

CHƯƠNG 1: CƠ SỞ MATLAB

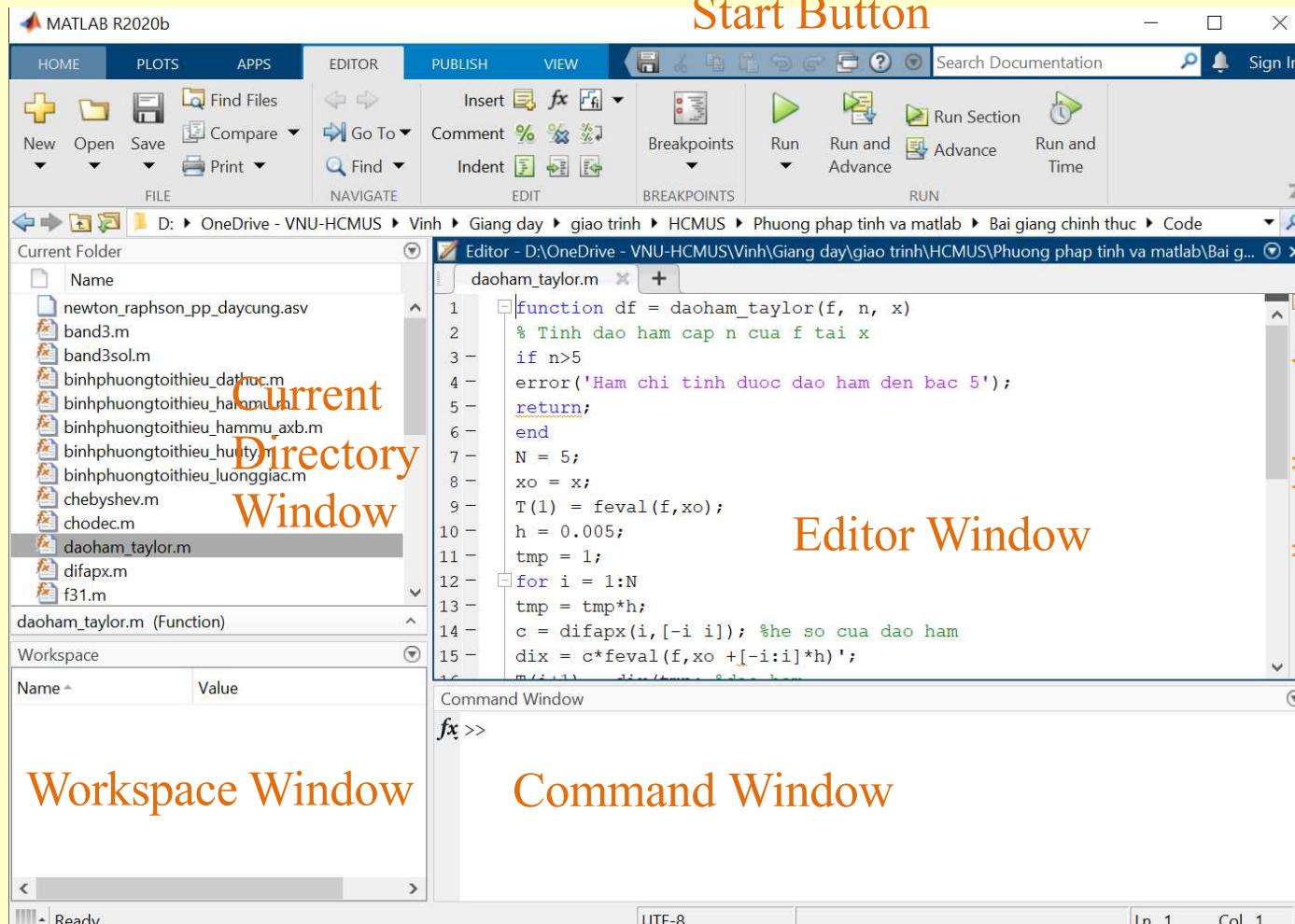
Tài liệu tham khảo:

Essential MATLAB for Engineers and Scientists

(Brian D.Hahn, Daniel T.Valentine)

=> Cung cấp cái nhìn tổng quan về Matlab

MATLAB desktop



Command Window

Editor Window

Workspace Window

Current Directory Window

Start Button

Các cửa sổ Matlab

- Cửa sổ trợ giúp (Help window)
- Cửa sổ nhập lệnh (Command window)
- Cửa sổ không gian làm việc (Workspace window)
- Cửa sổ quá trình lệnh (Command History window- lịch sử lệnh)
- Cửa sổ biên tập mảng, vectơ, ma trận (Array editor window)
- Cửa sổ địa chỉ thư mục hiện thời (Current directory window)

Command Window

- Command Window : cửa sổ lệnh.
- Trong cửa sổ này nhập lệnh, hàm,.. sau dấu >> rồi <Enter>, Matlab sẽ thực hiện lệnh và hiển thị kết quả.
- Một số lệnh phím thông dụng:
 - ↑ (up arrow) trả về lệnh cuối cùng được nhập
 - clc – xoá màn hình
 - whos – liệt kê danh sách biến
 - clear – xoá biến

Ví dụ

Gõ vào

>> 2+3 <Enter>

trong cửa sổ lệnh gõ vào

>> clc <Enter> xóa cửa sổ command window

>> whos <Enter> liệt kê biến

Quan sát các thay đổi ở các cửa sổ.

Command History Window

- Command History Window (cửa sổ lịch sử lệnh): lưu lại tất cả các lệnh được nhập vào trong Matlab. thực hiện các dòng lệnh đơn

Vd: dùng Command History Window để nhập lại lệnh $2+3$

- ⑦ Copy and paste or double click on $2+3$
- Dùng phím “Shift” để chọn nhiều dòng lệnh

Một số ví dụ MATLAB

```
>> 3-2      <Enter>
>> 3*2      <Enter>
>> 3/2      <Enter>
>> 3\2 ?    <Enter>
>> 3^2      <Enter>
>> 2/0 ?    <Enter>
>> 0/2      <Enter>
>> 3*Inf    <Enter>
```

```
>> a = 3    <Enter>
>> b = 2    <Enter>
>> a - b    <Enter>
>> a / b    <Enter>
>> a^2      <Enter>
>> c = a * b    <Enter>
>> d = c^(b+1)    <Enter>
>> who
```

Một số ví dụ MATLAB

```
>> clear; clc      <Enter>
>> whos              <Enter>
>> a = 3;            <Enter>
>> b = 2;            <Enter>
>> c = a * b;        <Enter>
>> d = c^(b+1);     <Enter>
>> who               <Enter>
>> % a, b, c, d are in workspace <Enter>
>> a, b, c, d       <Enter>
```

Vẽ y theo x

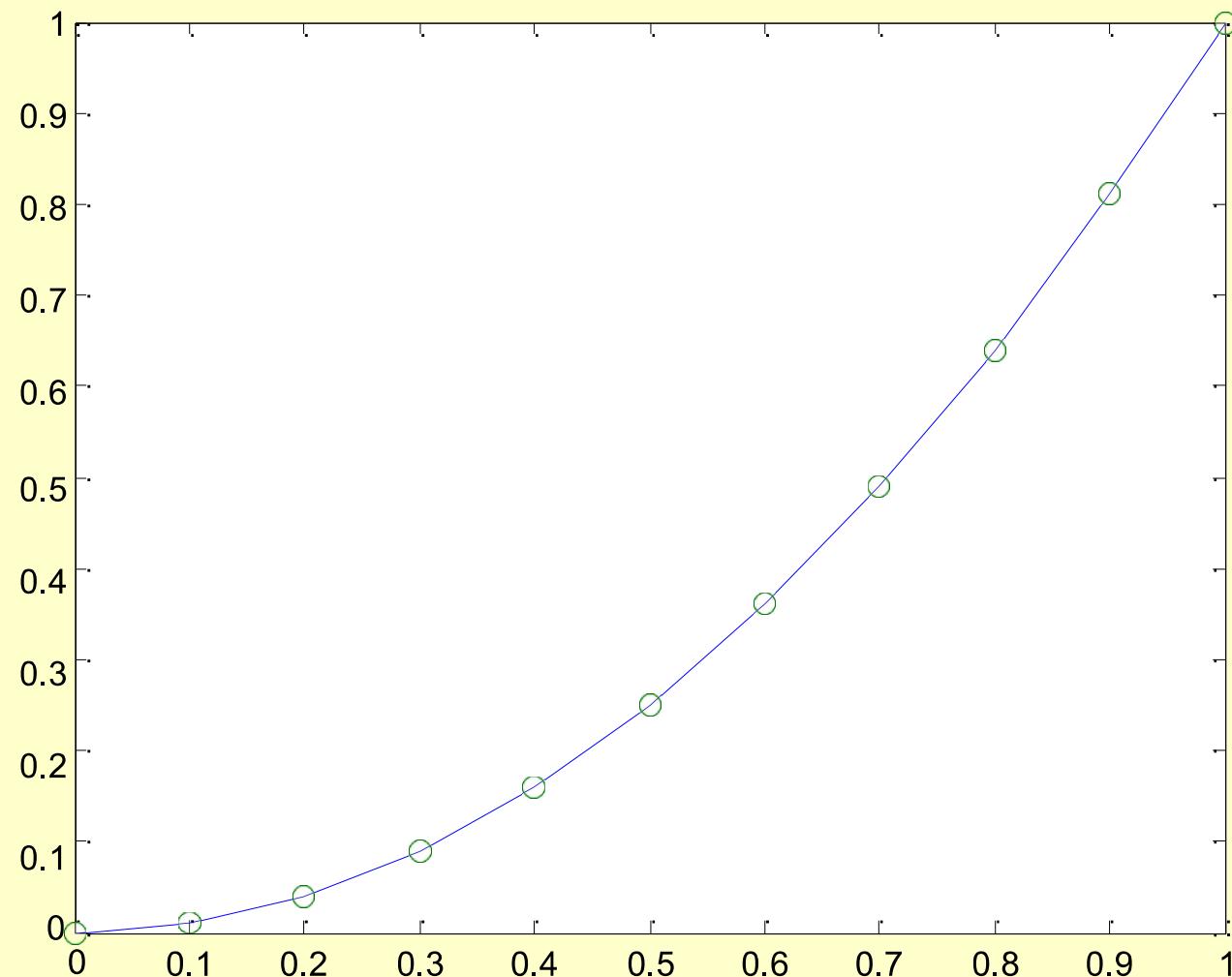
□ Xét đoạn chương trình:

```
>> clear; clc      <Enter>
>> x = 0:0.1:1;   <Enter>
>> y = x.^2;      <Enter>
>> whos            <Enter>
>> plot(x,y,x,y,'o')    <Enter>
>> disp(' '),disp('..... x ..... y .....'),disp([x' y'])  <Enter>
>> x    <Enter>
>> y    <Enter>
>> % x and y are 1-by-11 arrays of numbers!
```

Kết quả

```
x =  
Columns 1 through 9  
0 0.1000 0.2000 0.3000 0.4000 0.5000 0.6000 0.7000 0.8000  
Columns 10 through 11  
0.9000 1.0000  
  
y =  
Columns 1 through 9  
0 0.0100 0.0400 0.0900 0.1600 0.2500 0.3600 0.4900 0.6400  
Columns 10 through 11  
0.8100 1.0000  
>> plot(x,y,x,y,'o')
```

Kết quả



Một chương trình đơn giản

- Thể tích một hình nón:

Volume = (pi.*r.^2.*h)./3

radius = 6 inches

height = 12 inches

- Trong cửa sổ lệnh của Matlab:

```
>> clear; clc    <Enter>
>> r = 6        <Enter>
>> h = 12       <Enter>
>> v = (pi.*r.^2.*h)./3   <Enter>
>> whos        <Enter>
```

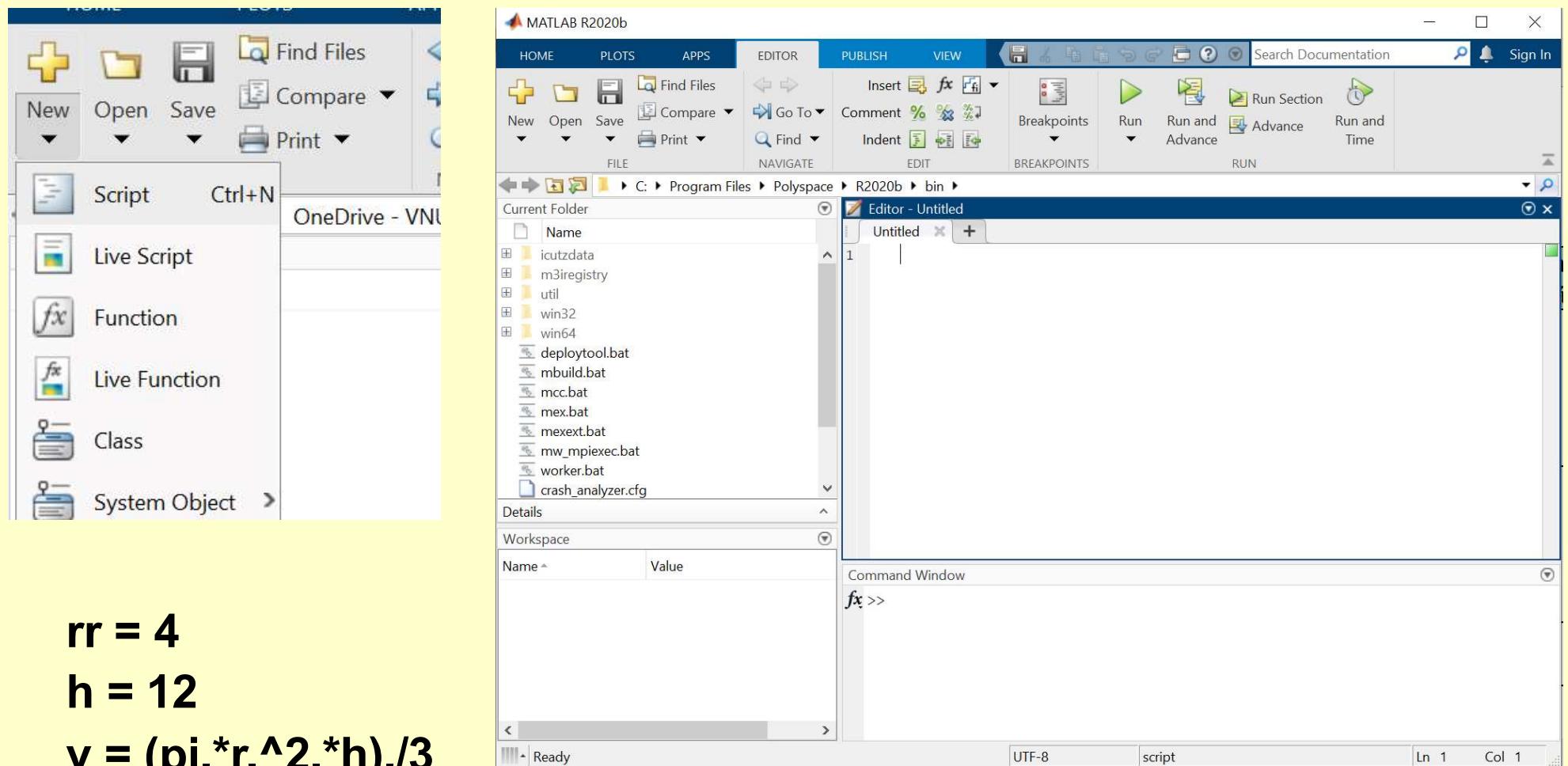
M-Files

- M-file trong MATLAB có dạng giống như txt-file trong Microsoft Notepad.
- M-file có thể được tạo và soạn thảo bằng MATLAB text editor.
- M-files:
 - Chứa tập hợp các lệnh của Matlab (**Scripts**)
 - ✓ Ưu điểm:
 - ✓ Khi thực hiện lại nhóm lệnh, chỉ cần gọi m-file
 - ✓ Mở lại và chỉnh sửa chương trình khi cần
 - ✓ Dễ gỡ rối chương trình
 - Hàm (Functions): có thể có input và output

Chú thích trong chương trình

- Dùng toán tử chú thích % ở đầu dòng
- MATLAB sẽ không thi hành lệnh sau %
- Dùng chú thích để:
 - Làm cho chương trình tường minh
 - Tạo thuận tiện cho việc chỉnh sửa chương trình sau này
 - %{ Mở đầu đoạn chú thích
 - %} Kết thúc đoạn chú thích

Ví dụ M-File



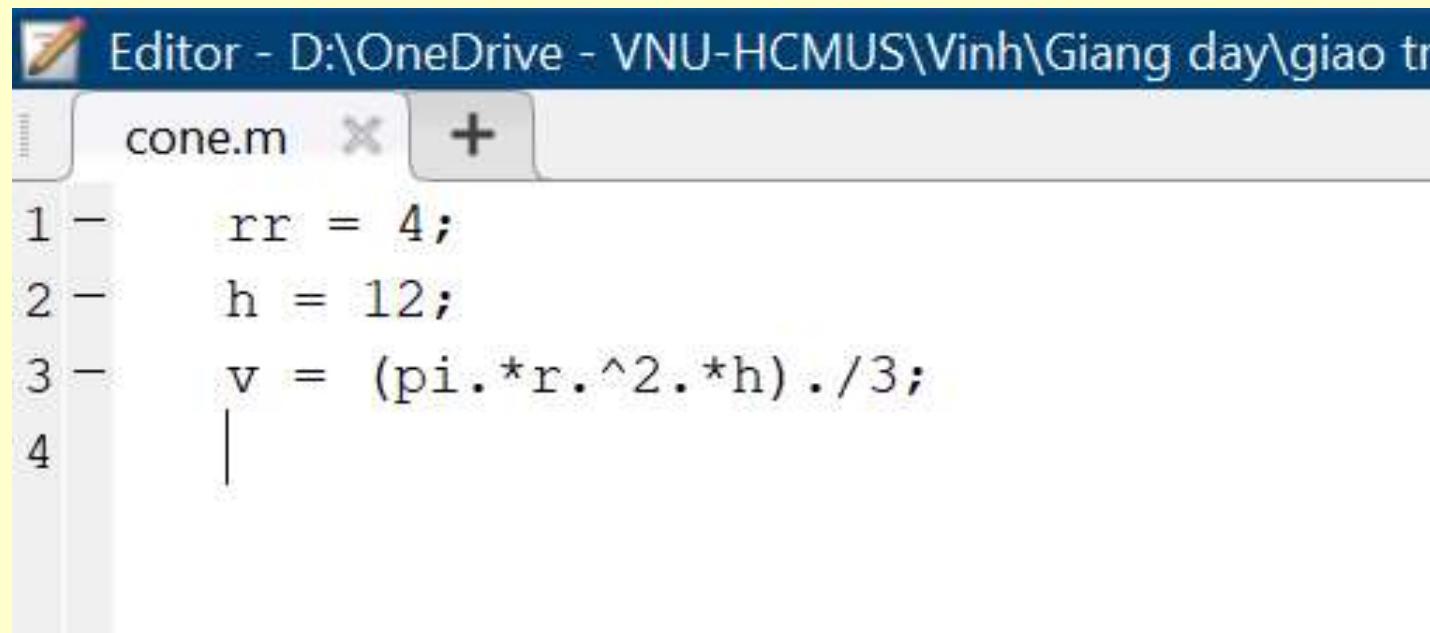
$$rr = 4$$

$$h = 12$$

$$v = (\pi \cdot r^2 \cdot h) / 3$$

- Lưu lại với tên file cone.m.

cone.m



The screenshot shows a MATLAB code editor window titled "Editor - D:\OneDrive - VNU-HCMUS\Vinh\Giang day\giao tri". The current file is "cone.m". The code contains four numbered lines:

```
1 - rr = 4;
2 - h = 12;
3 - v = (pi.*r.^2.*h)./3;
4 - |
```

Thực thi M-file

- Trong cửa sổ lệnh nhập vào:
 >> cone <Enter>
- Kết quả =?
- Sửa chương trình và chạy lại
- Thay giá trị height thành 24, lưu và chạy lại

BIẾN, MẢNG



Các lệnh Matlab

- Các lệnh MATLAB thường có dạng:

variable = expression

hoặc đơn giản

expression

variable: tên biến,

expression: biểu thức.

Quy tắc đặt tên biến

- Bắt đầu bằng chữ
- Có thể gồm chữ, số và gạch dưới (_)
- Không có khoảng trắng
- Chỉ có 63 ký tự đầu là có nghĩa, các ký tự khác phía sau bị cắt cụt
- Phân biệt chữ hoa, chữ thường (a và A khác nhau)
- Biến đặc biệt có tên là ans (answer), được tự động tạo ra nếu khuyết tên biến trong câu lệnh

Biến nào dưới đây hợp lệ?

- 12oclockRock
- tertiarySector
- blue cows
- Eiffel65
- red_bananas
- This_Variable_Name_Is_Quite_Possibly_Too_Long_To_Be_Considered_Good_Practice_However_It_Will_Work % (the green part is not part of the recognized name)

Một số thói quen khi đặt tên biến

- Viết liền hay cách bằng dấu “_”:
 - **lowerCamelCase**
 - **UpperCamelCase**
 - **underscore_convention**
- Nếu biến là hằng thường dùng chữ HOA
 - **CONSTANT**
- Tuỳ ở thói quen của người lập trình
- Tránh đặt tên biến trùng với tên hàm có sẵn của Matlab hay các từ khoá. Ở ví dụ sau khi gán giá trị cho biến **sqrt**, ta không dùng được hàm **sqrt** nữa.
- Tra cứu lệnh **help**

Một số thói quen khi đặt tên biến

□ Biến dài hơn 1 dòng

Final_Answer = BigMatrix(row_indices,column_indices) + ...

Another_vector*SomethingElse;

Mission = ['DSTO"s objective is to give advice that' ...

'is professional, impartial and informed on the' ...

'application of science and technology that is best' ...

'suited to Australia"s defence and security needs.'];

Keywords

```
>> clear
>> sqrt(9)

ans =
3

>> sqrt=1

sqrt =
1

>> sqrt(9)
??? Index exceeds matrix dimensions.
```

Tránh keywords, tên hàm,
hàng đã được định nghĩa sẵn

```
>> iskeyword
ans =

'break'
'case'
'catch'
'classdef'
'continue'
'else'
'elseif'
'end'
'for'
'function'
'global'
'if'
'otherwise'
'parfor'
'persistent'
'return'
'spmd'
'switch'
'try'
'while'

>> else=1
??? else=1
|
Error: Illegal use of reserved keyword "else".
```

Các hàm có sẵn của Matlab có thể tra cứu trong HELP

The screenshot shows the MATLAB Help browser interface. The title bar reads "Exponents and Logarithms". The main content area is titled "Documentation" and features a search bar labeled "Search Help". Below the search bar are tabs for "All", "Examples", and "Functions". The "Functions" tab is selected, displaying a table of functions related to exponents and logarithms. The table has two columns: the function name and its description. The functions listed are: exp, log, log10, log1p, log2, nextpow2, nthroot, pow2, reallog, realpow, realsqrt, and sqrt.

Function	Description
exp	Exponential
expm1	Compute $\exp(x)-1$ accurately for small values of x
log	Natural logarithm
log10	Common logarithm (base 10)
log1p	Compute $\log(1+x)$ accurately for small values of x
log2	Base 2 logarithm and floating-point number dissection
nextpow2	Exponent of next higher power of 2
nthroot	Real n th root of real numbers
pow2	Base 2 power and scale floating-point numbers
reallog	Natural logarithm for nonnegative real arrays
realpow	Array power for real-only output
realsqrt	Square root for nonnegative real arrays
sqrt	Square root

Các hàm có sẵn của Matlab có thể tra cứu trong HELP

The screenshot shows the MATLAB Documentation Center interface. The title bar says "Documentation". The left sidebar has a "CONTENTS" section with links to Documentation Home, MATLAB, Mathematics, Elementary Math, and Exponents and Logarithms. Under the "exp" heading, there are links to Syntax, Description, Examples, and Extended Capabilities, along with a "See Also" section. The main content area shows the "exp" function page. It includes a search bar at the top right. The "exp" function is described as "Exponential". The "Syntax" section shows the command `Y = exp(X)`. The "Description" section states that `Y = exp(X)` returns the exponential e^x for each element in array x . For complex elements $z = x + iy$, it returns the complex exponential $e^z = e^x(\cos y + i \sin y)$. The "Examples" section contains a "Numeric Representation of e" subsection with a live script example. The script shows the command `exp(1)` and the output `ans = 2.7183`. A blue button labeled "Open Live Script" is visible. The URL at the bottom is `file:///C:/Program%20Files/Polyspace/R2020b/help/documentation-center.html`.

Các hàm có sẵn của Matlab có thể tra cứu trong HELP

```
>> a=100  
  
a =  
  
100  
  
>> b=1.00001  
  
b =  
  
1.0000  
  
>> whos  
Name      Size            Bytes  Class       Attributes  
a          1x1              8  double  
b          1x1              8  double
```

Mặc định là số có độ chính xác kép (64 bit =8 bytes)

Ví dụ

```
>> a=100
a =
100

>> b=[1 2 3 4]
b =
1 2 3 4

>> c=10.111
```

Workspace

Name	Value	Size	Min	Max
a	100	1x1	100	100
b	[1,2,3,4,5]	1x5	1	5
c	10.1111	1x1	10.1111	10.1111

Details

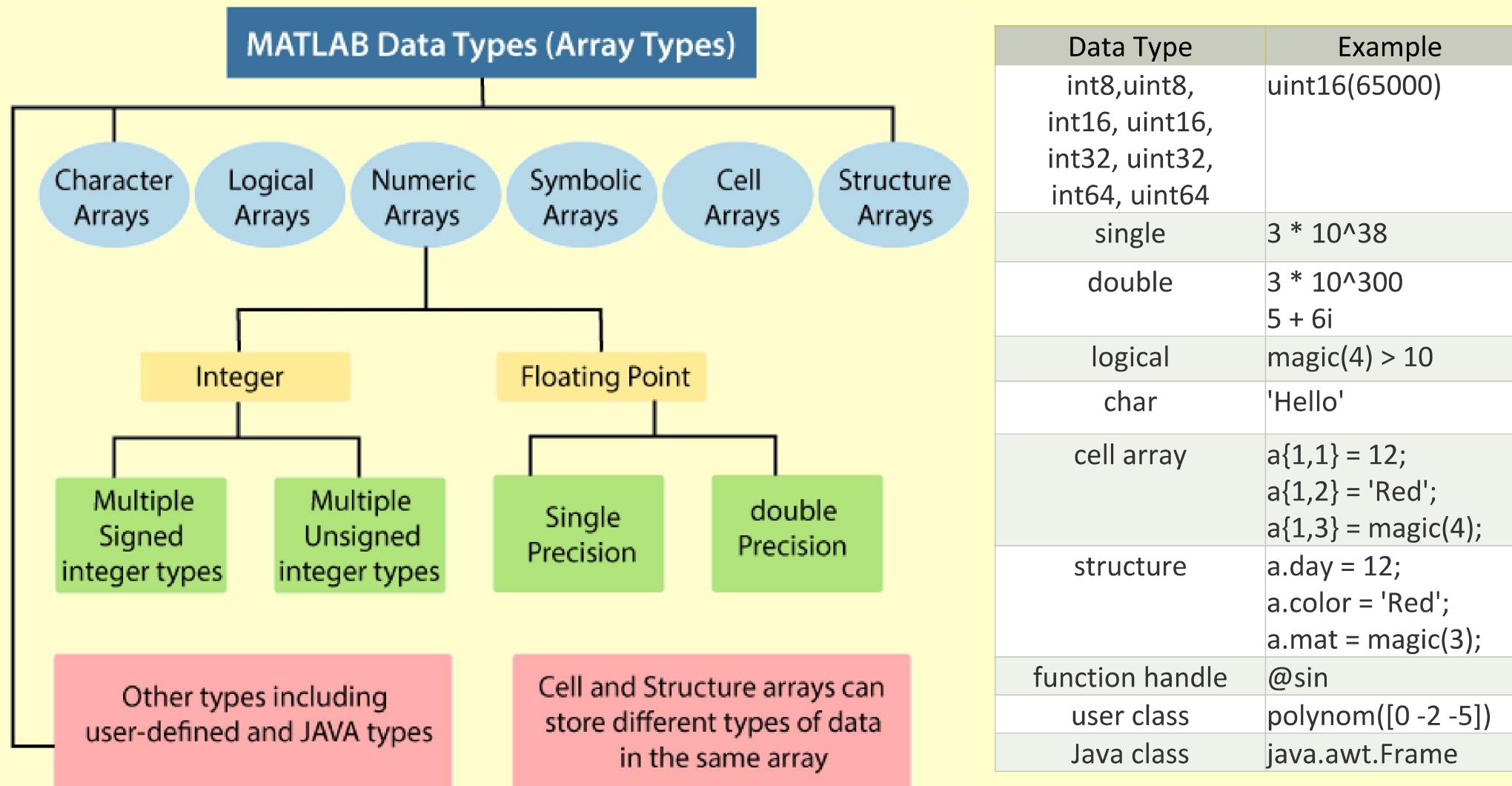
Name	Value	Size	Min	Max
a	100	1x1	100	100
b	[1,2,3,4,5]	1x5	1	5
c	10.1111	1x1	10.1111	10.1111

fx

Tai lieu pdf

1. Phan 1_Matlab...

Data Types Available in MATLAB



Định dạng kết quả hiển thị

format short : dấu phẩy thập phân cố định, 5 chữ số

format long : dấu phẩy cố định, 15 chữ số

format short e : ký hiệu khoa học, 5 chữ số

format long e : ký hiệu khoa học, 15 chữ số

format short g : dấu phẩy cố định hoặc di động, 5 chữ số

format long g : dấu phẩy cố định hoặc di động, 15 chữ số

format hex : format dạng Hexa (hệ 16)

format '+' : dương (+), âm (-), và ký tự trắng (blank) ứng với 0

format bank : Dollars và cents

format rat : tỷ lệ xấp xỉ integer

```
>> format rat
>> b
b =
1000001/1000000
```

>> help format

```
>> a=100
a =
100
>> b=1.00001
b =
1.0000
>> whos
  Name      Size            Bytes  Class       Attributes
  a         1x1              8  double
  b         1x1              8  double

>> format long
>> a
a =
100
>> b
b =
1.0000100000000000
>> whos
  Name      Size            Bytes  Class       Attributes
  a         1x1              8  double
  b         1x1              8  double
```

Ví dụ

Quan sát và nhận xét:

str = 'Hello World!'

n = 2345

d = double(n)

un = uint32(789.50)

rn = 5678.92347

c = int32(rn)

VECTOR, MÃNG & MA TRẬN

Mảng

Trong MATLAB, biến được lưu trữ như một mảng các số. Biến có thể là vô hướng, vector hay ma trận.

scalar

1×1

vector

$n \times 1$ or $1 \times n$

matrix

$n \times m$

Kích cỡ của mảng được xác định bởi số hàng và số cột của mảng, số đi trước chỉ hàng.

Vô hướng (Scalars)

- Vô hướng là trường hợp đặc biệt, khi đó kích cỡ mảng là 1×1 .
- Biến vô hướng chứa giá trị đơn, ví dụ:

$r = 6$

Height = 5.3

Width = 9.07

Vectors

- Vector là mảng 1 chiều.
- Vector có thể là $n \times 1$ or $1 \times n$.
- Các cột được phân tách bởi dấu phẩy (hay khoảng trắng):

$$\mathbf{h} = [1, 2, 3] \text{ hoặc } \mathbf{h} = [1 \ 2 \ 3]$$

- Hàng được phân biệt bởi dấu chấm phẩy

$$\mathbf{v} = [1; 2; 3]$$

Ma trận (Matrices)

- Ma trận là mảng hai chiều, là dạng dữ liệu chủ yếu dùng trong Matlab

- Ma trận 4×3 :

	Columns		
	1	2	3
1	3.0	1.8	3.6
2	4.6	-2.0	21.3
3	0.0	-6.1	12.8
4	2.3	0.3	-6.1

```
m = [3.0, 1.8, 3.6; 4.6, -2.0, 21.3; 0.0,  
-6.1, 12.8; 2.3, 0.3, -6.1]
```

Truy xuất vị trí trong ma trận

- Vị trí phần tử trong ma trận được truy xuất qua (row, column).

m(2 , 3)

ans =
21.3000

		Columns		
		1	2	3
Rows	1	3.0	1.8	3.6
	2	4.6	-2.0	21.3
3	0.0	-6.1	12.8	
4	2.3	0.3	-6.1	

Các loại ma trận

□ Nhập vào các biến sau trong Matlab:

□ Scalar: (đại lượng vô hướng)

```
a = 1
```

□ Vectors: (hàng hoặc cột)

```
b = [1, 0, 2]
```

```
c = [1 0 2]
```

□ Matrix: (hàng x cột)

```
d = [5, 4, 3; 0, 2, 8]
```

Ví dụ ma trận

- Nhập ma trận sau trong Matlab:

whiteRabbit =

-7	21	6
2	32	0
-5	0	-18.5

Các phép toán trên ma trận

Operation	Algebraic Syntax	MATLAB Syntax
Addition	$a + b$	$a + b$
Subtraction	$a - b$	$a - b$
Multiplication	$a \times b$	$a.*b$
Division	$a \div b$	$a./b$
Exponentiation	a^b	$a.^b$

Các phép toán trên mảng

- *, /, và ^
- **Thao tác trên phần tử mảng**
• *, . / và . ^

✓ Ví dụ:

x = [2, 1; 3, 4]

y = [5, 6; 7, 8]

z = x .* y

→ [10, 6; 21, 32];

z = x * y

→ [17, 20; 43, 50];

Mức độ ưu tiên của các phép toán

Trái qua phải, trong ngoặc, hàm mũ, nhân/chia, cộng/trừ

Ví dụ:

$$c = 2 + (3^2) + 1 / (1+2)$$

1st

$$c = 2 + 3^2 + 1 / (1+2)$$

2nd

$$c = 2 + 3^2 + 1 / (1+2)$$

3rd

$$c = 2 + 3^2 + 1 / (1+2)$$

4th

$$c = 2 + 3^2 + 1 / (1+2)$$

5th

$$c = 2 + 3^2 + 1 / 3$$

$$c = 2 + 9 + 1 / 3$$

$$c = 2 + 9 + 0.33333$$

$$c = 11 + 0.33333$$

$$c = 11.33333$$

Ví dụ

- Nhập vào 2 ma trận:

a =

10	5	5
2	9	0
6	8	8

b =

1	0	2
0	0	0
1	1	0

- Thực hiện nhân phần tử với phần tử trong Matlab

Mọi biến trong MATLAB là mảng !

- Mảng đơn hay vô hướng: 1×1
- Mảng hàng hay vector hàng: $1 \times n$
- Mảng cột hay vector cột: $n \times 1$
- Mảng n hàng và m cột hay ma trận: $n \times m$
- Truy xuất (hàng, cột)

Gán

- Phép gán

```
>> ax = 5;
```

```
>> bx = [1 2];
```

```
>> by = [3 4];
```

```
>> b = bx + by;
```

- Thực hiện các phép toán bên về phải rồi gán kết quả vào biến bên về trái.

size() và length() của mảng

```
b =  
1 2 3 4  
  
=> size(b)  
  
ans =  
1 4  
  
=> length(b)  
  
ans =  
4
```

Định nghĩa hay gán giá trị cho mảng

- Một mảng có thể được định nghĩa thông qua phép gán

A = [12, 18, -3] or A = [12 18 -3]

A =

$$\begin{matrix} 12 & 18 & -3 \end{matrix}$$

- Dấu , hay khoảng trắng để phân biệt các giá trị
- Dùng dấu ";" để thêm hàng mới.

B = [2, 5, 2; 1, 1, 2; 0, -2, 6]

B =

$$\begin{matrix} 2 & 5 & 2 \\ 1 & 1 & 2 \\ 0 & -2 & 6 \end{matrix}$$

Định nghĩa hay gán giá trị cho mảng

- Một cách khác để khai báo mảng

$$C = [A; B]$$

$$C =$$

$$\begin{matrix} 12 & 18 & -3 \\ 2 & 5 & 2 \\ 1 & 1 & 2 \\ 0 & -2 & 6 \end{matrix}$$

$$D = [C, C]$$

$$D = ?$$

$$\begin{matrix} 12 & 18 & -3 & 12 & 18 & -3 \\ 2 & 5 & 2 & 2 & 5 & 2 \\ 1 & 1 & 2 & 1 & 1 & 2 \\ 0 & -2 & 6 & 0 & -2 & 6 \end{matrix}$$

Định nghĩa hay gán giá trị cho mảng

- `linspace(0, pi/2, 10)` : creates a vector of 10 equally spaced points from 0 to $\pi/2$
- `y = [1 4 8 0 -1]'` : toán tử chuyển vị
- `x = 1:0.5:4`
- `x = 1:10`

Mảng toàn không và toàn một

- Tạo một mảng toàn không:

E = zeros(3,5)

E =

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

- Tạo một mảng toàn 1:

F = ones(2,3)

F =

1	1	1
1	1	1

Chú ý: **ones(4)** sẽ tạo ra một mảng 4x4 với toàn giá trị 1.

Truy xuất phần tử trong mảng

Tên mảng (vị trí phần tử)

G = [1, 2, 3; 4, 5, 6; 7, 8, 9]

G =

1	2	3
4	5	6
7	8	9

G(2,1)

ans = 4

G(3,2)

ans = 8

Thay đổi giá trị trong mảng

- Thay đổi giá trị của một phần tử trong mảng

$$A(2) = 5$$

A =

12 5 -3

- Mở rộng mảng:

$$A(6) = 8$$

A =

12 5 -3 0 0 8

- Lưu ý: những phần tử không khai báo sẽ được điền vào giá trị không

Toán tử hai chấm (:)

- Toán tử : có thể dùng như sau

first : last

$$H = 1:6$$

$$H =$$

1 2 3 4 5 6

- Khoảng nhảy mặc định là 1, để định nghĩa khoảng nhảy, dùng cấu trúc
first : increment : last

$$I = 1:2:11$$

$$I =$$

1 3 5 7 9 11

- Bước nhảy ở đây là 2

Trích dữ liệu trong mảng dùng toán tử hai chấm

- Xét ma trận:

G =

1	2	3
4	5	6
7	8	9

G(:,1)

ans =

1
4
7

G(:,3)

ans =

3
6
9

G(2,:)

ans =

4 5 6

Trích dữ liệu trong mảng dùng toán tử hai chấm

- Toán tử hai chấm cũng có thể được dùng để trích một khoảng trong hàng hay cột

G =

1	2	3
4	5	6
7	8	9

G(2:3, :)

G =

4	5	6
7	8	9

G(1, 2:3)

ans =

2	3
---	---

Phép toán trên mảng

□ Chuyển vị:

J = [1 , 3, 7]

J =
1 3 7

J'

ans =

1
3
7

Phép toán trên mảng

- ☐ `fliplr()` và `flipud()` lật ma trận left-to-right và top-to-bottom

G =
1 2 3
4 5 6
7 8 9

```
>> fliplr(G)
```

```
ans =
```

3 2 1
6 5 4
9 8 7

```
>> flipud(G)
```

```
ans =
```

7 8 9
4 5 6
1 2 3

Phép toán trên mảng

```
>> K=[G ; H]
```

K =

1	2	3
4	5	6
7	8	9
1	1	1

```
>> K=[G H']
```

K =

1	2	3	1
4	5	6	1
7	8	9	1

```
>> fliplr(K)
```

ans =

1	3	2	1
1	6	5	4
1	9	8	7

Ví dụ

- Tạo ma trận:

$W =$

1	2	3	4	5
10	12	14	16	18
6	5	4	3	2

$W = [1:5; 10:2:18; 6:-1:2]$

Ví dụ

- **Tạo ma trận:** $x = [1.2:1.1:5.6; 1.9:1.9:9.5; 0:-3:-12]$
$$\begin{matrix} 1.2 & 2.3 & 3.4 & 4.5 & 5.6 \\ 1.9 & 3.8 & 5.7 & 7.6 & 9.5 \\ 0 & -3 & -6 & -9 & -12 \end{matrix}$$
- **Chuyển vị ma trận x và gán kết quả vào biến Y.**
`>> Y = x'`
- **Lấy hàng 2 từ Y và gán vào biến Z**
`>> z = Y(2, :)`

Một số ví dụ về ma trận

$a=[1\ 2\ 3\ ;\ 4\ 5\ 6\ ;\ 7\ 8\ 9]$

a	=
1	2
4	5
7	8
	3
	6
	9

$b=[a\ 10*a; -a [1\ 0\ 0; 0\ 1\ 0; 0\ 0\ 1]]$

b	=																		
<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	4	5	6	7	8	9	<table border="1"><tr><td>10</td><td>20</td><td>30</td></tr><tr><td>40</td><td>50</td><td>60</td></tr><tr><td>70</td><td>80</td><td>90</td></tr></table>	10	20	30	40	50	60	70	80	90
1	2	3																	
4	5	6																	
7	8	9																	
10	20	30																	
40	50	60																	
70	80	90																	
<table border="1"><tr><td>-1</td><td>-2</td><td>-3</td></tr><tr><td>-4</td><td>-5</td><td>-6</td></tr><tr><td>-7</td><td>-8</td><td>-9</td></tr></table>	-1	-2	-3	-4	-5	-6	-7	-8	-9	<table border="1"><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td></tr></table>	1	0	0	0	1	0	0	0	1
-1	-2	-3																	
-4	-5	-6																	
-7	-8	-9																	
1	0	0																	
0	1	0																	
0	0	1																	

Một số ví dụ về ma trận

```
>> a = [1 2; 3 4]
a =
    1     2
    3     4
>> repmat(a,2,3)
```

replicate a matrix --- > help repmat

B = repmat(A,M,N) creates a large matrix B consisting of an M-by-N tiling of copies of A.

```
ans =
    1     2      1     2      1     2
    3     4      3     4      3     4
    1     2      1     2      1     2
    3     4      3     4      3     4
```

Ví dụ

```
>> a = zeros(2,3)
a =
    0     0     0
    0     0     0

>> b = ones(2,2)/2
b =
    0.5000    0.5000
    0.5000    0.5000

>> u = rand(1,5)
u =
    0.9218    0.7382    0.1763    0.4057    0.9355

>> n = randn(5,5)
n =
    -0.4326    1.1909   -0.1867    0.1139    0.2944
    -1.6656    1.1892    0.7258    1.0668   -1.3362
     0.1253   -0.0376   -0.5883    0.0593    0.7143
     0.2877    0.3273    2.1832   -0.0956    1.6236
    -1.1465    0.1746   -0.1364   -0.8323   -0.6918

>> eye(3)
ans =
    1     0     0
    0     1     0
    0     0     1
```

Ví dụ

```
>> a = [1 2 3;4 5 6;7 8 9]
a =
    1      2      3
    4      5      6
    7      8      9
>> a(:,2) = []
a =
    1      3
    4      6
    7      9
```

```
>> b = 10*a
b =
    10     30
    40     60
    70     90
```

```
>> a + b
ans =
    11     33
    44     66
    77     99
```

a, b cùng cỡ

```
>> a'
ans =
    1      4      7
    3      6      9
```

Chuyển vị

```
>> b = [[1 2 3]', [4 5 6]']
b =
    1      4
    2      5
    3      6
```

```
>> z=[ 1 -2i; i 1-i]
```

Số phức

`z =`

1.0000	0 - 2.0000i
0 + 1.0000i	1.0000 - 1.0000i

`>> z'`

Chuyển vị liên hợp phức

`ans =`

1.0000	0 - 1.0000i
0 + 2.0000i	1.0000 + 1.0000i

`>> z.'`

Chuyển vị không lấy liên hợp phức

`ans =`

1.0000	0 + 1.0000i
0 - 2.0000i	1.0000 - 1.0000i

Bài tập tự kiểm tra

a =[1 2 3]; b=[4;2;1]

Các lệnh nào sau đây hợp lệ và kết quả là gì:

a.*b, a./b, a.^b

a.*b', a./b', a.^b')

Lệnh diary

- **diary *filename***
Sẽ lưu mọi thứ trong Command Window vào
tập tin *filename*
- **diary off** : dừng lưu

Lệnh disp()

disp(name of a variable)

hoặc

disp('text as string')

Ví dụ:

```
disp( 'PP Tính , Matlab' );
```

```
x = 2;
```

```
disp( ['The answer is ', num2str(x)] );
```

```
disp( [x y z] )
```

Lệnh input()

```
variable_name = input('chuỗi sẽ hiển  
 thị trong Command Window')
```

VD:

```
Totalstudents=input('nhập vào số sinh viên')
```

Lệnh fprintf()

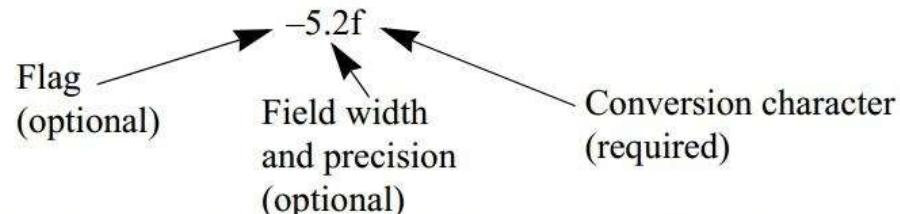
```
fprintf('text as string %-.5.2f additional text',
        variable_name)
```

The % sign marks the spot where the number is inserted within the text.

Formatting elements (define the format of the number).

The name of the variable whose value is displayed.

The formatting elements are:



The flag, which is optional, can be one of the following three characters:

e	Exponential notation using lowercase e (e.g., 1.709098e+001).
E	Exponential notation using uppercase E (e.g., 1.709098E+001).
f	Fixed-point notation (e.g., 17.090980).
g	The shorter of e or f notations.
G	The shorter of E or f notations.
i	Integer.

