Introduction to Engineering
MATLAB – 8
Plotting

Agenda

- Multiple curves
- Multiple plot

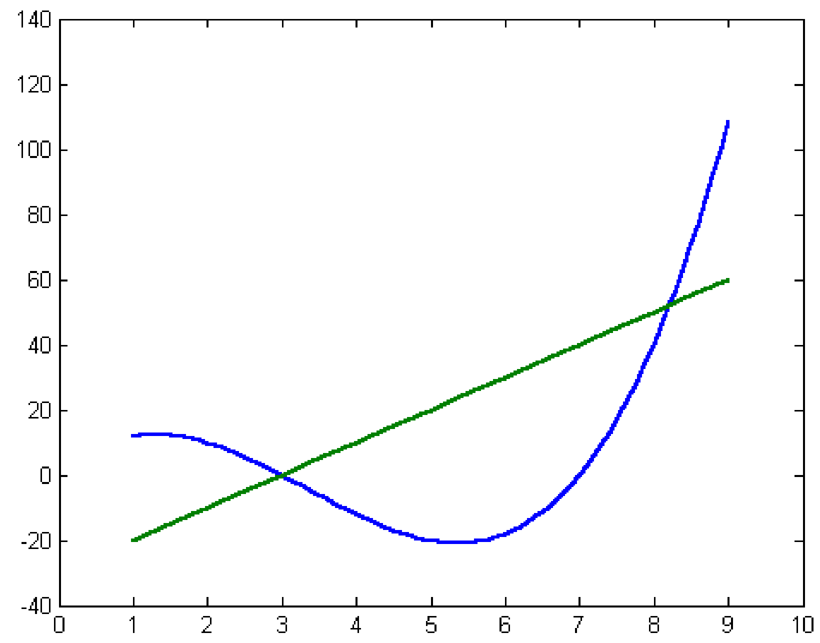
- Note: when working through this lesson on your own, type each example step by step in the command window to learn the effects of each individual step. Then create the script files shown for each example to see how it's run all at once.

PLOTTING MULTIPLE CURVES IN THE SAME PLOT

There are two methods for plotting two curves in one plot:

1. Using the plot command.

2. Using the hold on, hold off commands.



USING THE PLOT COMMAND TO PLOT MULTIPLE CURVES

The command:

plot(x,y,u,v)   plots y versus x and v versus u on the same plot.

By default, the computer makes the curves in different colors.

The curves can have a specific style by using:

plot(x,y,'color_linestyle_marker',u,v, 'color_linestyle_marker')

More curves can be added.

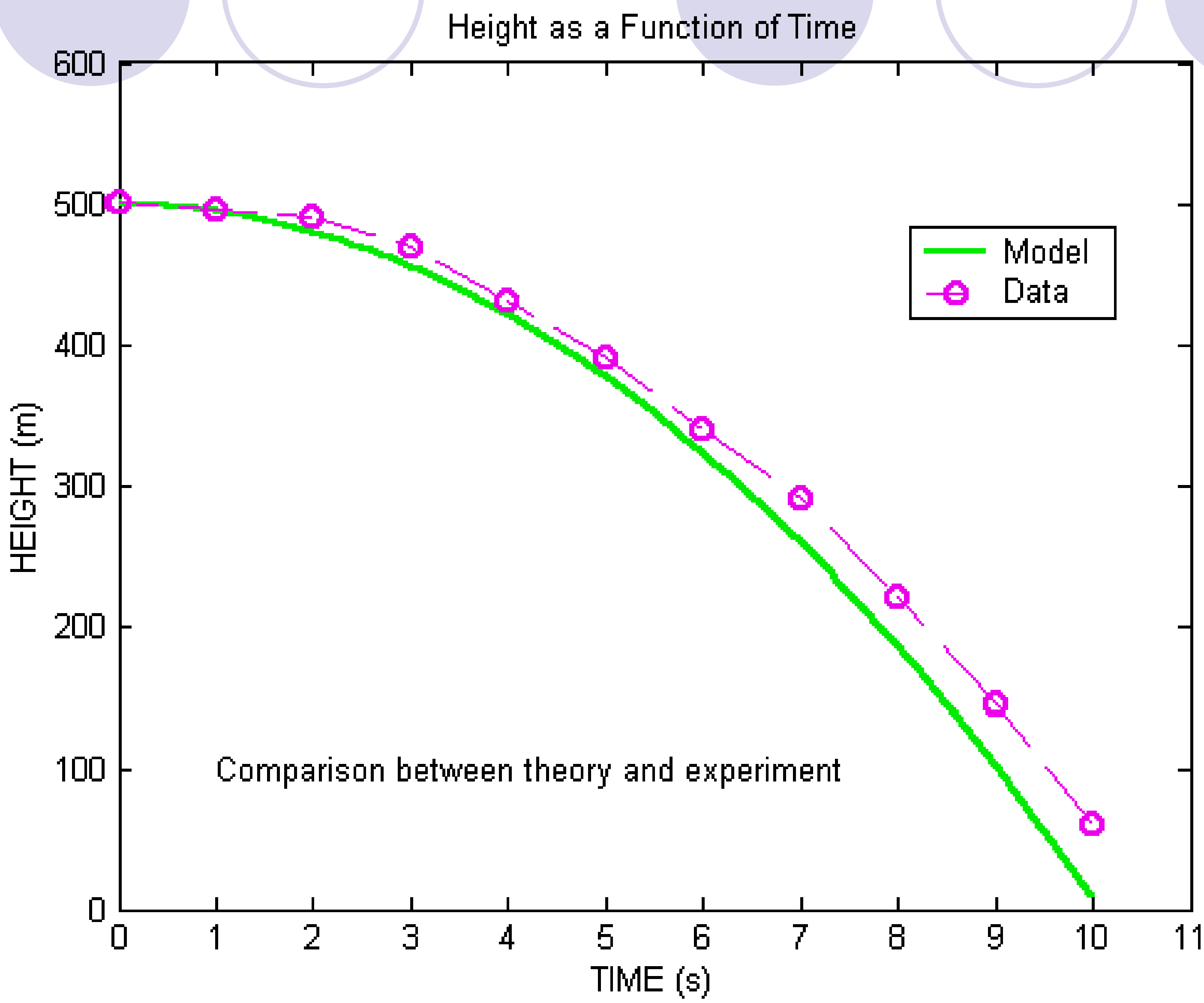
EXAMPLE OF A FORMATTED PLOT WITH TWO CURVES

