## LỆNH CƠ BẢN

**Chú ý:** Các lệnh đều viết bằng chữ thường, nhưng vì tác giả muốn viết hoa để người xem tiện theo dõi.

### Lệnh ANS

**a) Công dụng**

Là biến chứa kết quả mặc định.

**b) Giải thích**

Khi thực hiện một lệnh nào đó mà chưa có biến chứa kết quả, thì MATLAB lấy biến Ans làm biến chứa kết quả đó.

**c) Ví dụ**

|  |
| --- |
|  |

2-1

ans = 1

### Lệnh CLOCK

a**) Công dụng**

Thông báo ngày giờ hiện tại.

**b) Cú pháp**

c = clock

**c) Giải thích**

Để thông báo dễ đọc ta dùng hàm fix.

**d) Ví dụ**

|  |
| --- |
|  |

c = clock

c =

1.0e+003\*

2.0010 0.0040 0.0200 0.0030 0.0420 0.0501

c = fix(clock)

c = 2001 4 20 3 43 3

### Lệnh COMPUTER

**a) Công dụng**

Cho biết hệ điều hành của máy vi tính đang sử dụng Matlab.

**b) Cú pháp**

computer

[c,m] = computer

**c) Giải thích**

c: chứa thông báo hệ điều hành của máy.

m: số phần tử của ma trận lớn nhất mà máy có thể làm việc được với Matlab.

**d) Ví dụ**

|  |
| --- |
|  |

» [c,m]=computer

c =

PCWIN

m =

2.1475e+009

### Lệnh DATE

**a) Công dụng**

Thông báo ngày tháng năm hiện tại

**b) Cú pháp**

s = date

**c) Ví dụ:**

|  |
| --- |
|  |

» s=date

s =

20-Apr-2001

### Lệnh CD

**a) Công dụng**

Chuyển đổi thư mục làm việc.

**b) Cú pháp**

cd

cd diretory

cd ..

c) **Giải thích**

cd: cho biết thư nục hiện hành.

diretory: đường dẫn đến thư mục muốn làm việc.

cd .. chuyển đến thư mục cấp cao hơn một bậc.

### Lệnh CLC

**a) Công dụng**

Xóa cửa sổ lệnh.

**b) Cú pháp:**

clc

**c) Ví dụ:**

|  |
| --- |
|  |

clc, for i: 25, home, A = rand(5), end.

### Lệnh CLEAR

**a) Công dụng**

Xóa các đề mục trong bộ nhớ.

**b) Cú pháp**

clear

clear name

clear name1 name2 name3

clear functions

clear variables

clear mex

clear global

clear all

**c) Giải thích**

clear: xóa tất cả các biến khỏi vùng làm việc.

clear name: xóa các biến hay hàm được chỉ ra trong name.

clear functions: xóa tất cả các hàm trong bộ nhơ.

clear variables: xóa tất cả các biến ra khỏi bộ nhớ.

clear mex: xóa tất cả các tập tin .mex ra khỏi bộ nhớ.

clear: xóa tất cả các biến chung.

clear all: xóa tất cả các biến, hàm, và các tập tin .mex khỏi bộ nhớ. Lệnh này làm cho bộ nhớ trống hoàn toàn.

### Lệnh DELETE

**a) Công dụng**

Xóa tập tin và đối tượng đồ họa.

**b) Cú pháp**

delete filename

delete (n)

**c) Giải thích**

file name: tên tập tin cần xóa.

n: biến chứa đối tượng đồ họa cần xóa. Nếu đối tượng là một cửa sổ thì cửa sổ sẽ đóng lại và bị xóa.

### Lệnh DEMO

**a) Công dụng**

Chạy chương trình mặc định của Matlab.

**b) Cú pháp**

demo

**c) Giải thích**

demo: là chương trình có sẵn trong trong Matlab, chương trình này minh họa một số chức năng của Matlab.

