

CÁC PHÉP TOÁN CƠ BẢN CỦA MATLAB

Nhóm: **TERRI-MATH**

I. Các lệnh cơ bản nhất:

- Xóa màn hình:

>> *clc*

II. Các phép toán số thông thường:

1) Chia trái, chia phải:

>> *a/b*>> *%(Chia phải) nghĩa là $\frac{a}{b}$* >> *%(Cách viết khác là: `mrdivide(a,b)`*>> *a\b*>> *%(Chia trái) nghĩa là $\frac{b}{a}$* >> *%(Cách viết khác là: `mldivide(a,b)`*

2) Dấu:

>> *sign(0.2) = 1*>> *sign(-0.2) = -1*>> *sign(0) = 0*

3) Trị tuyệt đối:

>> *abs(-5) = 5*>> *abs(5) = 5*

4) Hiển thị dạng phân số:

>> *rats(0.2) = 1/5*

5) Hiển thị dạng:

$$d_1 + \frac{1}{d_2 + \frac{1}{d_3 + \frac{1}{\dots}}}$$

>> *rat(10/7) = 1 + 1/(2 + 1/3)*

6) Xấp xỉ (làm tròn):

a. Bình thường:

>> *round(1.4) = 1*>> *round(1.5) = 2*>> *round(1.6) = 2*>> *round(-1.4) = -1*>> *round(-1.5) = -2*>> *round(-1.6) = -2*

b. Theo chiều âm vô cùng (làm tròn xuống):

>> *floor(1.1) = 1*>> *floor(1.9) = 1*>> *floor(-1.1) = -2*>> *floor(-1.9) = -2*

c. Theo chiều dương vô cùng (làm tròn lên):

>> *ceil(1.1) = 2*>> *ceil(1.9) = 2*>> *ceil(-1.1) = -1*>> *ceil(-1.9) = -1*

d. Theo chiều về không (làm tròn lên với số âm, làm tròn xuống với số dương):

>> *fix(1.1) = 1*>> *fix(1.9) = 1*>> *fix(-1.1) = -1*>> *fix(-1.9) = -1*

7) Lấy phần dư:

a. >> *mod(a,b)*

b. \gg *idivide*

c. \gg *rem*

8) Số nguyên tố, BCNN, UCLN:

a. Kiểm tra số n có là số nguyên tố hay không?

\gg *isprime(n)*

b. Liệt kê tất cả các số nguyên tố bé hơn hay bằng n :

\gg *primes(n)*

c. Phân tích số nguyên dương n thành tích các thừa số nguyên tố:

\gg *factor(n)*

d. Ước chung lớn nhất của hai số nguyên dương a và b : (Greatest common divisor)

\gg *gcd(a, b)*

e. Bội chung nhỏ nhất của hai số nguyên dương a và b : (Least common multiples)

\gg *lcm(a, b)*

9) Hoán vị, chỉnh hợp, tổ hợp:

a. Hoán vị:

- Tính $n!$:

\gg *factorial(n)*

hoặc

\gg *prod(1:n)*

- Liệt kê tất cả các hoán vị của tập A :

\gg *perms(A)*

b. Chỉnh hợp:

c. Tổ hợp:

- Tính C_n^k :

\gg *nchoosek(n, k)*

- Liệt kê tất cả các tập con có k phần tử lấy từ tập hợp A :

\gg *nchoosek(A, k)*

III. Phép toán lượng giác:

- Tính hàm *sin*, *cos*, *tan* bình thường:

\gg *sin(x)*

\gg *cos(x)*

\gg *tan(x)*

- Tính hàm ngược *arcsin*, *arccos*, *arctan*:

\gg *asin(x)*

\gg *acos(x)*

\gg *atan(x)*

IV. Chuyển hệ cơ số:

- Chuyển cơ số n từ nhị phân sang thập phân:

\gg *bin2dec('n')*

- Chuyển cơ số n từ thập phân sang nhị phân:

\gg *dec2bin(n)*

- Chuyển cơ số n từ thập lục phân sang thập phân:

\gg *hex2dec('n')*

- Chuyển cơ số n từ thập phân sang thập lục phân:

`>> dec2hex(n)`

- Chuyển cơ số n từ hệ cơ số k sang thập phân:

`>> base2dec(n, k)`

- Chuyển cơ số n từ thập phân sang hệ cơ số k :

`>> dec2hex(n, k)`

V. Ma trận:

1) Nhập ma trận:

- Nhập thủ công:

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

A =

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

`>> A = [1 2 3`

4 5 6

7 8 9]

A =

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

2) Nhập ma trận đặc biệt:

- Ma trận không:

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

B =

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

`>> B = zeros(2)`

B =

```
0 0
0 0
```

- Ma trận một:

`>> C = ones(2,3)`

C =

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

`>> C = ones(2)`

C =

```
1 1
1 1
```

- Ma trận chéo:

`>> D = diag([7 8 9])`

D =

```
7 0 0
0 8 0
0 0 9
```

- Ma trận đơn vị:

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

E =

```
1 0 0
0 1 0
```

`>> E = eye(2)`

E =

```
1 0
0 1
```

- Ma trận kì ảo: (ma phương)

`>> F = magic(3)`

F =

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

- Ma trận Pascal: (đọc trên đường chéo phụ)

`>> G = pascal(3)`

G =

1 1 1
1 2 3
1 3 6

- Ma trận Hilbert:

`>> H = rats(hild(3))`

`H =`

1 $\frac{1}{2}$ $\frac{1}{3}$
 $\frac{1}{2}$ $\frac{1}{3}$ $\frac{1}{4}$
 $\frac{1}{3}$ $\frac{1}{4}$ $\frac{1}{5}$

- Ma trận ngẫu nhiên: (mỗi hệ số nằm trong đoạn $[0,1]$)

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

`I =`

0.4253 0.1615 0.4229
0.3127 0.1788 0.0942

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

`I =`

0.5985 0.6959 0.6385
0.4709 0.6999 0.0336

- Ma trận ngẫu nhiên: (mỗi hệ số nguyên nằm trong đoạn $[a, b]$)

