زانكۆی سەلاحەدین - هەولێر
Salahaddin University-Erbil

University of Salahaddin
College of Education
/Department of physics

Lecture 4
Plotting

Diyar Rasool

4. Plotting

- The graph plotting in MATLAB is quite powerful and easy to use.

- MATLAB can plot many common types of graphs (e.g., scatter, pie, histogram, polar, etc.), including both 2D and 3D graphs.

- It is not only very useful to provide visual data feedback to users but also a great tool for creating graphs for publications and presentations.

4.1 Creating Simple Plots

- A simple 2D graph can be plotted using the "**plot**" function.

$$\boldsymbol{plot(xvalues, yvalues,' style - option')}$$

- where xvalues and yvalues are vectors containing the x- and y-coordinates of points on the graph.

- The style option is an optional argument that specifies the color, the line style (e.g., solid, dashed, dotted), and the point-marker style (e.g., o, +,*).

- The two vectors xvalues and yvalues MUST have the **same length**.

4.1 Creating Simple Plots

Example 1: The vectors x = (1,2,3,4,5,6) and y = (3,1,2,4,5,1) produce the picture.

Ans: >> x = 1:6; >> y = [3 -1 2 4 5 1];

>> plot(x,y), plot(x,y,'r'), plot(x,y,':'), plot(x,y,'b- -') or plot(x,y,'*'), so on.

Example 2: Plot a parabola function $y = x^2$ for an x-value range between −10 and 10.

Ans: clc, clear all, close all

x=linspace(-10,10,11); y=x.^2; figure; plot(x,y,'ro')

4.2 Adding titles, axis labels, Legend and other text objects

❖ MATLAB enables to add axis labels and titles. Thus

`xlabel('Pipe Length')`	labels the x-axis with `Pipe Length`,
`ylabel('Fluid Pressure')`	labels the y-axis with `Fluid Pressure`,
`title('Pressure Variation')`	titles the plot with `Pressure Variation`, and
`text(2,6,'Note this dip')`	writes "`Note this dip`" at the location (2.0,6.0) in the plot coordinates.

❖ **Legend:** this command produces a boxed legend on a plot. The most commonly used forms of the command are listed here.

`legend(`*`string1, string2, ..`*`)`	produces legend using the text in *string1*, *string2*, etc., as labels,
`legend(`*`LineStyle1, string1, ..`*`)`	specifies the line style of each label,
`legend(..,` *`pos`*`)`	writes the legend outside the plot-frame if $pos = -1$ and inside if $pos = 0$, (there are other options for *pos* too), and
`legend off`	deletes the legend from the plot.

❖ **legend('First','Second')**

4.3 Specifying line styles and colors

❖ It is possible to specify line styles, colors, and markers (e.g., circles, plus signs, . . .) using the plot command:

$$\boldsymbol{plot(x, y, 'style\_color\_marker')}$$

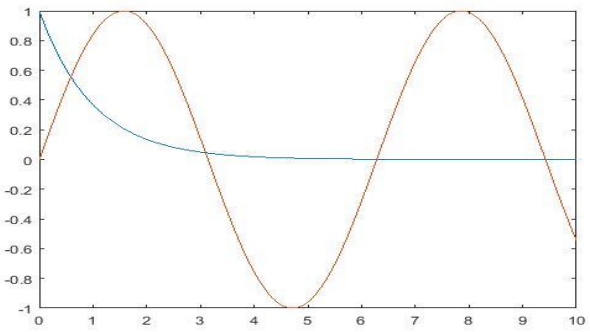
❖ To find additional information, type help plot or doc plot.

Symbol	Color	Symbol	Line Style	Symbol	Marker
k	Black	–	Solid	+	Plus sign
r	Red	– –	Dashed	o	Circle
b	Blue	:	Dotted	*	Asterisk
g	Green	–.	Dash-dot	.	Point
c	Cyan	none	No line	×	Cross
m	Magenta			s	Square
y	Yellow			d	Diamond

4.4 Multiple data sets in one plot

- Multiple (x; y) pairs arguments create multiple graphs with a single call to plot.

Example 1: x = 0:0.01:10; plot(x, exp(-x), x, sin(x))



