MATLAB TRAINING SESSION II

DATA PRESENTATION

MATLAB provides a variety of functions for displaying data as 2-D and 3-D plots of curves and 3-D mesh surface plots, as well as functions for annotating these graphs. On-line help for each of the commands shown below is available by typing `help 'command_name'` or `doc 'command_name'` at the MATLAB prompt.

2-D PLOTS:

The following list summarizes the functions that produce basic line plots of data. These functions differ only in the way they shape and scale the plot axes. Each accepts arguments in the form of vectors or matrices and automatically scales the axis to accommodate the data range.

- `plot(x,y)` Generates a linear plot of **x** versus **y**
- `errorbar(x,y,e)` Generates a linear plot of **x** versus **y** with with error bars that are symmetric about y and are 2*e(i) long
- `semilogx(x,y)` Generates a plot of **x** versus **y** using a logarithmic scale for x
- `semilogy(x,y)` Generates a plot of **x** versus **y** using a logarithmic scale for y
- `loglog(x,y)` Generates a plot of **x** vs. **y** using a logarithmic scale for x and y
- `polar(theta,r)` Generates a polar plot of angles **theta**(in rads) versus magnitudes **r**
- `bar(x,y)` Generates a bar graph of **y** at locations specified by **x**
- `hist(y,nb)` Generates a histogram of data in vector **y** in **nb** number of bins
- `stairs(x,y)` Generates a stair graph of **y** at locations specified by equally spaced **x**
- `stem(x,y)` Generates a discrete impulse plot of **y** at locations specified by **x**
- `rose(theta,nb)` Generates a angle histogram for angles in **theta** in **nb** number of bins
- `compass(z)` Generates a plot that displays the angle and magnitude of the complex elements of **z** as arrows emanating from the origin
- `quiver(x,y,dx,dy)` Generates a plot that displays little arrows at every (**x**,**y**) pair where **dx** and **dy** determine the direction and magnitude of the arrows
- `subplot(n,m,p)` Splits the graphics window into a **n**-by-*m* matrix of plots where **p** will be next window used for a plot command.

PLOTTING OPTIONS

- `'plot_command'(x,y,w,z)` Generates two plots on the same axes
- `'plot_command'(x,y,'indicator')` Establishes line or marker style and color

INDICATORS:

line type	indicator	point type	indicator	color	indicator
solid	-	point	.	red	r
dashed	--	plus	+	yellow	y
dotted	:	star	*	green	g
dashdot	-.	circle	O	blue	b
		x-mark	x	magenta	m
				cyan	c
				white	w
				black	k

SCALING

- `axis(axis)` Freezes the current axis scaling for subsequent plots
- `axis(v)` **v** is a 4-element vector containing [xmin, xmax, ymin, ymax]
- `axis square` Specifies the aspect ratio to be square
- `axis auto` Specifies the aspect ratio to return to the default
- `hold` Freezes the current axis scaling and plot for subsequent plots to share (another way to make multiple plots on the same graph)
  The second call to hold releases the current axis

ANNOTATION

The following list summarizes the commands for adding annotation to plots.

- `title('text')` Writes the text string as a title at the top of the current plot
- `xlabel('text')` Writes the text string beneath the x-axis on the current plot
- `ylabel('text')` Writes the text string beside the y-axis on the current plot
- `legend(str1,...)` Puts a legend on the current plot using the specified strings as labels
- `text('text')` Writes the text string at the point specified by (x,y) on the current plot
- `gtext('text')` Writes the text string at the point specified by a mouse click
- `grid` Adds grid lines to the current plot