`J = round(a + rand(m,n) * (b - a))`

`>> % Nhập ma trận ngẫu nhiên cấp 2 x 3, các hệ số nằm trong khoảng - 4 đến 5`

`>> J = round(-4 + rand(2,3) * 9)`

`J =`

3 -3 -3
0 -2 -1

3) Các phép toán cơ bản trên ma trận:

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

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

- Kích thước ma trận: (dòng cột)

`>> size(A)`

`ans =`

3 3

- Phép tổng theo cột:

`>> sum(A)`

`ans =`

12 15 18

- Phép tổng theo hàng:

`>> sum(A')`

`>> % hoặc sum(A, 2) tính tổng theo hướng 2, xuất theo hướng 2`

`>> % hướng 1: ngang, hướng 2: dọc`

`ans =`

6 15 24

- Phép cộng cả ma trận:

`>> sum(sum(A))`

* Phép toán lớn nhất ($\max(A)$), nhỏ nhất ($\min(A)$), trung bình cộng vector ở giữa ($\text{mean}(A)$), cũng tương tự như phép tổng ($\text{sum}(A)$).

- Trung bình cộng các phần tử của ma trận:

`>> sum(sum(A))/prod(size(A))`

- Phép cộng:

`>> %Phép cộng 2 ma trận thông thường`

`>> A + B`

`ans =`

```
2    7   10
6    9   15
10   14   17
```

`>> %Phép cộng mỗi hệ số của ma trận cho một số`

`>> A + 1`

`ans =`

```
2    3    4
5    6    7
8    9   10
```

- Phép trừ:

`>> %Phép trừ 2 ma trận thông thường`

`>> A - B`

`ans =`

```
0   -3   -4
2    1   -3
4    2    1
```

`>> %Phép trừ mỗi hệ số của ma trận cho một số`

`>> A - 1`

`ans =`

```
0    1    2
3    4    5
6    7    8
```

- Phép nhân:

`>> %Phép nhân 2 ma trận thông thường`

`>> A * B`

`ans =`

```
14   31   49
32   76  121
50  121  193
```

`>> %Phép nhân mỗi hệ số của ma trận cho một số`

`>> A * 2`

`ans =`

```
2    4    6
8   10   12
14   16   18
```

`>> %Phép nhân mỗi hệ số của ma trận này với
hệ số tương ứng của một ma trận khác.`

>> $A.*B$

ans =

```
1  10  21
8  20  54
21 48  72
```

- Ma trận nghịch đảo:

>> %Cách đơn giản nhất là B^{-1}

>> $\text{rats}(\text{inv}(B))$

ans =

```
- 2/3  2/33  17/33
1/3  -13/33  5/33
0  3/11  -2/11
```

- Ma trận giả đảo:

>> %sau khi làm xong phải kiểm tra lại

>> $C = [\quad];$

>> $\text{rats}(\text{pinv}(C))$

ans =

```
? ? ?
? ? ?
? ? ?
```

- Ma trận chuyển vị:

>> A'

ans =

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

- Ma trận bậc thang rút gọn:

>> $\text{rref}(A)$

ans =

```
1  0  -1
0  1  2
0  0  0
```

- Hạng:

>> $\text{rank}(A)$

ans =

2

- Số chiều:

>> $\text{ndims}(A)$

ans =

2

- Vết:

>> %Tổng các số trên đường chéo chính

>> $\text{trace}(A)$

```
ans =
    15
```

4) Trích ma trận:

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

- Trích vị trí (dòng i , cột j):

```
>> %Trích dòng 2, cột 3 của ma trận A
>> A(2,3)
ans =
    6
```

- Trích cả dòng:

```
>> %Trích dòng 2 của ma trận A
>> A(2,:)
ans =
    4    5    6
>> %Trích dòng 2, dòng 3 của ma trận A
>> A([2,3],:)
ans =
    4    5    6
    7    8    9
```

- Trích cả cột:

```
>> %Trích cột 2 của ma trận A
>> A(:,2)
ans =
    2
    5
    8
>> %Trích cột 2, cột 3 của ma trận A
>> A(:, [2,3])
ans =
    2    3
    5    6
    8    9
```

- Trích một khối:

```
>> %Trích vị trí (1,2); (1,3); (2,2); (2,3) của ma trận A
>> A([1,2], [2,3])
ans =
    2    3
    5    6
```

5) Các phép biến đổi sơ cấp trên dòng hoặc trên cột:

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

- Biến dòng i thành α lần dòng i :

```
>> A(i,:) =  $\alpha$  * A(i,:)
>> %Nhân dòng 1 cho 2:
```

```
>> A(1,:) = A(1,:) * 2
ans =
    2    4    6
    4    5    6
    7    8    9
```

- Biến dòng i thành dòng i cộng α lần dòng j :

```
>> A(i,:) = A(i,:) + \alpha * A(j,:)
>> %Dòng 2 thành dòng 2 cộng 2 lần dòng 3 của A:
>> A(2,:) = A(2,:) + A(3,:) * 2
ans =
    1    2    3
   18   21   24
    7    8    9
```

- Hoán vị dòng i và dòng j :

```
>> A([1,2,...,i,...,j,...,m],:) = A([1,2,...,j,...,i,...,m],:)
>> %Hoán vị dòng 2 và dòng 3 của A:
>> A = A([1,3,2],:)
ans =
    1    2    3
    7    8    9
    4    5    6
```

* Phép sơ cấp trên cột cũng tương tự.

