Chapter 2: Basic MATLAB (Cont.)

Yunong Zhang (张雨浓)

Email: zhynong@mail.sysu.edu.cn

Displaying Output Data

- Double-precision variable:

  It is not necessary (and, most of the time, impossible) to show the exact value of a variable stored in the computer.

  pi=3.14159265358979...

  pi=3.14 is enough

The default format:

Four digits after the decimal point

x=100.11 → x=100.1100

Displaying Output Data (Cont.)

- Changing the Default Format
- *format* command

x=12.345678901234567

format short: 4 decimal digits 12.3457
format long: 15 decimal digits 12.345678901234567
format short e: 4 decimal digits plus exponent
1.2346e+001
format long e: 15 decimal digits plus exponent
1.234567890123457e+001
format hex: hexadecimal display 4028b0fcd32f707a

hex head screw 六角头螺钉 hex socket screw key 六角凹头螺钉键 hexa 六 hexad 六个一组 hexadecimal 十六进制(的) hexadecimal multiplication 十六进制乘法 hexagon 六角/边形 hexahedron 六面体

Displaying Output Data (Cont.)

- disp function

  ```
  str=['The value of pi is:' num2str(pi)];
  disp(str);
  ```

- fprintf function

  ```
  fprintf(format, data)
  fprintf('The value of pi is %f \n',pi)
  fprintf('The value of pi is %6.2f \n',pi)
  ```

Displaying Output Data (Cont.)

```
>> fprintf('The value of pi is %6.2f \n',pi)
The value of pi is   3.14
>> fprintf('The value of pi is %6.3f \n',pi)
The value of pi is  3.142
>> fprintf('The value of pi is %6.9f \n',pi)
The value of pi is 3.141592654
>> fprintf('The value of pi is %7.9f \n',pi)
The value of pi is 3.141592654
>> fprintf('The value of pi is %1.9f \n',pi)
The value of pi is 3.141592654
>> fprintf('The value of pi is %0.9f \n',pi)
The value of pi is 3.141592654
>> fprintf('The value of pi is %2.16f \n',pi)
The value of pi is 3.1415926535897931
>>
```

Displaying Output Data (Cont.)

- About Special Characters in *fprintf* Format Strings:

Format String	Results
%d	Display value as an integer
%e	Display value in exponential format
%f	Display value in floating point format
%g	Display value in either floating point or exponential format, whichever is shorter
\n	Skip to a new line (=the `enter` key).

Displaying Output Data (Cont.)

```
x=2*(1-2i)^3;
str=['x=' num2str(x)];
disp(str);
fprintf('x=%8.4f \n', x);
```

MATLAB can handle complex numbers!

```
x=-22+4i
x=-22.0000
```

Note: fprintf function only displays the real part of a complex number.

Displaying Output Data (Cont.)

```
>> x=100000
>> fprintf('x=%8.4f \n', x);
x=100000.0000
>> fprintf('x=%2.1f \n', x);
x=100000.0
>> fprintf('x=%1.1f \n', x);
x=100000.0
>> fprintf('x=%.1f \n', x);
x=100000.0
>> fprintf('x=%1f \n', x);
x=100000.000000
```

Data Files

- *save* command:

1) save filename var1 var2 var3

   var1,var2, var3, and so forth, are saved in the file “filename.mat”.

2) save filename

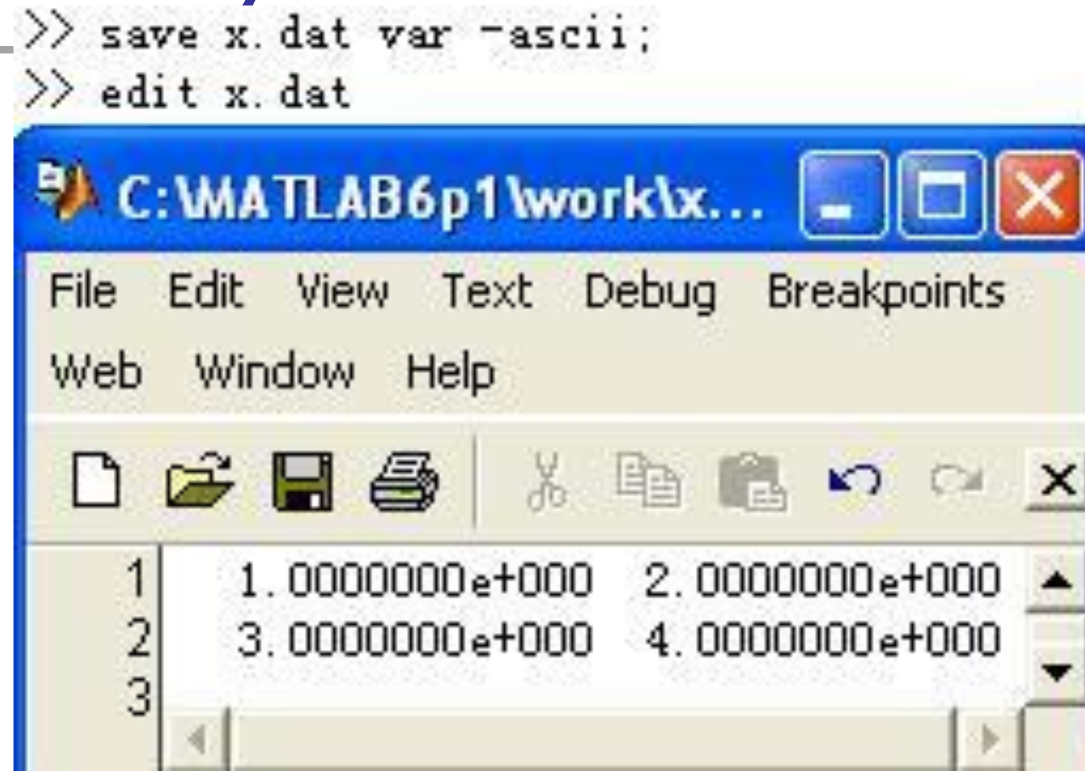
   All variables of the workspace are saved to the file “filename.mat”.

Data Files (Cont.)

- By default, the file name is given the extension "mat".
- Mat-file preserves many details, including the name, type, size, and data value of each variable.
- Mat-file cannot be read by other programs

- save filename var1 var2 var3 -ascii

Data Files (Cont.)



var=[1 2;3 4];

save x.dat var –ascii;

----

x contains the following data:

1.0000000e+000 2.0000000e+000

3.0000000e+000 4.0000000e+000

Data Files (Cont.)