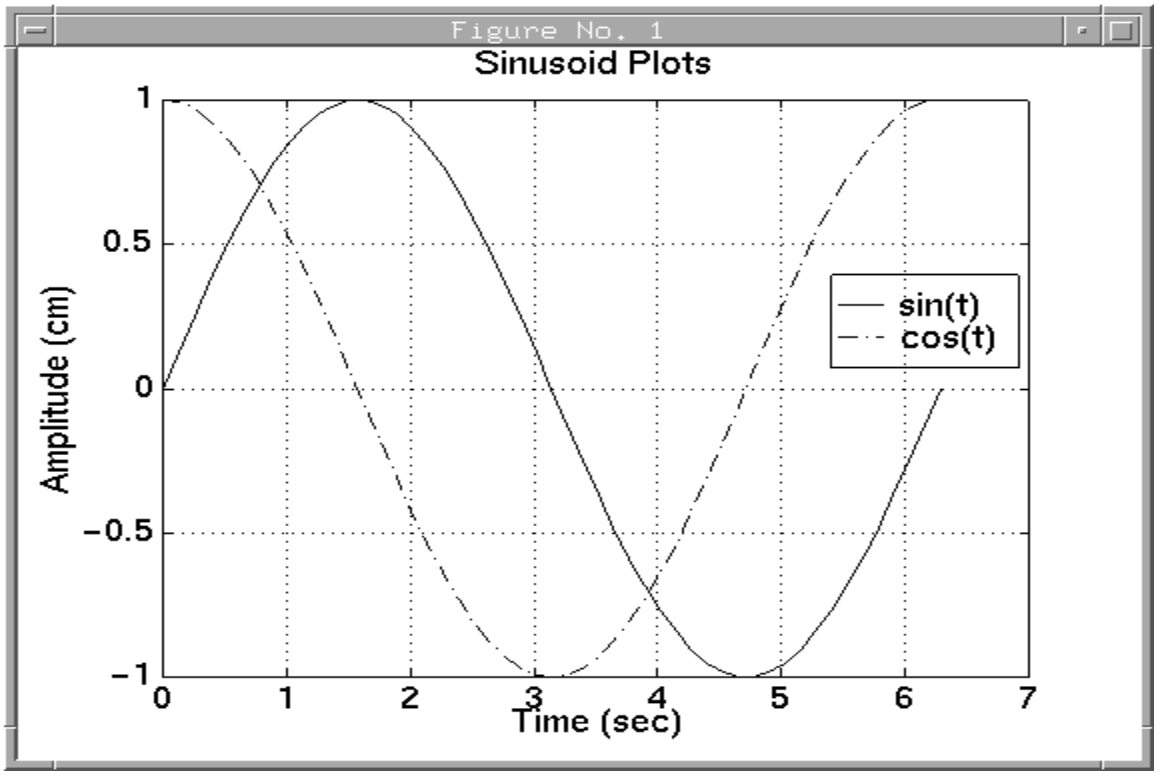
SCREEN CONTROL

- `shg` Show graph window
- `clg` Clear graph window
- `figure` Create a graph window
- `delete` Delete a graph window
- `axes` Control the graph axis properties
- `gcf` Get current figure handle
- `gca` Get current axis handle
- `set` Change object property values
- `get` Get object property values

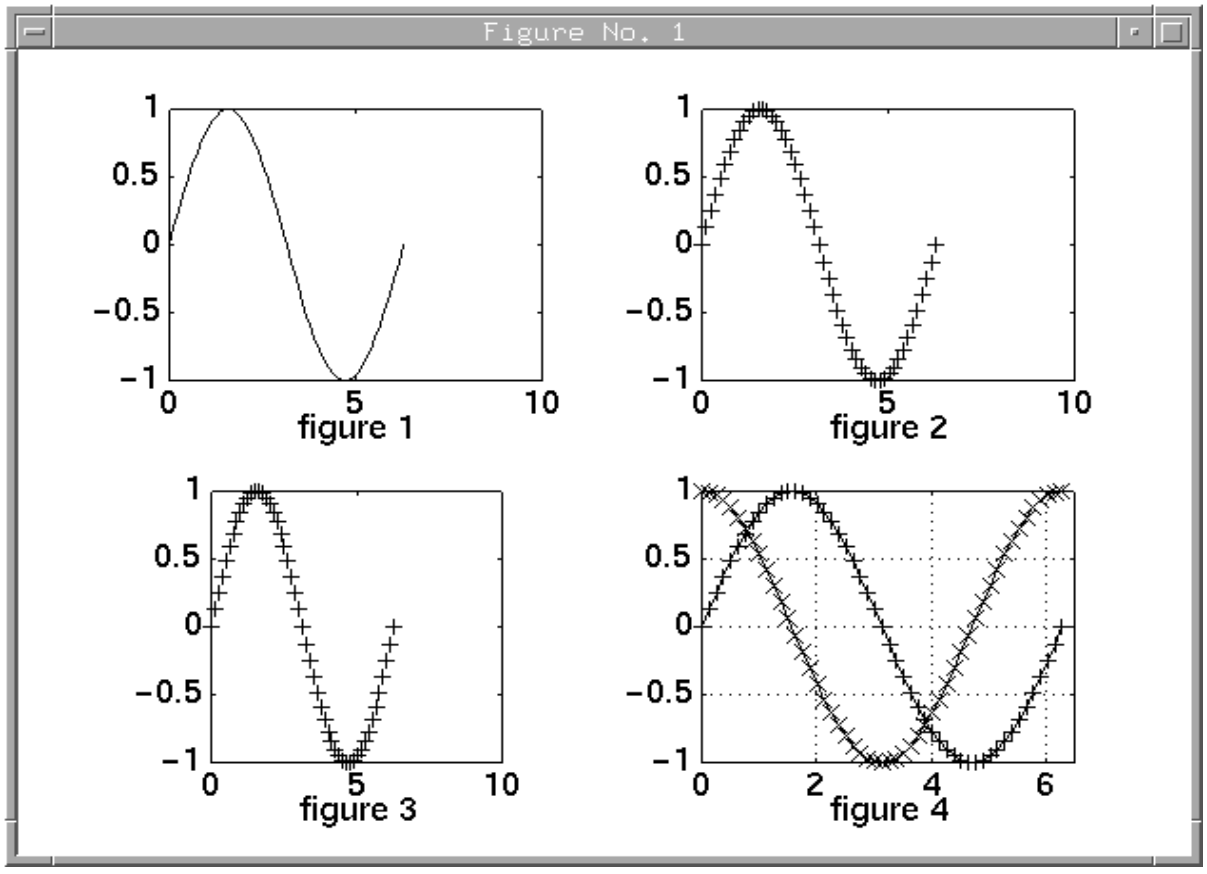
2-D EXAMPLES:

MATLAB_window

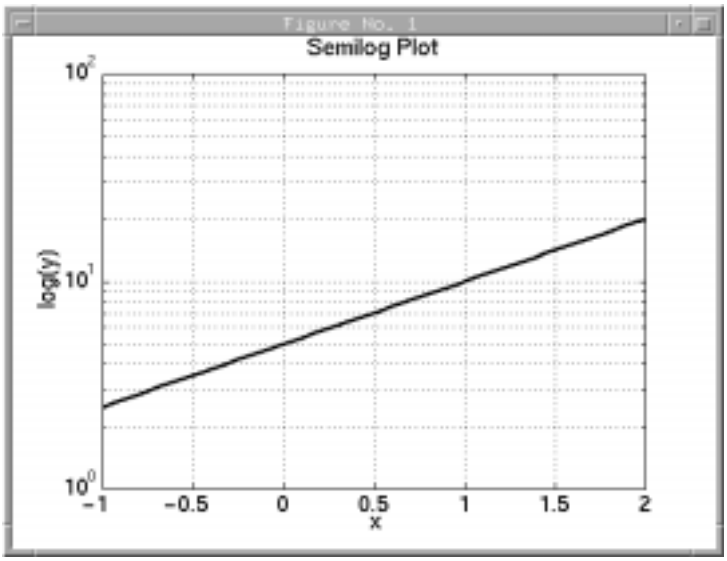
```
                                < M A T L A B (R) >
                    (c) Copyright 1984-94 The MathWorks, Inc.
                                All Rights Reserved
                                   Version 4.2a
                                    May 13 1994


MATLAB passcode expiration date of 01-jun-1995 is less than
Commands to get started: intro, demo, help help
Commands for more information: help, whatsnew, info, subscri

>> t = 0:pi/25:2*pi;
>> x = sin(t);
>> y = cos(t);
>> plot(t,x)
>> hold
Current plot held
>> plot(t,y,'-.')
>> title(' Sinusoid Plots ')
>> xlabel(' Time (sec) ')
>> ylabel(' Amplitude (cm) ')
>> legend(' sin(t) ',' cos(t) ')
>> grid, hold
Current plot released
>>
```



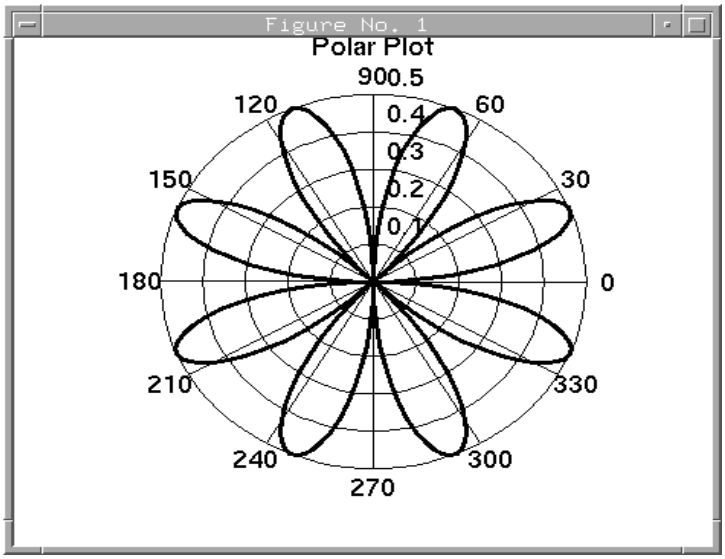
```
MATLAB_window
>>
>> clg
>> subplot(2,2,1), plot(t,x), xlabel('figure1')
>> subplot(2,2,2), plot(t,x,'+'), xlabel('figure2')
>> subplot(2,2,3), plot(t,x,'r+'), xlabel('figure3'), axis square
>> subplot(2,2,4), plot(t,x,t,x,'+',t,y,t,y,'x')
>> axis([0 2*pi -1 1]), xlabel('figure4'), grid
>>
```