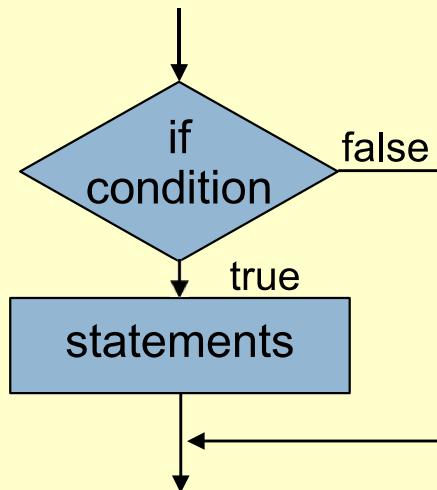
Character used for flag

- | | <u>Description</u> |
|----------------|--|
| - (minus sign) | Left-justifies the number within the field. |
| + (plus sign) | Prints a sign character (+ or -) in front of the number. |
| 0 (zero) | Adds zeros if the number is shorter than the field. |

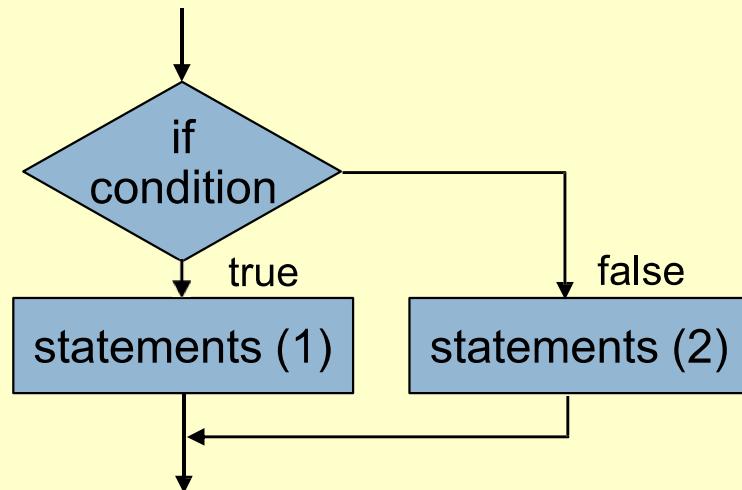
Chương 2

Các cấu trúc điều khiển

Cấu trúc if



```
if A>10  
    % computations;  
end
```



```
if A>10  
    % computations;  
else  
    % computations;  
end
```

Toán tử logic và quan hệ

Relational Operators

<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	equality
~=	not equal

Logical Operators

OP	Symbol
not	~
and	&
or	
xor	xor(A,B)

Note: 0 is false
Anything else is true

A	B	$\sim A$	$A B$	$A \& B$	$A \text{ xor } B$
0	0	1	0	0	0
0	1	1	1	0	1
1	0	0	1	0	1
1	1	0	1	1	0

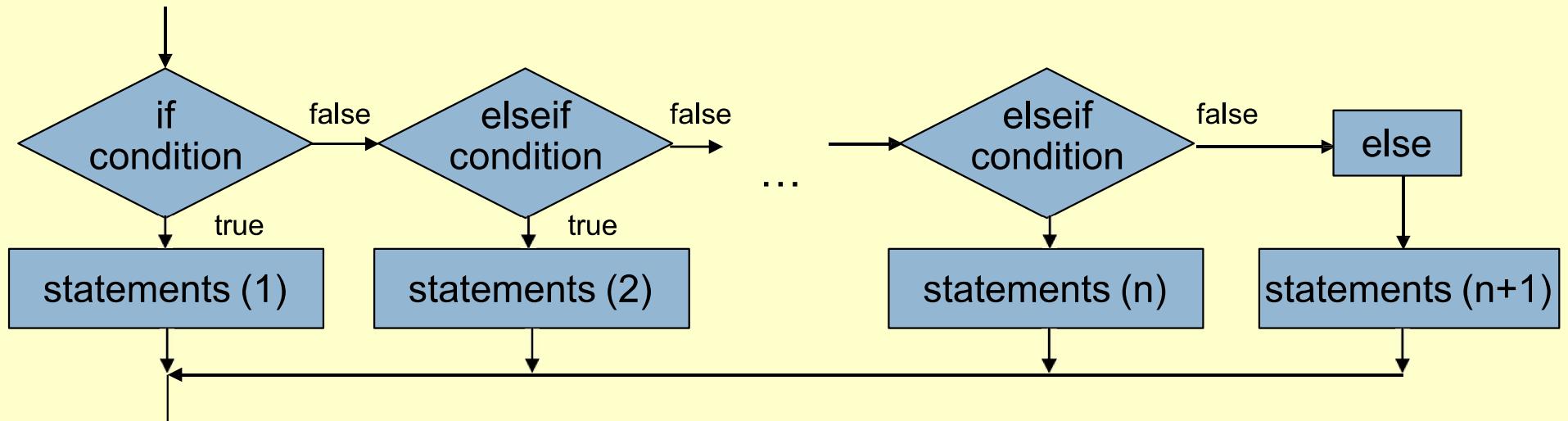
Ví dụ

```
>> apples = 10; % number of apples
>> cost = apples*25 % cost of apples
cost =
250

>> if apples>5 % give 20% discount for larger purchases
    cost = (1-20/100)*cost;
end

>> cost
cost =
200
```

Cấu trúc if-elseif



```
if A>10  
    % computations;  
elseif A<10  
    % computations;  
else  
    % computations  
end
```

Ví dụ

$$y = \sqrt{x} \quad x \geq 0$$

$$y = e^x - 1 \quad x < 0$$

$$y = \ln x, \quad x > 10$$

$$y = \sqrt{x}, \quad 0 \leq x \leq 10$$

$$y = e^x - 1, \quad x < 0$$

```

if x >= 0
    y = sqrt(x)
else
    y = exp(x) - 1
end

```

$$y = \ln x, \quad x \geq 5$$

$$y = \sqrt{x}, \quad 0 \leq x < 5$$

```

if x > 10
    y = log(x)
elseif x >= 0
    y = sqrt(x)
else
    y = exp(x) - 1
end

```

```

if x >= 5
    y = log(x)
else
    if x >= 0
        y = sqrt(x)
    end
end

```

```

if x >= 5
y = log(x)
elseif x >= 0
y = sqrt(x)
end

```

Ví dụ

```
%DEMO  
function output = DEMO(input)  
  
%put help info here!  
%Do stuff  
if input > 0  
    fprintf('Greater than 0')  
elseif input < 0  
    fprintf('Less than 0')  
else  
    fprintf('Equals Zero')  
end  
  
%Set return value if needed  
outvar = 1;
```

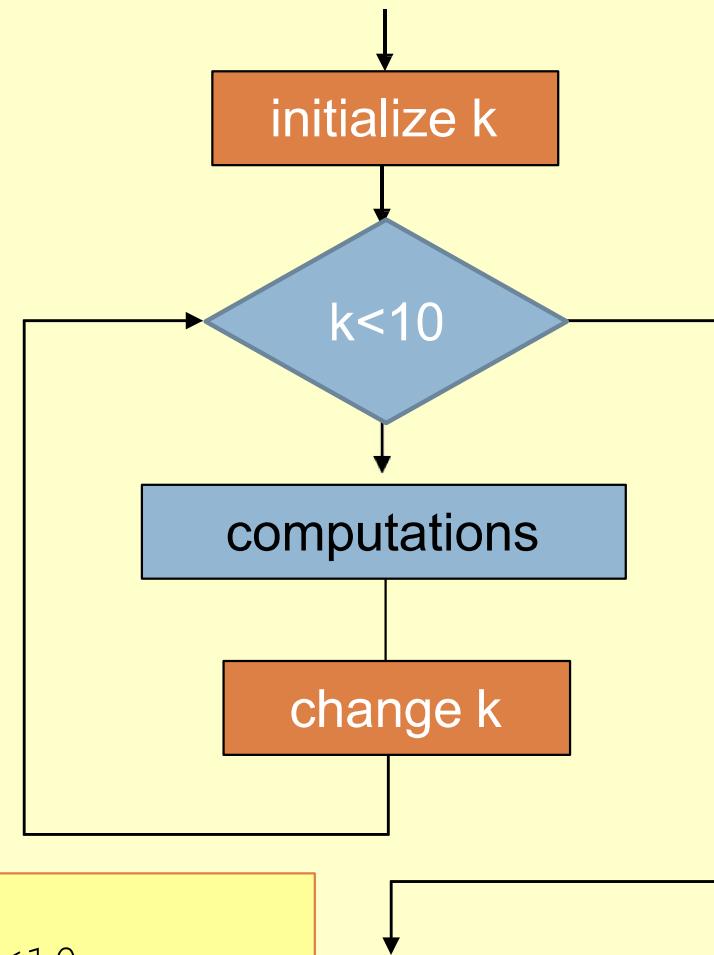
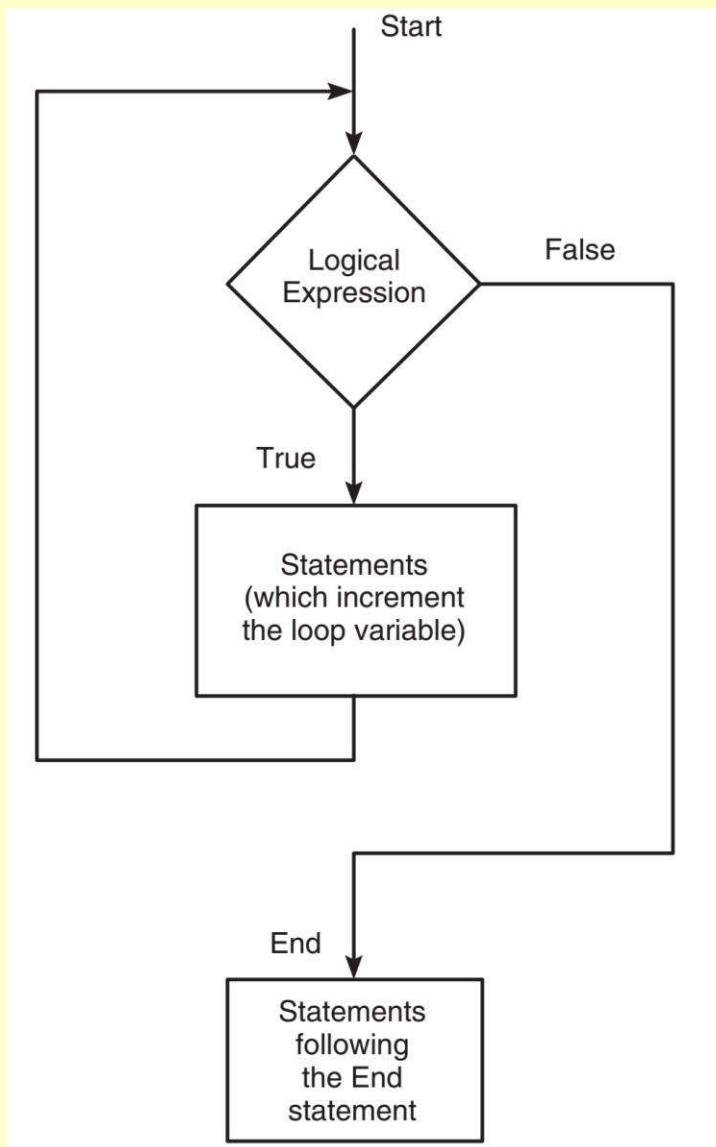
Cấu trúc switch-case

```
..... MATLAB program.  
.....  
  
switch switch expression  
    case value1  
        ..... ] Group 1 of commands.  
    .....  
    case value2  
        ..... ] Group 2 of commands.  
    .....  
    case value3  
        ..... ] Group 3 of commands.  
    .....  
    otherwise  
        ..... ] Group 4 of commands.  
    .....  
end  
..... MATLAB program.  
.....
```

Ví dụ cấu trúc switch-case

```
x=2.5;  
units='m'; %convert to centimeters  
switch units  
    case {'inch','in'}  
        y=x.*2.54;  
    case {'feet','ft'}  
        y=x.*2.54.*12;  
    case {'meter','m'}  
        y=x.*100;  
    case {'centimeter','cm'}  
        y=x;  
    otherwise  
        disp(['Unknown units: ' units])  
        y=NaN;  
end
```

Cấu trúc lặp while

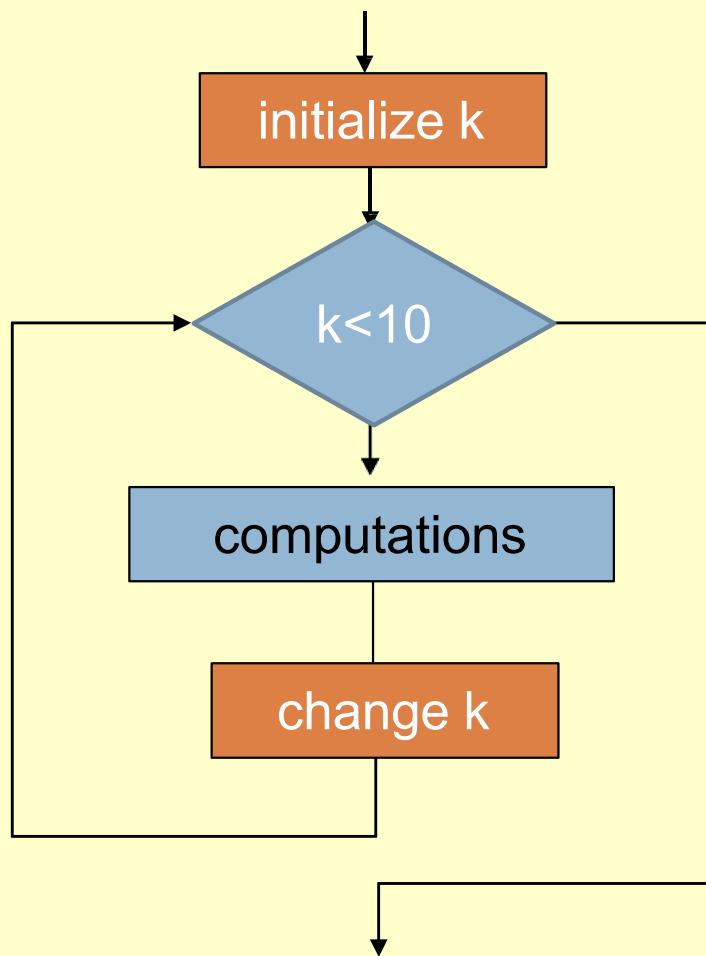


```

k=0;
while k<10
    % computations;
    k=k+1;
end
  
```

The corresponding pseudocode for the flowchart is shown in a yellow box with a red border:

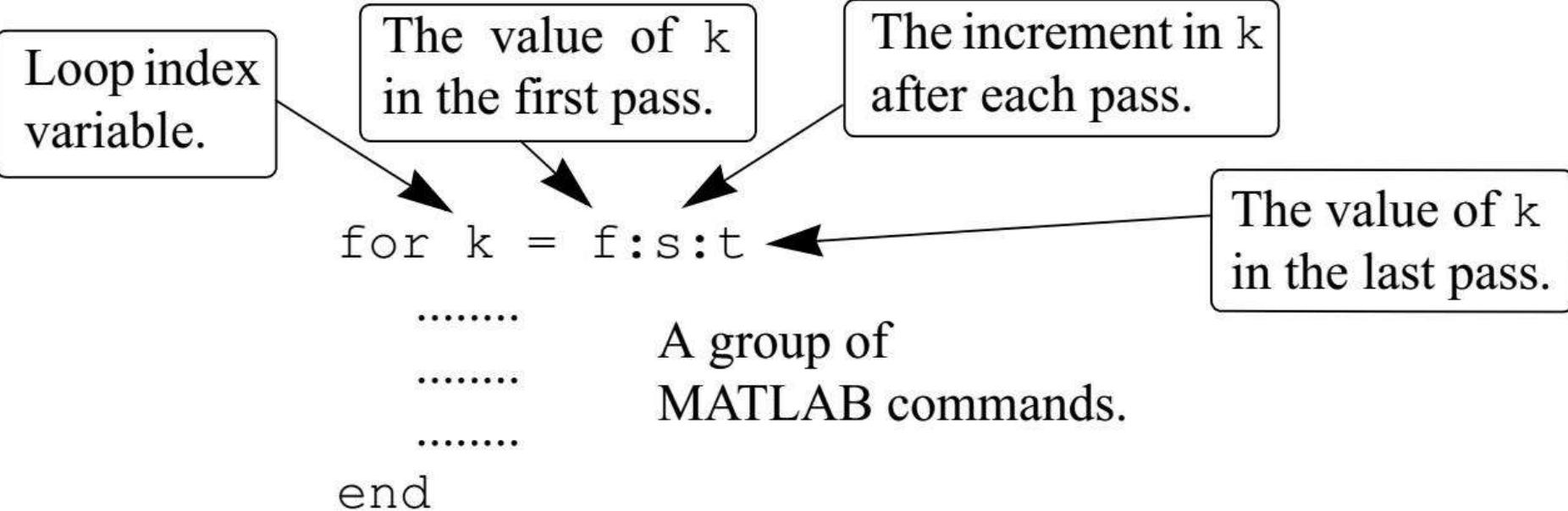
Cấu trúc lặp while



```
k=0;  
while k<10  
    % computations;  
    k=k+1;  
end
```

- Cấu trúc trong vòng lặp được thực thi khi điều kiện vẫn thỏa
- Lưu ý cần phải khởi tạo biến điều khiển lặp
- Vòng lặp có thể chạy vô hạn nếu biến điều khiển lặp không được cập nhật.

Cấu trúc for



```
for k=1:3:10
    x = k^2
end
```

```
>> x =
1
x =
16
x =
49
x =
100
```

Cấu trúc for

Use a for-end loop in a script file to calculate the sum of the first n terms of the series: $\sum_{k=1}^n \frac{(-1)^k k}{2^k}$. Execute the script file for $n = 4$ and $n = 20$.

```

n=input('Enter the number of terms ' );
S=0;
for k=1:n
    S=S+(-1)^k*k/2^k;
end
fprintf('The sum of the series is: %f',S)

```

Setting the sum to zero.

for-end
loop.

In each pass one element of the series is calculated and is added to the sum of the elements from the previous passes.

Disp() and fprintf()

- `disp(X)` – Xuất ra màn hình các phần tử của X
- `disp('hello world')` – Xuất ra màn hình chuỗi

- `fprintf(fid, format, A)` – Xuất có định dạng:
 - Ghi A vào file **fid** có định dạng theo **format**
(Nếu bỏ qua fid sẽ xuất ra màn hình)
 - **format** chứa các chỉ dẫn định dạng hay chuỗi cho các biến A
 - Các chỉ dẫn định dạng thông dụng gồm **d, i, o, u, x, X, f, e, E, g, G, c, và s.**
 - Một số ký tự đặc biệt gồm **\n,\r,\t,\b,\f**

Ví dụ

```
% This program will calculate the
% area and circumference of ten circles,
% allowing the radius as an input,
% but will only provide output for circles
% with an area that exceeds 20.
N = 0; R = 0.0; AREA = 0.0; CIRC = 0.0;
for J = 1:1:10
    R = input('Please enter the radius: ');
    AREA = pi * R^2;
    CIRC = 2.0 * pi * R;
    if AREA > 20.0
        fprintf('\n Radius = %f units',R)
        fprintf('\n Area = %f units squared', AREA)
        fprintf('\n Circumference = %f units\n', CIRC)
    else
        N = N + 1;
    end
end
fprintf('\n Number of circles that do not have area > 20: %.0f \n', N)
```

Một số từ khóa đặc biệt

pi – 3.1416

i,j – $\sqrt{-1}$

inf – infinity

NaN – “Not a number”

clock – date and time

date – Current date in string form

eps – “Epsilon” the smallest amount by which
two values can differ on the current computer

ans – just computed value

CHƯƠNG 3

Hàm trong Matlab



Hàm trong Matlab

- Sử dụng hàm trong MATLAB

input, output, and name

$$\mathbf{b} = \tan(\mathbf{x})$$

\mathbf{x} is the input, \mathbf{b} is the output, and \tan is the name of the function; in this case, it is the tangent function with argument in radians.

- Hàm gồm 2 loại: có sẵn (dựng sẵn) và hàm tự tạo
- Dạng của hàm:

variable = function(number or variable)

Hàm trong Matlab

- Để biết danh sách các hàm trong Matlab dùng lệnh help
- Để biết chi tiết cách sử dụng một hàm, dùng **help tên_hàm**

Ví dụ:

help tan

MATLAB Help

The screenshot shows the MATLAB Documentation interface. The title bar reads "Arithmetic Operations". The main content area is titled "Arithmetic Operations" and describes basic arithmetic operations like addition, subtraction, multiplication, division, power, and rounding. It also mentions functions for common calculations like summation, moving sums, modulo operations, and rounding. A "Functions" section is shown with a "Basic Arithmetic" subsection containing tables for "Addition" and "Subtraction".

Documentation Home

« MATLAB

« Language Fundamentals

« Operators and Elementary Operations

Category

Arithmetic Operations

Relational Operations

Logical Operations

Set Operations

Bit-Wise Operations

Help

Arithmetic Operations

Search Help

All Examples Functions

Arithmetic Operations

Addition, subtraction, multiplication, division, power, rounding

Arithmetic functions include operators for simple operations like addition and multiplication, as well as functions for common calculations like summation, moving sums, modulo operations, and rounding.

For more information, see [Array vs. Matrix Operations](#).

Functions

expand all

Basic Arithmetic

<code>+</code>	Addition or append strings
<code>sum</code>	Sum of array elements
<code>cumsum</code>	Cumulative sum
<code>movsum</code>	Moving sum

Addition

<code>+</code>	Addition or append strings
<code>sum</code>	Sum of array elements
<code>cumsum</code>	Cumulative sum
<code>movsum</code>	Moving sum

Subtraction

<code>-</code>	Subtraction

More Help

Command Window

```
>> help
```

New to MATLAB? See resources for [Getting Started](#).

To view the documentation, [open the Help browser](#).

```
>> help tan
```

tan Tangent of argument in radians.

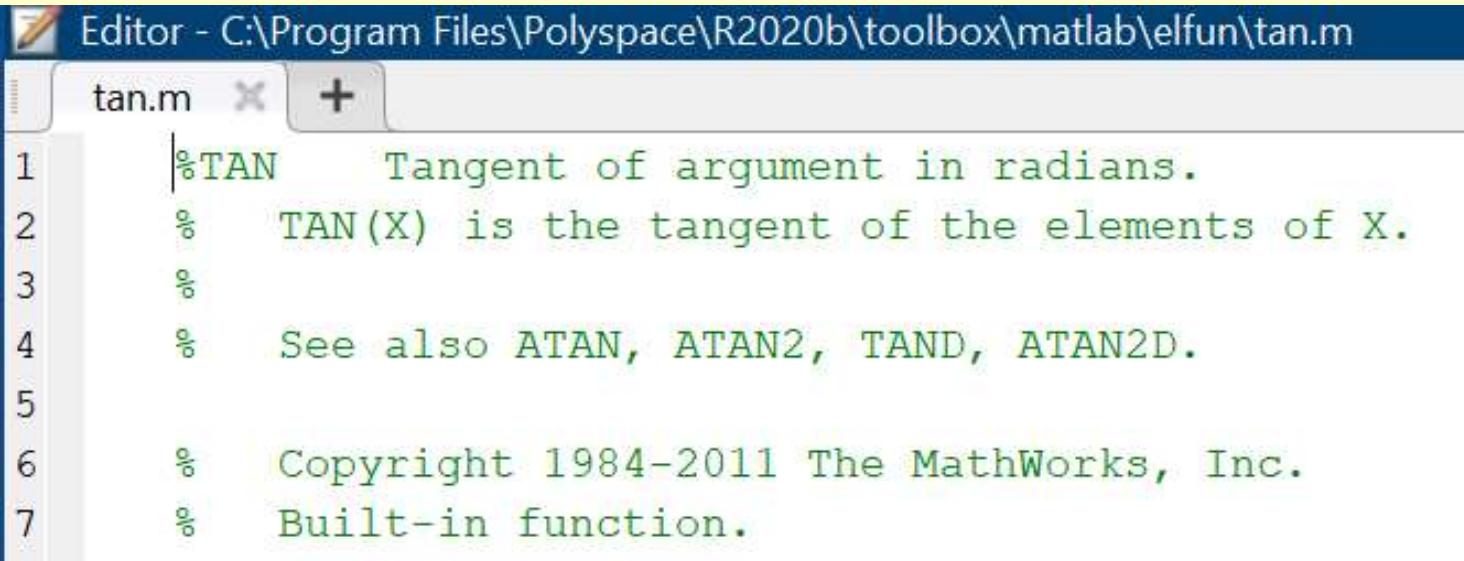
tan(X) is the tangent of the elements of X.

See also [atan](#), [atan2](#), [tand](#), [atan2d](#).

[Documentation for tan](#)

[Other functions named tan](#)

Edit tan



The screenshot shows a MATLAB code editor window titled "Editor - C:\Program Files\Polyspace\R2020b\toolbox\matlab\elfun\tan.m". The file contains the following code:

```
tan.m %TAN      Tangent of argument in radians.  
% TAN(X) is the tangent of the elements of X.  
%  
% See also ATAN, ATAN2, TAND, ATAN2D.  
  
% Copyright 1984-2011 The MathWorks, Inc.  
% Built-in function.
```

Đây là hàm có sẵn trong Matlab, không thay đổi được.
Có thể thay thế bằng một hàm khác tự tạo có cùng tên.

Help Navigator window

The screenshot shows the MATLAB Documentation Center window. A blue arrow points from the title "Help Navigator window" to the search bar at the top of the window. The search bar contains the text "tan". The main content area displays the documentation for the `tan` function. The title "tan" is highlighted in yellow. Below it, the text "Tangent of argument in radians" is shown. There is a "collapse all in page" link. The "Syntax" section includes the code snippet `Y = tan(X)`. The "Description" section states: "`Y = tan(X)` returns the tangent of each element of `X`. The `tan` function operates element-wise on arrays. The function accepts both real and complex inputs." It also includes a "example" link. A bulleted list under "Description" specifies: "For real values of `X`, `tan(X)` returns real values in the interval $[-\infty, \infty]$. For complex values of `X`, `tan(X)` returns complex values." The "Examples" section has a "Plot Tangent Function" button. The bottom right corner features a "Open Live Script" button. On the left sidebar, there are links for "CONTENTS", "All", "Examples", "Functions", and categories like "Documentation Home", "MATLAB", "Mathematics", "Elementary Math", "Trigonometry", and "tan". The "ON THIS PAGE" sidebar lists "Syntax", "Description", "Examples", "Input Arguments", "Output Arguments", "More About", "Tips", "Extended Capabilities", and "See Also".

Các hàm lượng giác:

sin	- hàm sin. Đơn vị góc là radian
sind	- sin của argument tính theo độ.
sinh	- sin hyperbolic.
asin	- arcsin, hay hàm nghịch đảo của hàm sin.
asind	- hàm nghịch đảo của hàm sin, kết quả theo độ.
asinh	- hàm nghịch đảo của hàm sin hyperbolic.
cos	- hàm cos.
cosd	- cos của argument tính theo độ.
cosh	- cos hyperbolic.
acos	- hàm nghịch đảo của hàm cos.
acosd	- hàm nghịch đảo của hàm cos, kết quả theo độ.
acosh	- hàm nghịch đảo của hàm cos hyperbolic.

NOTE: $\sin(\pi) = 1.2246e-16$ (không phải bằng 0)

Các hàm lượng giác (TT)

tan	- hàm tang.
tand	- tang của argument tính theo độ.
tanh	- tang hyperbolic.
atan	- hàm nghịch đảo của hàm tang.
atand	- hàm nghịch đảo của hàm tang, kết quả theo độ.
atan2	- hàm nghịch đảo của hàm tang 4 góc phần tư.
atanh	- hàm nghịch đảo của hàm tang hyperbolic.
cot	- hàm cötang.
cotd	- cötang của argument tính theo độ..
coth	- cötang hyperbolic.
acot	- hàm nghịch đảo của hàm cötang.
acotd	- hàm nghịch đảo của hàm cötang, kết quả theo độ.
acoth	- hàm nghịch đảo của hàm cötang hyperbolic.

Các hàm lũy thừa

<code>exp</code>	- hàm mũ.
<code>expml</code>	- tính chính xác $\exp(x)-1$.
<code>log</code>	- logarit cơ số tự nhiên.
<code>log1p</code>	- tính chính xác $\log(1+x)$.
<code>log10</code>	- logarit cơ số 10.
<code>reallog</code>	- loga cơ số tự nhiên của số thực.
<code>realsqrt</code>	- căn bậc hai của một số ≥ 0 .
<code>sqrt</code>	- căn bậc hai.
<code>nthroot</code>	- nghiệm thực bậc n của các số thực.

- Calculate e^7 : $\text{exp}(7)$
- Calculate $\ln(4)$: $\text{log}(4)$
- Calculate $\log_2(12)$: $\text{log2}(12) \Rightarrow \lg(10)?$

Các hàm liên quan đến số phức

abs	- giá trị tuyệt đối.
angle	- góc pha.
complex	- xây dựng dữ liệu về số phức từ các phần thực và ảo.
conj	- liên hợp của phức.
imag	- phần ảo của phức.
real	- phần thực của phức.
isreal	- hàm logic, trả về giá trị 'true' với mảng số thực.

Các hàm làm tròn và phần dư

fix	- làm tròn về phía 0.
floor	- làm tròn về phía âm vô cùng.
ceil	- làm tròn về phía dương vô cùng.
round	- làm tròn về phía số nguyên gần nhất.
mod	- mô đun (lấy phần dư của phép chia).
rem	- lấy phần dư của phép chia (tương tự mod)
sign	- hàm lấy dấu của một biến, trả về +1, 0, -1 (+, 0, -).

Chuỗi

□ Chuỗi hay mảng các ký tự

The screenshot shows the MATLAB environment with the Command Window and Workspace.

Command Window:

```
>> s1='hello'  
s1 =  
hello  
  
>> b='kitty'  
b =  
kitty  
  
>> s2=[s1 ' ' b]  
s2 =  
hello kitty
```

Workspace:

Name	Value	Size	By...	Min
ab\ b	'kitty'	1x5	10	
ab\ s1	'hello'	1x5	10	

Current Command Window:

```
>> s2(5)  
ans =  
o
```

Chuỗi – bảng mã Ascii



Name	Value	Size	Type	Min	Max
a	'1234'	1x4		8	
b	1234	1x1		8	1234
c	[98,100,1...	1x4		32	98

```

a =
1234
>> b=1234
b =
1234
>> a='1234'
a =
1234
>> b=1234
b =
1234
>> c=2*a
c =
98   100   102   104

```

Chuỗi và số

```
>> d= str2num(a)
```

```
d =
```

```
1234
```

```
>> e=num2str(b)
```

```
e =
```

```
1234
```

Name	Value	Size	Bytes	Min	Max
a	'1234'	1x4	8		
b	1234	1x1	8	1234	1234
c	[98,100,1...]	1x4	32	98	104
d	1234	1x1	8	1234	1234
e	'1234'	1x4	8		

Một số hàm thống kê

- Tự tìm hiểu công dụng của các hàm:

max()

min()

mean()

median()

sum()

prod()

sort()

sortrows()

size()

length()

std()

var()

Ví dụ

`x = [5, 3, 7, 10, 4]`

Giá trị lớn nhất? `[value,position] = max(x)`

`value = 10`
`position = 4`

highest value is 10
located in the 4th column

Median?

`median(x)`

`ans = 5`

Tổng các phần tử? `sum(x)`

`ans = 29`

Bài tập tự kiểm tra

v = [2, 24, 53, 7, 84, 9]

y = [2, 4, 56; 3, 6, 88]

1. Xếp các phần tử của v theo thứ tự giảm dần.
2. Tìm kích cỡ của y
3. Tìm **độ lệch chuẩn** của v : hàm std(x)
4. Tìm tích số tích luỹ của v : hàm cumprod(x,dim)

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}}$$

Bài tập tự kiểm tra

```
>> x= [3 5 7 10 4 12 4 9]  
  
x =  
    3      5      7      10      4      12      4      9  
  
>> median(x)  
  
ans =  
    6  
  
>> sort(x, 'descend')  
  
ans =  
   12      10      9      7      5      4      4      3  
  
>> sort(x, 'ascend')  
  
ans =  
    3      4      4      5      7      9      10     12
```

Bài tập tự kiểm tra

```
>> Y= [1 2 3; 4 5 6; 7 8 9]
```

```
Y =
```

1	2	3
4	5	6
7	8	9

```
>> Y2=cumprod(Y,1)
```

```
Y2 =
```

1	2	3
4	10	18
28	80	162

```
>> Y2=cumprod(Y,2)
```

```
Y2 =
```

1	2	6
4	20	120
7	56	504

```
X =
```

3	5	7	10	4	12	4	9
---	---	---	----	---	----	---	---

```
>> X1=cumprod(X,2)
```

```
X1 =
```

Columns 1 through 6

3	15	105	1050	4200	50400
---	----	-----	------	------	-------

Columns 7 through 8

201600	1814400
--------	---------

```
>> X2=cumprod(X,1)
```

```
X2 =
```

3	5	7	10	4	12	4	9
---	---	---	----	---	----	---	---

Tạo số ngẫu nhiên

- `rand(n)` tạo ma trận vuông nxn, mỗi phần tử là số ngẫu nhiên từ 0 - 1 .
- `rand(n,m)` tạo ma trận nxm, mỗi phần tử là số ngẫu nhiên từ 0 - 1

Để có được giá trị ngẫu nhiên lớn hơn/nhỏ hơn 1, hay số âm dùng định dạng sau:

$$w=40.*rand(1)$$

Số phức

- Số phức có dạng $a+b*i$:

- a là phần thực
- b là phần ảo

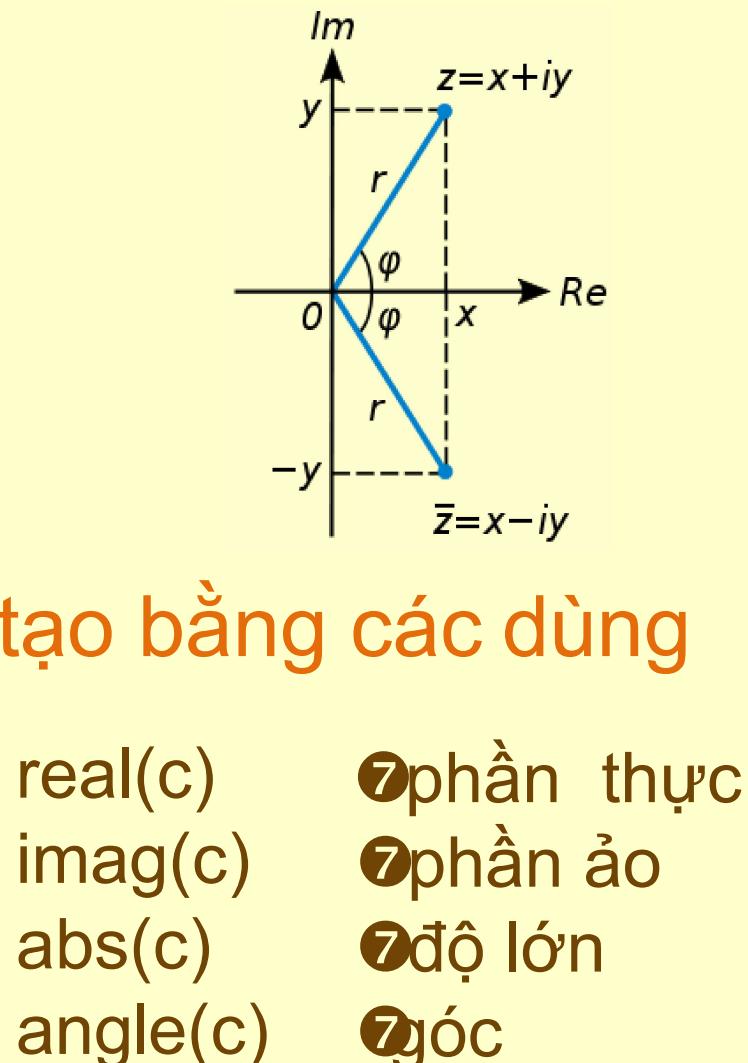
và $i = \sqrt{-1}$.

- Số phức cũng có thể được tạo bằng cách dùng hàm có sẵn **complex()**

`>> a = 2; b = 3;`

`>> c = a+b*i`

`>> c = complex(a,b)`



Một số hàm khác

- **clock:** produces an array that tells year, month, day, hour, min, sec.

```
>> clock ans =  
1.0e+003 *  
2.0160 0.0070 0.0280 0.0150 0.0180 0.0592
```

- **date:** tells date

```
>> date  
ans =  
28-Jul-2016
```

- **pi:** the number pi (3.141592653589.....)

```
>> pi  
ans =  
3.1416
```

eps: khoảng sai biệt nhở nhất giữa hai dấu chấm động
eps = 2.2204e-016.

Các phép toán so sánh

* Các toán tử quan hệ (so sánh):

Toán tử	Cú pháp	Mô tả
<	$A < B$	Nhỏ hơn
\leq	$A \leq B$	Nhỏ hơn hoặc bằng
>	$A > B$	Lớn hơn
\geq	$A \geq B$	Lớn hơn hoặc bằng
$=$	$A = B$	Bằng
\sim	$A \sim B$	Không bằng

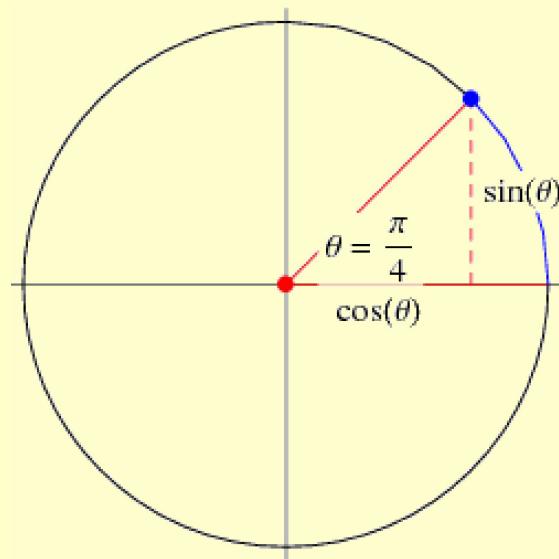
So sánh phần tử với phần tử.
Kết quả là mảng logic cùng cỡ.

Z	<3x3 logical>	3x3	9
Z1	<3x3 logical>	3x3	9
Z2	<3x3 logical>	3x3	9
-	1 1 1 0 0 0 1 1 1	1..4	9

```
>> X= 3*ones(3,3)
X =
3 3 3
3 3 3
3 3 3
>> Y= [1 2 3; 4 5 6; 7 8 9]
Y =
1 2 3
4 5 6
7 8 9
>> Z=X<Y
Z =
0 0 0
1 1 1
1 1 1
>> Z1=X>Y
Z1 =
1 1 0
0 0 0
0 0 0
```

Các phép toán so sánh

```
a = 0;  
b = sin(pi);  
a == b  
  
ans =  
0
```



Tại sao?

>> sin(pi)==cos(pi/2) => kết quả ?

Phép toán logic

- *logical* là một kiểu dữ liệu trong Matlab; nhận 1 trong 2 giá trị 1 (true) hay 0 (false).

Input	Output: ans =	Data type
$4 < 5$	1	Logical array
$4 \leq 5$	1	Logical array
$4 > 5$	0	Logical array
$4 \geq 5$	0	Logical array
$4 \approx 5$	1	Logical array

Các toán tử logic

Toán tử

&

|

~

Phép toán

Logical AND

Logical OR

Logical NOT

`z1 =`

1	1	0
0	0	0
0	0	0

`>> z2`

`z2 =`

0	0	1
0	0	0
0	0	0

`>> z3=z1|z2`

`z3 =`

1	1	1
0	0	0
0	0	0

Logical Operators: Examples

Input:

```
a = true;
b = false;
c = true;
```

Can you guess the results of:

a&b	0 (False)
b&c	0
a&c	1 (True)
a b	1
b c	1
a c	1
c a	1
b b	0

expression	result
(6>3) & (4>2)	true (or 1)
7==4 2<6	true (or 1)
~(2>4)	true (or 1)
(~(2>4)) 2>1	true (or 1)
(2<0)&((1>-1) (4==2))	false(or 0)

Các toán tử trên bit

Function	Purpose	
bitand(a, b)	Bit-wise AND of integers <i>a</i> and <i>b</i>	a = 60
bitcmp(a)	Bit-wise complement of <i>a</i>	>> b
bitor(a, b)	Bit-wise OR of integers <i>a</i> and <i>b</i>	b = 0.5000
bitxor(a, b)	Bit-wise XOR of integers <i>a</i> and <i>b</i>	>> b=13
Giả sử A = 60; and B = 13; A = 0011 1100 B = 0000 1101	A&B = 0000 1100 A B = 0011 1101 A^B = 0011 0001 ~A = 1100 0011	 b = 13
Giả sử A = 5; and B = 4; A = 0000 0101 B = 0000 0100 A&B = 0000 0100 (4) A B = 0000 0101 (5)		>> bitand(a,b) ans = 12

Trình tự các phép toán

Precedence

1 (highest)

Operation

Parentheses (if nested parentheses exist, inner ones have precedence)

2

Exponentiation

3

Logical NOT (\sim)

4

Multiplication, division

5

Addition, subtraction

6

Relational operators ($>$, $<$, \geq , \leq , \neq , \sim)

7

Logical AND ($\&$)

8 (lowest)

Logical OR ($|$)

```
>> x=-2; y=5;
>> -5<x<-1
```

```
ans =
```

```
0
```

```
>> -5<x & x<-1
```

```
ans =
```

```
1
```

```
>> ~(y<7)
```

```
ans =
```

```
0
```

```
>> ~y<7
```

```
ans =
```

```
1
```

```
>> ~(y>=8) | (x<-1)
```

```
ans =
```

```
0
```

```
>> ~(y>=8) | (x<-1)
```

```
ans =
```

```
1
```

Nếu cùng cấp → thực hiện từ trái qua phải

So sánh C và Matlab code

C code	MATLAB code
<pre>double a; a = 5.0;</pre>	<pre>a = 5.0;</pre>
<pre>// repeat which assigns values to // array elements for(i=1; i<=10; i++) A[i] = i;</pre>	<pre>% i = start:increment:end % where increment = 1 is default for i=1:10 A(i) = i; end</pre>
<pre>// repeat which prints a series of // values for(i=0; i<=10; i += 2) printf("%i \n", i);</pre>	<pre>for i=0:2:10 fprintf(' %i \n', i) end</pre>
<pre>// initialize identity matrix enum(N=100); double B[N]; for(r=0;r<N;r++) for(c=0;c<N;c++) B[r][c] = 0.0; for(r=0;r<N;r++) B[r][r] = 1.0;</pre>	<pre>% MATLAB has functions for % common array initializations B = eye(100);</pre>

```
// declare and initialize an array
double C[3] = {1,2,3};
```

C = [1 2 3]; % or C = [1, 2, 3]

```
// declare and initialize an array
double C[4] = {2,4,6,8};
```

% array name = [start:increment:end];
C = [2:2:8];

```
// print an array element on screen
printf("%g \n", C[2]);
```

C(2) % no ending ; displays value

```
// declare and initialize an array
// with fixed interval between values
double C[4] = {2,4,6,8};
```

C = linspace(2,8,4)
% third param is optional and = # points
% between and including 1st two points
% if third param left off, default
% is 100 points

```
// declare and initialize a 2D array
enum{Rows=3, Cols=3};
double D[Rows][Cols] = {{1,2,3},
{4,5,6},
{7,8,9}};
```

% these three examples accomplish the
% same thing
D = [1 2 3; 4 5 6; 7 8 9];
D = [1:3; 4:6; 7:9];
D = [1 2 3
 4 5 6
 7 8 9];

```
// print element of 2D array
```

```
printf("%g \n", D[2][2]);
```

D(2,2) % row 2, column 2

```
// print selected sub array of 2D array
```

```
for(r=1; r<=2; r++)
    printf("%g \n", D[r][1]);
```

D(1:2,1) % rows 1 to 2 of column 1

```
// print all rows of column 1 of 2D
// array
```

```
for(r=1; r<Rows; r++)
    printf("%g \n",
        D[r][1]);
```

D(:,1) % all rows of column 1

```
// logical expression
```

```
double a = 1;
double b = 2;
if (a == 1 || b == 3)
    printf("a = 1 or b = 3
\n");
```

```
a = 1;
b = 2;
if a == 1 || b == 3
    fprintf('a = 1 or b = 3 \n');
End
```

```
// logical expression
```

```
if a == 1 && b != 3 {
    printf("a=1 and b not 3 \n");
    printf("OK? \n");
}
```

```
if a == 1 && b ~= 3
    fprintf('a=1 and b not 3 \n');
    fprintf('OK? ');
End
```

```
// if structure
if a != 1
    fprintf('a is not 1 \n');
else if b != 3
    fprintf('b is not 3 \n');
else
    fprintf('huh? \n');
```

```
// switch structure
```

```
switch a+b {
    case 1:
        printf("a+b = 1 \n");
        break;
    case 2:
        printf("a+b = 2 \n");
        break;
    case 3:
        printf("a+b = 3 \n");
        break;
    default:
        printf("a+b > 3 \n");
}
```

```
// matrix multiplication
for(i=0; i<I; i++)
    for(k=0; k<K; k++)
        C[i][k] = 0;
// matrix multiplication, C = AB
for(k=0; k<K; k++)
    for(i=0; i<I; i++)
        for(j=0; j<J; j++)
            C[i][k] += A[i][j] * B[j][k];
```

```
if a ~= 1
    fprintf('a is not 1 \n')
elseif b ~= 3
    fprintf('b is not 3 \n')
else
    fprintf('huh? \n')
end
```

```
switch a+b
    case 1
        printf('a+b = 1 \n')
    case 2
        printf('a+b = 2 \n')
    case 3
        printf('a+b = 3 \n')
    otherwise
        printf('a+b > 3 \n')
```

End

```
C = A*B;
```

```
// program which calls a function
```

```
#include <stdio.h>
% variables defined outside a
% program unit are global
y = 1;
z = 2;

main() {
    int b;

    b = myfunc(3);

    printf("%i \n", b);
}

// function definition can be in
// same file or separate file

int myfunc(int x) {
    int a;
    a = x*y*z;
    return(a);
}
```

% program and function definition must
 % be in separate files and function file
 % must have same name as function

```
global y z
% declare global variables before
% first use
y = 1;
z = 2;
b = myfunc(3)
% function definition of myfunc
% must be in a separate file named
% myfunc.m (i.e., functionName.m)
LISTING OF FILE myfunc.m
function a = myfunc(x)

% function keyword, return variable
% = function name (inputs)
%
% 1st continuous group of
% comments in a function are
% displayed when you enter 'help'
% and name of function in the command
% window

global y z
% declare global variables used
% in this file
a = x*y*z;
% return last value of return variable
% in file Matlab functions can return an
% array
```

Hàm tự tạo (m-file)

Hàm tự tạo (m file)

Matlab Functions (m-functions)

- M-functions giống khái niệm “hàm” trong các ngôn ngữ lập trình

The screenshot shows the Matlab Editor window with the file `cylinder.m` open. The code defines a function to calculate the volume of a cylinder. Below the editor, the command window shows the execution of the function with input arguments `(1, 10)`, resulting in the output `ans = 31.4159`. A yellow box highlights the command in the window, and a red box contains the text "Phân biệt function và script!".

```
C:\matlabR12\work\cylinder.m
File Edit View Text Debug Breakpoints Web Window Help
 1 function volume=cylinder(radius, length)
 2 % CYLINDER computes volume of circular cylinder given radius and
 3 % Use:
 4 % vol=cylinder(radius, length)
 5 %
 6 - volume=pi.*radius.^2.*length;
 7 
```

Ready

```
>> cylinder(1,10)
```

ans =

31.4159

Phân biệt function và script!