- *load* command

  Expression: *load filename*

  It loads data from a file into the current workspace.

```
var1=1; var2=2; var3=3;
>> save x.mat var1 var2 var3
>>load x.mat
```

[The extension .mat cannot be omitted]

```
>>x
```

Data Files (Cont.)

```
>> var1=1;
var2=2;
var3=3;
save x.mat var1 var2 var3
>> whos
  Name       Size           Bytes  Class

  var1       1x1                8  double array
  var2       1x1                8  double array
  var3       1x1                8  double array

Grand total is 3 elements using 24 bytes

>> clear
>> whos
>> load x.mat
>> whos
  Name       Size           Bytes  Class

  var1       1x1                8  double array
  var2       1x1                8  double array
  var3       1x1                8  double array

Grand total is 3 elements using 24 bytes

>> x
??? Undefined function or variable 'x'.
```

Data Files (Cont.)

\>> load x.mat var1

\>> var1

var1 = 1

```
>> clear
>> whos
>> load x.mat var1
>> whos
  Name      Size           Bytes  Class

  var1      1x1                8  double array

Grand total is 1 elements using 8 bytes
```

Scalar and Array Operations

- Scalar Operations

Operation	Algebraic Form	Matlab Form
Addition	$a+b$	a+b
Subtraction	$a-b$	a-b
Multiplication	$a \times b$	a*b
Division	a/b	a/b
Exponentiation	a^b	a^b

Scalar and Array Operations (Cont.)

Operation	Matlab Form	Description
Array Addition	A+B	A and B have the same dim.
Array Subtraction	A-B	A and B have the same dim.
Array Multiplication	A.*B	element-by-element multiplication A(i,j)*B(i,j), dim(A)=dim(B)
Matrix Multiplication	A*B	num. col. of A=num. row. of B
Array Right Division	A./B	element-by-element division A(i,j)/B(i,j), dim(A)=dim(B), and ?
Matrix Right Division	A/B	defined by A*inv(B)

Scalar and Array Operations (Cont.)

Array Left Division A.\B element-by-element division

B(i,j)/A(i,j), dim(A)=dim(B), and ?

Marry Left Division A\B defined by inv(A)*B

Array Exponentiation A.^B

element-by-element exponentiation. A(i,j)^B(i,j), dim(A)=dim(B)

A+c, A-c,

Vague usage!

A*c, A/c

c./A

Vague usage!

where A is a matrix, and c is a scalar.

Scalar and Array Operations (Cont.)

Example:

A=[1 0;
2 1];

B=[-1 2;
0 1];

C=[3;2];

d=5;

What is the result of the following expressions?

(1) A+B (2) A.*B (3) A*B (4) A*C

(5) A+C (6) A+d (7) A.*d (8) A*d

矩阵大写，标量小写，向量小写黑体/粗体

Scalar and Array Operations (Cont.)

(1) A+B=[0 2; 2 2];

(2) A.*B=[-1 0; 0 1];

(3) A*B=[-1 2; -2 5];

(4) A*C=[3;8];

(5) illegal;

(6) A+d=[6 5;7 6];

(7) A.*d=[5 0; 10 5];

(8) A*d=[5 0;10 5];

Hierarchy of Operations

Operations' Precedence

1) The contents of all parentheses are evaluated, starting from the innermost parentheses and working outward.
2) All exponentials are evaluated, working from left to right.
3) All multiplications and divisions are evaluated, from left to right.
4) All additions and subtractions are evaluated, from left to right.

```
>> 7*3-5^3/7

ans =

    3.1429

>> 7*3-5^(3/7)

ans =

   19.0068
```

Built-in Functions

abs(x): $|x|$

cos(x): cosx, with x in rad.

exp(x): e^x

log(x): $\log_e^x = \ln x$

[value, index]=max(x): return the maximum value in x, and the location of that value.

[value, index]=min(x): return the minimum value in x, and the location of that value.

Built-in Functions (Cont.)

sqrt(x): square root of x

tan(x): tanx, with x in rad.

round(x): rounds x to the nearest integer.

int2str(x): converts an integer x into a character string

num2str(x): converts x into a character string representing the number with a decimal point

str2num(s): converts character string s into a number

Built-in Functions (Cont.)