### Lệnh DIARY

**a) Công dụng**

Lưu vùng thành file trên đĩa.

**b) Cú pháp**

diary filename

**c) Giải thích**

filename: tên của tập tin.

### Lệnh DIR

**a) Công dụng**

Liệt kê các tập tin và thư mục.

**b) Cú pháp**

dir

dir name

**c) Giải thích**

dir: liệt kê các tập tin và thư mục có trong thư mục hiện hành.

dir name: đường dẫn đến thư mục cần liệt kê.

### lệnh DISP

**a) Công dụng**

Trình bày nội dung của biến (x) ra màn hình

**b) Cú pháp**

disp (x)

**c) Giải thích:**

x: là tên của ma trận hay là tên của biến chứa chuỗi ký tự, nếu trình bày trực tiếp chuỗi ký tự thì chuỗi ký tự được đặt trong dấu ‘’

**d) Ví dụ**

|  |
| --- |
|  |

» num=('Matlab')

num =

Matlab

» disp(num)

Matlab

» num=[2 0 0 1]

num =

2 0 0 1

» disp(num)

2 0 0 1

» num='PHAM QUOC TRUONG'

num =

### Lệnh

PHAM QUOC TRUONG

### Lệnh ECHO

**a) Công dụng**

Hiển thị hay không hiển thị dòng lệnh đang thi hành trong file \*.m.

**b) Cú pháp**

echo on

echo off

**c) Giải thích**

on: hiển thị dòng lệnh.

off: không hiển thị dòng lệnh.

### Lệnh FORMAT

**a) Công dụng**

Định dạng kiểu hiển thị của các con số.

|  |  |  |
| --- | --- | --- |
| **Cú pháp** | **Giải thích** | **Ví dụ** |
| Format short | Hiển thị 4 con số sau dấu chấm | 3.1416 |
| Format long | Hiển thị 14 con số sau dấu chấm | 3.14159265358979 |
| Format rat | Hiển thị dạng phân số của phần nguyên nhỏ nhất | 355/133 |
| Format + | Hiển thị số dương hay âm | ­+ |

### Lệnh HELP

**a) Công dụng**

hướng dẫn cách sử dụng các lệnh trong Matlab.

**b) Cú pháp**

help

help topic

**c) Giải thích**

help: hiển thị vắn tắt các mục hướng dẫn.

topic: tên lệnh cần được hướng dẫn.

### Lệnh HOME

**a) Công dụng**

Đem con trỏ về đầu vùng làm việc.

**b) Cú pháp**

home

### Lệnh LENGTH

**a) Công dụng**

Tính chiều dài của vectơ.

**b) Cú pháp**

l = length (x)

**c) Giải thích**

l: biến chứa chiều dài vectơ.

**d) Ví dụ**

|  |
| --- |
|  |

tính chiều dài của vectơ x.

x = [0 1 2 3 4 5 6 7 8 9]

l = length (x)

l = 10

» x=[01 09 77,20 04 2001 ]

x =

1 9 77 20 4 2001

» l=length(x)

l =

6

### Lệnh LOAD

**a) Công dụng**

Nạp file từ đĩa vào vùng làm việc.

**b) Cú pháp**

load

load filename

load filename

load finame.extension

**c) Giải thích**

load: nạp file matlap.mat

load filename: nạp file filename.mat

load filename.extension: nạp file filename.extension

Tập tin này phải là tập tin dạng ma trận có nghĩa là số cột của hàng dưới phải bằng số cột của hàng trên. Kết quả ta được một ma trận có số cột và hàng chính là số cột và hàng của tập tin văn bản trên.

### Lệnh LOOKFOR

**a) Công dụng**

Hiển thị tất cả các lệnh có liên quan đến topic.

**b) Cú pháp**

lookfor topic

**c) Giải thích**

topic: tên lệnh cần được hướng dẫn.

### Lệnh PACK

**a) Công dụng**

Sắp xếp lại bộ nhớ trong vùng làm việc.

