

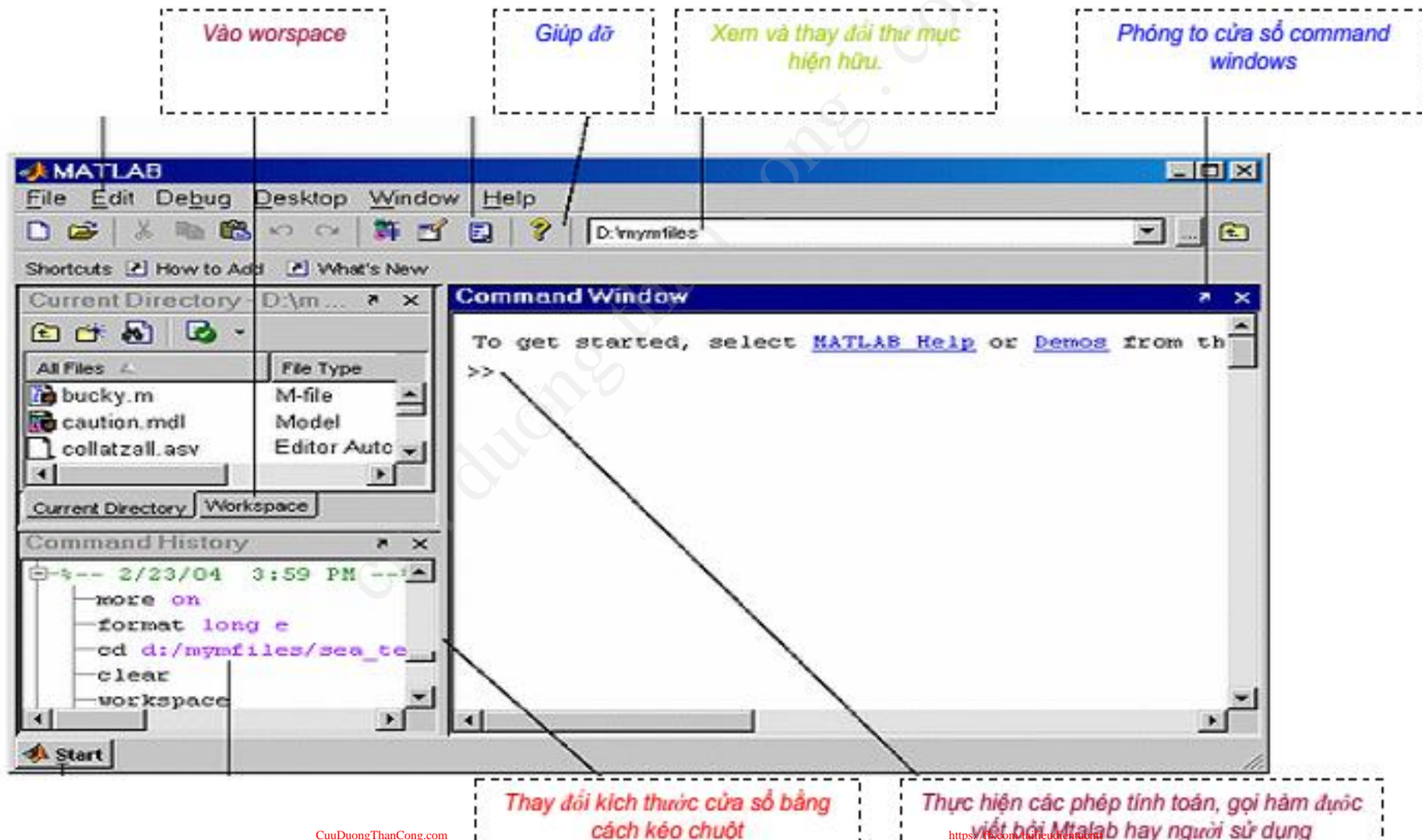
**NG**

i.

Email: [ongthanhhai@tdt.edu.vn](mailto:ongthanhhai@tdt.edu.vn)

# a Matlab

Khi chạy chương trình Matlab, giao diện đầu tiên xuất hiện với các thành phần cơ bản :



## a. Cửa sổ lệnh – command windows

Cửa sổ lệnh của Matlab cho phép người sử dụng thực hiện các phép tính toán, gọi các lệnh, hàm hoặc gọi các chương trình.

Trong quá trình sử dụng Matlab, ta có thể sử dụng một số hàm trợ giúp sau

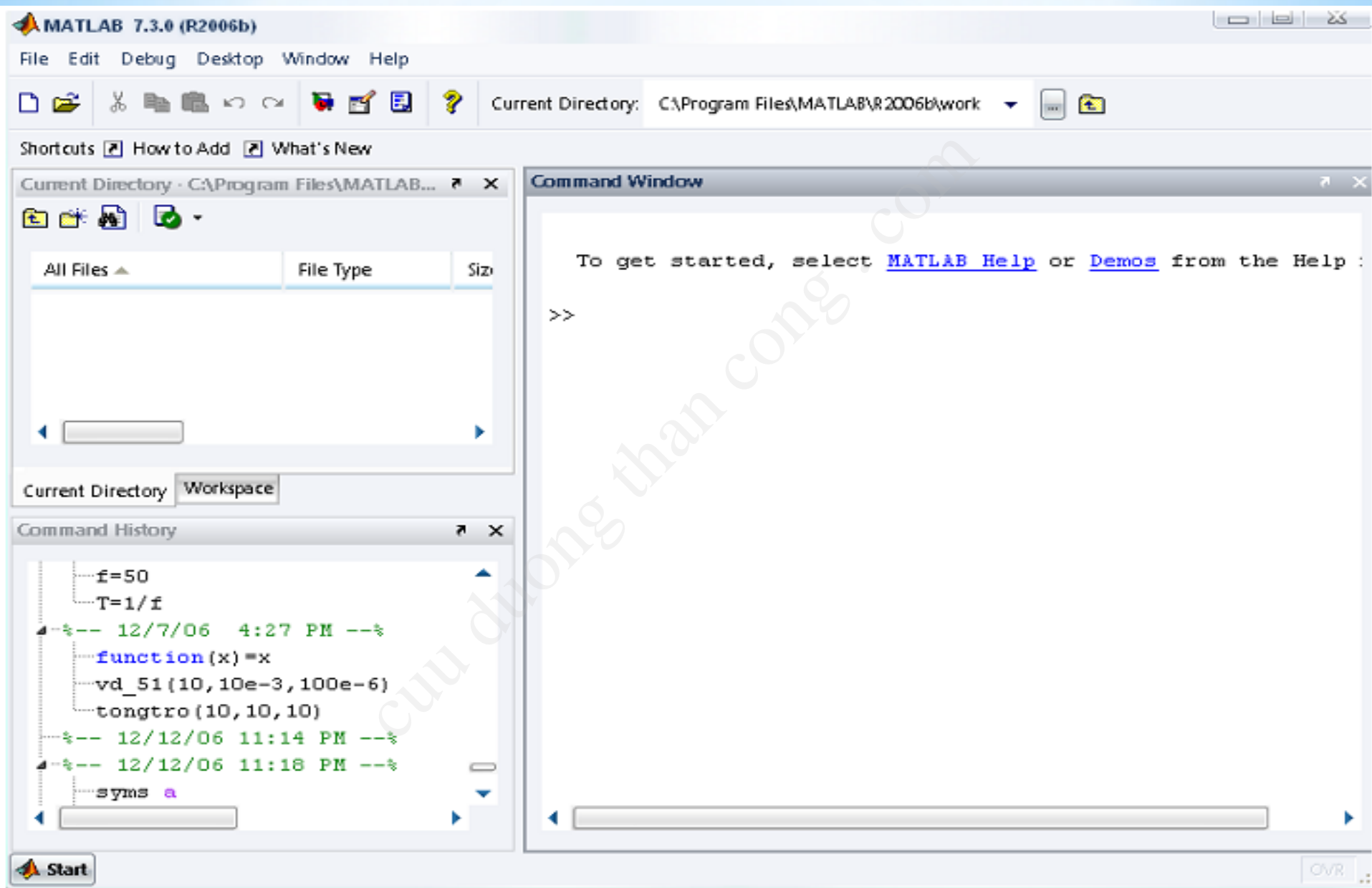
**help:** giúp đỡ, cho phép người dùng tra cứu bất cứ thông tin nào liên quan đến Matlab.

**demo:** các file, các chương trình đã được Matlab tạo sẵn, được tập hợp theo các chuyên đề .

**edit:** Mở chương trình soạn thảo hàm, chương trình.

**ver:** Xem thông tin về phiên bản của Matlab và các thành phần của nó .

**exit:** Thoát ra khỏi chương trình.





## **b. Cửa sổ không gian làm việc – work space**

Cửa sổ không gian làm việc liệt kê tất cả các biến hiện đang sử dụng trong chương trình

Muốn xóa tất cả các biến, chúng ta dùng lệnh `>> clear all`

## **c. Cửa sổ thư mục hiện tại – current directory:**

Hiển thị thư mục hiện tại mà chương trình Matlab đang dẫn đến. Thư mục mặc định là `C:\Programs\MATLAB\R2006\work`

Chương trình cho phép thiết lập đường dẫn đến thư mục bất kỳ trên máy tính.

## **d. Cửa sổ lịch sử lệnh – command history:**

Ghi nhớ các lệnh đã thực hiện trên cửa sổ lệnh, có thể copy và dán lại các mệnh lệnh đã thực thi được lưu lại vào ngược lại cửa sổ lệnh.

*Các phép toán cơ bản của Matlab được thực hiện trực tiếp trên cửa sổ lệnh command windows.*

*Các phép toán cơ bản gồm : các phép toán số học, các phép toán lượng giác, các phép toán làm tròn, các phép toán so sánh, các phép về số phức.*

## **2.1 Các phép toán số học**

Để tính toán với các phép tính số học đơn giản, tại ngay dấu nhắc >> của cửa sổ lệnh Command Windows, chúng ta gõ vào trực tiếp :

Stt	Tên hàm	Ý nghĩa	Ví dụ	Kết quả
1	+	Cộng	2+5	7
2	-	Trừ	1000-25	975
3	*	Nhân	10*10	100
4	/	Chia	100/5	20
5	^	Lũy thừa ( $a^b$ )	10^3	1000
6	sqrt(x)	Căn bậc 2	sqrt(144)	12
7	exp(x)	Hàm mũ ( $e^x$ )	exp(1)	2.7183
8	log(x)	Logarit tự nhiên (ln (x) )	log(exp(1))	1
9	log10(x)	Logarit thập phân ( $\lg_{10}(x)$ )	log10(100)	2

## 2.2 Các phép toán lượng giác :

Khi sử dụng các phép toán lượng giác, chúng ta chú ý là matlab hiểu các đối số của các hàm lượng giác là radian. Cũng như kết quả trả về của các hàm lượng giác ngược cũng là radian.

Stt	Tên hàm	Ý nghĩa	Ví dụ	Kết quả
1	sin	Sin	$\sin(30 \cdot \pi / 180)$	0.5000
2	cos	Cos	$\cos(0.5)$	0.8776
3	tan	Tang	$\tan(10 \cdot \pi / 180)$	0.1763
4	cot	Cotang	$\cot(45 \cdot \pi / 180)$	1
5	asin	arcsin	$\text{asin}(0.5) \cdot 180 / \pi$	30
6	acos	Arccos	$\text{acos}(0.86) \cdot 180 / \pi$	30
7	atan	Arctang	$\text{atan}(1) \cdot 180 / \pi$	45
8	acot	arccotang	$\text{acot}(1) \cdot 180 / \pi$	45

## 2.3 Các phép toán làm tròn và lấy phần dư :

Stt	Tên hàm	Ý nghĩa	Ví dụ	Kết quả
1	fix	làm tròn các thành phần thập phân về 0	fix(1.5680)	1
2	floor	làm tròn về số nguyên gần nhất nhỏ hơn	floor(1.5680)	1
3	ceil	làm tròn về số nguyên gần nhất lớn hơn	ceil(1.5680)	2
4	round	Làm tròn về số nguyên gần nhất	round(1.5680)	2
5	Mod(x,y)	Tính phần dư phép chia, lấy theo y	mod(13,5)	3
6	Rem(x,y)	Tính phần dư phép chia, lấy theo x	rem(13,2)	1
7	Sign(x)	Lấy dấu của x	sign(-2)	-1



## 2.4 Các phép toán so sánh :

Các phép toán so sánh sẽ so sánh giá trị của giá trị bên phải và bên trái của hàm so sánh, tùy theo từng trường hợp cụ thể mà giá trị trả về có thể là 1 hay 0

Stt	Tên hàm	Ý nghĩa	Ví dụ	Kết quả
1	>	So sánh lớn hơn	$1 > 2$	0
2	<	So sánh nhỏ hơn	$1 < 2$	1
3	==	So sánh bằng	$1 == 2$	0
4	~=	So sánh không bằng	$1 ~= 2$	1
5	>=	So sánh lớn hơn hay bằng	$1 >= 2$	0
6	<=	So sánh nhỏ hơn hay bằng	$1 <= 2$	1

## 2.5 Các phép toán logic

Các phép toán so sánh sẽ so sánh giá trị của giá trị bên phải và bên trái của hàm so sánh, tùy theo từng trường hợp cụ thể mà giá trị trả về có thể là 1 hay 0

Stt	Tên hàm	Ý nghĩa	Ví dụ	Kết quả
1	&	Phép giao	$(1>3)\&(2>4)$	0
2	and	Phép giao	$\text{and}(1>3,2>4)$	0
3		Phép hợp	$(1>3) (2>1)$	1
4	or	Phép hợp	$\text{or}(1>3,2>1)$	1
5	~	Phép phủ định	$\sim(1>2)$	1
6	not	Phép phủ định	$\text{not}(1>2)$	1
7	xor	Phép Xor	$\text{xor}(1<3,2<5)$	0

## 2.7 Phép toán gán

Toán tử gán được sử dụng trong Matlab dùng để thay thế cho giá trị hoặc một biểu thức tính toán.

Toán tử = trong Matlab được gọi là toán tử gán

Ví dụ: cần tính biểu thức

$$A = \frac{51^6 + 76^7}{\ln(1076)e^{15}}$$

Khi đó, chúng ta sẽ thay giá trị tử số bằng biến B, giá trị mẫu số bằng biến C, và giá trị A cần tính bằng B/C.

Cú pháp của toán tử gán :

**Tên biến = giá trị hay biểu thức tính toán**

Tên biến : Tối đa 31 ký tự, có phân biệt chữ hoa và chữ thường, có thể sử dụng các chữ số trong tên biến nhưng ký tự đầu tiên của tên biến phải là chữ.

Thực hiện phép gán giải quyết ví dụ trên như sau

Thực hiện phép gán giải quyết ví dụ trên như sau

$$\begin{aligned}B &= 51^6 + 76^7 \\ C &= \log(107) * \exp(15) \\ A &= B/C\end{aligned}$$

Ví dụ:

- Phép gán:  $x + 2 = 20$  là sai vì bên trái toán tử gán không phải là 1 tên biến.
- Phép gán:  $x = 5 + y$  chỉ đúng nếu giá trị của biến  $y$  đã được xác định từ trước, nếu không Matlab sẽ báo sai.
- $>> x = 5;$

$>> x = x + 3$

```
>> b=3/26;
```

```
>> format long; b
```

```
b =
```

```
0.11538461538462
```

```
>> format short e; b
```

```
b =
```

```
1.1538e-001
```

```
>> format bank; b
```

```
b =
```

```
0.12
```

```
>> format short eng; b
```

```
b =
```

```
115.3846e-003
```

```
>> format hex; b
```

```
b =
```

```
3fbd89d89d89d89e
```

format (định dạng)

```
>> format +; b
```

```
b =
```

```
+
```

```
>> format rat; b
```

```
b =
```

```
3/26
```

```
>> format short; b
```

```
b =
```

```
0.1154
```

```
>> format long eng; b
```

```
b =
```

```
115.384615384615e-003
```



ng

- **clc**

- **clear:**

a workspace

- **clear var1 var2**

■

- **exist('name')**

name

chưa

- **quit**

nh Matlab

- **who**

▪

- **whos**: t kê c n n nh va ch c  
a ng trong bộ nhớ va ro n o a  
ng u .