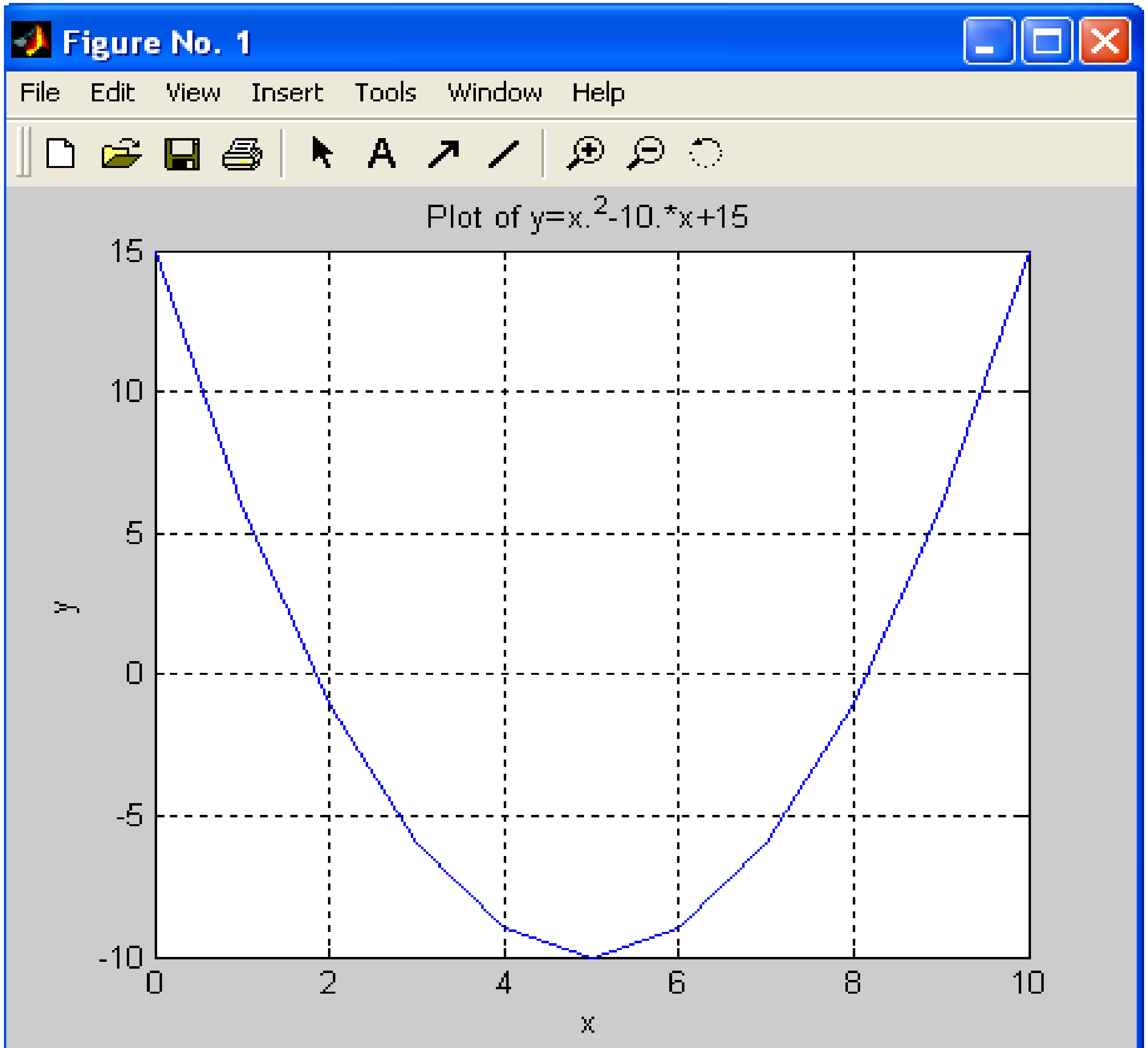
```
>> int2str(2)
ans = 2
>> int2str(200)
ans = 200
>> int2str(20)
ans = 20

>> str2num('3.1415926')
ans = 3.1416
>> format long
>> str2num('3.1415926')
ans = 3.14159260000000
>> str2num('who are u?')
ans = []
```

Drawing Simple XY Plots

```
x=0:1:10;
y=x.^2-10.*x+15;
plot(x,y);
title('Plot of y=x.^2-10.*x+15');
xlabel('x');
ylabel('y');
grid on;
```

Drawing Simple XY Plots (Cont.)



Printing a Plot

- *print* command

  print: Print the current figure to the computer-connected printer

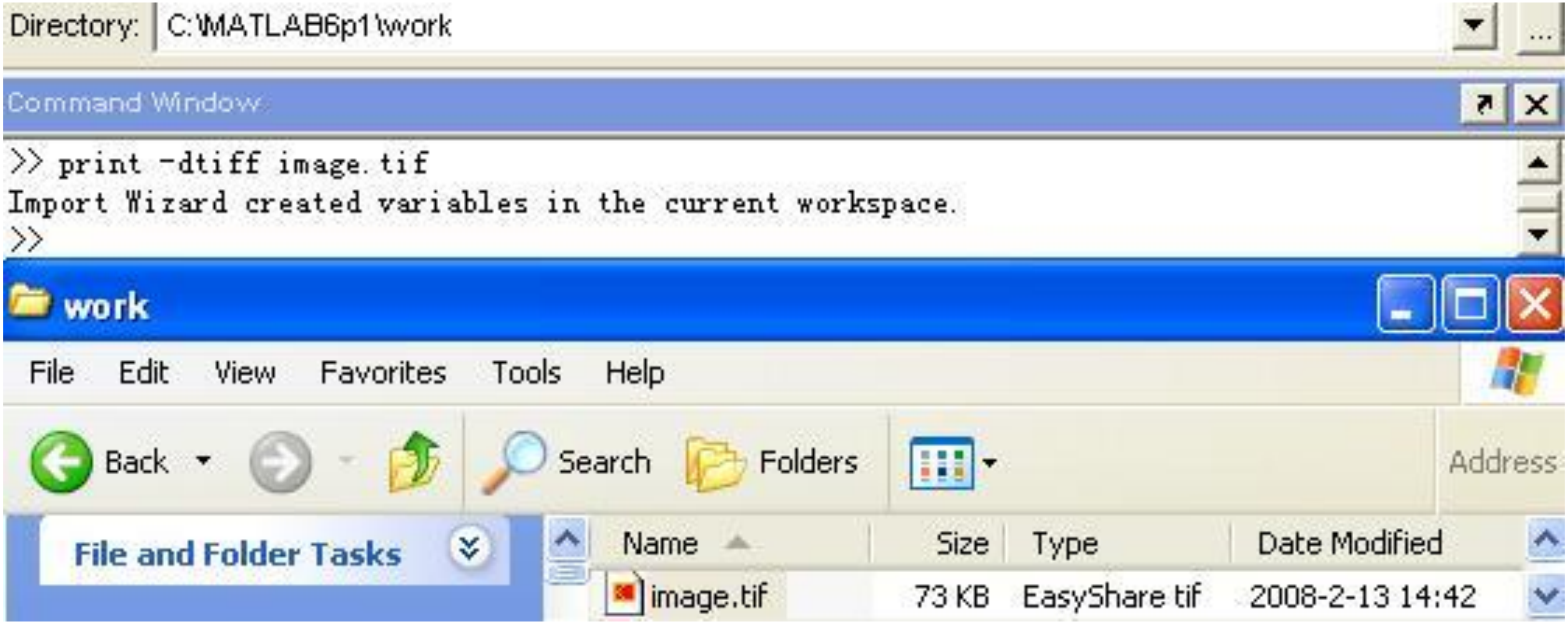
  print filename: print/save a copy of the current figure to the specified name

