ENGR 1100

Week 04 – Managing Data, 2D & 3D plots

Learning Objectives

Upon completion this module, students will be able to:

1. Demonstrate how to manage data in MATLAB,
   - ✓ Output Commands– input, disp, fprintf
   - ✓ Save & load commands
   - ✓ Importing and Exporting data
   - ✓ Example of MATLAB applications
2. Demonstrate how to create 2D and 3D Plots in MATLAB
   - ✓ The plot command
   - ✓ Example of MATLAB applications

Input & Output Commands

- The input command is used to assign value to a variable

MATLAB: 'input' command

- The disp command is used to display the elements of a variable **or** to display text.

MATLAB: 'disp' command

Input & Output Commands

- The fprintf command is used to display output (text and numerical data) on the screen or to save it to a file.

MATLAB: ‘fprintf' command

  - Output can be formatted. Text and numerical values of variables can be displayed in the same line.

Format Specifier	Meaning
s	String
d	Decimal notation
e	Exponential notation
f	Fixed point or decimal notation

MATLAB-- fprintf

- The fprintf command can also be used to save output to a file.
  1. Step1: Open the file to be written fid=fopen('file_name', 'permission')
  2. Step 2: write output to the file using fprintf

     fprintf(fid, 'format', variables)
  3. Step 3: Close the file when writing is complete

     fclose(fid)
- MATLAB example– **Velocity conversion**

Importing & Exporting Data with Excel

- Importing data from Excel

  variable_name=xlsread('filename', 'tab_name', 'range')

- Exporting data to Excel

  xlswrite('filename', variable_name)

- Use Import Wizard

Save AND Load command

MATLAB: ‘save’ command

MATLAB: ‘load’ command

- Use save command to save workspace or data,
- Use load command to retrieve stored workspace or data

```
save file_name OR save('file_name')
save file_name var1 var2 save('file_name','var1','var2’)

load file_name OR load('file_name')
load file_name var1 var2 load('file_name','var1','var2')
```

2D & 3D PLOT

2D plots

This section will cover 2D (two-dimensional) plots.
Many options:

- Linear, semi-logarithmic, logarithmic axes
- Line type, color, thickness
- Lots of different data-point markers
- Grid lines, titles, text comments, legends
- Subplots
- Bar, stair, polar plots

