EECE 231: GRAPHICS – 2D AND 3D PLOTS
READING: BIELAJEW, CHAPTER 12
MATLAB HELP FOR GRAPHICS AND 3D VISUALIZATION

OBJECTIVES

- Go over the basic graphics support in Matlab.
- Illustrate 2-D plotting Matlab functions and utilities.
- Illustrate 3-D plotting Matlab functions and utilities.
- Illustrate how to annotate plots with labels, texts and shapes.

OUTLINE

- BASIC ELEMENTS OF PLOTS
- TWO DIMENSIONAL PLOTTING
- MULTIPLE PLOTS IN THE SAME FIGURE
- THREE DIMENSIONAL PLOTTING
  - Examples
- GENERATING PLOTS USING THE GRAPHICAL USER INTERFACE

MATLAB FIGURES

- Matlab offers a variety of data plotting functions and GUI tools to create and modify plots.
    - GUI help to complement the annotation commands.
- A *figure* is a matlab window that contains the graphic display
    - Plots
    - User interface components
- A figure can be created explicitly with the figure command
    - Or implicitly when plotting while no figure is active.

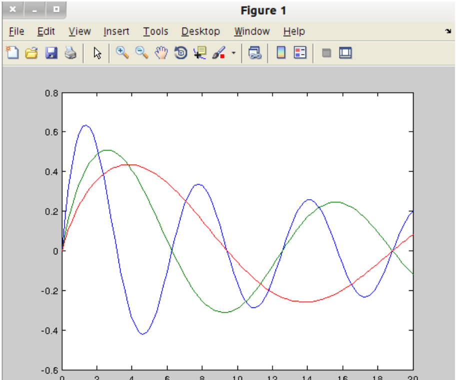
FIGURES, PLOTS AND AXES

- A *plot* is a graphic display you create in a figure window.
    - Tabular data, geometric, surface and image objects.
    - Annotations such as titles, legends, and colorbars.
- Each plot is created within a 2-D or a 3-D *axes* system.
    - You can create axes explicitly with the *axes* or the *subplot* functions.

PLOTTING EXAMPLE

- Array *x* forms the x-axis
- Array *y* has three rows each describes a plot.
- Matlab fills in between the values to create smooth curved lines.
- Matlab picks the colors for the various plots automatically.
- You can specify your own colors.

```
x = 0:.2:20;
y = sin(x)./sqrt(x+1);
y(2,:) = sin(x/2)./sqrt(x+1);
y(3,:) = sin(x/3)./sqrt(x+1);
plot(x,y)
```



OUTLINE

- BASIC ELEMENTS OF PLOTS
- TWO DIMENSIONAL PLOTTING
- MULTIPLE PLOTS IN THE SAME FIGURE
- THREE DIMENSIONAL PLOTTING
  - Examples
- GENERATING PLOTS USING THE GRAPHICAL USER INTERFACE

COLORS AND TOOLBARS

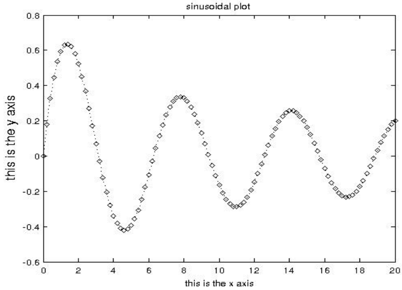
- You can specify your own colors.
- You can use the menu options to edit the plot.
- You can enable two additional toolbars from the View Menu
  - Camera: changes perspectives in 3D
  - Plot Edit: allows for annotations and labels.

```
%Using different line colors
%Choose from:
plot(x,y,'b'); %b: blue
plot(x,y,'g'); %g: green
plot(x,y,'r'); %r: red
plot(x,y,'c'); %c: cyan
plot(x,y,'m'); %m: magenta
plot(x,y,'y'); %y: yellow
plot(x,y,'k'); %k: black
plot(x,y,'w'); %w: white
```

COMMAND LINE PLOTTING OPTIONS

- You can add labels for the axes and the title from the commands line.
- You can control the line style.
- You can plot with point symbols.
    - `.` for a point,
    - `x` for a cross,
    - $\star$ for an asterisk.
    - `d` for a diamond,
    - `o` for a circle,
    - `S` for square,
    - `p` for a pentagram.
    - `h` for hexagram . . .

```
x = 0:.2:20;
y = sin(x)./sqrt(x+1);
plot(x,y,’k:’); % k is for color black
                % : is for the dotted line
                % - solid,
                % -. dash-dot line,
                % -- dashed line

title(’three sinusoidal plots’);
xlabel(’ this is the x axis’);
ylabel(’\fontsize{16} this is the y axis’);

plot(x,y,’k:d’); % d for a diamond point symbol
```