- **;** **u** **m** **y** i ng nh ngăn không  
cho Matlab n thi c t ra a số nh.

- **...** **u** **ba** **m** liên c, khi ng nh i  
n ng ng, u ba m i ng  
nh o cho Matlab t n p c ng  
p theo.

- **y**

ng

- **m**

u nhau.

- **n trăm**

nh

### 3. Sử dụng file .M và lập trình

Ở chương hai chúng ta đã sử dụng cửa sổ lệnh để thực hiện các phép tính toán bằng cách tính toán trực tiếp trên cửa sổ lệnh, nhưng để giải quyết các vấn đề phức tạp thì ta phải sử dụng các file được lập trình. **Các file được lập trình của Matlab được gọi là các M - file ( các file này có đuôi là : .m),** do đó trong toàn bộ giáo trình các file được lập trình sẽ gọi là M-file.



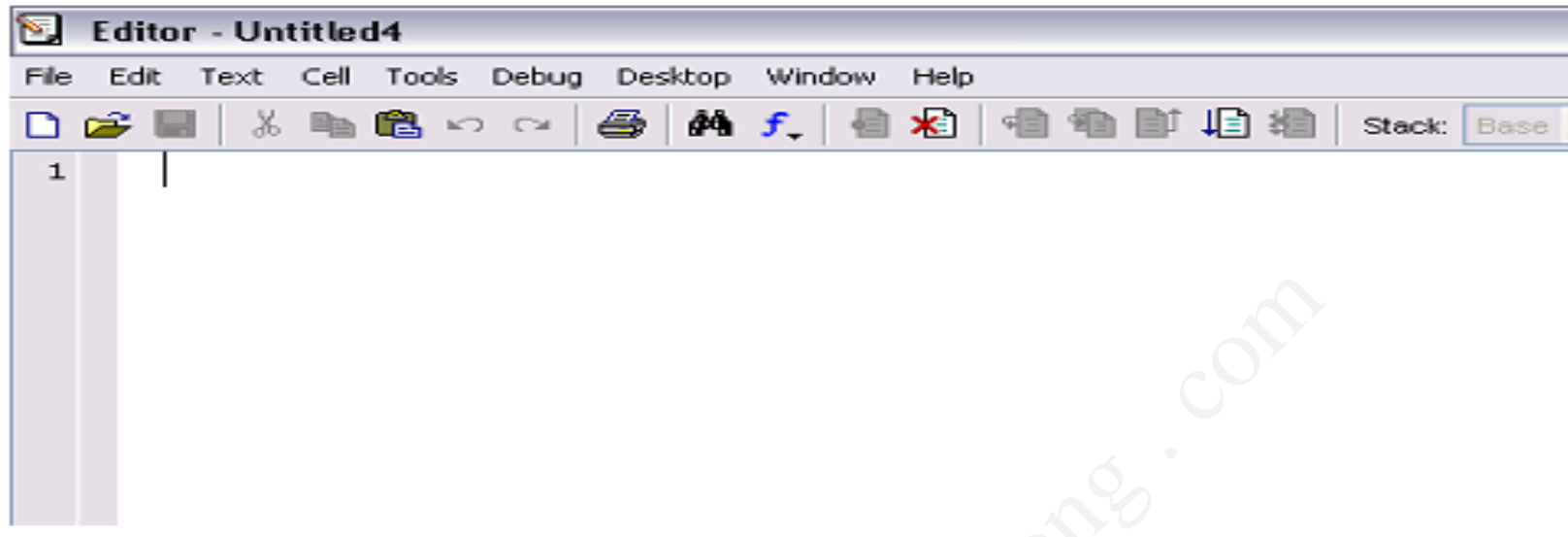
## 3.1 Cách tạo M-file

**Có 2 cách để khởi động một chương trình biên soạn M-file:**

**Cách 1 :** từ cửa sổ lệnh (command windows), gõ edit.

**Cách 2 :** Vào menu file, chọn New .

Khi đó, chương trình sẽ hiển thị một cửa sổ trắng để chúng ta soạn thảo



Sau khi soạn thảo xong M-file, chúng ta nhấn : **Ctrl + S hoặc File/Save để lưu file chương trình**. Khi đặt tên file chương trình phải đúng theo quy định của Matlab. Cụ thể, tên file phải được bắt đầu bằng chữ, sau đó có thể sử dụng số, và được dùng dấu gạch ngang dưới để phân biệt, ví dụ tên file : baitap\_21.m

Để chạy chương trình, có thể sử dụng một trong 2 cách sau :

**Cách 1 :** Trong môi trường soạn thảo M-file, chúng ta vào menu **debug/ Run** hoặc nhấn phím tắt F5.

**Cách 2 :** Trong cửa sổ lệnh command window, chúng ta **nhập vào đúng tên M-file đã được lưu**, sau đó nhấn Enter .

Nếu chương trình **được lập trình đúng**, sau khi chạy chương trình, người sử dụng **chuyển ra cửa sổ lệnh để xem kết quả**.

Còn ngược lại, chương trình sẽ báo lỗi, Matlab phát ra **1 tiếng bip báo hiệu**, đồng thời chương trình sẽ tự **chuyển sang cửa sổ lệnh, thông báo cho người lập trình vị trí bị lỗi**.

## 3.2 Các hàm nhập và hàm xuất dữ liệu ra màn hình

### a. hàm nhập dữ liệu

Cú pháp :

```
x = input ( 'prompt' )
```

Trong đó :

input : Từ khoá của hàm nhập dữ liệu

x : tên biến được gán giá trị nhập vào.

promp : dòng text mà người sử dụng đánh vào.

**Diễn đạt :** biến x sẽ có giá trị bằng giá trị mà người sử dụng nhập vào



## **b. hàm xuất dữ liệu ra màn hình**

### **Cú pháp :**

`disp ( x )`  
`disp( 'text' )`

Trong đó :

`disp` : Từ khoá của hàm xuất dữ liệu

`x` : tên biến hoặc các giá trị số cần xuất ra màn hình

`text` : dòng text mà người sử dụng cần xuất ra màn hình

**Ghi chú : Cú pháp : `clear` hoặc `clear all` sẽ xóa toàn bộ các biến đang được sử dụng trong matlab, giúp chương trình chạy đúng.**

## Ví dụ:

```
%chuong trinh tinh dthcn  
disp('tinh dien tich hinh chu nhac');  
a=input('nhap a=');  
b=input('nhap b=');  
S=a*b  
disp('dien tich hinh chu nhac:');  
disp(S)
```

```
%chuong trinh giai phuong trinh bac hai  
disp('CHUONG TRINH GIAI PHUONG TRINH BAC HAI')  
disp('nhap vao cac he so:')  
a=input('nhap he so a = ');  
b=input('nhap he so b = ');  
c=input('nhap he so c = ');  
delta = b^2-4*a*c;  
disp('cac nghiem so : ');  
x1=(-b+sqrt(delta))/(2*a)  
x2=(-b-sqrt(delta))/(2*a)
```

### 3.3 Các hàm con .

#### a. Hàm điều kiện:

Dạng đơn giản

if- elseif - else – end

hàm if – end (if- else – end)

Cú pháp :

```
if expression
    statements
end
```

Trong đó :

if, end : Từ khoá của hàm

expression : Điều kiện – biểu thức logic

statements : Lệnh hoặc nhóm lệnh cần thực thi

**Diễn đạt :**

Nếu thỏa điều kiện - biểu thức logic thì sẽ thực hiện lệnh.

```
a=input('nhap a = ');  
b=input('nhap b = ');  
if a<b  
    disp('a nho hon b');  
end  
if a==b  
    disp('a bang b');  
end  
if a>b  
    disp('a lon hon b');  
end
```

## Viết chương trình giải phương trình bậc nhất $ax + b = 0$

% Giải phương trình bậc nhất

a=input('nhap a=');

b=input('nhap b=');

if a~=0

x=-b/a;

fprintf('nghiem so x=%9.5g\n',x);

else

if b~=0

disp('ptvn')

else

disp('ptvsn')

end

end

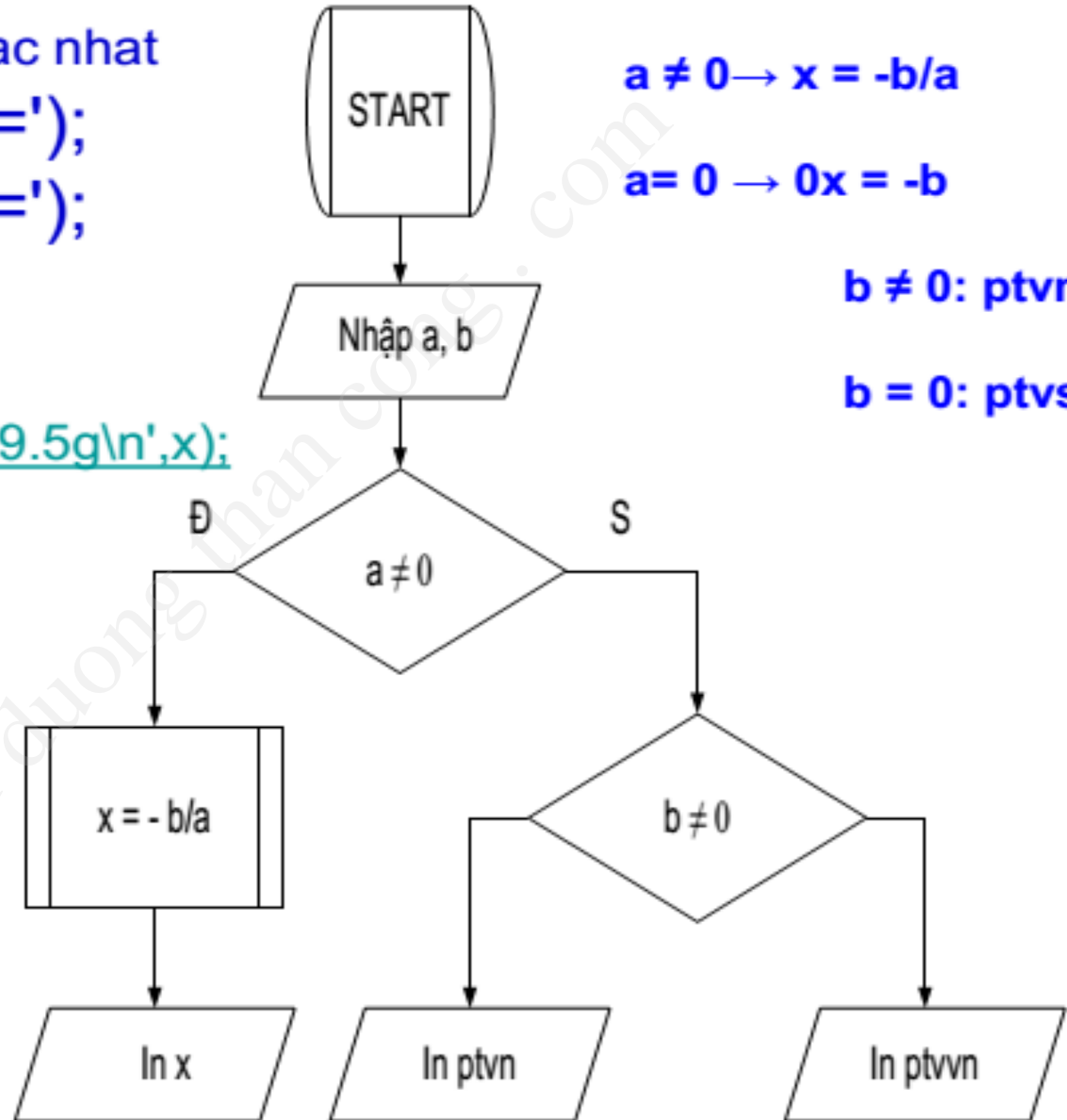
$$ax + b = 0 \rightarrow ax = -b$$

$$a \neq 0 \rightarrow x = -b/a$$

$$a = 0 \rightarrow 0x = -b$$

$b \neq 0$ : ptn

$b = 0$ : ptn



```
%chuong trinh giai phuong trinh bac hai
disp('giai phuong trinh bac hai')
a=input('nhap a=');
b=input('nhap b=');
c=input('nhap c=');
delta=b^2-4*a*c;
d=delta;
if d<0
disp('ptvn')
else
if d==0
x=-b/(2*a);
fprintf('nghiem so x=%9.5g\n',x);
else
x1=(-b-sqrt(d))/(2*a);
x2=(-b+sqrt(d))/(2*a);
fprintf('nghiem so x1=%9.5g\n',x1);
fprintf('nghiem so x2=%9.5g\n',x2);
end
end
end
```

## Dạng đầy đủ

if- elseif - else – end

## Cú pháp :

```
if expression1
    statements1
elseif expression2
    statements2
else
    statements3
end
```

**if, elseif, else, end** : Từ khoá của hàm

expression1, 2, 3 : Điều kiện – biểu thức logic 1, 2, 3

statements 1, 2, 3 : Lệnh hoặc nhóm lệnh thực thi 1, 2, 3

## Diễn đạt :

Nếu thỏa điều kiện hoặc biểu thức logic 1 thì sẽ thực hiện lệnh 1.

Nếu không thỏa điều kiện 1, và thỏa điều kiện 2 thì sẽ thực hiện lệnh 2.

Nếu không thỏa điều kiện 1 và điều kiện 2 thì sẽ thực hiện lệnh 3.