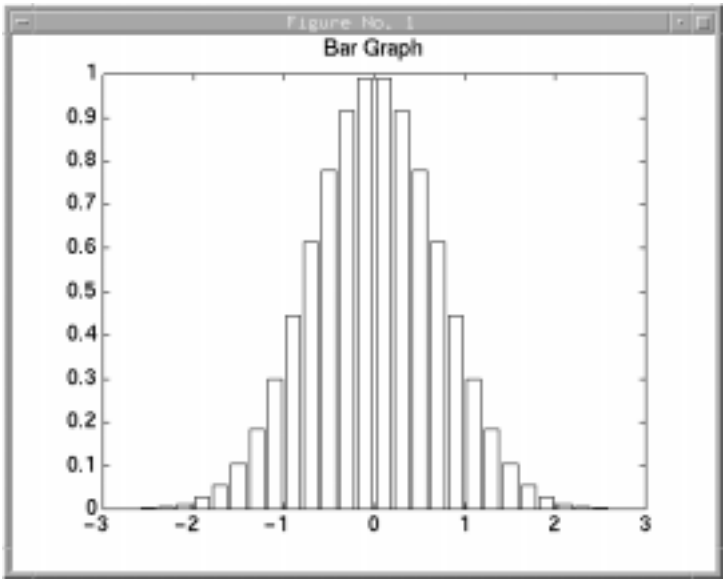
```
MATLAB_window
>>
>> delete(1)
>> x = linspace(-1,2,50);
>> y = 5*2.^x;
>> plot(x,y)
>> semilogy(x,y)
>> grid, title('Semilog Plot')
>> xlabel('x'), ylabel('log(y)')
>>
```




```
MATLAB_window
>>
>> t = linspace(0,2*pi,200);
>> polar(t,sin(2*t).*cos(2*t))
>> title('Polar Plot')
>>
```




```
MATLAB_window
>>
>> z = -2.9:0.2:2.9;
>> bar(z, exp(-z.^2))
>> title('Bar Graph')
>>
```



3-D PLOTS

MATLAB provides a variety of functions to display 3-D data. Some plot lines in 3-D, while others draw wire frame 3-D surfaces. The following list summarizes the basic functions.

- `plot3(x,y,z)` Plots lines and points on a 3-D axis
- `mesh(Z)` 3-D mesh surface plot of heights in matrix **Z**
- `mesh(X,Y,Z)` 3-D mesh surface plot of matrices **Z** vs. [**X,Y]** pairs
- `mesh(X,Y,Z,C)` 3-D mesh surface plot of matrices **Z** vs. [**X,Y]** pairs along with a color specification matrix

*Use [xgrid,ygrid] = meshgrid(xl:Δx:xh, yl:Δy:yh) to generate the sample plane [X,Y].

- `meshc` Same as mesh but is a combonation mesh/contour plot
- `meshc` Same as mesh but is a combonation mesh/contour plot
- `slice(V,sx,sy,sz,n)` Draws slices of the volume matrix **V** with cutting planes at indicies **sx**, **sy**, and **sz** where **V** has **n** rows.
- `cylinder(R,n)` Forms the unit cylinder with radius vector **R** and base resolution **2π/n**
- `sphere(N)` Forms the unit sphere with surface resolution **(N+1)x(N+1)**
- `contour(Z,n)` Contour plot of **Z** with **n** levels
- `contour(Z,v)` Contour plot of **Z** with levels indicated within the vector **v**
- `contour3` Same as contour plot the but contours are plotted with their corresponding height

Also note that the plotting options, scaling, annotation, and screen commands that are described in the 2-D plotting section are also valid and include the height dimension (i.e. like zlabel(‘ ’)).