EXAMPLE OF VARIOUS COMBINATIONS

```
s = linspace(0.01, 6, 20);
t = log(gamma(s));
plot(s,t)
title(...
'\fontsize{20}The log\Gamma(x) function')
xlabel('\fontsize{20}x')
ylabel('\fontsize{20}y')
plot(s,t,'k:.')
plot(s,t,'g-.o')
plot(s,t,'r--x')
plot(s,t,'b-+')
plot(s,t,'k:*')
plot(s,t,'g-.S')
```

OUTLINE

BASIC ELEMENTS OF PLOTS

TWO DIMENSIONAL PLOTTING

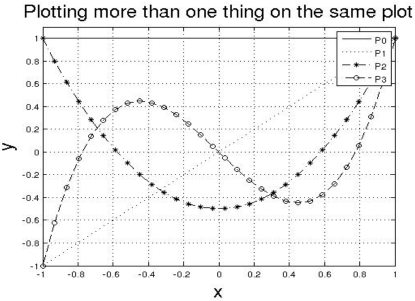
MULTIPLE PLOTS IN THE SAME FIGURE

THREE DIMENSIONAL PLOTTING
- Examples

GENERATING PLOTS USING THE GRAPHICAL USER INTERFACE

PLOTTING MULTIPLE THINGS AT ONCE

```
x = linspace(-1, 1, 30);
f0 = ones(1,length(x));
f1 = x;
f2 = (1/2)*(3*x.^2 - 1);
f3 = (1/2)*(5*x.^3 - 3*x);
plot(x,f0,'k-',x,f1,'k:',x,f2,'k-.*',...
  x,f3,'k--o')
title(...
'\fontsize{20} Plotting multiple things')
xlabel('\fontsize{20}x')
ylabel('\fontsize{20}y')
axis([-1,1,-1,1.1]) %[xmin,xmax,ymin,ymax]
grid on % Turn the grid on
legend('P0','P1','P2','P3')
```



Another way: **use `hold on`**

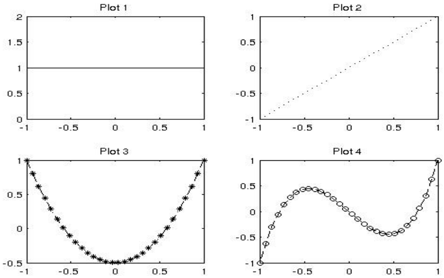
```
plot(x,f0,'k-')
hold on
plot(x,f1,'k:')
plot(x,f2,'k-.*')
plot(x,f3,'k--o')
axis([-1,1,-1,1.1]) %[xmin,xmax,ymin,ymax]
grid on % Turn the grid on
legend('P0','P1','P2','P3')
```

Yet another way is to place all in one array

```
f = f0;
f(2,:) = f1;
f(3,:) = f2;
f(4,:) = f3;
plot(x,f)
axis([-1,1,-1,1.1]) %[xmin,xmax,ymin,ymax]
grid on % Turn the grid on
legend('P0','P1','P2','P3')
```

FIGURE CAN BE A GRID OF PLOTS WITH *subplot*

- The `subplot(n,m,x)` command creates an *n* by *m* grid of plots
  - makes grid *x* active.
- Counting starts from top left and moves right, and then down.

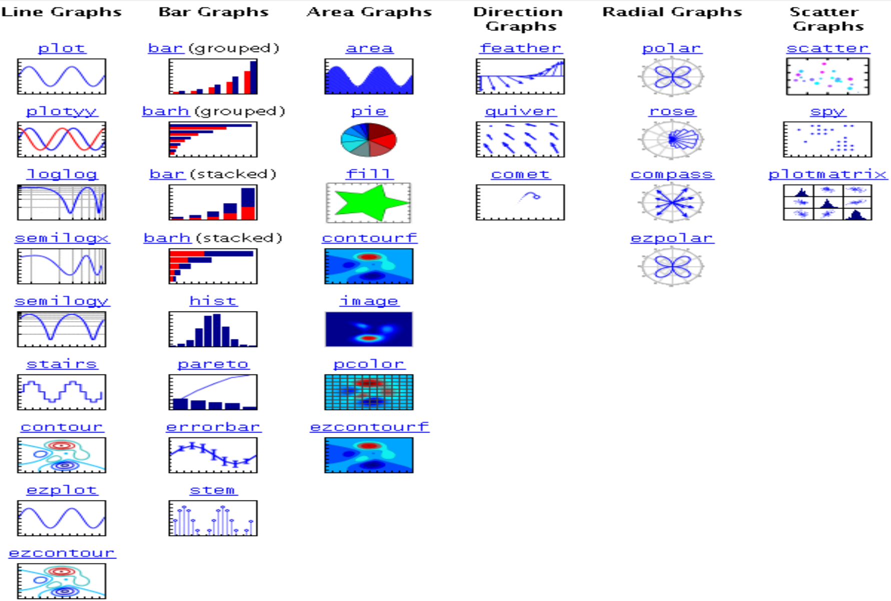


```
subplot(2,2,1),
plot(x,f0,’k-’),
title(’Plot 1’)

subplot(2,2,2),
plot(x,f1,’k:’),
title(’Plot 2’)

subplot(2,2,3),
plot(x,f2,’k-.*’),
title(’Plot 3’)

subplot(2,2,4),
plot(x,f3,’k--o’),
title(’Plot 4’)
zoom on
```