Specify the format of the output:

print -dtiff image.tif

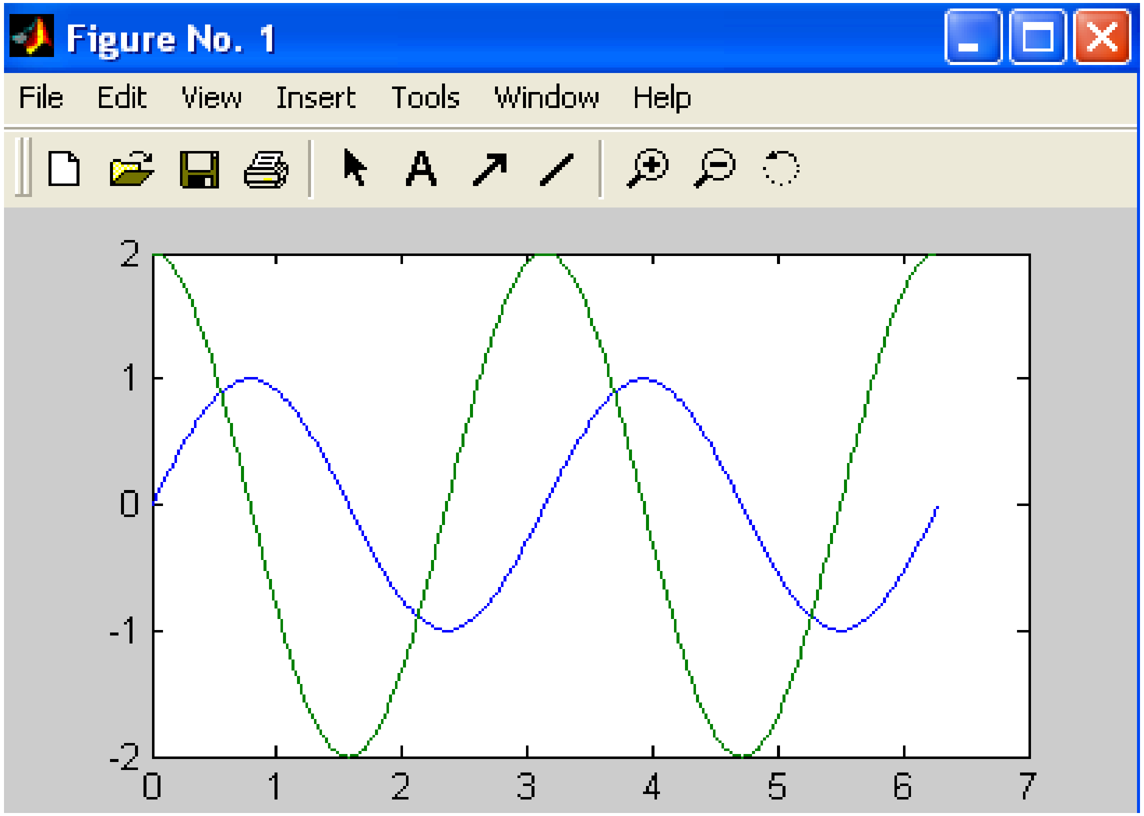
It creates a TIFF image of the current figure and store it in the file `image.tif`.

Printing a Plot (Cont.)