**b) Cú pháp**

pack

pack filename

**c) Giải thích**

Nếu như khi sử dụng Matlap máy tính xuất hiện thông báo “Out of memory” thì lệnh pack có thể tìm thấy một số vùng nhớ còn trống mà không cần phải xóa bớt các biến.

Lệnh pack giải phóng không gian bộ nhớ cần thiết bằng cách nén thông tin trong vùng nhớ xuống cực tiểu. Vì Matlab quản lý bộ nhớ bằng phương pháp xếp chồng nên các đoạn chương trình Matlab có thể làm cho vùng nhớ bị phân mảnh. Do đó sẽ có nhiều vùng nhớ còn trống nhưng không đủ để chứa các biến lớn mới.

Lệnh pack sẽ thực hiện:

+ lưu tất cả các biến lên đĩa trong một tập tin tạm thời là pack.tmp.

+ xóa tất cả các biến và hàm có trong bộ nhớ.

+ lấy lại các biến từ tập tin pack.tmp.

+ xóa tập tin tạm thời pack.tmp.

kết quả là trong vùng nhớ các biến được gộp lại hoặc nén lại tối đa nên không bị lãng phí bộ nhớ.

Pack.finame cho phép chọn tên tập tin tạm thời để chứa các biến. Nếu không chỉ ra tên tập tin tạm thời thì Matlab tự lấy tên tập tin đó là pack.tmp.

Nếu đã dùng lệnh pack mà máy vẫn còn báo thiếu bộ nhớ thì bắt buộc phải xóa bớt các biến trong vùng nhớ đi.

### Lệnh PATH

**a) Công dụng**

Tạo đường dẫn, liệt kê tất cả các đường dẫn đang có.

**b) Cú pháp**

path

p = path

path (p)

**c) Giải thích**

path: liệt kê tất cả các dường dẫn đang có.

p: biến chứa đường dẫn.

path (p): đặt đường dẫn mới.

**d) Ví dụ**

|  |
| --- |
|  |

đặt đường dẫn đến thư mục c:\lvtn\matlab

p = ‘d:\DA\matlab’;

path (p);

### Lệnh QUIT

**a) Công dụng**

Thoát khỏi Matlab.

**b) Cú pháp**

quit

### Lệnh SIZE

**a) Công dụng**

Cho biết số dòng và số cột của một ma trận.

**b) Cú pháp**

d = size (x)

[m,n] = size (x)

m = size (x,1)

n = size (x,2)

**c) Giải thích**

x: tên ma trận.

d: tên vectơ có 2 phần tử, phần tử thứ nhất là số dòng, phần tử còn lại là số cột.

m,n: biến m chứa số dòng, biến n chứa số cột

**d) Ví dụ**

|  |
| --- |
|  |

ta có ma trận a

x = 1 2 3 4

5 6 6 8

» x=[1 2 3 4,5 6 7 8]

x =

1 2 3 4 5 6 7 8

Các bạn chú ý về cách nhập 1 ma trận:

» x=[1 2 3 4;5 6 7 8]

x =

1 2 3 4

5 6 7 8

» d=size(x)

d =

2 4

» m=size(x,1)

m =

2

» n=size(x,2)

n =

4

» [m,n]=size(x)

m =

2

n =

4

### Lệnh TYPE

**a) Công dụng**

Hiển thị nội dung của tập tin.

**b) Cú pháp**

type filename

**c) Giải thích**

filename: tên file cần hiển thị nội dung.

Lệnh này trình bày tập tin được chỉ ra.

### Lệnh WHAT

**a) Công dụng**

Liệt kê các tập tin \*.m, \*.mat, \*.mex.

**b) Cú pháp**

what

what dirname

**c) Giải thích**

what: liệt kê tên các tập tin .m, .mat, .mex có trong thư mục hiện hành.

dirname: tên thư mục cần liệt kê.

