / (x(j+1) - x(j))

            y_interp(i) = ((x_interp(i)-x(j+1))/(x(j)-x(j+1)))*y(j)-((x_i

            break;
        end
    end
end

plot (x_interp,y_interp,'*g')
%% interpolate using interp1

clear
close all
clc

x=0:10;
y=sin(x);
plot(x,y,'-ob')
hold on


x_interp=0:0.25:10;
y_interp=interp1(x,y,x_interp,"linear");
plot (x_interp,y_interp,'*-g')

%% use POLYFIT (uses least square analysis - regression) for fitting high


clear
close all
clc

x=0:10;
y=sin(x);
plot(x,y,'-ob')
hold on

p = polyfit(x,y,3);

x1 = 0:0.25:10
y1 = polyval(p,x1);

plot(x1,y1,'-og')
```

Curve fitting

Least Square Regression

- only 1 best fitted curve

Regression:

1. Used to model relationships between variables, especially when data contains noise or inaccuracies.

2. Does not require the model to pass through all data points.

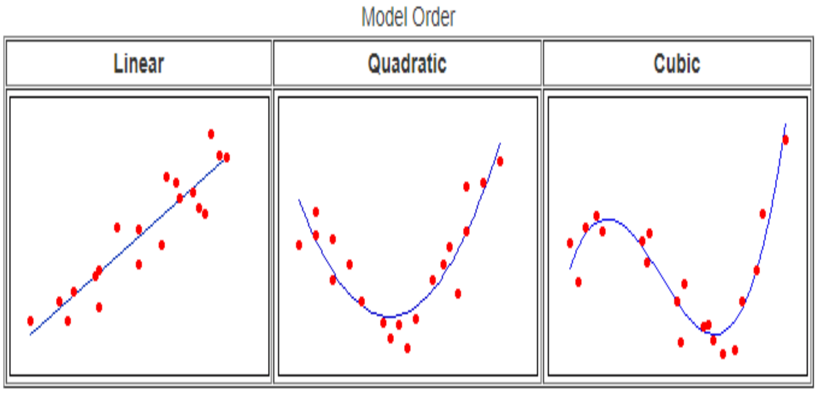
3. The goal is to find the best-fit curve or line that approximates the relationship between variables.

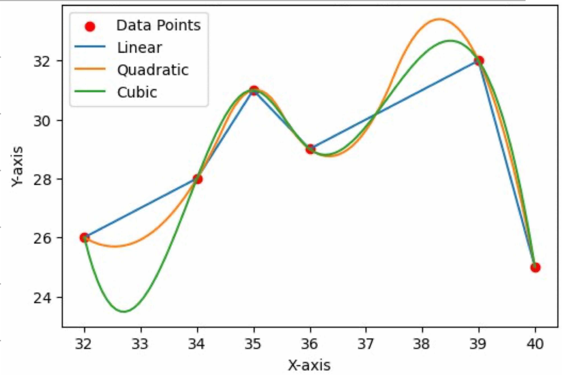
Interpolation:

1. Used to estimate unknown values within a given range of discrete data points.