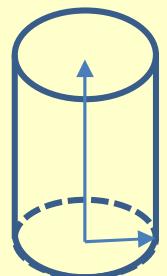
Cấu trúc M-function

Định nghĩa hàm

Biến trả về

Đối số

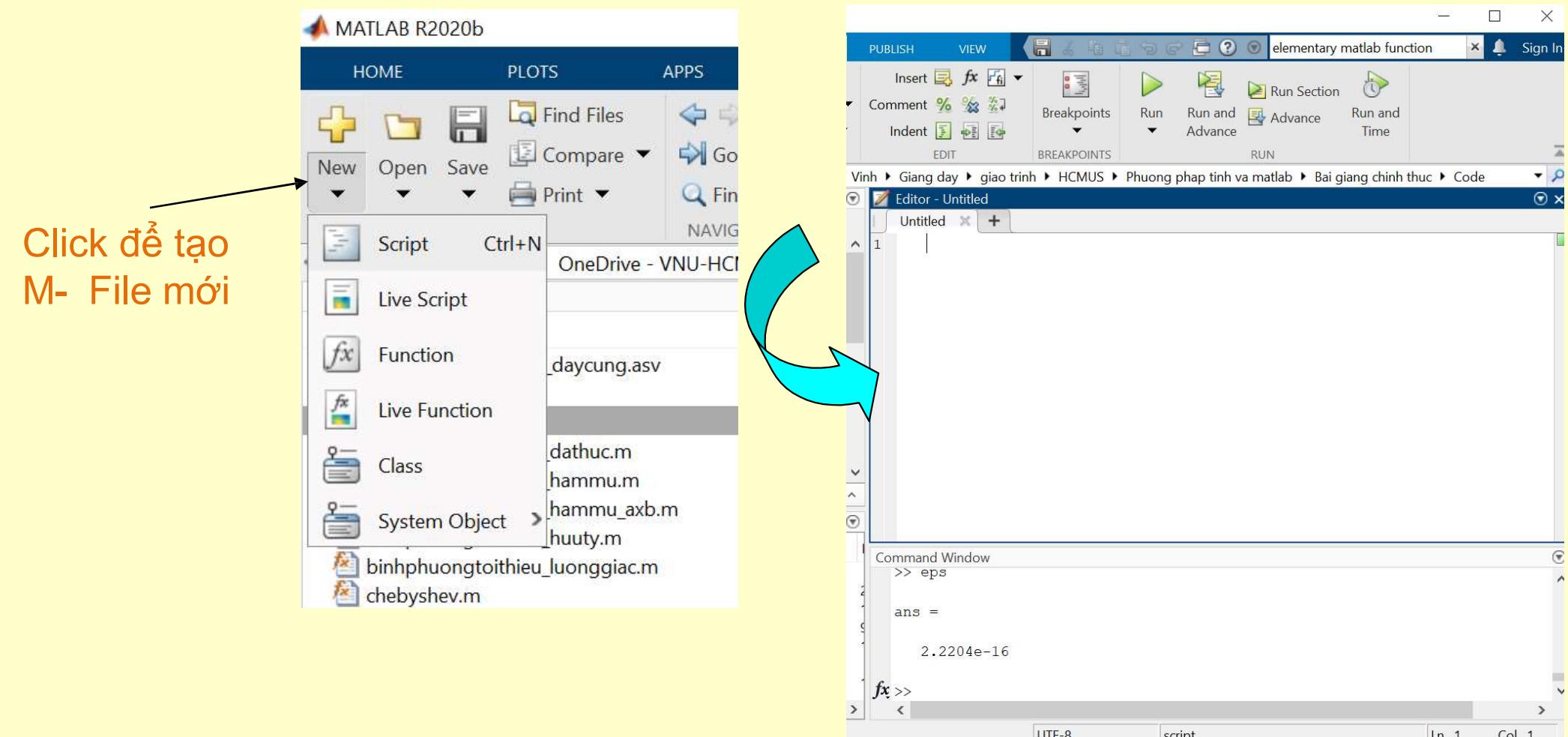
```
function volume=cylinder(radius, length)
    % CYLINDER computes volume of circular cylinder
    % given radius and length
    % Use:
    % vol=cylinder(radius, length)
    %
    volume=pi.*radius^2.*length;
```



Chú thích, xuất hiện
Khi dùng lệnh **help**

Các lệnh
Không cần lệnh **end**

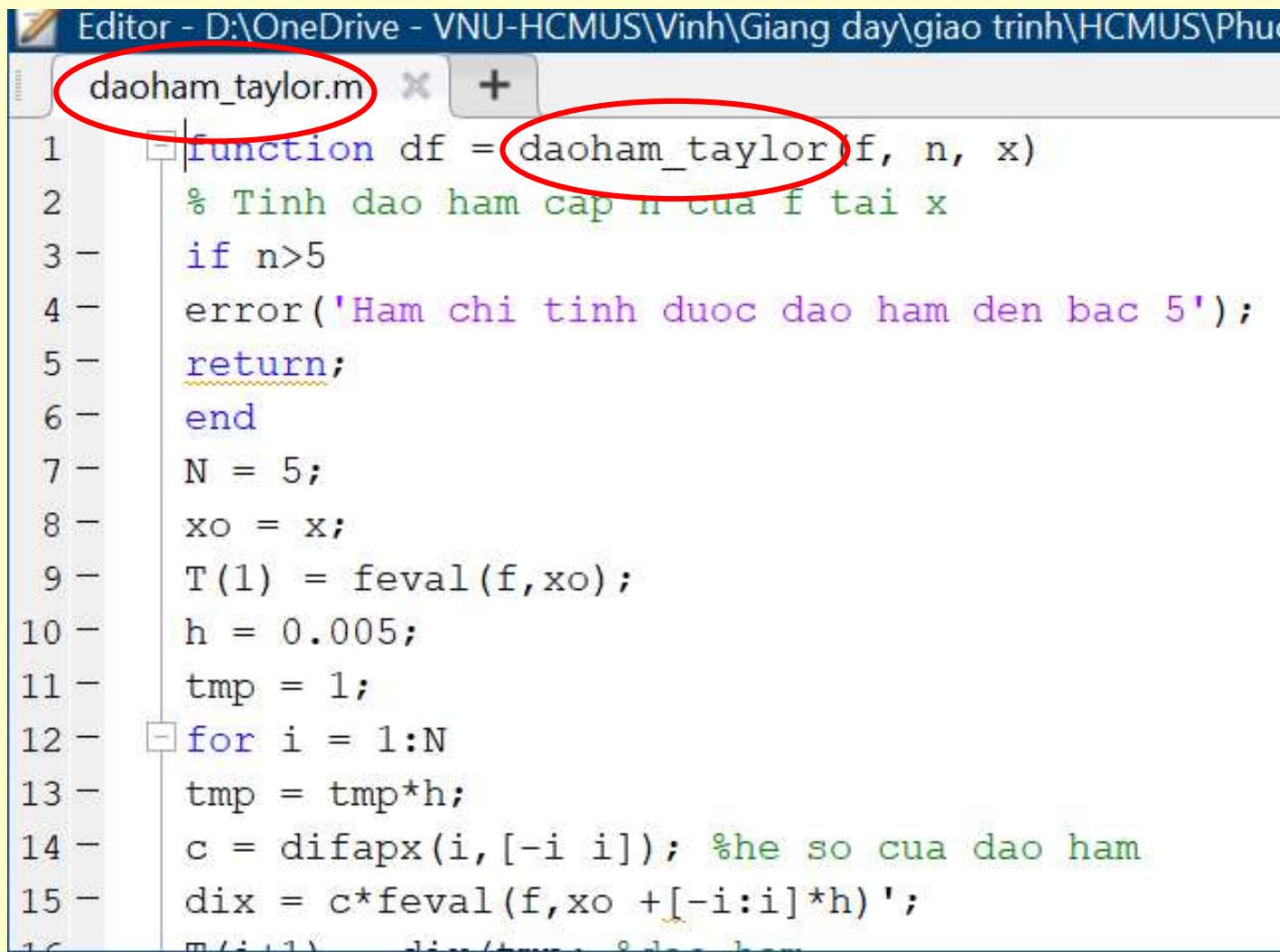
Dùng Editor để soạn thảo m-file



- Phần mở rộng trong tên file mặc định là “.m”
- Nội dung file có thể chứa script hay function hay program

Dùng Editor để soạn thảo m-file

- Tên file và tên hàm cần trùng nhau.



The screenshot shows a MATLAB code editor window. The title bar reads "Editor - D:\OneDrive - VNU-HCMUS\Vinh\Giang day\giao trinh\HCMUS\Phuc...". The current file is "daoham_taylor.m". The code itself is:

```
1 function df = daoham_taylor(f, n, x)
2 % Tinh dao ham cap n cua f tai x
3 if n>5
4 error('Ham chi tinh duoc dao ham den bac 5');
5 return;
6 end
7 N = 5;
8 xo = x;
9 T(1) = feval(f, xo);
10 h = 0.005;
11 tmp = 1;
12 for i = 1:N
13 tmp = tmp*h;
14 c = difapx(i, [-i i]); %he so cua dao ham
15 dix = c*feval(f, xo +[-i:i]*h);
```

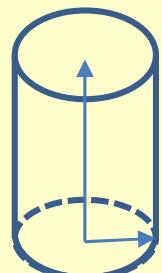
Two parts of the code are highlighted with red circles: the file name "daoham_taylor.m" in the title bar and the function name "daoham_taylor" in the first line of the script.

Biến global (tổng quát)

Editor - D:\OneDrive - VNU-HCMUS\Vinh\Giang day\giao trinh\HCMUS\Phuong phap tinh

cylinderwt.m

```
1 function weight=cylinderwt(radius, length)
2 % help command
3 %
4 global DENSITY
5 weight = DENSITY.*pi.*radius^2.*length;
6
```

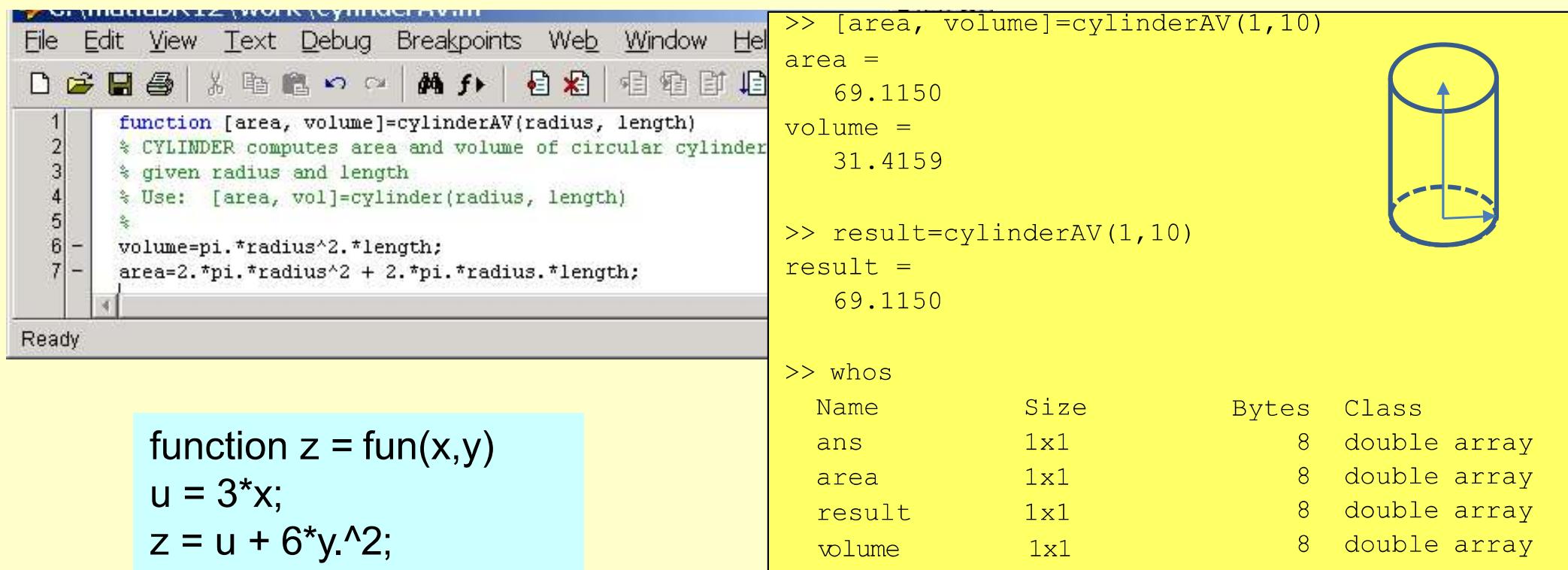


```
>> global DENSITY
>> DENSITY=100;
>> cylinderwt(1,10)

ans =
3.1416e+003
```

M-functions có thể trả về nhiều giá trị

- M-function có thể trả về một giá trị vô hướng hay một mảng
- M-functions dùng cấu trúc [] để trả về nhiều giá trị



The screenshot shows the MATLAB environment with the following components:

- Editor:** Displays the `cylinderAV.m` function code.
- Command Window:** Shows the execution of the function and its results.
- Figure:** A 3D plot of a cylinder with a dashed base, representing the geometric shape being calculated.
- Code Block:** A highlighted section of code in a light blue box:


```
function z = fun(x,y)
u = 3*x;
z = u + 6*y.^2;
```
- Output Block:** A light blue box at the bottom showing the command and its result:


```
>>z = fun(3,7)
z =
303
```

Các cách gọi hàm

1. Gọi tên m-file
2. Dùng function handle
3. Dùng “inline”
4. Dùng biểu thức

Ví dụ hàm fzero gọi hàm tự tạo fun1, $y = x^2 - 4$.

1. Gọi tên m file: Phải có hàm function trước

```
function y = fun1(x)
y = x.^2-4;
```

Để tìm zero để $y=0$ trong đoạn $0 \leq x \leq 3$:

```
>> [x, value] = fzero('fun1', [0, 3])
```

Các cách gọi hàm

2. function handle trả đến M-file:

Viết hàm trước

```
function y = fun1(x)
y = x.^2-4;
>> [x, value] = fzero(@fun1, [0, 3])
```

3. Như một đối tượng hàm “inline”: không cần hàm function

```
>> fun1 = 'x.^2-4';
>> fun_inline = inline(fun1);
>> [x, value] = fzero(fun_inline, [0, 3])
```

4. Như một biểu thức chuỗi ký tự: không cần hàm function

```
>> fun1 = 'x.^2-4';
>> [x, value] = fzero(fun1, [0, 3])
```

Hay

```
>> [x, value] = fzero('x.^2-4', [0, 3])
```

Ví dụ gọi hàm: Tìm giá trị zero của hàm

fzero(@function, x0)

@function là function handle, x0 là giá trị ước đoán.

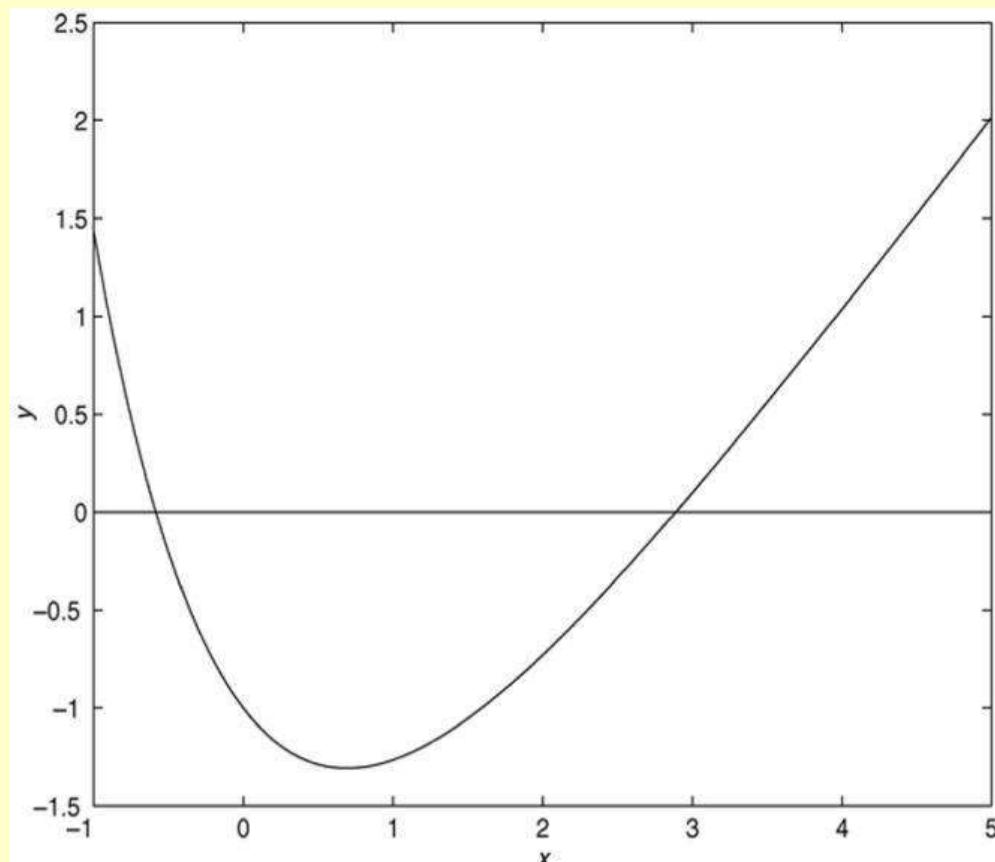
□ Ví dụ tìm nghiệm của $y = x + 2e^{-x} - 3$:

```
function y = f1(x)
```

```
y = x + 2*exp(-x) - 3;
```

```
>>x = fzero(@f1,-0.5)
```

Kết quả: $x = -0.5881$



Ví dụ gọi hàm: Tìm giá trị nhỏ nhất của hàm

- `fminbnd(@function, x1, x2)`

FMINBND Single-variable bounded nonlinear function minimization

- `@function` là function handle

$x_1 \leq x \leq x_2$ là đoạn tìm kiếm

- Ví dụ `fminbnd(@cos,0,4)` trả về giá trị 3.1416.

Hàm ẩn danh

Có thể tạo ra hàm mà không cần tạo m file. Các hàm như vậy gọi là **là ẩn danh anonymous function**. Các hàm này có thể được tạo ra ngay trong dòng lệnh của MATLAB hay trong một hàm khác. **Hàm ẩn danh được dùng thay thế cho cách khai báo hàm inline.**

Cú pháp:

```
fhandle = @(arglist) expr
```

`arglist` là danh sách đối số, cách nhau bởi dấu ','
`expr` là biểu thức hàm.

Hàm ẩn danh

Ví dụ tạo hàm tính bình phương

```
>>sq = @(x) x.^2;
```

Để gọi hàm dùng tên thẻ hàm function handle, kèm danh sách đối số y như gọi hàm thông thường.

Ví dụ:

```
>>sq([5, 7])  
ans =  
    25         49
```

Hàm ẩn danh

Ví dụ để tìm giá trị min của đa thức $4x^2 - 50x + 5$ trong đoạn $[-10, 10]$, có thể làm như sau:

```
>>poly1 = @(x) 4*x.^2 - 50*x + 5;  
>>fminbnd(poly1, -10, 10)  
ans = 6.2500
```

Hoặc không cần khai báo function handle

```
>>fminbnd(@(x) 4*x.^2 - 50*x + 5, -10, 10)
```

Hàm nhiều đối số

Có thể khai báo hàm nhiều đối số như ví dụ sau:

$$x^2 + y^2$$

```
>>sqrtsum = @(x,y) sqrt(x.^2 + y.^2);
```

Để truyền đối số cho hàm, dùng cách sau:

```
>>sqrtsum(3, 4)
```

```
ans =
```

```
5
```

Ví dụ khác:

```
>>A = 6; B = 4;
```

```
>>plane = @(x,y) A*x + B*y;
```

```
>>z = plane(2,8)
```

```
z = 44
```

Gọi hàm trong hàm khác

Ví dụ: hàm $h(x) = 5 \cdot \sin(x^3)$ gồm $g(y) = 5 \cdot \sin(y)$ và $y = x^3$

```
>>y = @(x) x.^3;  
>>g = @(x) 5*sin(x);  
>>h = @(x) g(y(x));  
>>h(2)  
ans =  
4.9468
```

Lưu ý: khi tạo hàm, ví dụ: $\text{plane} = @(x, y) A*x + B*y$.
A và B sẽ được lấy giá trị tại thời điểm tạo hàm và được duy trì theo function handle.
Sau khi hàm được tạo ra, nếu A và B thay đổi thì không ảnh hưởng tới giá trị này trong hàm

So sánh m script và m function

1. Đều được lưu dưới dạng file có phần mở rộng .m
2. Dòng lệnh đầu tiên trong m function là định nghĩa hàm
3. Biến trong m function là biến cục bộ, biến trong m script có thể dùng trong Command Window
4. Script có thể dùng các biến đã được định nghĩa trong workspace
5. Script file có thể chứa các lệnh và hàm matlab
6. Các function nhận dữ liệu qua đối số vào và trả về dữ liệu qua đối số ra
7. Tên của function nên trùng với tên m file

Truyền giá trị cho biến trong script

- Khi script file chạy các biến sử dụng trong script phải được gán giá trị. Nói cách khác các biến phải được liệt kê trong workspace.
- Có ba cách để gán giá trị cho biến

Các cách gán giá trị cho biến

- Cách 1: Biến được định nghĩa và gán giá trị ngay trong script file

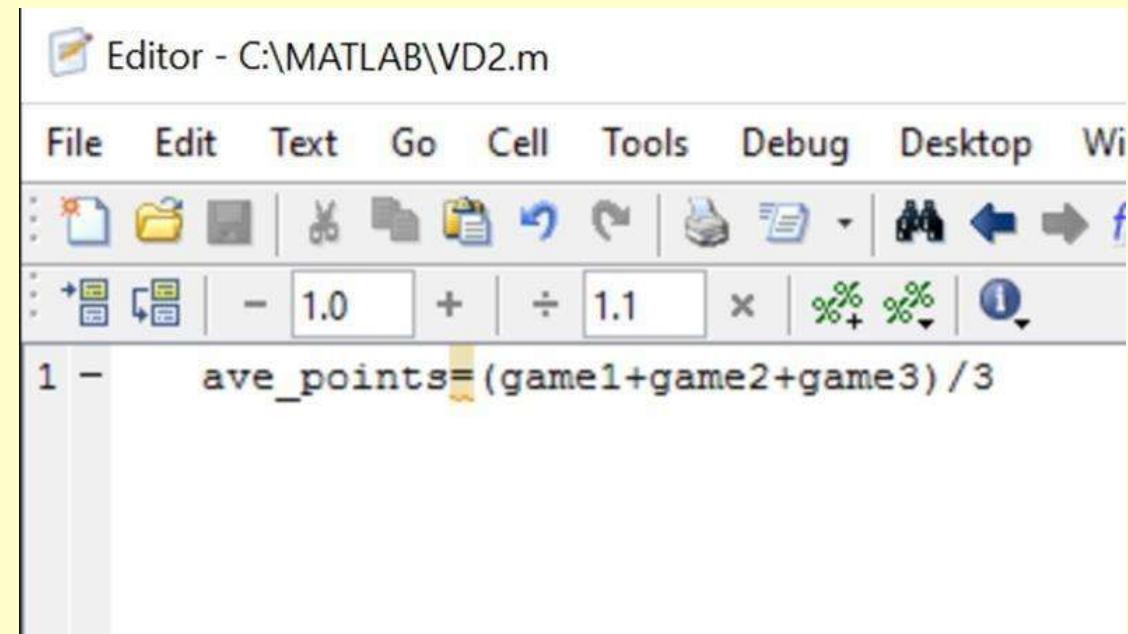
```
game1=75;    game2=93;
```

```
game3=68;
```

```
ave_points=(game1+game2+game3)/3
```

- Cách 2: Biến được định nghĩa và gán giá trị trong cửa sổ lệnh

```
>> game1=3;  
>> game2=13;  
>> game3=10;  
>> VD2  
  
ave_points =  
  
8.6667
```



Các cách gán giá trị cho biến

- Cách 3: Biến được định nghĩa trong script, giá trị được nhập thông qua cửa sổ lệnh.

The screenshot shows the MATLAB Editor window with the title "Editor - C:\MATLAB\VD3.m". The menu bar includes File, Edit, Text, Go, Cell, Tools, Debug, Desktop, Window, and Help. The toolbar below has various icons for file operations like Open, Save, and Print. Below the toolbar is a numeric toolbar with buttons for arithmetic operations: subtraction (-), addition (+), division (÷), multiplication (×), and percentage (%). The code in the editor is:

```
1 - game1=input('Enter the points scored in the first game ');
2 - game2=input('Enter the points scored in the first game ');
3 - game3=input('Enter the points scored in the first game ');
4 - ave_points=(game1+game2+game3)/3
```

```
>> VD3
Enter the points scored in the first game 1
Enter the points scored in the first game 3
Enter the points scored in the first game 4

ave_points =
5.6667
```

Các hàm lồng nhau

```
function y=A(a1,a2)
.....
    function z=B(b1,b2)
        .....
    end
.....
end
```

B có thể truy xuất workspace của A và
ngược lại
B có thể gọi A, A có thể gọi B

```
function y=A(a1,a2)
.....
    function z=B(b1,b2)
        .....
    end
.....
function w=C(c1,c2)
.....
end
.....
```

A,B,C có thể truy xuất workspace của nhau.
A,B,C có thể gọi lẫn nhau.

Lưu ý phải có lệnh **end** khi khai báo cáo hàm con. Lệnh **end** có thể bỏ qua nếu không khai báo hàm con

Hàm con nhiều cấp

```
function y=A(a1,a2)
.....
    function z=B(b1,b2)
        .....
            function w=C(c1,c2)
                .....
                    end
            end
        end
    function u=D(d1,d2)
        .....
            function h=E(e1,e2)
                .....
                    end
            end
        end
    .....
end
```

- Hàm con có thể được gọi từ hàm mẹ trên 1 cấp, function A có thể gọi B hay D, không gọi C hay E.)
- Hàm con có thể được gọi từ hàm con khác cùng cấp, B có thể gọi D và ngược lại
- Hàm con có thể được gọi từ các hàm con khác lồng trong nó
- Biến được định nghĩa trong hàm chính có thể được định nghĩa lại ở trong bất kỳ hàm con nào
- Biến được định nghĩa trong hàm con có thể được sử dụng hay định nghĩa lại trong các hàm có chứa hàm con đó

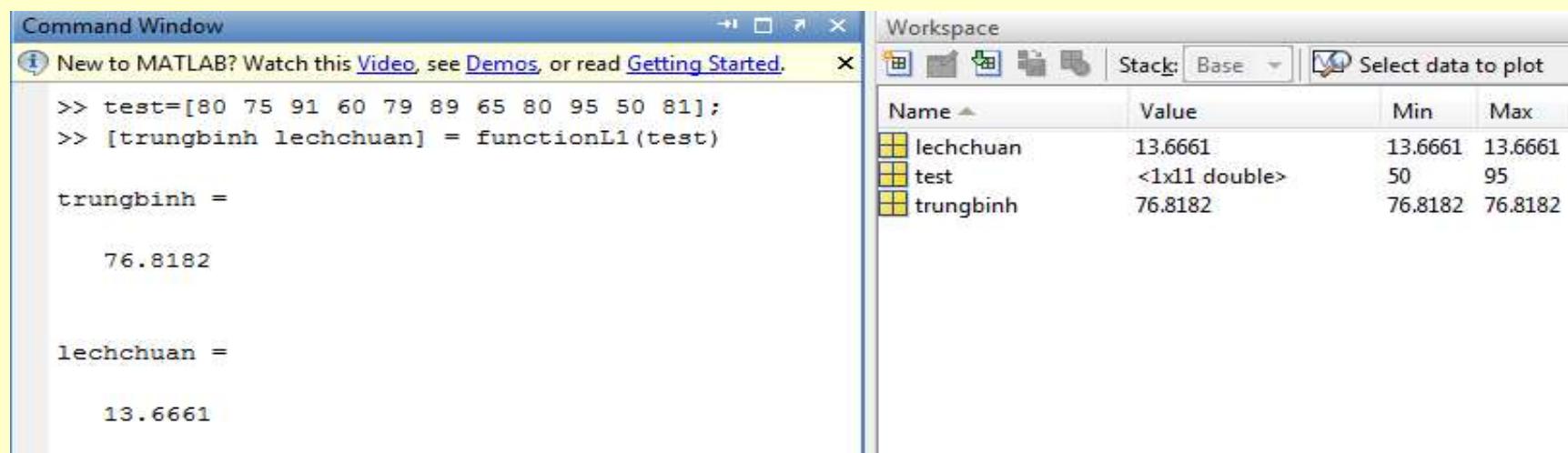
Ví dụ

```

function SD=lechchuan(x) %std(x)
n=length(x);
%Tinh gia tri mean
    function av=functionL2a(x)
        av=sum(x)/n;
    end
    function Sdiv=functionL2b(x)
        xdif=x-functionL2a(x);
        xdif2=xdif.^2;
        Sdiv= sqrt(sum(xdif2)/(n-1));
    end
%Tinh standard deviation
SD=functionL2b(x);
end

```

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$



Bài tập Matlab – Lần 1

File BaiTapMatlab1New.pdf (10%)

Mỗi SV làm 3 bài tập: m-file

Bài tập 1: Từ bài tập số 1 đến bài tập số 10.

Số thứ tự bài tập là số cuối MSSV (số 0 = 10)

Bài tập 2: Từ bài tập số 11 đến bài tập số 15.

Số thứ tự bài tập là = $11 + \text{làm tròn xuống} (\text{số kể cuối MSSV}/2)$

(số 0 => 11, số 9 => 15)

Bài tập 3: Từ bài tập số 16 đến bài tập số 25

Số thứ tự bài tập là = $16 + \text{số cuối MSSV} (\text{số 0} => 16)$

Nộp tại link google form chia sẻ trên group lớp. Nộp file nén (zip, rar)

Cú pháp m-file::: **BTsoTT.m** (3 bài => 3 file .m)

Tạo thư mục **HovaTen_MSSV_Lop_BTVN_lan1** chép các file m vào thư mục, nén thành file zip/rar theo cú pháp: **HovaTen_MSSV_Lop_BTVN_lan1.zip**

Link google form: đăng trên group lớp.

Thời gian nộp: **03/10/2022**

CHƯƠNG 4

Đồ thị trong Matlab

Mục tiêu

Khảo sát các cách biểu diễn đồ thị 2D trong Matlab

Chủ đề

- Vẽ đồ thị 2D
- Các kiểu đồ thị 2D
- Ví dụ...

Hàm plot (vẽ 2D cơ bản)

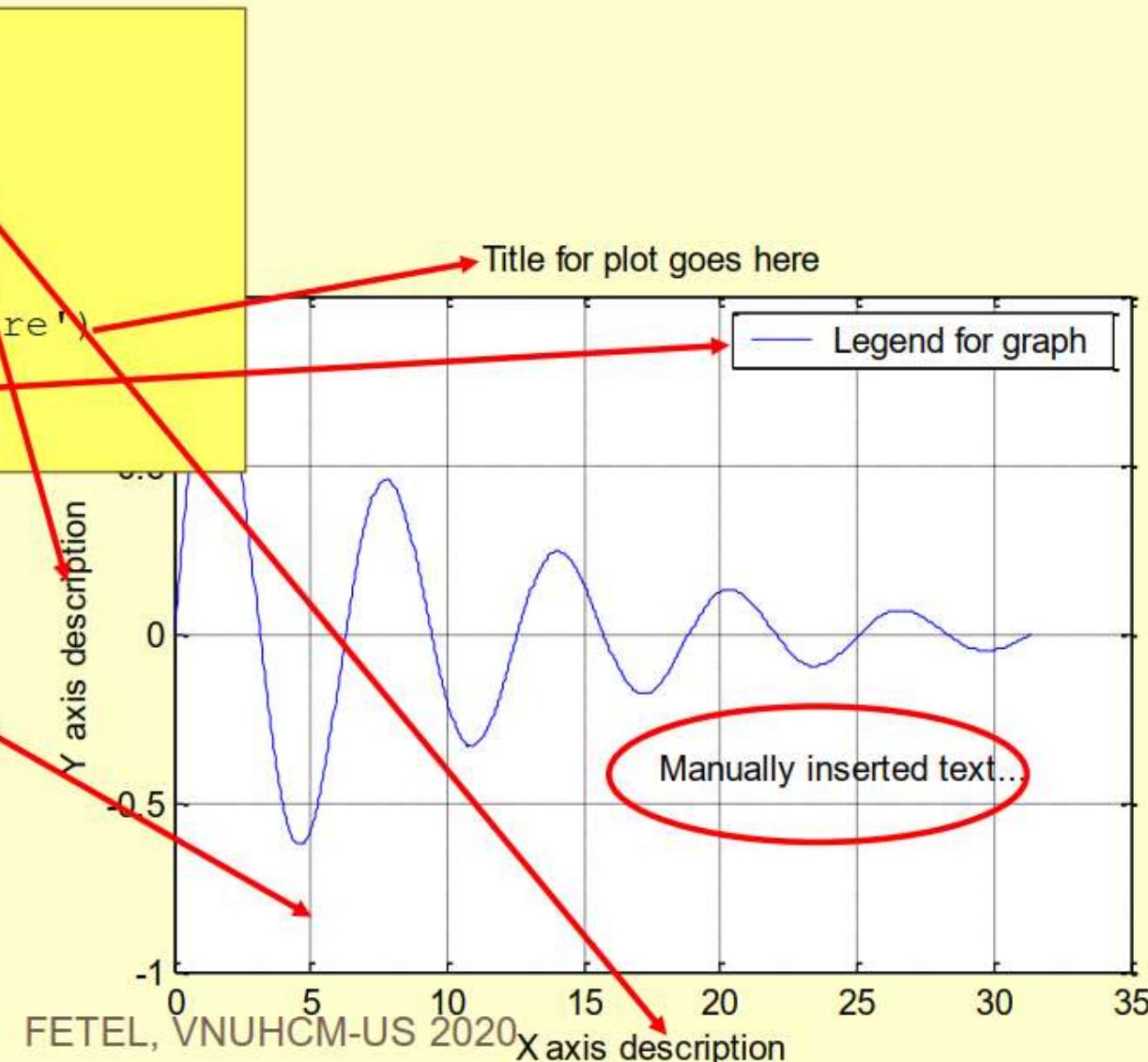
```
>> x=0:0.05:10*pi;
>> y=exp(-.1.*x).*sin(x);
>> plot(x,y)
>> xlabel('X axis description')
>> ylabel('Y axis description')
>> title('Title for plot goes here')
>> legend('Legend for graph')
>> grid on
```

NOTE #1:

Đảo thứ tự (x,y) thành
(y,x) sẽ quay đồ thị 90^0

NOTE #2:

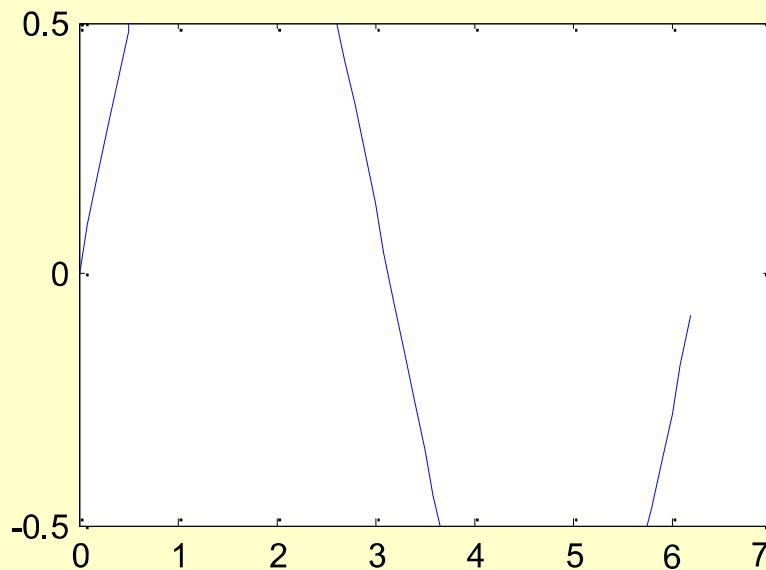
line(x, y) giống plot(x, y)
nhưng sẽ không có các lựa
chọn phụ



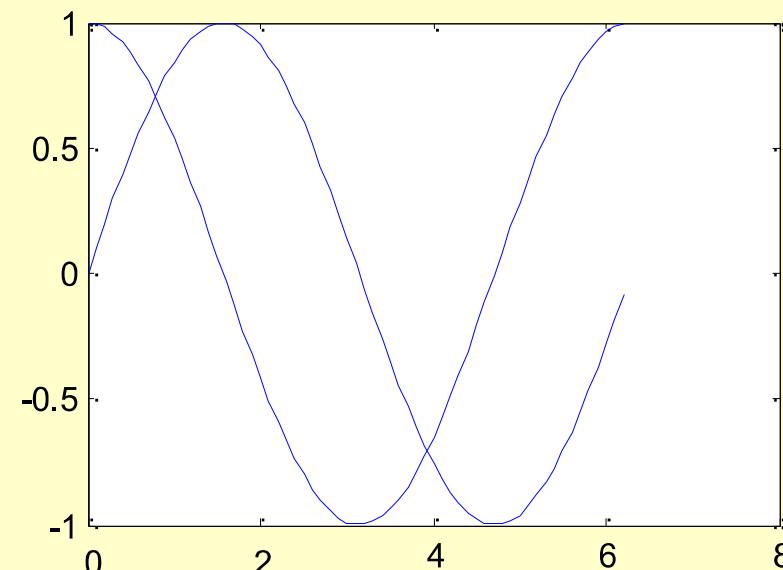
Các hàm hỗ trợ vẽ

- `axis()` xác định phạm vi hiển thị
- `hold on/off` cho phép vẽ thêm các đường mới trên cùng đồ thị

```
>> x=0:0.1:2*pi;
>> plot(x,sin(x))
>> axis
ans =
    0      7      -1      1
>> axis([0 7 -.5 .5])
```



```
>> x=0:0.1:2*pi;
>> plot(x,sin(x))
>> hold on
>> plot(x,cos(x))
```



Lines, Markers, Color

Line Style	Specifier
solid (default)	-
dashed	--

Line Style	Specifier
dotted	:
dash-dot	-.

Line Color	Specifier
red	r
green	g
blue	b
cyan	c

Line Color	Specifier
magenta	m
yellow	y
black	k
white	w

Cấu trúc hàm plot

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		

```
plot(x,y)
plot(x,y, 'r')
plot(x,y, '--y')
plot(x,y, '*')
plot(x,y, 'g:d')
```

`plot(x,y, 'line specifiers', 'PropertyName', PropertyValue)`

Vector Vector

(Optional) Specifiers that define the type and color of the line and markers.

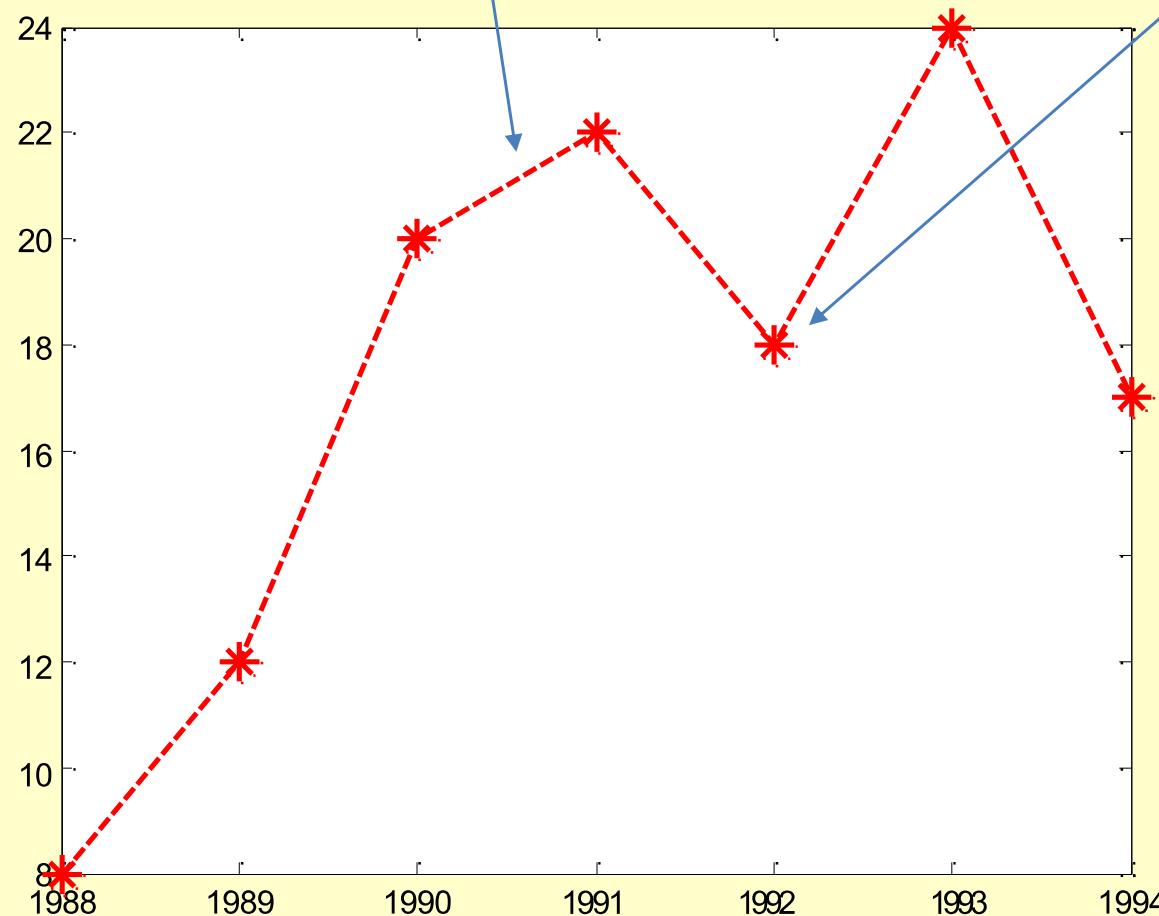
(Optional) Properties with values that can be used to specify the line width, and a marker's size and edge, and fill colors.

Cấu trúc hàm plot

Property name	Description	Possible property values
LineWidth (or linewidth)	Specifies the width of the line.	A number in units of points (default 0.5).
MarkerSize (or markersize)	Specifies the size of the marker.	A number in units of points.
MarkerEdgeColor (or markeredgecolor)	Specifies the color of the marker, or the color of the edge line for filled markers.	Color specifiers from the table above, typed as a string.
MarkerFaceColor (or markerfacecolor)	Specifies the color of the filling for filled markers.	Color specifiers from the table above, typed as a string.

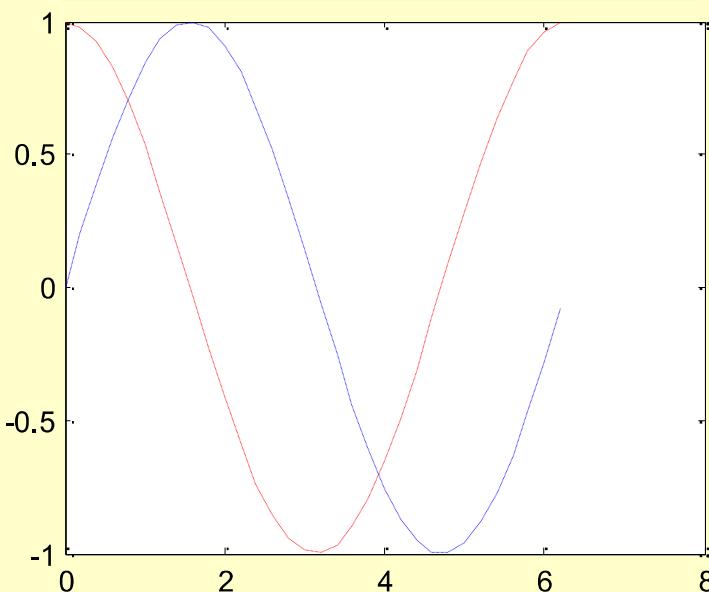
Ví dụ hàm plot

```
>> yr=[1988:1:1994];  
>> sle=[8 12 20 22 18 24 17];  
>> plot(yr, sle,'--r*', 'linewidth', 2, 'markersize', 12)
```

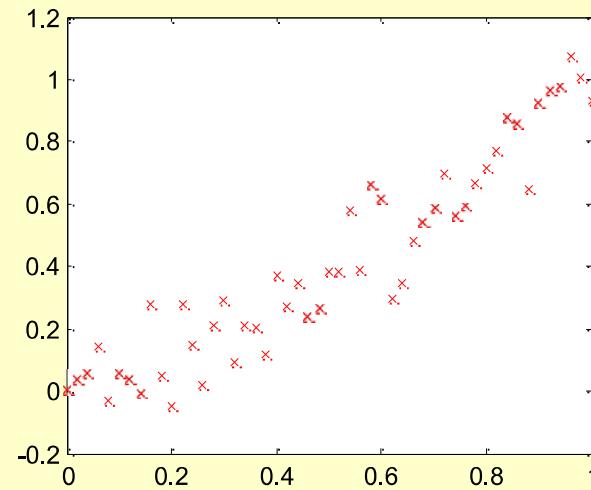


Dùng kết hợp Markers & Lines

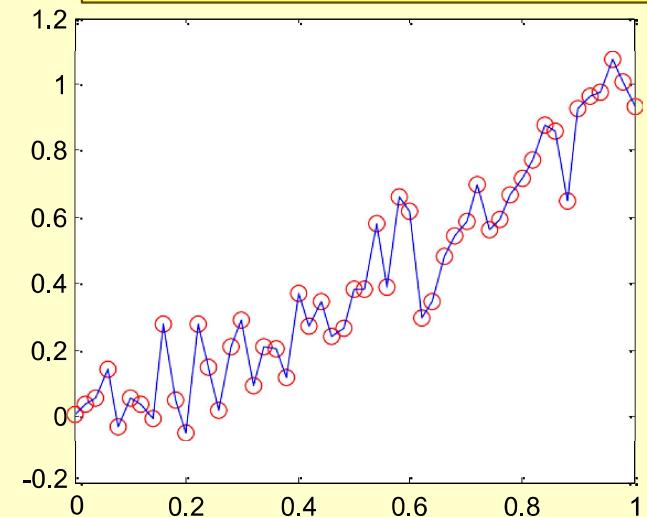
```
>> plot(x,cos(x), 'r:')
>> hold on
>> plot(x,sin(x), 'b--')
```



```
>> x=0:0.02:1;
>> y=x.^1.5;
>> yr=randn(size(x));
>> yy=y+0.1.*yr;
>> plot(x,yy,'ro',x,yy)
```



```
>> x=0:.02:1;
>> y=x.^1.5;
>> yr=randn(size(x));
>> yy=y+0.1.*yr;
>> plot(x,yy,'ro',x,yy)
```

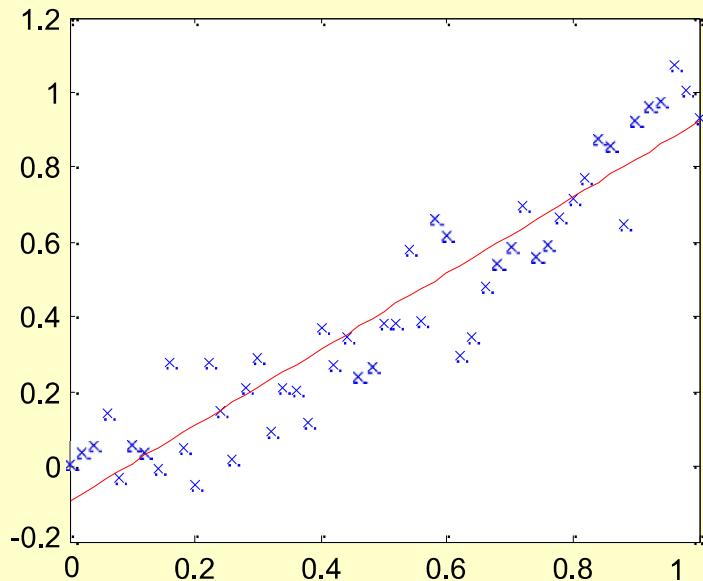


Dùng kết hợp Markers & Lines

```

>> x=0:.02:1;
>> y=x.^1.5;
>> yr=randn(size(x));
>> yy=y+0.1.*yr;
>> plot(x,yy,'x')
>> p=polyfit(x,yy,1)
p =
    1.0159   -0.0927
>> hold on
>> plot(x,polyval(p,x), 'r')

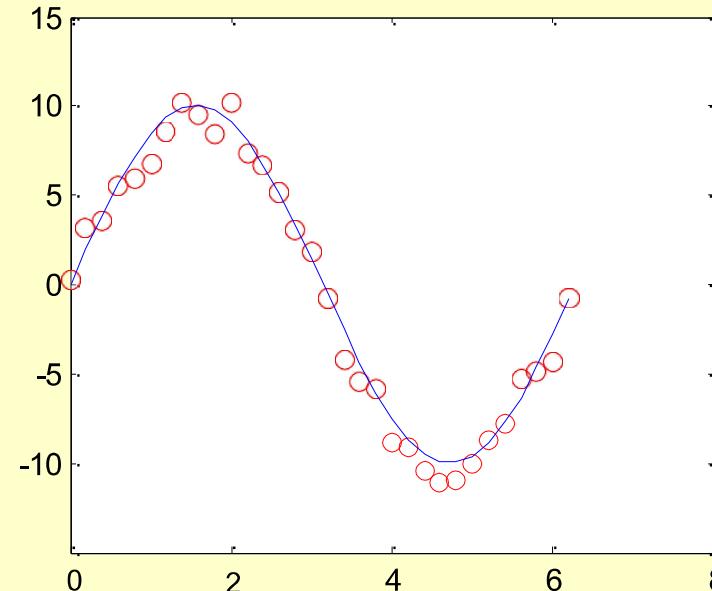
```



```

>> x=0:0.2:2.*pi;
>> y=sin(x);
>> yr=randn(size(x));
>> plot(x,10.*y+yr,'ro')
>> hold on
>> plot(x,10.*y)

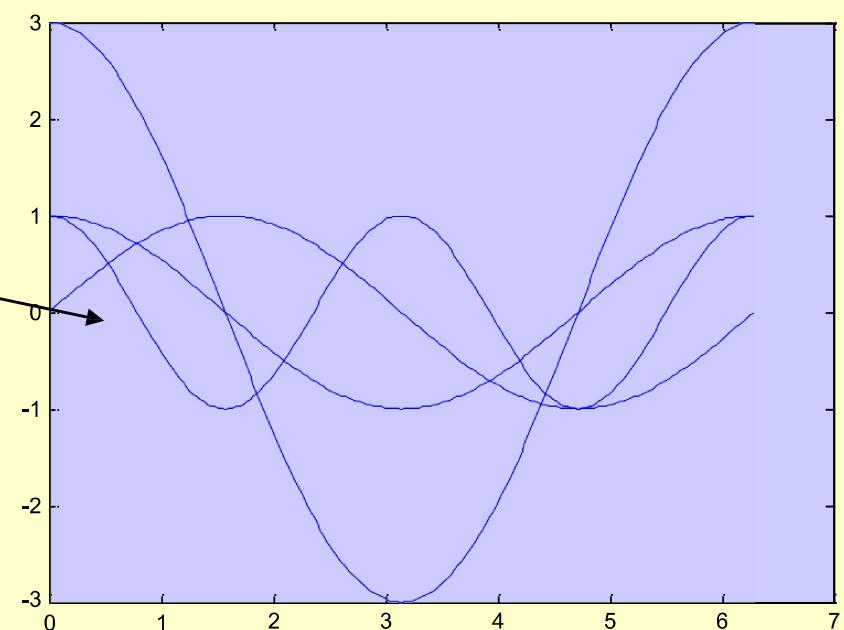
```