### Lệnh WHICH

**a) Công dụng**

Xác định chức năng của funname là hàm của Matlab hay tập tin.

**b) Cú pháp**

which funname

**c) Giải thích**

funname: là tên lệnh trong Matlab hay tên tập tin

**d) Ví dụ**

|  |
| --- |
|  |

which inv

inv is a build-in function

which f

c:\matlab\bin\f.m

### Lệnh WHO, WHOS

**a) Công dụng**

Thông tin về biến đang có trong bộ nhớ.

**b) Cú pháp**

who

whos

who global

whos global

**c) Giải thích**

who: liệt kê tất cả các tên biến đang tồn tại trong bộ nhớ.

whos: liệt kê tên biến, kích thước, số phần tử và xét các phần ảo có khác 0 không.

who global và whos: liệt kê các biến trong vùng làm việc chung.

## CÁC TOÁN TỬ VÀ KÝ TỰ ĐẶC BIỆT

### Các toán tử số học (Arithmetic Operators)

|  |  |
| --- | --- |
| Toán tử | Công dụng |
| + | Cộng ma trận hoặc đại lượng vô hướng (các ma trận phải có cùng kích thước). |
| - | Trừ ma trận hoặc đại lượng vô hướng (các ma trận phải có cùng kích thước). |
| \* | Nhân ma trận hoặc đại lượng vô hướng (ma trận 1 phải có số cột bằng số hàng của ma trận 2). |
| .\* | Nhân từng phần tử của 2 ma trận hoặc 2 đại lượng vô hướng (các ma trận phải có cùng kích thước). |
| \ | Thực hiện chia ngược ma trận hoặc các đại lượng vô hướng (A\B tương đương với inv (A)\*B). |
| .\ | Thực hiện chia ngược từng phần tử của 2 ma trận hoặc 2 đại lượng vô hướng (các ma trận phải có cùng kích thước). |
| / | Thực hiện chia thuận 2 ma trận hoặc đại lượng vô hướng (A/B tương đương với A\*inv(B)). |
| ./ | Thực hiện chia thuận từng phần tử của ma trận này cho ma trận kia (các ma trận phải có cùng kích thước). |
| ^ | Lũy thừa ma trận hoặc các đại lượng vô hướng. |
| .^ | Lũy thừa từng phần tử ma trận hoặc đại lượng vô hướng (các ma trận phải có cùng kích thước). |

\* ví dụ:

|  |  |
| --- | --- |
| Phép tính ma trận | Phép tính mảng |
| 1  x 2  3 | 4  y 5  6 |
| x’ 1 2 3 | y’ 4 5 6 |
| 5  x + y 6  7 | -3  x – y -3  -3 |
| 3  x + 2 4  5 | -3  x – 2 -3  -3 |
| x \* y phép toán sai | 4  x. \* y 10  18 |
| x’\* y 32 | x’.\* y phép toán sai |
| 4 5 6  x \* y’ 8 10 12  12 15 18 | x. \* y’ phép toán sai |
| 2  x \* 2 4  6 | 2  x.\* 2 4  6 |
| x \ y 16/7 | 4  x.\ y 5/2  2 |
| 1/2  2 \ x 1  3/2 | 2  2./ x 1  2/3 |
| 0 0 1/6  x / y 0 0 1/3  0 0 1/2 | 1/4  x./ y 2/5  1/2 |
| 1/2  x / 2 1  3/2 | 1/2  x./ 2 1  3/2 |
| x ^ y phép toán sai | 1/2  x.^ y 32  729 |
| x ^ 2 phép toán sai | 1  x.^ 2 4  9 |
| 2 ^ x phép toán sai | 2  2.^ x 4  8 |

### Toán tử quan hệ (Relational Operators)

|  |  |
| --- | --- |
| Toán tử | Công dụng |
| < | So sánh nhỏ hơn. |
| > | So sánh lớn hơn. |
| >= | So sánh lớn hơn hoặc bằng. |
| <= | So sánh nhỏ hơn hoặc bằng. |
| = = | So sánh bằng nhau cả phần thực và phần ảo. |
| -= | So sánh bằng nhau phần ảo. |

**a) Giải thích**

Các toán tử quan hệ thực hiện so sánh từng thành phần của 2 ma trận. Chúng tạo ra một ma trận có cùng kích thước với 2 ma trận so sánh với các phần tử là 1 nếu phép so sánh là đúng

và là 0 nếu phép so sánh là sai.

