The solution to Sample Problem 5-1, which is the plot in Figure 5-7, can be obtained by using the plot and line commands as shown in the following script file:

```
x=[-2:0.01:4];
y=3*x.^3-26*x+6;
yd=9*x.^2-26;
ydd=18*x;
plot(x,y,'LineStyle','-','color','b')
line(x,yd,'LineStyle','--','color','r')
line(x,ydd,'linestyle',':','color','k')
```

5.4 FORMATTING A PLOT

The plot and fplot commands create bare plots. Usually, however, a figure that contains a plot needs to be formatted to have a specific look and to display information in addition to the graph itself. This can include specifying axis labels, plot title, legend, grid, range of custom axis, and text labels.

Plots can be formatted by using MATLAB commands that follow the plot or fplot command, or interactively by using the plot editor in the Figure Window. The first method is useful when a plot command is a part of a computer program (script file). When the formatting commands are included in the program, a formatted plot is created every time the program is executed. On the other hand, formatting that is done in the Figure Window with the plot editor after a plot has been created holds only for that specific plot, and will have to be repeated the next time the plot is created.

5.4.1 Formatting a Plot Using Commands

The formatting commands are entered after the plot or the fplot command. The various formatting commands are:

The xlabel and ylabel commands:

Labels can be placed next to the axes with the xlabel and ylabel command which have the form:

```
xlabel(‘text as string’)
ylabel(‘text as string’)
```

The title command:

A title can be added to the plot with the command:

```
title(‘text as string’)
```

The text is placed at the top of the figure as a title.

The `text` command:

A text label can be placed in the plot with the `text` or `gtext` commands:

```
text(x,y,‘text as string’)
gtext(‘text as string’)
```

The `text` command places the text in the figure such that the first character is positioned at the point with the coordinates `x`, `y` (according to the axes of the figure). The `gtext` command places the text at a position specified by the user. When the command is executed, the Figure Window opens and the user specifies the position with the mouse.

The `legend` command:

The `legend` command places a legend on the plot. The legend shows a sample of the line type of each graph that is plotted, and places a label, specified by the user, beside the line sample. The form of the command is:

```
legend(‘string1’,‘string2’, ..... ,‘Location’,‘pos’)
```

The strings are the labels that are placed next to the line sample. Their order corresponds to the order in which the graphs were created. The `‘Location’,‘pos’` are optional strings that specifies where in the figure the legend is to be placed. Several options are:

`NE`	Places the legend at the upper-right corner of the plot (default).
`NW`	Places the legend at the upper-left corner of the plot.
`SE`	Places the legend at the lower-right corner of the plot.
`SW`	Places the legend at the lower-left corner of the plot.
`B`	Places the legend inside the plot in a location that interferes the least with the graphs.
`BO`	Places the legend in a least unused space outside the plot.

To read about other options for the position of the legend type `help legend` in the Command Window.

Formatting the text within the `xlabel, ylabel, title, text` and `legend` commands:

The text in the string that is included in the command and is displayed when the command is executed can be formatted. The formatting can be used to define the font, size, position (superscript, subscript), style (italic, bold, etc.), and color of the characters, the color of the background, and many other details of the display. Some of the more common formatting possibilities are described below. A complete explanation of all the formatting features can be found in the Help Window under Text and Text Properties. The formatting can be done either by adding modifiers inside the string, or by adding to the command optional `PropertyName` and `PropertyValue` arguments following the string.

The modifiers are characters that are inserted within the string. Some of the modifiers that can be added are:

Modifier	Effect
`\bf`	bold font
`\it`	italic style
`\rm`	normal font

Modifier	Effect
`\fontname{fontname}`	specified font is used
`\fontsize{fontsize}`	specified font size is used

These modifiers affect the text from the point at which they are inserted until the end of the string. It is also possible to have the modifiers applied to only a section of the string by typing the modifier and the text to be affected inside braces { }.

Subscript and superscript:

A single character can be displayed as a subscript or a superscript by typing _ (the underscore character) or ^ in front of the character, respectively. Several consecutive characters can be displayed as a subscript or a superscript by typing the characters inside braces { } following the _ or the ^.