Vẽ nhiều đường

```
>> X = 0.0:pi/100:2*pi;  
>> Y1 = cos(X);  
>> Y2 = 3*cos(X);  
>> Y3 = cos(2*X);  
>> Y4 = sin(X);
```

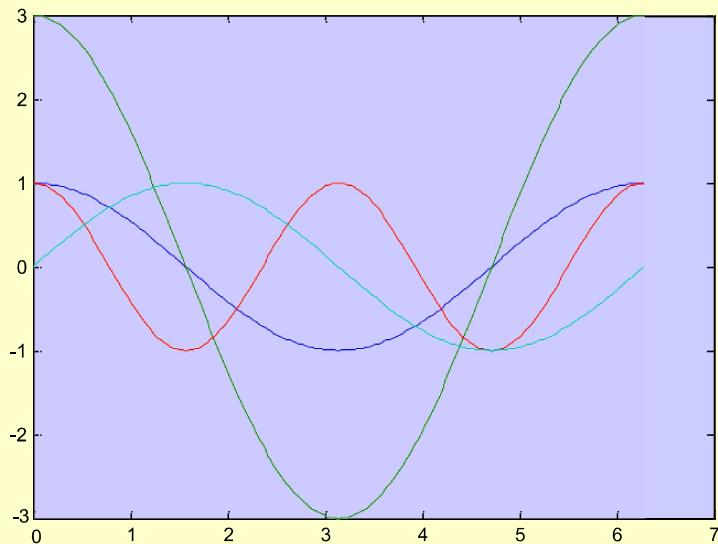
```
>> plot(X,Y1)  
>> hold on  
>> plot(X,Y2)  
>> plot(X,Y3)  
>> plot(X,Y4)
```



Vẽ nhiều đường

- Có thể dùng cách này:

```
>> plot(X, Y1, X, Y2, X, Y3, X, Y4)
```



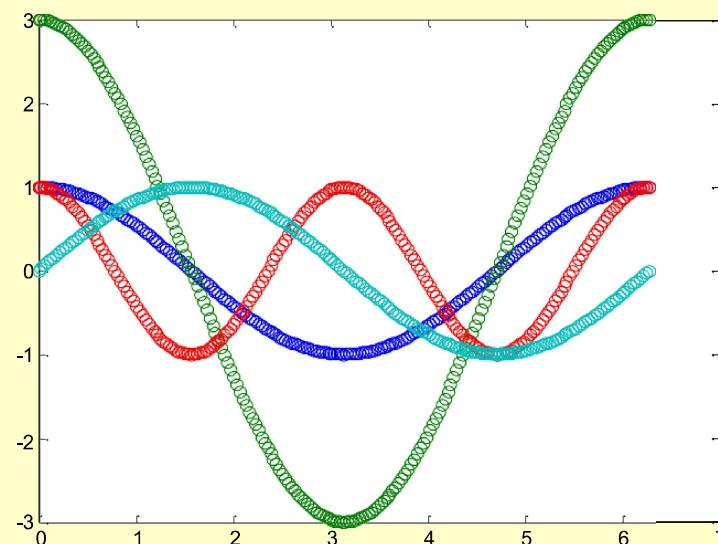
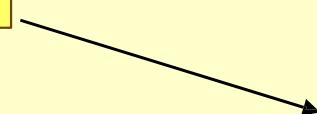
- Hoặc cách này

```
>> Z = [Y1;Y2;Y3;Y4];
>> plot(X, Z)
```



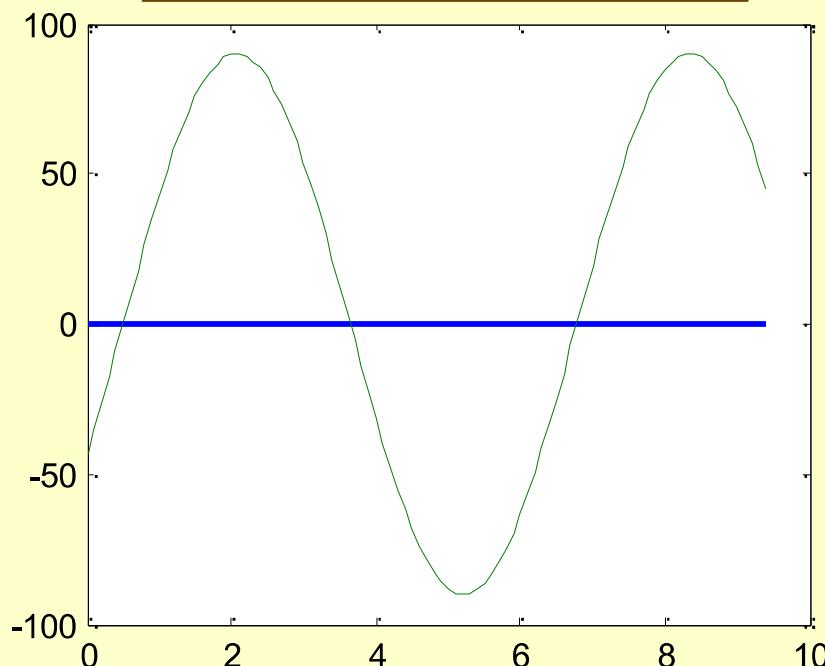
- Nếu dùng câu lệnh như dưới thì sao?

```
>> plot(X, Z, 'o')
```

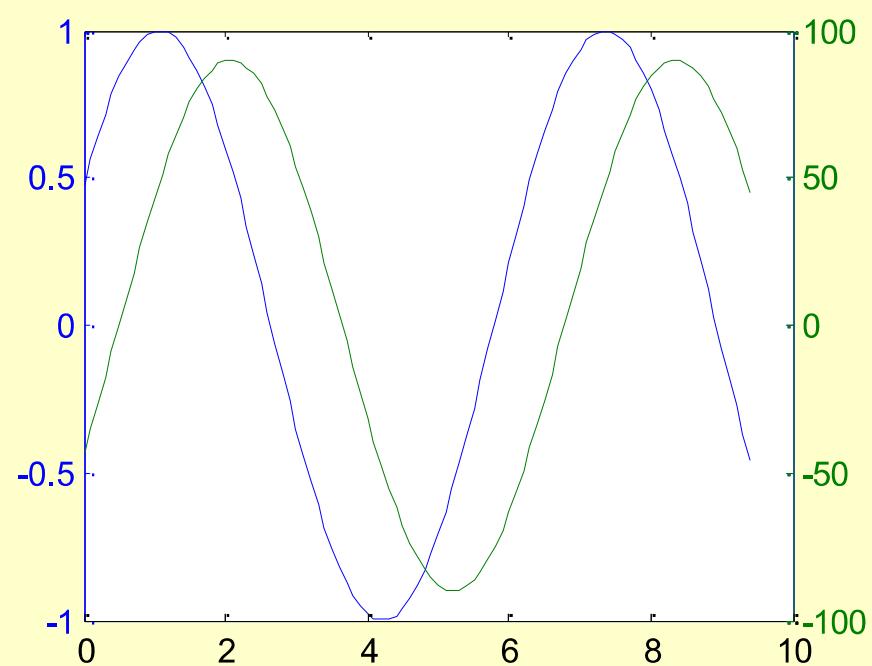


Vẽ 2 đường với hai trục Y

```
>> x=0:0.1:3.*pi;  
>> y1=sin(x+0.5);  
>> y2=90.*sin(x-0.5);  
>> plot(x,y1,x,y2)
```



```
>> x=0:0.1:3.*pi;  
>> y1=sin(x+0.5);  
>> y2=90.*sin(x-0.5);  
>> plotyy(x,y1,x,y2)
```



Một số lệnh liên quan đến đồ thị

- `axis ([xmin, xmax, ymin, ymax])` – Thiết lập giá trị giới hạn trên các trục.
- `axis` – giải phóng giới hạn ở các trục
- `axis off` – turns off display of axes
- `axis on` – turns on display of axes
- `grid on/off` – Tắt/mở hiển thị lưới
- `text (x, y, 'string')` – Hiện chuỗi tại (x,y)
- `gtext ('string')` mouse click
- line (x, y) – Thêm đường biểu diễn xác định bởi x & y vectors

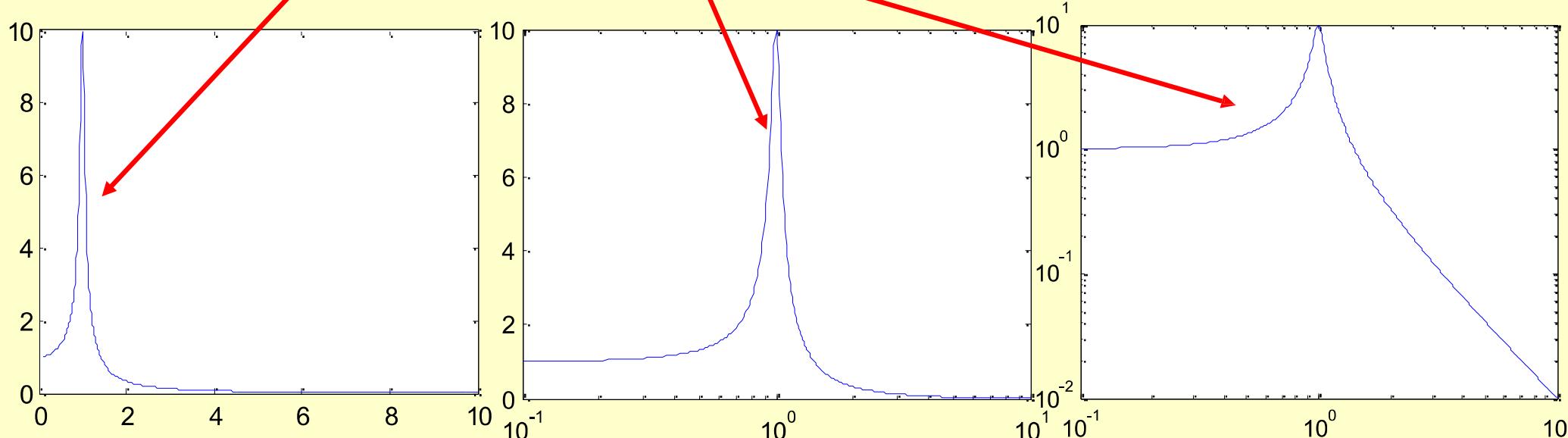
Vẽ đồ thị logarit

```

>> x=linspace (.1,10,1000);
>> damp=0.05;
>> y=1./sqrt((1-x.^2).^2 + (2.*damp.*x).^2);
>> plot(x,y)
>> semilogx(x,y) %log theo x
>> loglog(x,y)   %log theo x, y

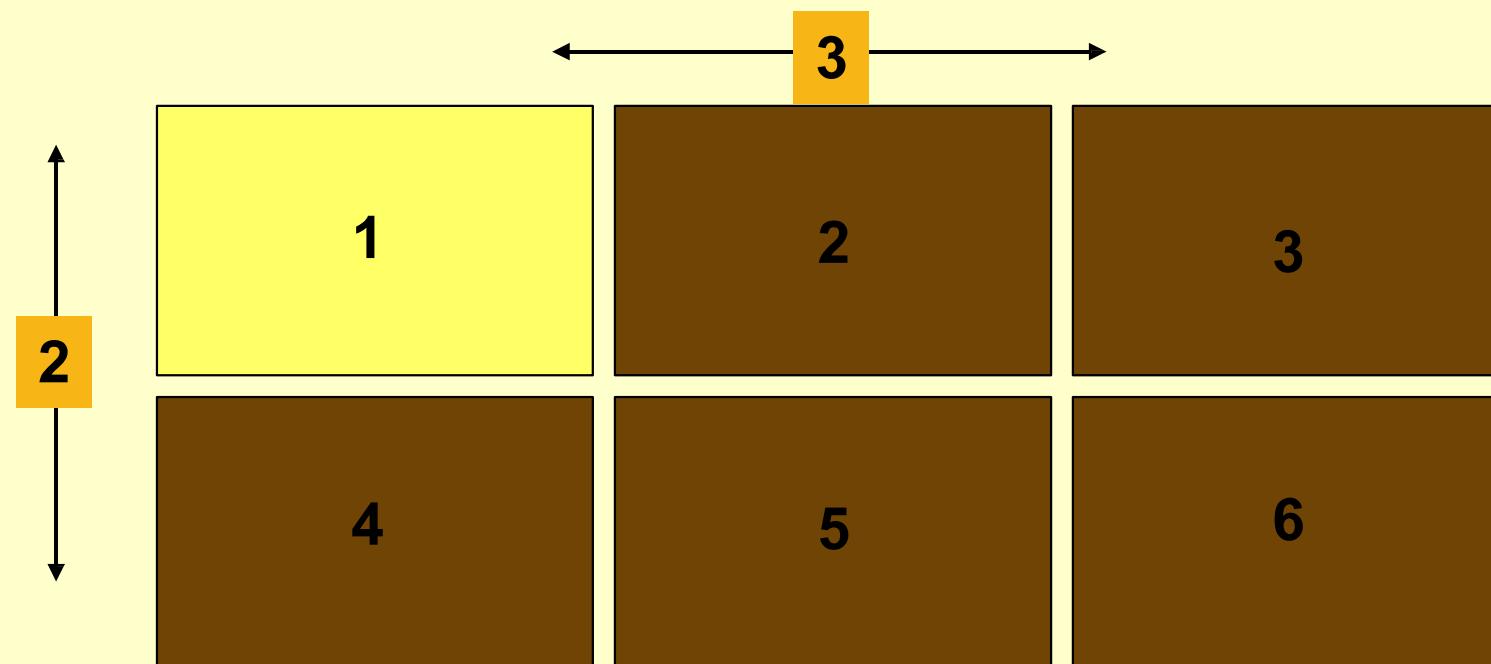
```

$$y = \frac{1}{\left[(1 - x^2)^2 + (2\zeta x)^2 \right]^{1/2}}$$



Vẽ đồ thị trên các cửa sổ con

- subplot (m, n, k)
 - m= hàng, n= cột
 - k= cửa sổ hiện tại
- Ví dụ: subplot (2, 3, 1)

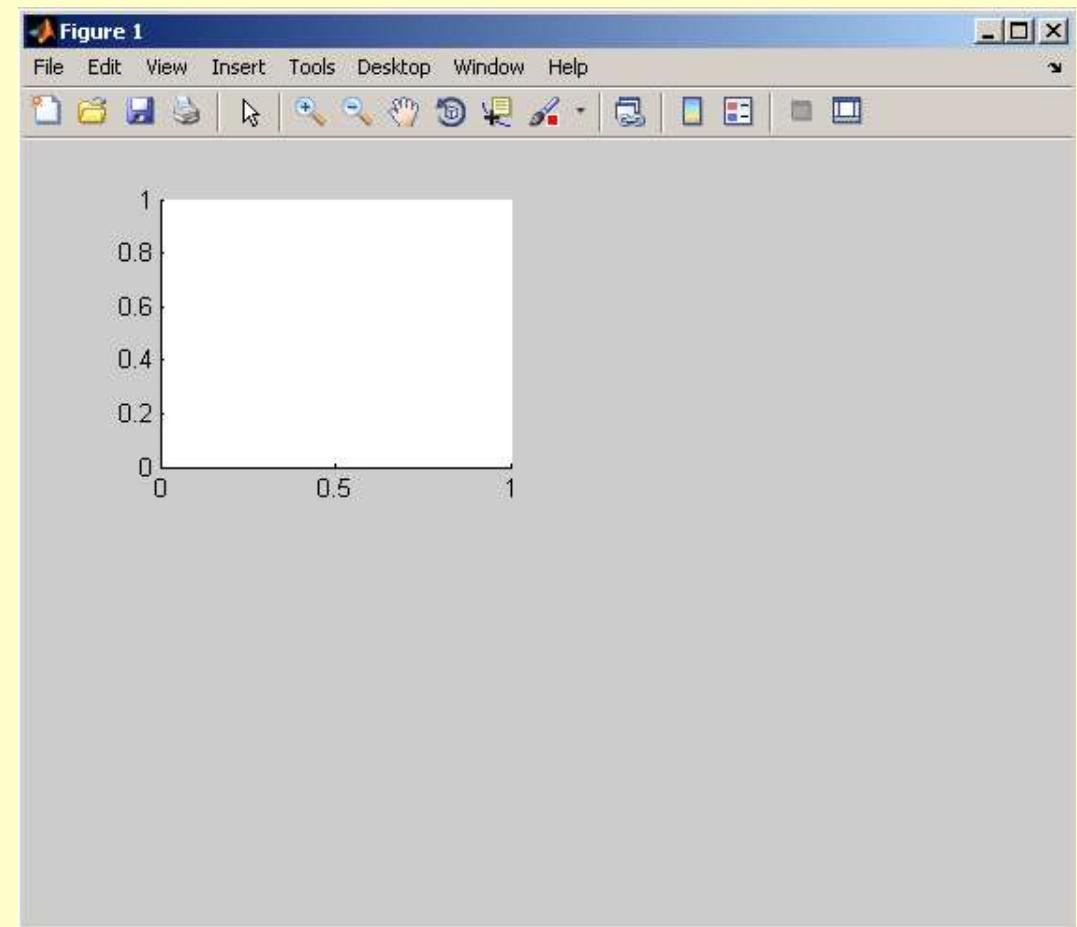


Ví dụ: chạy thử đoạn chương trình sau

```
x=0:0.5:50;  
Y=5*x.^2;  
  
subplot(2,2,1), plot(x,Y), title('Polynomial - Linear/Linear'), ...  
    ylabel('y'), grid  
  
subplot(2,2,2), semilogx(x,Y), title('Polynomial - Log/Linear'), ...  
    ylabel('y'), grid  
  
subplot(2,2,3), semilogy(x,Y), title('Polynomial - Linear/Log'), ...  
    ylabel('y'), grid  
  
subplot(2,2,4), loglog(x,Y), title('Polynomial - Log/Log'), ...  
    ylabel('y'), grid
```

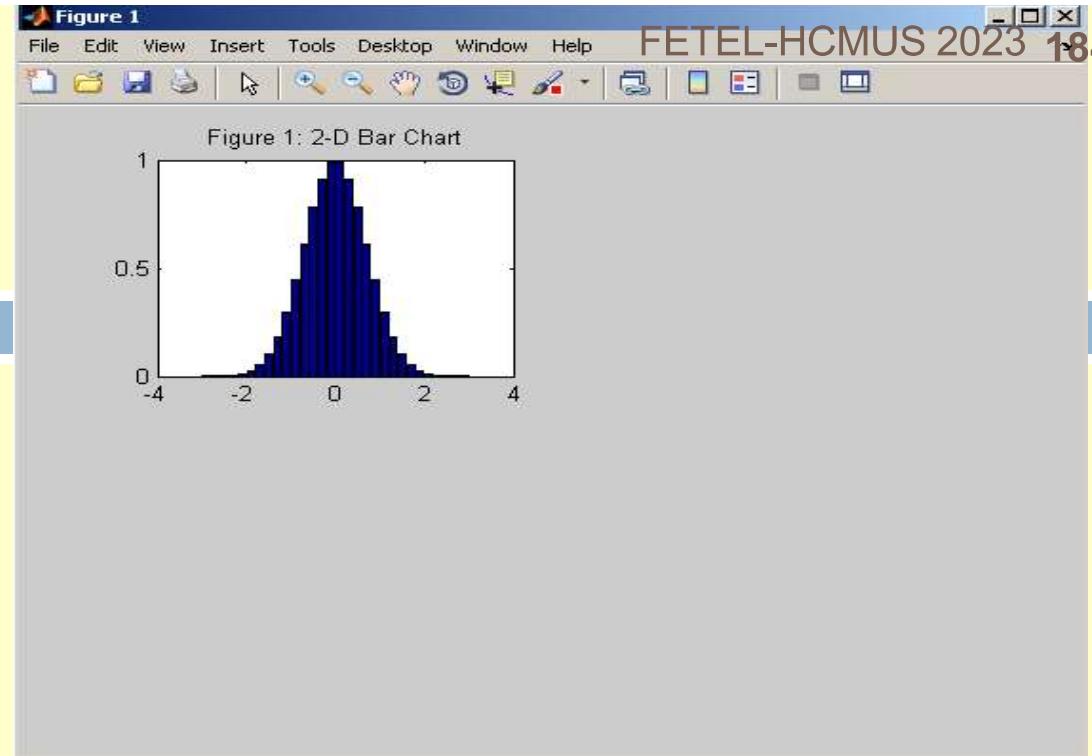
Một số vấn đề khác

```
>> x = -2.9:0.2:2.9;  
>> y = exp(-x.*x);  
>> subplot(2,2,1)
```

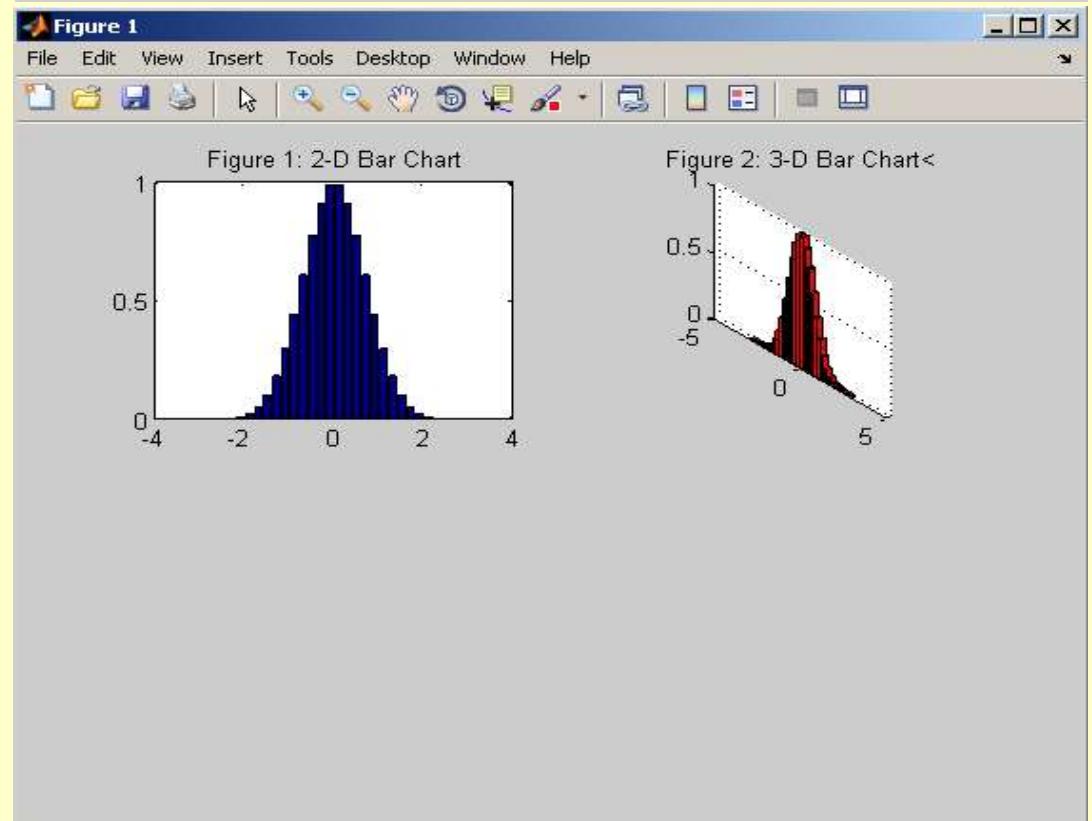


Ví dụ

```
>> bar(x,y)  
>> title('Figure 1: 2-D Bar Chart')
```

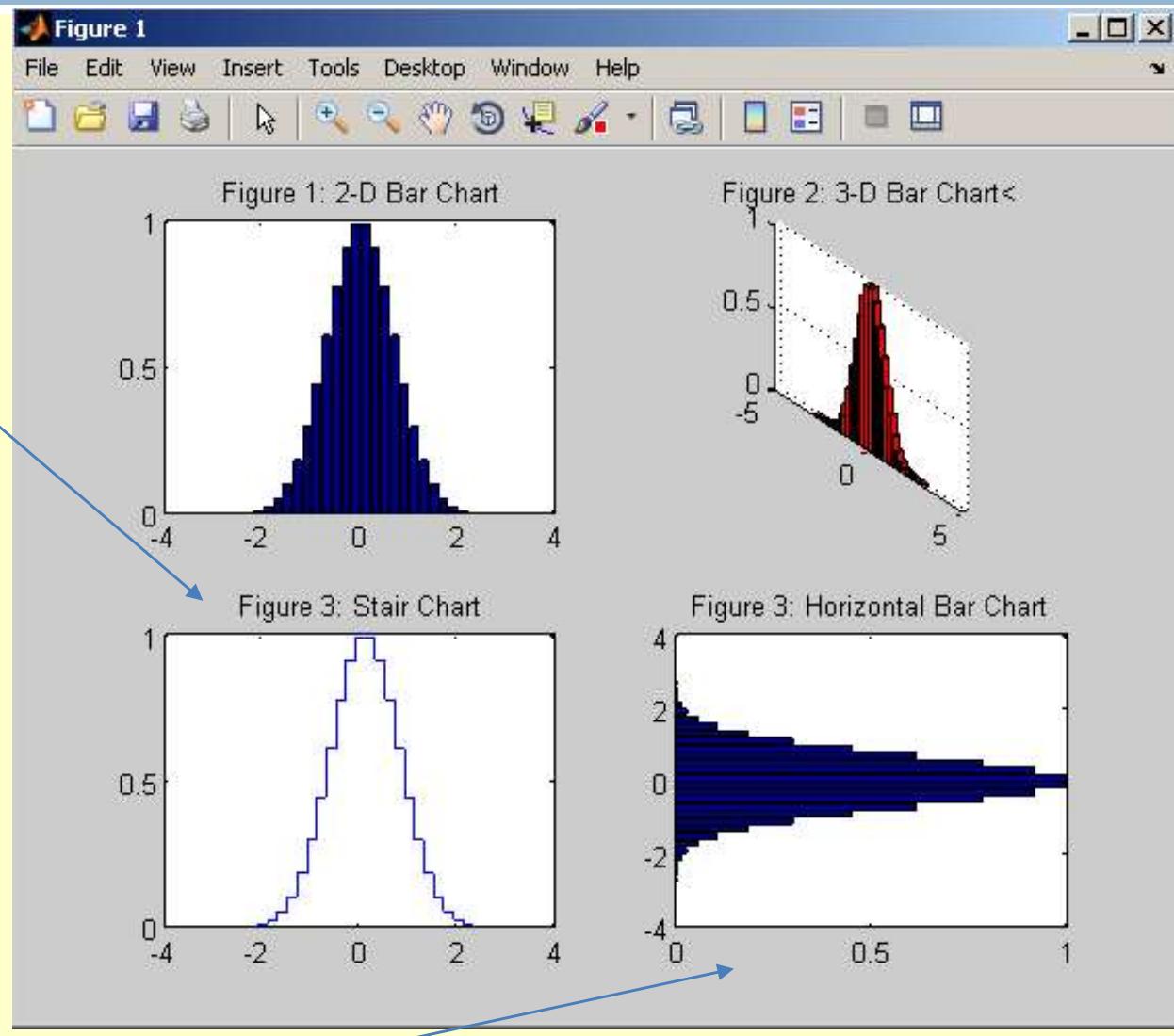


```
>> subplot(2,2,2)  
>> bar3(x,y,'r')  
>> title('Figure 2: 3-D Bar Chart<')
```



Ví dụ

```
>> subplot(2,2,3)  
>> stairs(x,y)  
>> title('Figure 3: Stair Chart')
```



```
>> subplot(2,2,4)  
>> barh(x,y)  
>> title('Figure 4: Horizontal Bar Chart')
```

Tóm lược một số lệnh cơ bản

- **plot** Vẽ đường tuyến tính x-y
- **loglog** Vẽ logarithm scale x-y
- **semilogx** Vẽ bán loga x-y (loga trực x)
- **semilogy** Vẽ bán loga x-y (loga trực y)
- **polar** Vẽ tọa độ cực
- **mesh** Vẽ mặt lưới 3 chiều
- **contour** Vẽ đường mức
- **bar** Vẽ biểu đồ
- **stairs** Vẽ đồ thị bậc thang

Tóm lược một số lệnh cơ bản

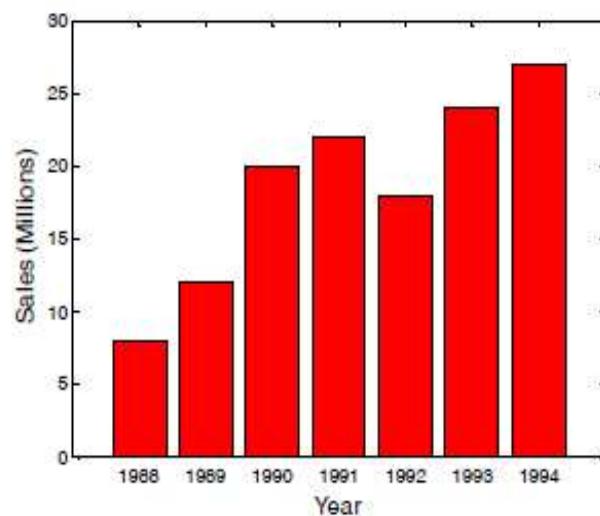
- **title** Tiêu đề đồ thị
- **xlabel** Nhãn trục x
- **ylabel** Nhãn trục y
- **text** Văn bản ở vị trí bất kỳ
- **gtext** Văn bản ở vị trí chuột
- **grid** Các dòng lưới

Ví dụ

Vertical Bar Plot

Function format:

bar(x, y)

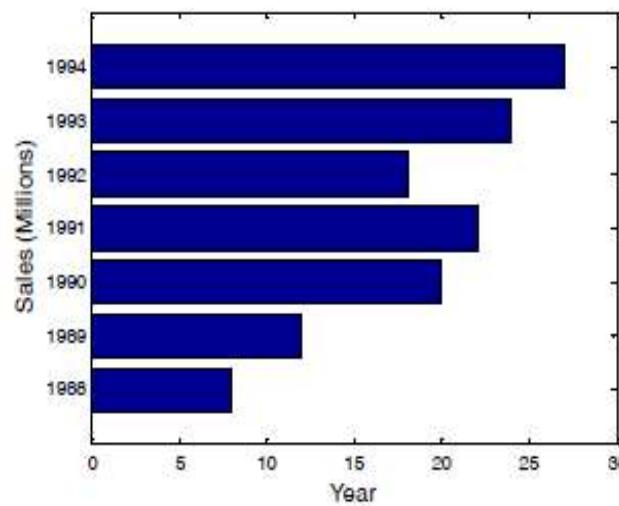


```
yr=[1988:1994];
sle=[8 12 20 22 18 24 27];
bar(yr,sle,'r') ← The
xlabel('Year')
ylabel('Sales (Mil-
lions)')
```

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')
```

Ví dụ

<p><u>Stairs Plot</u></p> <p>Function format: stairs(x, y)</p>		<pre>yr=[1988:1994]; sle=[8 12 20 22 18 24 27]; stairs(yr,sle)</pre>
<p><u>Stem Plot</u></p> <p>Function Format stem(x, y)</p>		<pre>yr=[1988:1994]; sle=[8 12 20 22 18 24 27]; stem(yr,sle)</pre>

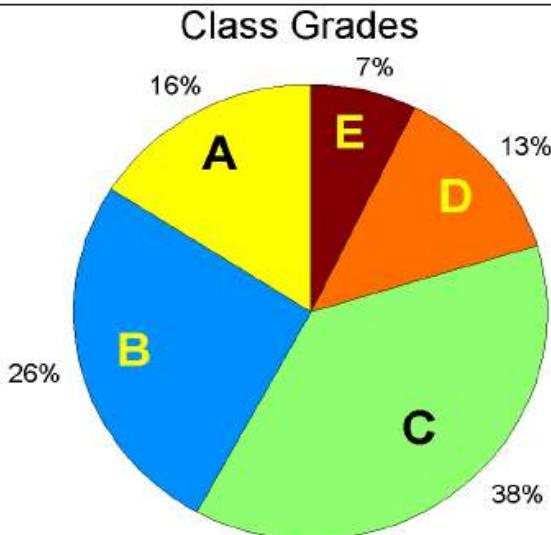
Ví dụ

Grade	A	B	C	D	E
Number of students	11	18	26	9	5

Pie Plot

Function format:

```
pie(x)
```

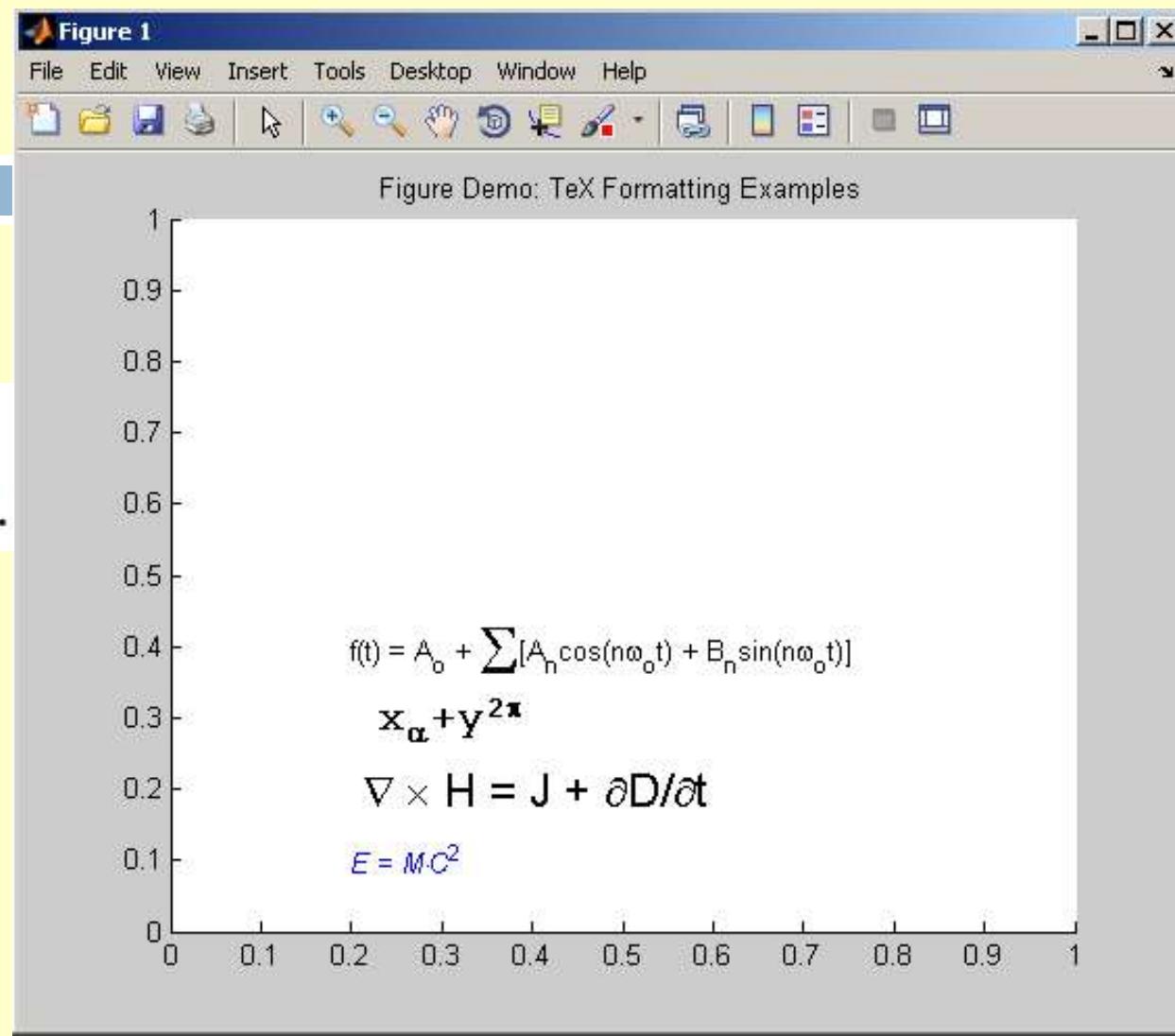


```
grd=[11 18 26 9 5];  
pie(grd)  
title('Class Grades')
```

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

Hiển thị chữ Hy lạp, chỉ số trên/dưới

```
\theta_{12} → θ12.  
\theta^{10} → θ10.
```



```
>> text(0.2,0.1,'color(blue)\itE = M\cdot C^(\rm2)')
>> text(0.2,0.2,'fontsize(16) \nabla \times H = J + \partial D / \partial t')
>> text(0.2,0.3,'fontname(courier)\bf x_{\alpha}+y^{2\pi}')
>> fastr = 'f(t) = A_0 + \text{fontSize}(30)_\Sigma\text{fontSize}(10)';
>> text(0.2,0.4,[fastr '[A_ncos(n\omega_0t) + B_nsing(n\omega_0t)]'])'
>> title('Figure Demo: TeX Formatting Examples')
```

Bảng ký tự Hy Lạp

Character Sequence	Symbol	Character Sequence	Symbol	Character Sequence	Symbol
\alpha	α	\pi	π	\upsilonon	υ
\angle	\angle	\rho	ρ	\phi	Φ
\ast	*	\sigma	σ	\chi	χ
\beta	β	\varsigma	ς	\psi	ψ
\gamma	γ	\tau	τ	\omega	ω
\delta	δ	\equiv	\equiv	\Gamma	Γ
\epsilon	ϵ	\Im	\Im	\Delta	Δ
\zeta	ζ	\otimes	\otimes	\Theta	Θ
\eta	η	\cap	\cap	\Lambda	Λ
\theta	θ	\supset	\supset	\Xi	Ξ
\vartheta	ϑ	\int	\int	\Pi	Π
\iota	ι	\rfloor	\rfloor	\Sigma	Σ
\kappa	κ	\lfloor	\lfloor	\Upsilon	Υ
\lambda	λ	\perp	\perp	\Phi	Φ
\mu	μ	\wedge	\wedge	\Psi	Ψ
\nu	ν	\rceil	\rceil	\Omega	Ω
\xi	ξ	\vee	\vee	\forall	\forall

3D Plotting in Matlab

Plot3(x, y, z)

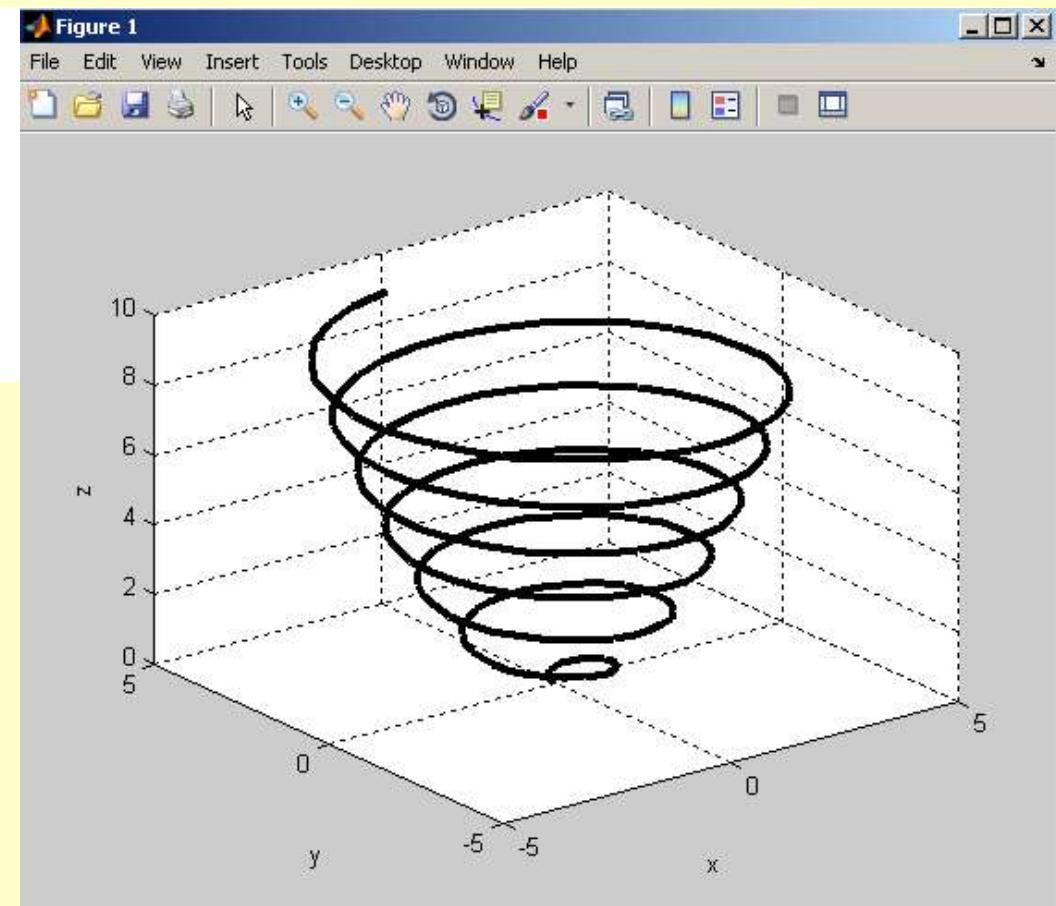
x,y,z là các vector chứa tọa độ điểm

```
t=0:0.1:6*pi;  
x=sqrt(t).*sin(2*t);  
y=sqrt(t).*cos(2*t);  
z=0.5*t;  
plot3(x,y,z,'k','linewidth',3)  
grid on  
xlabel('x'); ylabel('y'); zlabel('z')
```

$$x = \sqrt{t} \sin(2t)$$

$$y = \sqrt{t} \cos(2t)$$

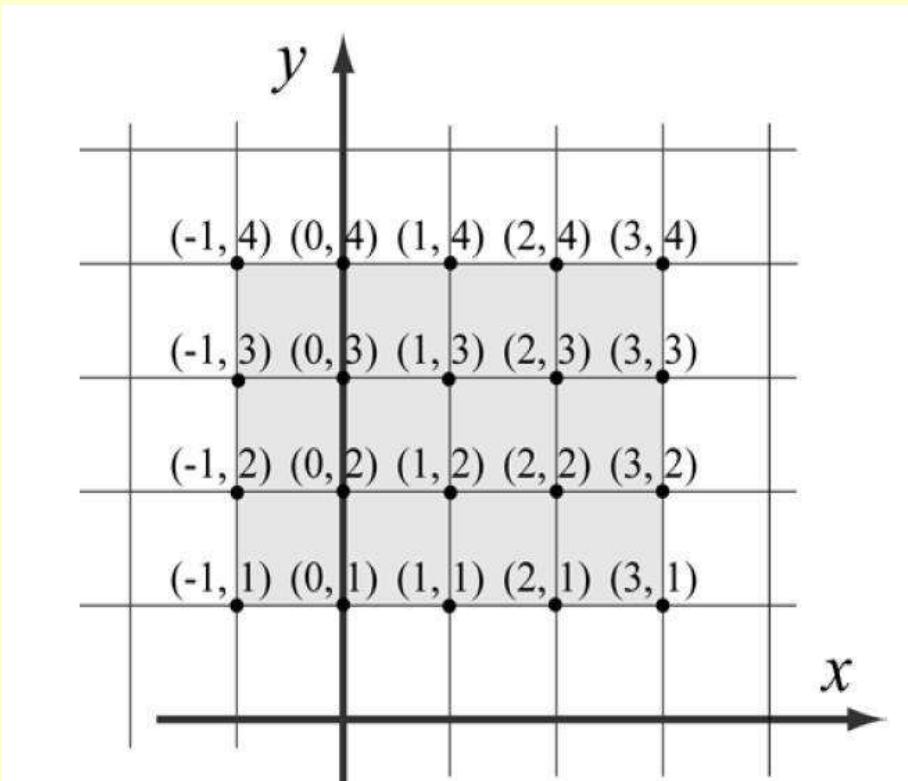
$$z = 0.5t$$



Tạo lưới

- Để tạo một lưới trong hệ tọa độ vuông góc dùng lệnh: meshgrid

```
>> x=-1:3;
>> y=1:4;
>> [X,Y]=meshgrid(x,y)
X =
    -1     0     1     2     3
    -1     0     1     2     3
    -1     0     1     2     3
    -1     0     1     2     3
Y =
    1     1     1     1     1
    2     2     2     2     2
    3     3     3     3     3
    4     4     4     4     4
```

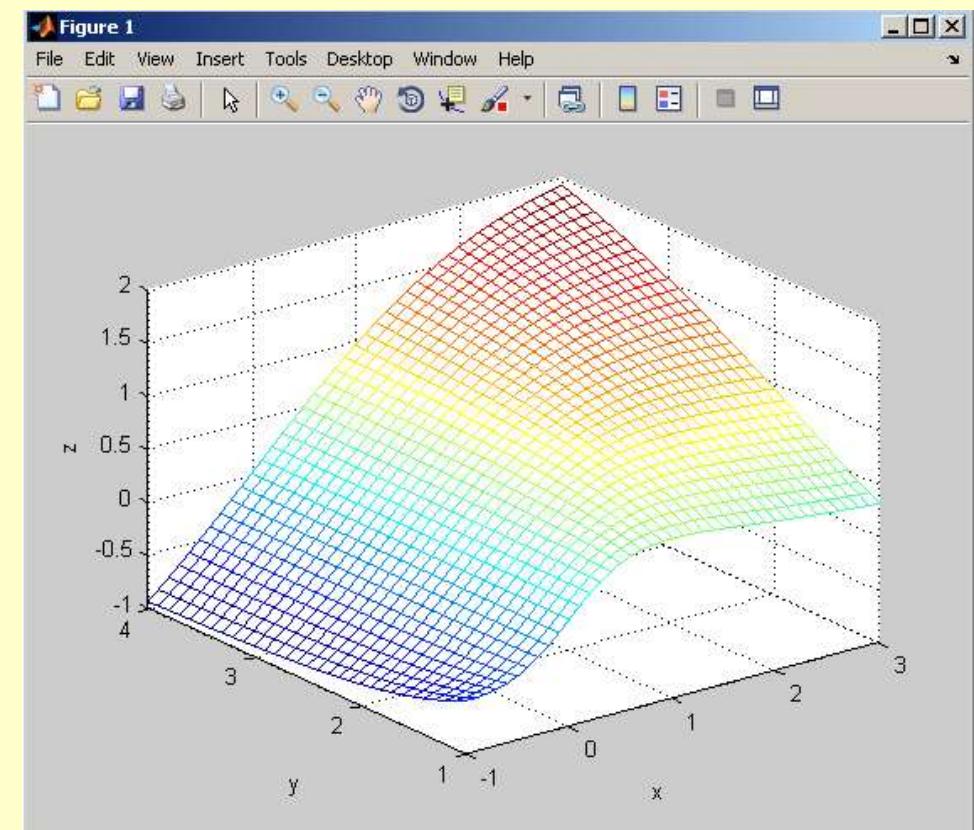
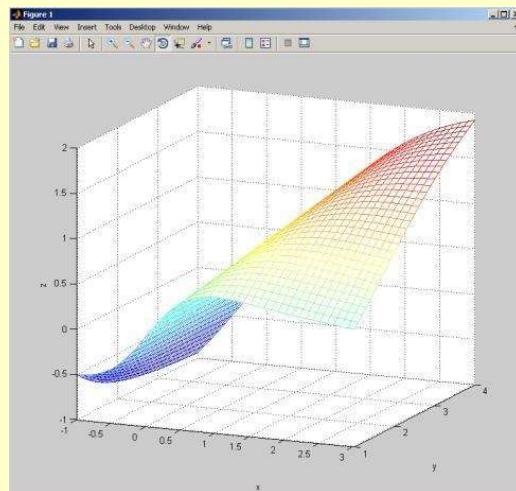


Lệnh mesh(X,Y,Z) và surf(X,Y,Z)

□ Ví dụ vẽ

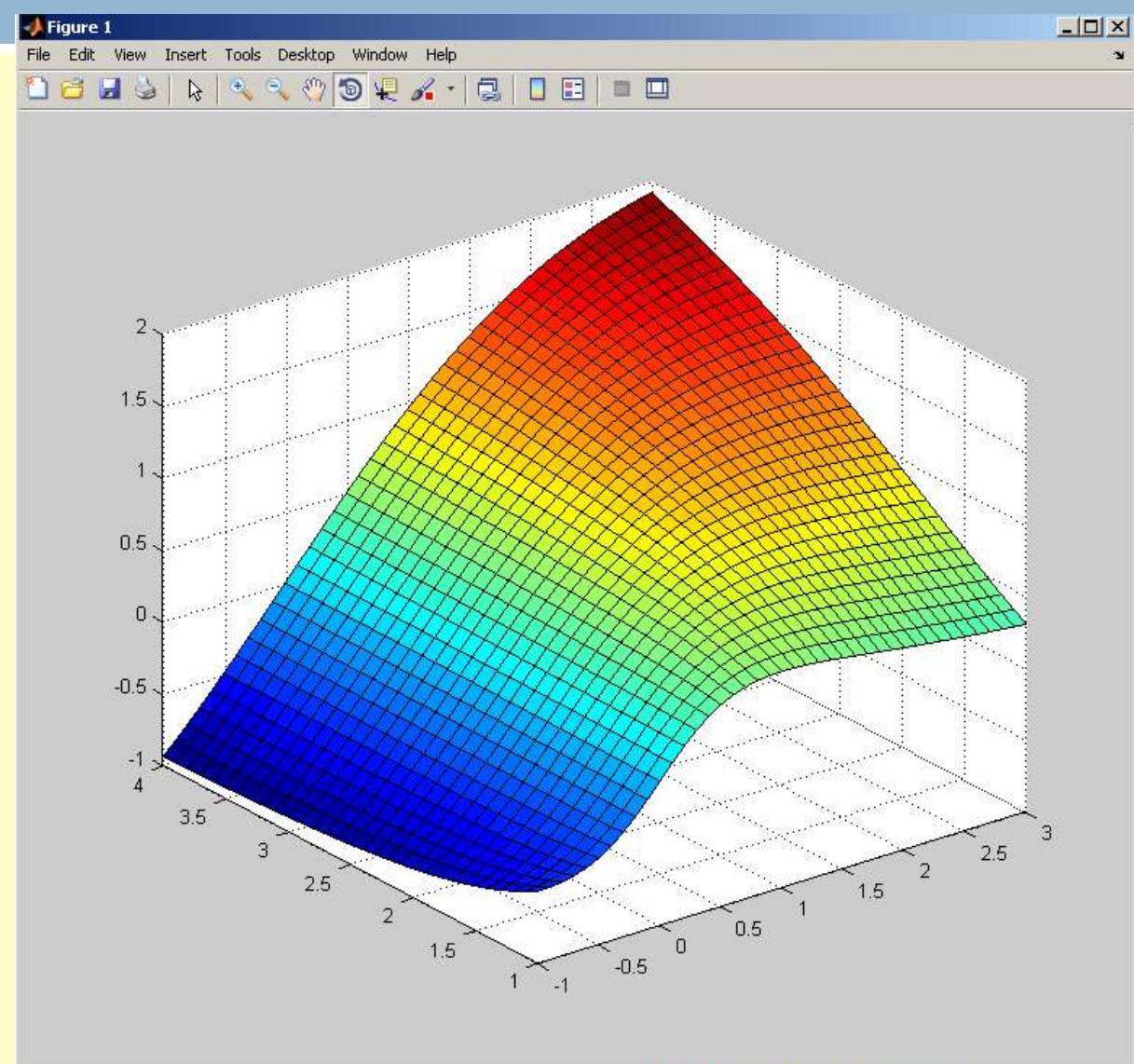
$$z = \frac{xy^2}{x^2 + y^2} \text{ over the domain } -1 \leq x \leq 3 \text{ and } 1 \leq y \leq 4$$