2. Assumes that the data points are exact.

3. The goal is to find a function that passes through all the data points.





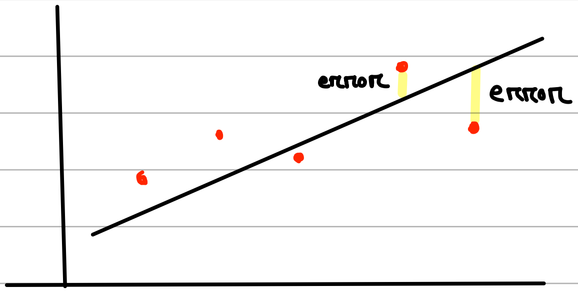
Linear Regression

$$y_m = a_0 + a_1 x$$

$$e = y_o - y_m$$

Observed ↗ ↖ Model

$$e = y_0 - a_0 - a_1 x$$



$$\sum_{i=1}^{n} e_i^2 = (y_i - a_0 - a_1 x_i)^2 \longrightarrow \text{sum of sq. errors}$$

$\longleftarrow S_r$

minimum

$$\frac{\partial S_r}{\partial a_0} = 0$$

$$\frac{\partial S_r}{\partial a_1} = 0$$

n = No. data points

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix}$$

Polynomial regression:

$$y_m = a_0 + a_1 x + a_2 x^2$$

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}$$

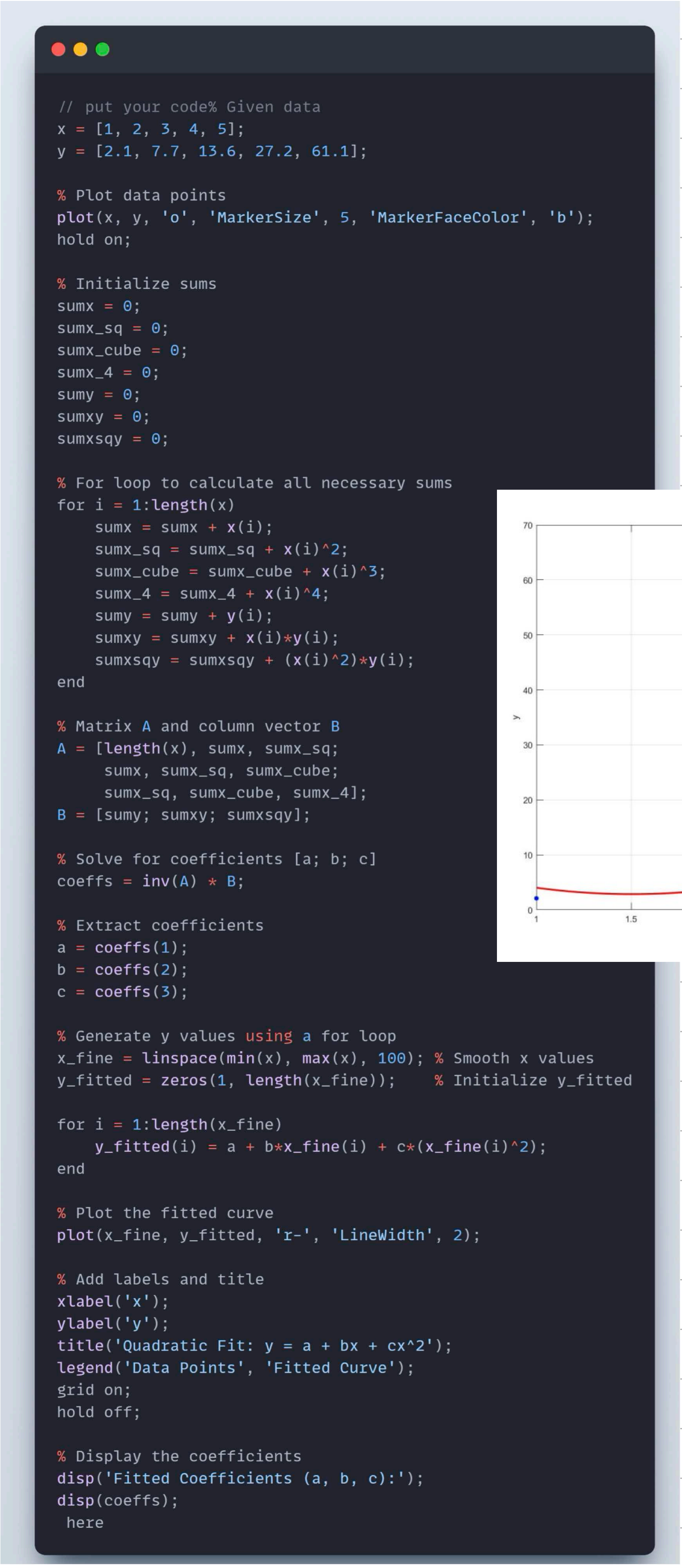
n → x_i গুণ

y_i

↓ x_i গুণ

Home work-3:

Fit a second order polynomial to the data given in table-4 and determine the value of the co-efficient



```
// put your code% Given data
x = [1, 2, 3, 4, 5];
y = [2.1, 7.7, 13.6, 27.2, 61.1];

% Plot data points
plot(x, y, 'o', 'MarkerSize', 5, 'MarkerFaceColor', 'b');
hold on;