MATLAB: 2D plots

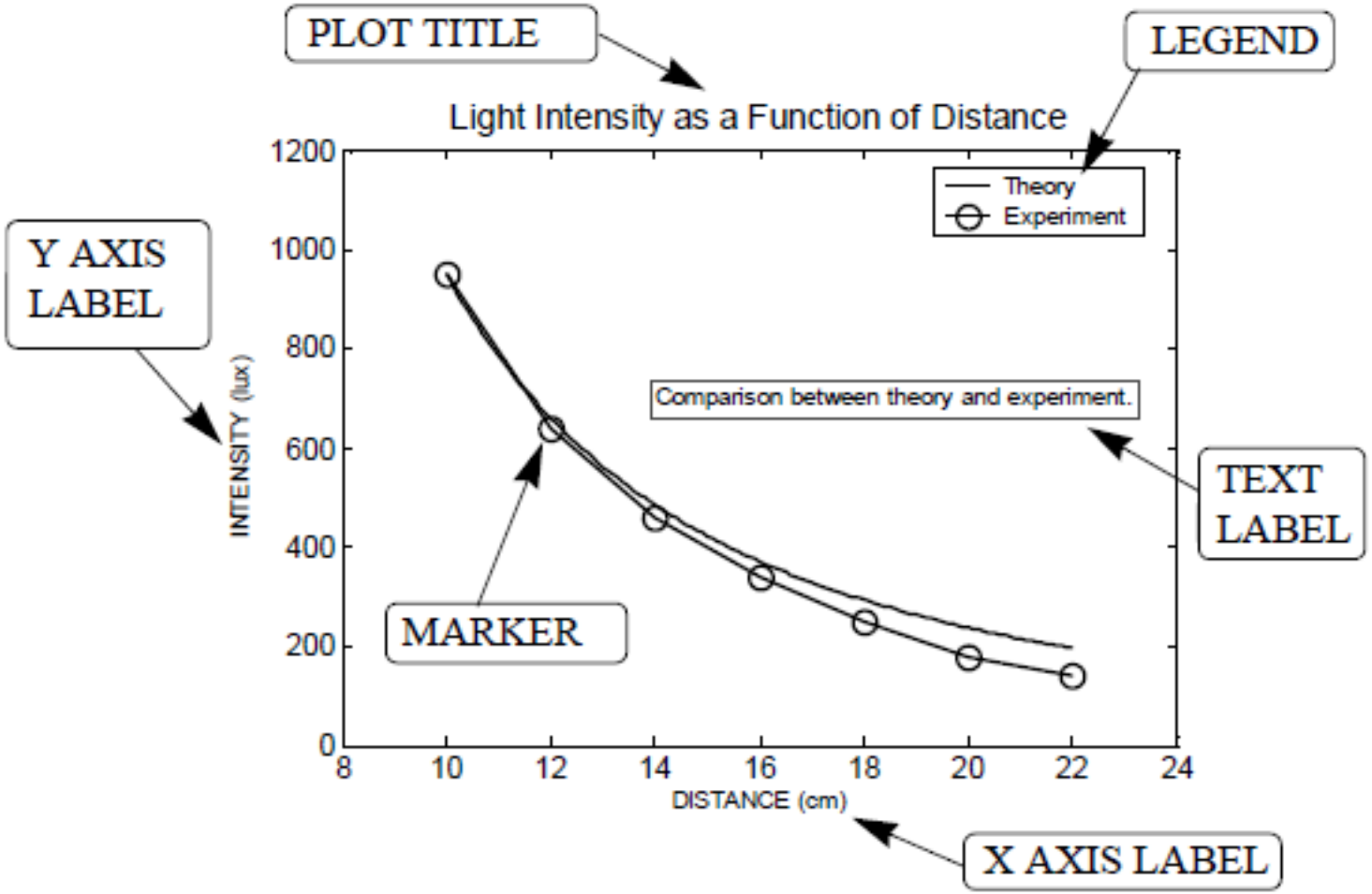


Figure 5-1: Example of a formatted two-dimensional plot.

The `plot` COMMAND

`plot` command used to make basic 2D plots. Simplest form is

`plot(y)`

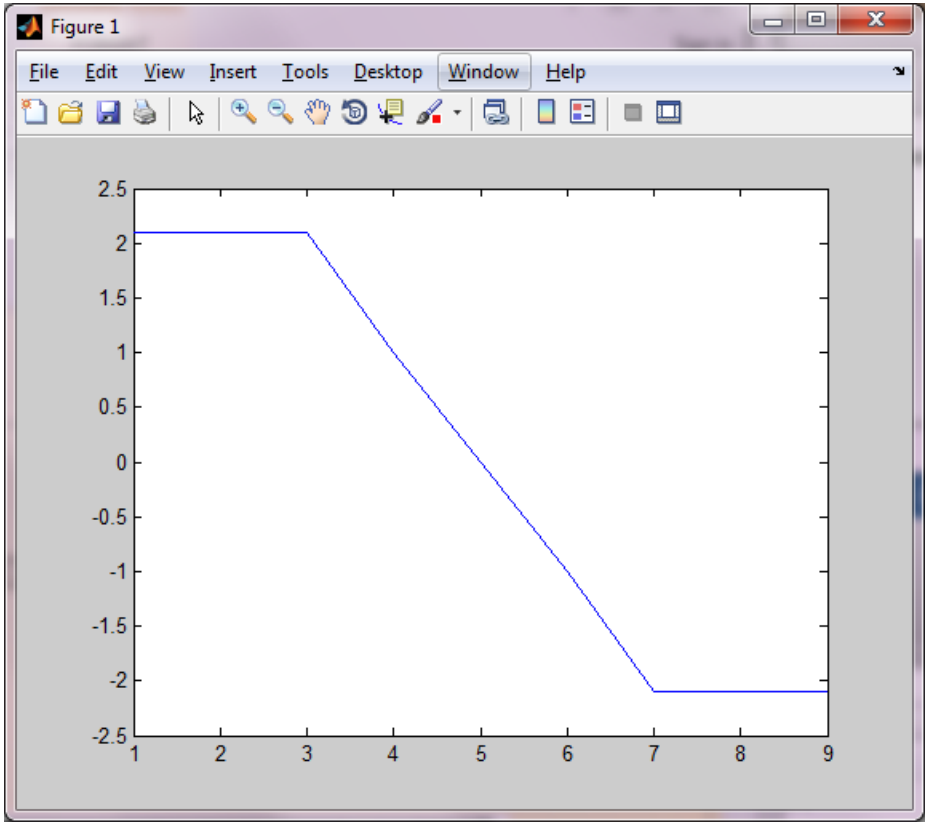
- Plots vector `y` on vertical axis, numbers 1 through N on horizontal axis (N = number of points in `y`)
- If there's a Figure Window, draws in it. Otherwise, creates a new Figure Window and draws in that