```
x=-1:0.1:3;
y=1:0.1:4;
[X,Y]=meshgrid(x,y);
Z=X.*Y.^2./(X.^2+Y.^2);
mesh(X,Y,Z)
xlabel('x'); ylabel('y'); zlabel('z')
```



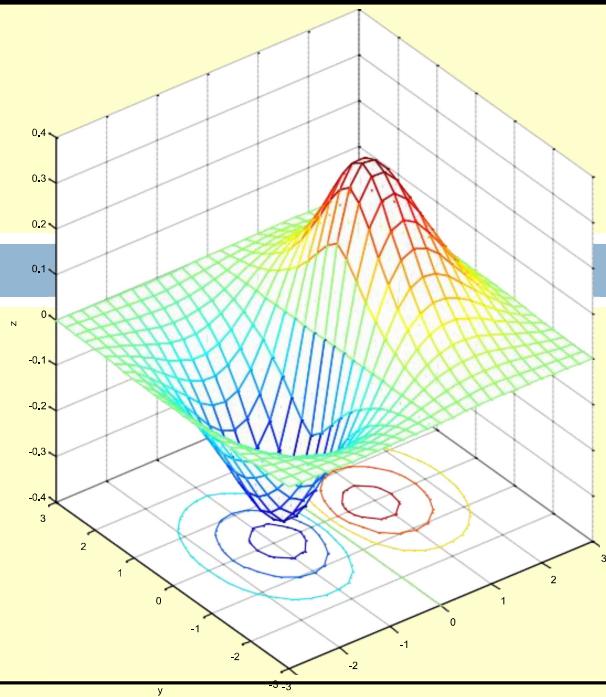
Lệnh mesh(X,Y,Z) và surf(X,Y,Z)

- $\text{surf}(X, Y, Z)$

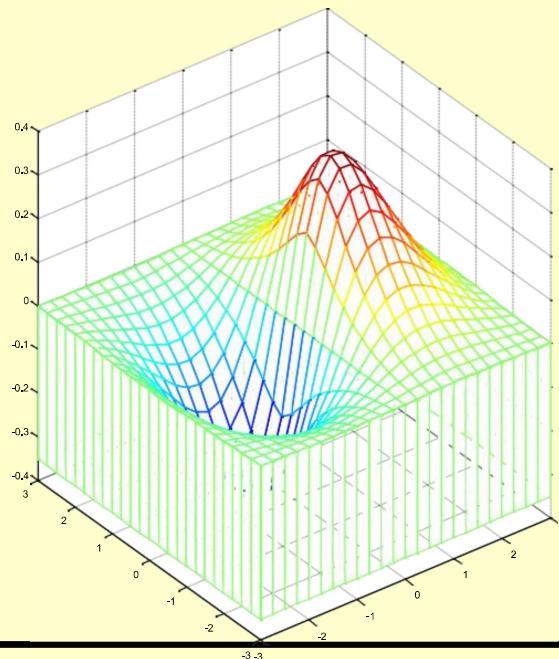


meshc(X,Y,Z)

Vẽ lưới và đường viền

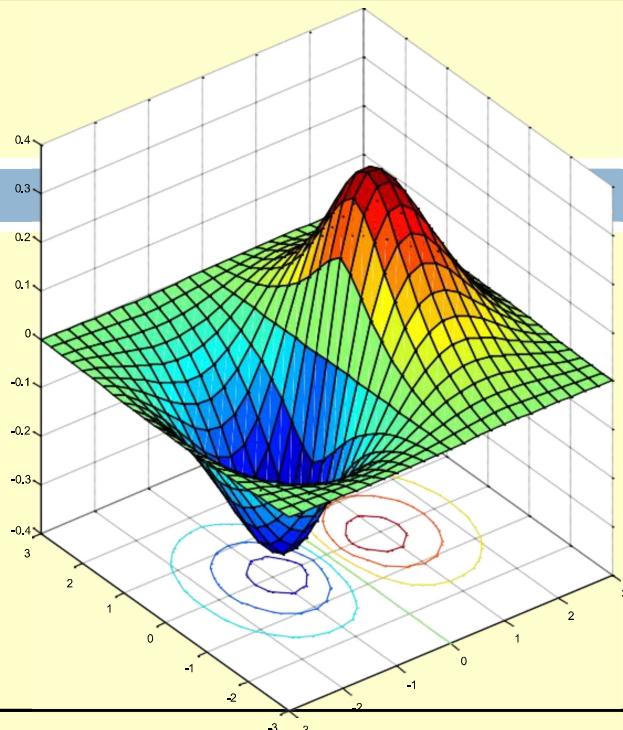
`x=-3:0.25:3;``y=-3:0.25:3;``[X,Y]=meshgrid(x,y);``Z=1.8.^(-1.5*sqrt(X.^2+Y.^2)).*cos(0.5*Y).*sin(X);`**meshc(X,Y,Z)**`xlabel('x');` `ylabel('y')``zlabel('z')`**meshz(X,Y,Z)**

Vẽ lưới và đường màn

`x=-3:0.25:3;``y=-3:0.25:3;``[X,Y]=meshgrid(x,y);``Z=1.8.^(-1.5*sqrt(X.^2+Y.^2)).*cos(0.5*Y).*sin(X);`**meshz(X,Y,Z)**`xlabel('x');` `ylabel('y')``zlabel('z')`

surf(X,Y,Z)

Vẽ mặt và đường viền



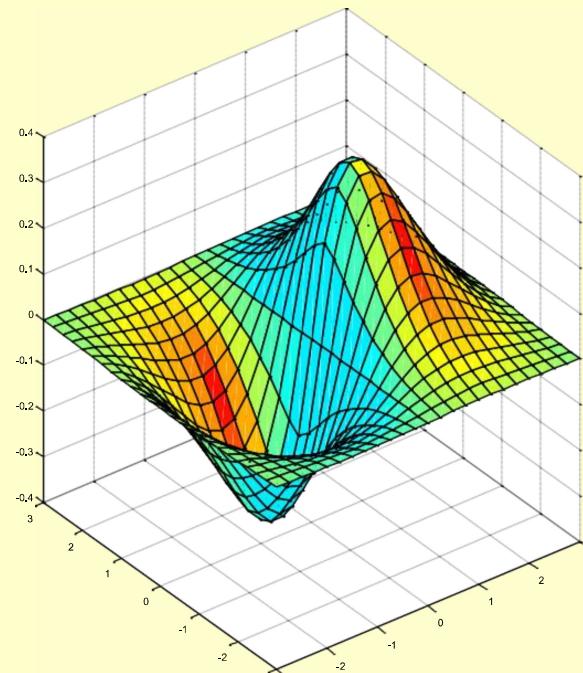
```
x=-3:0.25:3;
y=-3:0.25:3;
[X,Y]=meshgrid(x,y);
Z=1.8.^(-1.5*sqrt(X.^2+Y.^2)).*cos(0.5*Y).*sin(X);
```

surf(X,Y,Z)

```
xlabel('x'); ylabel('y')
zlabel('z')
```

surfl(X,Y,Z)

Surface Plot with
Lighting



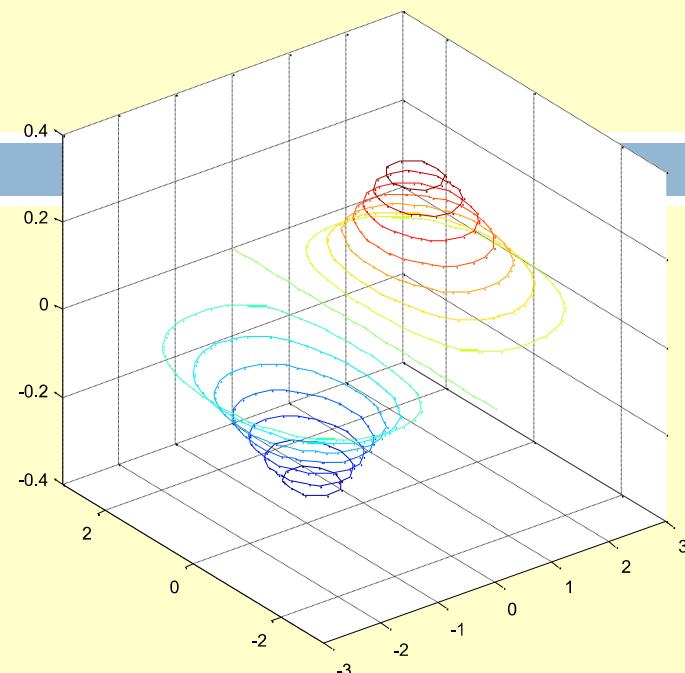
```
x=-3:0.25:3;
y=-3:0.25:3;
[X,Y]=meshgrid(x,y);
Z=1.8.^(-1.5*sqrt(X.^2+Y.^2)).*cos(0.5*Y).*sin(X);
```

surfl(X,Y,Z)

```
xlabel('x'); ylabel('y')
zlabel('z')
```

contour3(X, Y, Z, n)

n là số mức đường viền



```
x=-3:0.25:3;
```

```
y=-3:0.25:3;
```

```
[X, Y]=meshgrid(x, y);
```

```
Z=1.8.^(-1.5*sqrt(X.^2+Y.^2)).*cos(0.5*Y).*sin(X);
```

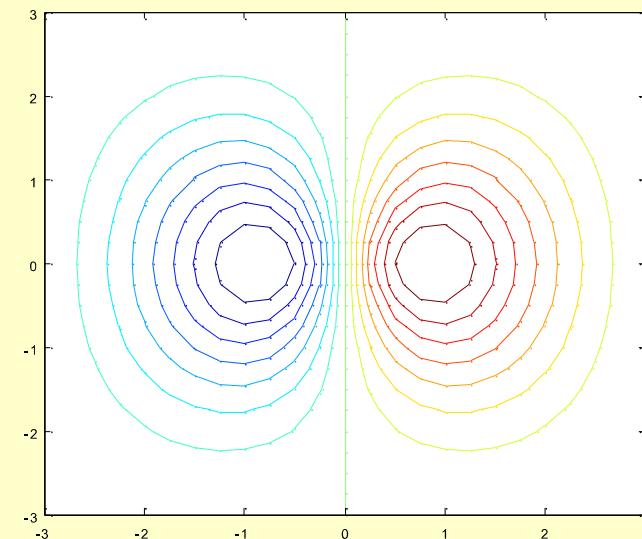
contour3 (X, Y, Z, 15)

```
xlabel('x'); ylabel('y')
```

```
zlabel('z')
```

contour(X, Y, Z, n)

n là số mức đường
viền



```
x=-3:0.25:3;
```

```
y=-3:0.25:3;
```

```
[X, Y]=meshgrid(x, y);
```

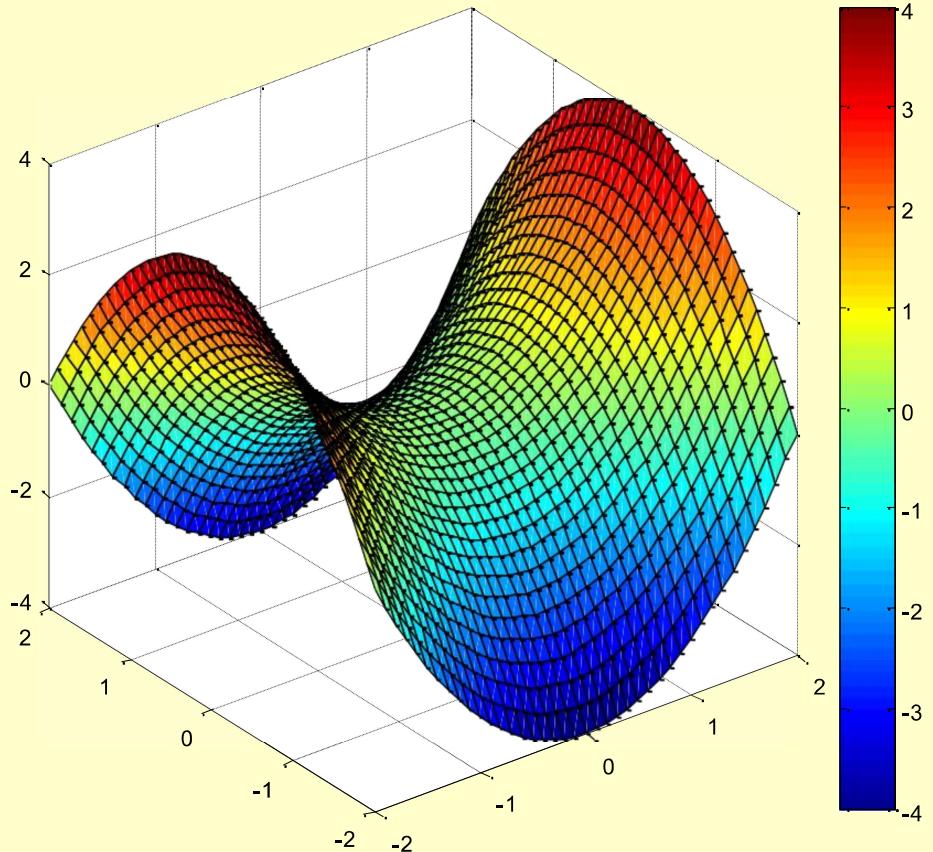
```
Z=1.8.^(-1.5*sqrt(X.^2+Y.^2)).*cos(0.5*Y).*sin(X);
```

contour (X, Y, Z, 15)

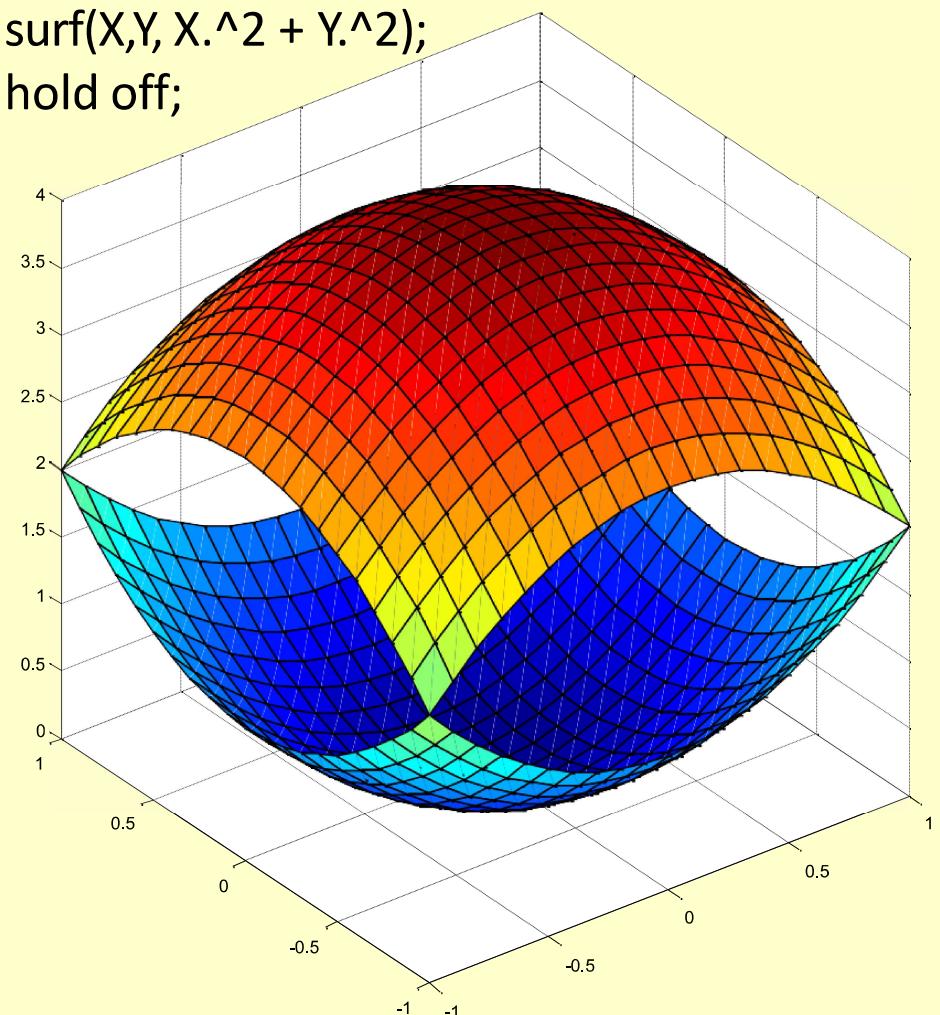
```
xlabel('x'); ylabel('y')
```

```
zlabel('z')
```

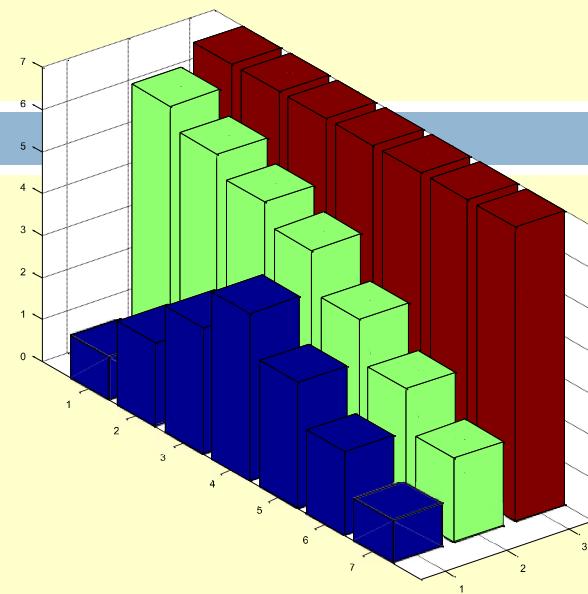
```
>> [X,Y] = meshgrid(-2:0.1:2);
>> surf(X, Y, X.^2 - Y.^2)
>> colorbar
```



```
>> [X,Y] = meshgrid(-1:0.1:1);
>> surf(X, Y, 4 - X.^2 - Y.^2);
>> hold on;
>> surf(X, Y, X.^2 + Y.^2);
>> hold off;
```



Bar3(Y)

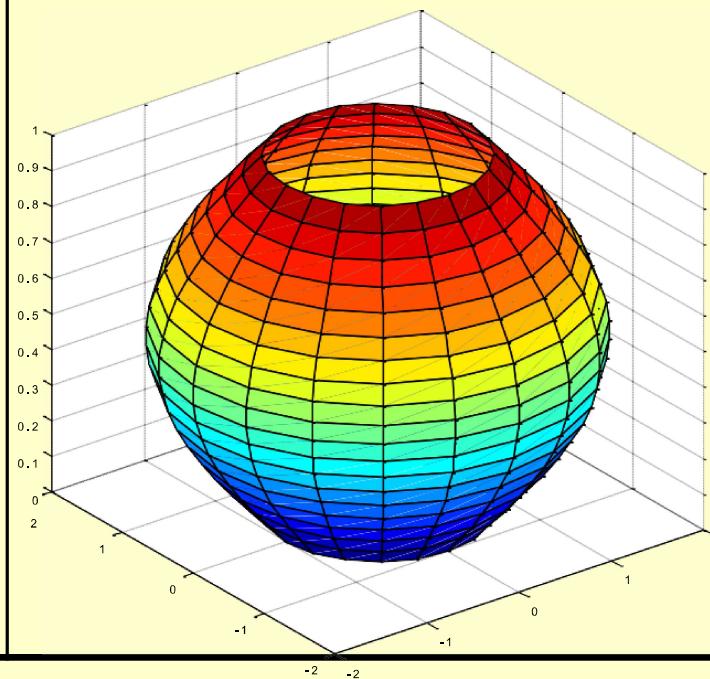


```
>> Y=[1 6.5 7; 2 6 7; 3 5.5 7; 4 5 7; 3  
4 7; 2 3 7; 1 2 7]
```

Y =

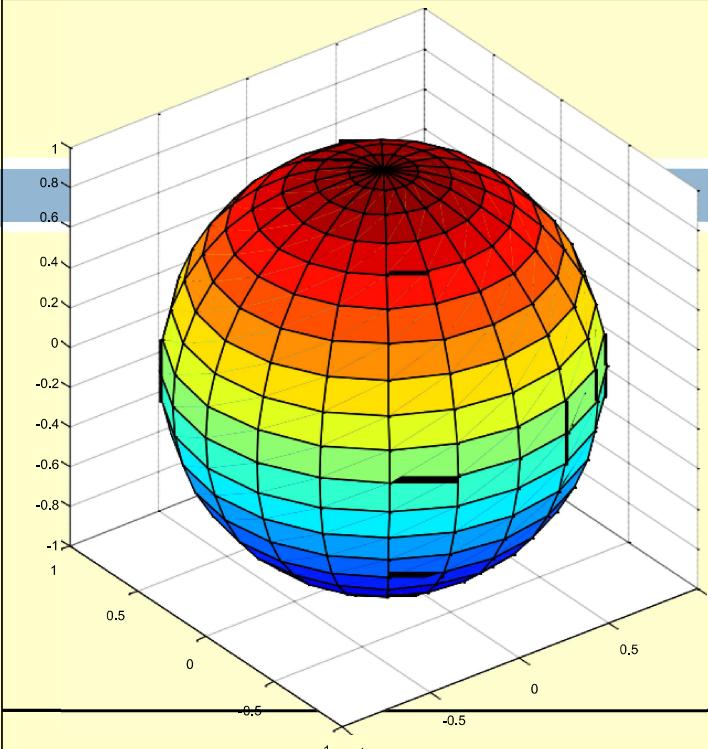
1.0000	6.5000	7.0000
2.0000	6.0000	7.0000
3.0000	5.5000	7.0000
4.0000	5.0000	7.0000
3.0000	4.0000	7.0000
2.0000	3.0000	7.0000
1.0000	2.0000	7.0000

Bài tập



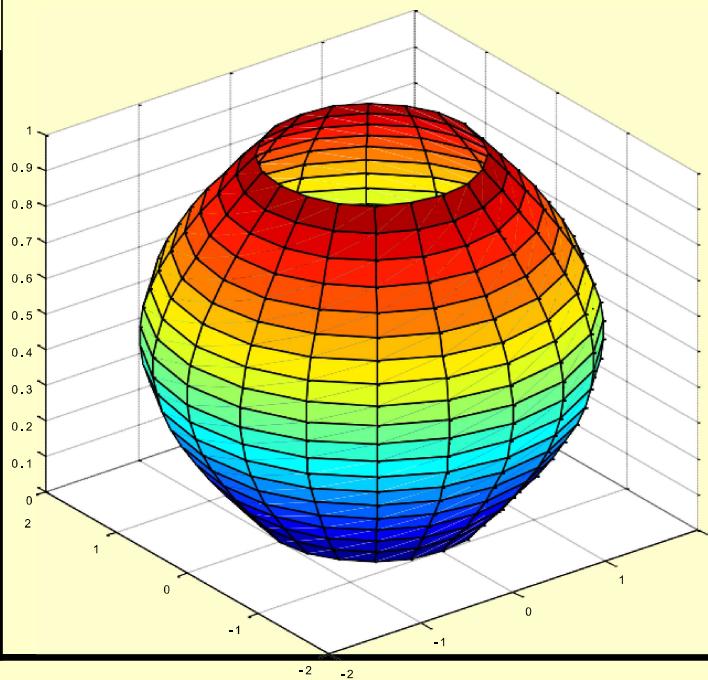
```
t=linspace(0,pi,20);  
r=1+sin(t);  
[x, y, z]=cylinder(r);  
surf(x,y,z)  
axis square
```

Bài tập



```
[x, y, z]=sphere(30);  
surf(x,y,z)  
axis square
```

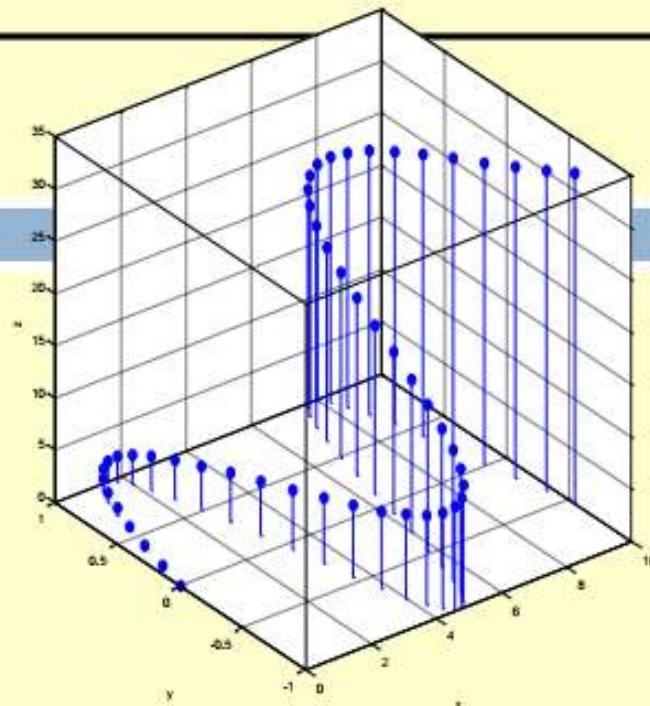
Bài tập



```
t=linspace(0,pi,20);  
r=1+sin(t);  
[x, y, z]=cylinder(r);  
surf(x,y,z)  
axis square
```

Bài tập

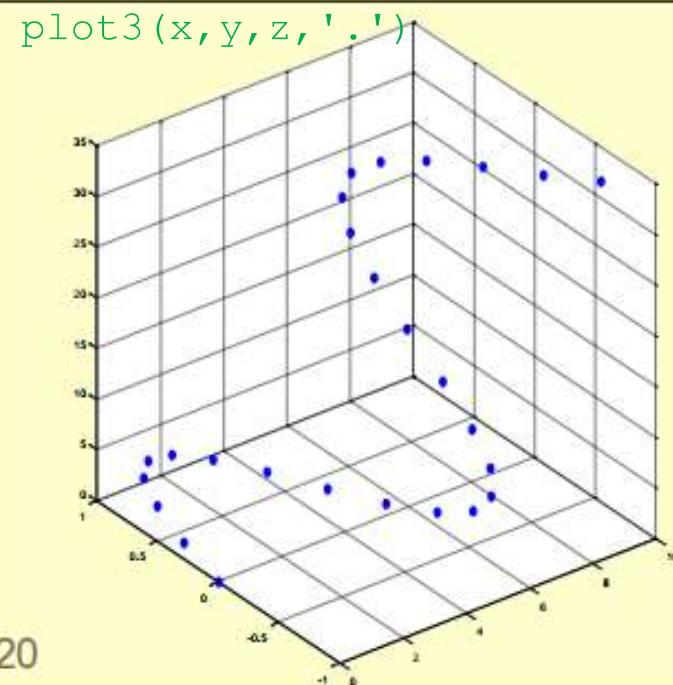
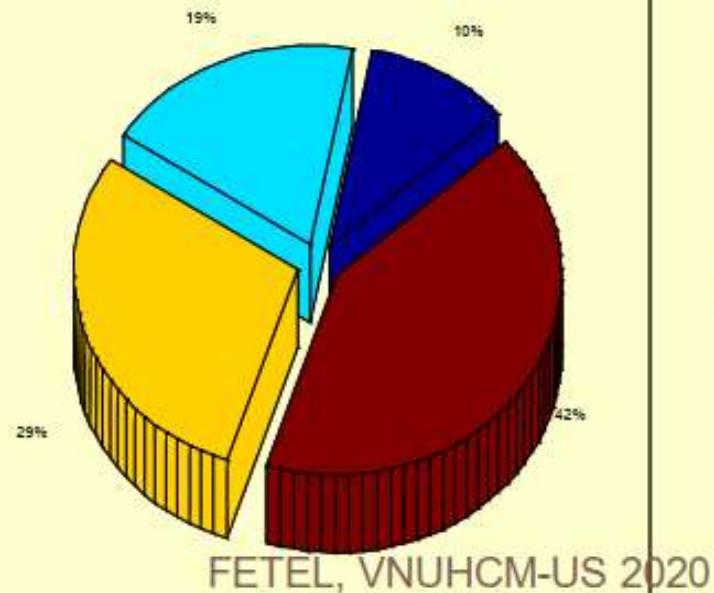
209



```
t=0:0.2:10;
x=t;
y=sin(t);
z=t.^1.5;
???
stem3(x,y,z)
```

Bài tập

```
p = [3.3 2.6 .69 .4 .3];
pie3(p)
title('Interesting Chart')
```



Chương 5

File Input - Output

Save

□ `save file_name` hay `save('file_name')`

→ Lưu tất cả các biến có trong workspace vào file, file mặc định có phần mở rộng .mat. Đây là file mã hóa.

```
>> x=1:10;
>> y=linspace(4,50,10);
>> z=rand(1,10)*10;
>> data=[x', y', z']

data =

```

1.0000	4.0000	7.0605
2.0000	9.1111	0.3183
3.0000	14.2222	2.7692
4.0000	19.3333	0.4617
5.0000	24.4444	0.9713
6.0000	29.5556	8.2346
7.0000	34.6667	6.9483
8.0000	39.7778	3.1710
9.0000	44.8889	9.5022
10.0000	50.0000	0.3445

	Name	Value
x	[1,2,3,4,5,6,7,8,9,10]	
y	[4,9,1111,14.2222,19.3333,24.4...	
z	[7.0605,0.3183,2.7692,0.4617,0...	
data	<10x3 double>	

Save

- `save file_name var1 var2`
- Hay `save('file_name','var1','var2')`

⑦ Sẽ lưu biến var1 và var2

```
>> save('datafile1','x', 'y')
```

Name	Value
x	[1,2,3,4,5,6,7,8,9,10]
y	[4,9.1111,14.2222,19...]

- Nếu muốn lưu file dưới dạng ascii dùng từ khóa -ascii

```
>> save('dataascii.dat','data','-ascii')
```

Import Wizard																																															
Select Column Separator(s)																																															
<input type="radio"/> Comma	<input checked="" type="radio"/> Space	<input type="radio"/> Semicolon	<input type="radio"/> Tab																																												
<input type="radio"/> Other	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																												
Number of text header lines: 0																																															
Preview of C:\Documents and Settings\Administrator\My Documents\MATLAB\dataascii.dat																																															
<table border="1"> <thead> <tr> <th>dataascii</th> <th>1</th> <th>2</th> <th>3</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>4</td><td>7.0605</td></tr> <tr><td>2</td><td>2</td><td>9.1111</td><td>0.3183</td></tr> <tr><td>3</td><td>3</td><td>14.2222</td><td>2.7692</td></tr> <tr><td>4</td><td>4</td><td>19.3333</td><td>0.4617</td></tr> <tr><td>5</td><td>5</td><td>24.4444</td><td>0.9713</td></tr> <tr><td>6</td><td>6</td><td>29.5556</td><td>8.2346</td></tr> <tr><td>7</td><td>7</td><td>34.6667</td><td>6.9483</td></tr> <tr><td>8</td><td>8</td><td>39.7778</td><td>3.1710</td></tr> <tr><td>9</td><td>9</td><td>44.8889</td><td>9.5022</td></tr> <tr><td>10</td><td>10</td><td>50</td><td>0.3445</td></tr> </tbody> </table>				dataascii	1	2	3	1	1	4	7.0605	2	2	9.1111	0.3183	3	3	14.2222	2.7692	4	4	19.3333	0.4617	5	5	24.4444	0.9713	6	6	29.5556	8.2346	7	7	34.6667	6.9483	8	8	39.7778	3.1710	9	9	44.8889	9.5022	10	10	50	0.3445
dataascii	1	2	3																																												
1	1	4	7.0605																																												
2	2	9.1111	0.3183																																												
3	3	14.2222	2.7692																																												
4	4	19.3333	0.4617																																												
5	5	24.4444	0.9713																																												
6	6	29.5556	8.2346																																												
7	7	34.6667	6.9483																																												
8	8	39.7778	3.1710																																												
9	9	44.8889	9.5022																																												
10	10	50	0.3445																																												
<input type="button" value="Help"/> <input type="button" value="Next >"/> <input type="button" value="Finish"/> <input type="checkbox"/> Generate MATLAB code <input type="button" value="Cancel"/>																																															

load

- Để tải lại các biến đã lưu trên file lên workspace, dùng lệnh load

load file_name

load('file_name')

load file_name var1 var2

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

- Khi tải lại các biến được lưu dưới dạng file ascii hay text, có thể dùng lệnh sau

load file_name

VarName=load('file_name')

load file_name.txt

VarName=load('file_name.txt')

Xuất và Tải dữ liệu từ file excel

```
>> x=1:10;
>> y=linspace(4,50,10);
>> z=rand(1,10)*10;
>> data=[x', y', z']
```

data =

1.0000	4.0000	4.3874
2.0000	9.1111	3.8156
3.0000	14.2222	7.6552
4.0000	19.3333	7.9520
5.0000	24.4444	1.8687
6.0000	29.5556	4.8976
7.0000	34.6667	4.4559
8.0000	39.7778	6.4631
9.0000	44.8889	7.0936
10.0000	50.0000	7.5469

```
>> xlswrite('Excellfile',data)
```

```
xlswrite('filename',variable_name)
variable_name = xlsread('filename')
```

```
variable_name = xlsread('filename','sheet_name')
variable_name = xlsread('filename','sheet_name','range')
```

Microsoft Excel - Excellfile

E12 fx

	A	B	C	D	E
1		1	4.387444		
2		2	9.111111	3.815585	
3		3	14.222222	7.655168	
4		4	19.333333	7.951999	
5		5	24.444444	1.868726	
6		6	29.555556	4.897644	
7		7	34.666667	4.455862	
8		8	39.777778	6.46313	
9		9	44.888889	7.093648	
10		10	50	7.546867	
11					
12					
13					
14					

>> data1=xlsread('Excellfile')

data1 =

1.0000	4.0000	4.3874
2.0000	9.1111	3.8156
3.0000	14.2222	7.6552
4.0000	19.3333	7.9520
5.0000	24.4444	1.8687
6.0000	29.5556	4.8976
7.0000	34.6667	4.4559
8.0000	39.7778	6.4631
9.0000	44.8889	7.0936
10.0000	50.0000	7.5469

Xuất và Tải dữ liệu từ file excel

```
>> data2=xlsread('Excelfile','A2:C8')

data2 =
```

2.0000	9.1111	3.8156
3.0000	14.2222	7.6552
4.0000	19.3333	7.9520
5.0000	24.4444	1.8687
6.0000	29.5556	4.8976
7.0000	34.6667	4.4559
8.0000	39.7778	6.4631

```
>> data2=xlsread('Excelfile','sheet2','A5:F8')

data2 =
```

5.0000	24.4444	1.8687	5.0000	24.4444	1.8687
6.0000	29.5556	4.8976	6.0000	29.5556	4.8976
7.0000	34.6667	4.4559	7.0000	34.6667	4.4559
8.0000	39.7778	6.4631	8.0000	39.7778	6.4631

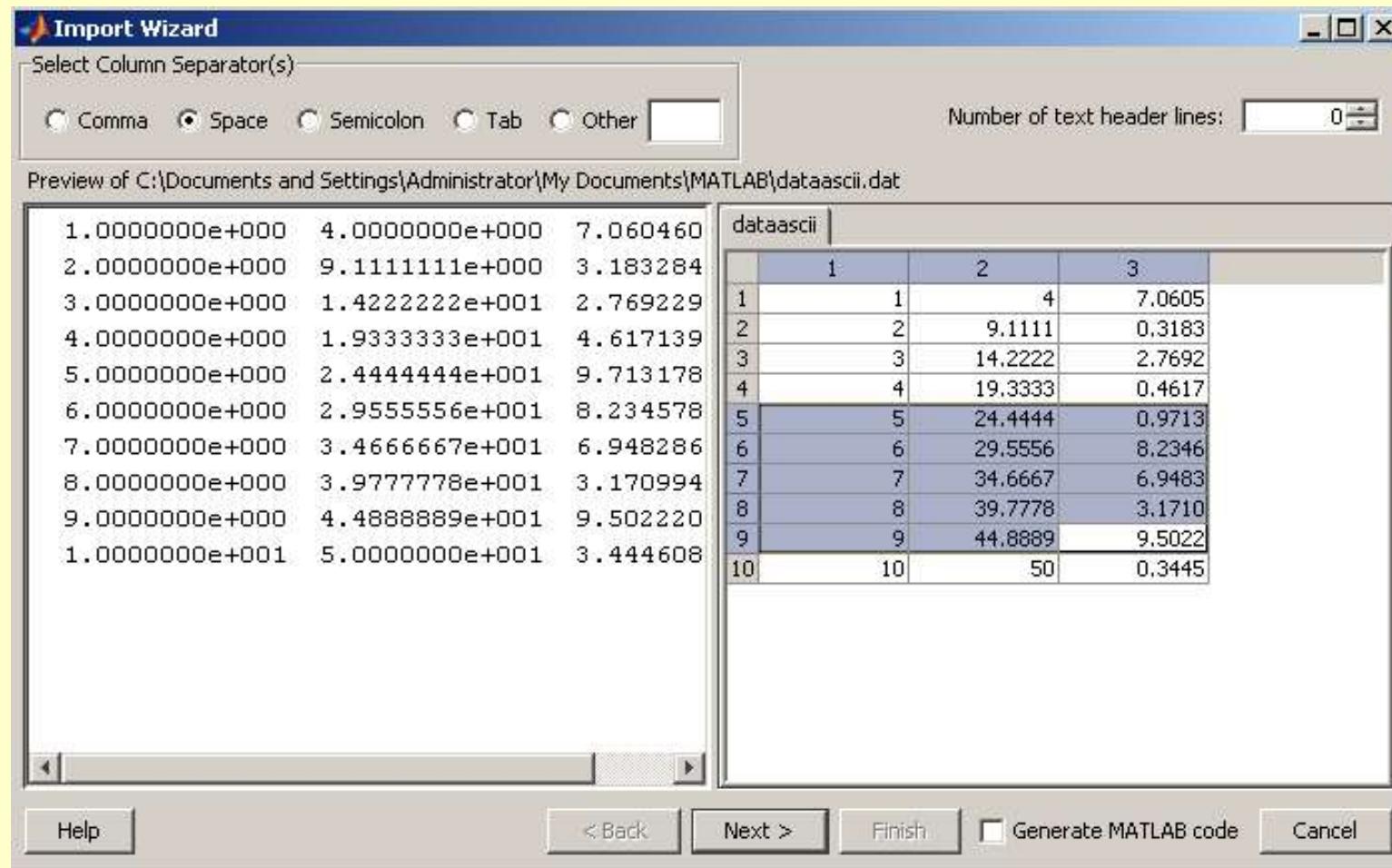
Microsoft Excel - Excelfile

	A	B	C	D	E	F
1	1	4	4.387444	1	4	4.387444
2	2	9.111111	3.815585	2	9.111111	3.815585
3	3	14.222222	7.655168	3	14.222222	7.655168
4	4	19.333333	7.951999	4	19.333333	7.951999
5	5	24.444444	1.868726	5	24.444444	1.868726
6	6	29.555556	4.897644	6	29.555556	4.897644
7	7	34.666667	4.455862	7	34.666667	4.455862
8	8	39.777778	6.46313	8	39.777778	6.46313
9	9	44.888889	7.093648	9	44.888889	7.093648
10	10	50	7.546867	10	50	7.546867
11	1	4	4.387444	1	4	4.387444
12	2	9.111111	3.815585	2	9.111111	3.815585
13	3	14.222222	7.655168	3	14.222222	7.655168
14	4	19.333333	7.951999	4	19.333333	7.951999
15	5	24.444444	1.868726	5	24.444444	1.868726
16	6	29.555556	4.897644	6	29.555556	4.897644
17	7	34.666667	4.455862	7	34.666667	4.455862
18	8	39.777778	6.46313	8	39.777778	6.46313
19	9	44.888889	7.093648	9	44.888889	7.093648
20	10	50	7.546867	10	50	7.546867
21						

Sheet2 / Sheet1 / Sheet3 /

Ready

Dùng Import Wizard để tải data



Dùng file handle

- Mở file: fid=fopen(filename,mode)

Mode	Meaning	Explanation	Already Exists	Does Not Exist
"r"	read	Open a file for reading	read from start	error
"w"	write	Create a file for writing	destroy contents	create new
"a"	append	Append to a file	write to end	create new
"r+"	read extended	Open a file for read/write	read from start	error
"w+"	write extended	Create a file for read/write	destroy contents	create new
"a+"	append extended	Open a file for read/write	write to end	create new

```
fp = fopen('file_name.txt', 'r');
```

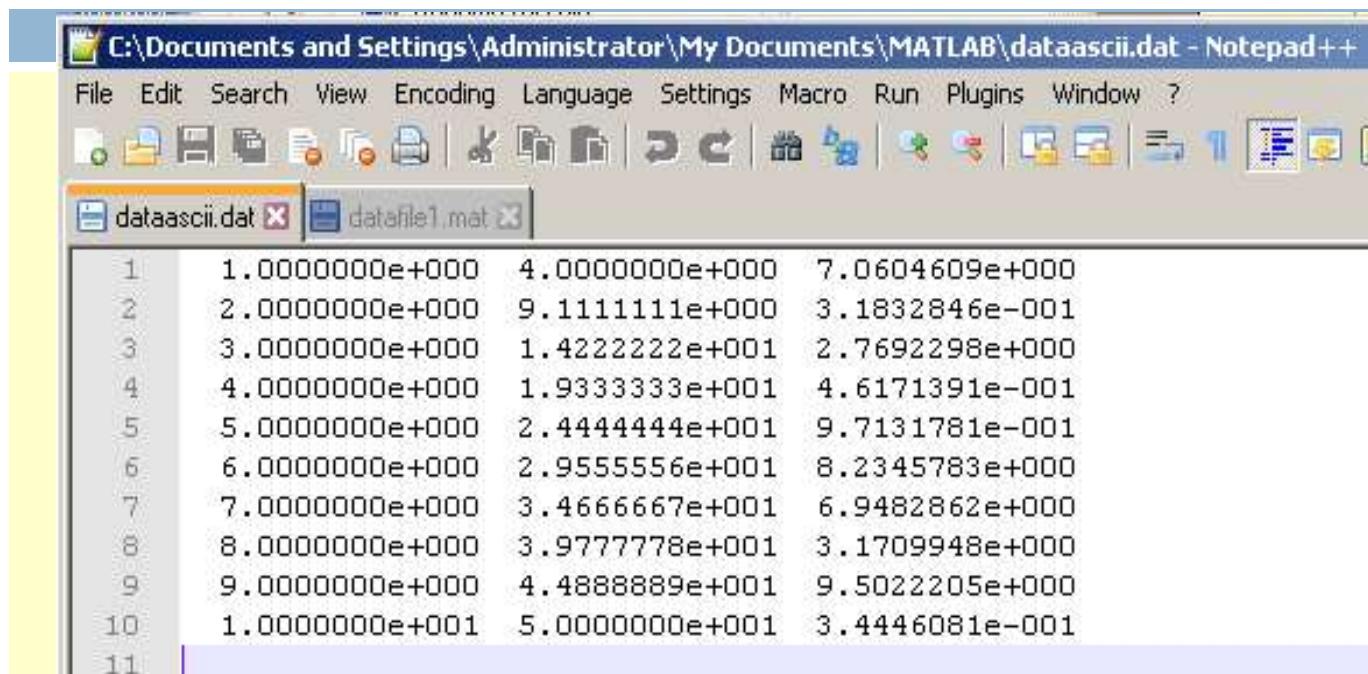
- Đóng file: fclose(fid)

fscanf

```
>> fp = fopen('dataascii.dat', 'r');
>> data=fscanf(fp, '%f')

data =
```

1.0000
4.0000
7.0605
2.0000
9.1111
0.3183
3.0000
14.2222
2.7692
4.0000
19.3333
0.4617
5.0000
24.4444
0.9713
6.0000
29.5556
8.2346
7.0000
34.6667
6.9483
8.0000
39.7778
3.1710
9.0000
44.8889
9.5022
10.0000
50.0000
0.3445



The screenshot shows a Notepad++ window titled "C:\Documents and Settings\Administrator\My Documents\MATLAB\dataascii.dat - Notepad++". The file contains 11 lines of data, each consisting of four floating-point numbers separated by spaces. The data corresponds to the values printed by the MATLAB code above.

1	1.0000000e+000	4.0000000e+000	7.0604609e+000
2	2.0000000e+000	9.1111111e+000	3.1832846e-001
3	3.0000000e+000	1.4222222e+001	2.7692298e+000
4	4.0000000e+000	1.9333333e+001	4.6171391e-001
5	5.0000000e+000	2.4444444e+001	9.7131781e-001
6	6.0000000e+000	2.9555556e+001	8.2345783e+000
7	7.0000000e+000	3.4666667e+001	6.9482862e+000
8	8.0000000e+000	3.9777778e+001	3.1709948e+000
9	9.0000000e+000	4.4888889e+001	9.5022205e+000
10	1.0000000e+001	5.0000000e+001	3.4446081e-001
11			

```
fprintf('text as string %-.5.2f additional text',
        variable_name)
```

The % sign marks the spot where the number is inserted within the text.

Formatting elements (define the format of the number).

The name of the variable whose value is displayed.

Flag (optional).
Field width and precision (optional).
Conversion character (required).

Character used for flag

- (minus sign)
- + (plus sign)
- 0 (zero)

Description

- | |
|--|
| Left-justifies the number within the field. |
| Prints a sign character (+ or -) in front of the number. |
| Adds zeros if the number is shorter than the field. |

- e Exponential notation using lower-case e (e.g., 1.709098e+001).
- E Exponential notation using upper-case E (e.g., 1.709098E+001).
- f Fixed-point notation (e.g., 17.090980).
- g The shorter of e or f notations.
- G The shorter of E or f notations.
- i Integer.