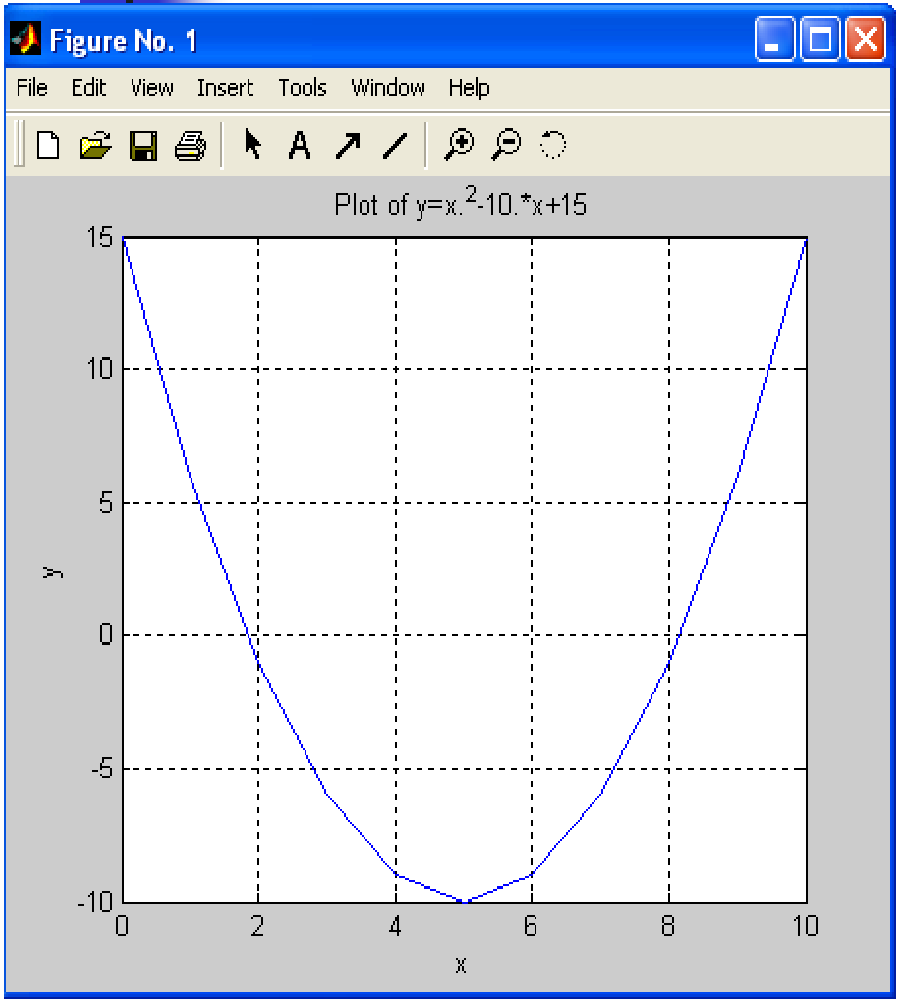
Multiple Plots

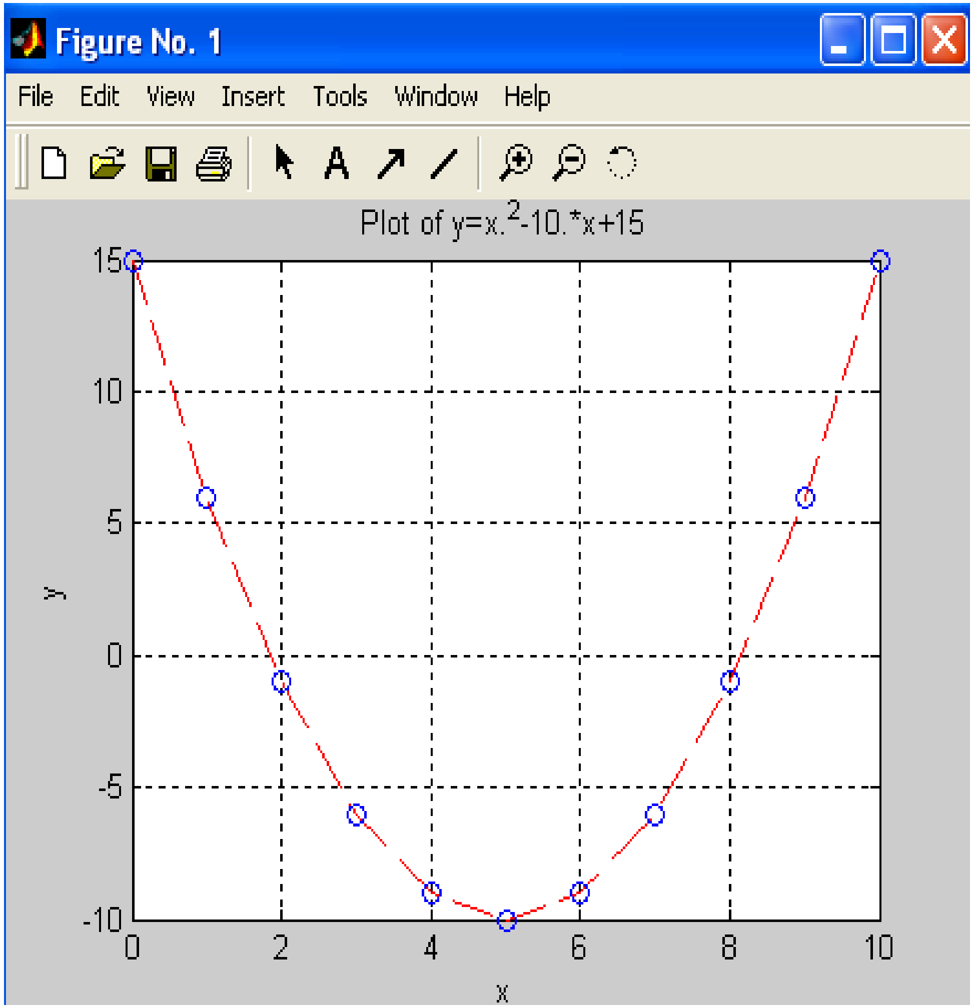
```
x=0:pi/100:2*pi;
y1=sin(2*x);
y2=2*cos(2*x);
plot(x,y1,x,y2);
```



Note that the following two commands are incorrect:
plot(x,y1,y2), plot(x,y1;x,y2)

Configurations of Plot





Configurations of Plot (Cont.)

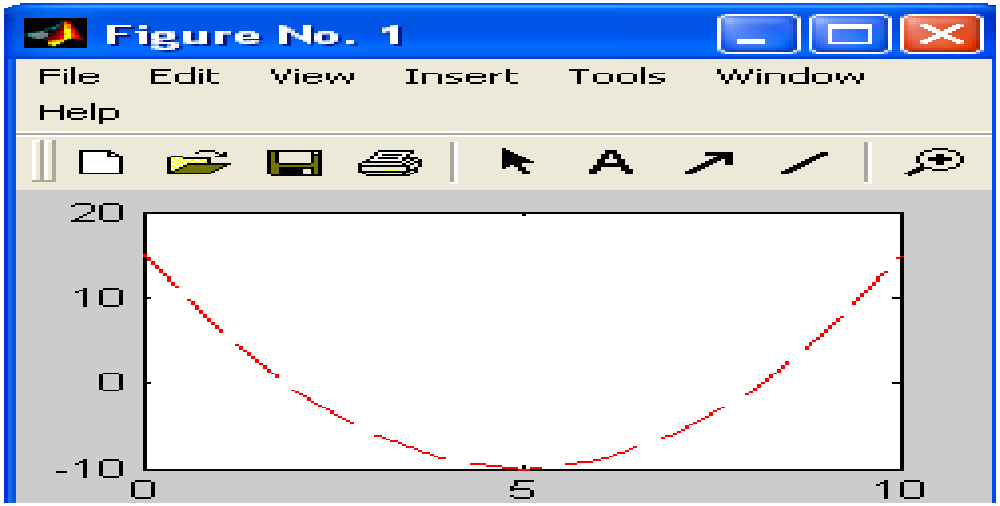
```
x=0:1:10;
y=x.^2-10.*x+15;
plot(x,y,'r--',x,y,'bo');
title('Plot of y=x.^2 ...10.*x+15');
xlabel('x');
ylabel('y');
grid on;
```

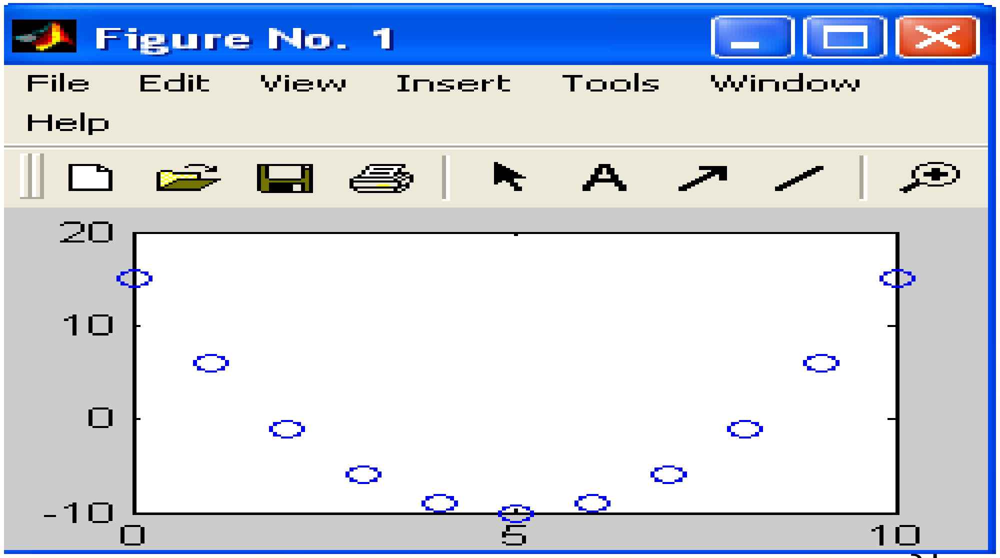
Configurations of Plot (Cont.)

plot(x,y,'r--',x,y,'bo');

=

plot(x,y,'r- -');
+ hold on;
+ plot(x,y,'bo);





Configurations of Plot (Cont.)

