

MATLAB: VẼ ĐỒ THỊ

Mục đích: Cho sinh viên làm quen với một số lệnh đồ họa trong Matlab.

1. Lệnh plot

a) Công dụng:

Phần lớn các câu lệnh để vẽ đồ thị trong mặt phẳng đều là lệnh plot. Lệnh plot vẽ đồ thị của một mảng dữ liệu trong một hệ trục thích hợp và nối các điểm bằng đường thẳng.

b) Cú pháp:

plot(x,y)

plot(x,y,'linetype')

c) Giải thích:

x,y: vẽ giá trị x theo giá trị y.

MATLAB mặc định đường vẽ là đường liền, không đánh dấu, màu xanh da trời. Ta có thể thay đổi kiểu đường vẽ và đánh dấu lên đồ thị bằng cách đưa vào một đối số thứ ba. Các đối số tùy chọn này là một chuỗi ký tự, có thể chứa một hoặc nhiều hơn theo bảng dưới đây.

Nếu một màu, dấu và kiểu đường tất cả đều chứa trong một chuỗi, thì kiểu màu chung cho cả dấu và kiểu nét vẽ. Để khai báo màu khác cho dấu, ta phải vẽ cùng một dữ liệu với các kiểu khai báo chuỗi khác nhau. linetype: kiểu phần tử tạo nên nét vẽ bao gồm 3 thành phần:

Ký tự	Màu
y	Vàng
m	Đỏ tươi
c	Lơ
r	Đỏ
g	Lục
b	Lam
w	Trắng
k	Đen

- Thành phần thứ hai là các ký tự chỉ nét vẽ của đồ thị:

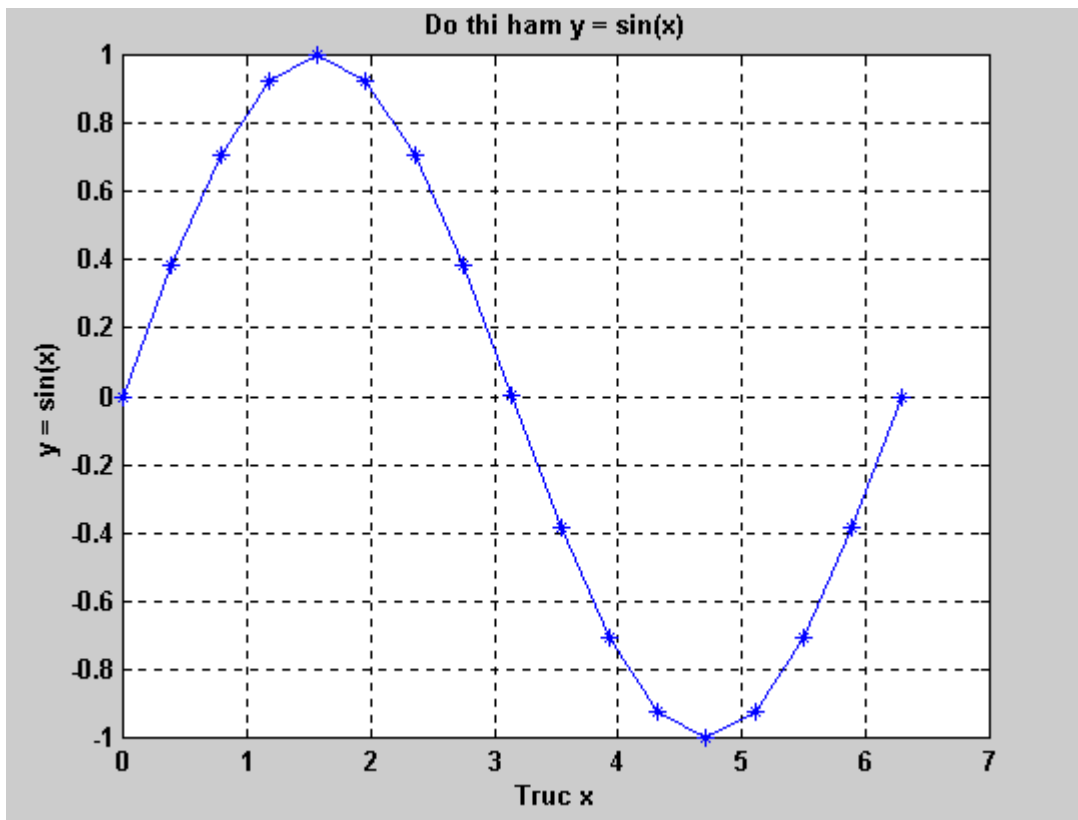
Ký tự	Loại nét vẽ
-	Đường liền nét
:	Đường chấm chấm
-.	Đường gạch chấm
--	Đường nét đứt đoạn

- Thành phần thứ ba là các ký tự chỉ loại điểm đánh dấu gồm:., o, x, +, *

d) Ví dụ:

Vẽ đồ thị hàm $y = \sin(x)$ với đồ thị màu lam, đường liền nét và đánh dấu các điểm được chọn bằng dấu *, trục x thay đổi từ 0 tới 2π , mỗi bước thay đổi là $\pi/8$

```
x = 0:pi/8:2*pi;  
y = sin(x);  
plot(x,y, 'b-* ')  
ylabel('y = sin(x)')  
xlabel('Trục x')  
title('Do thi ham y = sin(x)')  
grid on
```



Trong cửa sổ này nó sẽ tạo ra độ chia phù hợp với dữ liệu, vẽ đồ thị qua các điểm, và đồ thị được tạo thành bởi việc nối các điểm này bằng đường nét liền. Có thể vẽ nhiều hơn một đồ thị trên cùng một hình vẽ bằng cách đưa thêm vào plot một cặp đối số, plot tự động vẽ đồ thị thứ hai bằng màu khác trên màn hình. Nhiều đường cong có thể cùng vẽ một lúc nếu như cung cấp đủ cặp đối số cho lệnh plot.

2. Lệnh grid

a) Công dụng:

Tạo lưới tọa độ.

b) Cú pháp:

grid on

grid off

c) Giải thích:

on: hiển thị lưới tọa độ.

off: không hiển thị lưới tọa độ.

3. Lệnh title

a) Công dụng:

Đặt tiêu đề cho đồ thị.

b) Cú pháp:

title('text')

c) Giải thích:

text: tên tiêu đề.

4. Lệnh xlabel, ylabel, zlabel

a) Công dụng:

Đặt tên cho trục X, Y, Z.

b) Cú pháp:

xlabel('nx')

ylabel('ny')

zlabel('nz')

c) Giải thích:

nx, ny, nz: tên trục x, y, z

5. Lệnh legend

a) Công dụng:

Dòng ghi chú được đưa vào đồ thị nhờ hàm **legend**. Trong legend thì màu và kiểu của mỗi loại đường phù hợp với các đường đó trên đồ thị.

b) Ví dụ

```
x=linspace(0,2*pi,30);
```

```
y=sin(x);
```

```
z=cos(x);
```

```
plot(x,y,'mx-',x,z,'bp--')
```

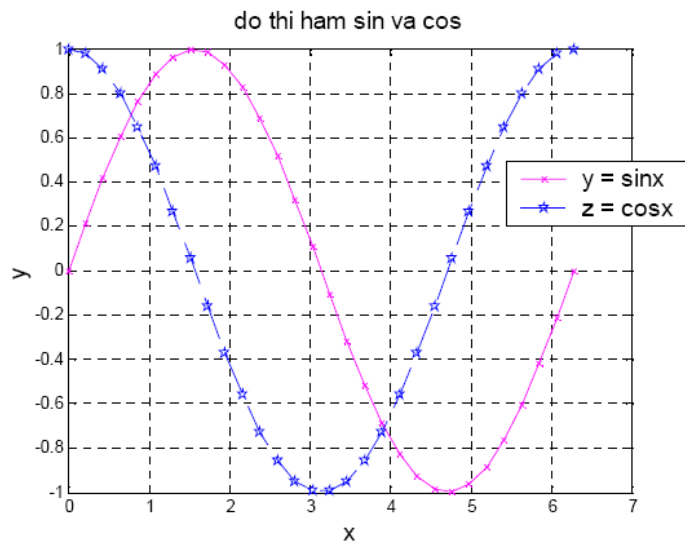
```
grid on
```

```
xlabel('x')
```

```
ylabel('y')
```

```
title('đồ thị hàm sin và cos')
```

```
legend('y = sinx','z = cosx')
```



6. Lệnh subplot

a) Công dụng:

Lệnh **subplot(m,n,p)** chia cửa sổ hiện tại thành một ma trận $m \times n$ khoảng để vẽ đồ thị, và chọn p là cửa sổ hoạt động. Các đồ thị thành phần được đánh số từ trái qua phải, từ trên xuống dưới, sau đó đến hàng thứ hai...

b) Cú pháp:

`subplot(m,n,p)`

c) Giải thích:

`subplot(m,n,p)` chia thành cửa sổ đồ họa thành $m \times n$ vùng để vẽ nhiều đồ thị trên cùng một cửa sổ.

m : số hàng được chia.

n : số cột được chia

p : số thứ tự vùng chọn để vẽ đồ thị.