**Chú ý :** các điều kiện 1, 2 không được trùng lặp nhau .



Ví dụ: Bài toán phân loại học sinh: điểm 9-10 xếp loại giỏi, điểm 7-8 xếp loại khá, điểm 5-6 xếp loại trung bình; điểm 1,2,3,4 xếp loại yếu; nếu điểm vào không phải số nguyên nằm giữa 1 và 10 thì thông báo điểm không hợp lệ.

```
diem = input('nhap diem = :')
if (diem == 9)|(diem == 10)
disp ('loai gioi ')
elseif (diem == 7)|(diem == 8)
disp ('loai kha ')
elseif (diem == 5)|(diem == 6)
disp ('loai trung binh ')
elseif (diem>=1)&(diem<=4)
disp ('loai yeu ')
else
disp ('diem vao khong hop le ')
end
```

## Qui tắc xử lý ưu tiên:

Trong một biểu thức Matlab vừa có toán tử số học, vừa có toán tử quan hệ và toán tử logic, thì thứ tự xử lý trong Matlab như sau:

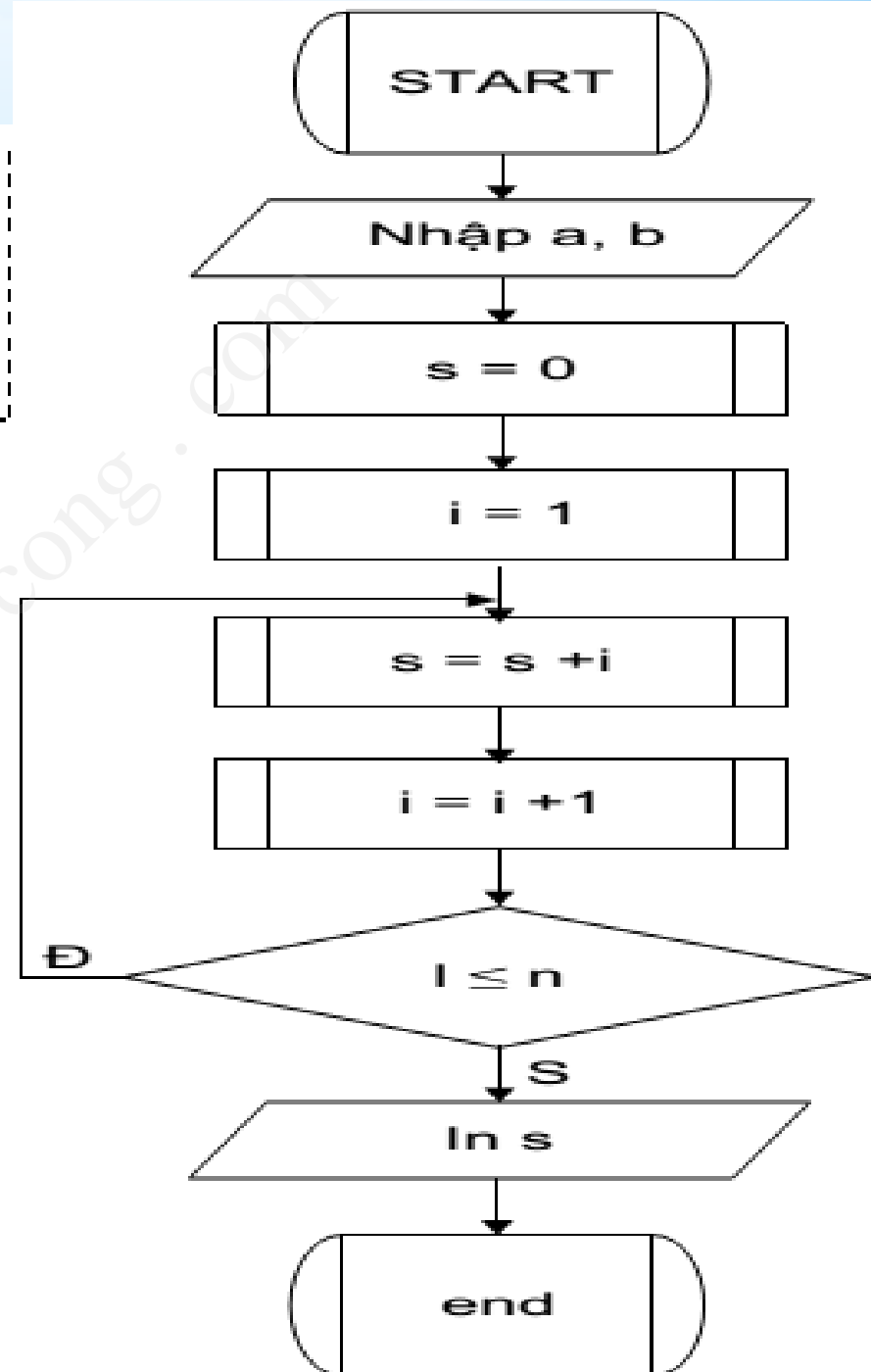
1. Các cặp dấu ngoặc, được tính từ cặp trong cùng nhất.
2. Các toán tử số học và toán tử NOT (~) được tính từ trái qua phải.
3. Các toán tử quan hệ được tính từ trái qua phải.
4. Toán tử AND
5. Toán tử OR

# Ví dụ cho vòng lặp for

$$S = 1 + 2 + 3 + \dots + n$$

$$S = 0 + 1 + 2 + 3 + \dots + n$$

```
clc
clear
disp('tinh tong');
n=input('nhap so hang can tinh tong n=');
s=0; % gia tri ban dau cua tong s
for i=1:n
    s=s+i;
end
fprintf('tong so s=%2.5g\n',s);
```



#### d. Hàm vòng lặp while

Được sử dụng khi số lần lặp không được biết trước. Quá trình lặp sẽ chấm dứt khi một điều kiện xác định nào đó được thỏa. Cấu trúc của vòng lặp while như sau:

**Cú pháp :**

```
while expression
statements
end
```

Trong đó :

expression : Điều kiện của vòng lặp

statements : Lệnh hoặc nhóm lệnh thực thi của vòng lặp.

**Diễn đạt :**

Nếu thỏa điều kiện thì sẽ thực hiện lệnh của vòng lặp.

Hàm while được sử dụng khi chưa biết số lần lặp, trong khi hàm for được sử dụng khi đã biết rõ số lần lặp.

Vd: tính  $n!$

$n! = 1.2.3.4.5.....n$

$GT = 1.1.2.3.4.5.....n$

```
clc
clear
disp('tinh giai thua');
n=input('nhap so hang can tinh giai thua n=');
GT=1;
i=1;
while i<=n
    GT=GT*i;
    i=i+1;
end
fprintf('tong so s=%2.5g\n',GT);
```

## e. Hàm vòng lặp switch-case

Cấu trúc switch-case cho phép chương trình có nhiều lựa chọn và thực hiện chỉ một trong những nhánh này, tùy thuộc vào giá trị của biểu thức đầu vào. Cấu trúc switch-case có dạng như sau:

### Cú pháp :

switch	biểu thức đầu vào (vô hướng hoặc chuỗi kí tự)
case	giá trị 1
	nhóm lệnh 1
case	giá trị 2
	nhóm lệnh 2
...	
case	giá trị n
	nhóm lệnh n
otherwise	
	nhóm lệnh n+1
end	

Ví dụ: trở lại bài toán phân loại học sinh, chương trình được viết với cấu trúc switch-case như sau:

```
n=input('cho biet diem : ');  
disp('phan loai:');  
switch n  
case {0,1,2,3,4}  
disp('loai yeu');  
case {5,6}  
disp('loai trung binh');  
case {7,8}  
disp('loai kha');  
case {9,10}  
disp('loai gioi');  
otherwise  
disp('khong hop le');  
end
```



### 3.4 Hàm trong Matlab (function)

- **Hàm** trong Matlab là một file.m có thể nhận tham số và trả về các giá trị.
- Tên hàm phải trùng với tên file.m. Gọi lệnh bằng cách gõ tên hàm (tên file.m)
- **Cú pháp:**  
`function tri_tra_ve = ten_ham(tham_so)`  
...  
■ **Ví dụ: % tbc.m**  
`function s = tbc(x)`  
`s = sum(x(:))/length(x(:));`
- **Gọi lệnh:** `>>t = tbc([2 3 4 5 6])`     *% t sẽ bằng 4*

## ■ **Script: % *UBC.m***

```
function [m,n] = UBC(a,b)
```

```
m = gcd(a,b);
```

```
n = lcm(a,b);
```

## ■ **Gọi hàm:**

```
>> [uoc, boi] = UBC(45,234)
```

```
>> uoc
```

```
ans = 9
```

```
>> boi
```

```
ans = 1170
```

## ■ Hàm phụ

- (khác tên hàm chính) chỉ nhằm hỗ trợ tính toán cho hàm chính trong script hàm
- Hàm phụ nằm sau hàm chính

## ■ Ví dụ: % *chinh.m*

```
function c = chinh(A)
```

```
c = phu(A) - 1
```

```
function d = phu(B)
```

```
d = min(B(:))
```

## ■ Gọi hàm:

```
>> chinh([23 2 34])
```

```
ans = 1
```

# n trong Matlab

- *Ma trận* là một mảng hình chữ nhật các con số
- Ma trận gồm các *dòng (row)* và các *cột (column)*. Các dòng hay cột gọi chung là *vector*.

$$\begin{pmatrix} 16 & 3 & 2 & 13 \\ 5 & 10 & 11 & 8 \\ 9 & 6 & 7 & 12 \\ 4 & 15 & 14 & 1 \end{pmatrix}$$

- Một con số trong Matlab là một ma trận  $1 \times 1$
- Thế mạnh của Matlab so với các ngôn ngữ lập trình khác là tính toán rất nhanh trên ma trận.

- Nhập trực tiếp danh sách các phần tử
- Phát sinh ma trận bằng các hàm sẵn có

- Nhập từ file
- Tạo ma trận bằng các file .m

$A = [16 \ 3 \ 2 \ 13; 5 \ 10 \ 11 \ 8; 9 \ 6 \ 7 \ 12; 4 \ 15 \ 14 \ 1]$

$A =$

16 3 2 13

5 10 11 8

9 6 7 12

4 15 14 1

- Dấu [ và ] mở đầu và kết thúc nhập ma trận.
- Dấu ; kết thúc một dòng.
- Các phần tử cách nhau bằng khoảng trắng hoặc dấu ,



⊙ A =

⊙ 16 3 2 13

⊙ 5 10 11 8

⊙ 9 6 7 12

⊙ 4 15 14 1

⊙ >> sum(A)

⊙ ans =

34 34 34 34

⊙ >> A'

⊙ ans =

16 5 9 4

3 10 6 15

2 11 7 14

13 8 12 1

⊙ >> sum(A')

⊙ ans =

34 34 34 34

n

⊙ A =

⊙ 16 3 2 13

⊙ 5 10 11 8

⊙ 9 6 7 12

⊙ 4 15 14 1

⊙ >> diag(A)

⊙ ans =

16

10

7

1



- Phân tử  $A_{ij}$  được trích bằng biểu thức  $A(i,j)$

- $A =$

16 3 2 13

5 10 11 8

9 6 7 12

4 15 14 1

- $A(4,2)$  là phân tử ở dòng 4 cột 2, tức là phân tử 15.

- Phép trích chỉ có một chỉ số sẽ theo thứ tự duyệt theo cột. (xem ma trận là một vector cột dài)

- $A(8)$  là phân tử thứ 8 duyệt theo cột từ trái qua phải, từ trên xuống dưới.

- `>> t = A(4,5)`
- *“Index exceeds matrix dimensions.”*
- Việc truy xuất phần tử vi phạm kích thước ma trận
- Nằm bên phải phép gán

- `>> X = A;`
- `>> X(4,5) = 17`
- `X =`

16	3	2	13	0
5	10	11	8	0
9	6	7	12	0
4	15	14	1	17
- Mở rộng ma trận
- Nằm bên trái phép gán

- ◎ Dấu hai chấm “:” là một trong những phép toán quan trọng nhất trong Matlab.
- ◎ Ví dụ:
  - 1:10 là một vector dòng gồm các số nguyên từ 1 đến 10
  - 1 2 3 4 5 6 7 8 9 10
- ◎ Để tạo bước tăng/giảm khác 1
  - 100:-7:50  
100 93 86 79 72 65 58 51
  - 0:pi/4:pi  
0 0.7854 1.5708 2.3562 3.1416

## ◎ Ví dụ:

- ◎  $A(1:k,j)$  gồm  $k$  số đầu tiên ở cột thứ  $j$  của ma trận  $A$ .
- ◎  $\text{sum}(A(1:4,4))$  tính tổng 4 số đầu tiên ở cột thứ 4 của ma trận  $A$ .
- ◎ Dấu hai chấm đứng một mình sẽ chỉ toàn bộ phần tử của dòng hoặc cột.
- ◎ Từ khoá “end” chỉ chỉ số cuối cùng của dòng hoặc cột.

## ◎ Ví dụ:

- ◎  $A(:,\text{end})$  chỉ toàn bộ phần tử ở cột cuối cùng.

- Sử dụng dấu “[,]” để liệt kê vị trí cần trích
- Ví dụ:
  - $A = [2\ 4\ 3; 8\ 6\ 7], x = [9\ 4\ 2\ 1]$
  - $A([2,1],2) \rightarrow [6\ 4]', x([2,4]) \rightarrow 4\ 1$
- Có thể sử dụng dấu “:” để trích dãy các phần tử
- Ví dụ:
  - $A(2,1:3) \rightarrow 8\ 6\ 7, x(3:-1:1) \rightarrow 2\ 4\ 9$



# nh load

- File .m là một file văn bản ghi các dòng lệnh Matlab.
- Có thể soạn thảo bằng MATLAB Editor hoặc bất kỳ trình soạn thảo văn bản nào.
- Lưu file có đuôi .m
- Gõ tên file để thực thi nội dung các dòng lệnh trong file.

## Ví dụ:

- tạo một file có nội dung như sau:

```
A=[ 16.0 3.0 2.0 13.0  
    5.0 10.0 11.0 8.0  
    9.0 6.0 7.0 12.0  
    4.0 15.0 14.0 1.0];
```

- Lưu với tên magik.m. Dòng lệnh

```
>>magik
```

sẽ đọc file và tạo biến A là ma trận như trên.

n

A	1 2 3 4 5 6	Thêm cột	Thêm dòng
B	10 12 11 13	$D = [A \ B]$	$E = [A; C]$
C	7 8 9 9 7 8 8 9 7	1 2 3 10 12 4 5 6 11 13	1 2 3 4 5 6 7 8 9 9 7 8 8 9 7



⊙  $A =$

```
16 3 2 13
5 10 11 8
9 6 7 12
4 15 14 1
```



⊙  $X(:, 2) = []$

⊙  $X =$

```
16 2 13
5 11 8
9 7 12
4 14 1
```

⊙  $X = A;$

⊙ Không được xoá 1 phần tử kiểu như  $X(1,2) = []$

⊙ Dùng chỉ số với dấu : để xoá 1 hay nhiều phần tử

⊙  $X(2:2:10) = []$        $X \rightarrow 16 \ 9 \ 2 \ 7 \ 13 \ 12 \ 1$

- zeros(n)
- zeros(m,n)
- zeros([m n])
- zeros(size(A))
- ones(n)
- ones(m,n)
- ones([m n])
- ones(size(A))
- eye(n)
- eye(m,n)
- eye(size(A))
- pascal
- magic
- numel(A)
- length(A)
- rand(m,n)
- diag(v,k), diag(v)
- tril, triu
- linspace(a,b),  
linspace(a,b,n)
- logspace(a,b,n)

Stt	Tên hàm	Ý nghĩa	Ví dụ :	Kết quả
1	zeros(a,b)	tạo ma trận axb, các phần tử = 0	zeros(2,3)	<div>0 0 0</div> <div>0 0 0</div>
2	ones(a,b)	tạo ma trận axb, các phần tử = 1	ones(2,3)	<div>1 1 1</div> <div>1 1 1</div>
3	eye(a,b)	tạo ma trận axb, các phần tử đường chéo =1	eye(3,3)	<div>1 0 0</div> <div>0 1 0</div> <div>0 0 1</div>

Stt	Tên hàm	Ý nghĩa	Ví dụ :	Kết quả
4	repmat(a,b)	tạo ma trận bxb các phần tử có giá trị a.	repmat(2,3)	<div>2 2 2</div> <div>2 2 2</div> <div>2 2 2</div>
5	rand(a,b)	tạo ma trận axb, các phần ngẫu nhiên	rand(2,3)	<div>0.9218 0.1763 0.9355</div> <div>0.7382 0.4057 0.9169</div>
6	randn(a,b)	tạo ma trận axb, các phần ngẫu nhiên phân bố đều	randn(2,3)	<div>-0.4326 0.1253 -1.1465</div> <div>-1.6656 0.2877 1.1909</div>
7	linspace(a,b,n)	tạo ma trận hàng n phần tử phân bố đều từ a đến b	linspace(1,1.1,4)	<div>1 2</div>
8	logspace(a,b,n)	tạo ma trận hàng n phần tử phân bố đều từ $10^a$ đến $10^b$	logspace(1,5,3)	<div>10 1000 100000</div>

```
>> D=eye(3)
```

```
D =
```

1	0	0
0	1	0
0	0	1

```
>> pascal(3)
```

```
ans =
```

1	1	1
1	2	3
1	3	6

```
>> eye(3,2)
```

```
ans =
```

1	0
0	1
0	0

```
>> magic(3)
```

```
ans =
```

8	1	6
3	5	7
4	9	2

```
>> A=zeros(3)
```

```
A =
```

```
0 0 0
0 0 0
0 0 0
```

```
>> B=zeros(2,3)
```

```
B =
```

```
0 0 0
0 0 0
```

```
>> size(A)
```

```
ans =
```

```
3 3
```

```
>> zeros(size(B))
```

```
ans =
```

```
0 0 0
0 0 0
0 0 0
```

```
>> numel(B)
```

```
ans =
```

```
6
```

```
>> length(B)
```

```
ans =
```

```
3
```

```
>> rand(3,2)
```

```
ans =
```

```
0.9501 0.4860
0.2311 0.8913
0.6068 0.7621
```

```
>> C=ones(3)
```

```
C =
```

```
1 1 1
1 1 1
1 1 1
```

# vector

Phép tính	Chú thích
<b><math>+, -</math></b>	Cộng hoặc trừ hai ma trận cùng kích thước
<b><math>A * B</math></b>	Nhân hai ma trận A và B
<b><math>A / B</math></b>	Chia hai ma trận (chia phải) A và B
<b><math>A \setminus B</math></b>	Chia trái hai ma trận B và A
<b><math>A .* B</math></b>	Nhân từng phần tử của hai ma trận A và B
<b><math>A ./ B</math></b>	Chia từng phần tử của hai ma trận A và B
<b><math>A . \setminus B</math></b>	Chia từng phần tử của hai ma trận B và A
<b><math>.^{\wedge}</math></b>	Mũ cho từng phần tử của mảng



Stt	Dòng lệnh	Ý nghĩa	Kết quả
1	A=[1 2;3 4]	Tạo ma trận	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
2	B=[1,2;3,4]	Tạo ma trận	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
3	2*A	Nhân một số với ma trận	$\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$
4	B/2	Chia ma trận cho 1 số	$\begin{bmatrix} 0.5000 & 1.0000 \\ 1.5000 & 2.0000 \end{bmatrix}$

- Khi đem ma trận nhân hoặc chia cho 1 số, ta sẽ được một ma trận cùng kích thước, và các phần tử sẽ có giá trị bằng phần tử tương ứng của ma trận cũ nhân hoặc chia cho số đó.