Below is the script file of a falling object.

```
x=[0:0.1:10];
y=500-0.5*9.81*x.^2;
xd=[0:10];
yd=[500 495 490 470 430 390 340 290 220 145 60];
plot(x,y,'g',xd,yd,'mo--')
xlabel('TIME (s)')
ylabel('HEIGHT (m)')
title('Height as a Function of Time')
legend('Model','Data')
axis([0 11 0 600])
text(1,100,'Comparison between theory and experiment')
```

- Creating a vector of time x
- Calculated height y for each x
- Creating a vector of time (data) xd
- Creating a vector of height (data) yd
- Plot y versus x in green, and plot yd versus xd in magenta, circle markers, and dashed line.

A PLOT WITH TWO CURVES



USING THE hold on, hold off, COMMANDS TO PLOT MULTIPLE CURVES

hold on Holds the current plot and all axis properties so that subsequent plot commands add to the existing plot.

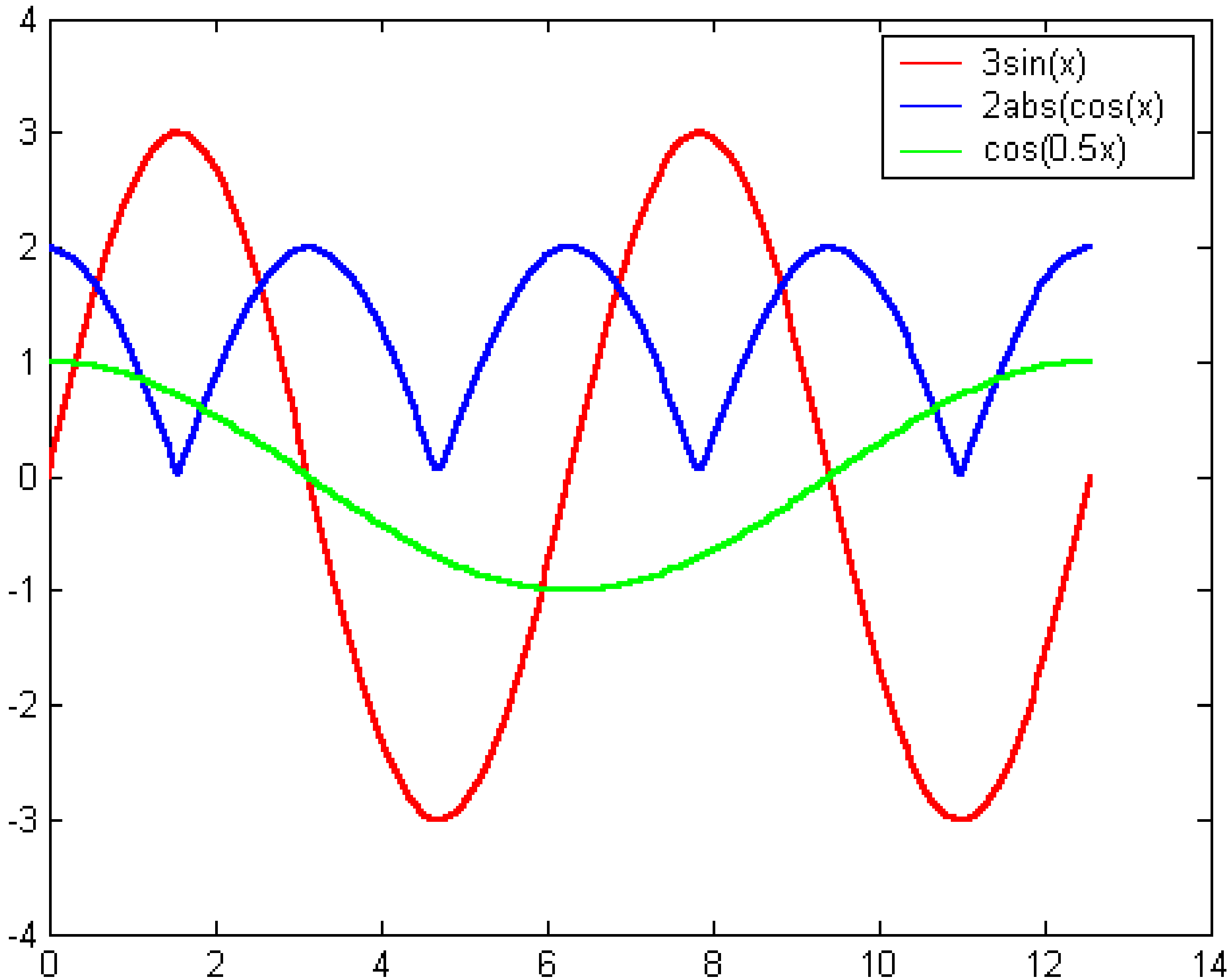
hold off Returns to the default mode whereby plot commands erase the previous plots and reset all axis properties before drawing new plots.

