



Khulna University of Engineering & Technology, Khulna

Department of Biomedical Engineering

SESSIONAL REPORT

Course No: **BME 2152**

Experiment No : **04**

Name of the Experiment: **Performing Curve Fitting using MATLAB.**

Remarks:

Date of performance : 8 January 2025

Date of submission : 15 January 2025

Name: **Md. Fuad Hasan Hamim**

Roll : **2215029**

Year : **2nd**

Term : **1st**

Objectives:

The main objectives of this sessional are:

- To learn about curve fitting.
- To learn how to construct curve diagram in MATLAB.
- To learn use of text, gtext, legend, hold, line command, residuals, find out least sequence line in MATLAB.

Introduction:

Curve fitting is the methodology through which a curve or a mathematical function is fitted to a set of data points, possibly with constraints. It may involve interpolation-where exact data is needed-or smoothing, mainly to reduce noise in data. Curve fitting is allied to regression, which is devoted to the estimation of the relationships between variables, especially when data is random or uncertain.

Fitted curves are helpful for data visualization, finding relationships, or making predictions. Extrapolation (extending the curve beyond observed data) carries uncertainty, as it assumes the curve reflects the underlying data accurately.

Commands Used in This Experiment:

Why not use ‘subplot’?

Using ‘subplot’ creates diagrams on separate pages, making comparison harder. Instead, we use commands like ‘plot(x, y, 'o')’ for scatter plots and ‘plot(x, y, '--')’ for dashed lines.

‘text’ Command

The ‘text’ command places labels or notes on a graph at a given set of coordinates. Syntax: ‘text(x, y, 'label')’. For 3D plots: ‘text(x, y, z, 'label')’.

‘gtext’ Command

The ‘gtext’ command allows you to interactively click on the graph where you want to place text. Type ‘gtext('label')’ then click the mouse on the graph where you want to place the text and press Enter.

‘legend’ Command

This constructs a legend for the graph. If no axes exist, an empty legend will be created.

‘hold’ Command

The ‘hold’ command toggles the behavior of whether new plots clear and replace the existing graph or build upon it. Default behavior clears graphs. Use ‘hold on’ to create overlaid plots, ‘hold off’ to return to default.

The ‘axis’ command changes graph scaling and aspect ratio. To have equal scaling along all dimensions, use ‘axis equal’ or ‘axis square’ for square axes.

‘line’ Command

The ‘line(x, y)’ command plots a line through given data points.

Residuals

Residuals represent the observed data minus the fitted curve. To obtain residuals:

1. Open Figure 1.
2. Select Tools > Basic Fitting.
3. Select a fit type.
4. Click Fit Results.
5. Choose ‘Error Estimation (Residuals)’.

Task 1 || Construction of a Scatter diagram, Straight line and dashed line.

Code:

```
% Construction of a Scatter diagram, Straight line and dashed line.%
```

```
clc;
```

```
clear all;
```

```
x = [2, 3, 5, 7, 9, 11];
```

```
y = [1,3,7,11,15,19];
```

```
z = y .^ (3/4);
```

```
p = y .^ (1/2);
```

```
plot(x, y, 'o', x, z, '--', x, p);
```

```
xlabel('X-axis');
```

```
ylabel('Y-axis');
```

```
title('X vs Y Graph');
```

```
grid on;
```