`plot(y)` default values

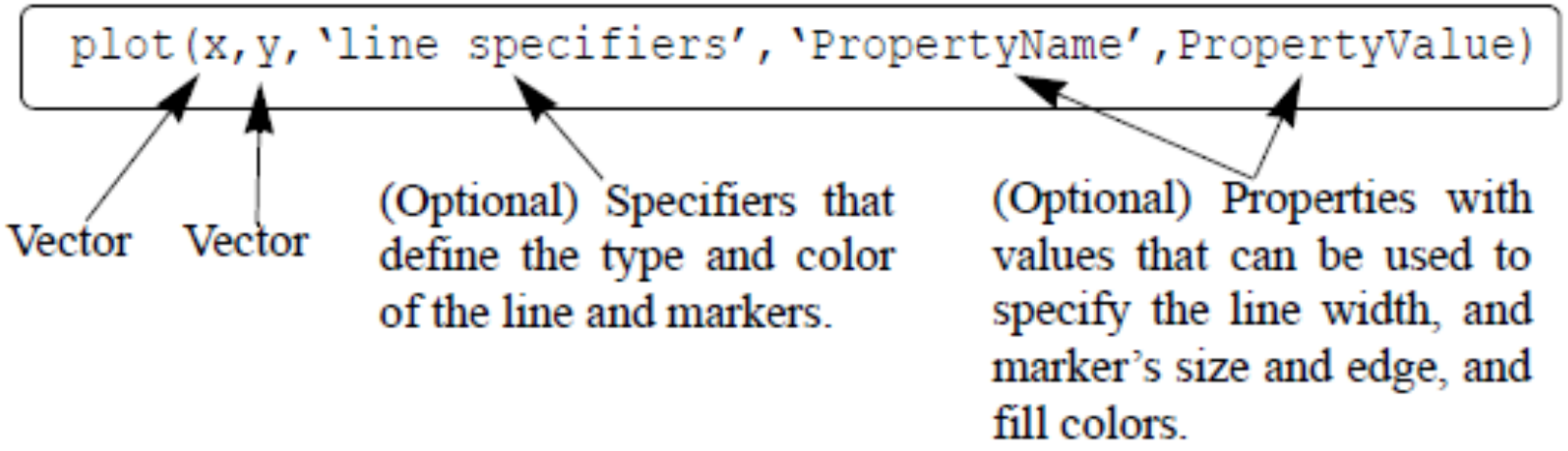
- Both axes linear
- Points connected by straight lines
- No point markers
- Points and lines in blue

Example: Plot

```
>> y = [ 2.1 2.1 2.1 1 0 -1 -2.1 -2.1 -2.1 ];
>> plot( y )
```



Plot command



Line Styles

Line Style	Specifier
solid (default)	-
dashed	--

Line Style	Specifier
dotted	:
dash-dot	-.