Stt	Dòng lệnh	Ý nghĩa	Kết quả
5	A+B	Cộng 2 ma trận	$\begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}$
6	A-B	Trừ 2 ma trận	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
<p>- 2 ma trận đem cộng hoặc trừ phải cùng kích thước</p> <p>- Kết quả là một ma trận có chung kích thước, các phần tử là tổng hoặc hiệu của 2 phần tử tương ứng.</p>			
7	A*B	Nhân 2 ma trận	$\begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}$
8	A\B	Chia 2 ma trận	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
<p>- Chỉ áp dụng khi giải phương trình <math>AX=B</math>, với A là ma trận vuông, B là ma trận cột có cùng kích thước với A</p>			

Stt	Tên hàm	Ý nghĩa	Ví dụ :	Kết quả
1	size	Kích thước ma trận	size(Z)	4 4
2	ndims	số chiều	ndims(Z)	2
3	length	chiều dài	length(Z)	4
4	numel	số phần tử ma trận.	numel(Z)	16
5	max	Vectơ hàng, chứa các phần tử lớn nhất theo từng cột	max(Z)	4 6 8 10

Stt	Tên hàm	Ý nghĩa	Ví dụ :	Kết quả
6	min	Vector hàng, chứa các phần tử nhỏ nhất theo từng cột	min(Z)	1    3    5    7
7	sum	Vector hàng, chứa tổng các phần tử theo từng cột	sum(Z)	10    18    26    34
8	sort	Sắp xếp theo thứ tự tăng dần trong từng cột.	sort(Z)	1    3    5    7 2    4    6    8 3    5    7    9 4    6    8    10

Cho ma trận  $A = \begin{pmatrix} 11 & 2 & 6 \\ 4 & 17 & 26 \\ 17 & 8 & 49 \end{pmatrix}$

Stt	Tên hàm	Ý nghĩa	Ví dụ :	Kết quả
1	det	Tính định thức ma trận	det(A)	5825
- Định thức ma trận không được định nghĩa ở đây, sinh viên tham khảo trong tài liệu toán cao cấp.				
2	'	Chuyển vị ma trận	$A'$	$\begin{pmatrix} 11 & 4 & 17 \\ 2 & 17 & 8 \\ 6 & 26 & 49 \end{pmatrix}$
- Ma trận chuyển vị là ma trận được tạo từ ma trận ban đầu bằng cách : + Chuyển các hàng thành cột + Hoặc chuyển các cột thành hàng				

Stt	Tên hàm	Ý nghĩa	Ví dụ :	Kết quả
3	tril	tạo ma trận tam giác dưới	tril(A)	<div>11   0   0</div> <div>4   17   0</div> <div>17   8   49</div>
- Ma trận tam giác dưới bỏ các thành phần từ đường chéo trở lên.				
4	triu	tạo ma trận tam giác trên	triu(A)	<div>11   2   6</div> <div>0   17   26</div> <div>0   0   49</div>
Ma trận tam giác trên bỏ các thành phần từ đường chéo trở xuống.				
5	chol	Phân tích ma trận thành thừa số cholesky	chol(A)	<div>3.3166   0.6030   1.8091</div> <div>0   4.0788   6.1070</div> <div>0   0   2.9037</div>

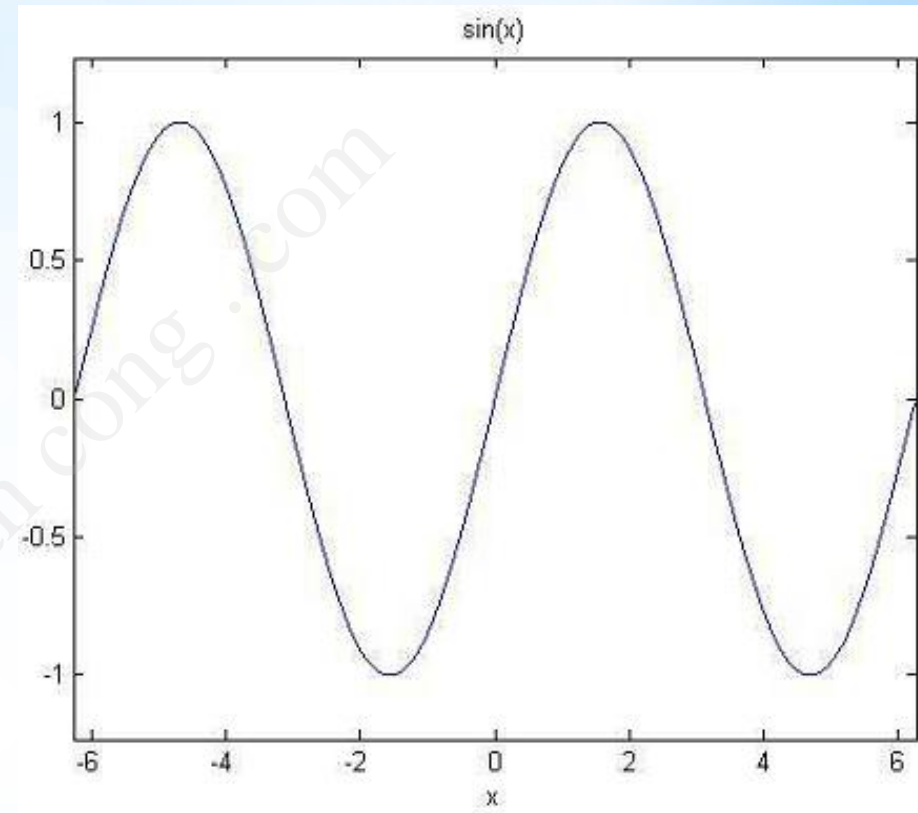
Stt	Tên hàm	Ý nghĩa	Ví dụ :	Kết quả
- Ma trận A sẽ được phân tích thành 2 ma trận : $R'.R$ , với R là ma trận tam giác trên.				
6	lu	Phân tích ma trận thành thừa số LU	$[L,U]=lu(A)$	L : 0.6471   -0.2101 1.0000 0.2353   1.0000    0 1.0000        0    0 U : 17.0000   8.0000 49.0000 0   15.1176   14.4706 0        0   -22.6654
- Ma trận A sẽ được phân tích thành 2 ma trận : $L.U$ với L là ma trận tam giác dưới, U là ma trận tam giác trên..				
7	inv	Nghịch đảo ma trận	$inv(A)$	0.1073   -0.0086   -0.0086 0.0422   0.0750   -0.0450 -0.0441   -0.0093   0.0307
- Ma trận nghịch đảo của ma trận A là ma trận B sao cho $A*B=B*A=I$ Với <b>I</b> là ma trận đơn vị				

## 5. Đồ Thị 2D trong Matlab

- Hàm `plot(x, y)` vẽ các điểm  $x$ ,  $y$  tương ứng lên mặt phẳng
- Ví dụ:  

```
>> x = -2*pi:pi/100:2*pi;  
>> plot(x, sin(x))
```
- Chú ý: đối với các hàm sơ cấp có thể dùng lệnh `ezplot('hàm số cần vẽ')`. Lệnh sau vẽ đồ thị hàm  $\sin(x)$   

```
>> ezplot('sin(x)')
```



# Vẽ nhiều đồ thị

Để vẽ nhiều đồ thị, chúng ta cũng sử dụng hàm plot với cú pháp sau

Cú pháp : **Plot(t,x,t,y)**

Matlab sẽ vẽ 1 đường x theo t và một đường y theo t.

Hoặc chúng ta cũng có thể sử dụng lệnh hold

Cú pháp : **hold**

Lệnh hold sẽ lưu giữ đồ thị hiện hữu, khi chúng ta thực hiện lệnh vẽ tiếp theo thì đồ thị mới sẽ được thêm vào đồ thị cũ.

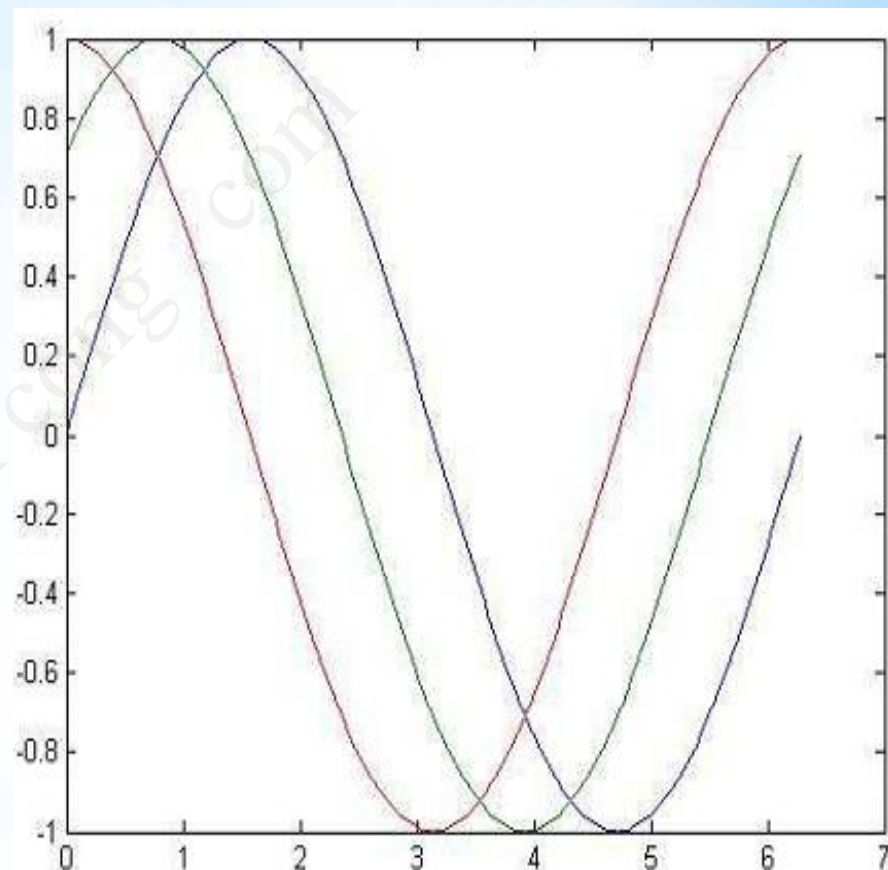


- Dùng dạng `plot(x1, y1, x2, y2, ...)`
- Ví dụ:

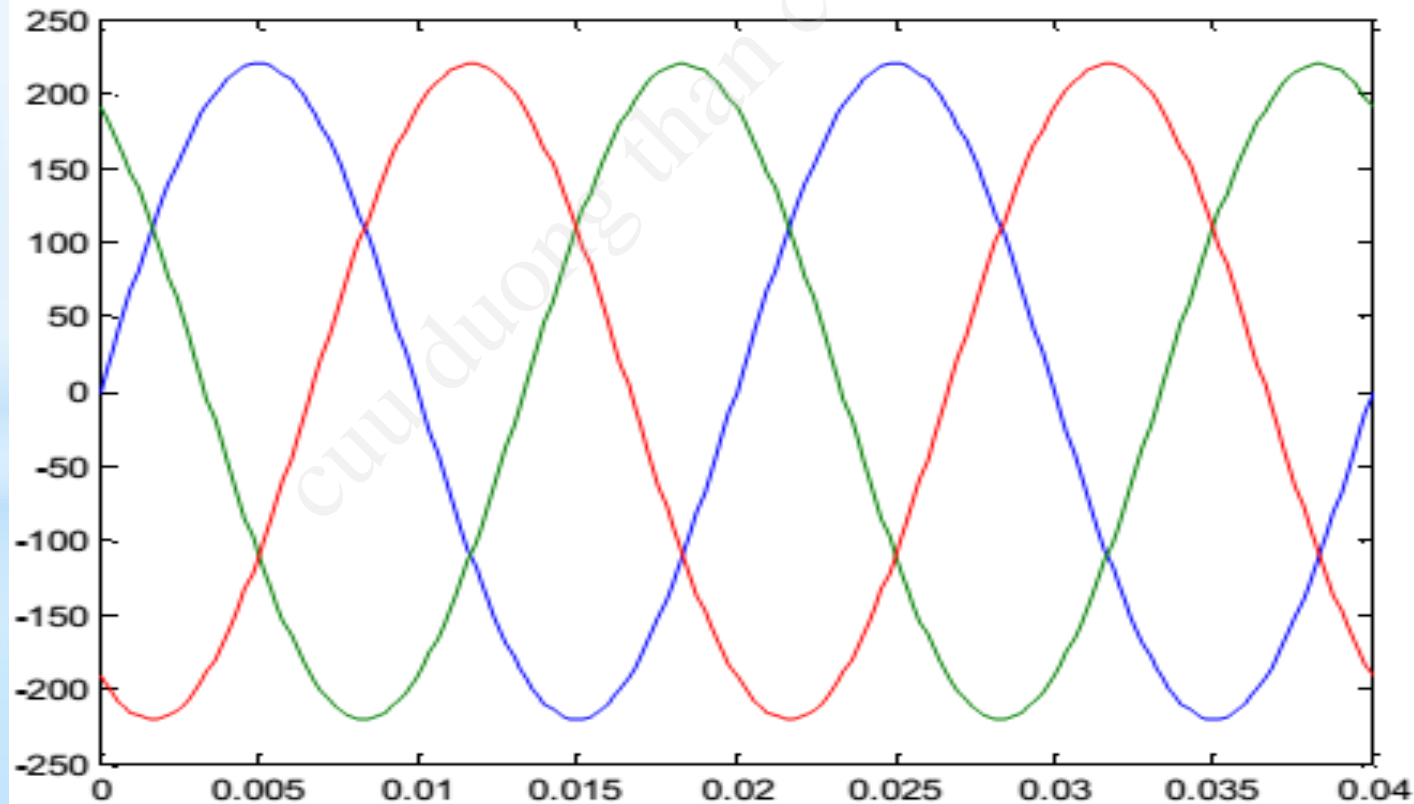
```
>> x=0:pi/100:2*pi;  
>> y1 = sin(x); y2 = sin(x+pi/4);  
>> y3 = sin(x+pi/2);  
>> plot(x,y1,x,y2,x,y3)
```