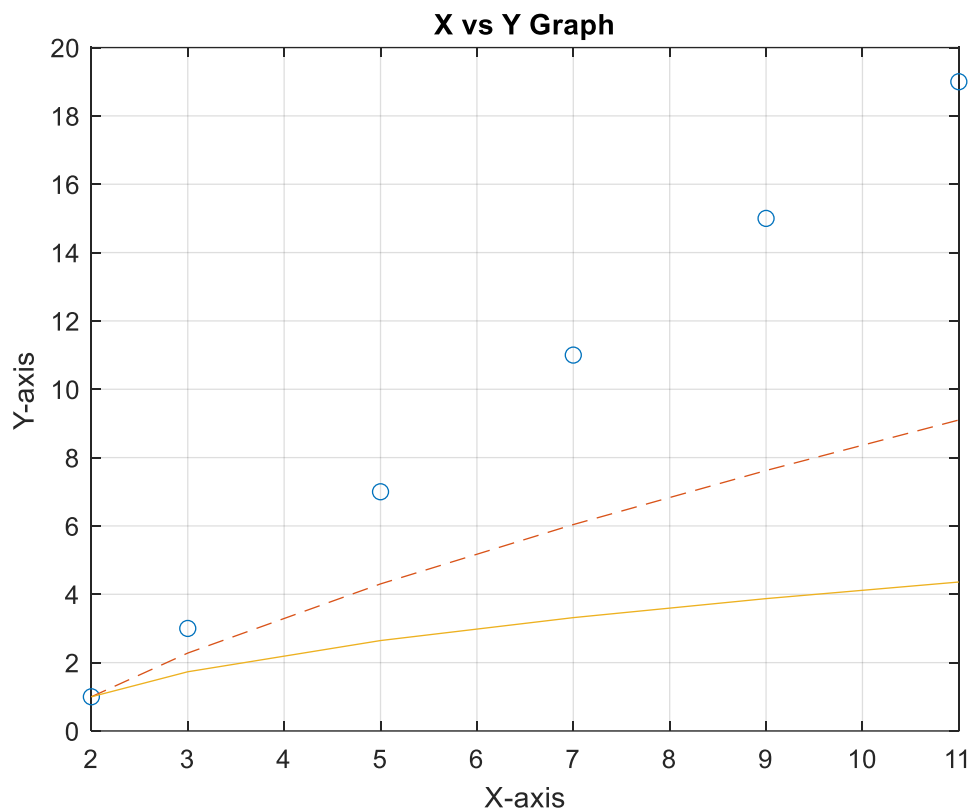


Figure 4,1(a) : 3 Graph with axis information and title.

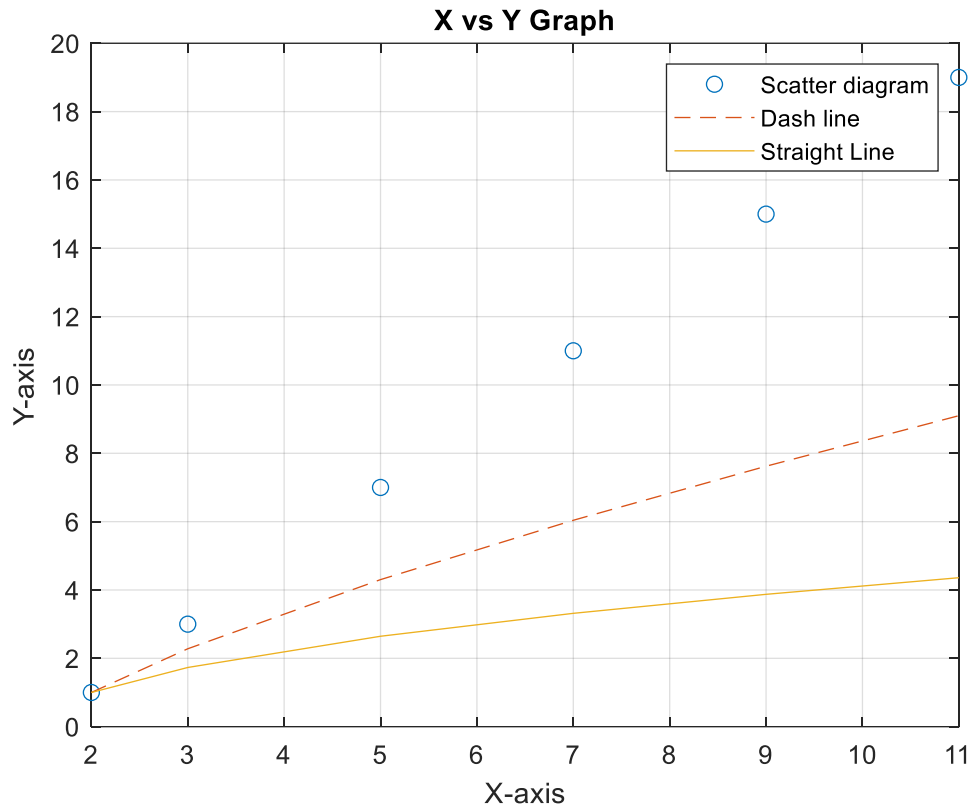


Figure 4,1(b) : 3 Graph with axis information, title and legend.

Comment: Extra code needed here are- legend('Scattered Diagram','Dash line','Straight line');