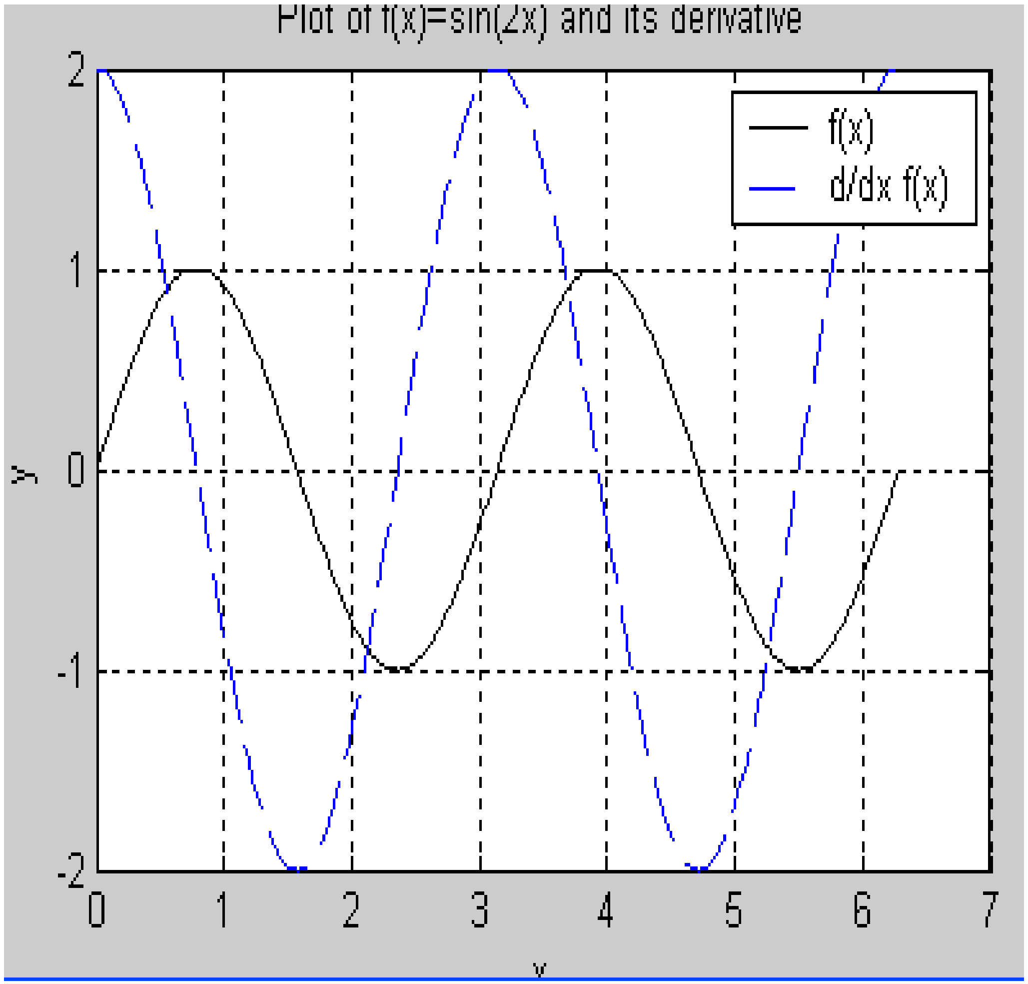
Color		Marker style		Line sytle	
y	yellow	.	point	-	**solid**
m	magenta洋红	o	circle	:	dotted
c	cyan青色	x	x-mark	-.	dash-dotted
r	red	+	plus	--	dashed
g	green	*	star		
b	blue	s	square		
w	white	d	diamond		
k	black	^	triangle (up)		

cyan
['saɪən]
n. 蓝绿色，青色

magenta
[məˈdʒentə]
红紫色，洋红

Configurations of Plot (Cont.)

```
x=0:pi/100:2*pi;
y1=sin(2*x);
y2=2*cos(2*x);
plot(x,y1,'k-',x,y2,'b--');
title('Plot of f(x)=sin(2x)...
and its derivative');
xlabel('x');
ylabel('y');
legend('f(x)','d/dx f(x)');
grid on;
```



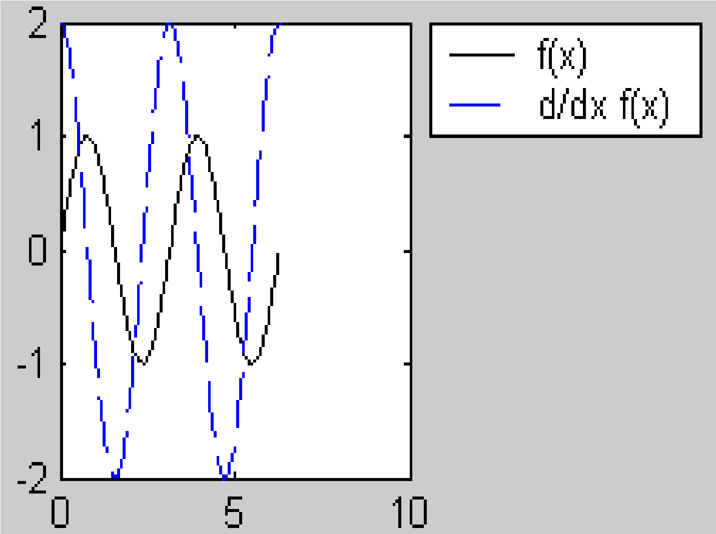
The other major difference between university-level and high-school-level knowledge!!

Configurations of Plot (Cont.)