- Cũng có thể dùng `plot(x,y)`, trong đó

```
>> y = [y1; y2; y3]  
>> plot(x,y)
```



```
f=50  
T=1/f  
t=0:T/100:2*T  
va=220*sin(2*pi*f*t)  
plot(t,va)  
hold on  
vb=220*sin(2*pi*f*t+120*pi/180)  
plot(t,vb)  
hold on  
vc=220*sin(2*pi*f*t-120*pi/180)  
plot(t,vc)
```



# Các thuộc tính nét vẽ

Cú pháp : **plot(X1,Y1,LineSpec,...)**

Hàm plot có sử dụng thuộc tính nét vẽ LineSpec cho phép thiết lập các thuộc tính của đồ thị. Các thuộc tính là :

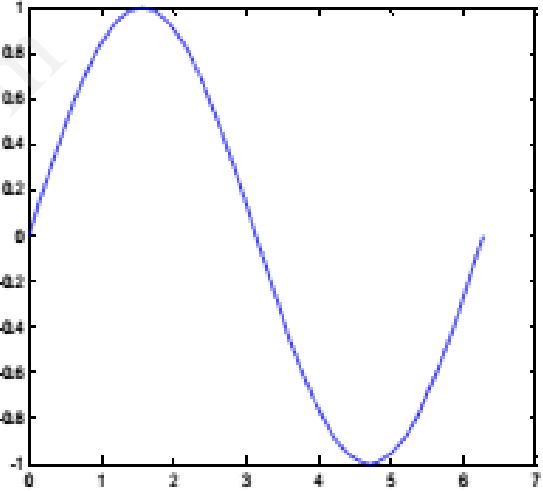
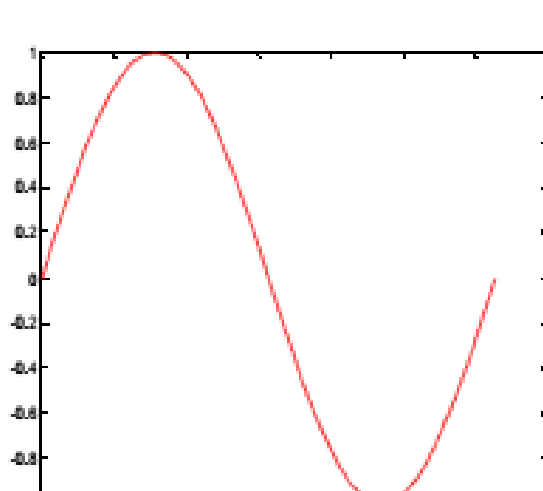
Line style : định dạng kiểu nét , ví dụ nét chấm, nét đứt, ....

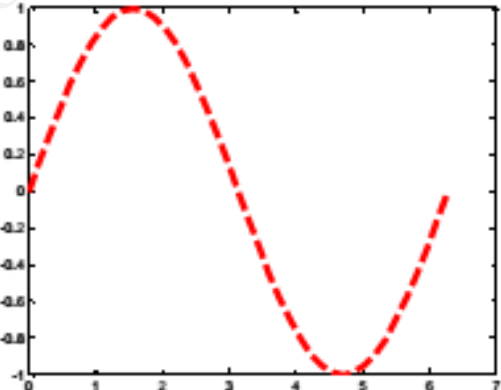
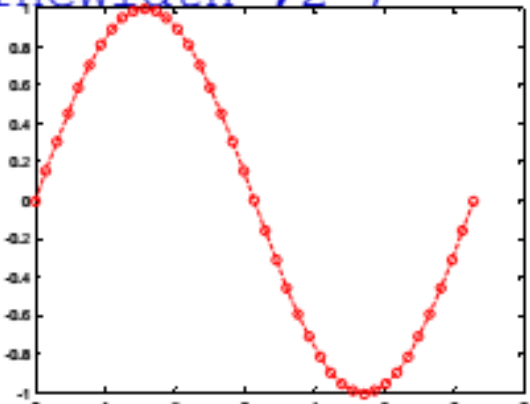
Line width : định dạng độ rộng nét vẽ, đơn vị là point

Color : định dạng màu của nét vẽ, ví dụ màu xanh, đỏ,

...

Marker type : định dạng điểm vẽ.

Thuộc tính	Lập trình trong Matlab	Đồ thị
<b>Loại nét vẽ</b> - : nét liền -- : nét đứt : : nét chấm -. : nét chấm gạch	<pre>t = 0:pi/20:2*pi; plot(t,sin(t),'--')</pre>	 <p>The plot shows a sine wave, <math>y = \sin(x)</math>, over the interval <math>x \in [0, 2\pi]</math>. The x-axis ranges from 0 to 7 with major ticks every 1 unit. The y-axis ranges from -1 to 1 with major ticks every 0.2 units. The curve is plotted with a dashed blue line, starting at (0,0), reaching a peak at <math>(\pi/2, 1)</math>, crossing the x-axis at <math>(\pi, 0)</math>, reaching a trough at <math>(3\pi/2, -1)</math>, and ending at <math>(2\pi, 0)</math>.</p>
<b>Màu nét vẽ</b> r : đỏ y : vàng w : trắng b : xanh nước biển k : đen	<pre>t = 0:pi/20:2*pi; plot(t,sin(t),'-- r')</pre>	 <p>The plot shows a sine wave, <math>y = \sin(x)</math>, over the interval <math>x \in [0, 2\pi]</math>. The x-axis ranges from 0 to 7 with major ticks every 1 unit. The y-axis ranges from -1 to 1 with major ticks every 0.2 units. The curve is plotted with a dashed red line, starting at (0,0), reaching a peak at <math>(\pi/2, 1)</math>, crossing the x-axis at <math>(\pi, 0)</math>, reaching a trough at <math>(3\pi/2, -1)</math>, and ending at <math>(2\pi, 0)</math>.</p>

Thuộc tính	Lập trình trong Matlab	Đồ thị
<b>Độ lớn nét vẽ</b> - Tính theo đơn vị là point	<pre>t = 0:pi/20:2*pi; plot(t,sin(t),'-- r','LineWidth',5)</pre>	 <p>The plot shows a sine wave <math>y = \sin(x)</math> for <math>x</math> in the range <math>[0, 2\pi]</math>. The x-axis ranges from 0 to 7 with major ticks every 1 unit. The y-axis ranges from -1 to 1 with major ticks every 0.2 units. The line is a thick dashed red line, representing the 'LineWidth' property set to 5.</p>
<b>Marker</b> + : dấu cộng o : vòng tròn * : hoa thị x : chữ x	<pre>t = 0:pi/20:2*pi; plot(t,sin(t),'-- r o', 'LineWidth',2)</pre>	 <p>The plot shows a sine wave <math>y = \sin(x)</math> for <math>x</math> in the range <math>[0, 2\pi]</math>. The x-axis ranges from 0 to 7 with major ticks every 1 unit. The y-axis ranges from -1 to 1 with major ticks every 0.2 units. The line is a thin dashed red line with open circle markers at each data point, representing the 'LineWidth' property set to 2 and the 'Marker' property set to 'o'.</p>

# Lệnh cho trục tọa độ

- Lệnh axis

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

- Tùy chỉnh các kiểu trục tọa độ

- axis on/off/auto

- axis normal/square/equal/tight

- axis ij/xy

- grid on/off

- Xác định giới hạn của trục Ox và Oy.

```
xlim([xmin xmax])
```

```
ylim(ymin ymax)
```

**Ví dụ:**

```
>>x = 0:.025:pi/2;
```

```
>>y = tan(x);
```

```
>>plot(x,tan(x),'-ro');
```

```
>> axis([0 pi/2 0 5])
```

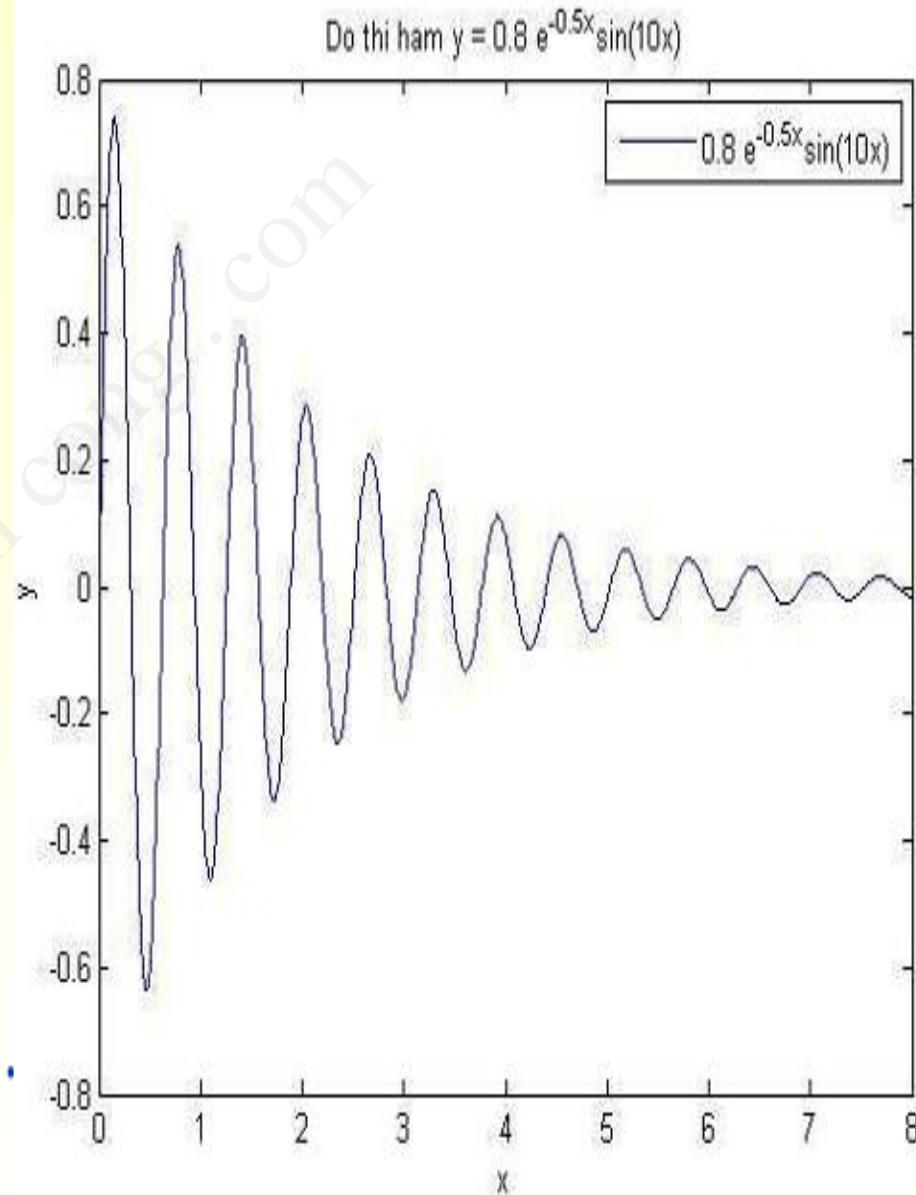
```
>> grid on
```

# Chú thích đồ thị

- xlabel, ylabel, title, legend

- Ví dụ:

```
>> x = 0:0.01:20;  
>> y = 0.8*exp(-0.5*x).*sin(10*x);  
>> plot(x,y);  
>> xlim([0 8]);  
>> xlabel('x');  
>> ylabel('y');  
>> legend('0.8 e{-0.5x}sin(10x)');  
>> title('Do thi ham y = 0.8 e{-0.5x} ...  
sin(10x)');
```

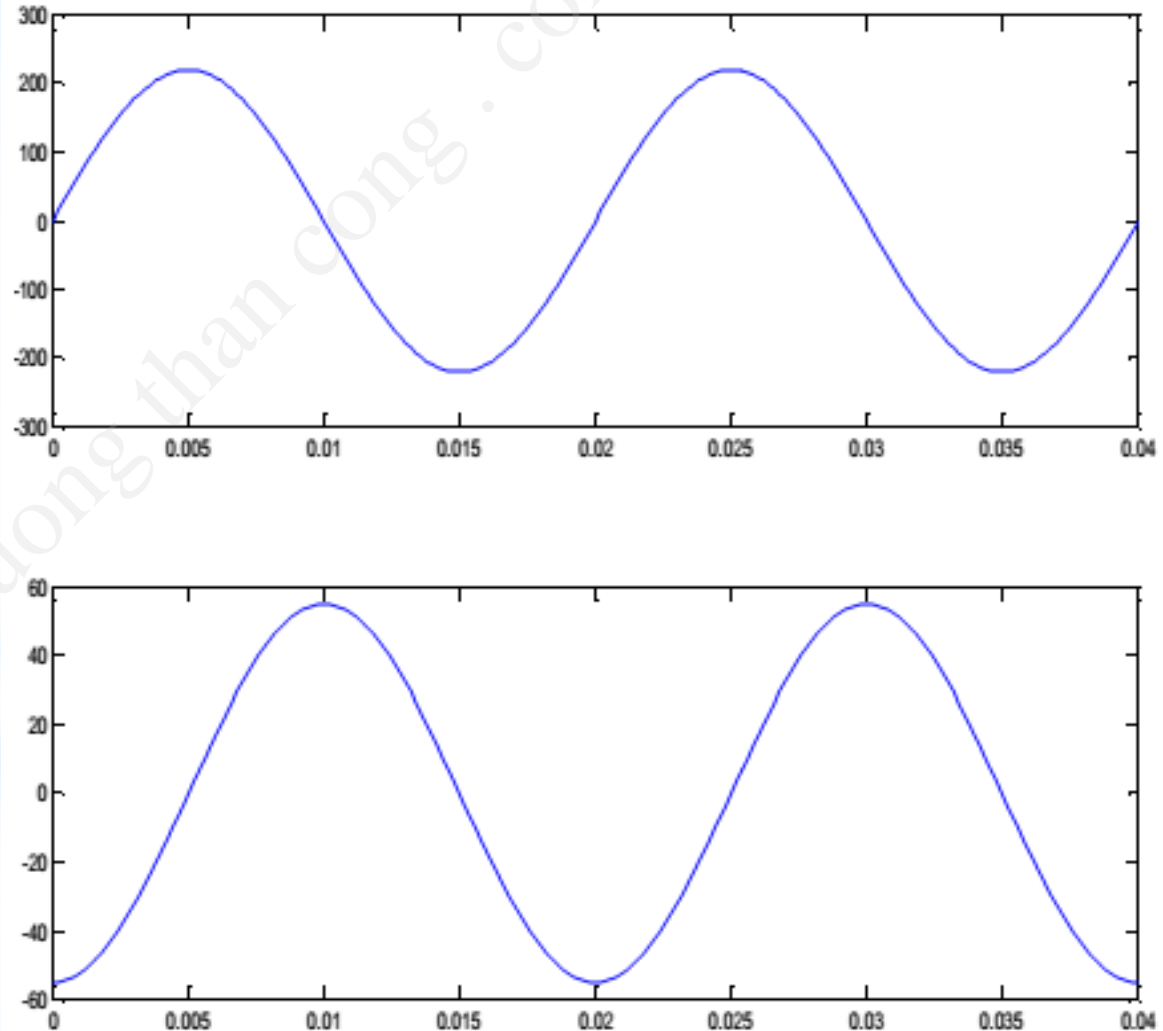


# Hàm subplot

Cú pháp : **subplot(p,q,i)**

Hàm subplot sẽ chia cửa sổ đồ họa ra thành các ô nhỏ. Và đồ thị sẽ được vẽ vào ô thứ i.