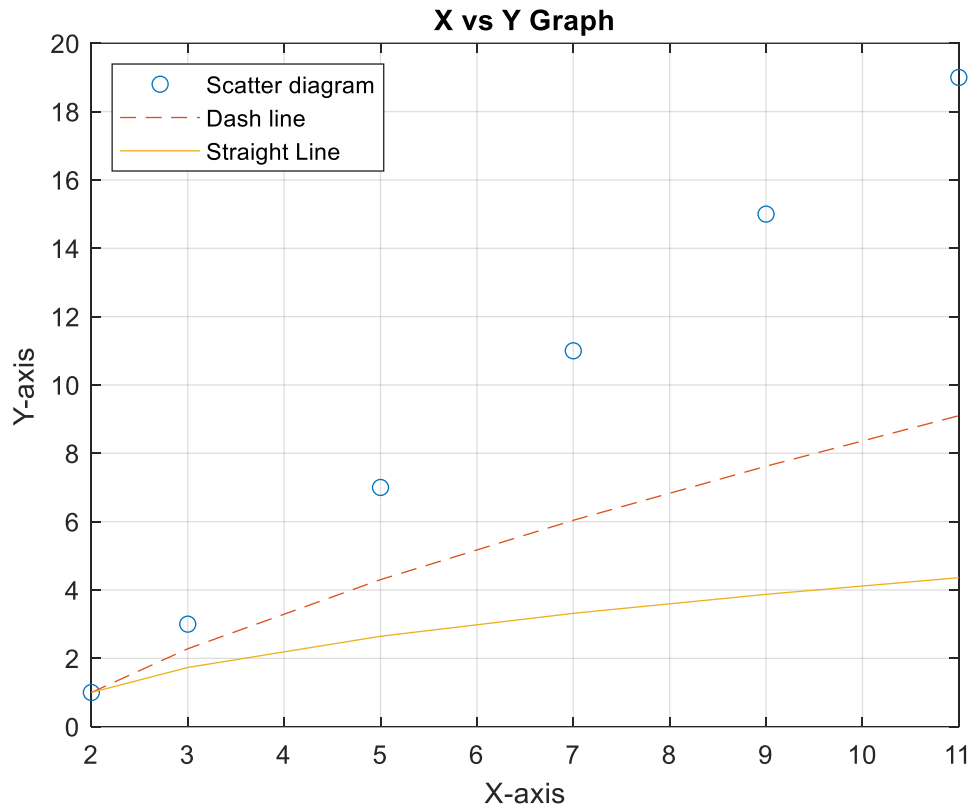


Figure 4,1(c) : 3 Graph with axis information, title and legend with changing legend location.

Comment: Extra coded needed for changing location-
legend('Scatter diagram','Dash line', 'Straight Line', 'Location', 'northwest');

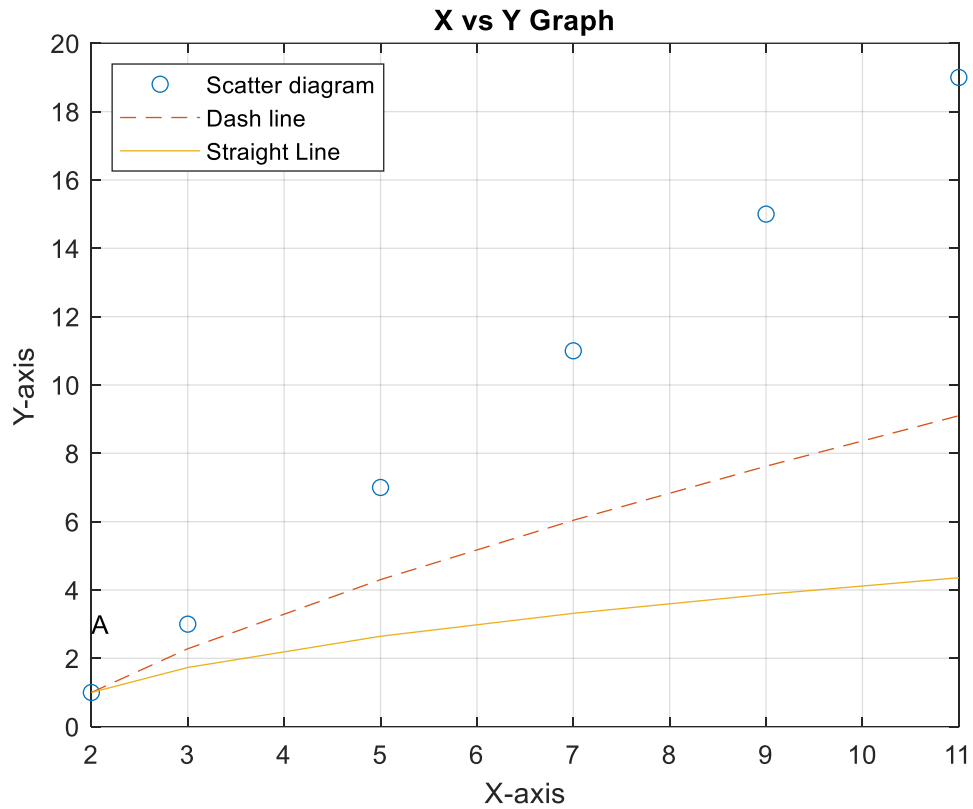


Figure 4,1(d) : 3 Graph with axis information, title, legend and text.