Nếu khai báo $p > m \times n$ thì sẽ xuất hiện một thông báo lỗi.

d) Ví dụ:

```
x=linspace(0,2*pi,30);
```

```
y=sin(x);
```

```
z=cos(x);
```

```
subplot(2,2,1)
```

```
plot(x,y)
```

```
subplot(2,2,2)
```

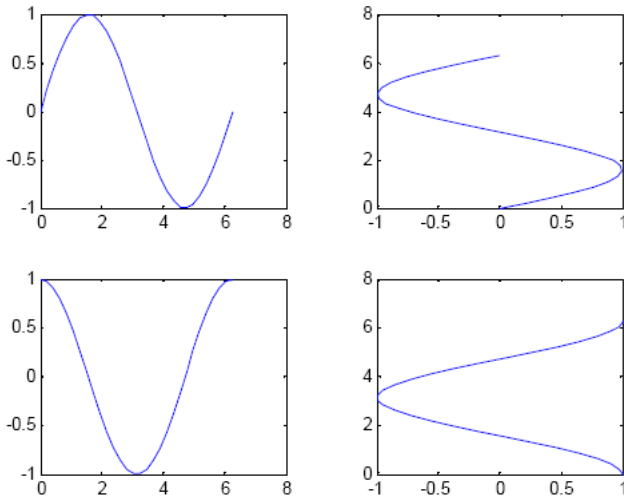
```
plot(y,x)
```

```
subplot(2,2,3)
```

```
plot(x,z)
```

```
subplot(2,2,4)
```

```
plot(z,x)
```



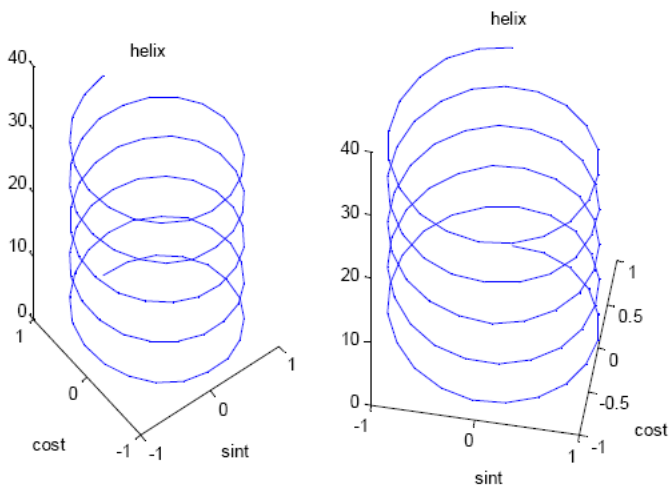
7. Lệnh plot3

a) Công dụng:

Hàm **plot3** cho phép vẽ các điểm và đường trong không gian. Ngoài việc có thêm trục z, cách sử dụng hàm này giống như cách sử dụng hàm plot.

b) Ví dụ:

```
t=linspace(0,10*pi);
subplot(1,2,1)
plot3(sin(t),cos(t),t)
xlabel('sint')
ylabel('cost')
title('helix')
subplot(1,2,2)
plot3(sin(t),cos(t),t)
view([10,35])
xlabel('sint')
ylabel('cost')
title('helix')
```



8. Lệnh view

a) Công dụng:

view($[\alpha, \beta]$): α là góc phương vị tính bằng độ ngược chiều kim đồng hồ từ phía âm của trục y. Giá trị mặc định của α là -37.5° . β là góc nhìn tính bằng độ xuống mặt phẳng x, y. Giá trị mặc định của β là 30° . Khi thay đổi các giá trị α và β sẽ nhìn được hình vẽ dưới các góc độ khác nhau. Với tập hợp lệnh trên, khi cho các giá trị α và β lần lượt là 0° và 90° ta sẽ thấy rõ hàm vẽ 2D là một trường hợp đặc biệt của hàm vẽ 3D.

b) Ví dụ:

Xem ví dụ trong lệnh plot3

9. Lệnh pie

a) Công dụng:

Để vẽ đồ thị bánh trong mặt phẳng ta dùng hàm pie, còn muốn vẽ trong không gian, ta dùng hàm pie3. Về mặt cú pháp hai hàm pie và pie3 giống nhau.

b) Cú pháp:

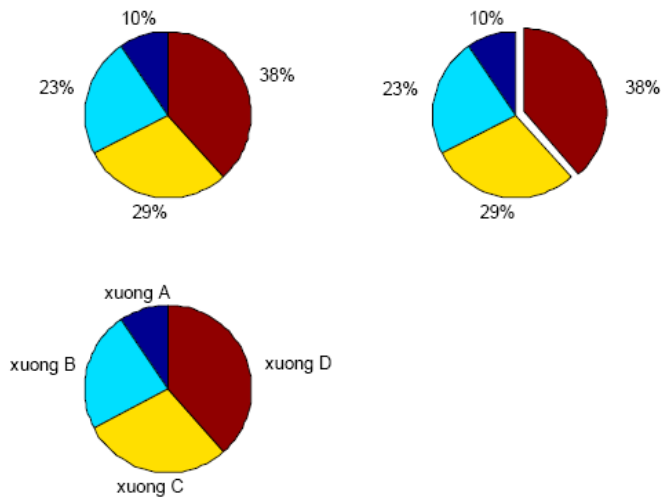
pie(V)

c) Giải thích:

V là vector chứa các phần tử được thể hiện trên đồ thị bánh. Nếu tổng các phần tử trong vector nhỏ hơn hoặc bằng 1 thì đồ thị bánh sẽ thể hiện các phần tử như là thành phần phần trăm. Nếu tổng các phần tử lớn hơn 1, thì mỗi phần tử được chia cho tổng đó để xác định phần chia trên đồ thị bánh ứng với mỗi phần tử. Thứ tự phân chia trên đồ thị bánh theo đúng thứ tự phần tử mô tả trong vector. Đường chia đầu tiên là đường nối tâm và điểm cao nhất trên đường tròn, các đường kế tiếp được phân chia theo thứ tự ngược chiều kim đồng hồ. Muốn tách phần chia nào đó ra khỏi đồ thị thì ta thêm vào hàm pie một vector nữa có cùng kích thước với vector được mô tả ở trên. Phần tử của vector này tương ứng với phần cần tách ra khỏi đồ thị thì ta cho giá trị khác 0, phần tử tương ứng với phần không tách ra ta cho giá trị bằng 0. Các màu của từng phần trong đồ thị bánh được MATLAB lựa chọn không trùng nhau và rất dễ phân biệt.

d) Ví dụ:

```
subplot(2,1,1)
pie([5 12 15 20])
subplot(2,1,2)
pie([5 12 15 20],[0 0 0 1])
pie([5 12 15 20],{'xuống A','xuống B','xuống C','xuống D'})
```



10. Lệnh bar/bar3

a) Công dụng:

Hàm barh và hàm barh3 cho phép vẽ đồ thị cột nằm ngang trong mặt phẳng và trong không gian.

b) Cú pháp:

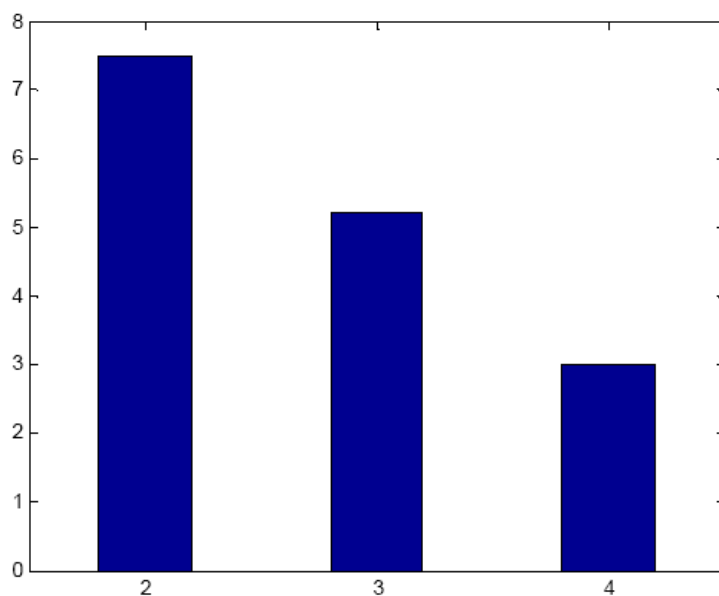
$\text{bar}(V_x, V_y, \text{kích thước})$

c) Giải thích:

V_x và V_y là những vectơ có cùng kích thước, các giá trị độ cao của cột trong V_y sẽ tương ứng với các giá trị trên trục ngang của V_x , điều chú ý quan trọng là các giá trị trong V_x phải đơn điệu tăng hoặc giảm. Tham số kích thước xác định bề rộng của cột.

d) Ví dụ:

$\text{bar}([2\ 3\ 4], [7.5\ 5.2\ 3], 0.4)$



11. Lệnh axes

a) Công dụng:

Đặt các trục tọa độ tại vị trí định trước.

b) Cú pháp:

axes('propertyname', propertyvalue ...)

c) Giải thích:

Tương ứng với một propertyname đi kèm với 1 propertyvalue.

1. 'position', [left, bottom, width, height]: định vị trí và kích thước của trục.

left: khoảng cách từ mép trái cửa sổ đến trục đứng.

bottom: khoảng cách từ mép dưới cửa sổ đến trục ngang.

width: chiều dài của trục ngang.

height: chiều cao trục đứng.

Ghi chú:

Luôn lấy điểm [0,0] làm gốc tọa độ.

Trục ngang và trục đứng có giá trị trong khoảng [0 1] và chia theo tỷ lệ thích hợp

*/ Ví dụ:

axes('position', [.1 .1 .8 .6])

2. 'xlim', [min,max]: định giá trị nhỏ nhất và lớn nhất trên trục x.

*/ Ví dụ:

axes('xlim', [25])

3. 'ylim', [min,max]: định giá trị nhỏ nhất và lớn nhất trên trục y.