Line Colors

Line Color	Specifier
red	r
green	g
blue	b
cyan	c

Line Color	Specifier
magenta	m
yellow	y
black	k
white	w

Plot- Marker Types

Marker Type	Specifier		Marker Type	Specifier
plus sign	+		square	s
circle	o		diamond	d
asterisk	*		five-pointed star	p
point	.		six-pointed star	h
cross	x		triangle (pointed left)	<
triangle (pointed up)	^		triangle (pointed right)	>
triangle (pointed down)	v			

Notes about using the specifiers:

- The specifiers are typed inside the `plot` command as strings.
- Within the string the specifiers can be typed in any order.
- The specifiers are optional. This means that none, one, two, or all the three can be included in a command.

Some examples:

`plot(x,y)`	A blue solid line connects the points with no markers (default).
`plot(x,y,'r')`	A red solid line connects the points.
`plot(x,y,'--y')`	A yellow dashed line connects the points.
`plot(x,y,'*')`	The points are marked with * (no line between the points).
`plot(x,y,'g:d')`	A green dotted line connects the points that are marked with diamond markers.

YEAR	1988	1989	1990	1991	1992	1993	1994
SALES (millions)	8	12	20	22	18	24	27

```
>> yr=[1988:1:1994];
>> sle=[8  12  20  22  18  24  27];
>> plot(yr,sle,'--r*','linewidth',2,'markersize',12)
>>
```

Line Specifiers: dashed red line and asterisk marker.

Property Name and Property Value: the line width is 2 points and the markers size is 12 point.

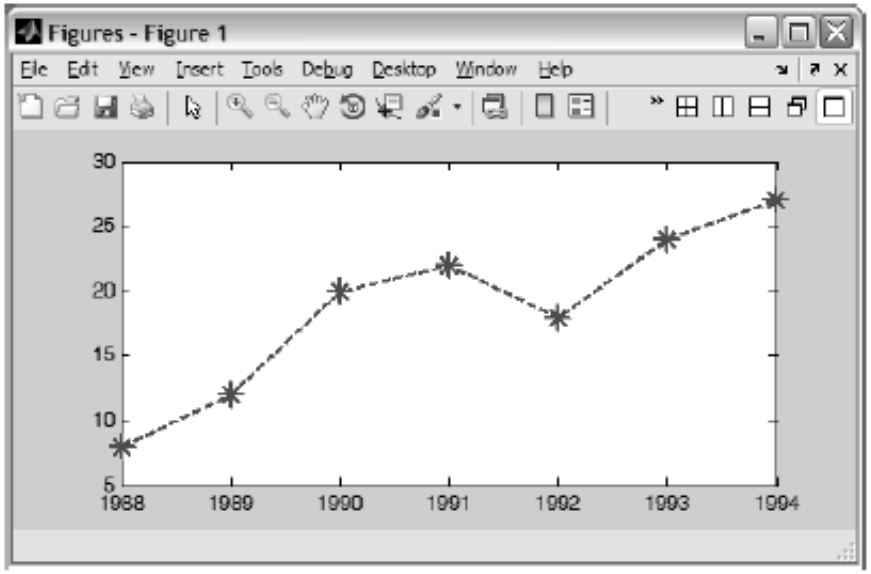
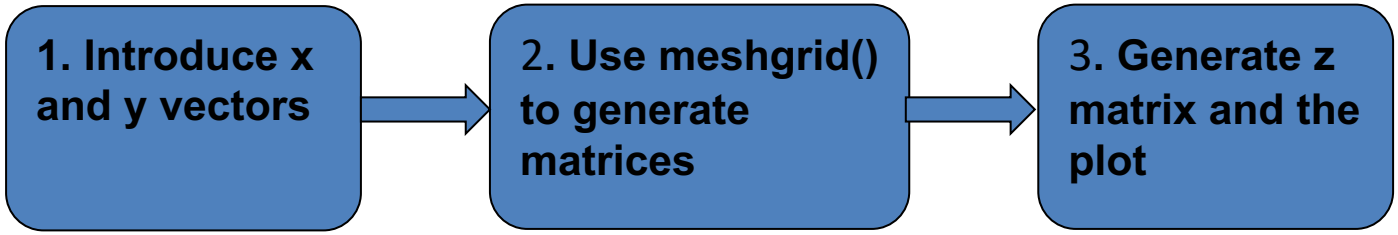