This method is useful when all the information (vectors) used for the plotting is not available at the same time.

EXAMPLE OF USING THE hold on, hold off, COMMAND

```
x=linspace(0,4*pi,200);
y1=3*sin(x);
plot(x,y1,'r')                  <- Plotting the first curve
hold on                         <- hold on command
y2=2*abs(cos(x));
plot(x,y2,'b')                  <- Plotting the second curve
y3=cos(0.5*x);
plot(x,y3,'g')                  <- Plotting the third curve
axis([0 14 -4 4])
legend('3sin(x)','2abs(cos(x)','cos(0.5x)')
hold off                        <- hold off command
```

The plot created by the script file above is shown in the next slide.


3sin(x)
2abs(cos(x)
cos(0.5x)
4
3
2
1
0
-1
-2
-3
-4
0
2
4
6
8
10
12
14

PLOTTING MULTIPLE PLOTS ON ONE PAGE

Several plots on one page can be created with the subplot command.

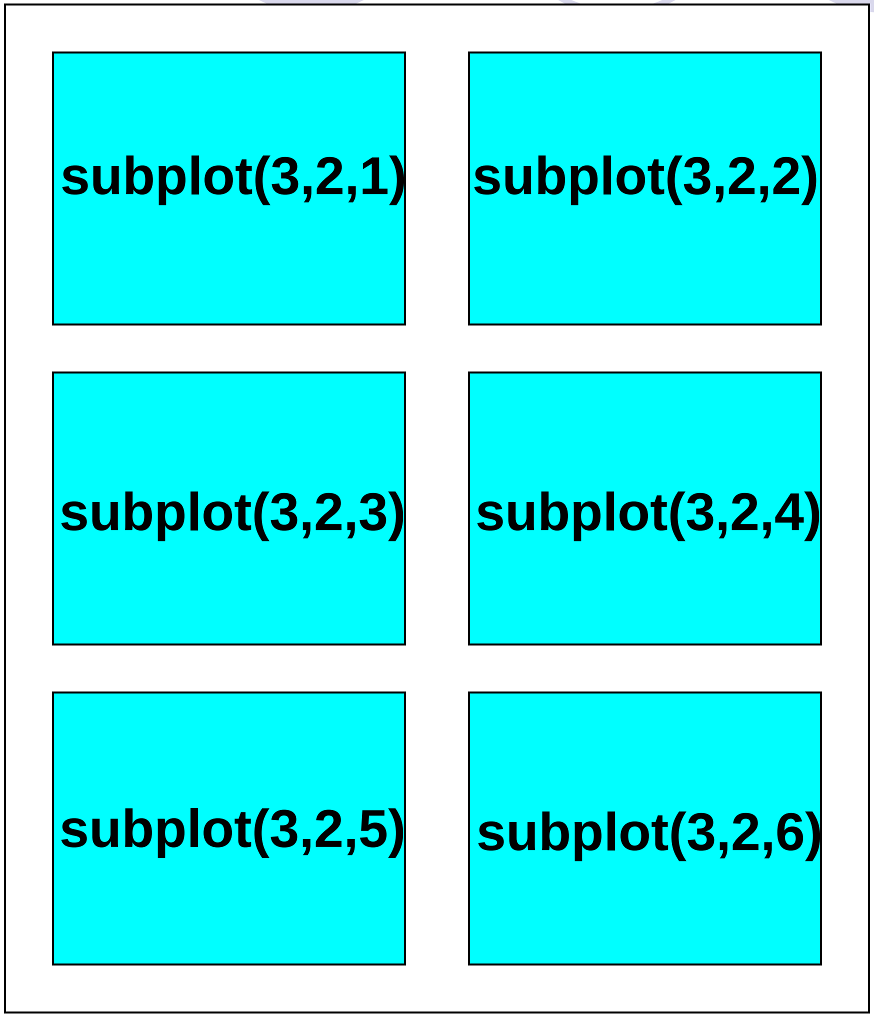
subplot(*m*,*n*,*p*) This command creates *m*xn plots in the Figure Window. The plots are arranged in *m* rows and *n* columns. The variable *p* defines which plot is active. The plots are numbered from 1 to *m*x*n*. The upper left plot is 1 and the lower right plot is *m*x*n*. The numbers increase from left to right within a row, from the first row to the last.