*/ Ví dụ:

axes('ylim', [25])

định giá trị trên cả hai trục

axes('xlim', [min,max], 'ylim', [min,max])

12. Lệnh AXIS

a) Công dụng:

Chia lại trục tọa độ.

b) Cú pháp:

axis([xmin xmax ymin ymax])

axis([xmin xmax ymin ymax zmin zmax])

axis on

axis off

c) Giải thích:

xmin, ymin, zmin: là giá trị nhỏ nhất của các trục x, y, z.

xmax, ymax, zmax: là giá trị lớn nhất của các trục x, y, z.

on: cho hiển thị trục tọa độ.

off: không cho hiển thị trục tọa độ.

13. Lệnh bar

a) Công dụng:

Vẽ đồ thị dạng cột.

b) Cú pháp:

`bar(x,y)`

c) Giải thích:

Vẽ giá trị x theo giá trị y.

d) Ví dụ:

`x = -pi:0.2:pi;`

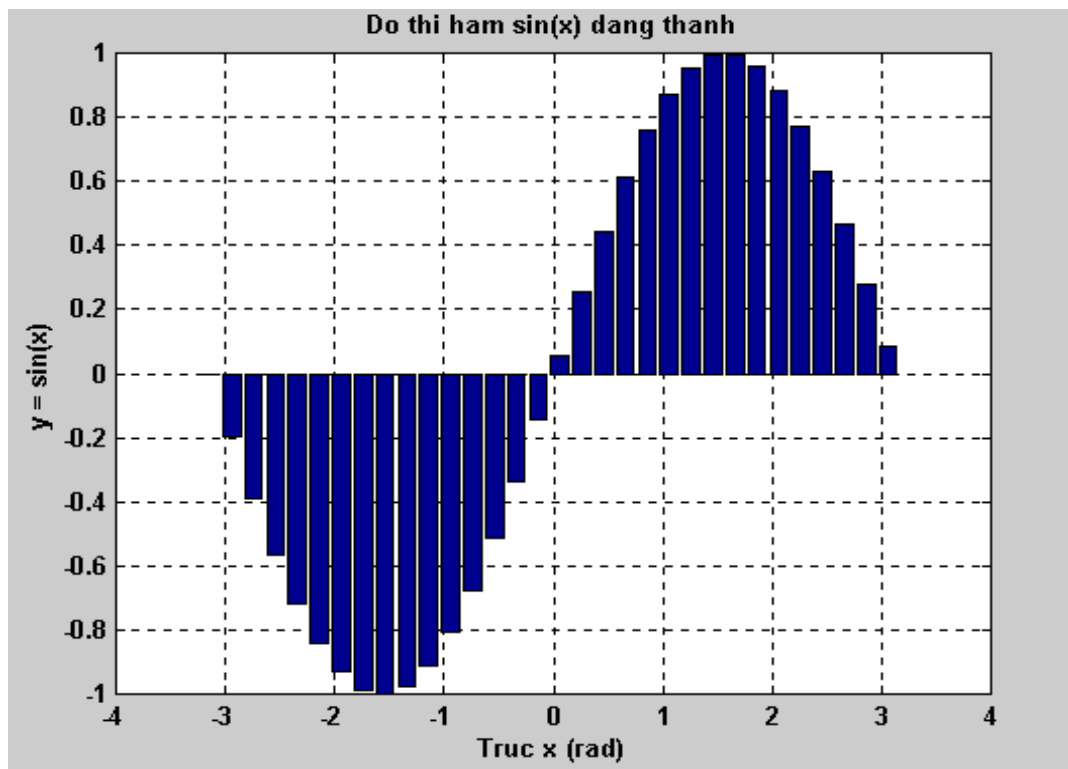
`bar(x,sin(x));`

`grid on`

`title('Do thi ham sin(x) dang thanh')`

`xlabel('truc x (rad)')`

`ylabel('y = sin(x)')`



14. Lệnh cla

a) Công dụng:

Xóa tất cả các đối tượng như: đường đồ thị, tên đồ thị...nhưng không xóa trục tọa độ.

b) Cú pháp:

`cla`

15. Lệnh clf

a) Công dụng:

Xóa hình ảnh (đồ thị) hiện tại.

b) Cú pháp:

clf

16. Lệnh close

a) Công dụng:

Đóng hình ảnh (đồ thị) hiện tại.

b) Cú pháp:

close

17. Lệnh colormap

a) Công dụng:

Tạo màu sắc cho đồ thị trong không gian 3 chiều.

b) Cú pháp:

colormap(map)

colormap('default')

c) Giải thích:

Colormap là sự trộn lẫn của 3 màu cơ bản: red, green, blue. Tùy theo tỷ lệ của 3 màu cơ bản mà cho ra các màu sắc khác nhau.

'default': màu có được là màu mặc định.

map: biến chứa các thông số sau:

Map	màu có được
Bone	gray + blue
Cool	cyan + magenta
Flag	red + white + blue + black
Gray	gray
Hot	black + red + yellow + white
Pink	pink

18. Lệnh figure

a) Công dụng:

Tạo mới hình ảnh (đồ thị).

b) Cú pháp:

figure

19. Lệnh gca

a) Công dụng:

Tạo các đặc tính cho trục.

b) Cú pháp:

h = gca

c) Giải thích:

h: là biến gán cho lệnh gca.