VI. Giải phương trình ma trận:

$$AX = B$$

- Có 2 cách giải:

```
>> X = A^(-1) * B
>> %hoặc tốc độ tính nhanh hơn:
>> A\B
```

VII. Giới hạn, vi – tích phân:

1) Giới hạn:

```
%Tính  $\lim_{x \rightarrow 5} \frac{1}{x}$ :
>> syms x
>> limit(1/x, x, 5)
ans =
    1/5
```

```
%Tính  $\lim_{x \rightarrow 0^+} \frac{1}{x}$ 
>> syms x
>> limit(1/x, x, 0, 'right')
ans =
    infinity
```

```
%Tính  $\lim_{x \rightarrow 0^-} \frac{1}{x}$ 
>> syms x
>> limit(1/x, x, 0, 'left')
ans =
    -infinity
```



```
%Tính  $\lim_{x \rightarrow +\infty} \frac{1}{x}$ 
>> syms x
>> limit(1/x, x, inf)
ans =
0
```

```
%Tính  $\lim_{x \rightarrow -\infty} \frac{1}{x}$ 
>> syms x
>> limit(1/x, x, -inf)
ans =
0
```

2) Đạo hàm:

```
%Công thức tổng quát:
>> limit((f(x+h) - f(x))/h, h, 0)

%Hoặc:
>> diff(f, x)
```

```
%Tính đạo hàm của sin(x)
>> syms x h
>> limit((sin(x+h) - sin(x))/h, h, 0)
ans =
cos(x)

>> diff(sin(x), x)
ans =
cos(x)
```

```
%Tính đạo hàm của  $x^x$ 
>> syms x h
>> limit(((x+h)^(x+h) - x^x)/h, h, 0)
ans =
x^x.log(x) + x^x

>> diff(x^x, x)
ans =
x^x.(log(x) + 1)
```

```
%Đạo hàm của hàm nhiều biến:
>> F = [f1(x), ..., fn(x)]
>> diff(F, x)
```

3) Khảo sát tính liên tục:

```
%Hàm f gọi là liên tục tại a khi và chỉ khi:
%  $\lim_{x \rightarrow a} f(x) = f(a)$ 
```

```
%Khảo sát tính liên tục của  $\frac{1}{x^3}$  tại 0:
>> limit(1/x^3, x, 0)
ans =
undefined
%Vậy f không liên tục tại 0
```

4) Khảo sát chuỗi hội tụ:

%Cú pháp chung:

>> `symsum(S(n), n, a, b)`

%Tính tổng 20 phần tử đầu tiên của dãy $\frac{1}{n}$.

%Cụ thể là tính: $\sum_{n=1}^{20} \frac{1}{n}$

>> `syms n`

>> `symsum($\frac{1}{n}$, n, 1, 20)`

`ans =`

$$\frac{55835135}{15519504}$$

%Xét tính hội tụ của chuỗi $\sum_{n=1}^{\infty} \frac{1}{n^2}$ bằng đồ thị, và tính toán

>> `syms n`

>> `symsum($\frac{1}{n^2}$, n, 1, inf)`

`ans =`

$$\frac{1}{6} \cdot \pi^2$$

>> `N = 1000;`

>> `for i = 1:N`

`X(i) = $\frac{1}{i^2}$;`

`end`

>> `plot(X)`

5) Tích phân:

%Tính nguyên hàm (không phải lúc nào cũng có kết quả)

%Ví dụ, tìm nguyên hàm của $\sin(x)$: $\int \sin(x) dx$

>> `int(sin(x), x)`

`ans =`

$$-\cos(x)$$

%Tính tích phân xác định. Thường là ra, nhưng cũng không chắc.

%Ví dụ, tính tích phân từ 0 đến $\frac{\pi}{2}$ của $\sin(x)$: $\int_0^{\frac{\pi}{2}} \sin(x) dx$

>> `int(sin(x), x, 0, 90)`

`ans =`

$$1 - \cos(90)$$

VẼ ĐỒ THỊ TRONG MATLAB

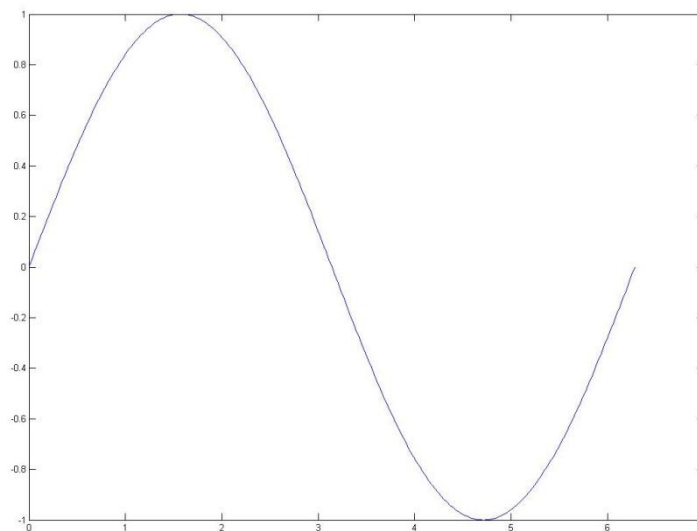
I. Không gian hai chiều:

1) Lệnh cơ bản:

- Hàm **plot** có ba dạng khác nhau tùy thuộc vào đối số đưa vào. Ví dụ nếu y là một vec tơ thì **plot(y)** tạo ra một đường thẳng quan hệ giữa các giá trị của y và hình vẽ của nó. Nếu ta có hai vec tơ x, y thì **plot(x, y)** tạo ra đồ thị quan hệ giữa x và y . Nếu ta có f là một hàm số phụ thuộc vào x thì **plot(x, f(x))** sẽ tạo thành một đồ thị của f theo x .