PLOTTING MULTIPLE PLOTS ON ONE PAGE

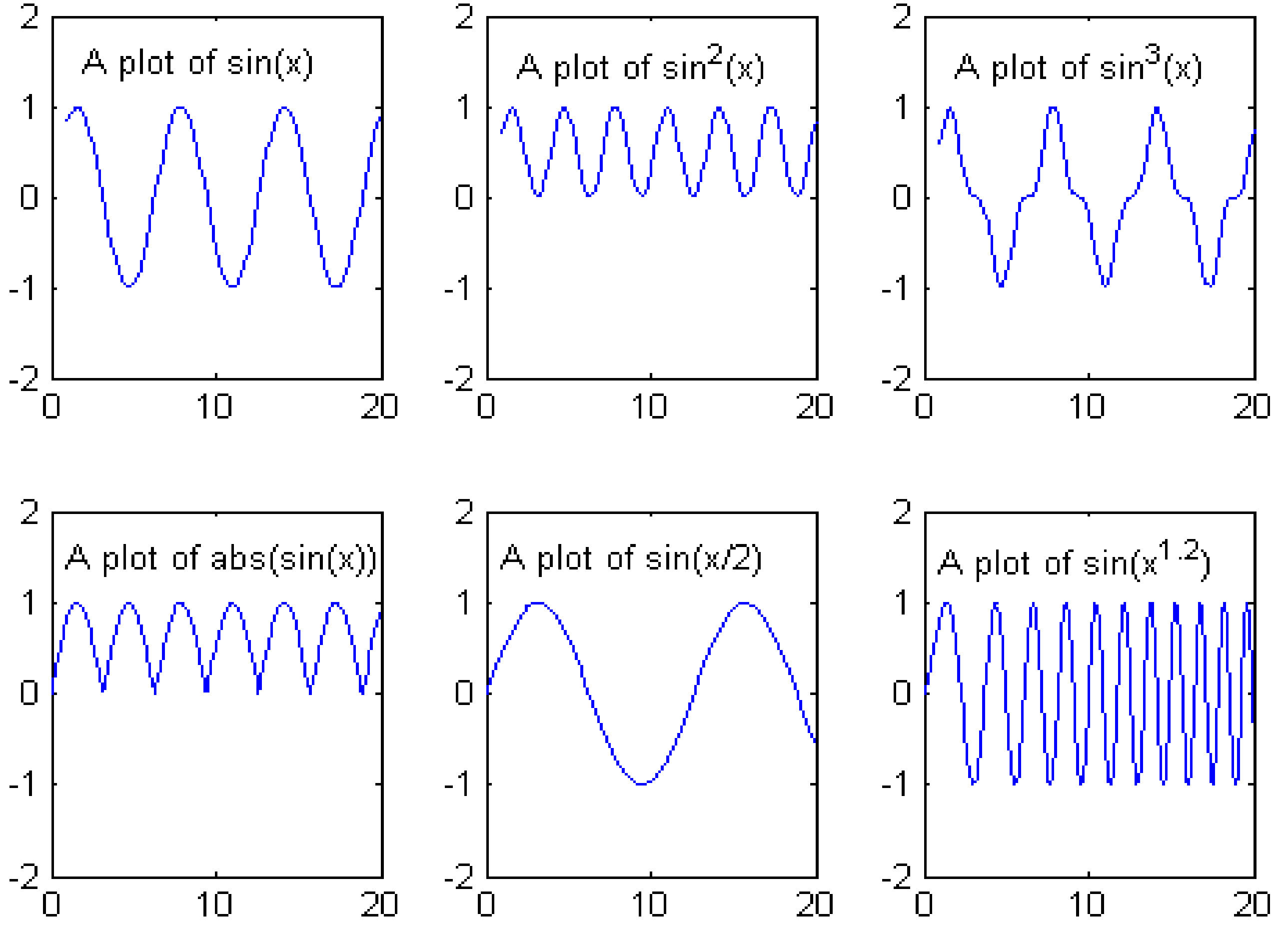
For example, the command:

subplot(3,2,*p*)

Creates 6 plots arranged in 3 rows and 2 columns. Changing the value of p activates a particular subplot.



EXAMPLE OF MULTIPLE PLOTS ON ONE PAGE



The script file of the figure above is shown in the next slide.