HARD COPY

diary *filename* -------- diary off

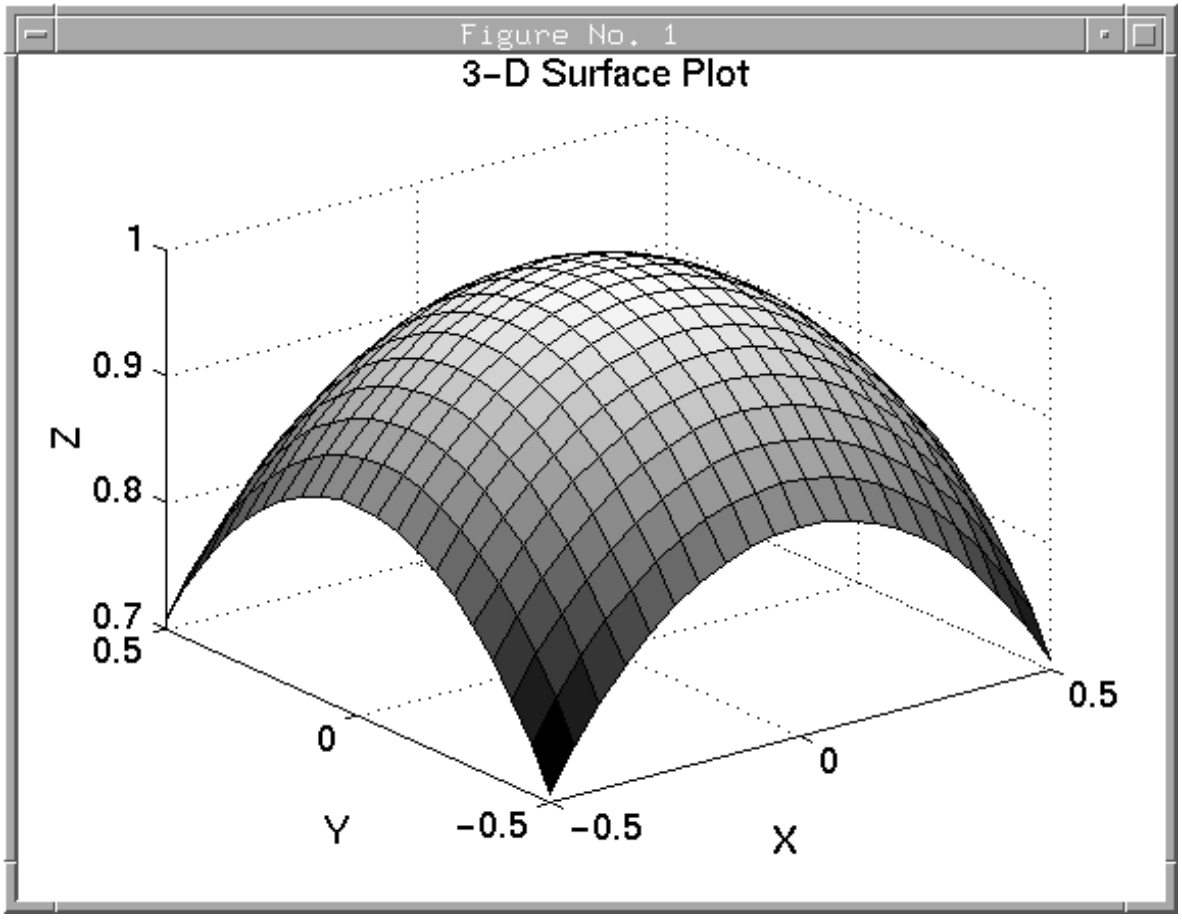
The `diary` command creates a text file version of the matlab session. This file is a log of the commands you gave and the answers MATLAB returned. The diary contains no graphics.

print [-ddevicetype] [-options] *filename*

`print` *`filename`* saves the current figure window to the designated filename in post script device format. If filename does not include an extension, then anappropriate extension like .ps or .eps is appended to it, depending on the device format setting. Use the optional [**-d*devicetype*]** in the print command to change device format of the resulting graphics file such as *`-deps`*. After execution of the command you will find the graphics file in your directory. Use the print command **lp -d(*destination*) *filename*.ps** to send the post script file to your ***destination*** printer such as `-deb119_ps2`.

3-D EXAMPLES:

```
MATLAB_window
>>
>> x = -0.5:0.05:0.5;
>> y = -0.5:0.05:0.5;
>> [X,Y] = meshgrid(x,y);
>> Z = sqrt(1 - X.^2 - Y.^2);
>> mesh(x,y,Z), title('3-D Surface Plot'), grid
>> xlabel('X'), ylabel('Y'), zlabel('Z')
>>
```



```
MATLAB_window
>>
>> t = 0:pi/20:10*pi;
>> plot3(sin(t),cos(t),t)
>> title(' 3-D Line Plot ')
>> grid
>>
```