Phép so sánh có chế độ ưu tiên sau phép toán số học nhưng trên phép toán logic.

**b) Ví dụ**

|  |
| --- |
|  |

thực hiện phép so sánh sau:

» x=5 % đầu tiên ta nhập x=5

x =

5

» x>=[1 2 3;4 5 6;7 8 9] %so sánh trực tiếp x (x là 5) với ma trận

ans = % rõ ràng các phầ tử 1,2,3,4,5 đều <= 5

1 1 1

1 1 0

0 0 0

» x=5

x =

5

» A=[1 2 3;4 5 6;7 8 9] % ta đặt ma trận A

A =

1 2 3

4 5 6

7 8 9

» x>=A

ans =

1 1 1

1 1 0

0 0 0

» x=A % dòng lệnh này tức là cho x= ma trận A

x =

1 2 3

4 5 6

7 8 9

» x==A % so sánh x và A

ans = % tất cả các phần tử đều đúng

1 1 1

1 1 1

1 1 1

» x=5 % cho lại x=5

x =

5

» x==A % so sánh x = A

ans =

0 0 0

0 1 0 % chỉ duy nhất phần tử 5=x (vì x=5)

0 0 0

» x<A

ans =

0 0 0

0 0 1

1 1 1

### Toán tử logig (Logical Operators)

|  |  |
| --- | --- |
| Toán tử | Công dụng |
| & | Thực hiện phép toán logic AND. |
| | | Thực hiện phép toán logic OR. |
| ~ | Thực hiện phép toán logic NOT. |

**a) Giải thích**

Kết quả của phép toán là 1 nếu phép logic là đúng và là 0 nếu phép logic là sai.

Phép logic có chế độ ưu tiên thấp nhất so với phép toán số học và phép toán so sánh.

**b) Ví dụ**

|  |
| --- |
|  |

Khi thực hiện phép toán 3>4 & 1+ thì máy tính sẽ thực hiện 1+2 được 3, sau đó tới 3>4 được 0 rồi thực hiện 0 & 3 và cuối cùng ta được kết qủa là 0.

### Ký tự đặc biệt (Special Characters)

|  |  |
| --- | --- |
| Ký hiệu | Công dụng |
| [] | Khai báo vector hoặc ma trận. |
| () | Thực hiện phép toán ưu tiên, khai báo các biến và các chỉ số của vector. |
| = | Thực hiện phép gán. |
| ‘ | Chuyển vị ma trận tìm lượng liên hiệp của số phức. |
| . | Điểm chấm thập phân. |
| , | Phân biệt các phần tử của ma trận và các đối số trong dòng lệnh. |
| ; | Ngăn cách giữa các hàng khi khai báo ma trận. |
| % | Thông báo dòng chú thích. |
| ! | Mở cửa sổ MS – DOS. |

### Dấu ‘:’

**a) Công dụng**

Tạo vector hoặc ma trận phụ và lặp đi lặp lại các giá trị.