TWO DIMENSIONAL PLOTTING FUNCTIONS

- Line graphs
  - plot, plotyy, loglog, semilogx, semilogy, stairs, countour, ezplot, ezcountour
- Bar graphs
  - bar (grouped, stacked), barh (grouped, stacked), hist, parto, errorbar, stem
- Area graphs
  - area, pie, fill, countourf, image, pcolor, ezcountourf
- Direction graphs
  - feather, quiver, comet
- Radial graphs
  - polar, rose, compass, ezpolar
- Scatter graphs
  - scatter, spy, plotmatrix

MULTI -



Line Graphs	Bar Graphs	Area Graphs	Direction Graphs	Radial Graphs	Scatter Graphs
plot	bar (grouped)	area	feather	polar	scatter
plotyy	barh (grouped)	pie	quiver	rose	spy
loglog	bar (stacked)	fill	comet	compass	plotmatrix
semilogx	barh (stacked)	contourf		ezpolar	
semilogy	hist	image			
stairs	pareto	pcolor			
contour	errorbar	ezcontourf			
ezplot	stem				
ezcontour					

OUTLINE

BASIC ELEMENTS OF PLOTS

TWO DIMENSIONAL PLOTTING

MULTIPLE PLOTS IN THE SAME FIGURE

THREE DIMENSIONAL PLOTTING

- Examples

GENERATING PLOTS USING THE GRAPHICAL USER INTERFACE

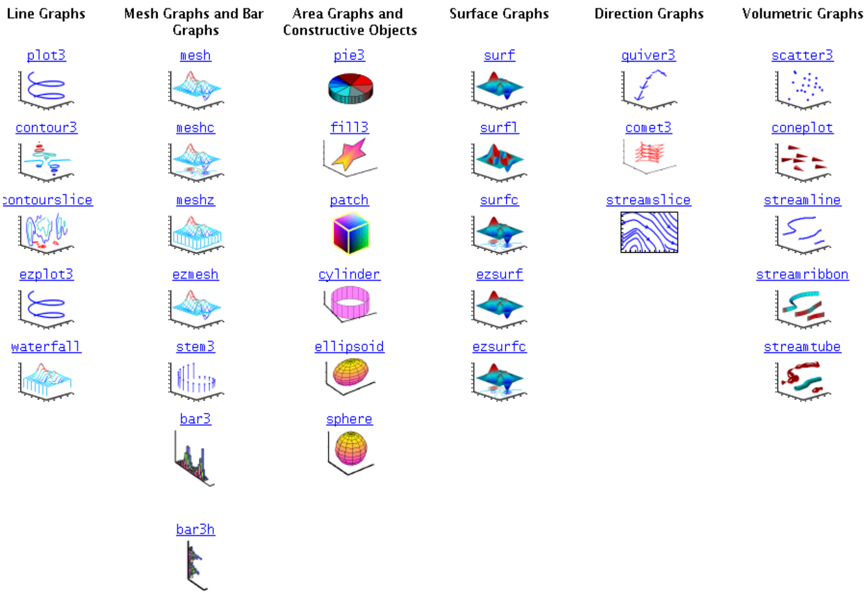
THREE DIMENSIONAL PLOTTING

- Also referred to as *volumetric* plotting.
- Functions exist to generate data for basic 3D shapes
  - cylinder, sphere, ellipsoid
  - You can superimpose your data upon them.