%Vẽ đồ thị: $y = \sin(x)$ từ $[0, 2\pi]$, mỗi điểm cách nhau $\frac{\pi}{100}$ (trên trục Ox)

```
>> x = 0: pi/100: 2 * pi;
y = sin(x);
plot(x, y);
```



2) Các tùy chỉnh nét vẽ, dấu, màu sắc:

- Lệnh:

```
>> plot(x, f(x), 'Nét vẽ – Dấu – Màu sắc')
```

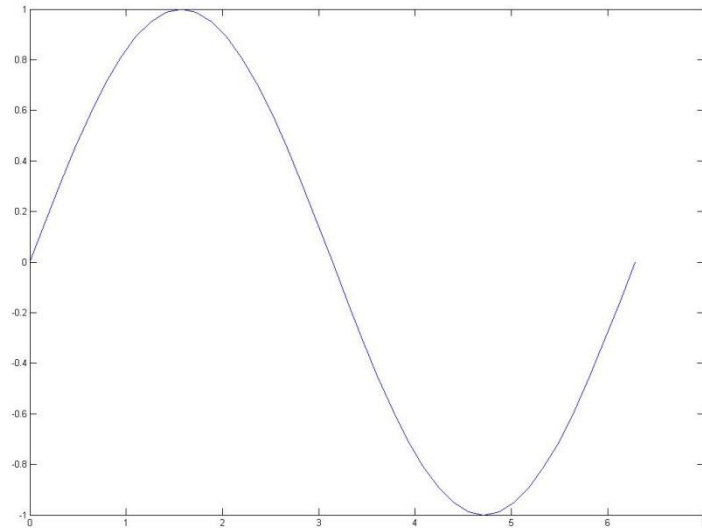
- Nét vẽ:

LineWidth: độ rộng của nét vẽ, tính bằng point (pt).

"_"	_____
"_ _"	-----
". ."
"_ ."	-. -.-

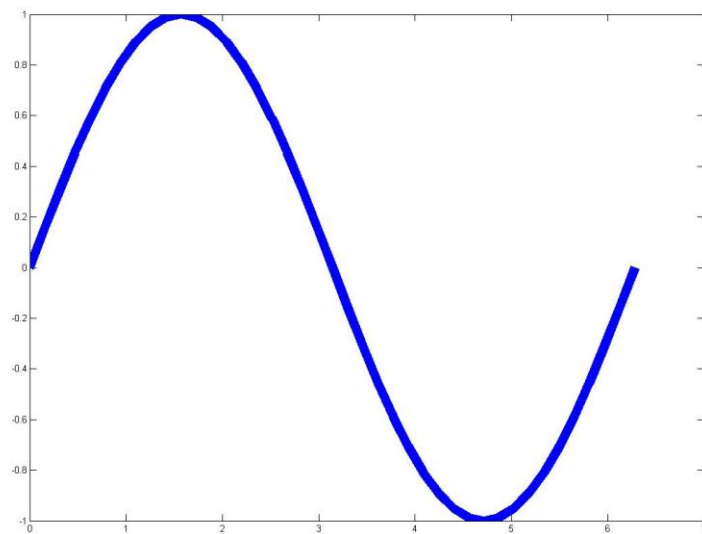
%Vẽ đồ thị: $y = \sin(x)$ từ $[0, 2\pi]$. Với nét vẽ liền.

```
>> x = 0: pi/20: 2 * pi;
y = sin(x);
plot(x, y, '-');
```



%Vẽ đồ thị: $y = \sin(x)$ từ $[0, 2\pi]$. Với nét vẽ liền. Dày 10 pt.

```
>> x = 0: pi/20: 2 * pi;
y = sin(x);
plot(x,y,'-', 'LineWidth', 10);
```



- Dấu (marker):

MarkerFacecolor: màu bên trong marker.

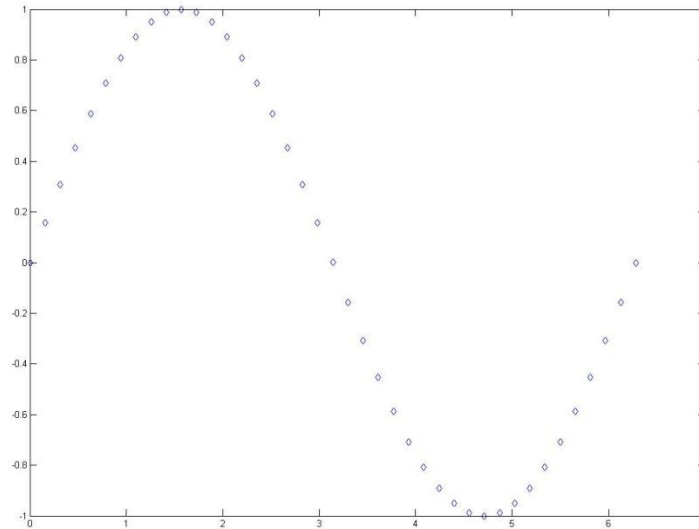
MarkerEdgecolor: màu của đường viền marker.

Markersize: độ lớn của marker, tính bằng pt.

"+"	+	"p" / "pentagram"	★
"o"	●	"h" / "hexagram"	☆
"*"	*	^	▲
"."	.	v	▼
"s" / "square"	□	>	►
"d" / "diamond"	◆	<	◄

%Vẽ đồ thị: $y = \sin(x)$ từ $[0, 2\pi]$. Chấm điểm hình kim cương.

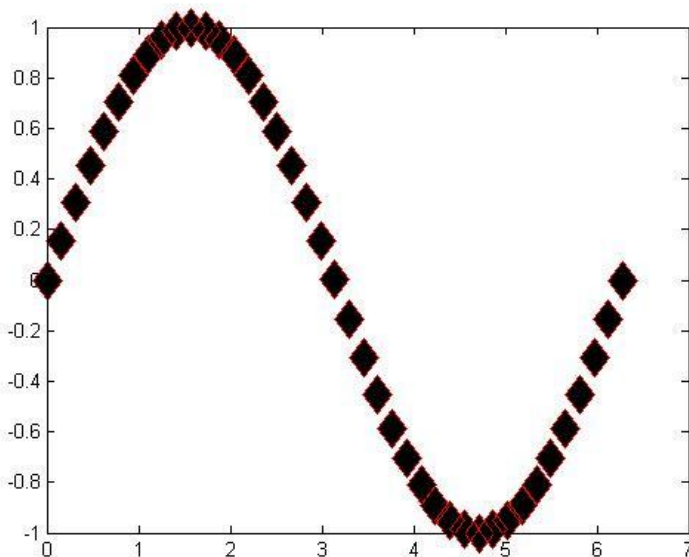
```
>> x = 0: pi/20: 2 * pi;
y = sin(x);
plot(x,y,'d');
```



%Vẽ đồ thị: $y = \sin(x)$ từ $[0, 2\pi]$. Chấm điểm hình kim cương.

