sible to assign 8 to a variable, and then use the variable when the function is typed in the `fplot` command.

- The function to be plotted can be typed as a function of any letter. For example, the function in the previous paragraph can be typed as `@ (t) 8*t.^2+5*cos(t)` or `@ (z) 8*z.^2+5*cos(z)'`.

`limits:` The limits argument is a vector with two elements that specify the domain of x [`xmin,xmax`], or a vector with four elements that specifies the domain of x and the limits of the y-axis [`xmin,xmax,ymin,ymax`].

`line specifiers:` The line specifiers are the same as in the `plot` command. For example, a plot of the function $y = x^2 + 4\sin(2x) - 1$ for $-3 \le x \le 3$ can be created with the `fplot` command by typing:

```
>> fplot(@ (x) x.^2+4*sin(2*x)-1,[-3 3])
```

in the Command Window. The figure that is obtained in the Figure Window is shown in Figure 5-6.

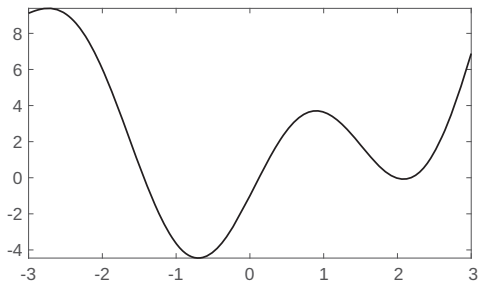

Figure 5-6: A plot of the function $y = x^2 + 4\sin(2x) - 1$.

5.3 PLOTTING MULTIPLE GRAPHS IN THE SAME PLOT

In many situations, there is a need to make several graphs in the same plot. This is shown, for example, in Figure 5-1 where two graphs are plotted in the same figure. There are three methods to plot multiple graphs in one figure. One is by using the `plot` command, the second is by using the `hold on` and `hold off` commands, and the third is by using the `line` command.

5.3.1 Using the `plot` Command

Two or more graphs can be created in the same plot by typing pairs of vectors inside the `plot` command. The command

```
plot(x,y,u,v,t,h)
```

creates three graphs—`y` vs. `x`, `v` vs. `u`, and `h` vs. `t`—all in the same plot. The vectors of each pair must be of the same length. MATLAB automatically plots the graphs in different colors so that they can be identified. It is also possible to add

line specifiers following each pair. For example the command

```
plot(x,y,'-b',u,v,'--r',t,h,'g:')
```

plots `y` vs. `x` with a solid blue line, `v` vs.`u` with a dashed red line, and `h` vs. `t` with a dotted green line.

Sample Problem 5-1: Plotting a function and its derivatives

Plot the function $y = 3x^3 - 26x + 10$, and its first and second derivatives, for $-2 \le x \le 4$, all in the same plot.

Solution

The first derivative of the function is: $y' = 9x^2 - 26$.
The second derivative of the function is: $y'' = 18x$.
A script file that creates a vector x and calculates the values of y, y', and y'' is:

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

Create vector `x` with the domain of the function.

Create vector `y` with the function value at each `x`.

Create vector `yd` with values of the first derivative.

Create vector `ydd` with values of the second derivative.

Create three graphs, `y` vs. `x`, `yd` vs. `x`, and `ydd` vs. `x`, in the same figure.

The plot that is created is shown in Figure 5-7.

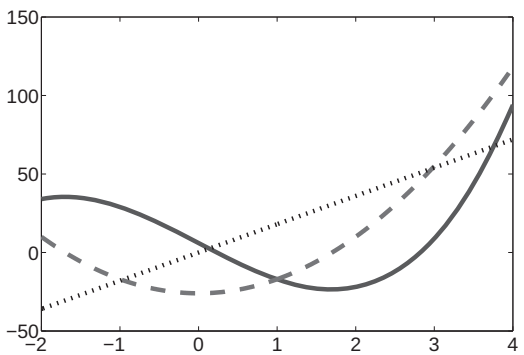

Figure 5-7: A plot of the function $y = 3x^3 - 26x + 10$ and its first and second derivatives.

5.3.2 Using the `hold on` *and* `hold off` *Commands*

To plot several graphs using the `hold on` and `hold off` commands, one graph is plotted first with the `plot` command. Then the `hold on` command is typed. This keeps the Figure Window with the first plot open, including the axis

properties and formatting (see Section 5.4) if any was done. Additional graphs can be added with `plot` commands that are typed next. Each `plot` command creates a graph that is added to that figure. The `hold off` command stops this process. It returns MATLAB to the default mode, in which the `plot` command erases the previous plot and resets the axis properties.

As an example, a solution of Sample Problem 5-1 using the `hold on` and `hold off` commands is 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,'-b')
hold on
plot(x,yd,'--r')
plot(x,ydd,':k')
hold off
```

The first graph is created.

Two more graphs are added to the figure.

5.3.3 *Using the* `line` *Command*

With the `line` command additional graphs (lines) can be added to a plot that already exists. The form of the line command is:

```
line(x,y,'PropertyName',PropertyValue)
```

(Optional) Properties with values that can be used to specify the line style, color, and width, marker type, size, and edge and fill colors.

The format of the `line` command is almost the same as the `plot` command (see Section 5.1). The `line` command does not have the line specifiers, but the line style, color, and marker can be specified with the Property Name and property value features. The properties are optional, and if none are entered MATLAB uses default properties and values. For example, the command:

```
line(x,y,'linestyle','--','color','r','marker','o')
```

will add a dashed red line with circular markers to a plot that already exists.

The major difference between the `plot` and `line` commands is that the `plot` command starts a new plot every time it is executed, while the `line` command adds lines to a plot that already exists. To make a plot that has several graphs, a plot command is typed first and then line commands are typed for additional graphs. (If a line command is entered before a plot command, an error message is displayed.)