**b) Giải thích**

|  |  |
| --- | --- |
| Khai báo | Công dụng |
| j : k | Tạo ra chuỗi j, j+1, j+2,…., k-1, k |
| j : i : k | Tạo ra chuỗi j, j+i, j+2I,….,k-i, k |
| A(: , j) | Chỉ cột thứ j của ma trận A |
| A(i , :) | Chỉ hàng thứ i của ma trận |
| A(: , :) | Chỉ toàn bộ ma trận A |
| A(j , k) | Chỉ phần tử A(j), A(j+1)…A(k) |
| A(: , j , k) | Chỉ các phần tử A(:, j), A(:, j+1)…A(:, k) |
| A(:) | Chỉ tất cả các thành phần của ma trận A |

**c) Ví dụ**

|  |
| --- |
|  |

khi khai báo D = 1 : 10

ta được kết quả:

D = 1 2 3 4 5 6 7 8 9 10

còn khi khai báo D = 0 : 2 :10

thì ta được kết quả:

D = 0 2 4 6 8 10

## CÁC HÀM LOGIC (LOGICAL FUNCTION)

### Lệnh ALL

**a) Công dụng**

Kiểm tra vector hay ma trận có giá trị 0 hay không.

**b) Cú pháp**

y = all(x)

**c) Giải thích**

y: biến chứa kết quả

x: tên vedtor hay ma trận

y = 1 khi tất cả các phần tử khác 0

y = 0 khi có 1 phần tử bằng 0

**d) Ví dụ**

|  |
| --- |
|  |

» a=[1 2 3]

a =

1 2 3

» y=all(a)

y =

1

» a=[1 0 3]

a =

1 0 3

» y=all(a)

y =

0

» a=[1 2 3;4 0 6;7 8 9]

a =

1 2 3

4 0 6

7 8 9

» y=all(a)

y =

1 0 1

» a=[1 2 0;0 3 5;2 6 8]

a =

1 2 0

0 3 5

2 6 8

» y=all(a)

y =

0 1 0

### Lệnh ANY

**a) Công dụng**

Kiểm tra vector hay ma trận có giá trị khác 0 hay không.

**b) Cú pháp**

y = any(x)

**c) Giải thích**

y: biến chứa kết quả.

x: tên vector, hay ma trận.

y = 1 khi có 1 phần tử khác 0.

y = 0 khi có 1 phần tử bằng 0.

**d) Ví dụ**

|  |
| --- |
|  |

» a=[1 2 3];

» y=any(a)

y =

1

» b=[1 0 3 0];

» y=any(b)

y =

1

» c=[1 2 0 4;0 2 0 4;1 2 3 4;3 4 5 6]

c =

1 2 0 4

0 2 0 4

1 2 3 4

3 4 5 6

» y=any(c)

y =

1 1 1 1

» d=[0 0 0 0;0 1 3 0]

d =

0 0 0 0

0 1 3 0

» y=any(d)

y =

0 1 1 0

### Lệnh EXIST

**a) Công dụng**

Kiểm tra biến hay file có tồn tại hay không.

**b) Cú pháp**

e = exist(‘item’)

**c) Giải thích:**

item: là tên file hay tên biến.

e: biến chứa giá trị trả về.

|  |  |
| --- | --- |
| e | Y nghĩa |
| 0 | item không tồn tại trong vùng làm việc |
| 1 | item là biến đang tồn tại trong vùng làm việc |
| 2 | item đang tồn tại trên đĩa (chỉ kiểm tra trong thư mục hiện hành) |
| 3 | item là MEX-file |
| 4 | item là file được dịch từ phần mềm Simulink |
| 5 | item là hàm của Matlab |

**d) Ví dụ:**

|  |
| --- |
|  |

e = exist(‘dir’)

e = 5

### Lệnh FIND

**a) Công dụng**

Tìm phần tử trong vector hay ma trận theo yêu cầu.

**b) Cú pháp:**

k = find(x)

[i,j] = find(x)

[i,j,s] = find(x)

**c) Giải thích:**

k: chỉ vị trí của phần tử cần tìm trong vector.

i,j: chỉ số hàng và số cột tương ứng của phần tử cần tìm.

s: chứa giá trị của phần tử cần tìm.

x: tên vector, ma trận hay là yêu cầu đề ra. Nếu không nêu ra yêu cầu thì mặc nhiên là tìm các phần tử khác 0.

