A. The `plot` command

The basic MATLAB graphing command is the `plot` command. You follow a 3-step procedure:

1. generate some x values;
2. generate corresponding y values; and
3. plot the points.

Here is an example. (Remember, you do not have to type in the comments.)

```
>>x = linspace(0, 2*pi, 100);
```

```
>>y = sin(x);
```

```
>>plot(x, y)  %This plots the (x, y) points connected with a straight line
```

To plot more than one graph on the same set of axes, you can try something like this:

```
>>z = cos(x);
```

```
>>plot(x, y, '-', x, z, ':')   %This plots the (x, y) points connected with a solid line
>>                     % ('-') and the (x, z) points connected with a dotted line (':')
```

You can add a title and axis labels and you can label your curves using the following commands. DO NOT close the figure window containing the plot.

```
>>title( 'Graphs of sine and cosine functions' ) %Creates a title at the top of the graph
```

```
>>xlabel('x')  %Puts a label under the horizontal axis
```

```
>>ylabel('y')  %Puts a label next to the vertical axis
```

```
>>legend('y = sin(x)', 'y = cos(x)')  %Creates a legend indicating which graph is which
```

If you don't like where MATLAB places the legend box, you can use the mouse to drag the box wherever you want it.

In addition to solid lines and dotted lines, you can generate dashdot graphs (`'.-'`) or dashed graphs (`'--'`). **You can control the color of the graph** by adding a letter after the line type (**b**lue, **g**reen, **r**ed, **c**yan, **m**agenta, **y**ellow, or blac**k**). For example, to plot the sine graph in red using a dashdot line and the cosine graph in black using a dashed line, type

```
plot(x, y, '.-r', x, z, '--k')
```

MATLAB has a trace feature. If you click on the Data Cursor toolbar icon in the figure window (the icon that has a + sign in the lower left corner) and then click on a graph, you will see the coordinates of the point you clicked on. You can use the arrow keys to move the data cursor to other points on the graph.

B. Plotting curves described by parametric equations

You can use the `plot` command to plot curves described by parametric equations. For example, to plot the curve given by $x = \cos(t)$, $y = \sin(t)$ for $0 \le t \le 2\pi$ try the following commands.

```
>>t = linspace(0, 2*pi, 100);
```

```
>>x = cos(t);
```

```
>>y = sin(t);
```

```
>>plot(x,y)
```

C. The `subplot` command

The *subplot* command allows you to create a single figure containing multiple plot windows. The command *subplot(m,n,p)* opens a figure with an $m \times n$ array of windows, and it activates window number p. (Windows are numbered left to right, starting in the first row, then moving to the second row, etc.) You can then issue commands to generate a plot in window number p.

Try the following commands. Do not close the figure until after you enter the last command.

```
>>subplot(2,2,1)
>>x=linspace(-10,10,100);
>>y=2*x;
>>plot(x,y)
>>subplot(2,2,2)
>>x=linspace(-3,3,100);
>>y=x.^2;
>>plot(x,y)
>>subplot(2,2,3)
>>x=linspace(-2,2,100);
>>y=x.^3;
>>plot(x,y)
>>subplot(2,2,4)
>>y=x.^4;
>>plot(x,y)
```

D. The `plotyy` command

Sometimes you might need to plot two quantities with different units and different magnitudes on the same set of axes. For example, suppose you wanted to plot both temperature (in °C) and atmospheric pressure (in millibars) at Logan Airport as functions of time for one day. Try the following commands. Do not close the figure until after you enter the last command.

```
>>t = linspace(0,24,100);
>>T = 20-5*cos(2*pi*t/24);
>>p = 1000+20*(t/24).^2;
>>[AX] = plotyy(t,T,t,p);
>>set(get(AX(1),'Ylabel'),'String','Temperature ( ^\circC)')
>>set(get(AX(2),'Ylabel'),'String','Pressure (mbars)')
>>xlabel('Time (hours)')
>>title('Temperature and Air Pressure at Logan Airport, Midnight 2/30/99 - Midnight 2/31/99')
```

E. Logarithmic axis scales

MATLAB can generate graphs in which one or both axes are scaled logarithmically. This is useful if your domain and/or range extends over several orders of magnitude.

For example, to draw a loglog plot of $y = x^2$ over the interval $10^{-3} \le x \le 10^4$, try these commands.

```
>>x = logspace(-3,4,100); %If the horizontal axis is scaled logarithmically
>> %it is better to use logspace than linspace
>>y = x.^2;
>>loglog(x,y)
```

To draw a graph in which the horizontal axis is scaled logarithmically and the vertical axis is scaled linearly, use the command `semilogx`, and to draw a graph in which the horizontal axis is scaled linearly and the vertical axis is scaled logarithmically, use the command `semilogy`.

For example, try these commands.

```
>>x = logspace(-2,5,100);
>>y = log(x);
>>semilogx(x,y)
>>x = linspace(-3,3,100);
>>y = exp(x);
>>semilogy(x,y)
```

Practice Problems. (Some are taken from Gilat, *MATLAB: An Introduction with Applications*)

1. Plot the function $f(x) = \dfrac{(x+5)^2}{4+3x^2}$ for $-3 \le x \le 5$.

2. Plot the curve described by the parametric equations
   $x = 1.5\sin(5t), \ y = 1.5\cos(3t)$ for $0 \le t \le 2\pi$. Use the command `>>axis([-2 2 -2 2])` to format the plot so that both axes range from -2 to 2.

3. Plot the function $f(x) = \cos(x)\sin(2x)$ and its derivative, both on the same plot, for $-\pi \le x \le \pi$. Plot the function with a solid line and the derivative with a dashed line. Add a legend and label the axes x and y.

4. Draw a loglog plot of the function $f(x) = \dfrac{x}{\sqrt{x^2+1}}$ for $10^{-2} \le x \le 10^3$

5. Suppose the height of a falling object (in meters) is given by $y = 490 - 4.9t^2$ where t denotes time (in seconds). Use *plotyy* to plot the height and velocity ($v = -9.8t$) of the object for $0 \le t \le 10$. Label all axes.