Comment: Extra code needed to write text on the figure is `-text(2,3,'A');`

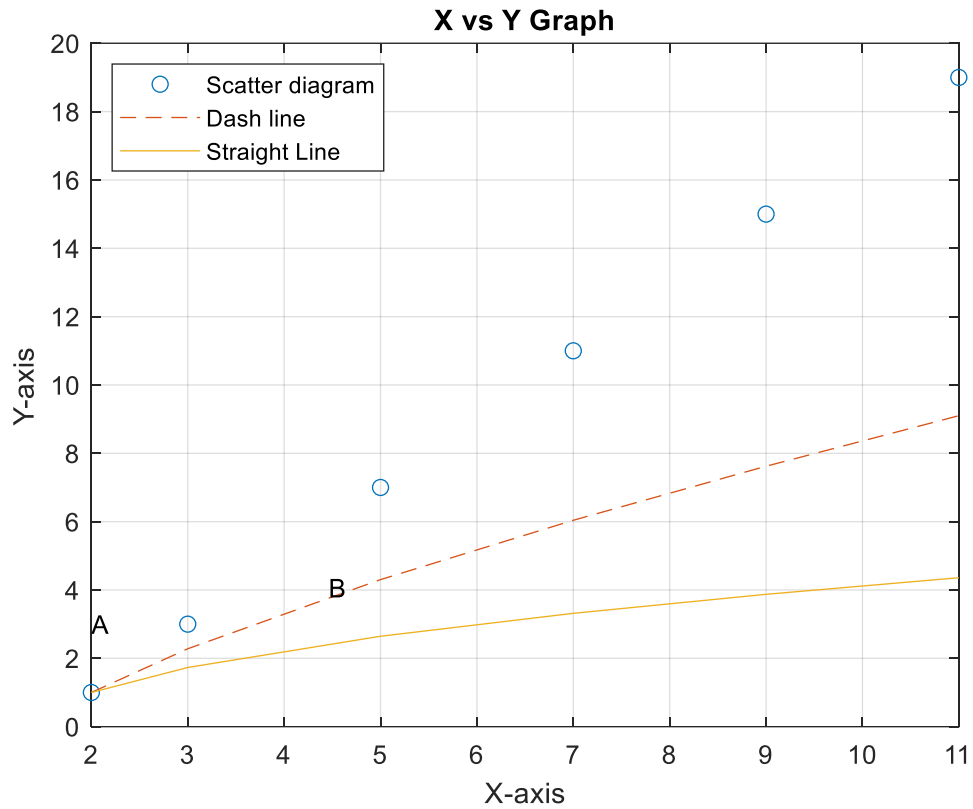


Figure 4,1(e) : 3 Graph with axis information, title, legend and text using gtext.

Comment: To create a gtext we needed to write-gtext('B');

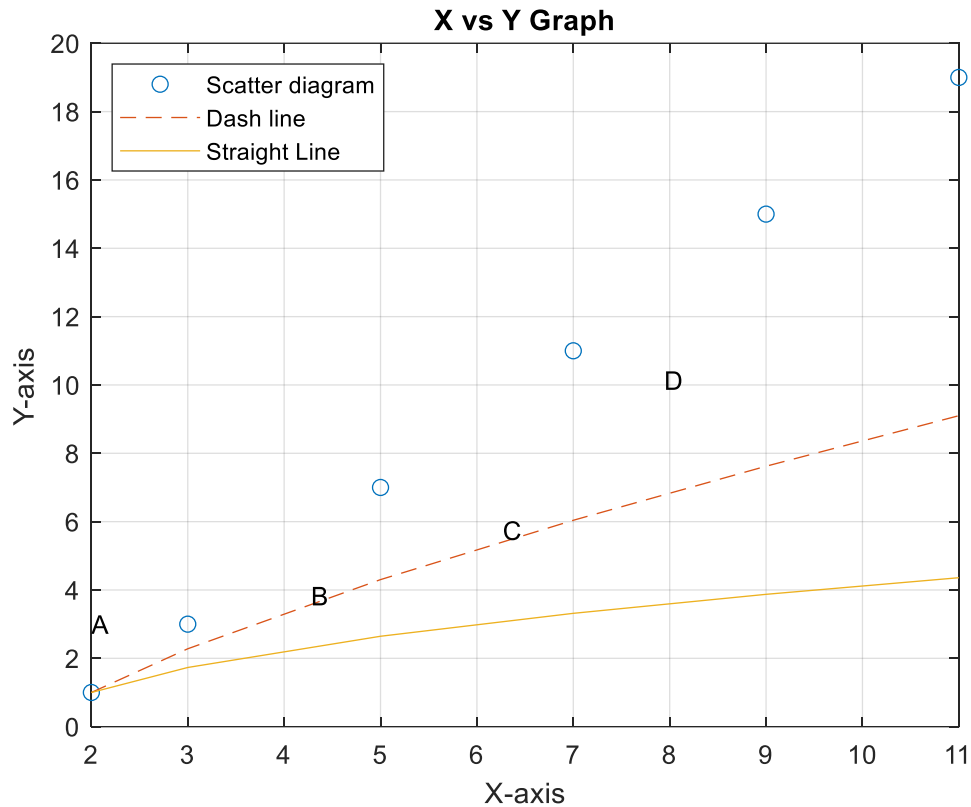


Figure 4,1(f) : 3 Graph with axis information, title, legend and text using gtext.

Comment: Same as before-

gtext('C');

gtext('D');

Task 2 || Construction of Scatter diagram and dashed line using hold on command and subplot.

(a) Using 'Subplot'

Code:

```
% Construction of Scatter diagram and dashed line using hold on command and
%subplot. %
clc;
clear all;
x = [2, 3, 5, 7, 9, 11];
y = [1,3,7,11,15,19];

hold on;
subplot 311;
plot(x, y,'O');
```

```

xlabel('Figure A');
grid on;
hold off;

```

```

subplot 312;
hold on;
plot(x, y, '--');
xlabel('Figure B');
grid on;
hold off;

```

```

subplot 313;
hold on;
plot(x,y,'--');
plot(x, y,'O');
xlabel('Figure C');
grid on;
hold off;