%Viên kim cương đen, viền đỏ, to 15pt.

```
>> x = 0: pi/20: 2 * pi;
y = sin(x);
plot(x,y,'d','MarkerFaceColor','k','MarkerEdgeColor','r','MarkerSize',15);
```



- Màu sắc:

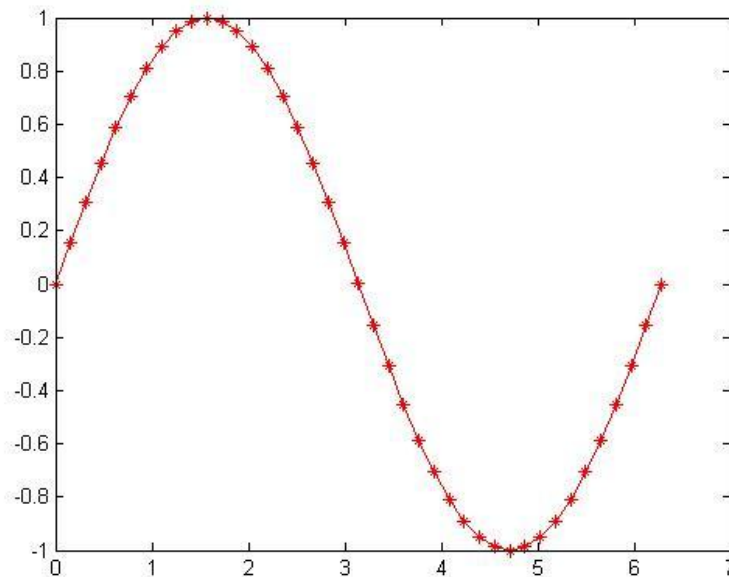
'r' - red	
'k' - black	
'w' - white	
'y' - yellow	

'c' - cyan	
'b' - blue	
'g' - green	
'm' - magenta	

- Ví dụ tổng hợp:

%Vẽ đồ thị: $y = \sin(x)$ từ $[0, 2\pi]$. Với nét vẽ liền, dấu *, màu đỏ.

```
>> x = 0: pi/20: 2 * pi;
y = sin(x);
plot(x,y,'-* r');
```



3) Vẽ nhiều hàm số, đồ thị trong cùng 1 cửa sổ:

- Vấn đề 1: vẽ nhiều hàm số trên cùng một hệ trục tọa độ.

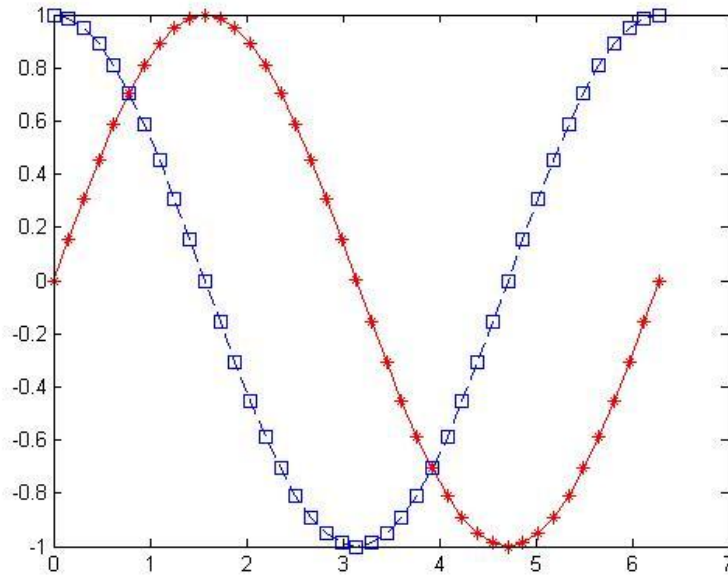
Vẽ đồ thị thứ nhất + **hold on**.

Vẽ tiếp đồ thị thứ hai + **hold off**.

>> %Vẽ đồ thị: $y = \sin(x)$ từ $[0, 2\pi]$. Với nét vẽ liền, dấu *, màu đỏ.

>> %Vẽ cùng đồ thị $y = \cos(x)$ từ $[0, 2\pi]$. Với nét vẽ đứt, dấu □, màu xanh dương.

```
>> x = 0: pi/20: 2 * pi;
y = sin(x);
plot(x,y,'-* r');
hold on;
x = 0: pi/20: 2 * pi;
y = cos(x);
plot(x,y,'-sb');
hold off;
```

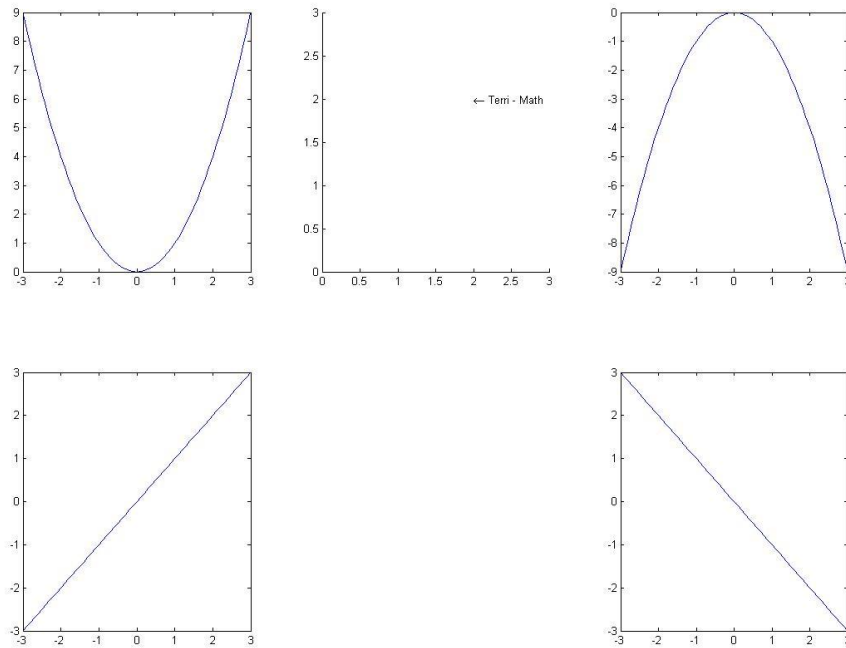


- Vấn đề 2: vẽ nhiều hệ trục tọa độ trong một cửa sổ.