SCRIPT FILE OF MULTIPLE PLOTS ON ONE PAGE

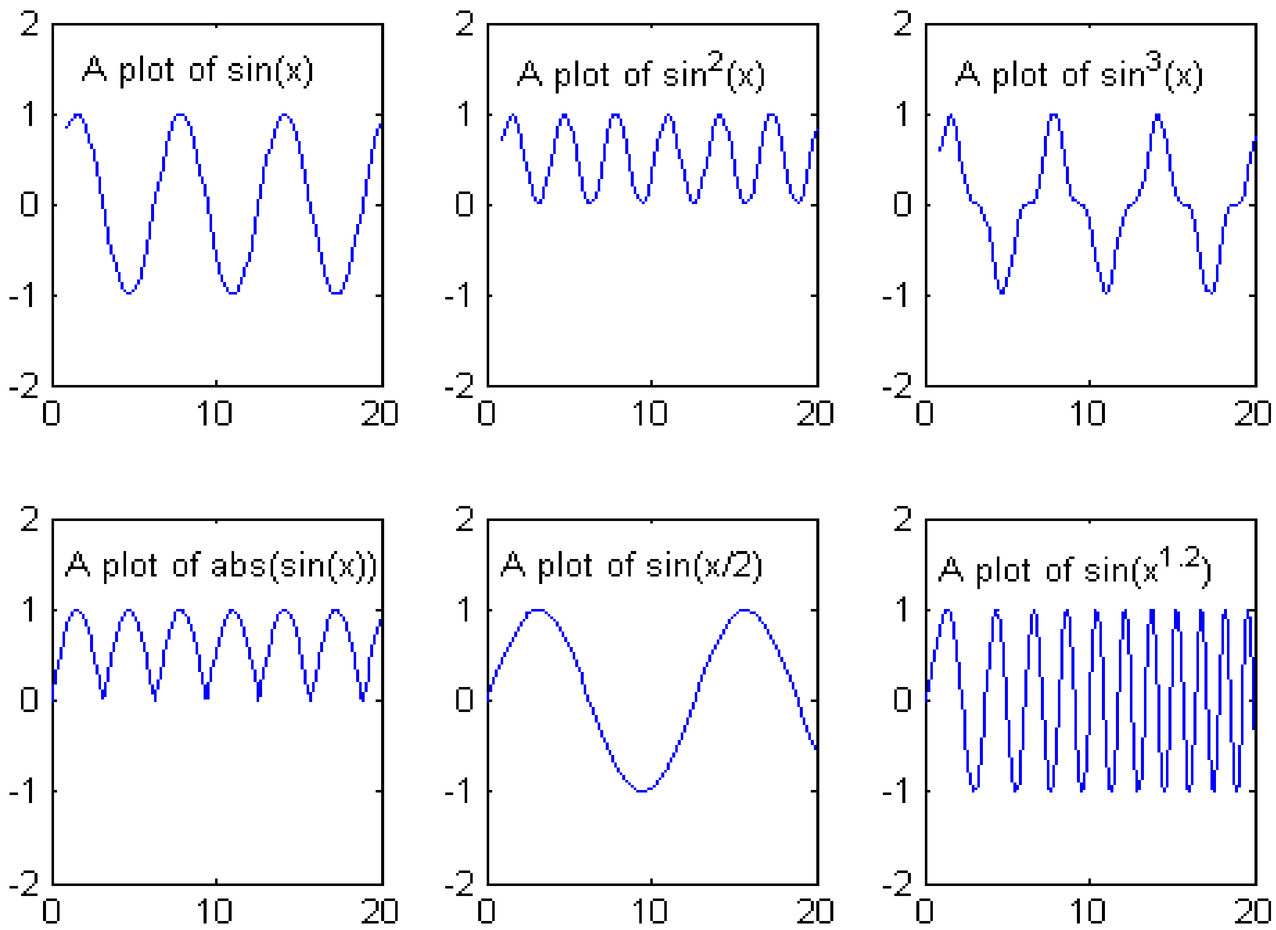
```
% Example of using the subplot command.
x1=linspace(1,20,100);          % Creating a vector x1
y1=sin(x1);                     % Calculating y1
subplot(2,3,1)                  % Creating the first plot
plot(x1,y1)                     % Plotting the first plot
axis([0 20 -2 2])               % Formatting the first plot
text(2,1.5,'A plot of sin(x)')
y2=sin(x1).^2;                  % Calculating y2
subplot(2,3,2)                  % Creating the second plot
plot(x1,y2)                     % Plotting the second plot
axis([0 20 -2 2])               % Formatting the second plot
text(2,1.5,'A plot of sin^2(x)')
y3=sin(x1).^3;                  % Calculating y2
```

(The file continues on the next slide)

SCRIPT FILE OF MULTIPLE PLOTS ON ONE PAGE (CONT.)

```
subplot(2,3,3)                        % Creating the third plot
plot(x1,y3)                           % Plotting the third plot
axis([0 20 -2 2])                     % Formatting the third plot
text(2,1.5,'A plot of sin^3(x)')
subplot(2,3,4)                        % Creating the fourth plot
fplot('abs(sin(x))',[0 20 -2 2])      % Plotting the fourth plot
text(1,1.5,'A plot of abs(sin(x))')   % Formatting the fourth plot
subplot(2,3,5)                        % Creating the fifth plot
fplot('sin(x/2)',[0 20 -2 2])         % Plotting the fifth plot
text(1,1.5,'A plot of sin(x/2)')      % Formatting the fifth plot
subplot(2,3,6)                        % Creating the sixth plot
fplot('sin(x.^1.4)',[0 20 -2 2])      % Plotting the fifth plot
text(1,1.5,'A plot of sin(x^1^.^2)')  % Formatting the sixth plot
```

EXAMPLE OF MULTIPLE PLOTS ON ONE PAGE
(CONT.)