THREE DIMENSIONAL PLOTTING FUNCTIONS

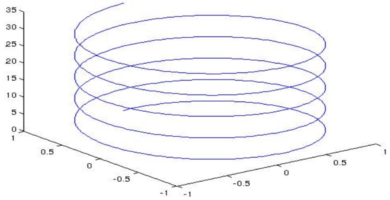
- Line graphs
  - plot3, contour3, contourslice, ezplot3, waterfall
- Bar graphs
  - mesh, meshc, meshz, ezmesh, stem3, bar3, bar3h
- Area graphs
  - pie3, fill3, patch, cylinder, ellipsoid, sphere
- Surface graphs
  - surf, surfl, surfc, ezsurf, ezsurfc
- Direction graphs
  - quiver3, comet3, streamslice
- Volumetric graphs
  - scatter3, coneplot, streamline, streamribbon, streamtube

3D -

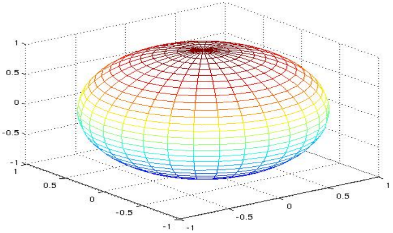


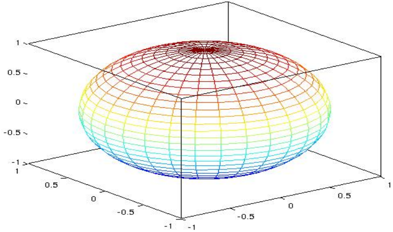
Line Graphs	Mesh Graphs and Bar Graphs	Area Graphs and Constructive Objects	Surface Graphs	Direction Graphs	Volumetric Graphs
plot3	mesh	pie3	surf	quiver3	scatter3
contour3	meshc	fill3	surfl	comet3	coneplot
contourslice	meshz	patch	surfc	streamslice	streamline
ezplot3	ezmesh	cylinder	ezsurf		streamribbon
waterfall	stem3	ellipsoid	ezsurfc		streamtube
	bar3	sphere			
	bar3h				

EXAMPLES WITH PLOT3 AND MESH

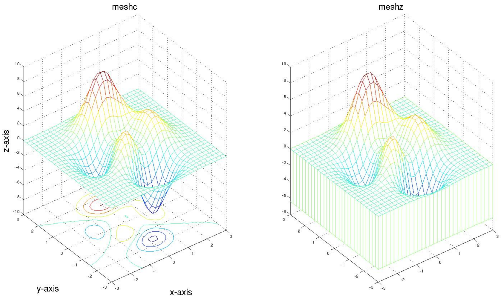


```
% Example using plot3
t = linspace(0, 10*pi, 1000);
plot3(sin(t),cos(t),t)

% Mesh plots
[x,y,z] = sphere(30);
mesh(x,y,z)

grid off % Turn off the grid
box on % Turn on the box

grid on % Turn on the grid
box off % Turn off the box
```



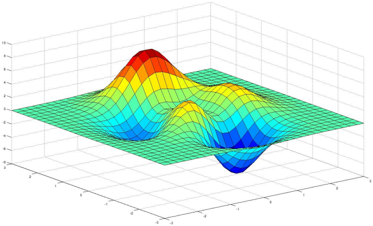


LABELING AXIS, SUBPLOT, MESHC, AND MESHZ

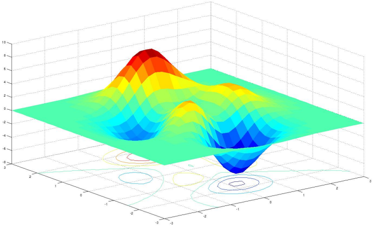


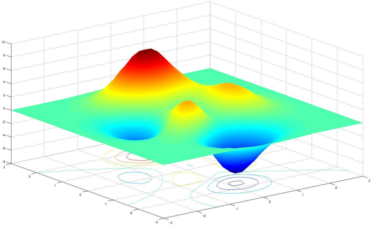
```
% Two other forms
% of mesh plots

% meshc
[x,y,z] = peaks(30);

subplot(1,2,1)
meshc(x,y,z)
title('\fontsize{20}meshc')
xlabel('\fontsize{20}x-axis')
ylabel('\fontsize{20}y-axis')
zlabel('\fontsize{20}z-axis')

% and meshz
subplot(1,2,2)
meshz(x,y,z)
title('\fontsize{20}meshz')
```

FUNCTION *surf* AND SHADING

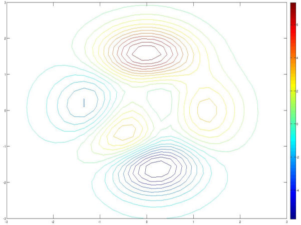


```
[x,y,z] = peaks(30);
surf(x,y,z)

surfc(x,y,z) % surf with contours
shading flat % Flat shading

shading interp

% the results of the following are
% not shown. try them out
surfl(x,y,z) % surf with lighting
colormap gray
```