Figure 5-3: The Figure Window with a plot of the sales data.

3D plot

MATLAB: 3D plots

- How to plot a 3D surface



- 3D MATLAB commands:
  - mesh
  - surf
  - contour

MATLAB Plots With Special Graphics

MATLAB: Different type of plots

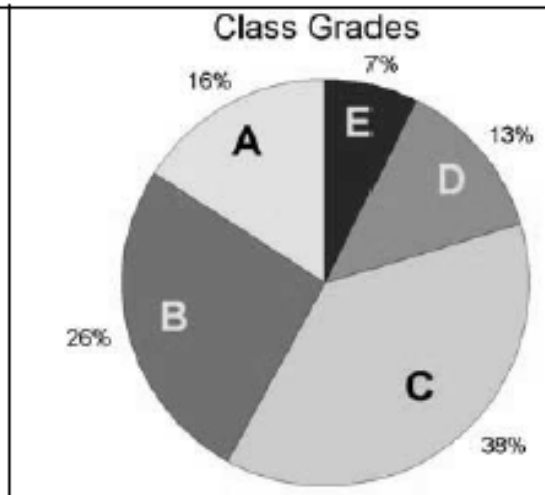
MATLAB has lots of special types of plots, e.g., bar, stairs, stem, pie

- For more information on types of plots, click on the Help icon, then on MATLAB, then scroll down to the Graphics section and click on 2-D and 3-D plots

Pie Plot

Function format:

```
pie(x)
```




```
grd=[11 18 26 9 5];
pie(grd)
title('Class Grades')
```

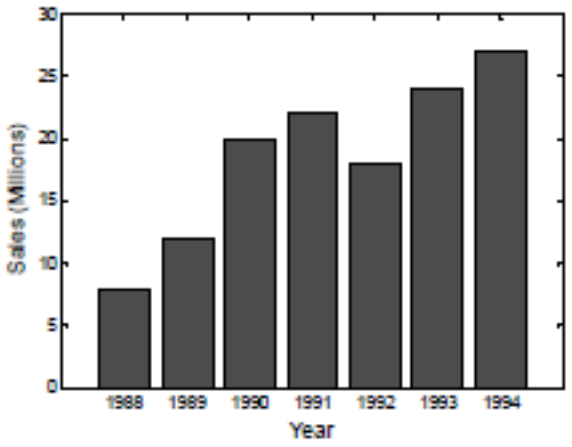
MATLAB draws the sections in different colors. The letters (grades) were added using the Plot Editor.

Examples of specialized plots

Vertical Bar Plot

Function format:

`bar(x,y)`



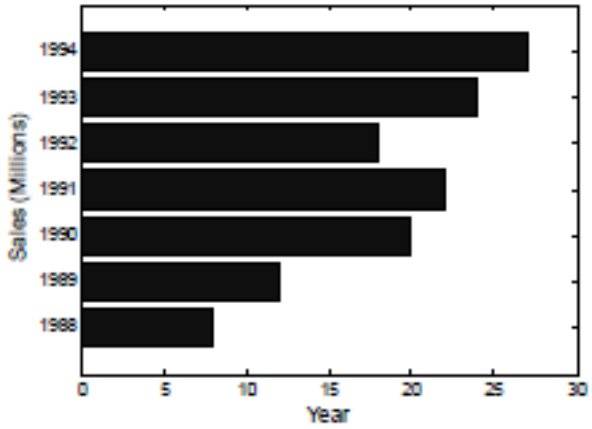

```
yr=[1988:1994];
sle=[8 12 20 22 18 24 27];
bar(yr,sle,'r')
xlabel('Year')
ylabel('Sales (Millions)')
```

The bars are in red.