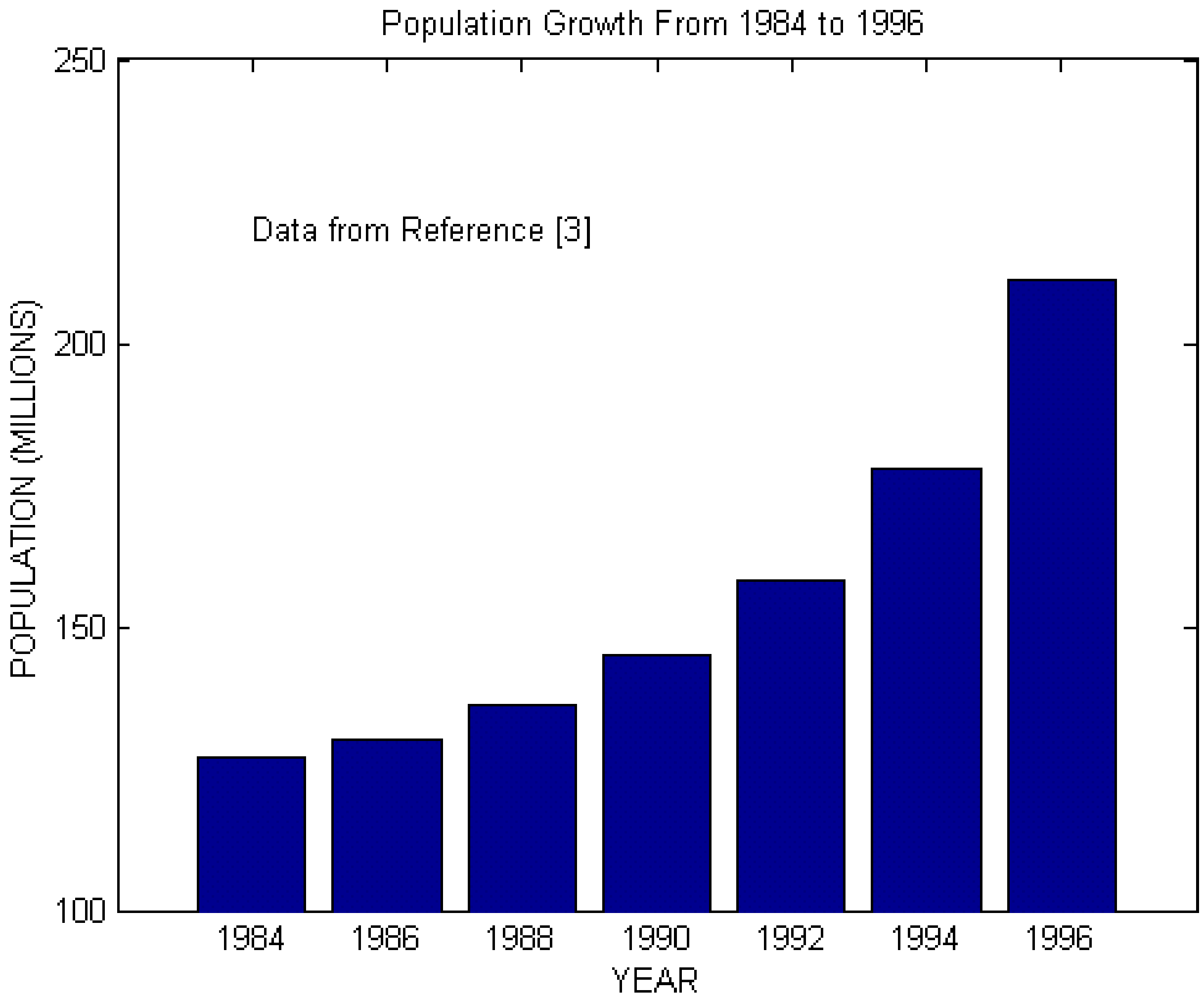
PLOTS WITH SPECIAL GRAPHICS

Commands for plots with special geometry:

bar(x,y)	Creates a bar chart of *y* vs. *x*.
stairs(x,y)	Creates a stair chart of *y* vs. *x*. (example follows)
stem(x,y)	Creates a stem chart of *y* vs. *x*. (example follows)
polar(theta,r)	Creates a polar plot. The vectors theta and r contain the polar coordinates θ and *r*, respectively.