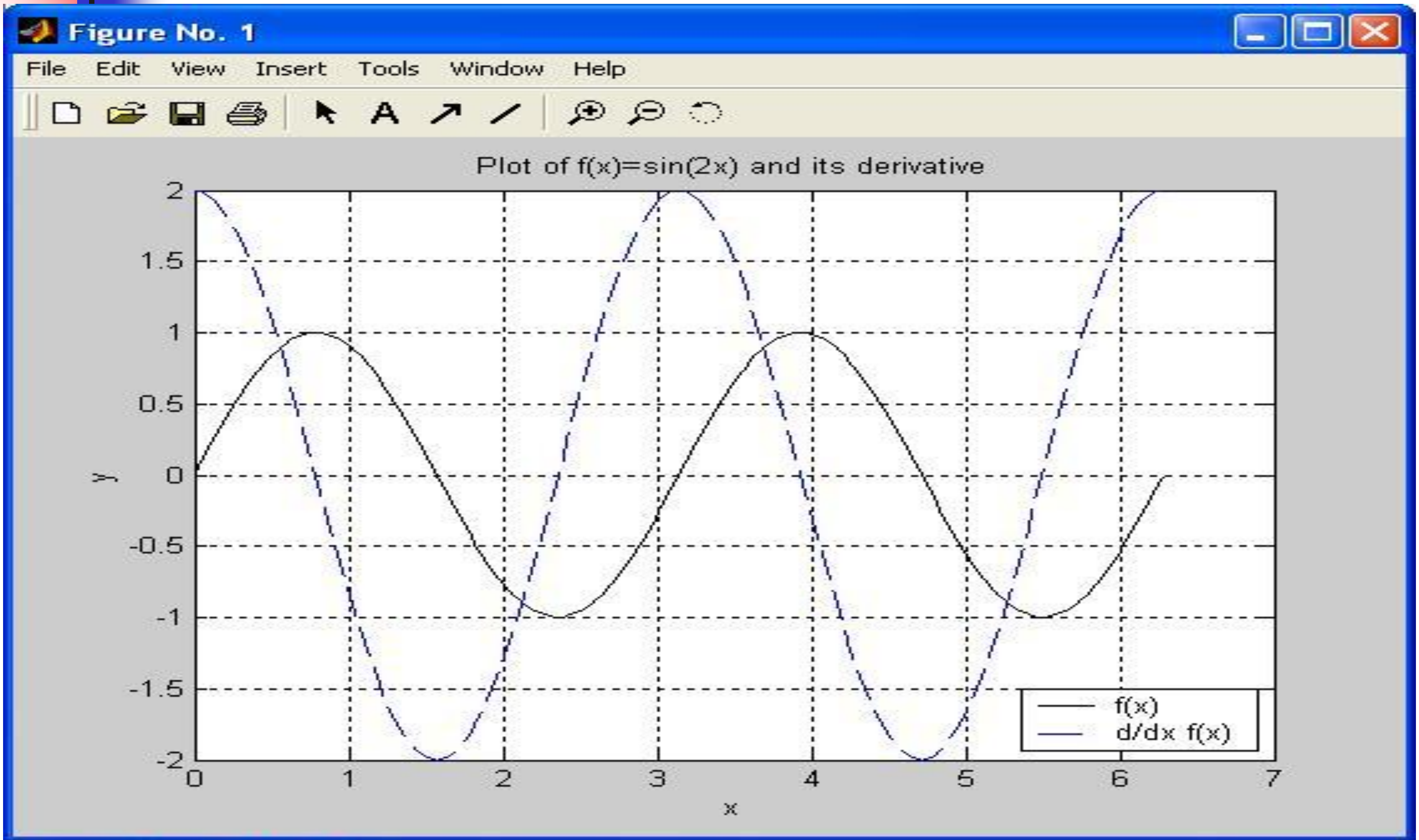
- legend ('string1','string2',..., pos)

  pos is an integer specifying where to place the legend.

Value	Description
0	automatic set -- automatic "best" placement
1	Upper right-hand corner (default)
2	Upper left-hand corner
3	Lower left-hand corner
4	Lower right-hand corner
-1,>4	To the right of the plot



Configurations of Plot (Cont.)




```
legend('f(x)','d/dx f(x)',0);
```

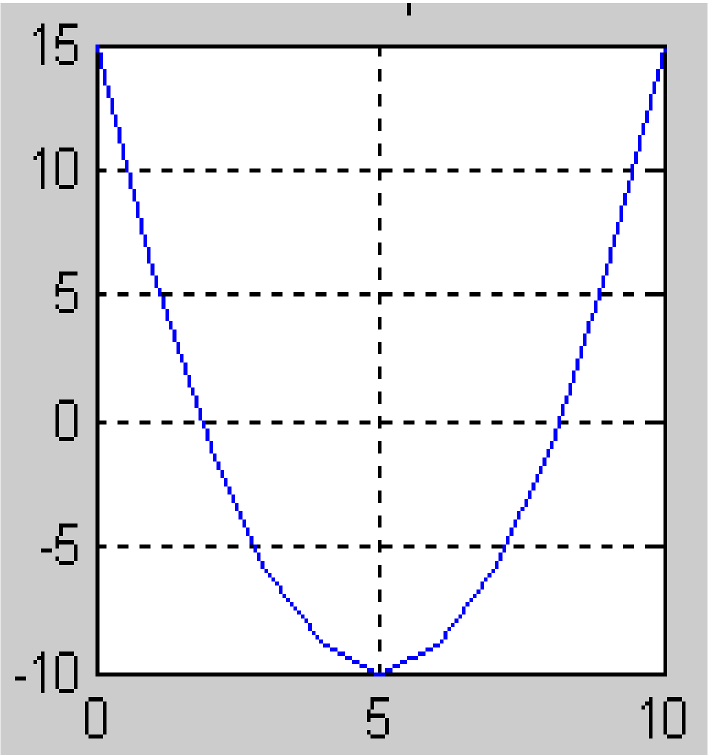
Configurations of Plot (Cont.)

