**INTRODUCTION TO GRAPHICS**

The objective of this tutorials is to introduce you to MATLAB’s high-level 2-D and 3-D plotting facilities.

**Basic 2-D graphs**

# Graphs (in 2-D) are drawn with the plot function or command. The plot function in Matlab is used to create a graphical representation of some data.

# The Plot Function

# The plot function usually takes two arguments, plot(x, y), (but can take one).

# The first is the X values of the points to plot, and the second is the Y value of the points to plot.

# The plot command can be executed if and only if the arrays x and y have the same length (number of elements).

# The resulting plot may not be smooth, unless a sufficiently large number of points are employed.

# NB: in terms of coordinates, the x-values represent the Easting coordinates and the y-values represent the Northing coordinates.

# PLOTS SYNTAX

# TWO OR MORE

# PLOT(X, Y) plots vector Y versus vector X. If X or Y is a matrix, then the vector is plotted versus the rows or columns of the matrix. Various line types ,plot symbols and colors may be obtained with

# PLOT(X,Y,S) where S is a character string made from one element from any or all the following:

### Line Style Specifiers

| **Specifier** | **Line Style** |
| --- | --- |
| - | Solid line (default) |
| -- | Dashed line |
| : | Dotted line |
| -. | Dash-dot line |

### Marker Specifiers

| **Specifier** | **Marker Type** |
| --- | --- |
| + | Plus sign |
| o | Circle |
| \* | Asterisk |
| . | Point (see note below) |
| x | Cross |
| 'square' or s | Square |
| 'diamond' or d | Diamond |
| ^ | Upward-pointing triangle |
| v | Downward-pointing triangle |
| > | Right-pointing triangle |
| < | Left-pointing triangle |
| 'pentagram' or p | Five-pointed star (pentagram) |
| 'hexagram' or h | Six-pointed star (hexagram) |

### Color Specifiers

| **Specifier** | **Color** |
| --- | --- |
| r | Red |
| g | Green |
| b | Blue |
| c | Cyan |
| m | Magenta |
| y | Yellow |
| k | Black |
| w | White |

# i.e.

# plot(x,y,lineSpec,lineWidth,markerEdge,markerFace,markerSize)

For example:

plot(x,y,'-.or')

# plots y versus x using a dash-dot line (-.), places circular markers (o) at the data points, and colors both line and marker red (r).

### Plotting Data Points with No Line

If you specify a marker, but not a line style, only the markers are plotted. For example:

plot(x,y,'d')

**Labeling your Plot**

The following functions are almost always used with plot to make the output more readable.

1. xlabel, ylabel : Provide a labeling for the graph axii.
2. title : Provide a title for the graph.
3. legend: Provide a legend telling what multiple graph lines mean.
4. grid: Put a checkered grid over the graph (add more lines to (sometimes) make it easier to see values.
5. axis: Tell how big of an area the graph should display.

## Examples

Plot a sine curve:

x = -pi:.1:pi;

y = sin(x);

plot(x,y)

xlabel( ‘x-axis’ , ‘FontSize’ ,14, ‘FontWeight’,’b’);

ylabel( ‘y-axis’ ,’ FontSize’ ,14, ‘FontWeight’,’b’);

title( ‘Sine Curve’ , ‘FontSize ‘,14, ‘FontWeight’, ‘b ‘);

legend(‘sine’)

**specifying line color,marker type**

plot(x,y,'-o','color','r')

or

plot(x,y,’-or’)

**Create line plot using specific line width, marker color, and marker size:**

plot(x,y,'--rs','[LineWidth](http://www.mathworks.com/access/helpdesk/help/techdoc/ref/lineseriesproperties.html#LineWidth)',2,...

'[MarkerEdgeColor](http://www.mathworks.com/access/helpdesk/help/techdoc/ref/lineseriesproperties.html" \l "MarkerEdgeColor" \t "_top)','k',...

'[MarkerFaceColor](http://www.mathworks.com/access/helpdesk/help/techdoc/ref/lineseriesproperties.html" \l "MarkerFaceColor" \t "_top)','g',...