```

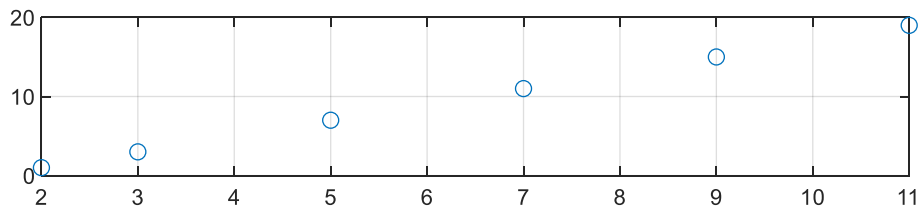


Figure A

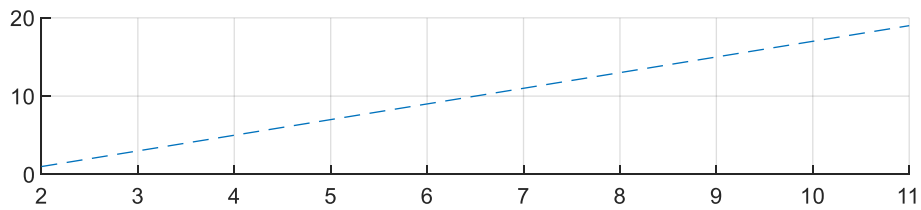


Figure B

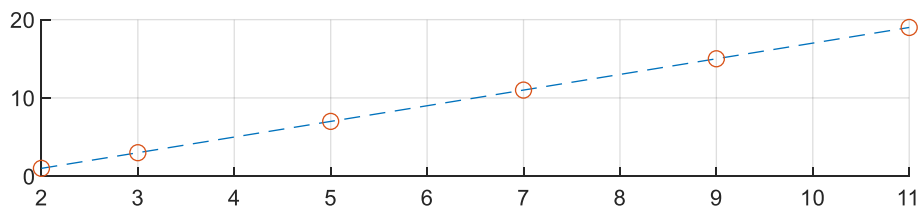


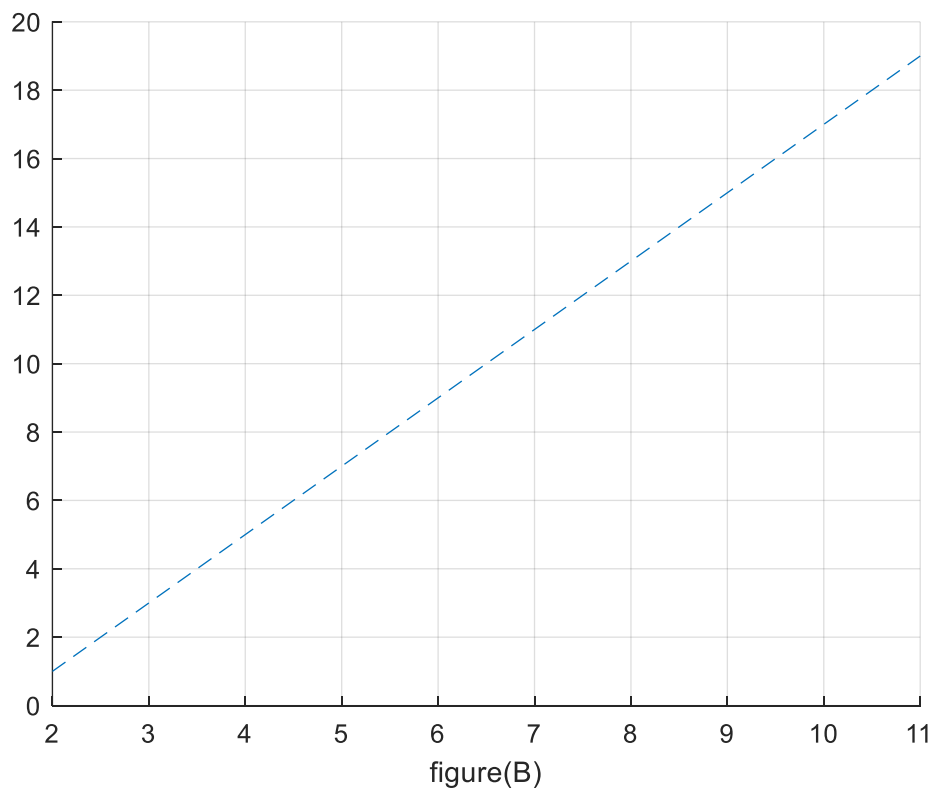
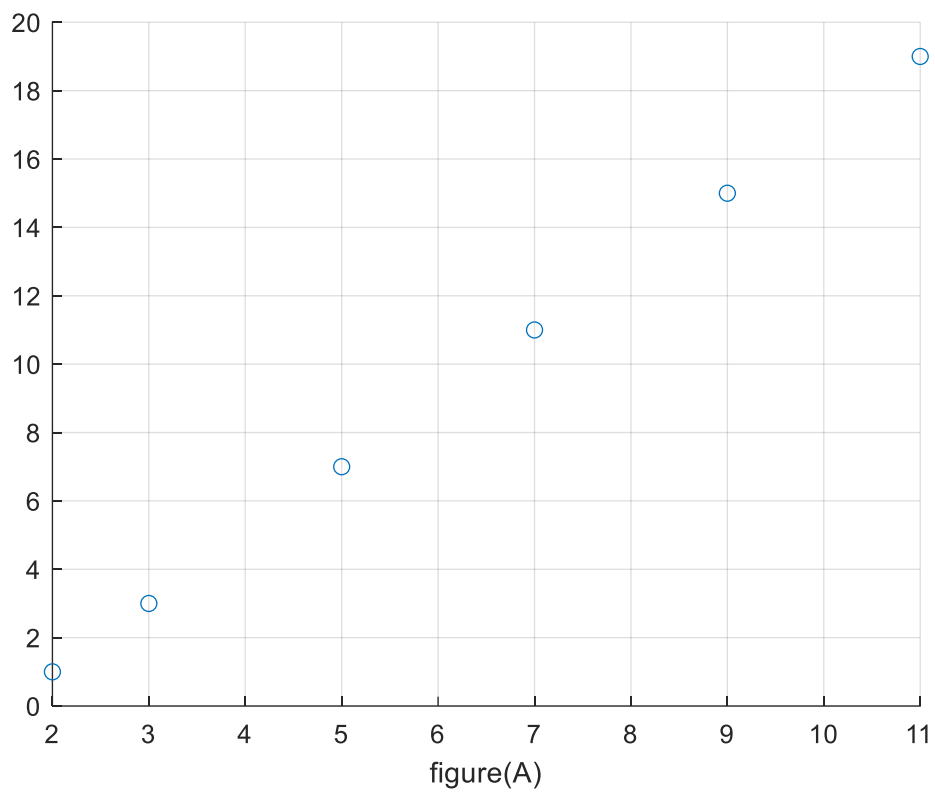
Figure C