Greek characters:

Greek characters can be included in the text by typing `\name of the letter` within the string. To display a lowercase Greek letter, the name of the letter should be typed in all lowercase English characters. To display a capital Greek letter, the name of the letter should start with a capital letter. Some examples are:

Characters in the string	Greek letter
`\alpha`	α
`\beta`	β
`\gamma`	γ
`\theta`	θ
`\pi`	π
`\sigma`	σ

Characters in the string	Greek letter
`\Phi`	Φ
`\Delta`	Δ
`\Gamma`	Γ
`\Lambda`	Λ
`\Omega`	Ω
`\Sigma`	Σ

Formatting of the text that is displayed by the `xlabel`, `ylabel`, `title`, and `text` commands can also be done by adding optional `PropertyName` and `PropertyValue` arguments following the string inside the command.

With this option, the `text` command, for example, has the form:

```
text(x,y,'text as string',PropertyName,PropertyValue)
```

In the other three commands the `PropertyName` and `PropertyValue` arguments are added in the same way. The `PropertyName` is typed as a string, and the `PropertyValue` is typed as a number if the property value is a number and as a string if the property value is a word or a letter character. Some of the Property Names and corresponding possible Property Values are:

Property name	**Description**	**Possible property values**
`Rotation`	Specifies the orientation of the text.	Scalar (degrees) Default: 0
`FontAngle`	Specifies italic or normal style characters.	`normal`, `italic` Default: `normal`
`FontName`	Specifies the font for the text.	Font name that is available in the system.
`FontSize`	Specifies the size of the font.	Scalar (points) Default: 10
`FontWeight`	Specifies the weight of the characters.	`light`, `normal`, `bold` Default: `normal`
`Color`	Specifies the color of the text.	Color specifiers (see Section 5.1).
`Background-Color`	Specifies the background color (rectangular area).	Color specifiers (see Section 5.1).
`EdgeColor`	Specifies the color of the edge of a rectangular box around the text.	Color specifiers (see Section 5.1). Default: none.
`LineWidth`	Specifies the width of the edge of a rectangular box around the text.	Scalar (points) Default: 0.5

The `axis` command:

When the `plot(x,y)` command is executed, MATLAB creates axes with limits that are based on the minimum and maximum values of the elements of `x` and `y`. The `axis` command can be used to change the range and the appearance of the axes. In many situations, a graph looks better if the range of the axes extend beyond the range of the data. The following are some of the possible forms of the `axis` command:

`axis([xmin,xmax,ymin,ymax])` Sets the limits of both the *x* and *y* axes (`xmin`, `xmax`, `ymin`, and `ymax` are numbers).

`axis equal` Sets the same scale for both axes.

`axis square` Sets the axes region to be square.

`axis tight` Sets the axis limits to the range of the data.

The `grid` command:

`grid on` Adds grid lines to the plot.

`grid off` Removes grid lines from the plot.

An example of formatting a plot by using commands is given in the following script file that was used to generate the formatted plot in Figure 5-1.

```
x=[10:0.1:22];
y=95000./x.^2;
xd=[10:2:22];
yd=[950  640  460  340  250  180  140];
plot(x,y,'-','LineWidth',1.0)
xlabel('DISTANCE (cm)')
ylabel('INTENSITY (lux)')
title('\fontname{Arial}Light Intensity as a Function of Distance','FontSize',14)
axis([8 24 0 1200])
text(14,700,'Comparison  between  theory  and  experiment.','EdgeColor','r','LineWidth',2)
hold on
plot(xd,yd,'ro--','linewidth',1.0,'markersize'
legend('Theory','Experiment',0)
hold off
```

Formatting text inside the `title` command.

Formatting text inside the `text` command.

5.4.2 Formatting a Plot Using the Plot Editor

A plot can be formatted interactively in the Figure Window by clicking on the plot and/or using the menus. Figure 5-8 shows the Figure Window with the plot of Figure 5-1. The Plot Editor can be used to introduce new formatting items or to modify formatting that was initially introduced with the formatting commands.