```
fprintf(fid, 'text %-5.2f additional text', variable_name)
```

fid is added to the fprintf command.

fprintf

```
>> data

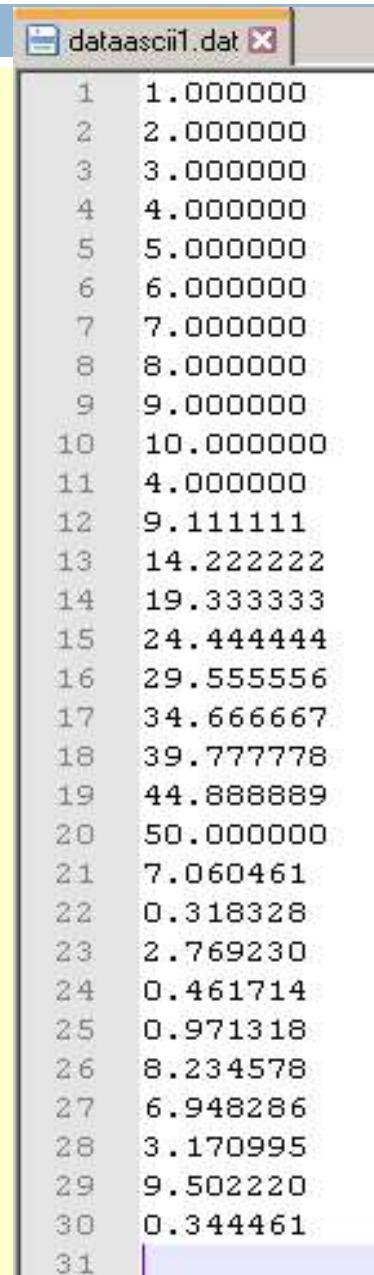
data =

1.0000    4.0000    7.0605
2.0000    9.1111    0.3183
3.0000   14.2222    2.7692
4.0000   19.3333    0.4617
5.0000   24.4444    0.9713
6.0000   29.5556    8.2346
7.0000   34.6667    6.9483
8.0000   39.7778    3.1710
9.0000   44.8889    9.5022
10.0000  50.0000    0.3445

>> fp = fopen('dataascii1.dat', 'w');
>> fprintf(fp, '%f \n', data)

ans =

309
```

A screenshot of a Windows File Explorer window. The window title is "dataascii1.dat". The file size is listed as 309. The file was modified on 11/11/2023 at 10:46 AM.

	Value
1	1.000000
2	2.000000
3	3.000000
4	4.000000
5	5.000000
6	6.000000
7	7.000000
8	8.000000
9	9.000000
10	10.000000
11	4.000000
12	9.111111
13	14.222222
14	19.333333
15	24.444444
16	29.555556
17	34.666667
18	39.777778
19	44.888889
20	50.000000
21	7.060461
22	0.318328
23	2.769230
24	0.461714
25	0.971318
26	8.234578
27	6.948286
28	3.170995
29	9.502220
30	0.344461
31	

```
x = 0:0.1:1;
y = [x; exp(x)];
fileID = fopen('exptable.txt','w');
fprintf(fileID, 'Exponential Function\n\n');
fprintf(fileID,'%f %f\n',y);
fclose(fileID);
type exptable.txt
```

https://uk.mathworks.com/help/matlab/import_export/writing-to-text-data-files-with-low-level-io.html

Kiểm tra lỗi mở/đóng file

□ Kiểm tra mở file (not use)

- `fopen()` will return NULL if there is an error opening the file

```
fp = fopen("file_name.txt", "r");  
if(fp == NULL) {  
    printf("Error: can't open file to read\n");  
    return 1;  
}
```

□ Kiểm tra đóng file (not use)

- `int fclose(FILE *fp);`

✓ `fclose()` will return EOF if there is an error closing the file

```
if(fclose(fp) == EOF) {  
    printf("Error closing file\n");  
    return 1;  
}
```

Closing a File

222

□ Function header: (not use)

- `int fclose(FILE *fp) ;`

- Good idea to test the file was closed without error

- ✓ Test the return value of `fclose`

- ✓ `fclose()` will return EOF if there is an error closing the file

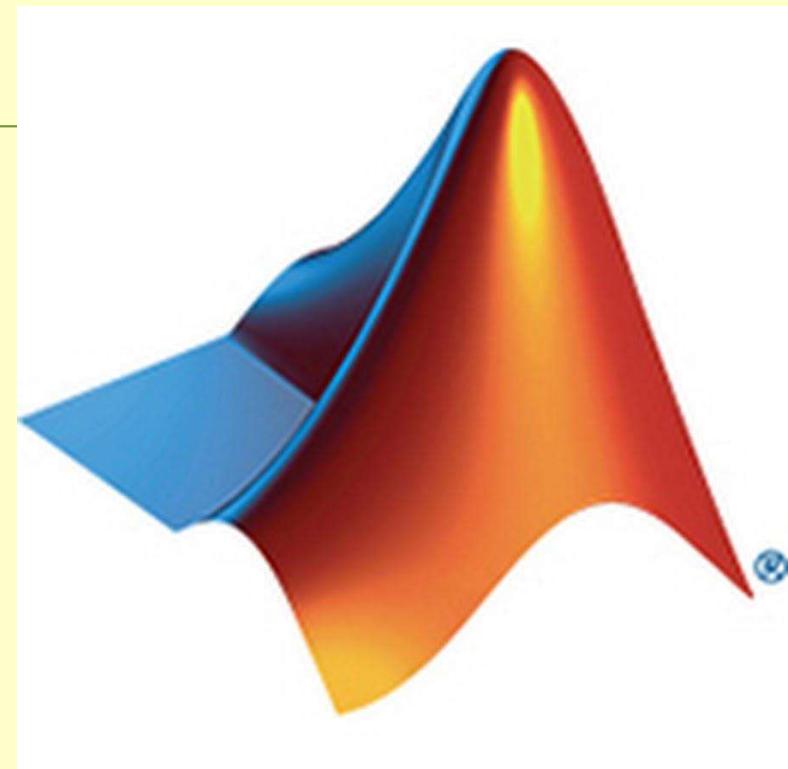
```
if(fclose(fp) == EOF) {  
    printf("Error closing file\n");  
    return 1;  
}
```

- It is best practice to close all files that you opened, somewhere in your program

Chương 6

Symbolics – GUI – Simulink - Debugging

Symbolic GUI Simulink



Symbolic

```
syms x y
```

Symbolic

Rút gọn biểu thức sau:

$$4\cos^2(x) + \cos(2x) + \sin^2(x)$$

```
>> syms x
>> simplify(4*cos(x)^2+ cos(2*x)+sin(x)^2)
ans =
5 - 5*sin(x)^2
```

Khai báo biến x

Đơn giản hóa biểu thức

Symbolic

Giải phương trình sau với biến x và biến y

$$(x - a)^2 + (y - b)^2 = r^2$$

```
>>syms x y a b r  
>>solve((x-a)^2+(y-b)^2 == r^2,x) %giai da thuc
```

ans =

$$\begin{aligned} &a + (b + r - y)^{(1/2)} * (r - b + y)^{(1/2)} \\ &a - (b + r - y)^{(1/2)} * (r - b + y)^{(1/2)} \end{aligned}$$

```
>> solve((x-a)^2+(y-b)^2 == r^2,y)
```

ans =

$$\begin{aligned} &b + (a + r - x)^{(1/2)} * (r - a + x)^{(1/2)} \\ &b - (a + r - x)^{(1/2)} * (r - a + x)^{(1/2)} \end{aligned}$$

Symbolic

Tính tích phân Q với
cận a=0 và b=2

$$Q = \int_a^b xe^x dx$$

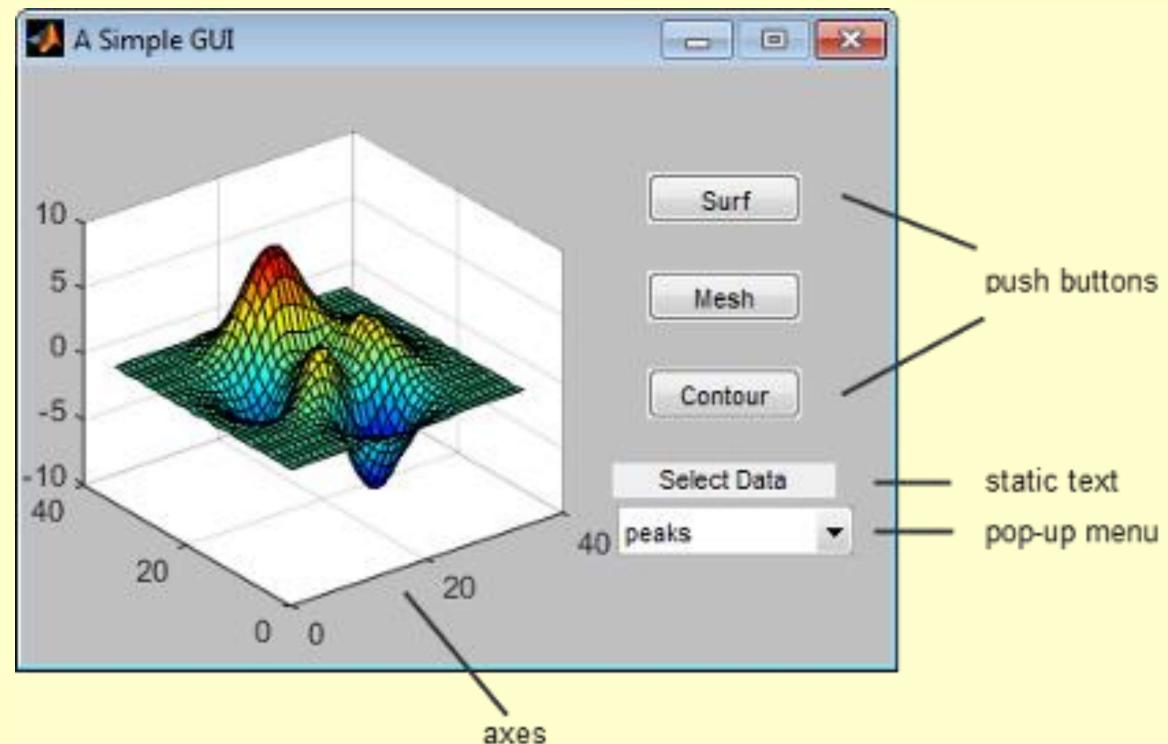
```
>>syms x a b
>>Q=int(x*exp(x), a, b)
Q=
exp(b)*(b-1)-exp(a)*(a-1)
>>subs(Q, [a b], [0 2])
ans =
exp(2) + 1
8.3891
```

Giải toán ký hiệu và toán số

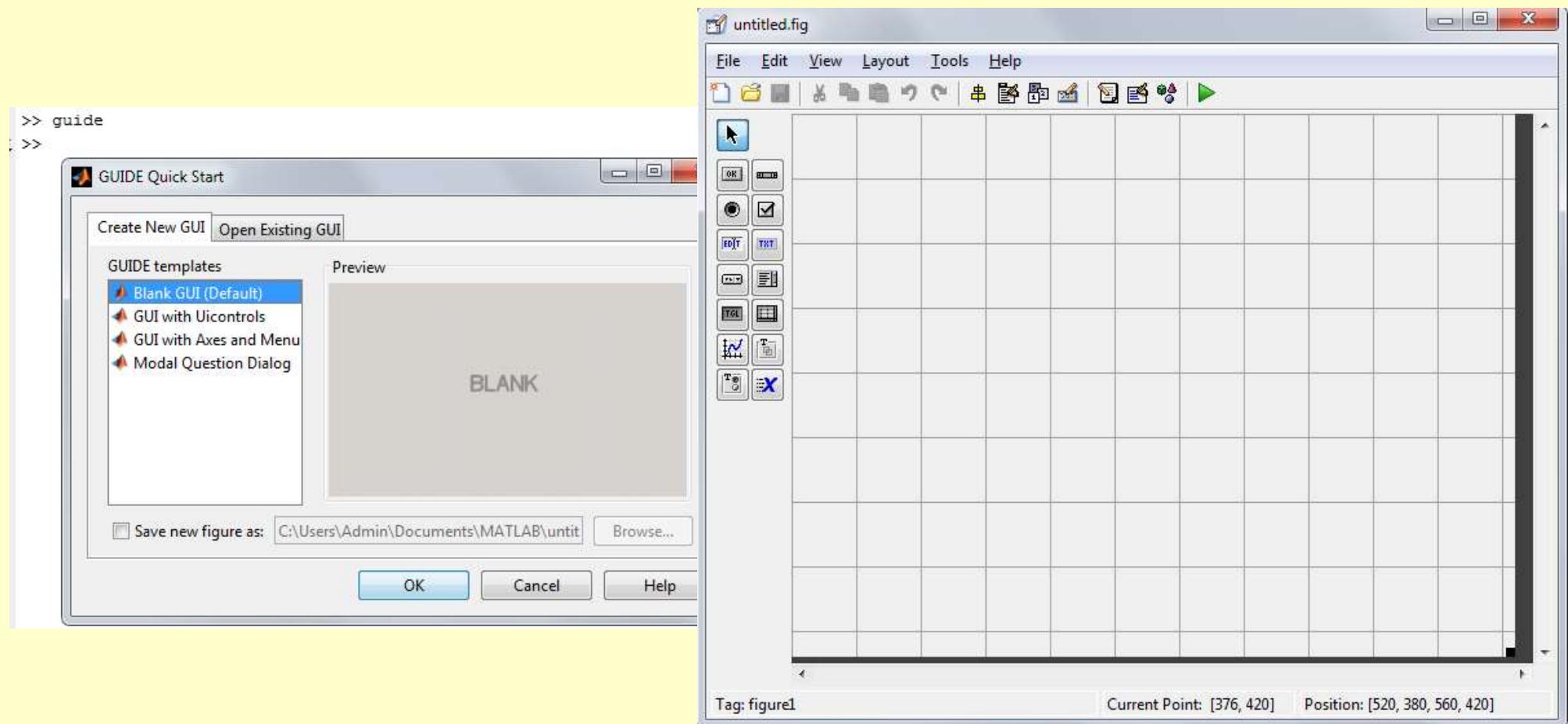
	Ưu điểm	Khuyết điểm
Symbolic	<ul style="list-style-type: none"><input type="checkbox"/> Cho lời giải giải tích<input type="checkbox"/> Cho cái nhìn trực quan về lời giải	<ul style="list-style-type: none"><input type="checkbox"/> Đôi khi không giải được<input type="checkbox"/> Có thể rất phức tạp
Numeric gần đúng	<ul style="list-style-type: none"><input type="checkbox"/> Luôn cho lời giải<input type="checkbox"/> Có thể cho lời giải chính xác<input type="checkbox"/> Dễ hiện thực bằng code	<ul style="list-style-type: none"><input type="checkbox"/> Thường không cho thấy bản chất<input type="checkbox"/> Lời giải số đôi khi sai<input type="checkbox"/> Cần thời gian tính toán

GUIDE

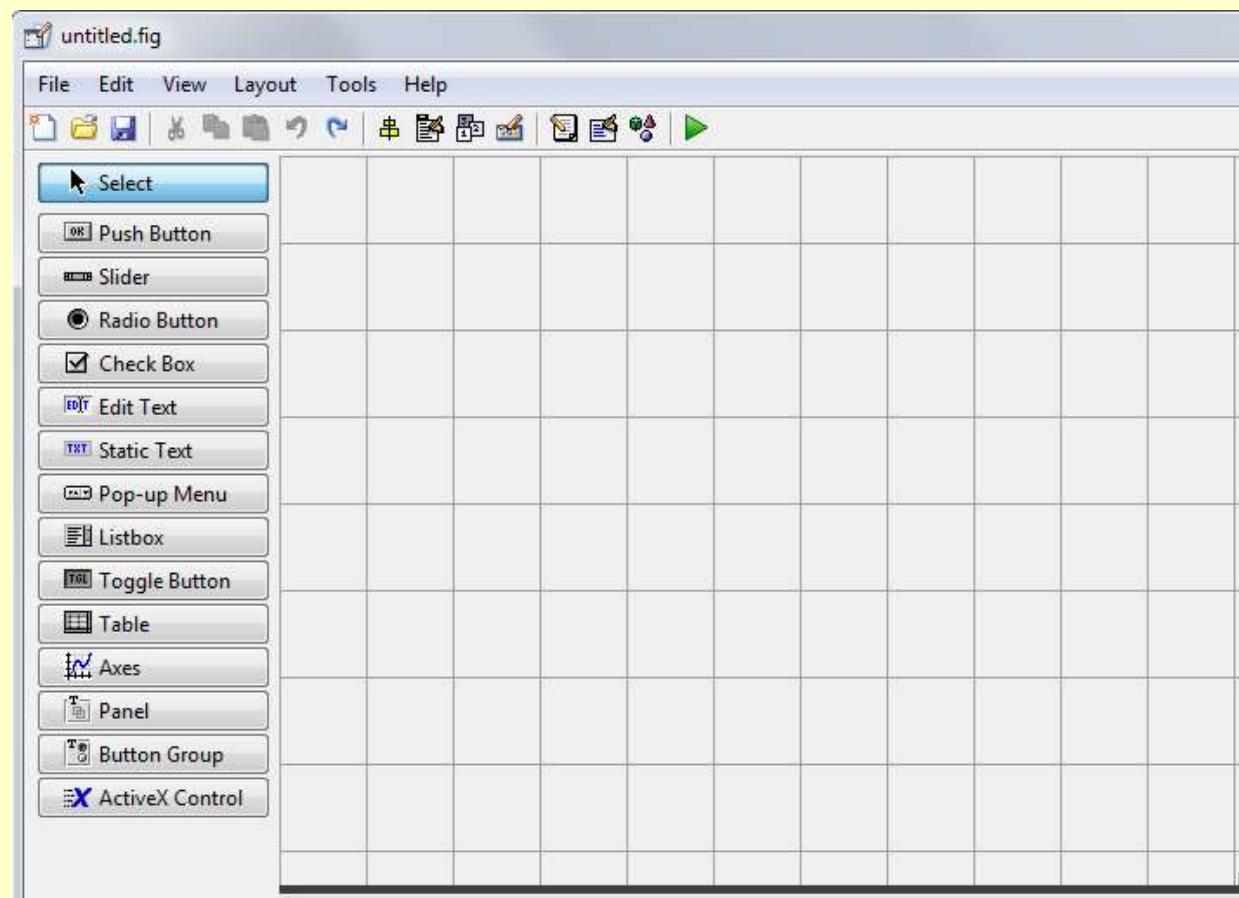
Thiết kế với giao diện



Khởi động GUIDE



Guide Layout



Alignment tool

Toolbar editor

Property Inspector

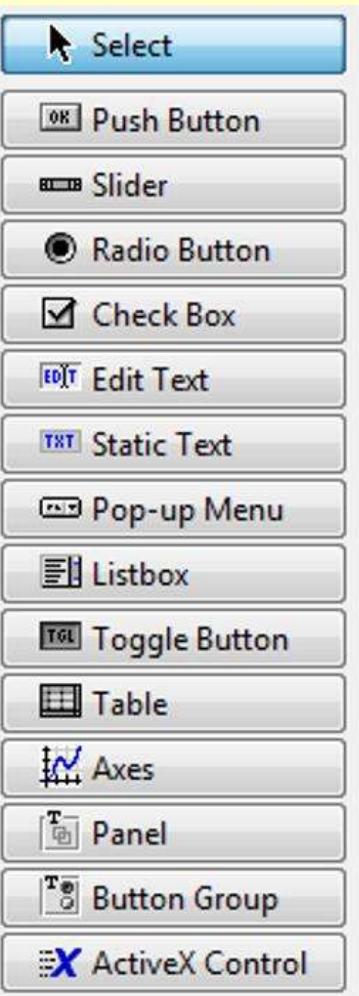


Menu Editor

M-file editor

RUN

Interface



Toolbars & Menus

Input Control components such as

- Push Buttons, Radio Buttons, Check Boxes (SWITCHES)
- Pop-up Menus, List-Boxes (SELECTIONS)
- Sliders (CONTINIOUS CONTROL)
- Edit Text (TEXT method of input)

Graphical Objects

- Axes Objects

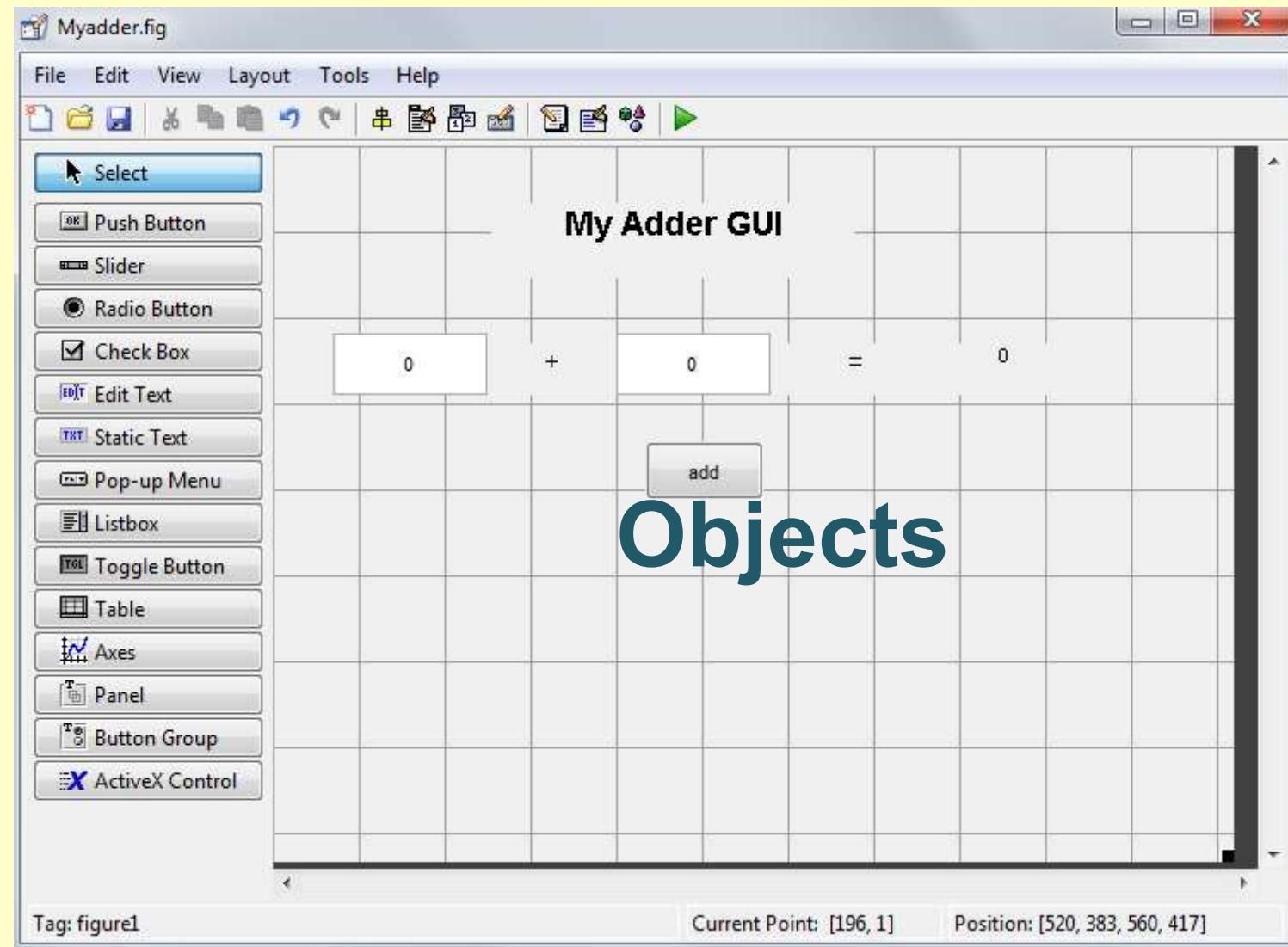
Text Objects

- Static Text

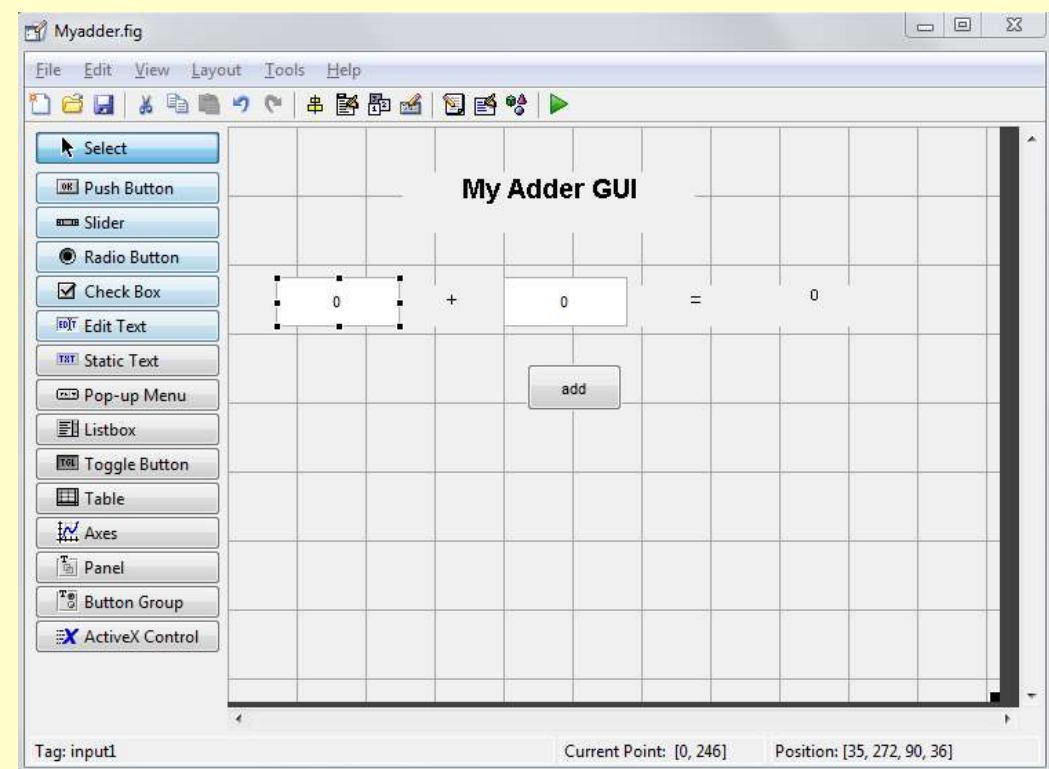
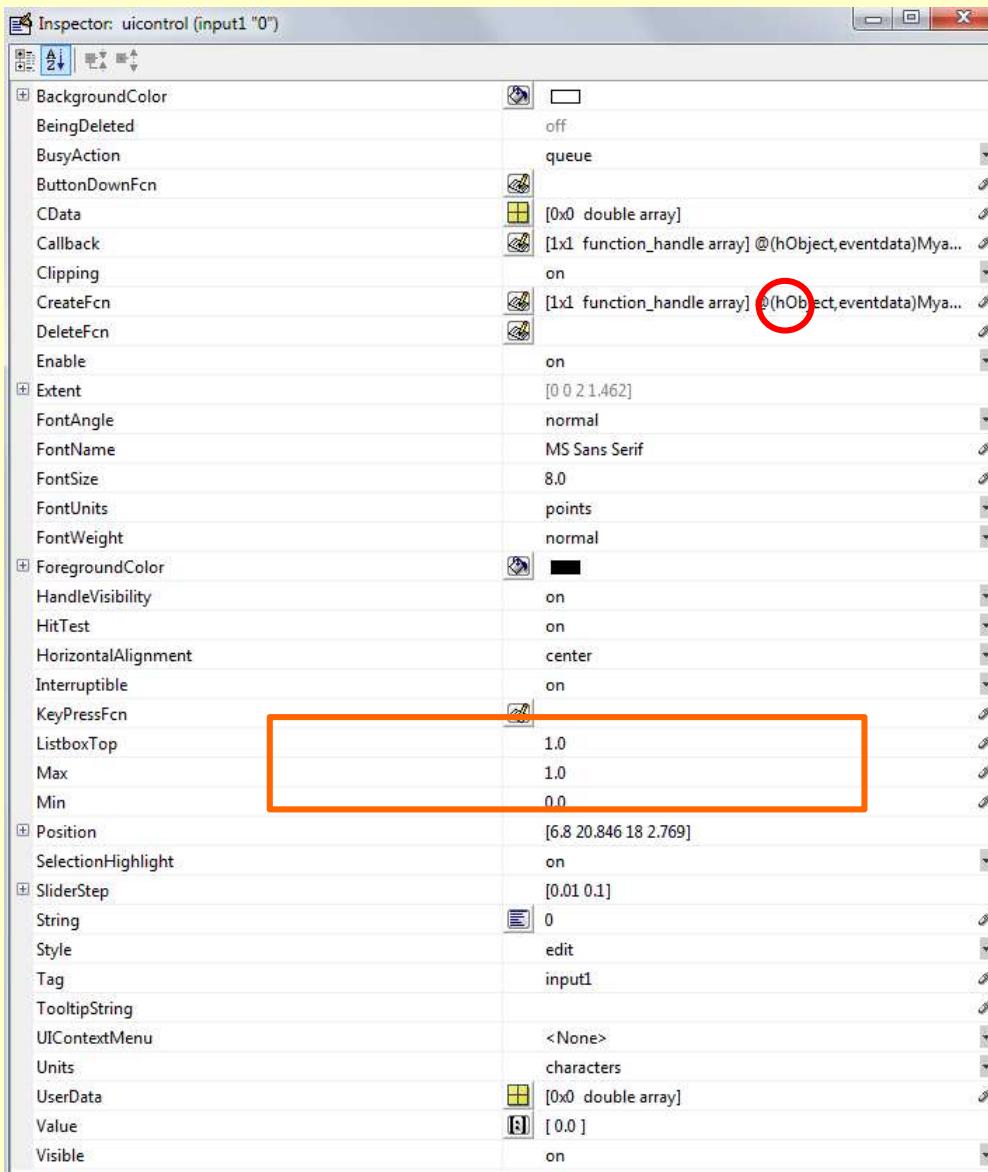
Objects-Properties-Event

Đối tượng	Push button, Sliders, Axes, Radio button...
Thuộc tính	Tag, String, FontSize, Value, BackgroundColor...
Sự kiện	Create, Callback,

Thiết kế giao diện



Properties



Event

```

 function input1_Callback(hObject, eventdata, handles) Cần khai báo global
 % hObject    handle to input1 (see GCBO)
 % eventdata reserved - to be defined in a future version of MATLAB
 % handles    structure with handles and user data (see GUIDATA)

 % Hints: get(hObject,'String') returns contents of input1 as text
 %         str2double(get(hObject,'String')) returns contents of input1 as a double
 input=str2num(get(hObject,'String'))

```

String	<input type="text"/> 0	<input type="button"/>
Style	<input type="button"/> edit	<input type="button"/>
Tag	input1	<input type="button"/>

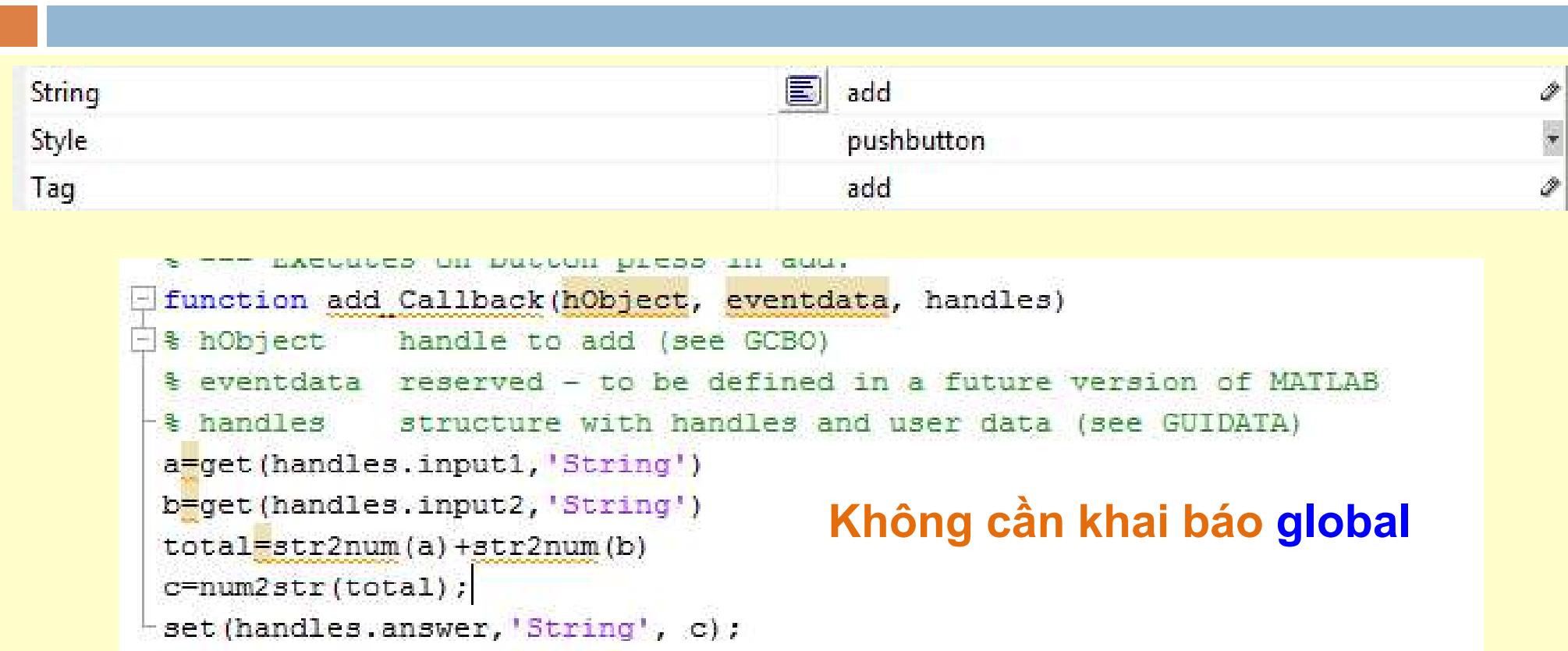
```

 function input1_Callback(hObject, eventdata, handles) Cần khai báo global
 % hObject    handle to input1 (see GCBO)
 % eventdata reserved - to be defined in a future version of MATLAB
 % handles    structure with handles and user data (see GUIDATA)

 % Hints: get(hObject,'String') returns contents of input1 as text
 %         str2double(get(hObject,'String')) returns contents of input1 as a double
 input=str2num(get(hObject,'String'))

```

Event



The screenshot shows the MATLAB GUI editor. At the top, there are three tabs: 'String', 'Style', and 'Tag'. The 'Tag' tab is selected, showing the handle 'add' for a pushbutton. Below the tabs is a code editor window containing the following MATLAB code:

```
% --- Executes on button press in add.
function add_Callback(hObject, eventdata, handles)
% hObject    handle to add (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
a=get(handles.input1,'String')
b=get(handles.input2,'String')
total=str2num(a)+str2num(b)
c=num2str(total);
set(handles.answer,'String', c);
```

Không cần khai báo global

Hãy phân biệt cách gọi hàm dùng
handles....và hObject?

Handles vs hObject

Handles

Truy cập các đối tượng khác qua Tag name

hObject

Áp dụng trên chính đối tượng

Cần khai báo global
để sử dụng

Get, set

```
Variable = get(hObject , 'Properties');
```

```
a = get(hObject, 'String');  
b = get(handles.inputB, 'String');
```

```
get(hObject , 'Properties', Value);
```

```
set(hObject, 'String', 'ABC');  
set(handles.staticTextC, 'String', c);  
% c là một biến chứa chuỗi
```

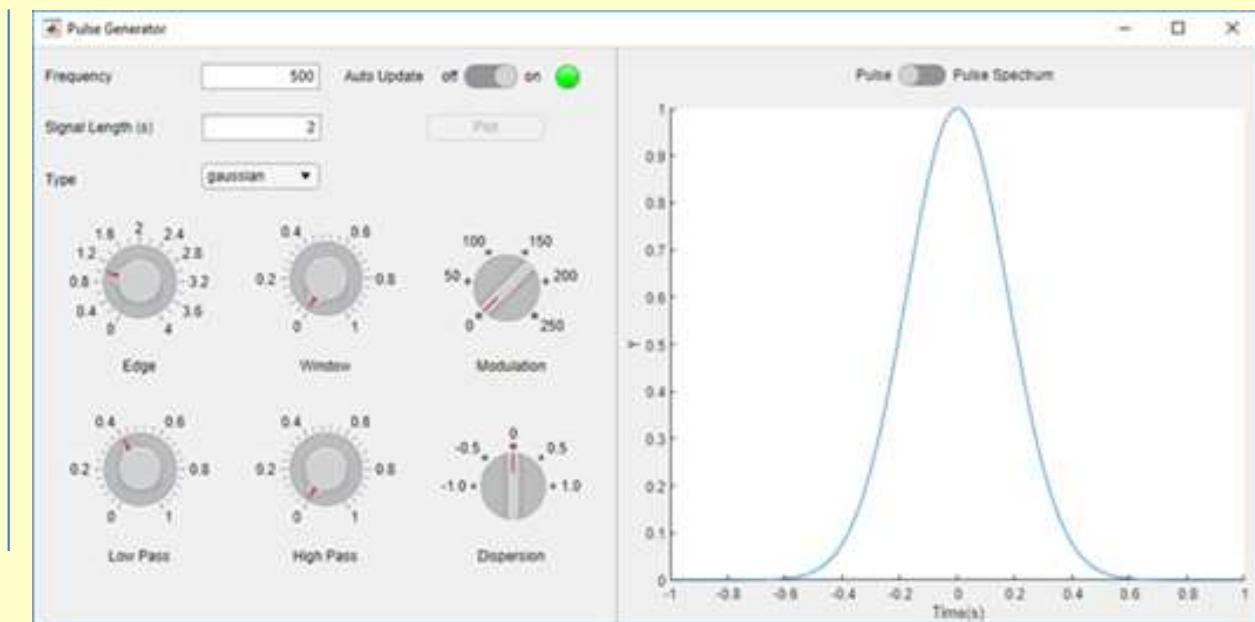
Axes

```
axes(handles.axesTag);
plot(x, y)
guidata(hObject, handles);
```

```
axes(handles.axes1);
x=0:10
y=x.^2
plot(x,y)
```

App Designer

Thiết kế với giao diện



App Designer
Create desktop and web apps in MATLAB

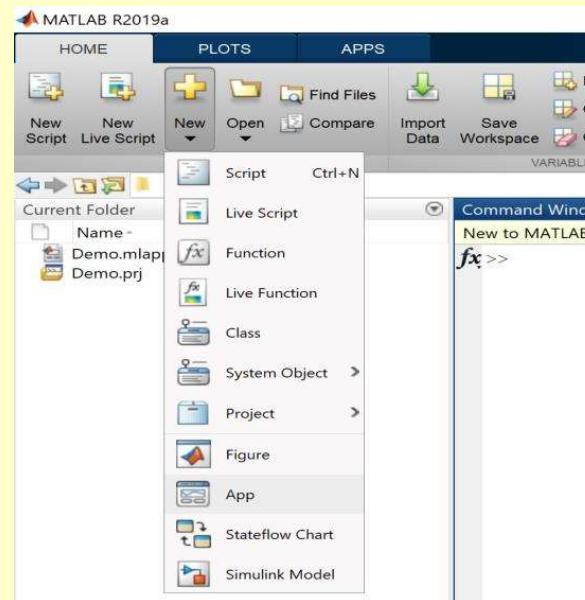
Khởi động App Designer

Home -> New -> App

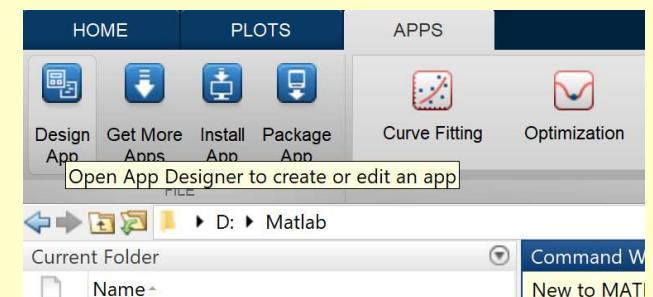
Gõ lệnh appdesigner

```
VARIABLE          CODE          SIMULIN

Command Window
New to MATLAB? See resources for Getting Started.
>> appdesigner
fx >> |
```



App -> Design App



Các đối tượng giao diện trong App Designer

The screenshot shows the MATLAB App Designer interface. On the left is the 'COMPONENT LIBRARY' pane, which is highlighted with a red border. It contains sections for 'COMMON' components like Axes, Button, Check Box, Date Picker, Drop Down, Edit Field (Numeric), Edit Field (Text), Image, Label, List Box, Radio Button Group, Spinner, State Button, and Text Area. In the center is the 'CANVAS' area where a plot titled 'Tín hiệu' (Signal) is displayed. The plot shows a sine wave over time. Below the plot are three input fields: 'Nhập hàm cần vẽ theo x' with value 'sin(x)+cos(2*x)', 'Nhập vào tần số lấy mẫu (Hz)' with value '1000', and 'Nhập vào thời gian vẽ (t)' with value '5'. A 'Vẽ tín hiệu' button is also present. On the right is the 'COMPONENT BROWSER' pane, which is highlighted with a blue border. It lists components under 'app.UIFigure' such as app.EditFieldFs, app.EditFieldT, app.buttonDraw, app.UIAxes, and app.EditFieldFunction. It also includes tabs for 'Inspector' and 'Callbacks', and sections for 'WINDOW APPEARANCE', 'POSITION', 'PLOTTING', 'CALLBACK EXECUTION CONTROL', 'IDENTIFIERS', and 'INTERACTIVITY'.

Giao diện

Tùy chỉnh thuộc tính của đối tượng

Get, set, axis

```
Variable = app.ComponentName.Properties
```

```
a = app.EditFieldA.Value;  
b = app.EditFieldA.BackgroundColor;
```

```
app.ComponentName.Properties= value
```

```
app.EditFieldTextA.Value = 'ABC';  
app.EditFieldTextA.Value = c;  
% c là một biến chứa chuỗi
```