- plot: plots both x and y on linear axes
- semilogx: plots x on logarithmic axes and y on linear axes
- semilogy: plots x on linear axes and y on logarithmic axes
- loglog: plots both x and y on logarithmic axes

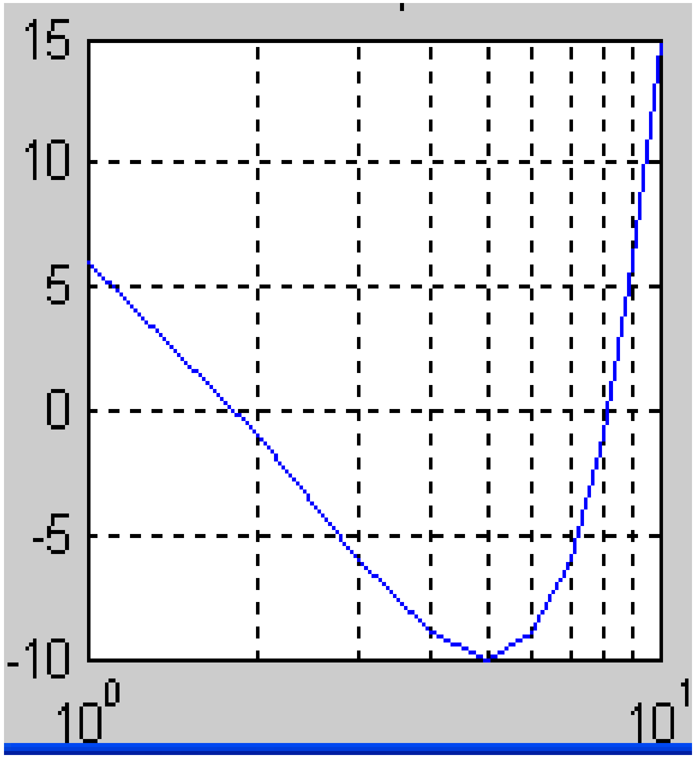
Configurations of Plot (Cont.)

```
x=0:1:10;
y=x.^2-10.*x+15;
```

plot(x,y);



semilogx(x,y);

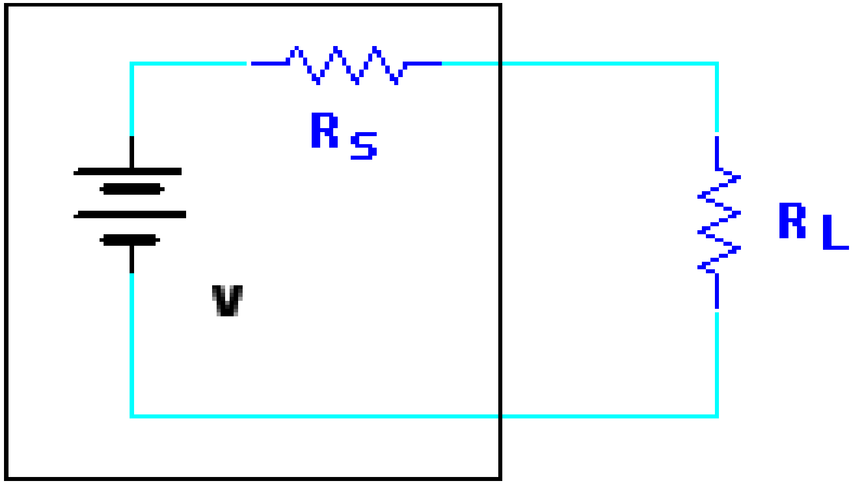


semilogy(x,y);

loglog(x,y)

Additional Example

Voltage source: V=120V
Internal resistance:
R_S=50 ohms
load of resistance: R_L



Q1: Find the max. possible power supplied to the load
Q2: Plot the power supplied to the load as a function of the load resistance R_L

Additional Example (Cont.)

Power supplied to the load given by

$$P_L = I^2 R_L$$

where the current passing through the load is

$$I = \frac{V}{R_S + R_L}.$$

So, $P_L = I^2 R_L = \dfrac{V^2 R_L}{(R_S + R_L)^2}$

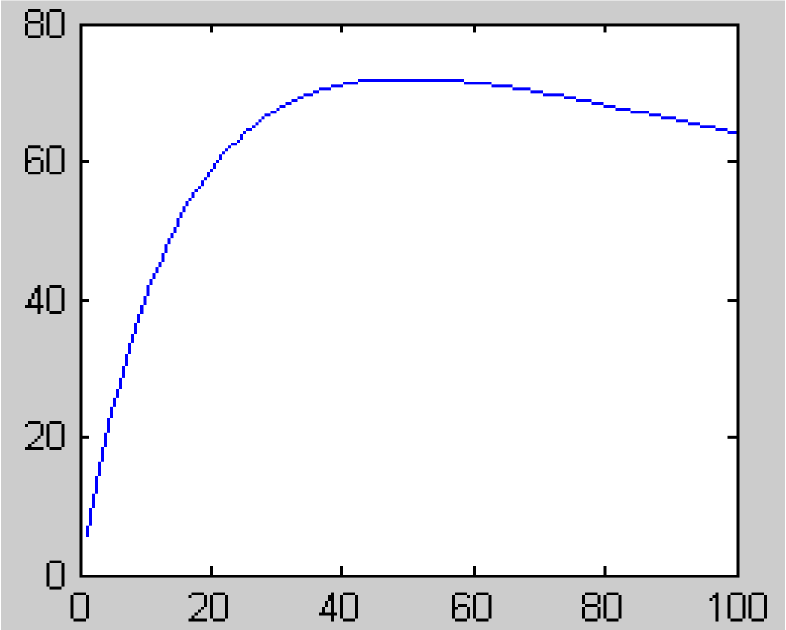
Given V=120 and R_S=50, to maximize P_L, R_L=?

Additional Example (Cont.)

```
% definitions of variables:
%   curflow: current flow to the load;
%   powersup: power supplied to the load;
%   resiload: resistance of the load;
%   resisource: internal resistance of the power source;
%   volts: voltage of the power source.

volts=120;
resisource =50;

resiload=1:100;
curflow =volts./(resisource+resiload);
powersup=(curflow.^2).* resiload;
plot(resiload, powersup);
```



$R_L = R_S = 50$

Sincere Thanks!

- Using this group of PPTs, please read

- [1] Yunong Zhang, Weimu Ma, Xiao-Dong Li, Hong-Zhou Tan, Ke Chen, MATLAB Simulink modeling and simulation of LVI-based primal-dual neural network for solving linear and quadratic programs, Neurocomputing 72 (2009) 1679-1687

- [2] Yunong Zhang, Chenfu Yi, Weimu Ma, Simulation and verification of Zhang neural network for online time-varying matrix inversion, Simulation Modelling Practice and Theory 17 (2009) 1603-1617