Các đặc tính của trục gồm có:

Cú pháp	Giải thích
Set(gca,'XScale','log', 'Yscale','linear')	Định đơn vị trên trục tọa độ: trục x có đơn vị là log và trục y có đơn vị tuyến tính.
Set(gca,'Xgrid','on','YGrid', 'nomal')	Tạo lưới cho đồ thị: trục x có tạo lưới và trục y không tạo lưới.
Set(gca,'XDir','reverse', 'YDir','normal')	Đổi trục tọa độ: đổi trục x về phía đối diện, trục y giữ nguyên.
Set(gca,'XColor','red', 'Ycolor','yellow')	Đặt màu cho lưới đồ thị: đặt lưới trục x màu đỏ, lưới trục y màu vàng. Gồm có các màu: yellow, magenta, cyan, red, green, blue, white, black.

20. Lệnh semilogx, semilogy

a) Công dụng:

Vẽ đồ thị theo logarith.

b) Cú pháp:

semylogx(x,y)

semylogx(x,y,'linetype')

semylogy(x,y)

semylogy(x,y,'linetype')

c) Giải thích:

semylogx và semylogy giống như lệnh plot nhưng chỉ khác một điều là lệnh này vẽ đồ thị theo trục logarith. Do đó, ta có thể sử dụng tất cả các loại 'linetype' của lệnh plot.

d) Ví dụ:

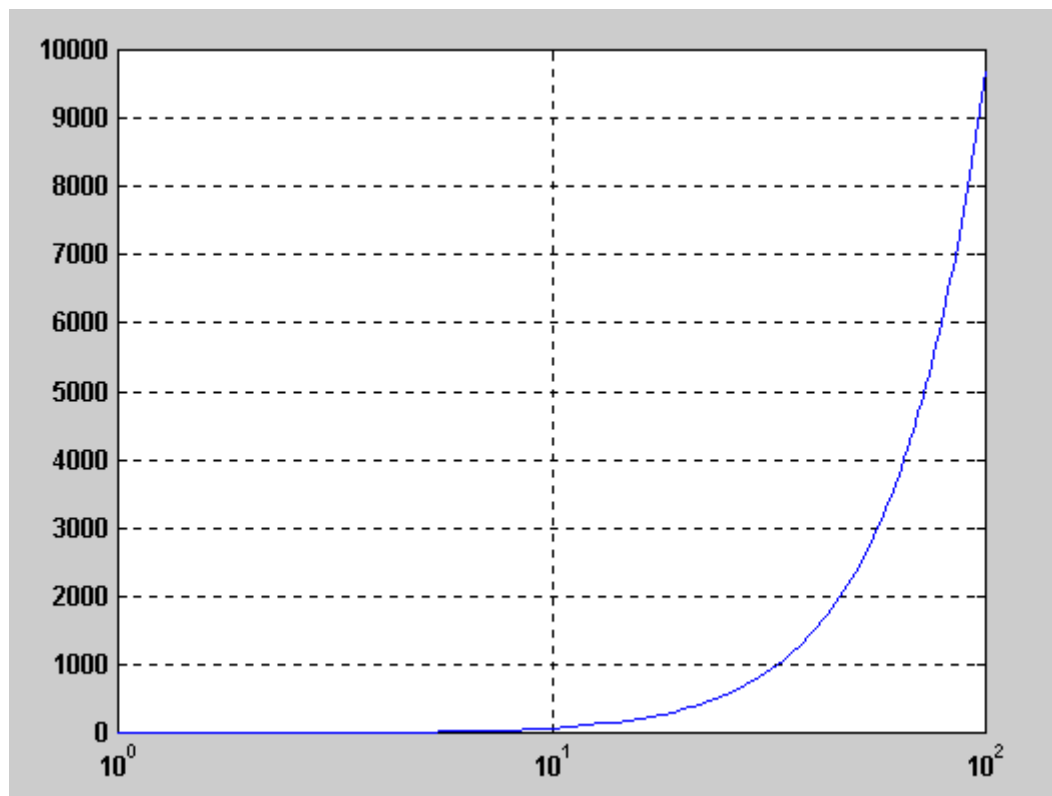
Vẽ đồ thị hàm $y = x^2 - 3x + 2$ theo trục logarith của x.

x = 0:100;

y = x.^2-3*x+2;

semylogx(x,y,'b')

grid on



21. Lệnh polar

a) Công dụng:

Vẽ đồ thị trong hệ trục tọa độ cực.

b) Cú pháp:

polar(theta,rho)