'[MarkerSize](http://www.mathworks.com/access/helpdesk/help/techdoc/ref/lineseriesproperties.html#MarkerSize)',10)

**To give the a black background, use the command set(gcf,'color',...)**

set(gcf, 'color','black')

**To set the background of the graph itself, use the command set(gca,'color',...). Notice this command uses gca while the window background uses gcf.**

set(gcf,'color',[0,0,0])  
set(gca,'color',[0.2,0.2,0.2])

set(gcf,'color',[1,0.5,0]) % plot color to orange

set(gca,'color',[0,0.1,0.2]) ]) %background color to dark blue.

**To create a graphic with the title of your choice**

set(gcf,'name','Sine Plot ')

**scaling Axis**

axis([-2,5,-1,4]) % NOTICE this goes AFTER the plot() command [xmin ,xmax ,ymin, ymax]

**To combine two or more graphs**

there are two methods.

* the first method uses one line and is of the form plot(x, function, style#1 ,x, function#2, style#2, ). For example,

y1=cos(x)

plot(x,y,'-or',x,y1,'-\*g')

legend('sine','cos')

* The second method involves the use of the "hold on" and "hold off" commands

After creating the first graph, use the "hold on" command, forcing subsequent graphing commands to be placed in the same window. Now type in your second plot command. When you are through, type "hold off". For example

plot(x,y,'-or')

hold on

Plot(x,y1,'-\*g')

hold off

grid on % adding grid line

**The "clf" command**

Sometimes you will want to "clear" the figure. This is normally done by simply "plotting" the next thing you want. Alternatively, you can use the "clf" command:

>> clf; % clear the plot

**The "subplot" command**

Sometimes you will want to place multiple plots side by side on a single figure. You can achieve this by using the Matlab subplot function.

The parameters used by the subplot function determine how many rows and columns for the overall "matrix" of figures, and then which of these positions to put the next plot in. The parameters are: 1) The number of rows, 2) the number of columns, 3) which of the sub plots to use.

i.e. subplot(m,n,i) creates *mn* plots, arranged in an array with *m* rows and *n* columns. It also sets the next plot command to go to the *i*th axes for the current plot The axes

are counted along the top row of the Figure window, then the second row,etc (counting across the rows).

Here is an example of how to create a 2x3 matrix (6 figures) of plots and to address each one of them by a number (shown below). Notice how row 1, column 1, is identified by the number 1. Row one, column three is identified by 3. Row two, column one, "wraps" and is identified by 4.:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Columns** | | |
|  |  | one | two | three |
| **Rows** | **one** | 1 | 2 | 3 |
| **two** | 4 | 5 | 6 |

**Example**

>> t = (0:.1:2\*pi)';

>> subplot(2,2,1)

>> plot(t,sin(t))

>> subplot(2,2,2)

>> plot(t,cos(t))

>> subplot(2,2,3)

>> plot(t,exp(t))

>> subplot(2,2,4)

>> plot(t,1./(1+t.^2))

### Creating Simple Contour Plots

display 20 contour levels of the peaks function

[X,Y,Z] = peaks;

contour(X,Y,Z,20)

NB: You can ignore the 20

### Labeling Contours

display 10 contour levels of the peaks function

[X,Y,Z] = peaks;

[C,h]=contour(X,Y,Z,10)

Clabel(C,h)

Label the contours and display a title:

clabel(C,h)

title({'Contour Labeled Using','clabel(C,h)'})

**surface plot(surf)**

**1.>> surf(X,Y,Z)**

**2.>> [x,y]=meshgrid(-1:.1:1,-1:.1:1);**

**surf(x,y,x.^2+y.^2)**

**CREATING BAR GRAPHS**

>>a = [1 2 3 4 6 4 3 4 5]

>>b = a + 2

bar(b)

xlabel('Sample #')

ylabel('Pounds')

NB: matlab has a lot on plot functions.read more on them.