Figure 4.2 (a): Construction of Scatter diagram and dashed line using hold on command and subplot.

(b) Using ‘Hold on’

Code:

```
clc;
clear all;
X =[2, 3, 5, 7, 9, 11];
Y =[1, 3, 7, 11, 15, 19];
figure(1);
hold on;
plot(X, Y, 'o');
xlabel('figure(A)');
grid on;
hold off;
figure(2);
hold on;
plot(X, Y, '--');
xlabel('figure(B)');
grid on;
hold off;
figure(3);
hold on;
plot(X, Y, 'o');
plot(X, Y, '--');
xlabel('figure(C)');
grid on;
hold off;
```



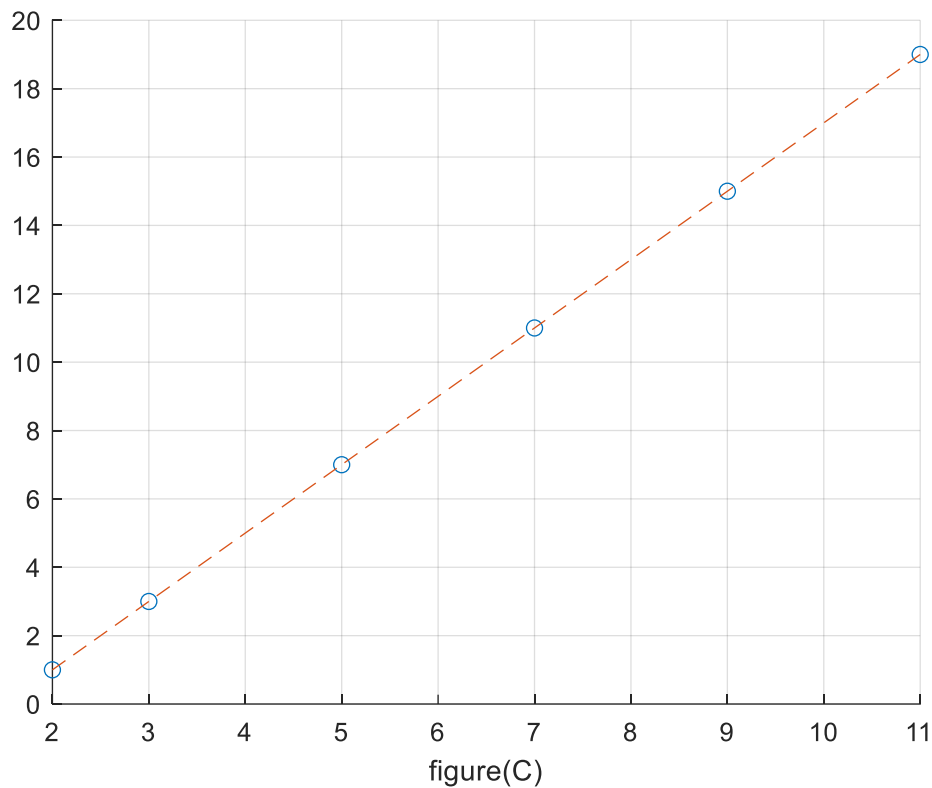


Figure 4.2(b) : Three graph on construction of Scatter diagram and dashed line using hold on command.

Comment: Hold on and Subplot functions provide almost same output. The only difference is 'Subplot' provides all plots in one figure whereas 'Hold on' provides all plots in individual figures.

Task 3 || Construction of scatter diagram and straight line using 'line command'.