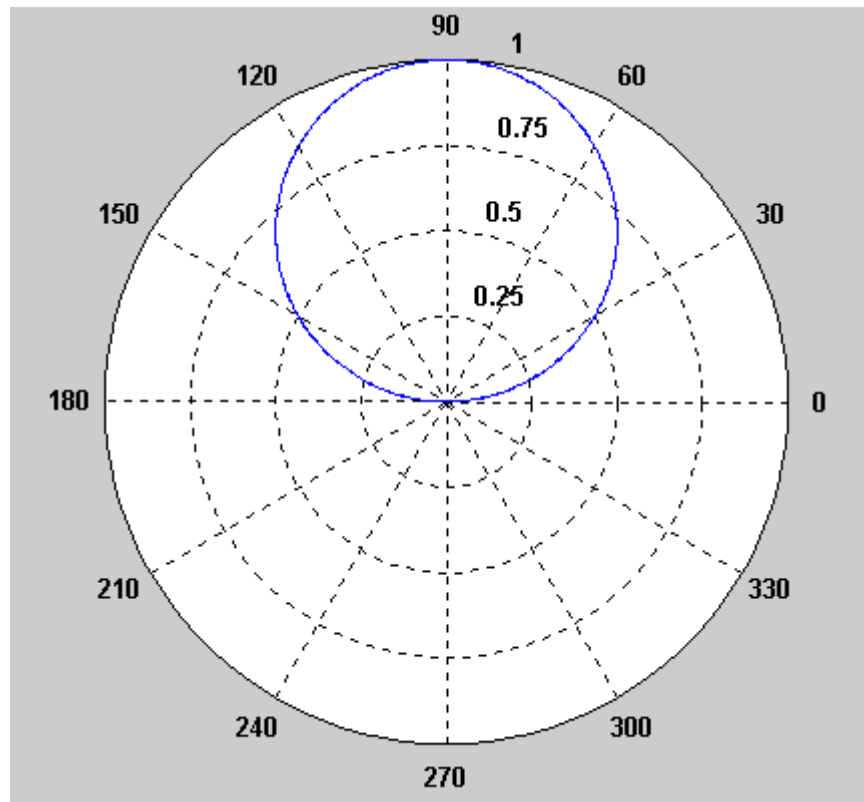
c) Giải thích:

Vẽ giá trị x theo giá trị y.

d) Ví dụ:

t = -pi:0.01:pi;

polar(t, sin(t))



22. Lệnh set

a) Công dụng:

Thiết lập các đặc tính chất cho đối tượng nào đó.

b) Cú pháp:

set(h, 'propertyname', propertyvalue,...)

c) Giải thích:

h: biến chứa đối tượng.

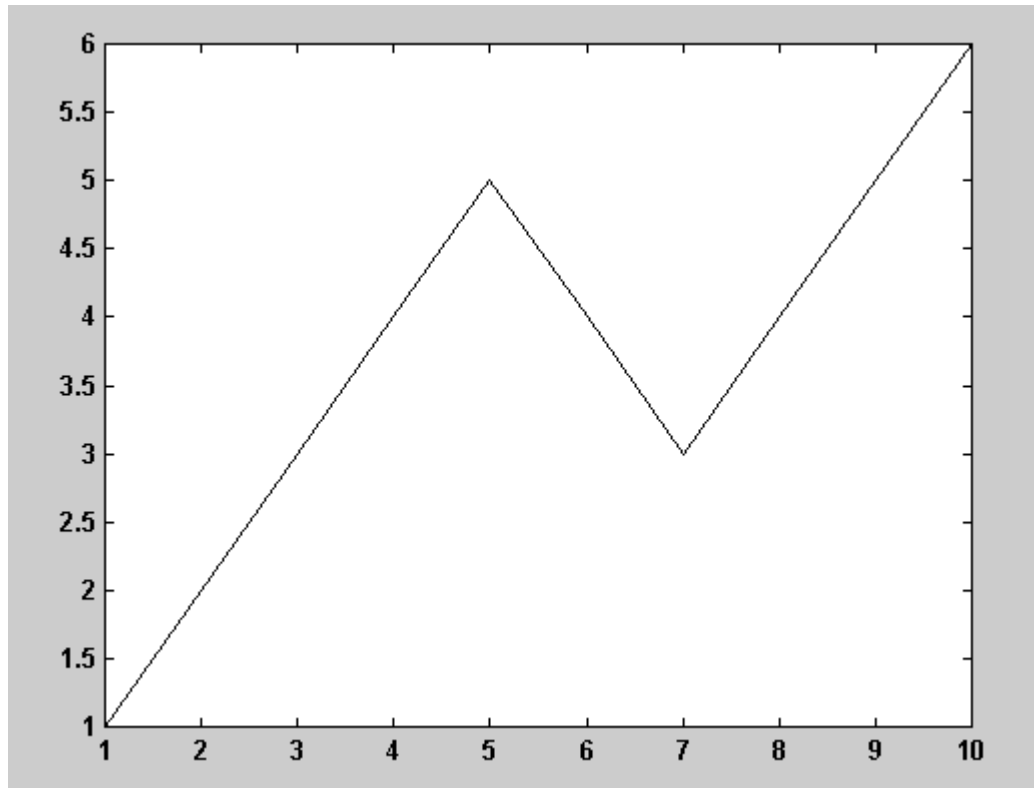
PropertyName và PropertyValue được cho trong bảng sau:

Cú pháp	PropertyName	PropertyValue	Giải thích
Set(h,'Marker','+')	Marker	-, --, :, -, o, x, +, *	Chọn kiểu phần tử
Set(h,'LineWidth',1)	LineWidth	1, 2, 3,...	Độ dày nét vẽ
Set(h,'MarkerSize',9)	MarkerSize	1, 2, 3,...	Kích thước các điểm tạo nên h
Set(h,'color','cyan')	Color	yellow,magenta, red,green,blue, cyan,white,black	Chọn màu cho đối tượng h

d) Ví dụ:

a = [1 2 3 4 5 4 3 4 5 6];

```
h = plot(a)
set(h,'color','black')
```



23. Lệnh stairs

a) Công dụng:

Vẽ đồ thị dạng bậc thang.

b) Cú pháp:

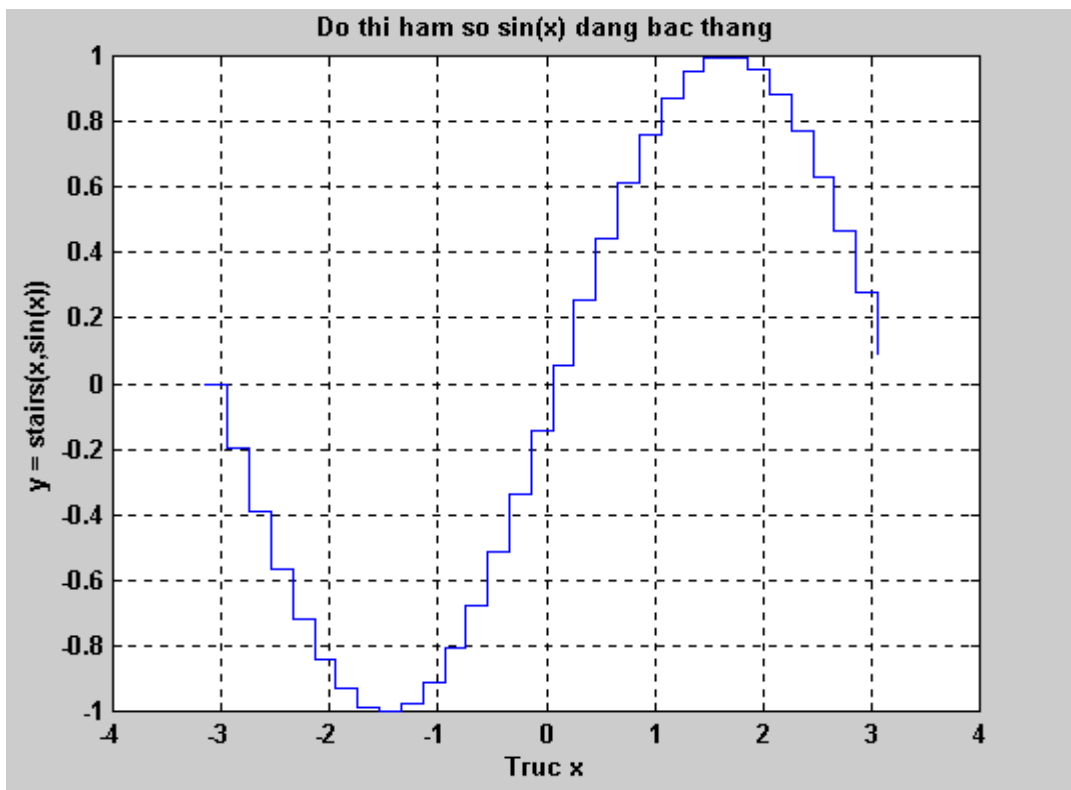
`stairs(x,y)`

c) Giải thích:

Vẽ giá trị x theo giá trị y.

d) Ví dụ:

```
x = -pi:0.2:pi;
stairs(x,sin(x))
xlabel('Trục x')
ylabel('y = stairs(x,sin(x))')
grid on
```



24. Lệnh whitebg

a) Công dụng:

Thay đổi màu nền của cửa sổ đồ họa.

b) Cú pháp:

`whitebg`

`whitebg('color')`

c) Giải thích:

`whitebg` chuyển đổi qua lại màu nền của cửa sổ đồ họa giữa trắng và đen.

`whitebg('color')` chuyển màu nền của cửa sổ đồ họa thành màu của biến `color`.

`color` có thể là các màu: yellow (vàng), magenta (đỏ tươi), cyan (lơ), red (đỏ), green (lục), blue (lam), white (trắng), black (đen).