Tạo ra 1 ma trận m hàng, n cột chứa $m \times n$ đồ thị, p là vị trí của từng đồ thị, thứ tự từ trên xuống dưới theo hàng.

`>> subplot(m,n,p)`

```
>> %Tạo ra 2 dòng, 3 cột vị trí.
%Vẽ parabola (P)  $y = x^2$ , ở vị trí 1.
x = -3:0.1:3;
y = x.^2;
subplot(2,3,1);
plot(x,y);
%Chú thích chữ Terri – Math, ở vị trí 2.
subplot(2,3,2);
text(2,2,'\leftarrow Terri – Math');
axis([0 3 0 3]);
%Vẽ parabola (P)  $y = -x^2$ , ở vị trí 3.
subplot(2,3,3);
plot(x,-x.^2);
%Vẽ đường thẳng (d)  $y = x$ , ở vị trí 4.
subplot(2,3,4);
plot(x,x);
%Vẽ đường thẳng (d)  $y = -x$ , ở vị trí 6.
subplot(2,3,6);
plot(x,-x);
```

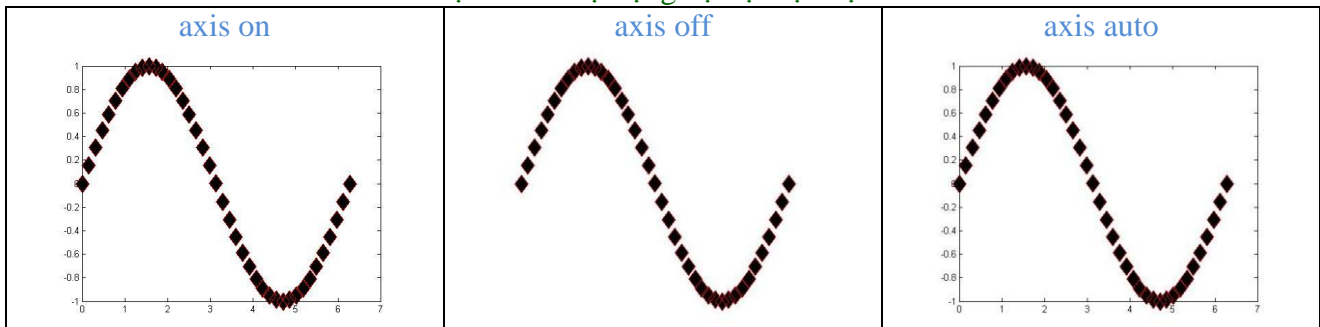


4) Xác định tọa độ:

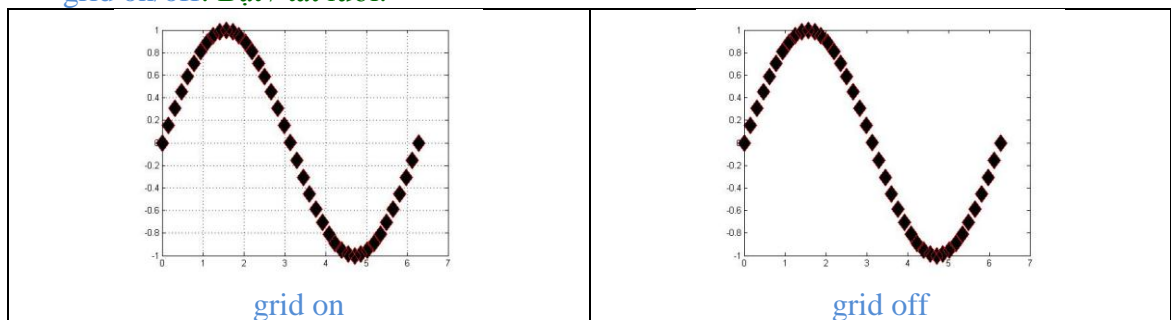
- `axis([xmin xmax ymin ymax])`: Mặc định $xmin = ymin = 0$. Tuy nhiên, đó không phải là điều bắt buộc.
- `xlim([xmin xmax])`: Bạn muốn trục Ox của bạn dài ra hay ngắn lại?
- `ylim([ymin ymax])`: Bạn muốn trục Oy của bạn dài ra hay ngắn lại?

5) Tùy chỉnh các kiểu trục tọa độ:

- `axis on/off/auto`: Bật / Tắt / Tự động hệ trục tọa độ.



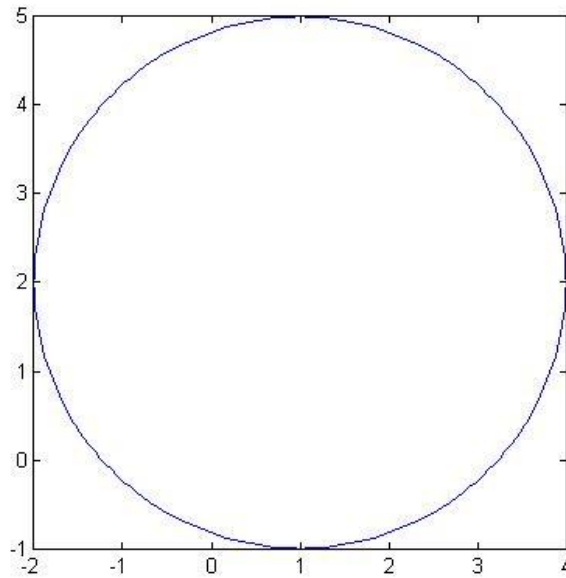
- `grid on/off`: Bật / tắt lưới.



- `axis normal/square/equal/tight`: hệ trục tọa độ bình thường, hình vuông, như nhau, hẹp.