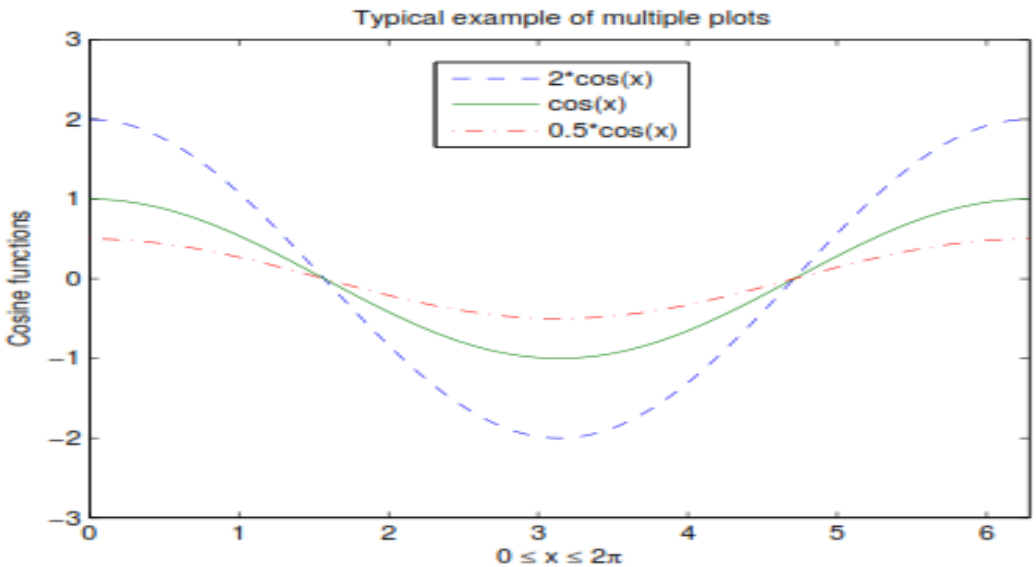
Example 2: x=linspace(-10,10,11);

figure; plot(x,x.^2,'ro'); hold on

plot(x, abs(10*x),'b-'); hold off

Example

Plot three related functions of x: $y_1 = 2\cos(x)$, $y_2 = \cos(x)$, and $y_3 = 0.5\cos(x)$, in the interval $0 \leq x \leq 2\pi$. Give the label names, title and legend?



Ans: >> x = 0:pi/100:2*pi;

>> y1 = 2*cos(x);

>> y2 = cos(x);

>> y3 = 0.5*cos(x);

>>plot(x,y1,'--',x,y2,'-',x,y3,':');>>xlabel(‘$0 \leq x \leq$‘) ;

>>ylabel('Cosine functions'); >>legend(‘2*cos(x)’,’cos(x)’,0.5*cos(x)’)

>> title (‘typical example of multiple plots’);

>> axis ([0 2*pi -3 3]); axis is used to control axis.

4.5 Axis control

❖ Once a plot is generated, you can change the axes limits with the axis command. Typing

```
axis([xmin xmax ymin ymax])
```

Example:

axis([-5 10 2 22]); means $x - axis$ from -5 to 10, $y - axis$ from 2 to 22.

❖ There are also some useful predefined string arguments for the axis command:

`axis('equal')`	sets equal scale on both axes,
`axis('square')`	sets the default rectangular frame to a square,
`axis('normal')`	resets the axis to default values,
`axis('axis')`	freezes the current axes limits, and
`axis('off')`	removes the surrounding frame and the tick marks.

4.6 Finding the slope of a plot

1. Ginput is a way to get the coordinates of a point by typing **ginput(1)** which allows you to click on any point in the figure window.

❖ The **ginput** can also used to compute slope and intersection.

Example: Find the slope for these two sets of data.

```
c=[650,580,540,500,490,450,400];     t=[76,156,66,46,34,24,56];

plot(c,t)      title 'ginput';      xlabel 'c';      ylabel t;

y=ginput(2);          slope=(y(1,2)-y(2,2)/y(1,1)-y(2,1))
```

4.7 Three dimensional plots

4.10.1: Curves in Three-Dimensional Space: The basic command is **plot3** which works like a **plot**, except that it takes three vectors (same lengths) instead of two.

Example1: **>> x=[1 5 12 34]; y=1:4; z=linspace(2,8,4); plot3(x,y,z)**

Example2: A Three Dimensional Parametric plot:

>> t = -2:0.01:2; **>> plot3(cos(2*pi*t), sin(2*pi*t), t)**

4.10.2 Surfaces in Three-Dimensional Space: There are two basic commands for plotting surfaces in 3- space: **mesh** **(**transparent) and **surf** **(opaque)**. Input vectors can have different lengths:

Example: **>> [x,y] = meshgrid(-2:0.1:2, -2:0.1:2);**

>> z = x.^2 - y.^2; mesh(x, y, z); surf(x,y,z)

4.8 Combining Plots in One Window

- The command **subplot** divides the figure window into an array of smaller plots.

- The first two arguments give the dimensions of the array of subplots, and the last argument gives the number of the subplot.

Example1: x = 0:0.05:40; subplot(2,1,1), plot(x, exp(x))
subplot(2,1,2), plot(x, sin(x))

Example2: x = linspace(0,10); y1 = sin(x); y2 = sin(2*x); y3 = sin(4*x); y4 = sin(8*x);

```
Figure, subplot(2,2,1) plot(x,y1) title('Subplot1');
 subplot(2,2,2) plot(x,y2) title('Subplot2')
subplot(2,2,3) plot(x,y3) title('Subplot3')
 subplot(2,2,4) plot(x,y4) title('Subplot 4')
```