% Initialize sums
sumx = 0;
sumx_sq = 0;
sumx_cube = 0;
sumx_4 = 0;
sumy = 0;
sumxy = 0;
sumxsqy = 0;

% For loop to calculate all necessary sums
for i = 1:length(x)
    sumx = sumx + x(i);
    sumx_sq = sumx_sq + x(i)^2;
    sumx_cube = sumx_cube + x(i)^3;
    sumx_4 = sumx_4 + x(i)^4;
    sumy = sumy + y(i);
    sumxy = sumxy + x(i)*y(i);
    sumxsqy = sumxsqy + (x(i)^2)*y(i);
end

% Matrix A and column vector B
A = [length(x), sumx, sumx_sq;
     sumx, sumx_sq, sumx_cube;
     sumx_sq, sumx_cube, sumx_4];
B = [sumy; sumxy; sumxsqy];

% Solve for coefficients [a; b; c]
coeffs = inv(A) * B;

% Extract coefficients
a = coeffs(1);
b = coeffs(2);
c = coeffs(3);

% Generate y values using a for loop
x_fine = linspace(min(x), max(x), 100); % Smooth x values
y_fitted = zeros(1, length(x_fine));    % Initialize y_fitted

for i = 1:length(x_fine)
    y_fitted(i) = a + b*x_fine(i) + c*(x_fine(i)^2);
end

% Plot the fitted curve
plot(x_fine, y_fitted, 'r-', 'LineWidth', 2);

% Add labels and title
xlabel('x');
ylabel('y');
title('Quadratic Fit: y = a + bx + cx^2');
legend('Data Points', 'Fitted Curve');
grid on;
hold off;

% Display the coefficients
disp('Fitted Coefficients (a, b, c):');
disp(coeffs);
 here
```

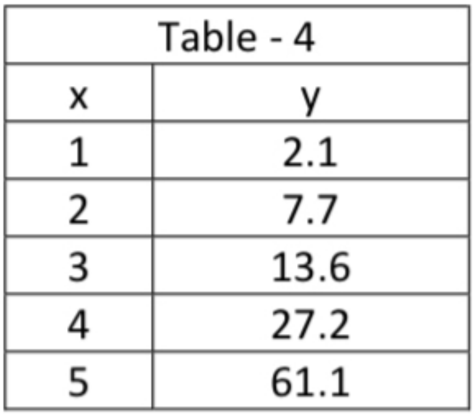
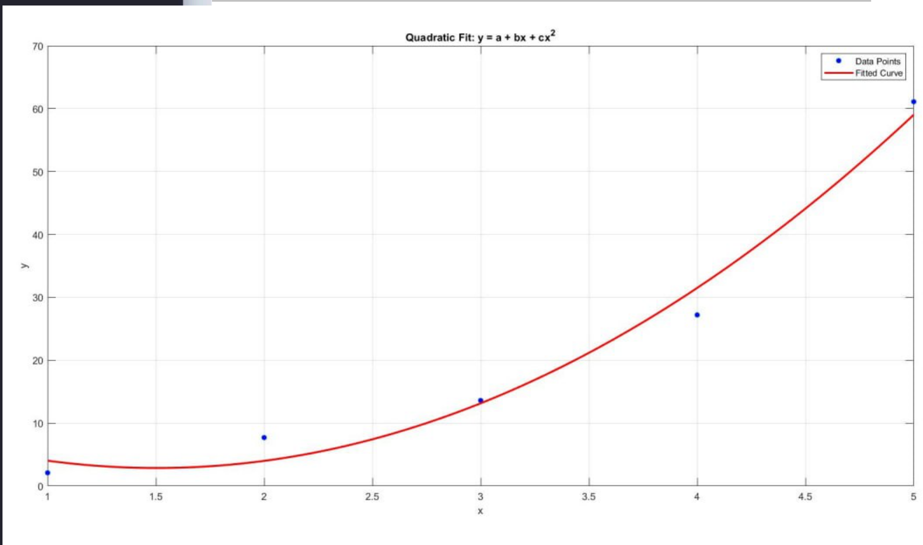


Table - 4	
x	y
1	2.1
2	7.7
3	13.6
4	27.2
5	61.1


Quadratic Fit: y = a + bx + cx^2
Data Points
Fitted Curve
y
x