```
plot(app.UIAxes, x, y);
```

```
x=0:10  
y=x.^2  
plot(app.AxesB, x,y)
```

Bài tập 1

Thiết kế giao diện và viết chương trình để vẽ tín hiệu

$s = 2 * \sin(2 * \pi * f * n)$ với các thông số được nhập vào từ giao

diện: f: tần số tín hiệu

fs: tần số lấy mẫu tín hiệu

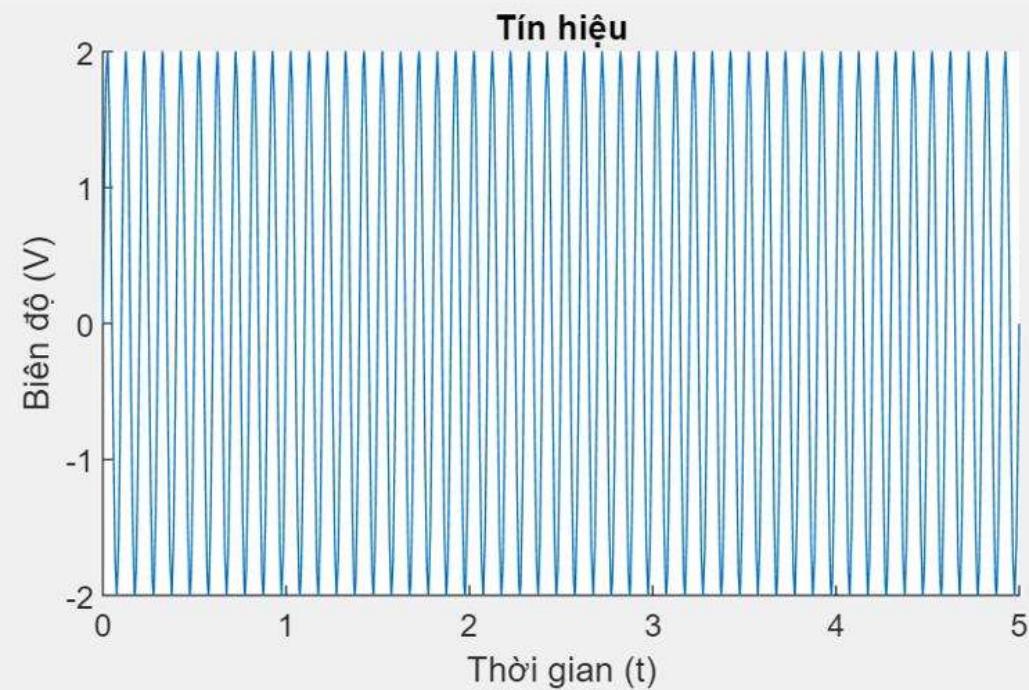
t: thời gian vẽ tín hiệu

Nhập vào tần số tín hiệu (Hz)

Nhập vào tần số lấy mẫu (Hz)

Nhập vào thời gian vẽ (t)

Vẽ tín hiệu



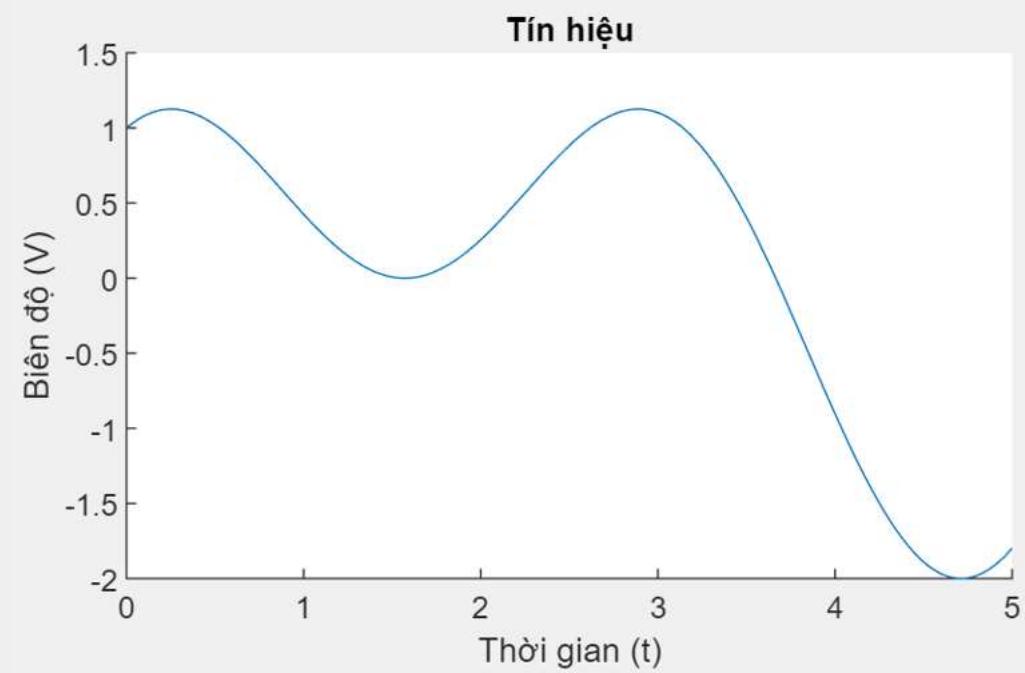
Bài 2

Thực hiện lại bài 1 với hàm số được nhập vào từ giao diện
Lưu ý: Hàm số là hàm một biến phụ thuộc x

Nhập hàm cần vẽ theo x

Nhập vào tần số lấy mẫu (Hz)

Nhập vào thời gian vẽ (t)



Bài 3

Viết giao diện

Nhập vào 1 hàm số 1 biến

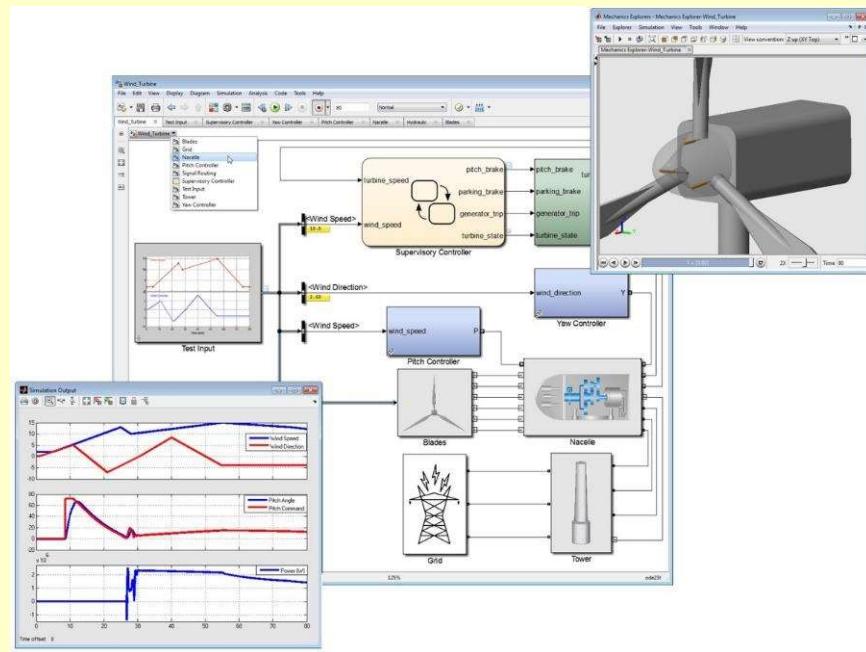
Các nút nhấn có các chức năng: Tìm nghiệm, Phân tích
thành nhân tử

Label để hiển thị kết quả

Viết chương trình thực hiện chức năng trên

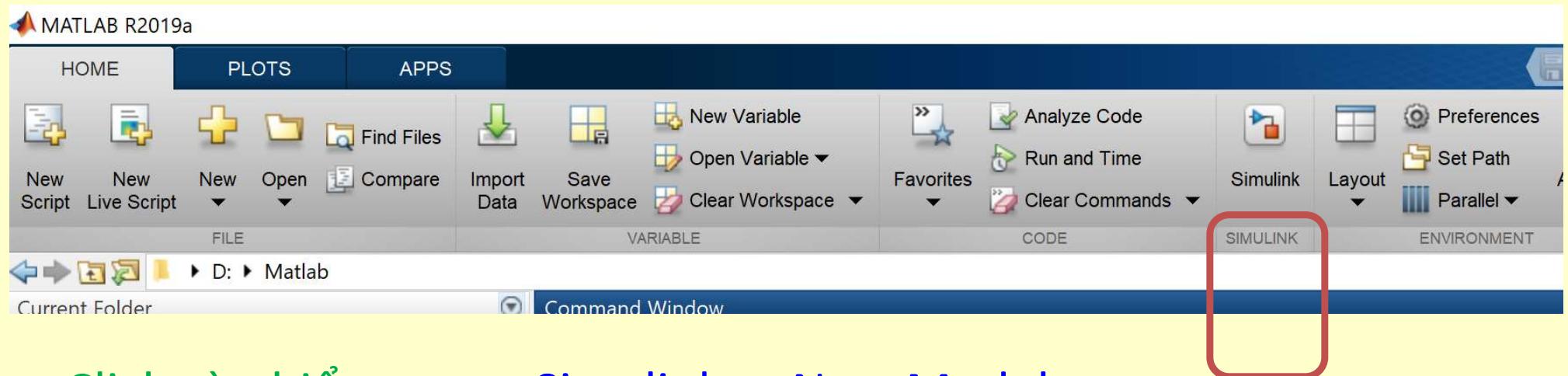
Simulink

Thiết kế và mô phỏng
với mô hình

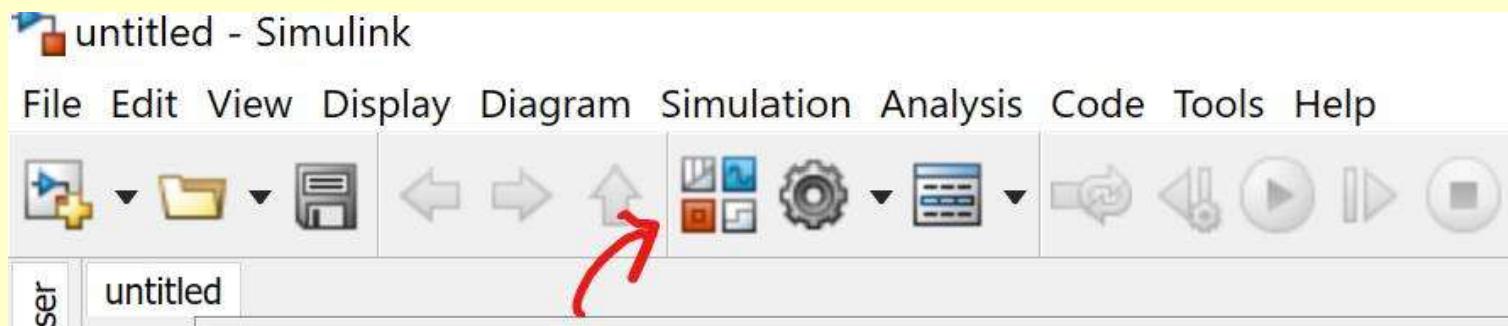


Start Simulink

Gõ lệnh **simulink** trong Command Windows
Hoặc click vào biểu tượng Simulink

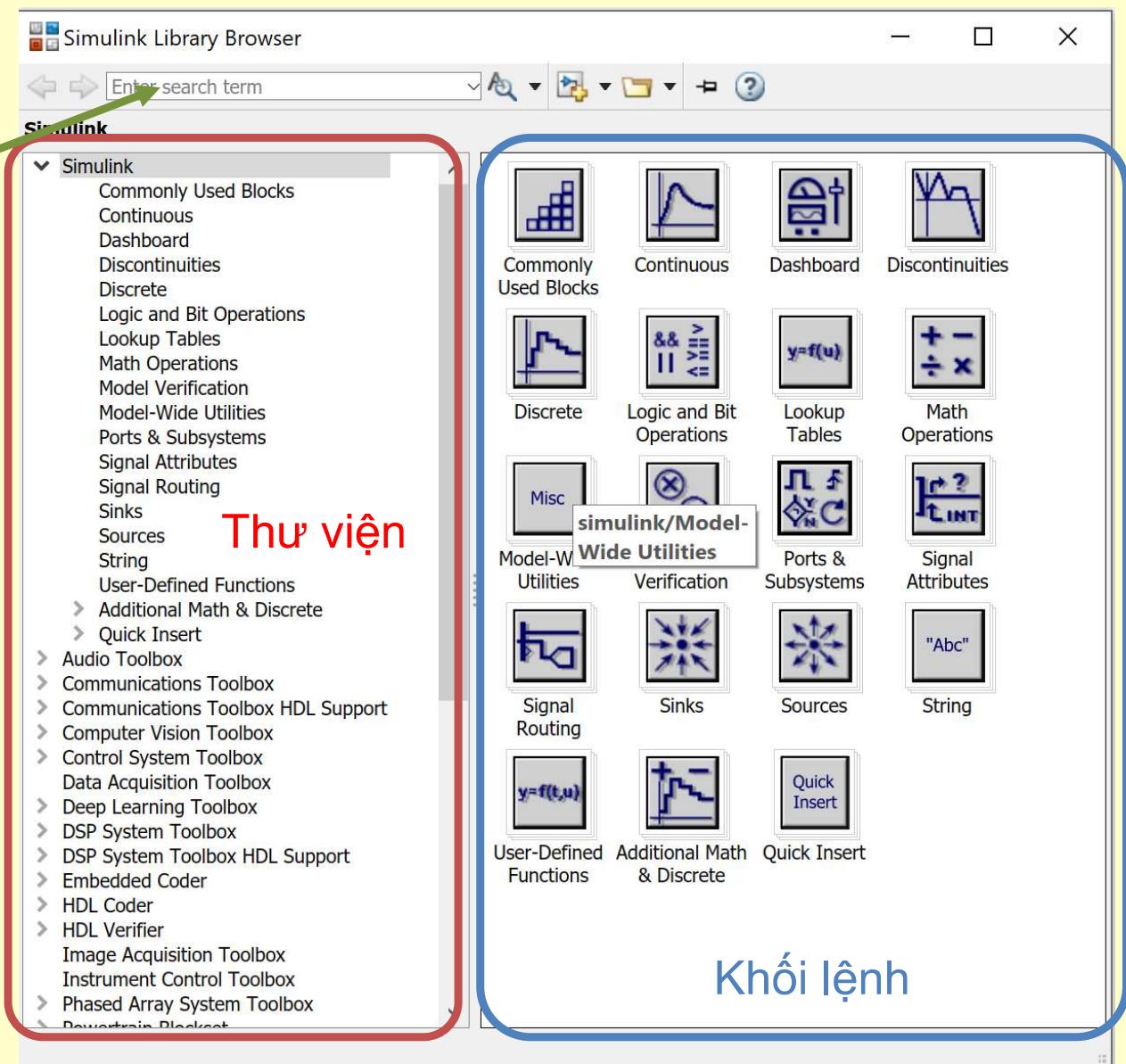


Click vào biểu tượng Simulink -> New Model



Library Browser

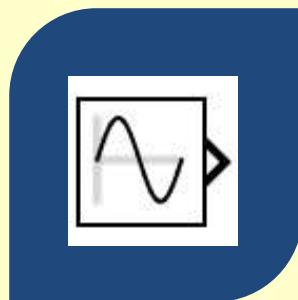
Tìm kiếm



Thư viện

Khối lệnh

Thư viện hay dùng



SOURCES

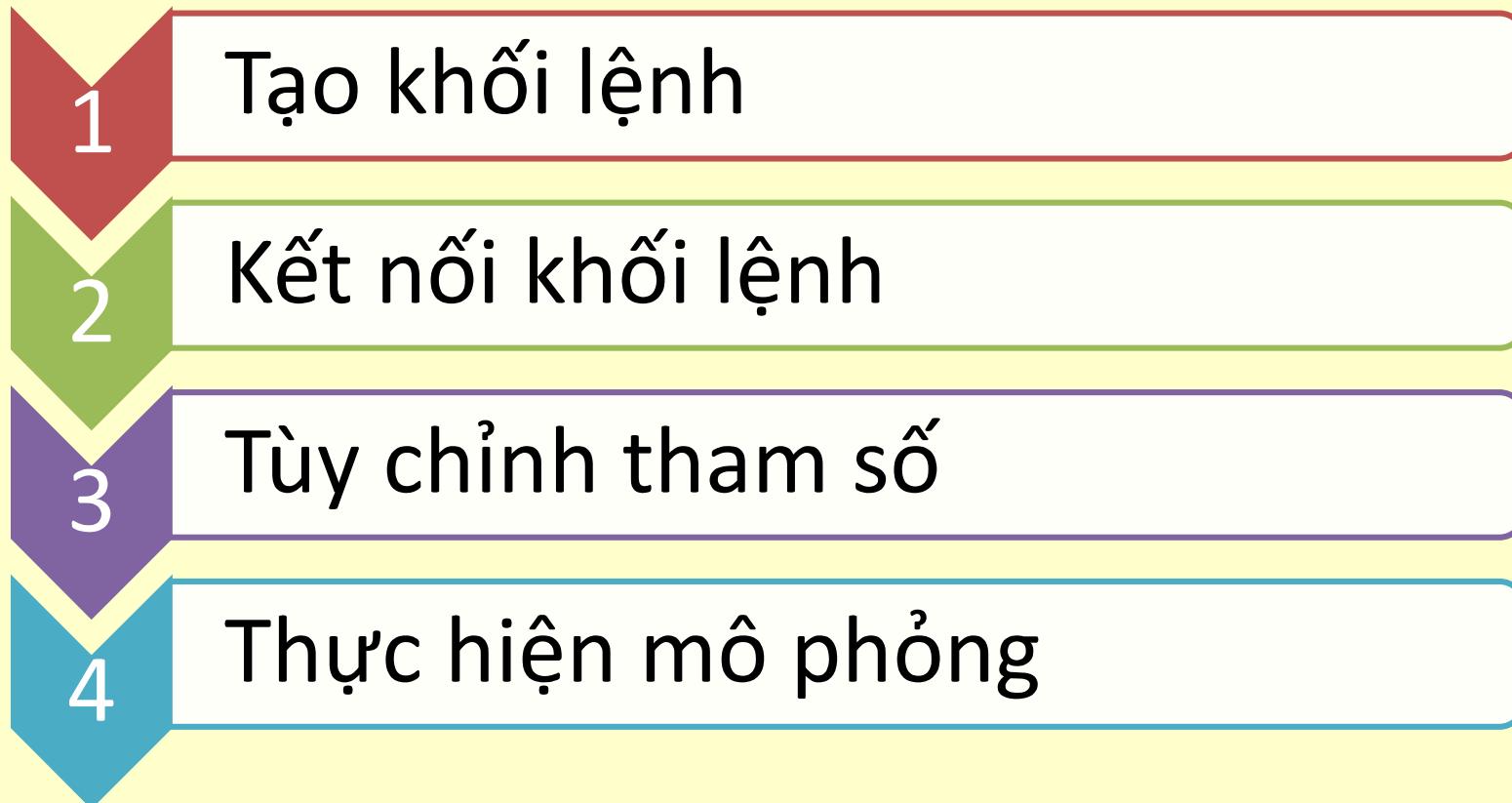
Nguồn tín hiệu, đầu vào



SINKS

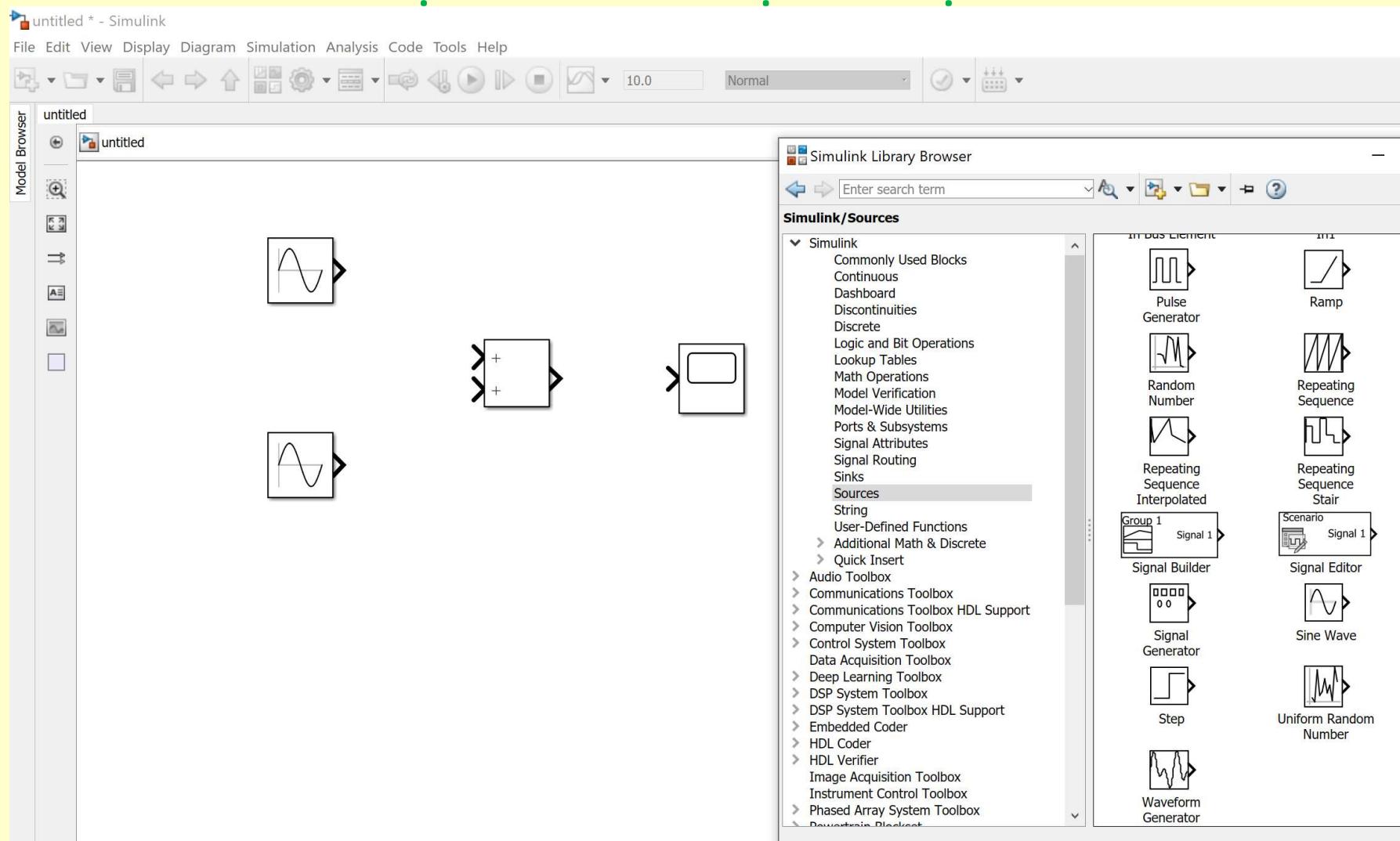
Hiển thị trực quan, thiết bị đo

Building a Model

- 
- 1 Tạo khối lệnh
 - 2 Kết nối khối lệnh
 - 3 Tùy chỉnh tham số
 - 4 Thực hiện mô phỏng

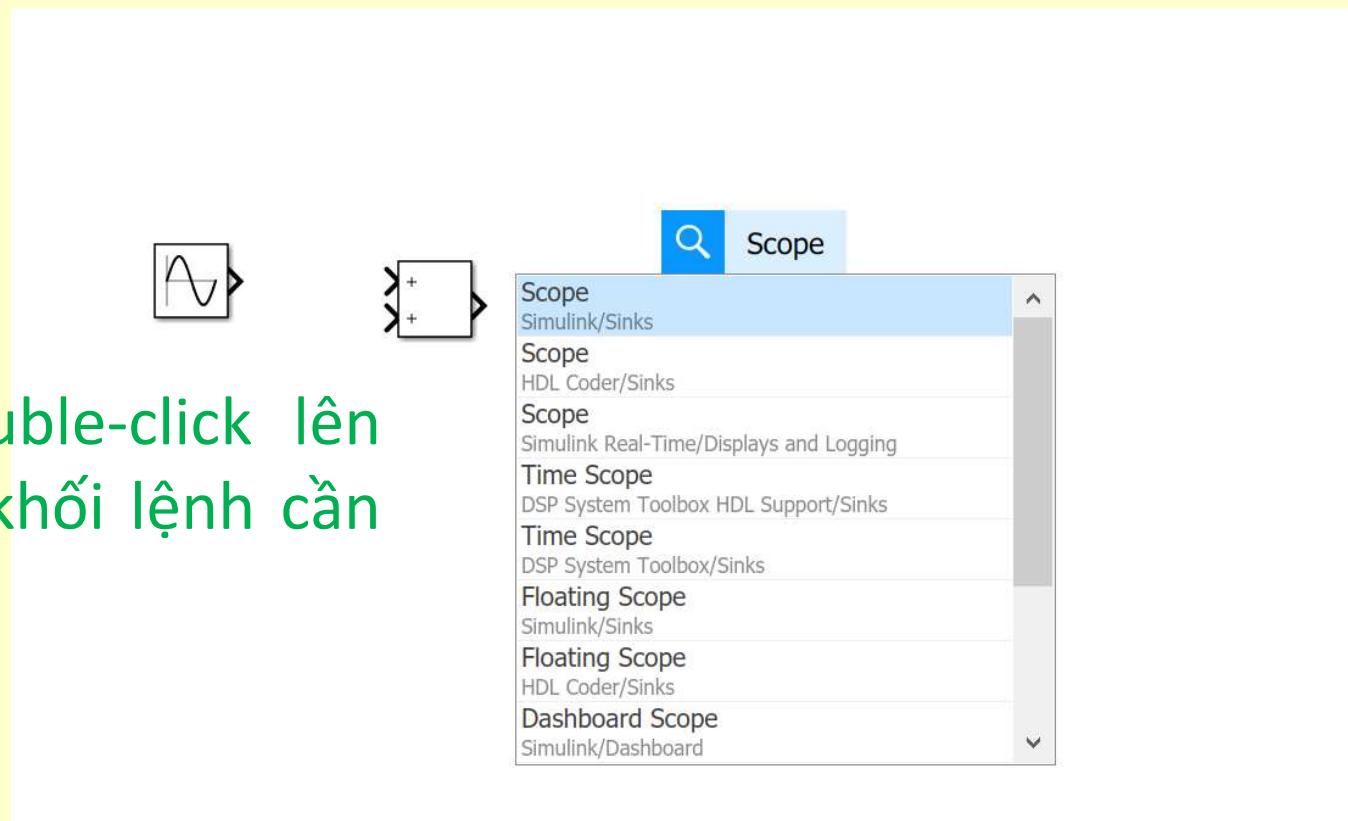
Tạo khối lệnh

Kéo thả khối lệnh từ thư viện để tạo mô hình

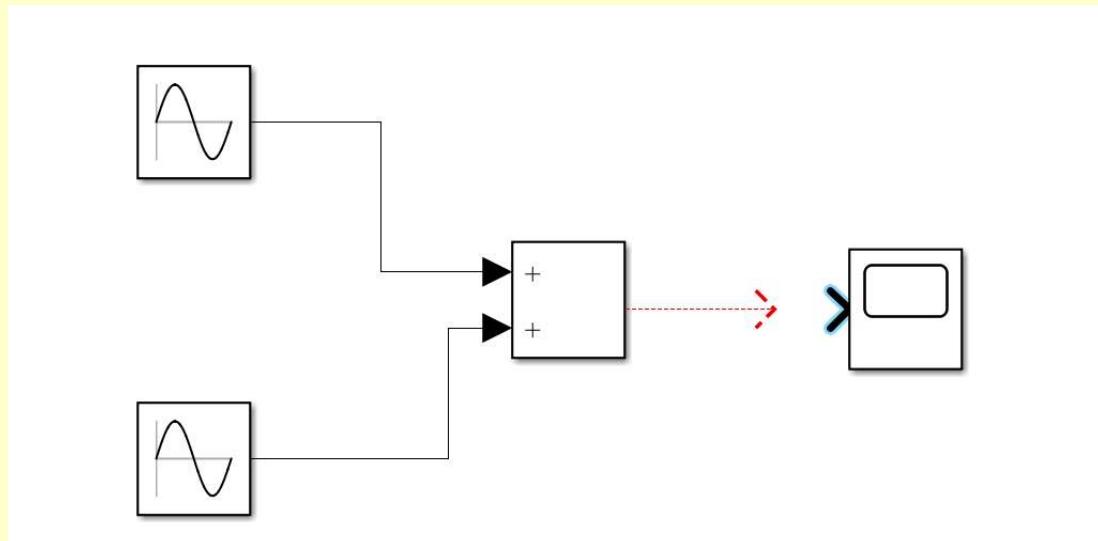


Lưu ý

Chúng ta có thể double-click lên vùng vẽ để tìm kiếm khối lệnh cần dùng



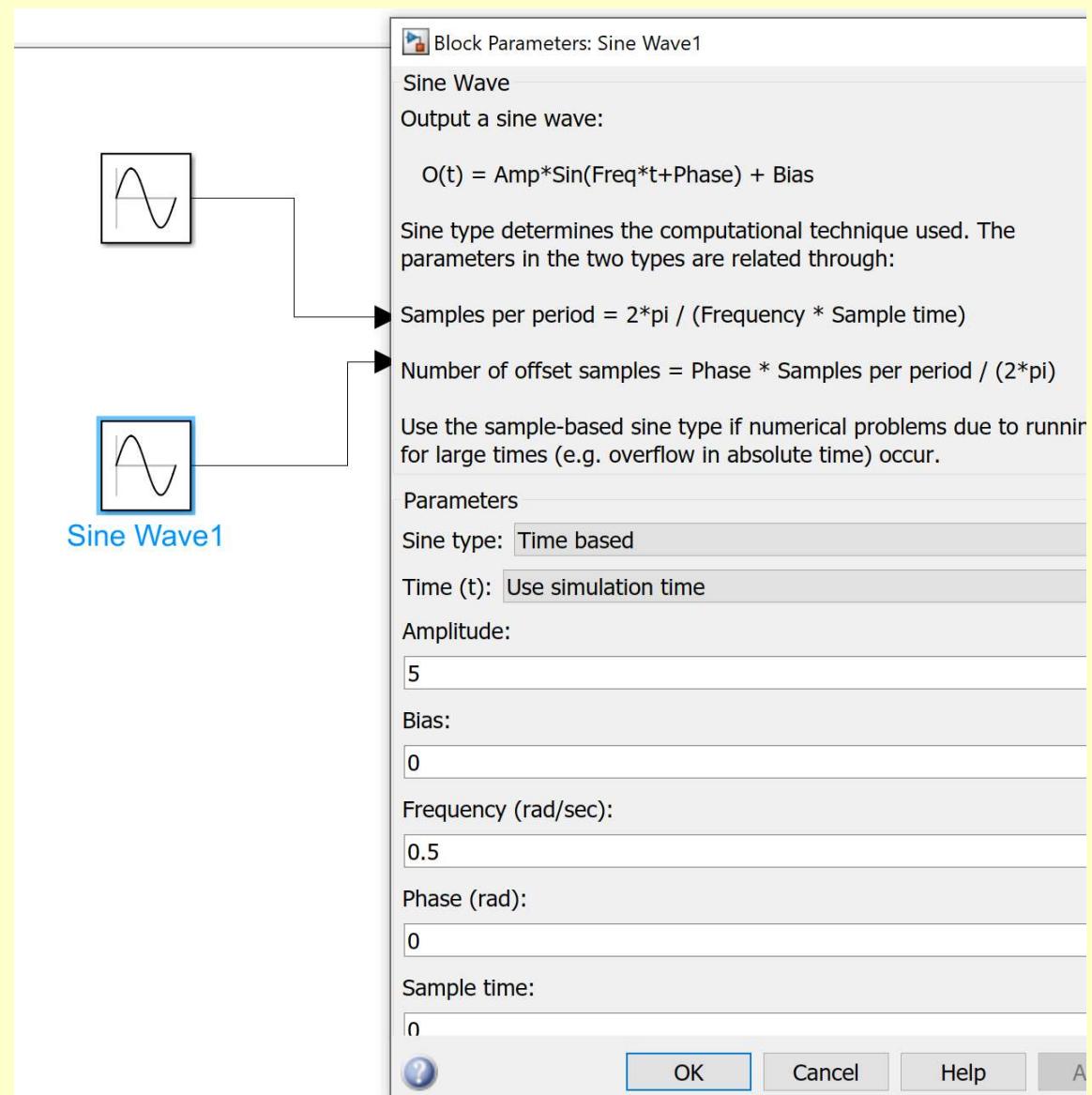
Kết nối khối lệnh



Kéo chuột từ các đầu vào ra để kết nối

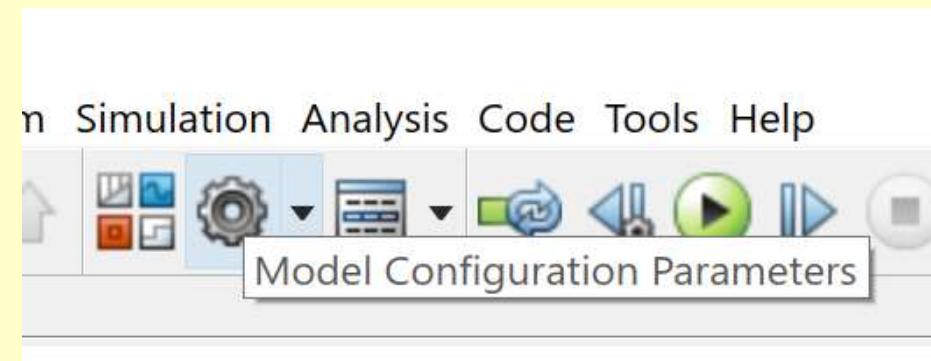
Tùy chỉnh tham số

Double-click lên khối lệnh để tùy chỉnh tham số



Cấu hình mô phỏng

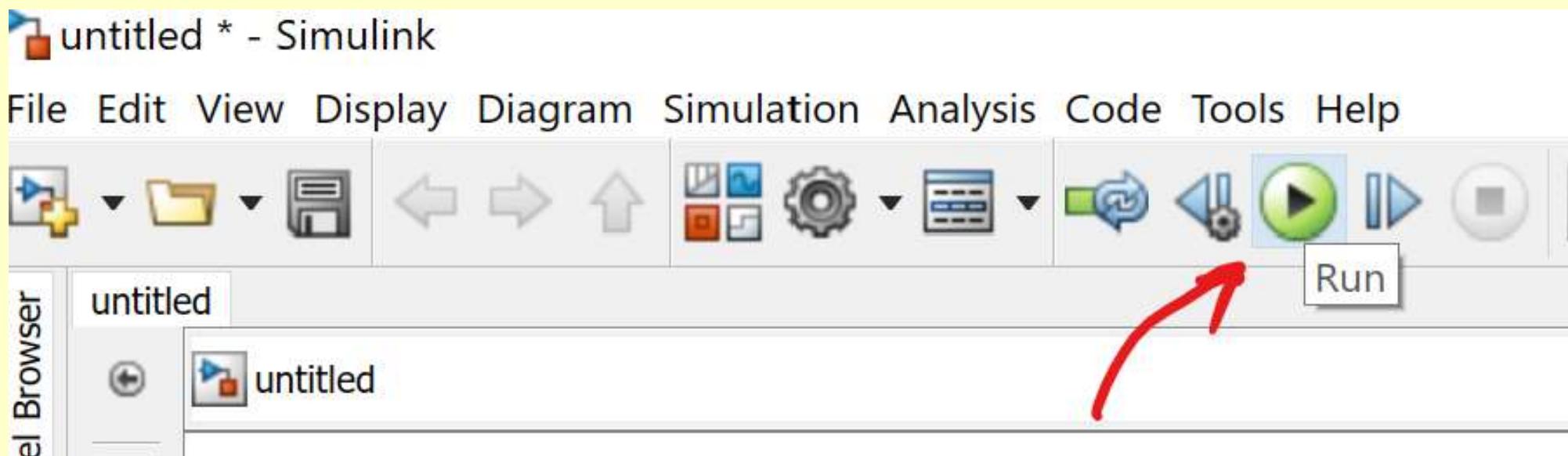
Vào biểu tượng setting, sau đó
cấu hình thời gian mô phỏng



The screenshot shows the 'Configuration Parameters' dialog box for a Simulink model named 'untitled/Configuration (Active)'. The left sidebar has a 'Solver' tab selected, showing a list of other tabs: Data Import/Export, Math and Data Types, Diagnostics, Hardware Implementation, Model Referencing, Simulation Target, Code Generation, and Coverage. The main panel displays solver settings under the 'Solver' tab. It includes fields for 'Start time' (0.0) and 'Stop time' (50.0). Under 'Solver selection', 'Type' is set to 'Variable-step' and 'Solver' is set to 'auto (Automatic solver selection)'. There is also a 'Solver details' section and an ellipsis (...).

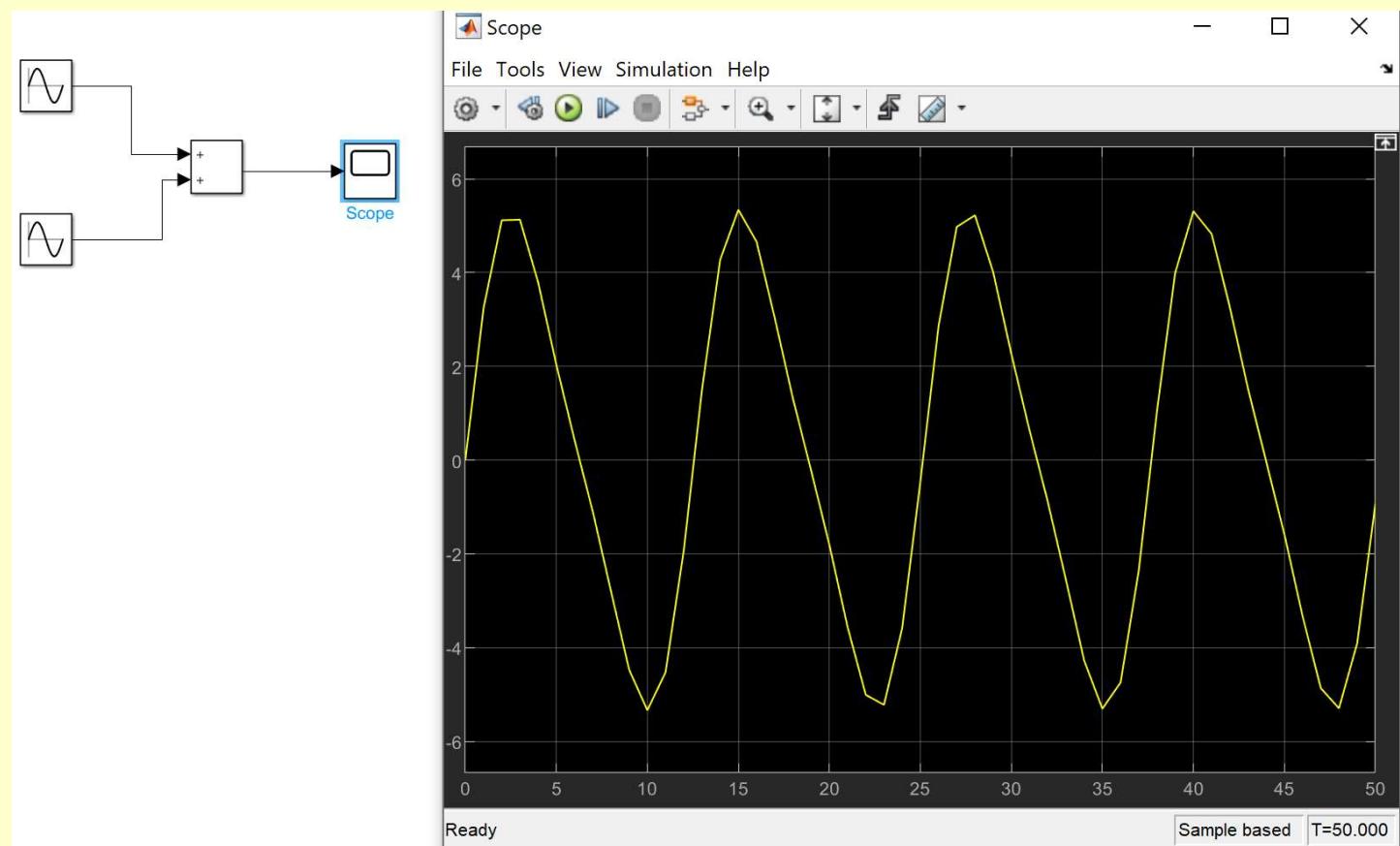
Thực hiện mô phỏng

Nhấn vào biểu tượng Run để mô phỏng

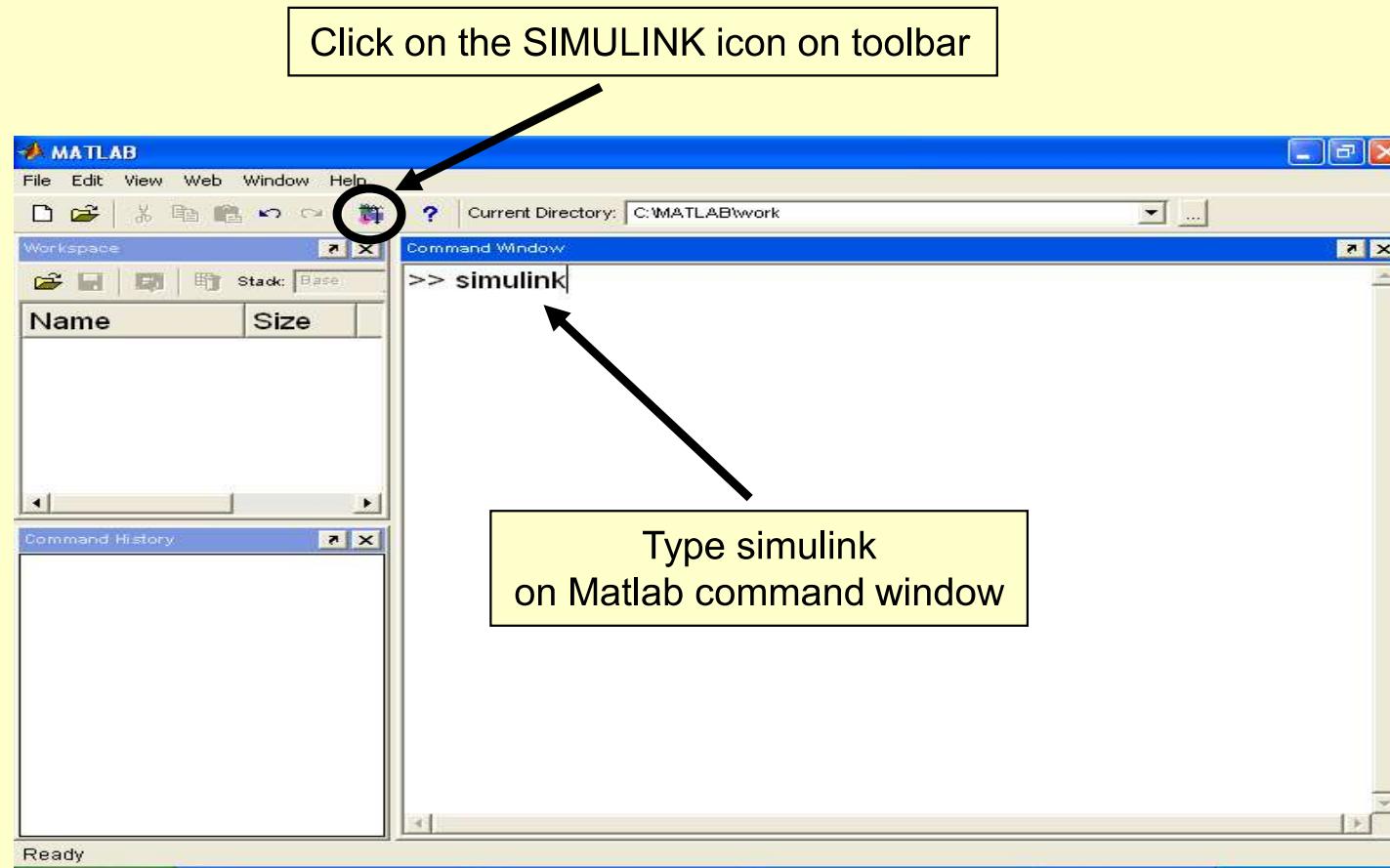


Quan sát kết quả

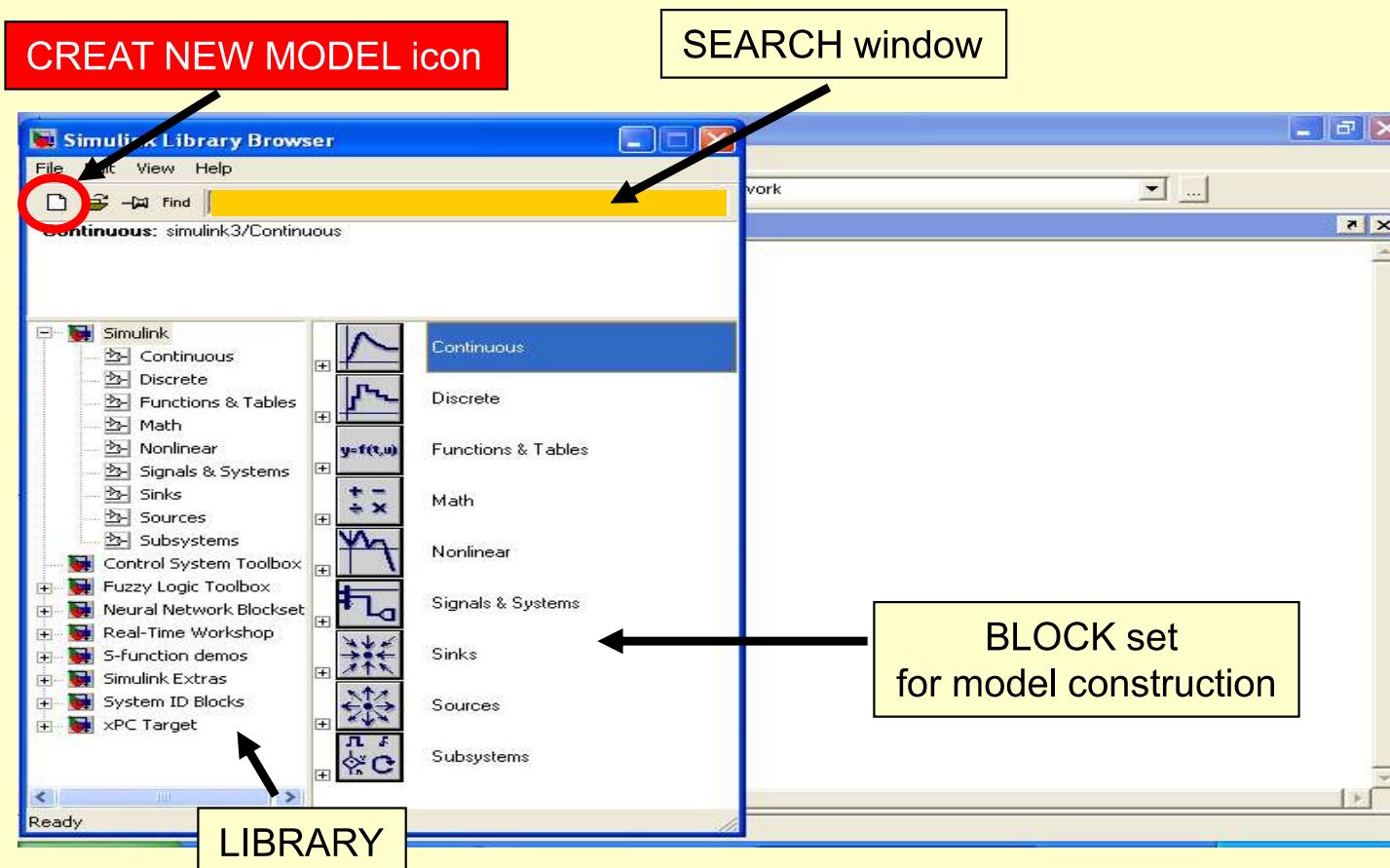
Double-click vào khối hiển thị để xem kết quả



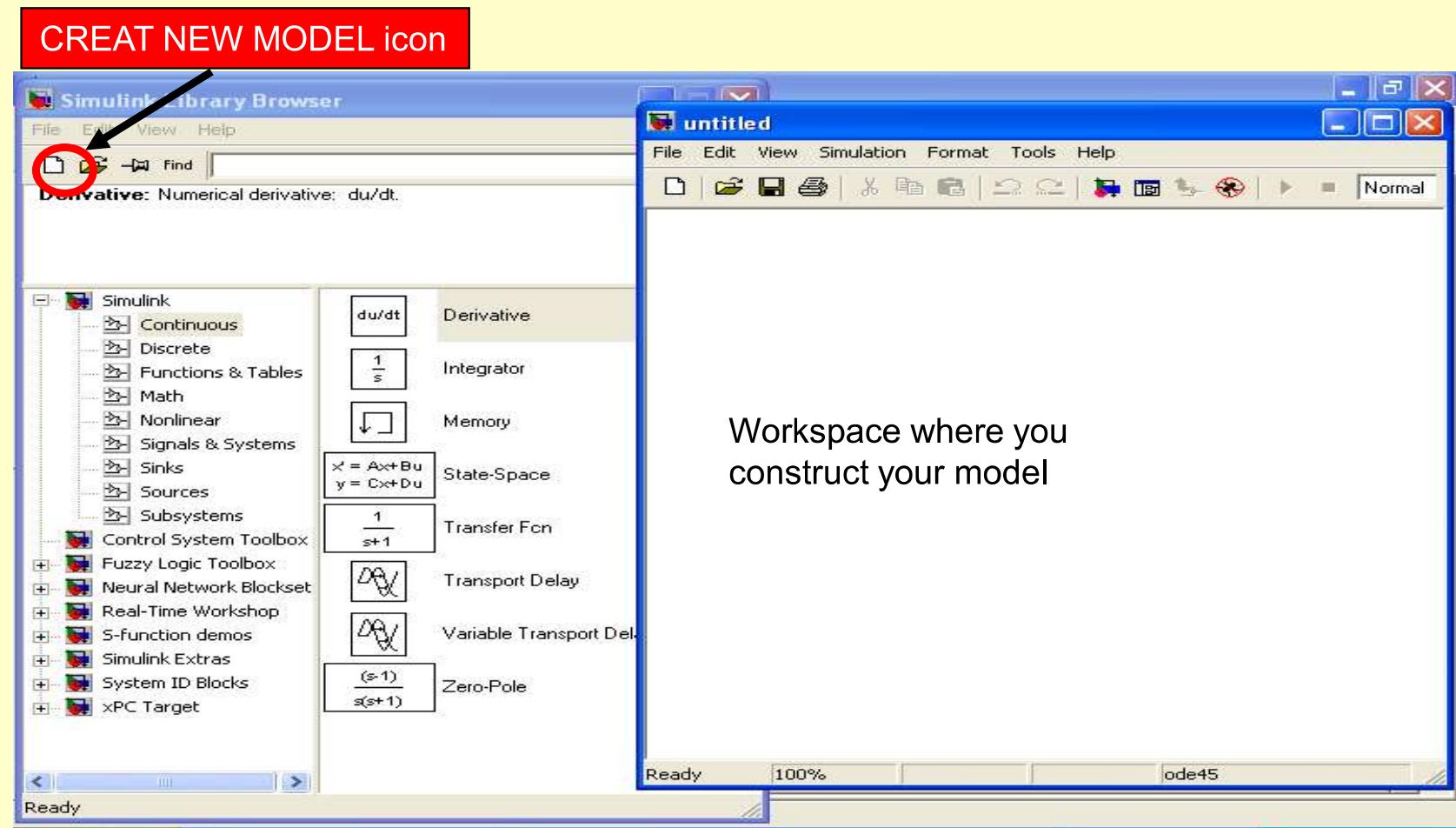
Phiên bản Matlab cũ Start a Simulink Session



Simulink Library Browser

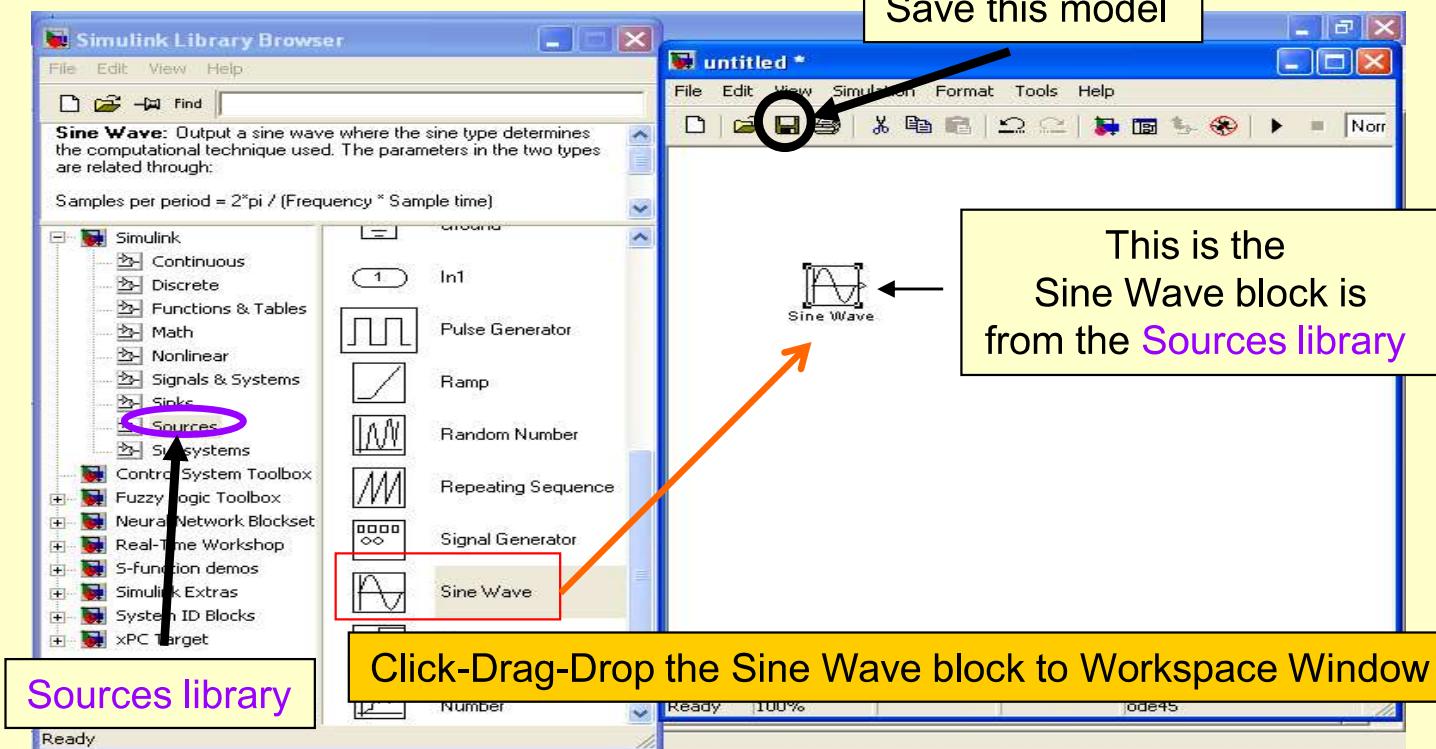


Create a New Model



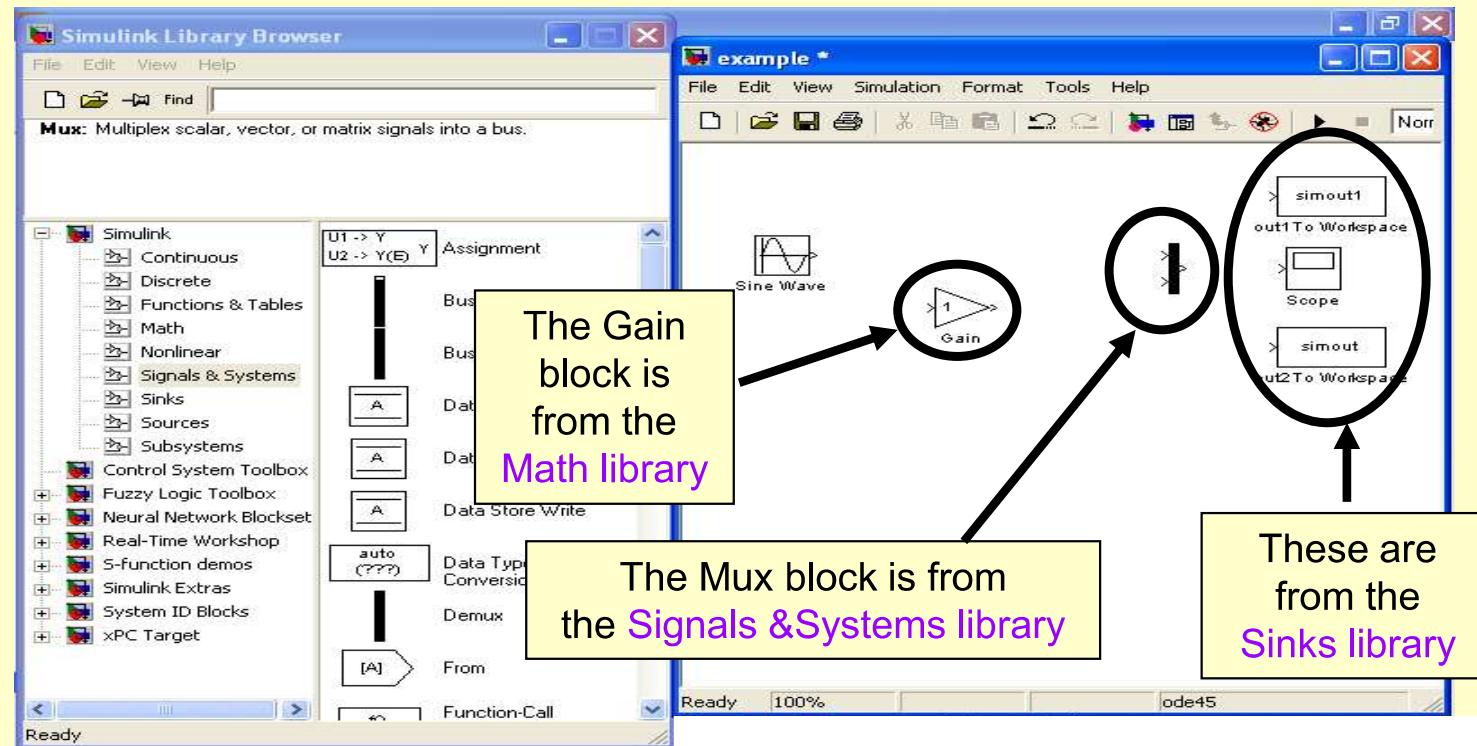
Building a Model

Step 1: Creating Blocks



Building a Model

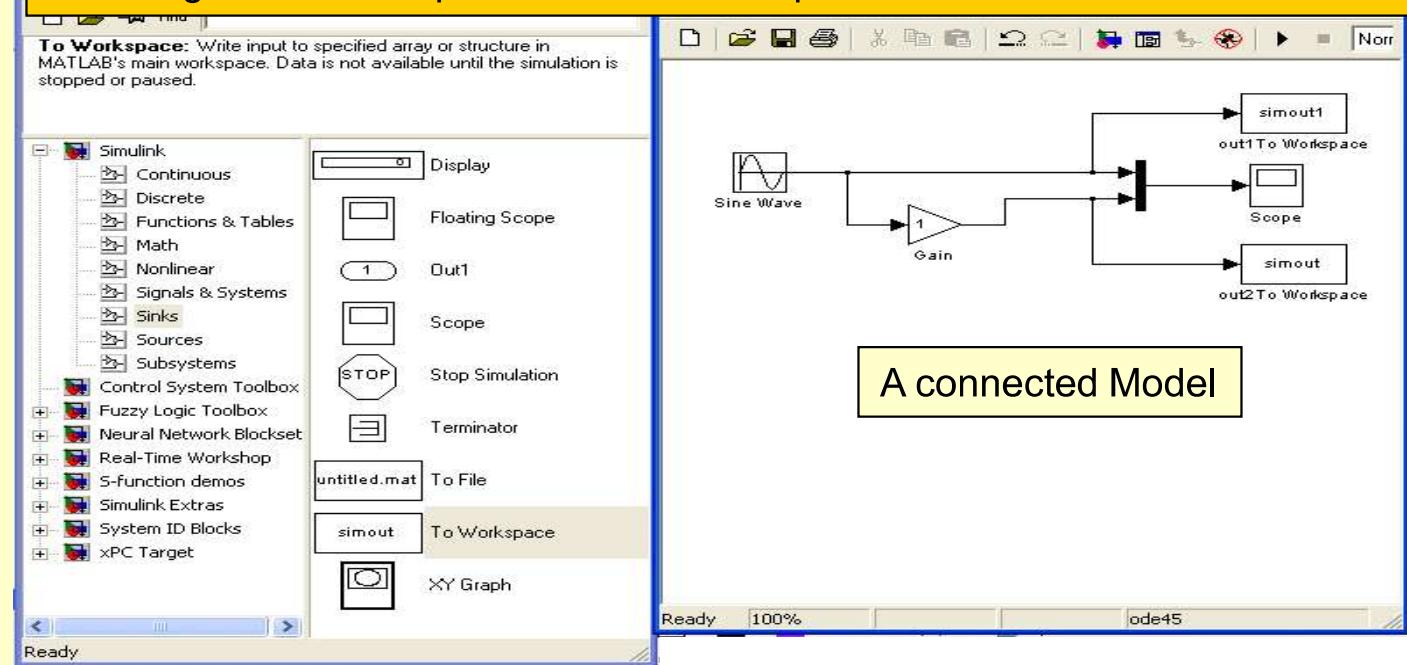
Step 1: Creating Blocks



Building a Model

Step 2: Making connections

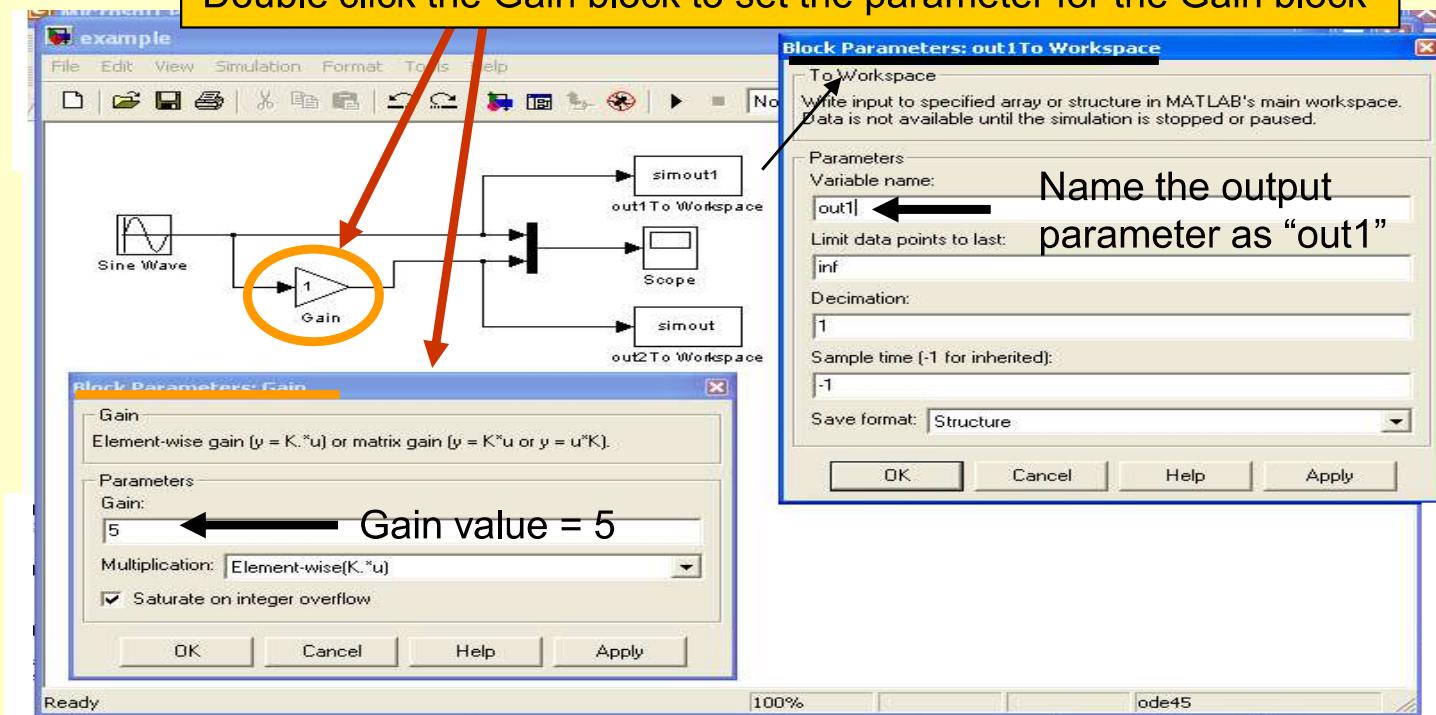
To make connection: left-click while holding down *control key* (on keyboard) and drag from source port to a destination port



Building a Model

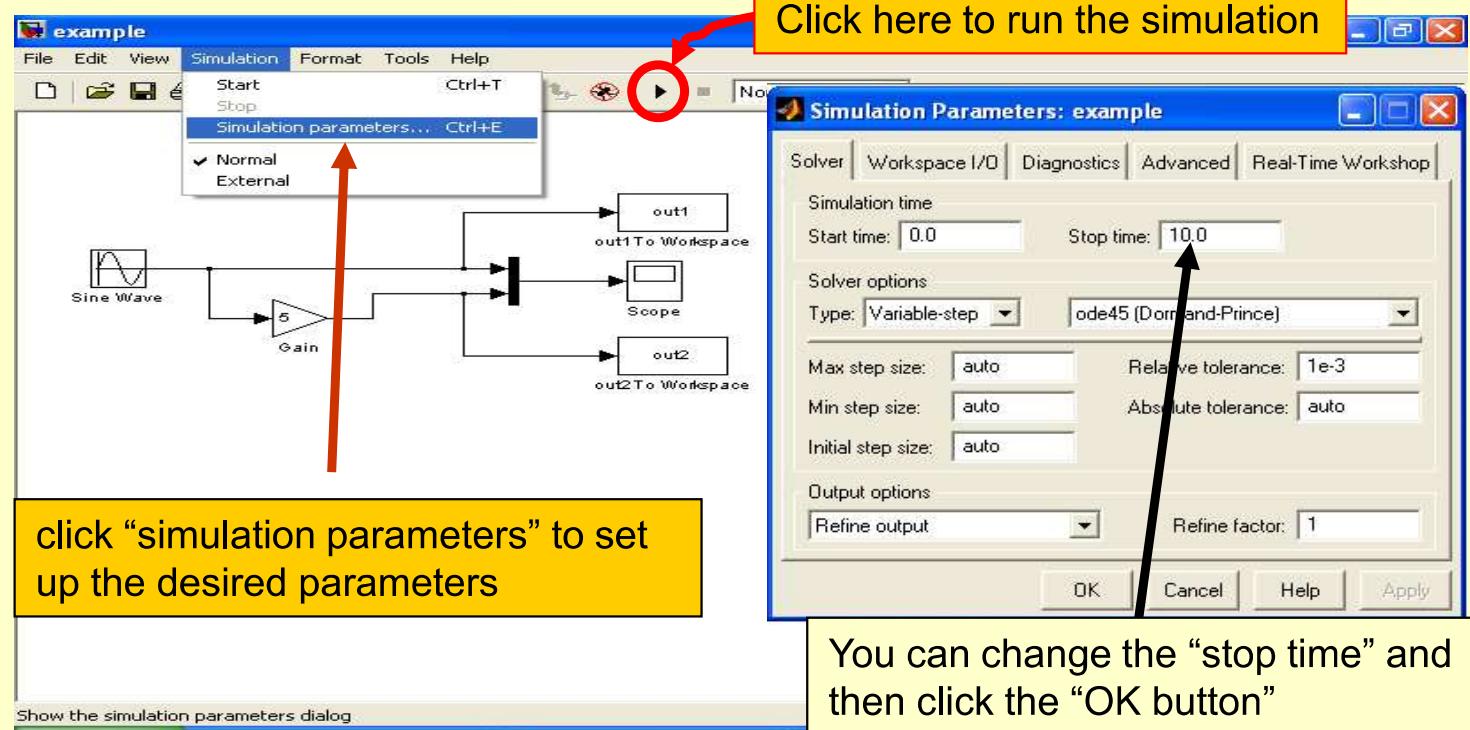
Step 3: Set Parameters

Double click the Gain block to set the parameter for the Gain block



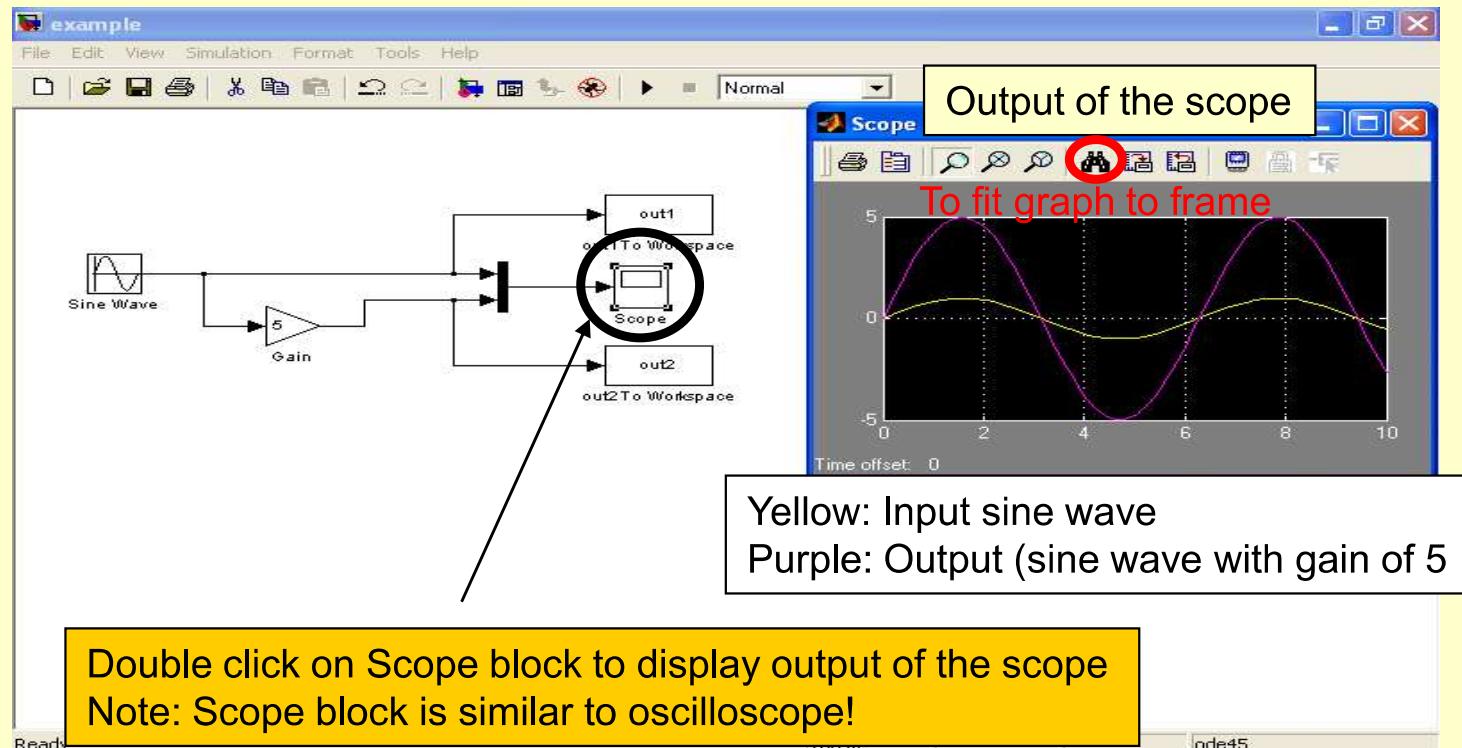
Building a Model

Step 4: Running Simulation



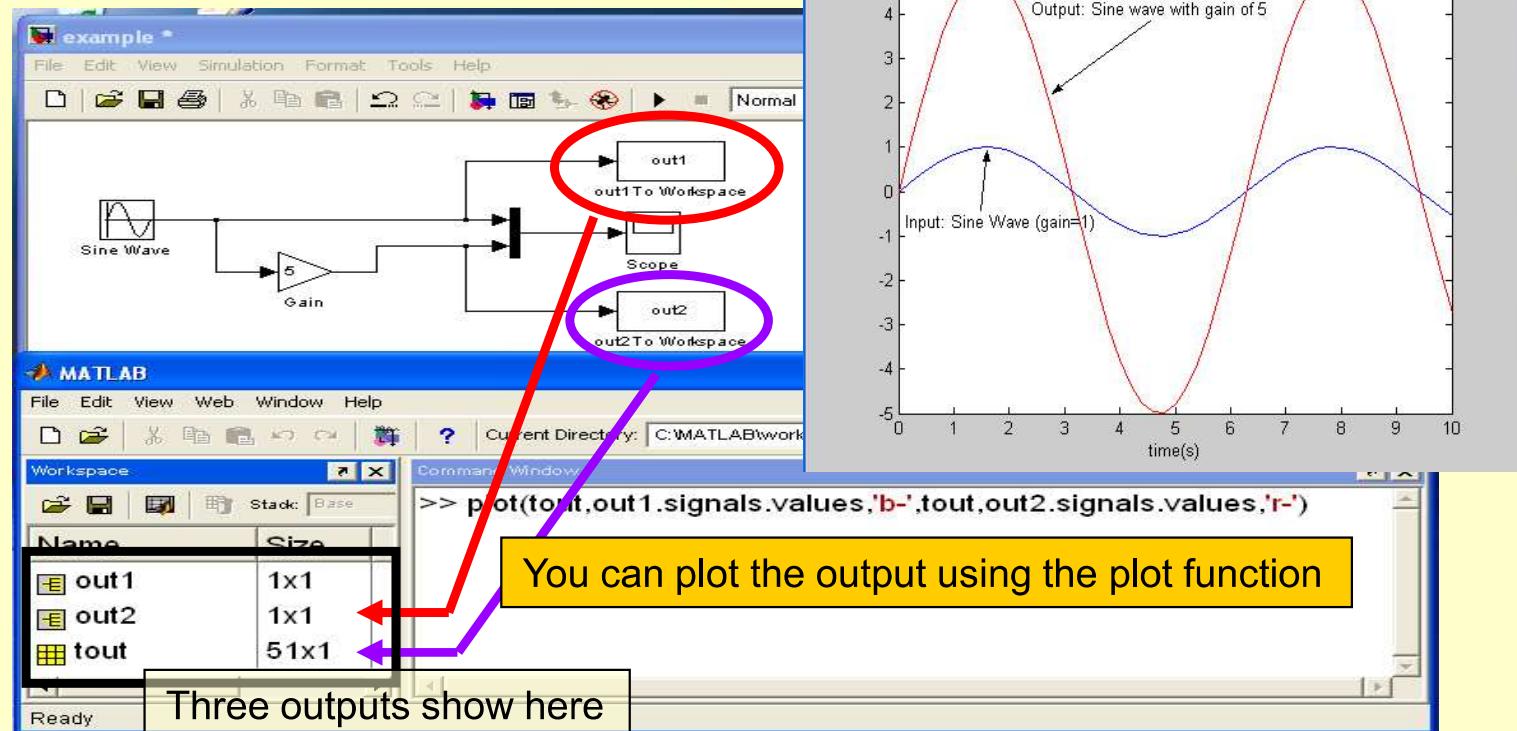
Building a Model

View output via Scope block

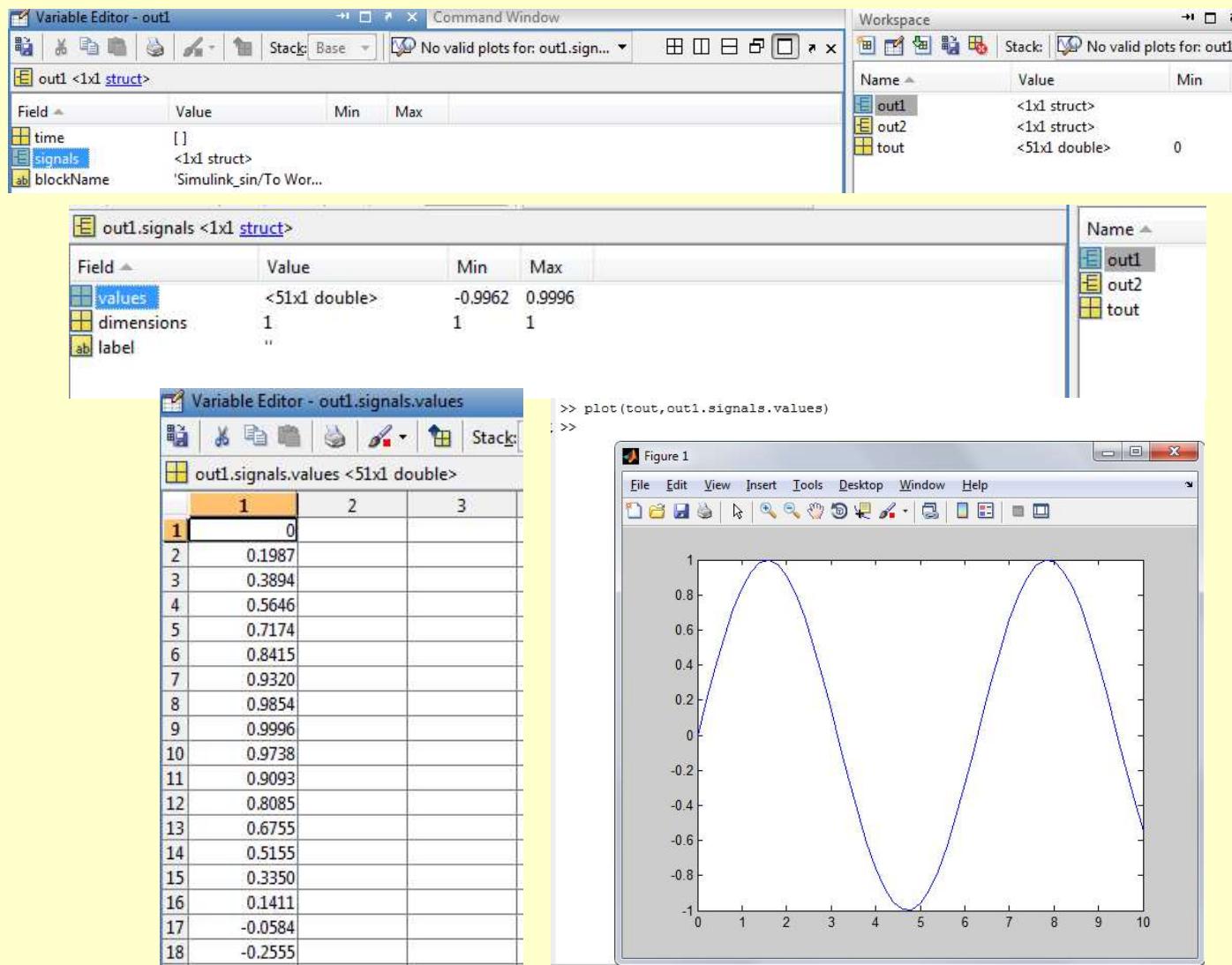


Building a Model

View output (*workspace*)



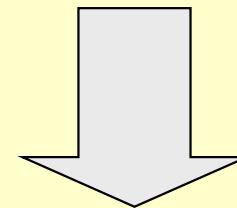
Biến, workspace



Example: Transfer Function

$$\ddot{x} = \frac{1}{2}(-2\dot{x} - 2x + f(t))$$

The transfer function of the equation
(assume all initial conditions =0)



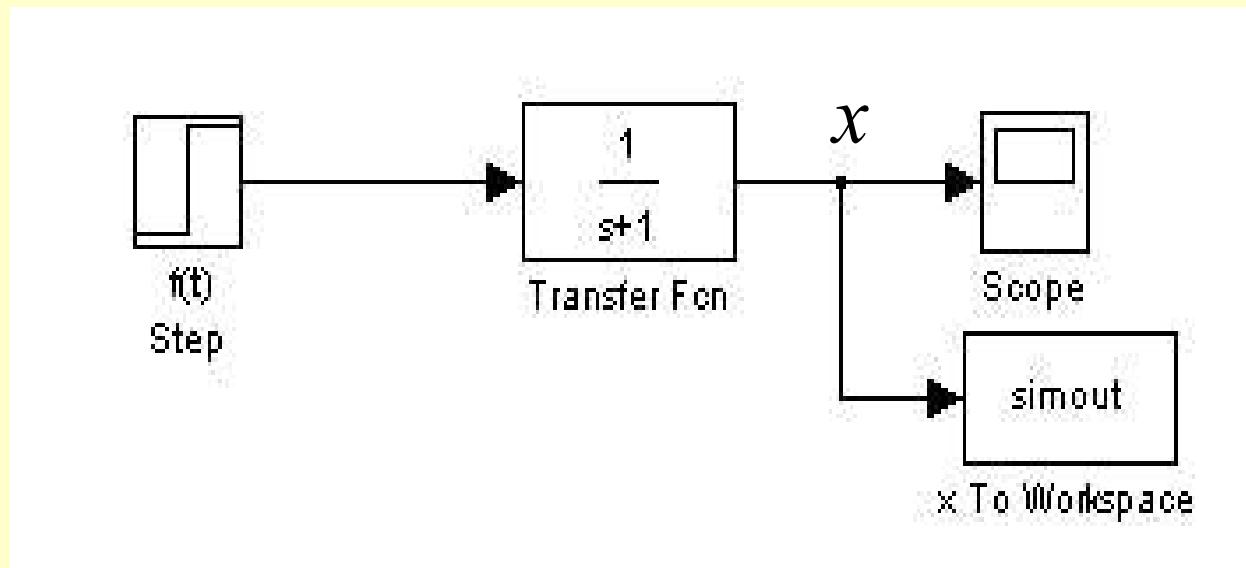
$$\frac{X(s)}{F(s)} = \frac{1}{2s^2 + 2s + 2}$$

STEP 1: Creating Blocks

Select BLOCK set	Location in Simulink Library
Step	Sources
Transfer Function	Continuous
Scope & To Workspace	Sinks

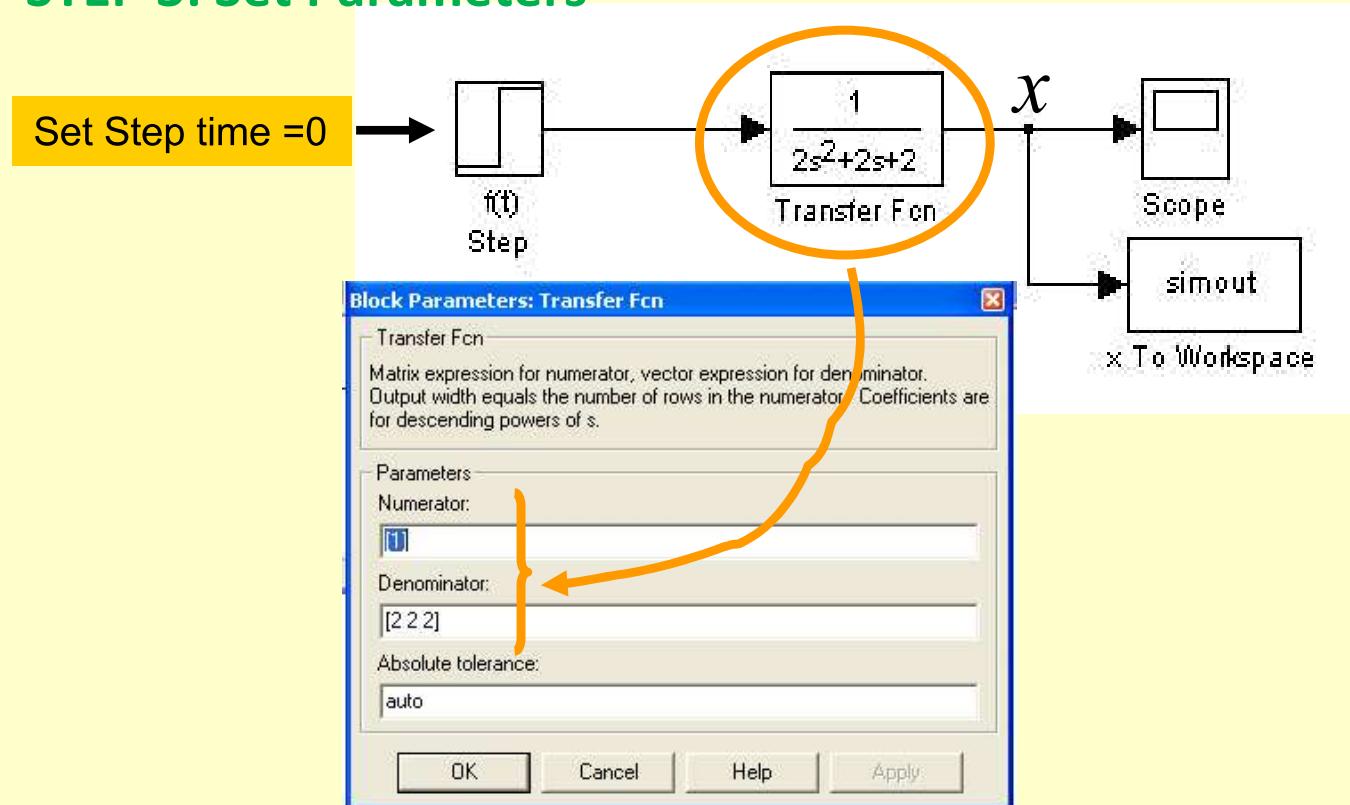
Example 2: Transfer Function

STEP 2: Making connections



Example 2: Transfer Function

STEP 3: Set Parameters

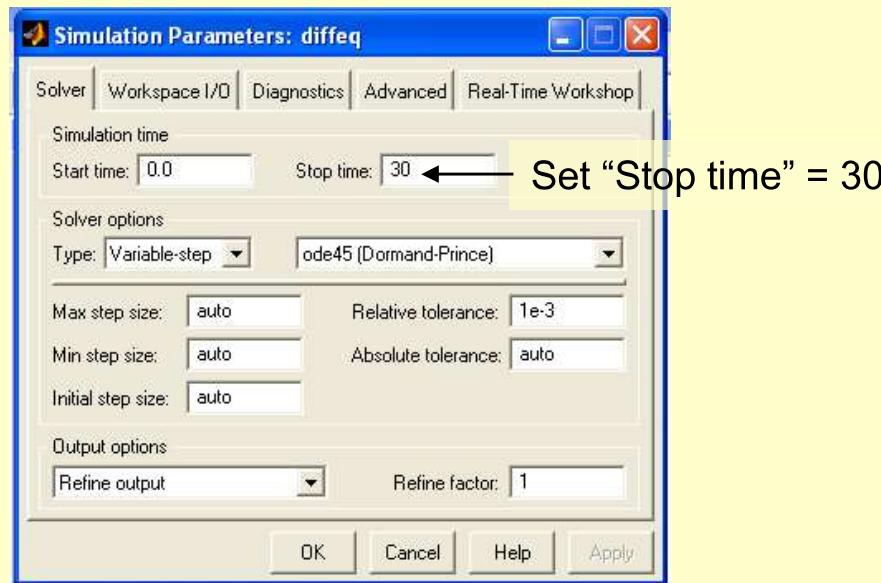


Example 2: Transfer Function

STEP 4: Running Simulation

1

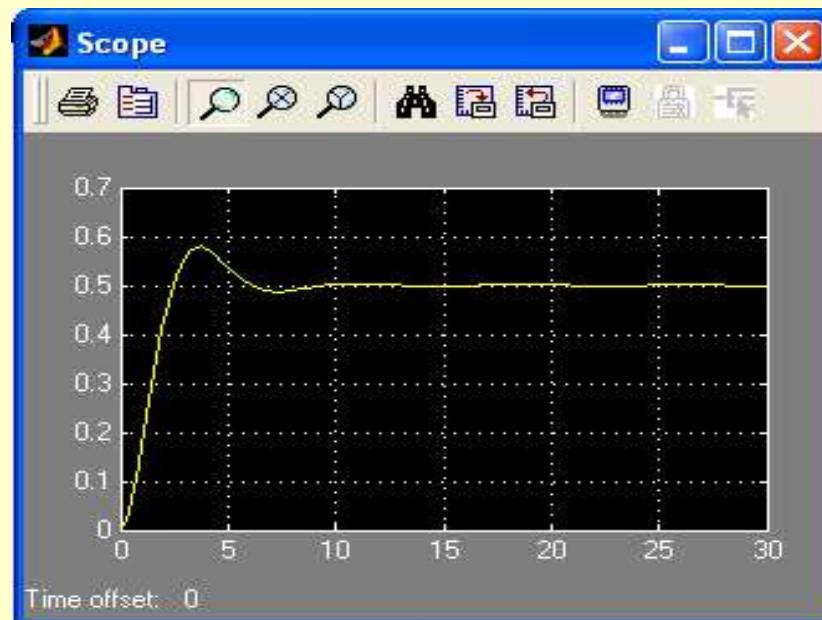
Open “simulation parameters” window



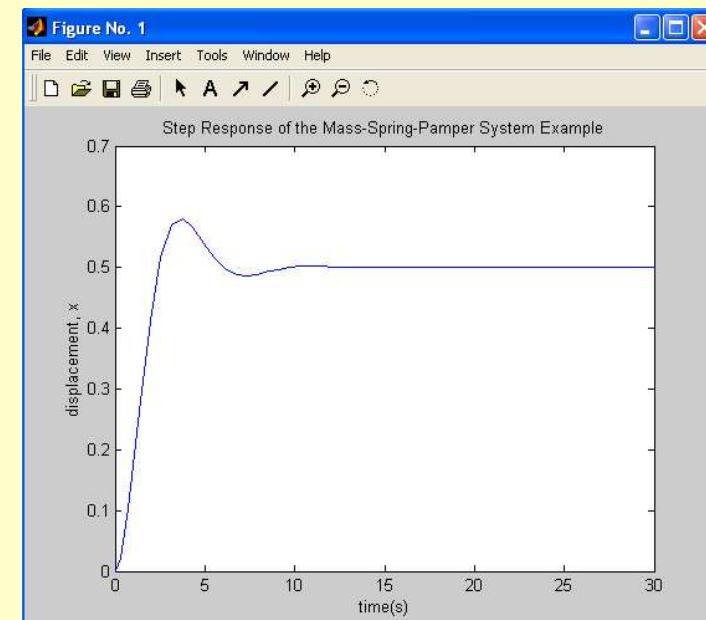
2

RUN
Simulation

Example 2: Transfer Function



Output from Scope block



Plot system response

Debugging

■ Tạo hàm:

```
function fact_of_n = factorialN(n)
fact_of_n = 0;
for i = 2:n
    fact_of_n = fact_of_n * i;
end
end
```

Gọi hàm để tính $6!$ → Kết quả ?

Tạo file test như sau:

The screenshot shows the MATLAB 7.10.0 (R2010a) interface. The title bar reads "MATLAB 7.10.0 (R2010a)". The menu bar includes File, Edit, Text, Go, Cell, Tools, Debug, Parallel, Desktop, and Window. The toolbar has various icons for file operations like Open, Save, and Print. The current folder is set to "C:\Users\Admin\Documents\MATLAB". Below the toolbar is a toolbar with icons for zoom, orientation, and other functions. The main area is the Editor, displaying the code "test_giaithua.m". The code contains the following lines:

```
1 - clear; clc;
2 - n=3
3 - giaithua= factorialN(3)
4 - n=4
5 - giaithua= factorialN(3)
6
```

Thêm vào các breakpoint dùng
Icon hay F12

This screenshot shows the same MATLAB Editor window as above, but with breakpoints added. The first four lines of the script now have black circular markers at the start, indicating they are breakpoints. The "Set/clear breakpoint" button in the toolbar is also highlighted.

```
1 - clear; clc;
2 ● n=3
3 ● giaithua= factorialN(3)
4 ● n=4
5 ● giaithua= factorialN(3)
6
```

F5 để RUN
F10 để chạy Step



Command Window

3

giaithua =

0

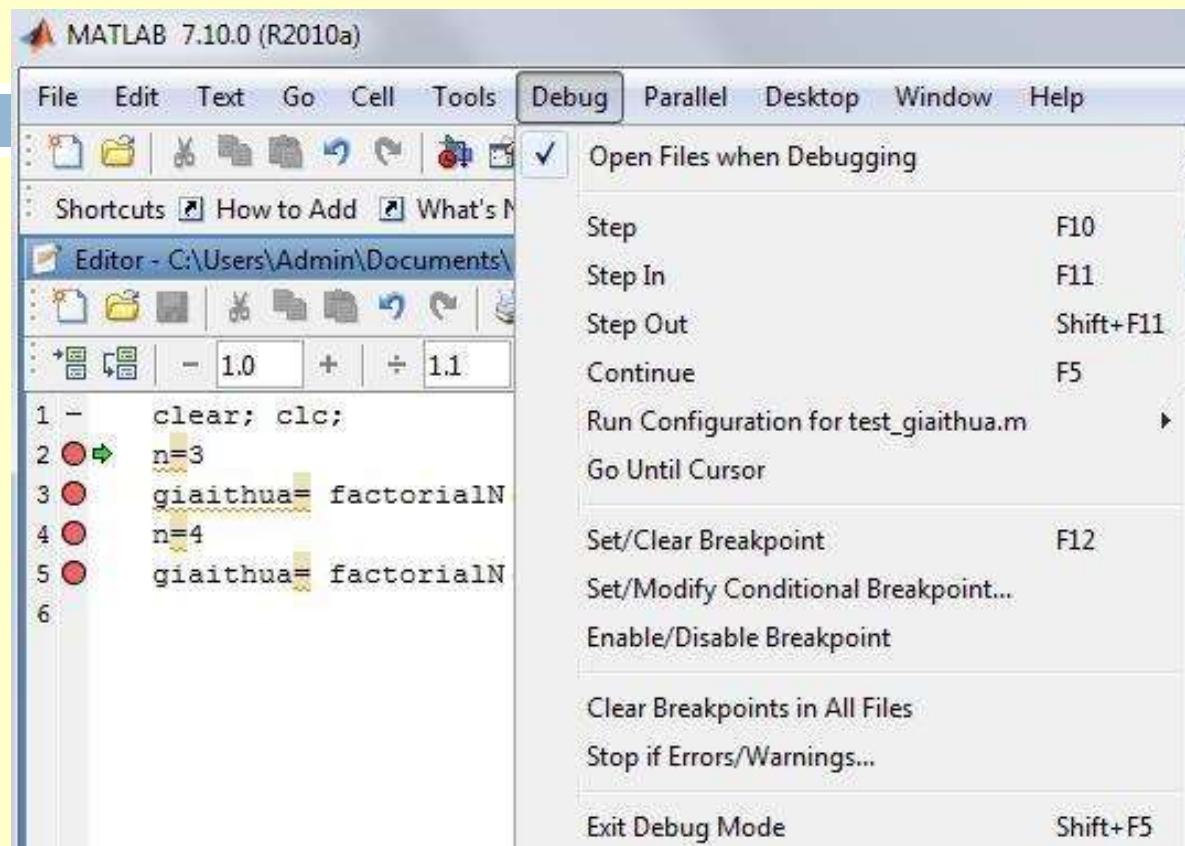
n =

4

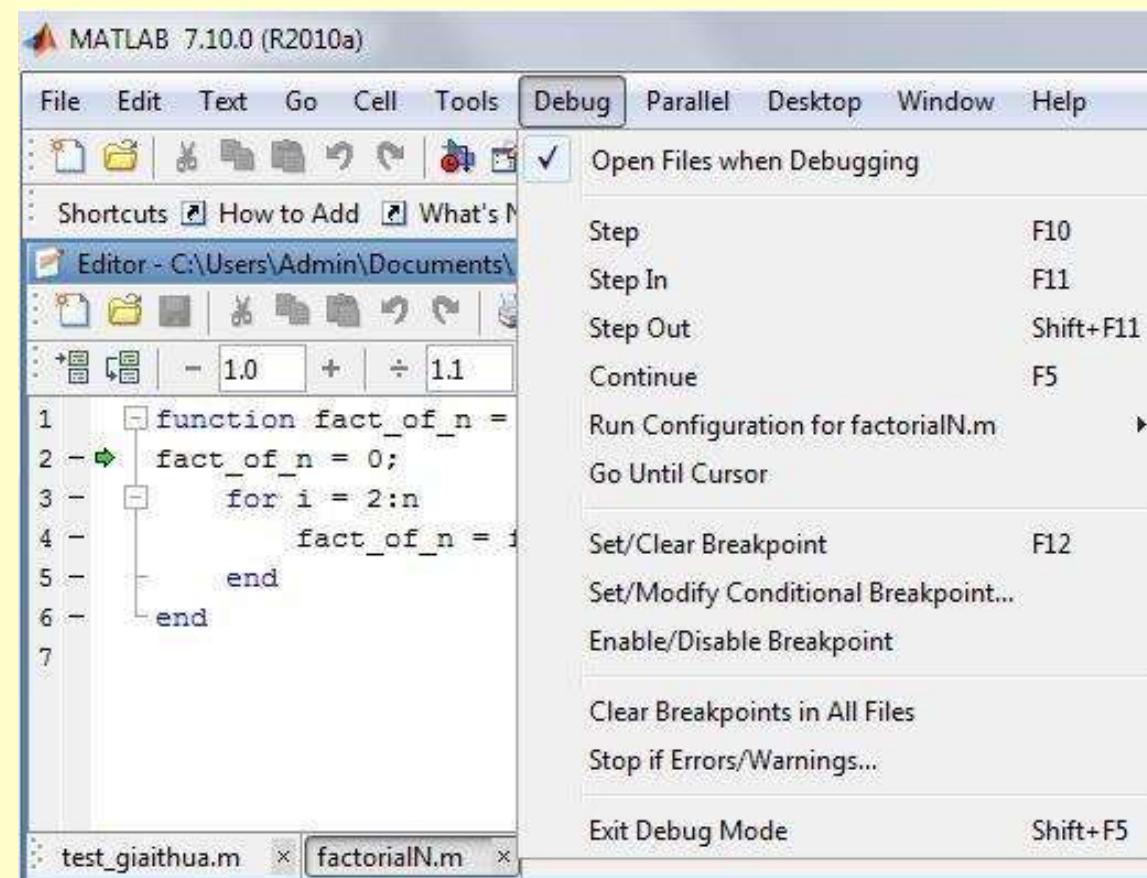
giaithua =

0

K>>



Để debug bên trong hàm (Step in) → F11



Syntax errors

```
x = input( 'x: ' );
if x < 0
    disp( 'negative' )
else if x == 0
    disp( 'zero' )
else
    disp( 'positive' )
end
```

```
>> [1 2 3] * [4 5 6]
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

```
y = sin()
```

```
>> y = sin ()
??? Error using ==> sin
Not enough input arguments.
```

```
disp( "Hi there" )
```

```
x = 0:pi/20:3*pi;  
y = sin(x);  
x = 0:pi/40:3*pi;  
plot(x,y)
```

??? Error using ==> plot
Vectors must be the same lengths.

Error in ==> debug2 at 4
plot(x,y)

```
>> x=x*(1+2*x));  
>> x=1; x=x(x+1)  
>> y='hello'  
>> z=1; disp(z);
```

```
clear;clc;close all  
x = -10:10;  
for k = 0:length(x)  
    if (x(k) > 0)  
        x(k) = -x(k);  
    end  
end  
disp(x)
```

Hãy cho biết các lỗi trong các chương trình

```
x = 0.1;  
while x ~= 0.2  
x = x + 0.001;  
fprintf('%g %g\n', x, x - 0.2 )  
end
```

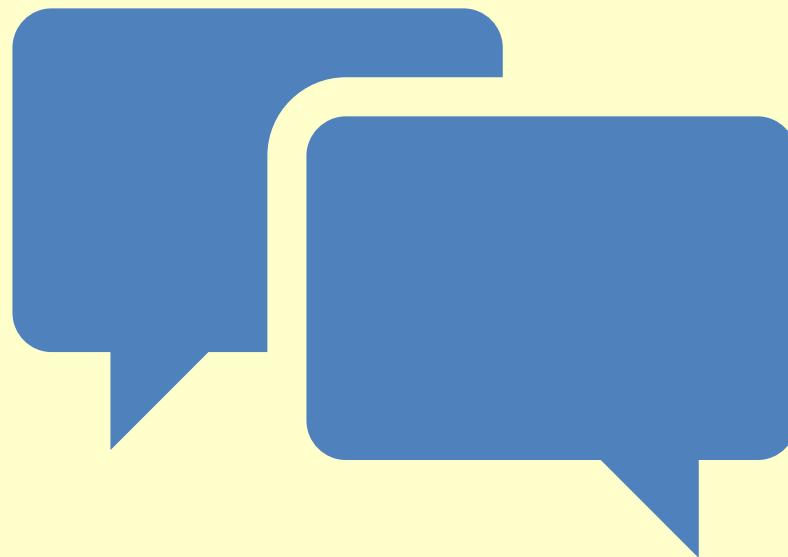
0.192 -0.008
0.193 -0.007
0.194 -0.006
0.195 -0.005
0.196 -0.004
0.197 -0.003
0.198 -0.002
0.199 -0.001
0.2 8.32667e-017

while x <= 0.2

while abs(x - 0.2) > 1e-6

```
if abs((a-b)/a) < 1e-6 disp('a bằng b...'),  
end
```

CÂU HỎI



FETEL-HCMUS

Thank you!