Code:

```
clc;
clear all;

x = [2, 3, 5, 7, 9, 11];
y = [1, 3, 7, 11, 15, 19];
z = y .^ (3/4);
p = y .^ (1/2);

plot(x, y, 'o');
hold on;
```

```

line(x, z); % Line for z
line(x, p); % Line for p

% Labels and title
xlabel('X-axis');
ylabel('Y-axis');
title('Scatter Diagram and Straight Lines');
grid on;

axis equal;

hold off;

```

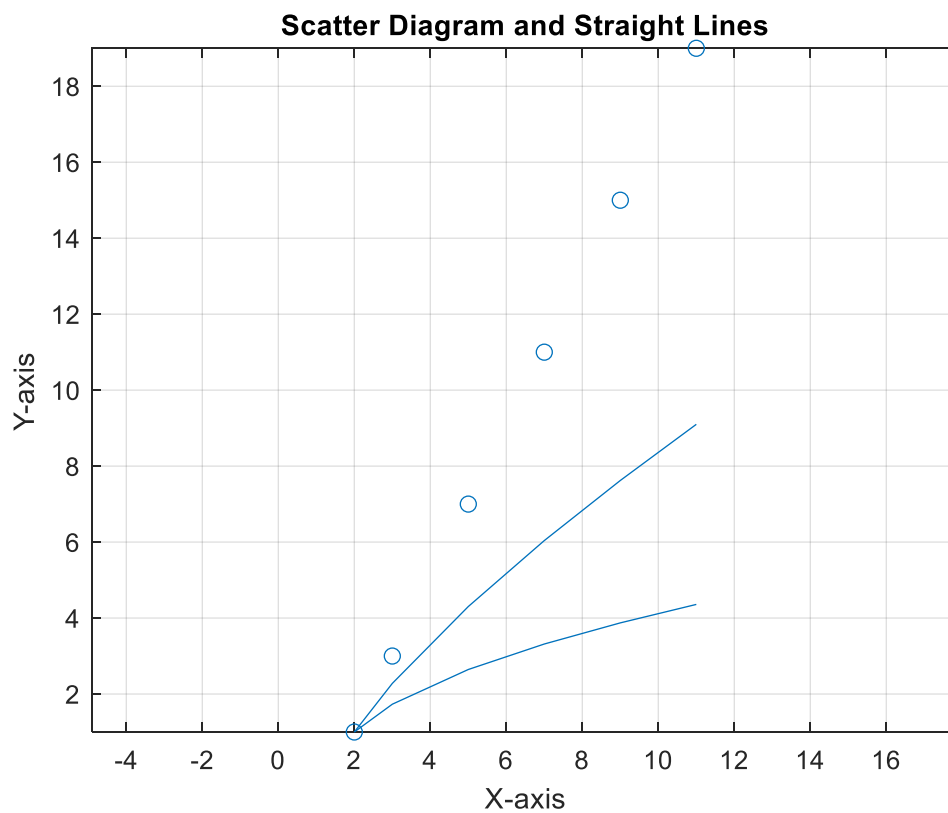


Figure 4.3: Construction of scatter diagram and straight line using 'line command'.

Task 4 || Performing curve fitting with plot residuals.

Code:

```
% Performing curve fitting with plot residuals. %  
clc;  
clear all;  
x = [2, 3, 5, 7, 9, 11];  
y = [1,3,7,11,15,19];  
plot(x, y,'o');  
xlabel('X- axis');  
ylabel('Y- axis');  
title('Performing curve fitting with plot residuals.');
```

%Figure Window >> Tools >> Basic Fitting >> Set TYPES OF FIT = Linear
%Select FIT RESULTS = Equation
%Select ERROR ESTIMATION(RESIDUALS) = Line