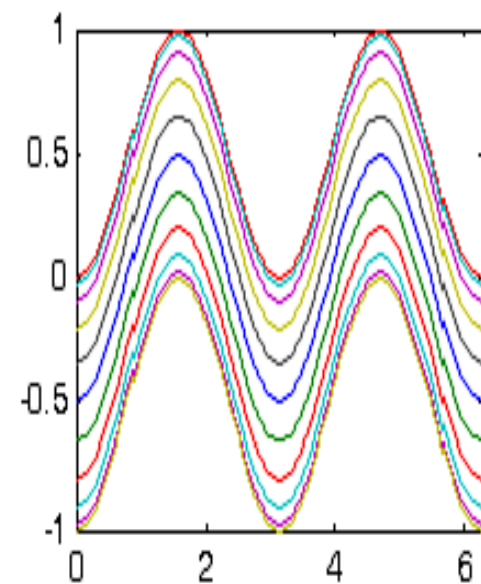
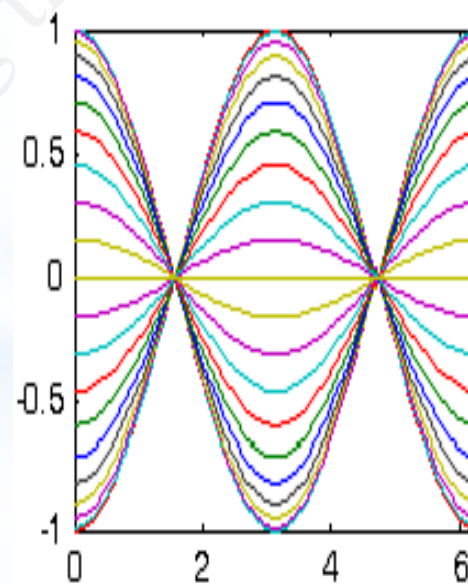
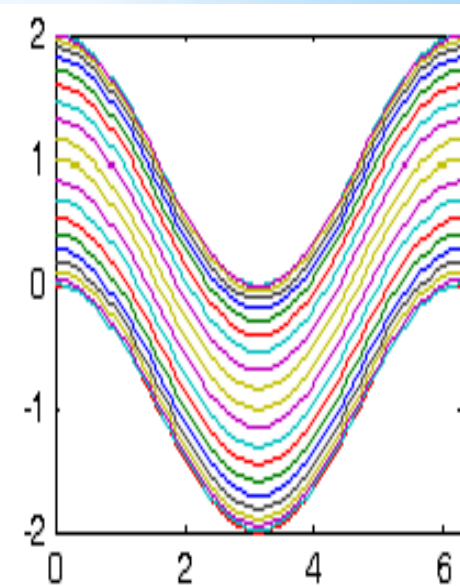
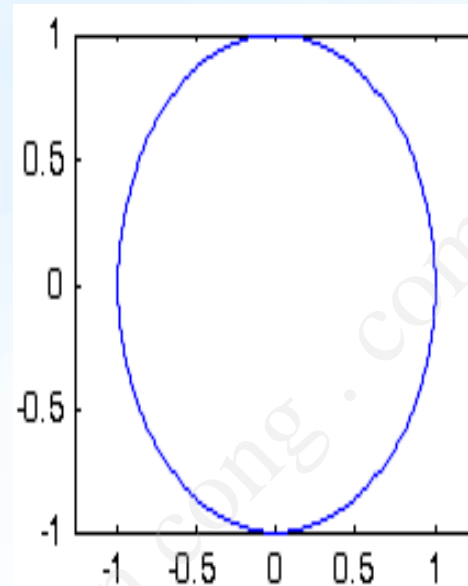
```
f=50  
T=1/f  
t=0:T/100:2*T  
v=220*sin(2*pi*f*t)  
i=55*sin(2*pi*f*t-  
90*pi/180)  
subplot(2,1,1)  
plot(t,v)  
subplot(2,1,2)  
plot(t,i)
```





## Ví dụ

```
t = 0:pi/20:2*pi;  
[x,y] = meshgrid(t);  
subplot(2,2,1)  
plot(sin(t),cos(t))  
axis equal  
subplot(2,2,2)  
z = sin(x)+cos(y);  
plot(t,z)  
axis([0 2*pi -2 2])  
subplot(2,2,3)  
z = sin(x).*cos(y);  
plot(t,z)  
axis([0 2*pi -1 1])  
subplot(2,2,4)  
z = (sin(x).^2)-(cos(y).^2);  
plot(t,z)  
axis([0 2*pi -1 1])
```



# Các hàm vẽ đồ thị khác

## a. Hàm bar : vẽ đồ thị thanh đứng

Cú pháp: **bar(x,Y)**

Hàm bar sẽ vẽ các thanh đứng có độ cao là yi tương ứng tại vị trí xi

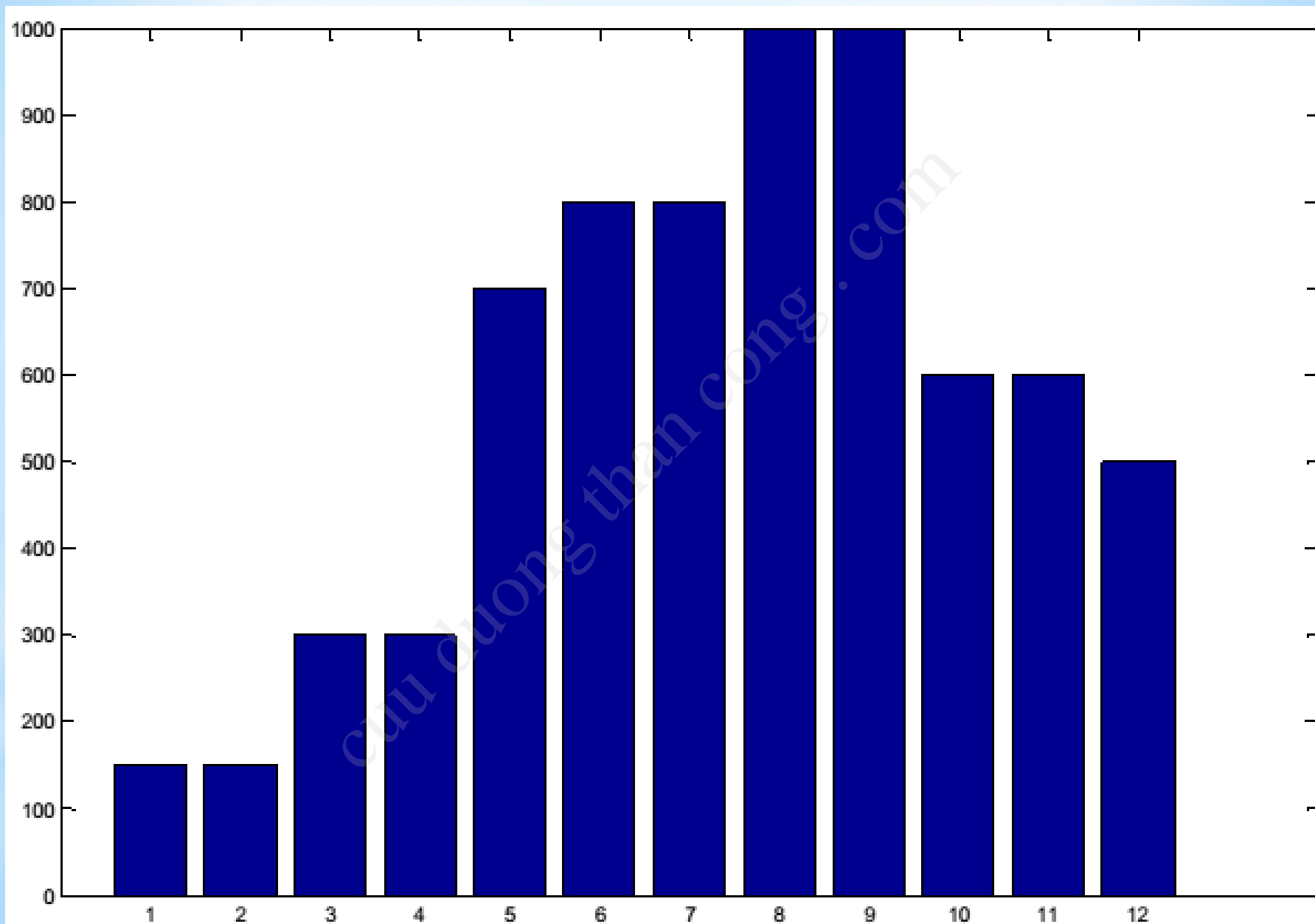
**Ví dụ 5.8** Vẽ đồ thị phụ tải của một xí nghiệp theo số liệu sau :

Tháng	1	2	3	4	5	6	7	8	9	10	11	12
Công suất -kw	150	150	300	300	700	800	800	1000	1000	600	600	500

```
x=[1 2 3 4 5 6 7 8 9 10 11 12]
```

```
y=[150 150 300 300 700 800 800 1000 1000 600 600  
500]
```

```
bar (x,y)
```



## b. Hàm pie : vẽ đồ thị dạng hình quạt

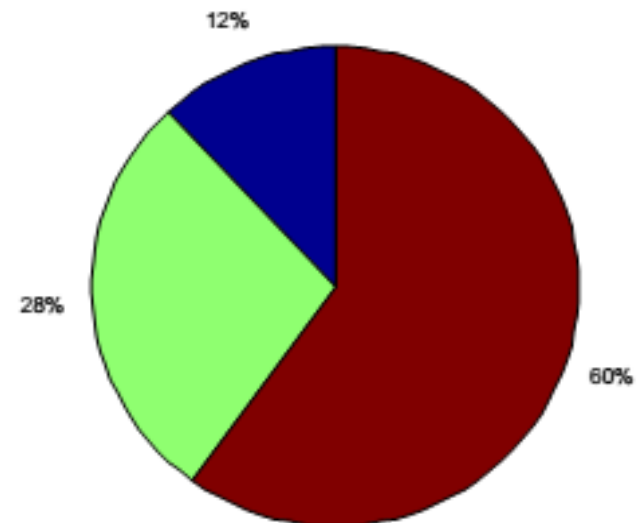
Cú pháp : **pie(X)**

Hàm pie sẽ vẽ đồ thị dạng quạt với diện tích tỉ lệ với phần trăm.

**Ví dụ 5.9:** Cho xí nghiệp có điện năng tiêu thụ trong hàng tháng được tính toán theo giờ cao điểm, thấp điểm và trung bình như sau :

Điện năng	Cao điểm	Thấp điểm	Trung bình
Điện năng-kwh	15000	35000	75000

**x=[15000 35000 75000]**  
**pie(x)**



## Vẽ các mặt

Bước đầu tiên là đưa ra đồ thị lưới của hàm hai biến  $z = f(x,y)$ , tương ứng với ma trận  $X$  và  $Y$  chứa các hàng và các cột lặp đi lặp lại, MATLAB cung cấp hàm **meshgrid** cho mục đích này:

**$X,Y] = \text{meshgrid}(x,y)$ :** tạo một ma trận  $X$ , mà các hàng của nó là bản sao của vector  $x$ , và ma trận  $Y$  có các cột của nó là bản sao của vector  $y$ . Cặp ma trận này sau đó được sử dụng để ước lượng hàm hai biến sử dụng đặc tính toán học về mảng của MATLAB.

Để vẽ bề mặt ta sử dụng các hàm:

**mesh (X,Y,Z):** nối các điểm với nhau trong một lưới chữ nhật.

**meshc (X,Y,Z):** vẽ các đường contour bên dưới đồ thị.

**meshz (X,Y,Z):** vẽ các đường thẳng đứng viền quanh đồ thị.

**waterfall X,Y,Z):** vẽ mặt với hiệu ứng như thác đổ.

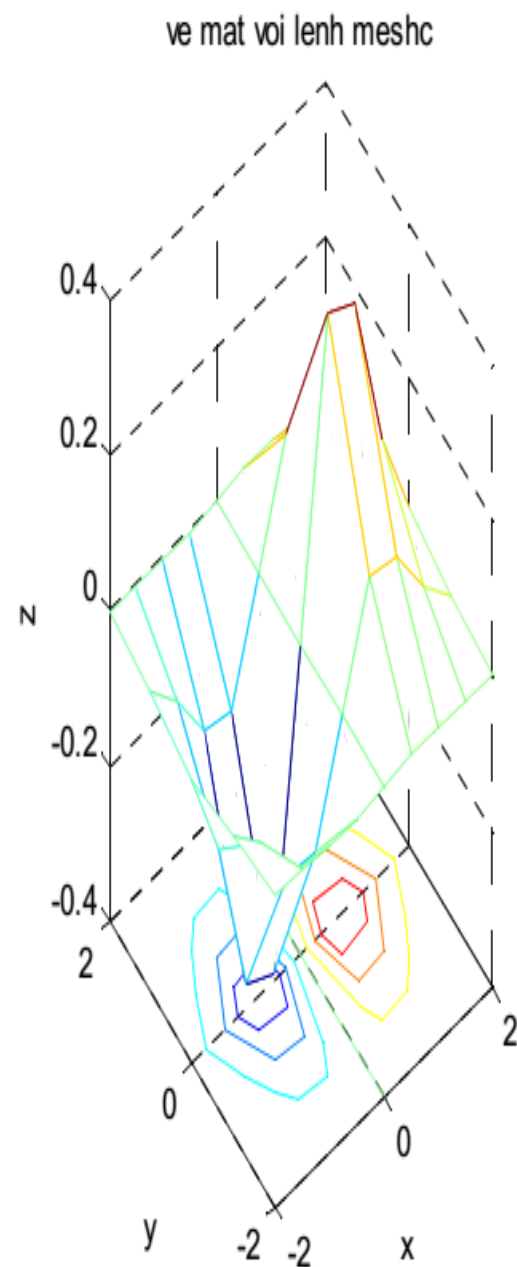
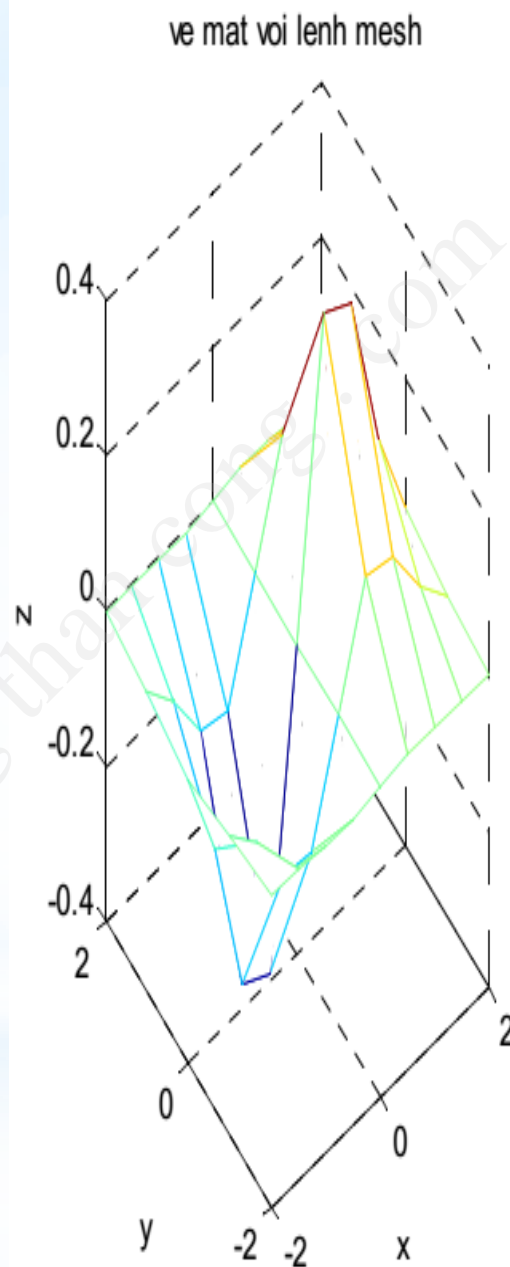
**Ví dụ:** Vẽ mặt xác định bởi phương trình:  $z(x, y) = xe^{-x^2-y^2}$

```
>> x=-2:0.5:2;
```

```
>> y=-2:1:2;
```