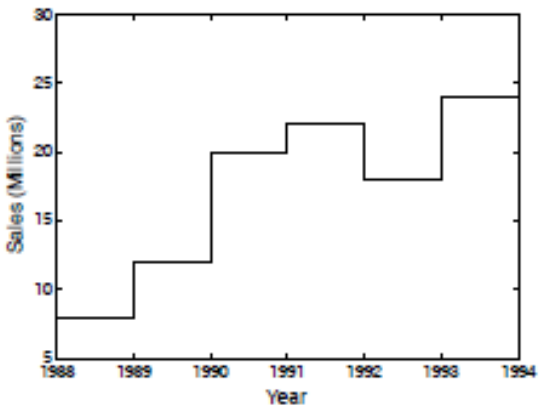
Horizontal Bar Plot

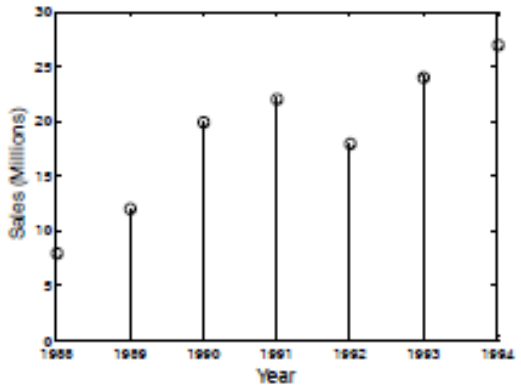
Function format:

`barh(x,y)`




```
yr=[1988:1994];
sle=[8 12 20 22 18 24 27];
barh(yr,sle)
xlabel('Sales (Millions)')
ylabel('Year')
```

Stairs Plot
 	
Function format:
 	
`stairs(x,y)`
		`yr=[1988:1994];`
		`sle=[8 12 20 22 18 24 27];`
		`stairs(yr,sle)`

Stem Plot
 	
Function Format
 	
`stem(x,y)`
		`yr=[1988:1994];`
		`sle=[8 12 20 22 18 24 27];`
		`stem(yr,sle)`

MATLAB Example 1: Average game points

You are asked to write a MATLAB script that calculate average points earned in 3 games. How would you approach it? How would you design your code?

Options:

1. Assign the values of each game scores in the script file
2. Assign the values of each game scores in the command windows (define variables first, then run script)
3. Ask user to input the values of each game 

MATLAB Example 2: Velocity Conversion

Write a MATLAB program that creates the velocity conversion table (miles/hour vs. kilometers/hour) and save the conversion table to a text file.

[Hint] The fprintf command can also be used to save output to a file.

1. Step1: Open the file to be written fid=fopen('file_name', 'permission')
2. Step 2: write output to the file using fprintf
   fprintf(fid, 'format', variables)
3. Step 3: Close the file when writing is complete
   fclose(fid)

MATLAB Example 3: Data exchange with Excel

Import a matrix from Excel to MATLAB, add one row with arbitrary numbers at the end of this matrix, and then write this updated matrix back into Excel.