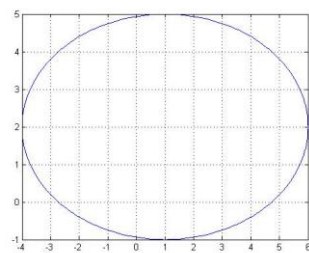
>> %Vẽ đường tròn tâm I(1;2) bán kính 3:


```
>> t = 0:  $\frac{\pi}{100}$ : 2 * pi;
x = 1 + 3 * sin(t);
y = 2 + 3 * cos(t);
plot(x,y);
```

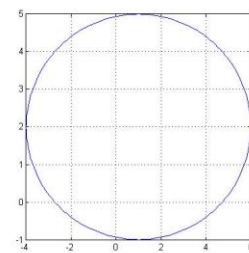


>> %Vẽ đường ellipse tâm I(1; 2) có độ dài trục dài 5, trục ngắn 3:

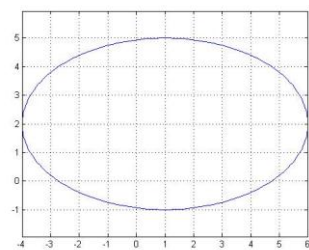
```
>> t = 0:  $\frac{\pi}{100}$ : 2 * pi;
x = 1 + 5 * sin(t);
y = 2 + 3 * cos(t);
plot(x,y);
grid on;
axis tight;
```



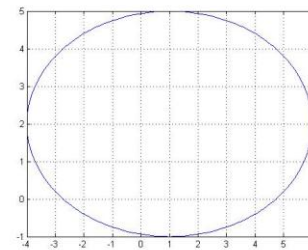
normal



square



equal



tight

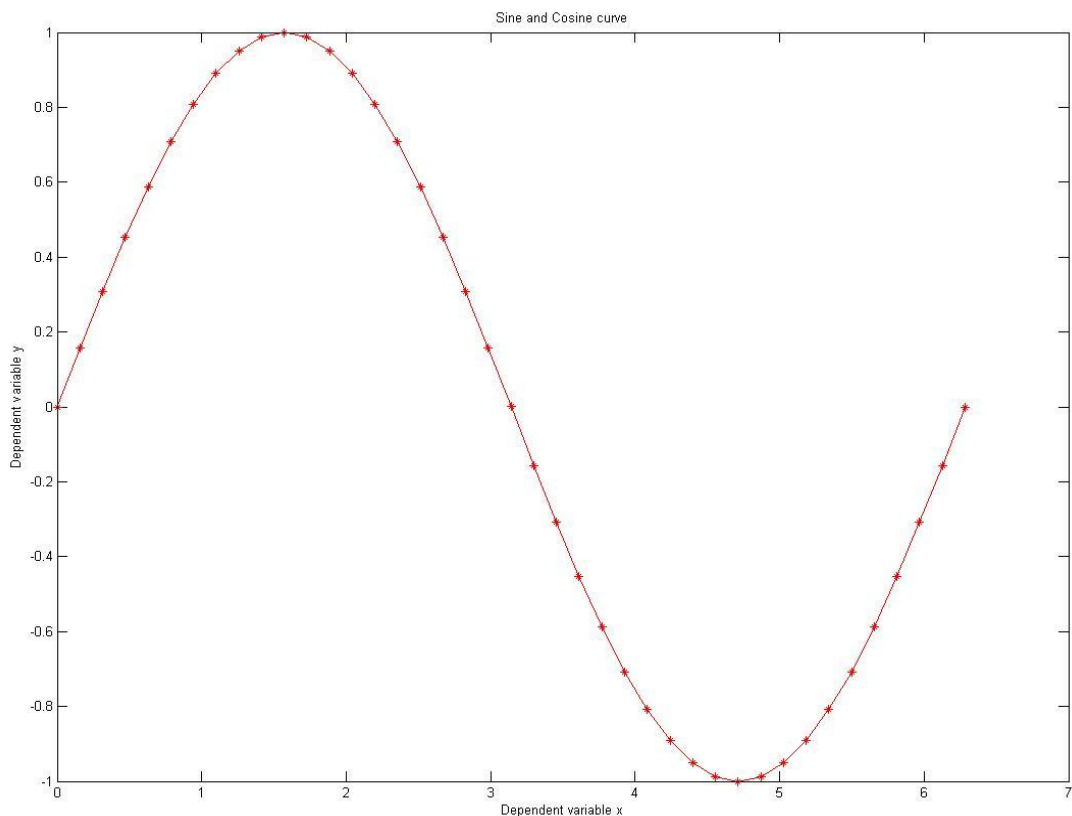
- axis ij/xy:

6) Chú thích trên đồ thị:

- `title('string')`: Tiêu đề, nằm trên cùng của đồ thị.
- `xlabel('string')`: Nhãn cho trục Ox. Nằm bên cạnh trục Ox (dù có xoay đồ thị).
- `ylabel('string')`: Nhãn cho trục Oy. Nằm bên cạnh trục Oy (dù có xoay đồ thị).
- `zlabel('string')`: Nhãn cho trục Oz. Nằm bên cạnh trục Oz (dù có xoay đồ thị).
- `gtext('string')`: Đặt chú thích lên đồ thị, vị trí được xác định bởi click chuột.
- `text(x,y, 'string')`: Đặt một chú thích (trong dấu ') lên đồ thị tại tọa độ (x,y).
- `legend`: Thêm chú giải vào đồ thị.
- `\bf`: bold font.
- `\it`: italics font.
- `\sl`: oblique font.

>> Ví dụ cho việc đặt các nhãn (label):

```
>> x = 0:  $\frac{\pi}{20}$ : 2 * pi;
y = sin(x);
plot(x,y,'-r');
title('Sine and Cosine curve');
xlabel('Dependent variable x');
ylabel('Dependent variable y');
```



II. Không gian 3 chiều:

1) Lệnh:

>> plot3(x,y,z)

Ta cần xác định các vector x, y, z . Để vẽ mặt $(x, y, z = f(x, y))$, lệnh

`>> meshgrid(x, y)`

sẽ tạo mảng giá trị X, Y từ miền giá trị của x, y .