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

```
>> subplot(1,2,1)
>> mesh(X,Y,Z)
>> xlabel('x')
>> ylabel('y')
>> zlabel('z')
>> title('ve mat voi lenh mesh')
>> subplot(1,2,2)
>> meshc(X,Y,Z)
>> xlabel('x')
>> ylabel('y')
>> zlabel('z')
>> title('ve mat voi lenh meshc')
```

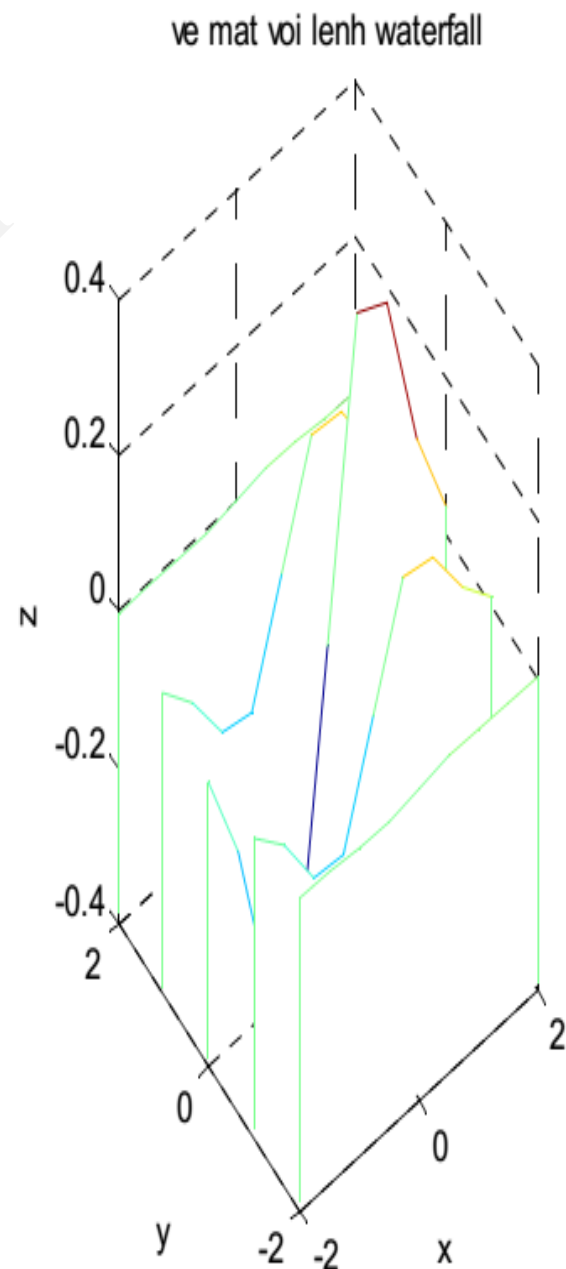
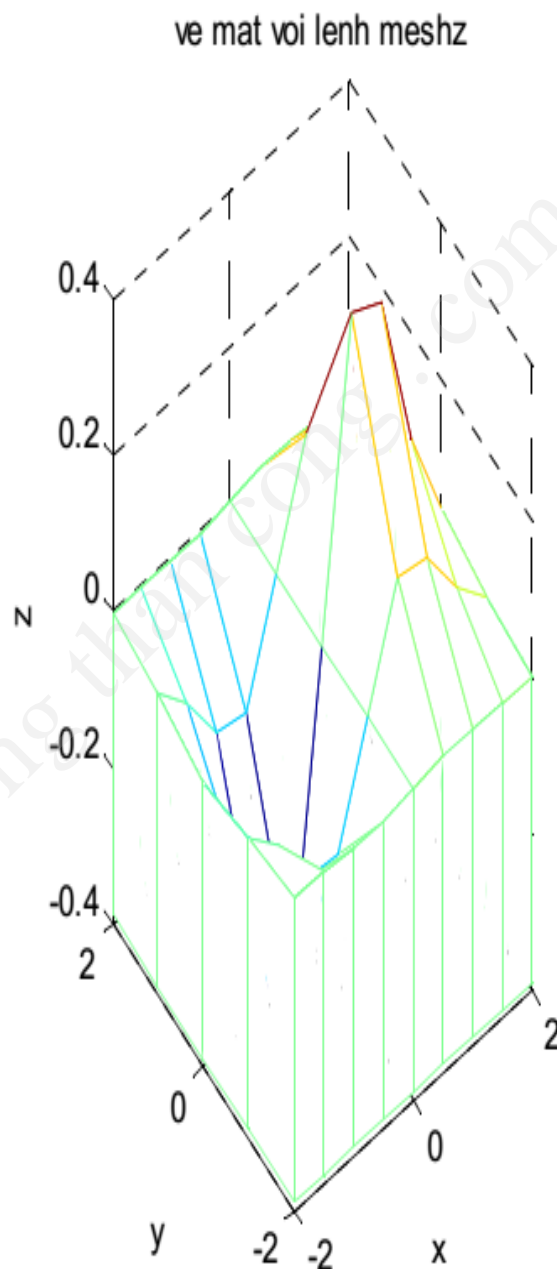




```

>> subplot(1,2,1)
>> meshz(X,Y,Z)
>> xlabel('x')
>> ylabel('y')
>> zlabel('z')
>> title('ve mat voi lenh meshz')
>> subplot(1,2,2)
>> waterfall(X,Y,Z)
>> xlabel('x')
>> ylabel('y')
>> zlabel('z')
>> title('ve mat voi lenh
waterfall')

```

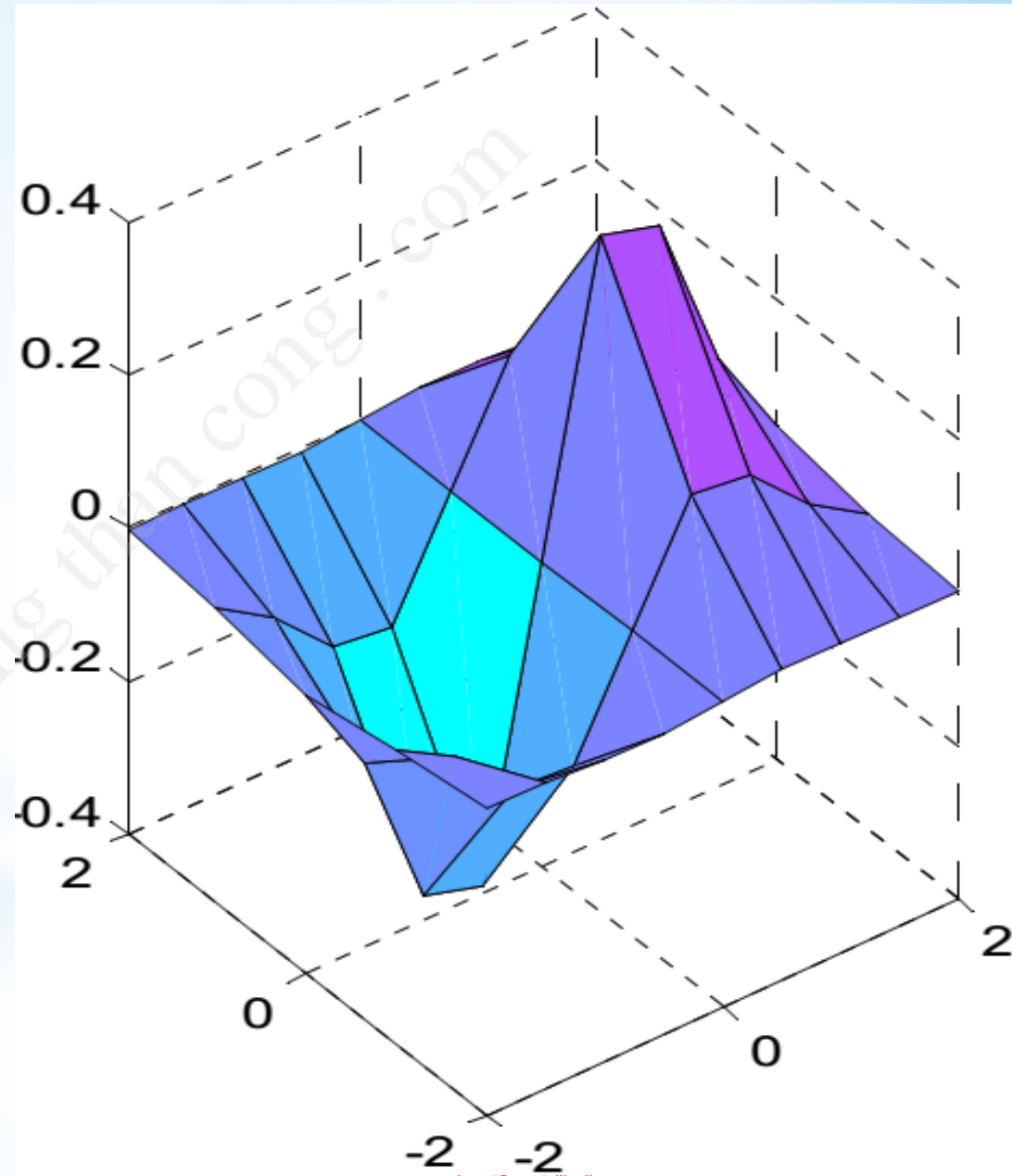




# Vẽ các mặt được tô bóng từ một ma trận bằng hàm surf, surfc

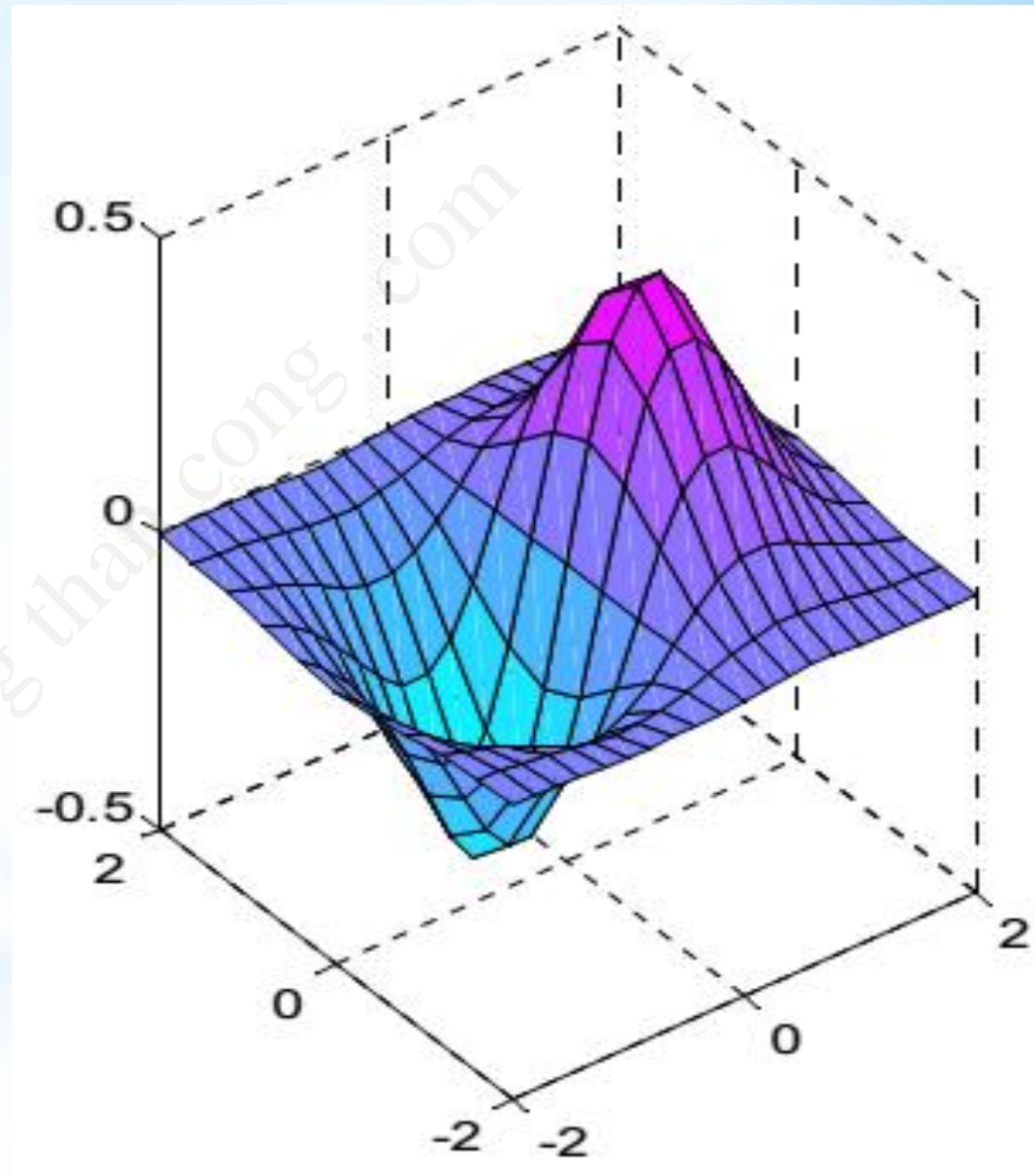
## Ví dụ:

```
>> x=-2:0.5:2;  
>> y=-2:1:2;  
>> [X,Y]=meshgrid(x,y);  
>> Z=X.*exp(-X.^2-Y.^2);  
>> surf(X,Y,Z)  
>> colormap(hot)
```



Ta có thể tạo nhiều lưới hơn để mặt mịn hơn

```
>> x=-2:0.2:2;  
>> y=-2:0.4:2;  
>> [X,Y]=meshgrid(x,y);  
>> Z=X.*exp(-X.^2-Y.^2);  
>> surf(X,Y,Z)  
>> colormap(cool)
```



Lệnh **surfc** (X,Y,Z): vẽ mặt có các đườn contour phía dưới.

Lệnh **surfl** (X,Y,Z,s): vẽ mặt có bóng sáng. Đối số s xác định hướng của nguồn sáng trên bề mặt vẽ. s là một vectơ tùy chọn trong hệ tọa độ decac hay trong tọa độ cầu. Nếu không khai báo giá trị mặc định của s là  $45^\circ$  theo chiều kim đồng hồ từ vị trí người quan sát.