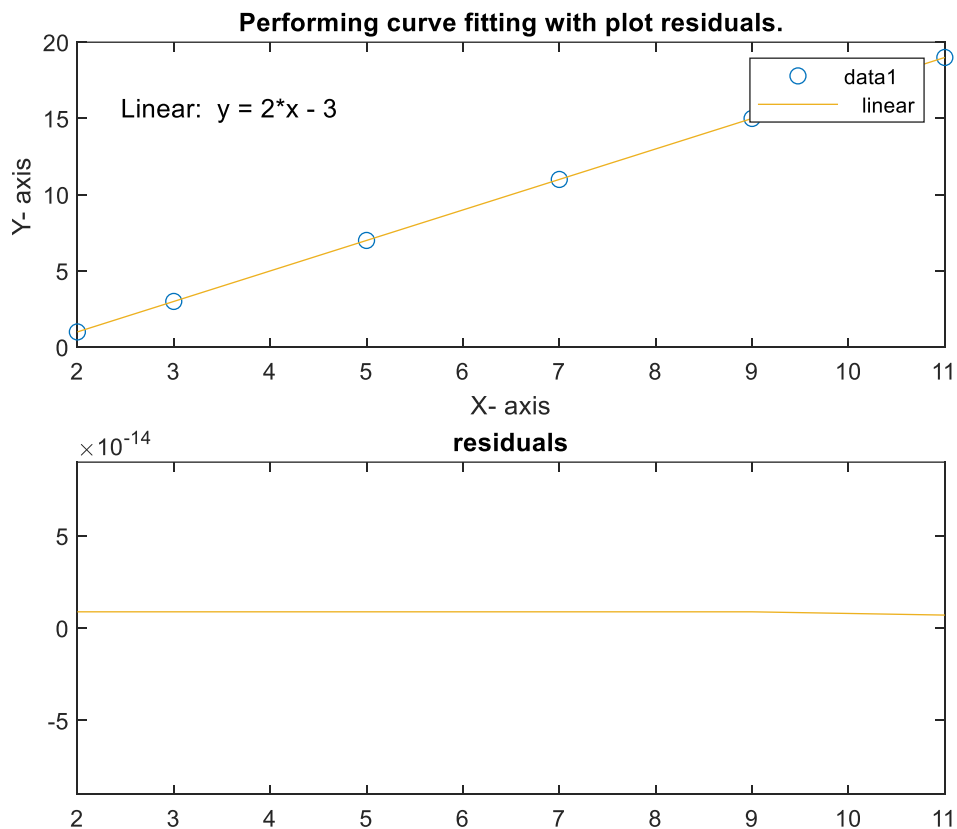


Figure 4.4: Performing curve fitting with plot residuals.

Comment: Extra operation needed are-

1. Figure Window >> Tools >> Basic Fitting >> Set TYPES OF FIT = Linear

2. Select FIT RESULTS = Equation
3. Select ERROR ESTIMATION(RESIDUALS) = Line

Task 5 || Performing curve fitting (First-order polynomial).

Code:

```
% Performing curve fitting (First-order polynomial).
```

```
clc;  
clear all;  
x=[2 3.5 5 7 9 11];  
y=[1 3 7 12 15 17];  
a=polyfit(x,y,1);%1 is the order of the equation  
x_fit=0:0.1:12;  
y_fit=polyval(a,x_fit);  
plot(x,y,'o',x_fit,y_fit);  
grid;  
title('curve fitting of 1st order polynomial');  
xlabel('x axis');  
ylabel('y axis');
```

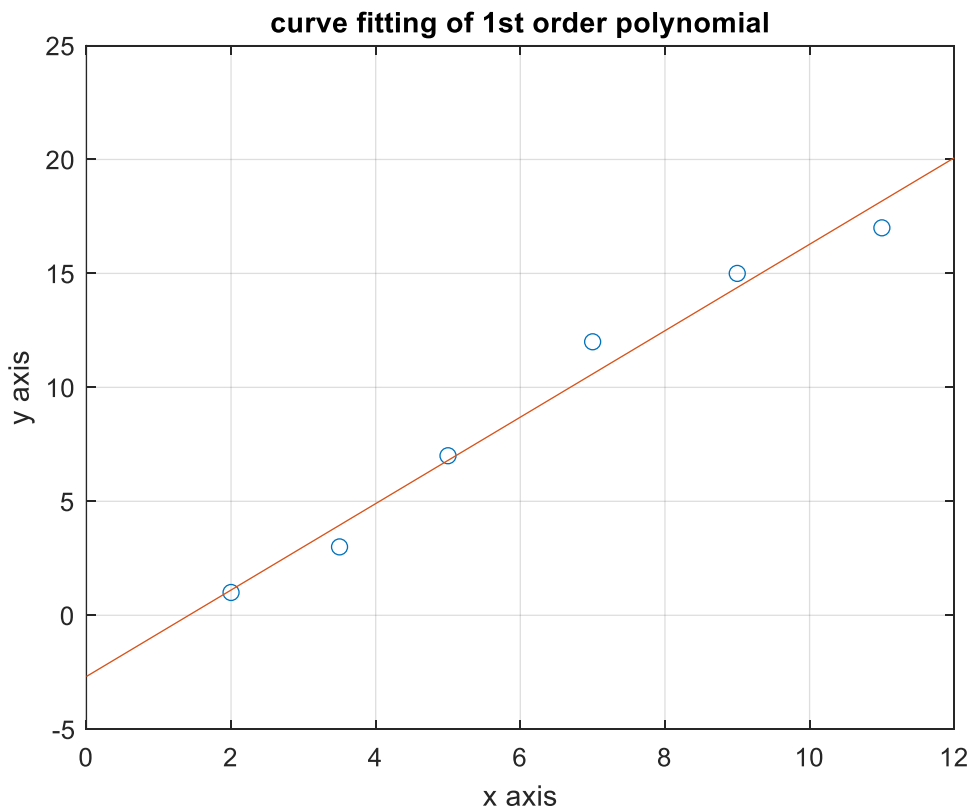


Figure 4.5: Performing curve fitting (First-order polynomial).

Conclusion:

In this session, we learned how to use MATLAB for curve fitting and data visualization. We practiced creating different diagrams like scatter plots, straight lines, and dashed lines. We also used important commands such as text, gtext, legend, hold, line, and axis. Using the polyfit function, we fitted a first-order polynomial to the data and checked the accuracy by analyzing residuals. We also learned how to calculate errors and residuals using the basic fitting tools. This session helped us understand how to use MATLAB for analyzing and visualizing data. The skills we gained will be helpful in future programming and data analysis tasks.