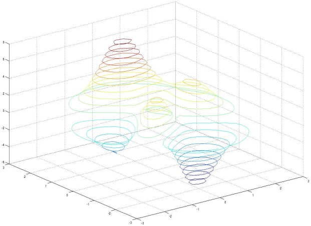


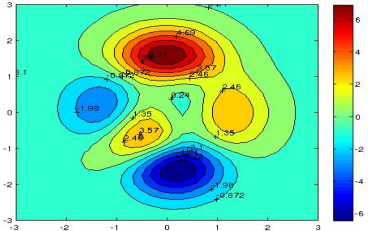


CONTOUR PLOTS



```
[x,y,z] = peaks(30);
% treats values of Z as heights above a plane
% uses colors from colormap to encode values
contour(x,y,z,20) % draw 20 contour lines

colorbar % Show the colorbar

%3D contours
contour3(x,y,z,20)

% Example using filled contour
C = contourf(x,y,z,12);
clabel(C)
colorbar % Show the colorbar
```





meshgrid FUNCTION

- Generates coordinate space:

```
[X,Y] = meshgrid(x,y)
[X,Y] = meshgrid(v)
    % means [X,Y] = meshgrid(x,y), where x = v and y = v
```

- *meshgrid* outputs $m \times n$ matrices X and Y constructed from repetitions of the $1 \times n$ input vector x and $1 \times m$ input vector y in such a way that $X(i,j) = x(i)$ and $Y(i,j) = y(j)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$.
- Examples:

```
[X,Y] = meshgrid(-8:.5:8, -8:.5:6)
Z = X.*Y+X;
mesh(X,Y,Z);
```

meshgrid FUNCTION (CONTINUED)

- Why turn x and y into matrices X and Y?
- So that we can fill Z by point-wise operations as above. Without using *meshgrid*, we need loops

```
x =  -8:.5:8;
y =-8:.5:6;
for i=1:length(x),
    for j  = 1:length(y);
       Z(i,j) = x(i)*y(i) +x(i);
    end;
 end;
 mesh(x,y,Z);
```

  Too long and not the Matlab way!

- Using point-wise operations

```
Z = X.*Y+X;
```

  is much simpler

OUTLINE

- BASIC ELEMENTS OF PLOTS
- TWO DIMENSIONAL PLOTTING
- MULTIPLE PLOTS IN THE SAME FIGURE
- THREE DIMENSIONAL PLOTTING
  - Examples
- GENERATING PLOTS USING THE GRAPHICAL USER INTERFACE

GENERATING THE PLOTS FROM THE GUI

- Interactively generate any of the plots using
  - The *PlotSelector* and *PlotCatalog* tools.
- Select one or more numeric workspace variables in
  - The *WorkspaceBrowser*,
  - The *VariableEditor* or
  - The plotting tools *FigurePalette*.
- Open the *PlotSelector* or *PlotCatalog* tool
  - Scroll to the graph you want to create.
  - Click the icon to plot the selected variables.
- The graph displays in the current figure,
  - The command that generated it displays in the *CommandWindow*.
- Alternatively, right-click a selected variable and choose *PlotCatalog* from the context menu.

INTERESTING AND USEFUL STUFF

- You can start more than one figure with the `figure` command.
    - `figure (x)` makes figure *x* the active and current figure.
- Advanced 2D plotting (not required):
    - Define multiple axes
    - Draw with your own colormap
- Advanced 3D topics (not required):
    - Camera and viewpoint positioning
    - Lighting and transparency
    - Texture and faces
    - Coloring: RGB, Indexed, CYMK, HSU, ...

HAVE FUN WITH MORE MATLAB GRAPHICS (OPTIONAL)!

TRY THE FOLLOWING

```
load earth % earth is a workspace saved by default
sphere;
h = findobj('Type','surface');
hemisphere = [ones(257,125),...
              X,...
              ones(257,125)];
set(h,'CData',flipud(hemisphere),'FaceColor','texturemap')
colormap(map)
axis equal
view([90 0])
pause % press enter for this to return

set(gca,'CameraViewAngleMode','manual')
view([65 30])
pause % press enter for this to return
view([65 45])
```

EECE 231: GRAPHICS – 2D AND 3D PLOTS

READING: BIELAJEW, CHAPTER 12

MATLAB HELP FOR GRAPHICS AND 3D VISUALIZATION