2) Một số lệnh vẽ đồ thị trong không gian ba chiều:

- Các hàm contour tạo, hiển thị và ghi chú các đường đẳng mức của một hay nhiều ma trận. Bao gồm:
 - ✓ `clabel`: tạo các nhãn sử dụng ma trận contour và hiển thị nhãn
 - ✓ `contour`: hiển thị các đường đẳng mức tạo bởi một giá trị cho trước của ma trận Z.
 - ✓ `contour3`: hiển thị các mặt đẳng mức tạo bởi một giá trị cho trước của ma trận Z.
 - ✓ `contourf`: hiển thị đồ thị contour 2D và tô màu vùng giữa 2 các đường
 - ✓ `contourc`: hàm cấp thấp để tính ma trận contour
- Hàm `mesh/surf`: dùng để vẽ mặt lưới 3D.
- `meshc`: giống mesh nhưng có thêm đường viền.
- `waterfall`: chỉ vẽ lưới theo một hướng.
- `bar`: hiển thị các cột của ma trận $m \times n$ như là m nhóm, mỗi nhóm có n bar.
- `bar3`: hiển thị các cột của ma trận $m \times n$ như là m nhóm, mỗi nhóm có n bar dạng 3D.
- `barh`: hiển thị các cột của ma trận $m \times n$ như là m nhóm, mỗi nhóm có n bar nằm ngang.
- `bar3h`: hiển thị các cột của ma trận $m \times n$ như là m nhóm, mỗi nhóm có n bar dạng 3D nằm ngang.
- `pie/pie3`: Đồ thị pie hiển thị theo tỉ lệ phần trăm của một phần tử của một vec tơ hay một ma trận so với tổng các phần tử.

LẬP TRÌNH TRONG MATLAB

I. Lệnh `disp` (tương đương lệnh `write` của Pascal, lệnh `printf` của C++):

%Xuất một câu ra ngoài màn hình

`>> disp('comment')`

- Lệnh này, dùng để xuất ra màn hình câu nói nào đó.

- Hàm `disp` chỉ có thể có duy nhất một đối số. (ví dụ: `disp('Terri – math', x)` là hàm sai)

Ví dụ:

`>> disp('Terri – math')`

Terri – math

II. Hàm lựa chọn:

1) Cấu trúc `if – elseif – ... – else – end`:

- Cú pháp bình thường:

`>> if biểu thức điều kiện`

khối các lệnh được thực hiện nếu biểu thức điều kiện là đúng;

`else`

khối các lệnh được thực hiện nếu biểu thức điều kiện là sai;

`end`

- Cú pháp mở rộng:

`>> if biểu thức điều kiện 1`

khối các lệnh được thực hiện nếu biểu thức điều kiện 1 là đúng;

`elseif biểu thức điều kiện 2`

khối các lệnh được thực hiện nếu biểu thức điều kiện 2 là đúng;

`elseif biểu thức điều kiện 3`

khối các lệnh được thực hiện nếu biểu thức điều kiện 3 là đúng;

`...`

`else`

khối các lệnh được thực hiện không có biểu thức điều kiện nào kể trên đúng.

`end`

- Ví dụ:

+ vd1:

%Tìm giá trị lớn nhất giữa 2 số a, b

`>> a = 5; b = 10;`

`if a > b`

disp('Max value is: ');

disp(a);

`else`

disp('Max value is: ');

disp(b);

`end`

%Màn hình sẽ hiện ra:

Max value is:

10

+ vd2:

%Tìm giá trị lớn nhất giữa 2 số a, b

>> a = 5; b = 5;

if a > b

disp('Max value is:');

disp(a);

elseif a == b

disp('Max is min.');

else

disp('Max value is:');

disp(b);

end

%Màn hình sẽ hiện ra:

Max is min.

2) Cấu trúc switch – case – ... – otherwise – end:

- Cú pháp:

>> switch biểu thức điều kiện

case giá trị thử 1

khối lệnh 1;

case {giá trị thử 2, giá trị thử 3, giá trị thử 4}

khối lệnh 2;

otherwise

khối lệnh 3;

end

- Ví dụ:

%Viết chương trình logic, nếu là 1 xuất ra 'Right'

nếu là 0 xuất ra là 'Wrong', nếu là - 1, hoặc 2 xuất ra là 'I am thinking'

các giá trị khác thì xuất ra 'Are you crazy?'

>> a = 2;

switch a

case 1

disp('Right.');

case 0

disp('Wrong.');

case {-1,2}

disp('I am thinking.');

otherwise

disp('Are you crazy?');

end

%Màn hình sẽ hiện ra:

I am thinking.

III. Vòng lặp:

1) Vòng lặp **for**:

- Cú pháp:

```
>> for biến = giá trị bắt đầu: bước nhảy: giá trị kết thúc
khởi các lệnh ...;
end
```

- Ví dụ:

```
%Tính tổng bình phương các số từ 1 đến 10
>> S = 0;
for i = 1:1:10
S = S + i * i;
end
disp(S);
%Màn hình sẽ hiện ra:
385
```

2) Vòng lặp **while**:

- Cú pháp:

```
>> while biểu thức điều kiện
khởi các lệnh ...;
end
```

- Ví dụ:

```
%Tính tổng bình phương các số từ 1 đến 10
>> S = 0; i = 1;
while i ≤ 10
S = S + i * i;
i = i + 1;
end
disp(S);
%Màn hình sẽ hiện ra:
385
```

IV. Tự viết hàm:

- B1: Ta chọn File\New\M-File.

- B2: Lập trình theo cú pháp:

```
function [các giá trị xuất] = Tên hàm(các giá trị nhập)
các lệnh lập trình ...
end
```

- B3: lưu file dưới dạng: Tên hàm.m.

Chú ý phải lưu trong đúng thư mục:

- B4: để sử dụng hàm, ta chỉ việc ghi ra: Tên hàm(các giá trị nhập).