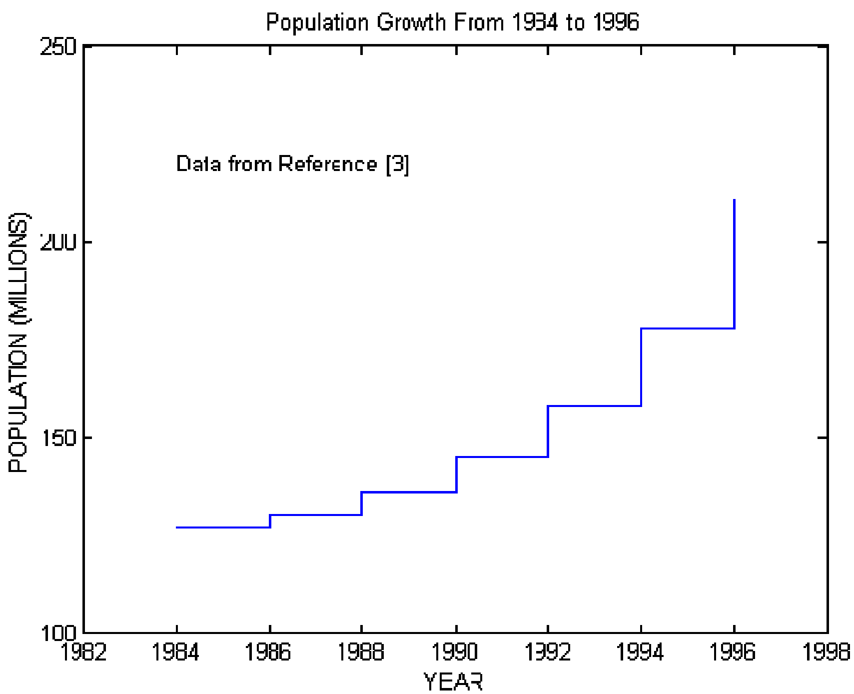
EXAMPLE OF PLOTS WITH SPECIAL GRAPHICS

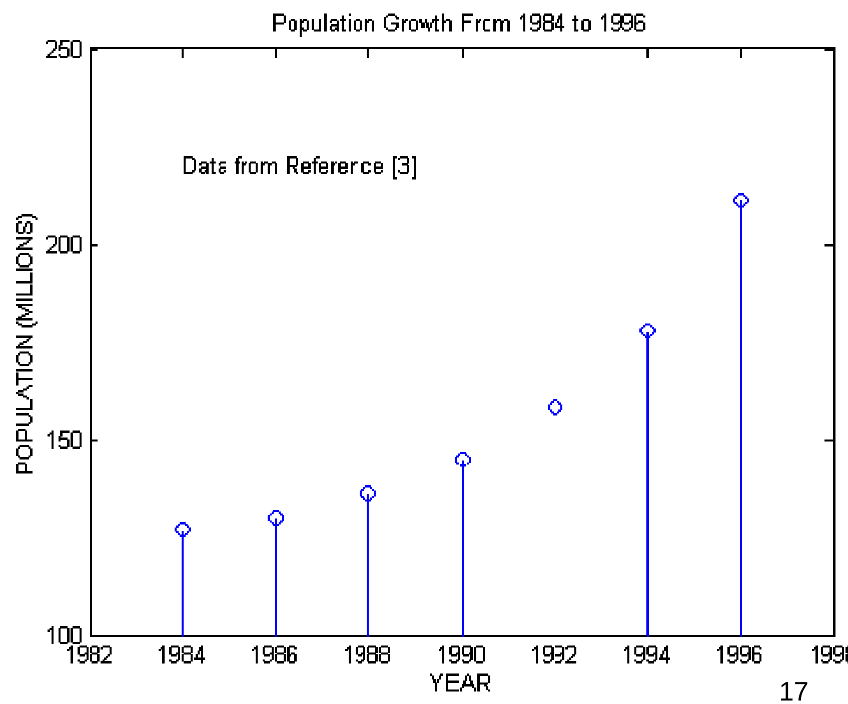
Bar plot of the population growth data.



EXAMPLE OF PLOTS WITH SPECIAL GRAPHICS

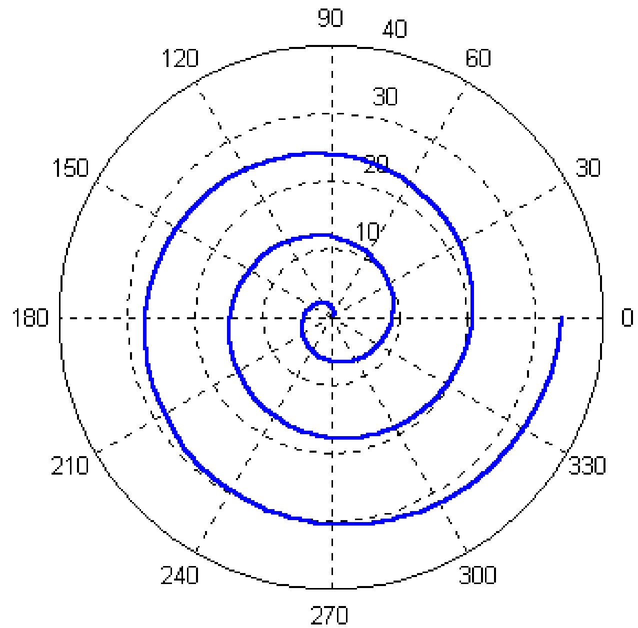
Stairs and stem plots of the population growth data.