- Importing data from Excel

  variable_name=xlsread('filename', 'tab_name', 'range')
- Exporting data to Excel

  xlswrite('filename', variable_name)
- Use Import Wizard

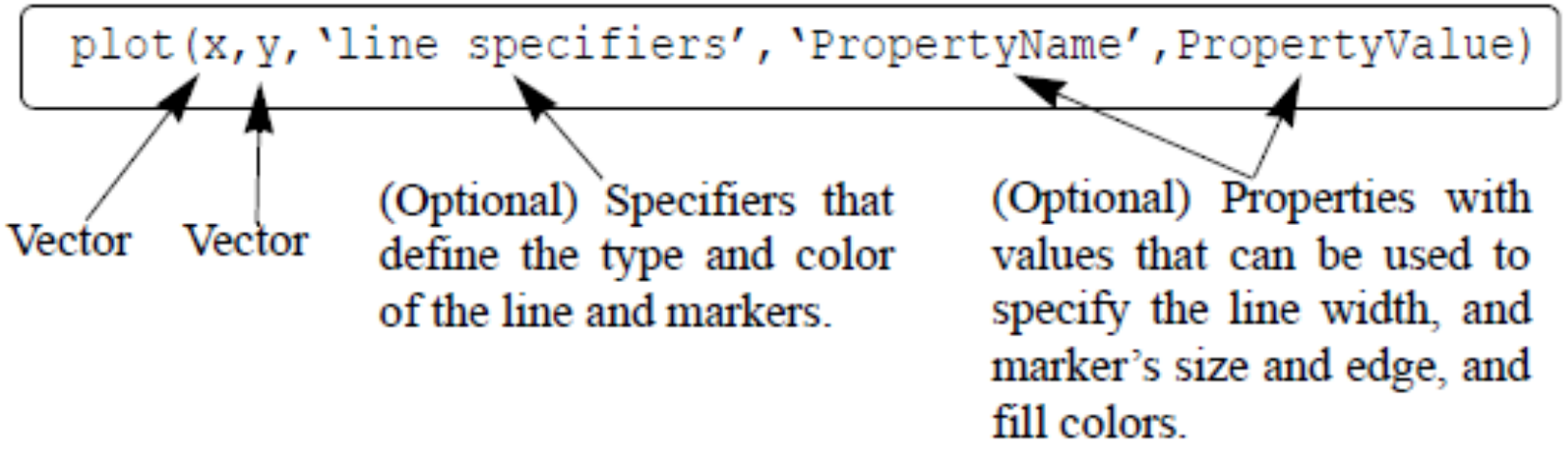
MATLAB Example 4: Save AND Load command

- Use save command to save workspace or data,
- Use load command to retrieve stored workspace or data

```
save file_name OR save('file_name')
save file_name var1 var2
save('file_name','var1','var2')

load file_name OR load('file_name')
load file_name var1 var2
load('file_name','var1','var2')
```