## 6. Phép tính với biến symbolic

### Khai báo

#### ● Khai báo biến:

> `syms a b c x`

hoặc

> `a = sym('a')`

> `b = sym('b')`

> `c = sym('c')`

> `x = sym('x')`

#### ● Khai báo biến phức

> `x = sym('x','real'); y = sym('y','real')`

hoặc `syms x y real`

> `z = x + i*y`

# Khai báo biểu thức:

● Khai báo biểu thức:  $f = 2*x + b$

> `syms x b`

> `f = 2*x + b`

hoặc

> `f = sym('2*x + b')`

> `sym('(sqrt(2) + 1)/3')`

> `g = syms('5')` (khác `g = 5`)

> `syms x y`

> `h = x^2 + y^2`

⦿ **Lệnh findsym: tìm biến hình thức trong biểu thức.**

⦿ **Ví dụ**

> syms a b n t x z

> s = x^n; g = sin(a\*t + b)

> findsym(f)

> ans = x n

> findsym(g)

> ans = a b t

⦿ **findsym(g,1): tìm biến hình thức mặc định**

> findsym(g,1)

> ans = t

# Hiển thị biến hình thức dưới dạng số học

⊙  $t = 0.1$

- > sym(t, 'f')
- > ans = '1.999999999999999a'\*2^(-4)
- > sym(t, 'r')
- > ans = 1/10
- > sym(t, 'e')
- > ans = 1/10+eps/40
- > sym(t, 'd')
- > ans = .1000000000000000000000555111512312578
- > digits(7)
- > sym(t, 'd')
- > ans = .1000000

Các phép toán cơ bản

Stt	Lệnh trong Matlab	Ý nghĩa	Kết quả
1	<code>x=sym('x','real')</code>	Tạo biến x là số thực	
2	<code>x=sym('x','positive')</code>	Tạo biến x là số thực dương	
3	<code>syms x y;</code>	Định nghĩa 2 biến x,y	
4	<code>syms</code>	Liệt kê các biến mà chương trình quản lý	'x' 'y'
2	<code>A=x+1</code> <code>B=y^2-1</code>	Tạo 2 biến symbolic mới A và B	
3	<code>A+B</code>	cộng 2 biến A và B	$x+y^2$
4	<code>A-B</code>	Trừ 2 biến A và B	$x+2-y^2$
5	<code>A*B</code>	Nhân 2 biến A và B	$(x+1)*(y^2-1)$
6	<code>A/B</code>	Biến A chia biến B	$(x+1)/(y^2-1)$
7	<code>A^B</code>	A lũy thừa B	$(x+1)^(y^2-1)$



## c. Các phép tính đạo hàm và tích phân

Stt	Lệnh trong Matlab	Ý nghĩa	Kết quả
1	diff(B)	Tính đạo hàm biểu thức B	$2*y$
2	diff(B,2)	Tính đạo hàm bậc 2 biểu thức B	2
3	diff(A*B,y)	Tính đạo hàm biểu thức A*B theo biến y	$2*(x+1)*y$
4	diff(A*B,y,2)	Tính đạo hàm bậc 2 biểu thức A*B theo biến y	$2*x+2$
5	int(A)	Tích tích phân biến A	$1/2*x^2+x$
6	int(A,0,5)	Tích tích phân biến A từ 0 đến 5	$35/2$
7	int(A*B,x)	Tích tích phân biến A*B theo biến x	$(y^2-1)*(1/2*x^2+x)$

# Đạo hàm

## ⊙ **diff(Y)**

Y: hàm số hoặc biến hình thức cần lấy đạo hàm.

## ⊙ **Ví dụ**

- > `syms x; f = sin(5*x)`
- > `diff(f)`
- > `ans = 5*cos(5*x)`
- > `g = exp(x)*cos(x)`
- > `diff(g)`
- > `ans = exp(x)*cos(x) - exp(x)*sin(x)`
- > `c = sym('5'); diff(c)`
- > `ans = 0`

> diff(5)

> ans = [ ] vì 5 không phải là biến hình thức

## ◎ Lấy đạo hàm cấp 2

> diff(g,2)

hoặc

> diff(diff(g))

> ans = -2exp(x)\*sin(x)

## ◎ Đạo hàm đa biến

Gọi  $f = f(x,y)$  thì

■ Đạo hàm theo x: diff(f,x)

■ Đạo hàm theo y: diff(f,y)

- Đạo hàm cấp 2 theo x:  $\text{diff}(f,x,2)$
- Đạo hàm cấp 2 theo y:  $\text{diff}(f,y,2)$
- Nếu x là biến mặc định của f thì  $\text{diff}(f,2)$  tương đương với  $\text{diff}(f,x,2)$ .

## ○ Ví dụ

- `syms s t`
- `f = sin(s*t)`
- `diff(f,t)`  $\Rightarrow \text{ans} = \cos(s*t)*s$
- `diff(f,s)`  $\Rightarrow \text{ans} = \cos(s*t)*t$
- `diff(f,t,2)`  $\Rightarrow \text{ans} = -\sin(s*t)*s^2$
- `findsym(f,1)`  $\Rightarrow \text{ans} = t$

Suy ra biến mặc định là t do đó  $\text{diff}(f,2) = \text{diff}(f,t,2)$

## ○ Đạo hàm đối với ma trận

- `syms a x`
- `A = [cos(a*x) sin(a*x); -sin(a*x) cos(a*x)]`
- `A =`  
$$\begin{bmatrix} \cos(ax) & \sin(ax) \\ -\sin(ax) & \cos(ax) \end{bmatrix}$$
- `diff(A)`
- `ans =`  
$$\begin{bmatrix} -\sin(ax)*a & \cos(ax)*a \\ -\cos(ax)*a & -\sin(ax)*a \end{bmatrix}$$

# Tích phân

◉  $\text{int}(f,x)$  hoặc  $\text{int}(f)$  : Tìm nguyên hàm của hàm  $f = f(x)$ .

◉  $\text{int}(f,a,b)$  : Tính tích phân của  $f$  từ  $a \rightarrow b$ .

◉ Ví dụ

> `syms x n a b t`

> `f = x ^ n`

> `int(f)` ( hoặc `int(f,x)` )

> `ans = x^(n+1)/(n+1)`

> `g = cos(a*t + b)`

> `int(g)`

> `ans = sin(a*t + b)/a`

> `h = sin(2*x)`

> `int(h,0,pi/2)`

> `ans = 1`

> `u = exp(-x^2)`

> `int(u,0,inf)`

> `ans = 1/2*pi^(1/2)`

# Giới hạn

⊙ `limit(f)` :  $\lim_{x \rightarrow 0} f(x)$

⊙ `limit(f,x,a)` :  $\lim_{x \rightarrow a} f(x)$

hoặc `limit(f,a)`

⊙ `limit(f,x,a,'left')` :  $\lim_{x \rightarrow a^+} f(x)$

⊙ `limit(f,x,a,'right')` :  $\lim_{x \rightarrow a^-} f(x)$

## ⊙ Ví dụ

- > `sym h n x`
- > `limit((cos(x + h) - cos(x))/h,h,0)`
- > `ans = -sin(x)`
- > `limit((1 + x/n)^n,n,inf)`
- > `ans = exp(x)`
- > `limit(x/abs(x),x,0,'left')`
- > `ans = -1`
- > `limit(x/abs(x),x,0,'right')`
- > `ans = 1`
- > `limit(x/abs(x),x,0)`
- > `ans = NaN`

# Tổng chuỗi

⊙ Tính:  $1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots$   
 $1 + x + x^2 + \dots$

- > `syms x k`
- > `s1 = symsum(1/k^2,1,inf)`
- > `s2 = symsum(x^k,k,0,inf)`
- > `s1 = 1/6*pi^2`
- > `s2 = -1/(x-1)`



## d. Các phép tính toán thu gọn.

Stt	Lệnh trong Matlab	Ý nghĩa	Kết quả
1	<code>collect((x+1)^2+(x+y)^2,y)</code>	Gom các lũy thừa cùng bậc theo biến y	$y^2+2*x*y+(x+1)^2+x^2$
2	<code>expand((x+y+1)^2)</code>	Khai triển các biểu thức	$x^2+2*x*y+2*x+y^2+2*y+1$
3	<code>factor(x^3-y^3)</code>	Phân tích thành thừa số	$(x-y)*(x^2+x*y+y^2)$
4	<code>simplify((x^3-y^3)/(x^2-y^2))</code>	Thu gọn biểu thức	$(x^2+x*y+y^2)/(x+y)$

⦿  $\text{collect}(f) - f = f(x)$

⦿  $\text{collect}(f, y) - f = f(x, y, \dots)$

- Đơn giản hàm  $f$  bằng các nhóm các biến  $x$  có cùng số mũ.
- Trường hợp  $f$  có nhiều biến  $\text{collect}(f, y)$  sẽ chỉ định gom nhóm theo biến  $y$ .
- $\text{collect}(f)$  gom nhóm theo biến mặc định được chỉ ra trong  $\text{findsym}(f)$ .

## ● Ví dụ

- > `syms x t`
- > `f = x^3 - 6*x^2 + 11*x - 6`
- > `g = (x - 1)*(x - 2)*(x - 3)`
- > `h = -6 + (11 + (-6 + x)*x)*x`
- > `pretty(f), pretty(g), pretty(h)`
- > `collect(f) => ans = x^3 - 6*x^2 + 11*x - 6`
- > `collect(g) => ans = x^3 - 6*x^2 + 11*x - 6`
- > `collect(h) => ans = x^3 - 6*x^2 + 11*x - 6`
- > `f = (1 + x)*t + x*t`
- > `collect(f) => ans = 2*x*t + t`
- > `collect(f,t) => ans = 2*x*t + t`

● **expand(f) : phân tích biểu thức f.**

● **Ví dụ**

- > `syms x y a b`
- > `f = a*(x + y)`
- > `expand(f) => ans = a*x + a*y`
- > `g = (x - 1)*(x - 2)*(x - 3)`
- > `expand(g) => ans = x^3 - 6*x^2 + 11*x - 6`
- > `h = exp(a + b)`
- > `expand(h) => ans = exp(a)*exp(b)`
- > `cos(3*x) => ans = 4*cos(x)^3 - 3*cos(x)`

⊙ **factor(f)** : phân tích đa thức f thành nhân tử chung

⊙ **Ví dụ**

>  $f = x^3 - 6x^2 + 11x - 6$

>  $g = x^3 - 6x^2 + 11x - 5$

>  $h = x^6 + 1$

> factor(f)

> ans =  $(x - 1)(x - 2)(x - 3)$

> factor(g)

> ans =  $x^3 - 6x^2 + 11x - 5$  ??

> factor(h)

> ans =  $(x^2 + 1)(x^4 - x^2 + 1)$

◎ **simplify(f): đơn giản biểu thức f.**

◎ **Ví dụ**

- >  $f = x*(x*(x - 6) + 11) - 6$
- >  $\text{simplify}(f) \Rightarrow \text{ans} = x^3 - 6*x^2 + 11*x - 6$
- >  $g = (1 - x^2)/(1 - x)$
- >  $\text{simplify}(g) \Rightarrow \text{ans} = x + 1$
- > `syms x y positive`
- >  $\text{simplify}(\log(x*y)) \Rightarrow \log(x) + \log(y)$
- >  $h = \cos(x)^2 + \sin(x)^2$
- >  $\text{simplify}(h) \Rightarrow \text{ans} = 1$

● **simple(f)**: rút gọn biểu thức f, kết hợp các phép toán của simplify, collect, factor.

● **Ví dụ**

>  $f = (1/a^3 + 6/a^2 + 12/a + 8)^{1/3}$

>  $\text{simplify}(f) \Rightarrow \text{ans} = ((2*a + 1)^3/a^3)^{1/3}$

>  $\text{simple}(f) \Rightarrow \text{ans} = (2*a + 1)/a$

>  $\text{syms } x \ y \text{ positive}$

>  $h = \log(x*y)$

>  $\text{simplify}(h) \Rightarrow \text{ans} = \log(x) + \log(y)$

>  $\text{simple}(h) \Rightarrow \text{ans} = \log(x*y)$

⊙ **poly2sym(a,x)**: tạo một đa thức theo biến x với các hệ số được lấy lần lượt từ mảng a.

⊙ **Ví dụ**

- > syms x; a = [1 4 -7 -10]
- > p = poly2sym(a,x)
- > p =  $x^3 + 4x^2 - 7x - 10$

⊙ **x = sym2poly(p)**: trích các hệ số của đa thức p chứa vào mảng s.

⊙ **Ví dụ**

- > syms x; p =  $4s^2 - 2s^2 + 5s - 16$
- > x = sym2poly(p)
- > x = 4 -2 5 -16



# 7. Ứng dụng Matlab

## 7.1 Tính toán trong đại số tuyến tính

### ⦿ Khai báo ma trận

- > `syms a b c d t`
- > `A = [a b; c d]`
- > `B = [cos(t) sin(t); -sin(t) cos(t)]`
- > `C = [t 1 0; 1 t 1; 0 1 t]`
- > `d = round(rand(3,3))`
- > `D = sym(D)`

## ◎ Các phép toán: với 2 ma trận A và B

- $A + B$
- $A - B$
- $A * B$
- $A \setminus B$  (  $= A * \text{inv}(B)$  )
- $A / B$  (  $= \text{inv}(A) * B$  )
- $A^n$
- $A.'$

## ◎ Các hàm xử lý ma trận:

- $\text{inv}(A)$
- $\text{det}(A)$
- $\text{rank}(A)$
- $\text{diag}(A)$
- $\text{tril}(A)$
- $\text{triu}(A)$

## ◉ Ví dụ

```
> c = floor(10*rand(4))  
> D = sym(c)  
> A = inv(D)  
> inv(A)*A  
> det(A)  
> b = ones(1,4)  
> x = b/A  
> x*A  
> A^3
```

## 7.2 Giải phương trình đại số

⊙ **solve(f)** : giải phương trình  $f(x) = 0$ .

⊙ **Ví dụ**

> `syms a b c x`

> `f = a*x^2 + b*x + c;`

> `solve(f)`

> `ans =`

$\left[ \frac{1}{2}a(-b + (b^2 - 4ac)^{1/2}) \right]$

$\left[ \frac{1}{2}a(-b - (b^2 - 4ac)^{1/2}) \right]$

◎ **solve(f)** : giải phương trình theo biến mặc định được chỉ ra trong hàm findsym(f), ở đây findsym(f) -> ans = x. solve(f,a): giải theo biến được chỉ định là a (tương tự cho b, c).

◎ **Ví dụ**

> solve(f,b)

> ans =  $-(a*x^2 + c)/x$

◎ **solve(' f(x) = g(x) ')**: giải phương trình  $f(x) = g(x)$ . Lưu ý: phải đặt trong dấu nháy.

## ◉ Ví dụ

```
> s = solve('cos(2*x) + sin(x) = 1')
```

```
> s =
```

```
[      0]
```

```
[      pi]
```

```
[ 1/6*pi]
```

```
[ 5/6*pi]
```

◉ `solve('f(x)', 'g(x)', 'h(x)', ...)`: giải hệ nhiều phương trình.



## ● Ví dụ

Giải hệ: 
$$\begin{cases} x^2 y^2 = 0 \\ x - y/2 = \alpha \end{cases}$$

> `syms x y alpha`

> `[x y] = solve('x^2*y^2=0','x - y/2 = alpha')`

`x =`

`[ 0]`

`[ 0]`

`[ alpha]`

`[ alpha]`

`y =`

`[-2*alpha]`

`[-2*alpha]`

`[ 0]`

`[ 0]`

- Nghiệm: `v = [x, y]`

● Giải hệ: 
$$\begin{cases} u^2 + v^2 = a^2 \\ u + v = 1 \\ a^2 - 2a = 3 \end{cases}$$

> S = solve('u^2+v^2=a^2','u+v=1','a^2-2\*a=3')

> S =

a: [2x1 sym]

u: [2x1 sym]

v: [2x1 sym]

> S.a

ans =

[ 3]

[ -1]

## 7.3 Giải phương trình đạo hàm riêng

- **Hàm: dsolve**

- **Ví dụ**

- **Giải:**  $\frac{dy}{dt} = 1 + y^2, y(0) = 1$

- > `dsolve('Dy=1+y^2','y(0)=1')`

- > `y = tan(t + 1/4*pi)`

- **Giải:**  $\frac{d^2y}{dx^2} = \cos(2x) - y, y(0)=1, \frac{d}{dx}y(0) = 0$

- > `y = dsolve('D2y=cos(2*x) - y','y(0)=1','Dy(0)=0','x')`

- > `simplify(y); ans = 4/3*cos(x) - 2/3*cos(x)^2+1/3`

◉ Giải: 
$$\begin{cases} \frac{d^3 u}{dx^3} = u \\ u(0) = 1; u'(0) = -1; u''(0) = \pi \end{cases}$$

> `dsolve('D3u=u','u(0)=1','Du(0)=-1','D2u(0)=pi'),'x')`

◉ Giải: 
$$\begin{cases} \frac{df}{dt} = 3f(t) + 4g(t) & , f(0) = 0 \\ \frac{dg}{dt} = -4f(t) + 3g(t) & , g(0) = 1 \end{cases}$$

> `[f g] = dsolve('Df = 3*f + 4*g','Dg = -4*f + 3*g',...  
'f(0) = 0','g(0) = 1')`

> `f = exp(3*t)*sin(4*t); g = exp(3*t)*cost(4*t)`