EXAMPLE OF A POLAR PLOT

A polar plot of the function: $r = \theta^{1.2}$




```
>> theta=linspace(0,6*pi,150);
>> r=theta.^1.2;
>> polar(theta,r)
```

THE fplot COMMAND

The command:

fplot('the function as a string',[xmin xmax])

plots the function between the x-axis limits specified by [xmin,xmax].

The command can also include the limits of the y-axis and specification of the line (color,type, etc.).

Example:

```
fplot('cos(8*x+2)/exp(-0.8*x)',[0 3 -10 15],'r')
```

The function:

$$y = \frac{\cos(8x + 2)}{\exp(-0.8x)}$$

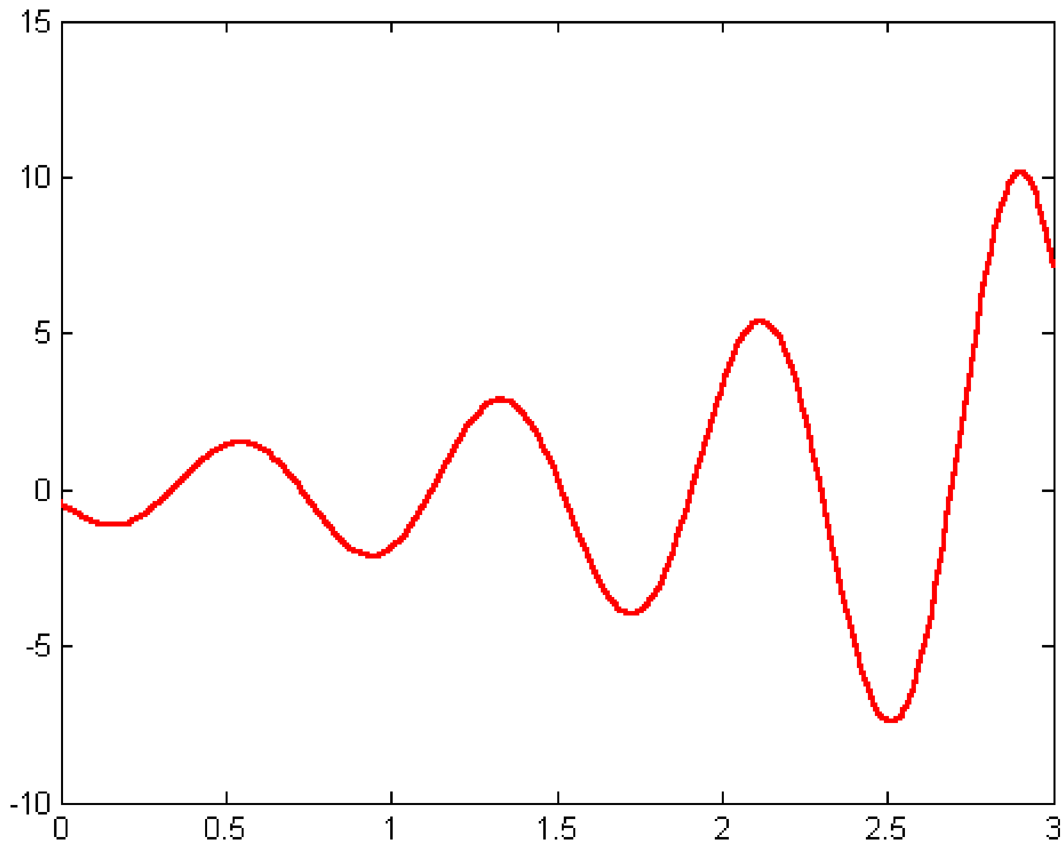
Range of the independent variable *x*

Limits of The *y* axis

Plot with red line

THE fplot COMMAND

The plot created by fplot('cos(8*x+2)/exp(-0.8*x)',[0 3 -10 15],'r')



LOGARITHMIC PLOTS

Plot commands for plots with logarithmic scales.

loglog(x,y)	Creates a *y* vs. *x* plot with log scale on both axes.
semilogx(x,y)	Creates a *y* vs. *x* plot with log scale on the *x* axis and linear scale on the *y* axis.
semilogy(x,y)	Creates a *y* vs. *x* plot with log scale on the *y* axis. and linear scale on the *x* axis.

IMPORTANT FACTS ABOUT LOGARITHMIC PLOTS

1. Negative numbers can not be plotted on log scales (since log of a negative number is not defined).

2. The number zero can not be plotted on a log scale.

3. The tick-mark labels on a log scale are the actual values being plotted (they are not the logarithms of the numbers).

4. Equal distances on a log scale correspond to multiplication by the same constant (in linear scale equal distances correspond to addition of the same constant).

5. Tick marks are not evenly spaced.

PLOT OF THE FUNCTION $y = 8^{(1-x)}$ WITH LINEAR AND LOG SCALES