2D Plot command



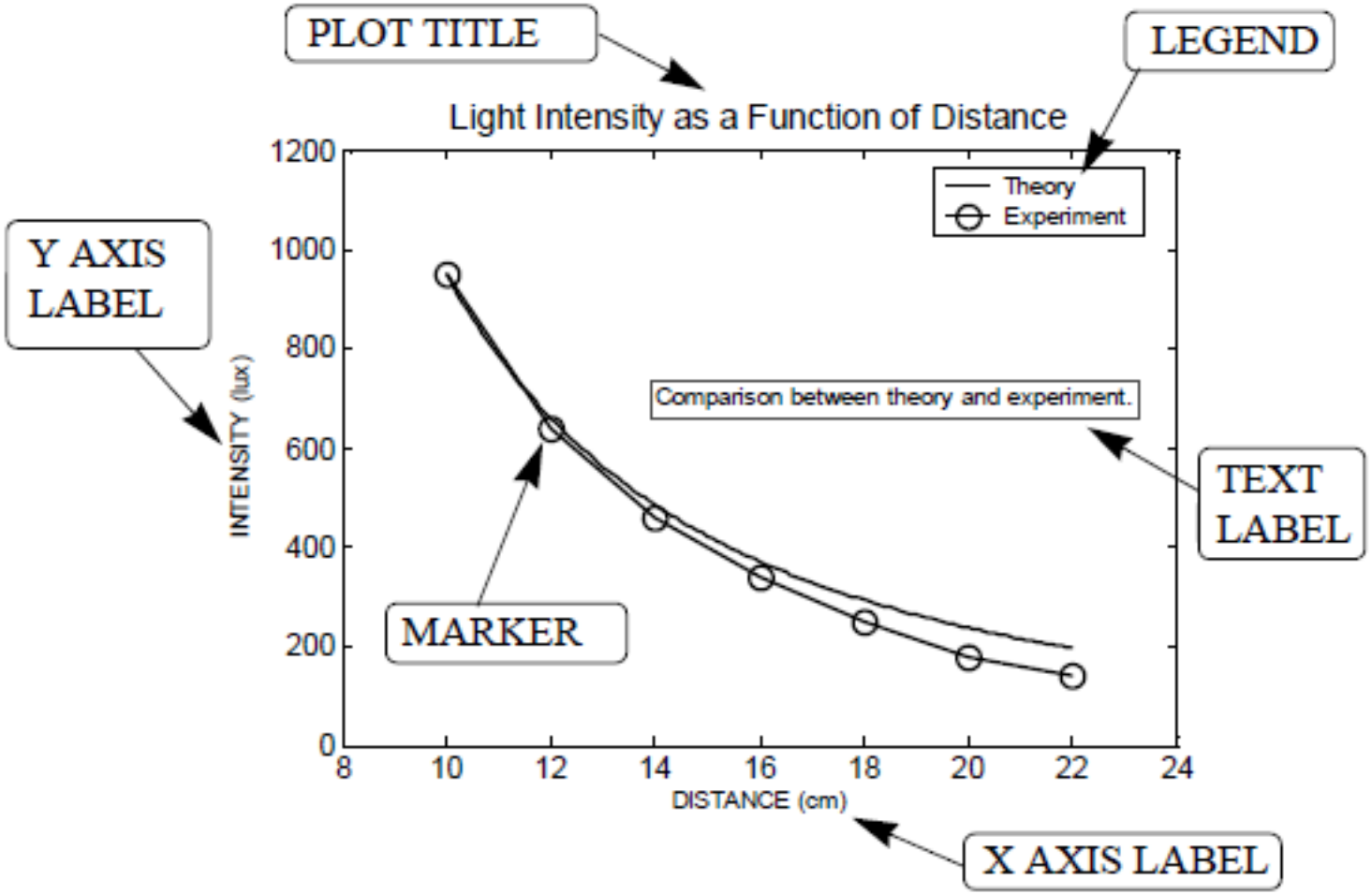


Figure 5-1: Example of a formatted two-dimensional plot.

MATLAB Example 5: 2D Plot

- **Plot** Example:

  Plot a trig function $\cos(x)$, where $0 \leq x \leq 6\pi$

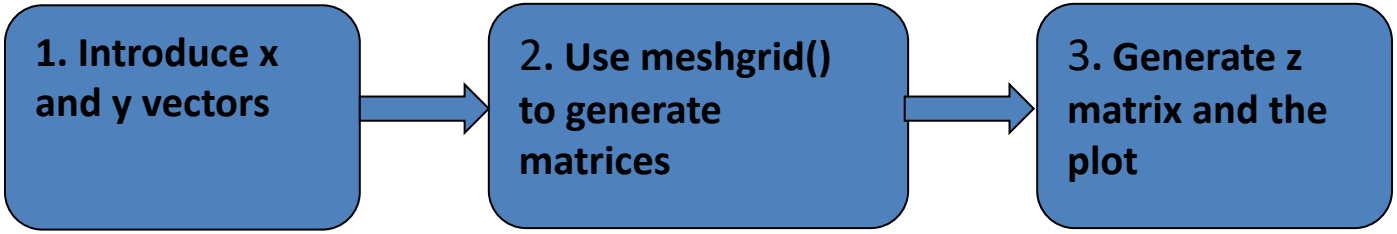
- **Subplot**

MATLAB Example 6: 3D plot

- MATLAB example:

  Plot the surface generated by $z = 2x^2 + 2y^2 - 4$ using `mesh` and `surf` command.

- How to plot a 3D surface



- 3D MATLAB commands:
  - mesh
  - surf
  - contour