**d) Ví dụ:**

|  |
| --- |
|  |

» x=[1 8 0 2 3 0]

x =

1 8 0 2 3 0

» k=find(x)

k =

1 2 4 5

» k=[3 6]

k =

3 6

» a=[5 0 0;8 0 3]

a =

5 0 0

8 0 3

» [i,j,k]=find(a)

i =

1

2

2

j =

1

1

3

k =

5

8

3

## NHÓM LỆNH LẬP TRÌNH TRONG MATLAB

### Lệnh EVAL

**a) Công dụng:**

Chuyển đổi chuỗi ký tự thành biểu thức.

**b) Cú pháp:**

kq = eval(‘string’)

**c) Giải thích:**

kq: biến chứa kết quả.

Nếu ‘string’ là các ký số thì chuyển thành những con số.

Nếu ‘string’ là câu lệnh thì chuyển thành các lệnh thi hành được.

**d) Ví dụ:**

|  |
| --- |
|  |

» a='199999999';

» eval(a)+1

ans =

200000000

### Lệnh FOR

**a) Công dụng:**

Dùng để thực hiện 1 công việc cần lặp đi lặp lại theo một quy luật, với số bước lặp xác định trước.

**b) Cú pháp:**

for biến điều khiển = giá trị đầu : giá trị cuối,

thực hiện công việc;

end

**c) Giải thích:**

Công việc chính là các lệnh cần thi hành, có thể có nhiều lệnh, kết thúc lệnh phải có dấu;

**d) Ví dụ:**

|  |
| --- |
|  |

In ra màn hình 5 dòng ‘PHAM QUOC TRUONG chao cac ban’.

for i = 1:5,

disp(‘PHAM QUOC TRUONG chao cac ban’);

end

PHAM QUOC TRUONG chao cac ban

PHAM QUOC TRUONG chao cac ban

PHAM QUOC TRUONG chao cac ban

PHAM QUOC TRUONG chao cac ban

PHAM QUOC TRUONG chao cac ban

### Lệnh FUNCTION

**a) Công dụng:**

Tạo thêm hàm mới.

**b) Cú pháp:**

function s = n(x)

**c) Giải thích:**

s: tên biến chứa giá trị trả về sau khi thi hành hàm.

n: tên gợi nhớ.

**d) Ví dụ:**

|  |
| --- |
|  |

( ở phần lập trong M.file)

### Lệnh INPUT

**a) Công dụng:**

Dùng để nhập vào 1 giá trị.

**b) Cú pháp:**

tên biến = input (‘promt’)

tên biến = input (‘promt’, ‘s’)

**c) Giải thích:**

tên biến, là nơi lưu giá trị ngập vào.

‘promt’: chuỗi ký tự muốn nhập vào.

‘s’: cho biết giá trị nhập vào là nhiều ký tự.

**d) Ví dụ1:**

|  |
| --- |
|  |

x = input(‘nhập giá trị của biến x: ’)

nhập giá trị của biến x: 5

x = 5

**e) Ví dụ2:**

|  |
| --- |
|  |

trả\_lời = input(‘bạn có muốn tiếp tục không ? ’,’s’)

bạn có muốn tiếp tục không ? không

trả\_lời = không

### Lệnh IF …ELSEIF …ELSE

**a) Công dụng:**

Thực hiện lệnh khi thỏa điều kiện.

**b) Cú pháp:**

if biểu thức luận lý 1

thực hiện công việc 1;

elseif biểu thức luận lý 2

thực hiện công việc 2;

else

thực hiện công việc 3;

end

**c) Giải thích:**

Khi biểu thức luận ký 1 đúng thì thực hiện công việc 1 tương tự cho biểu thức luận lý 2. Nếu cả hai biểu thức sai thì thực hiện công việc sau lệnh else.

Biểu thức luận lý là các phép so sánh ==, <, >, <=, >=

công việc chính là các lệnh cần thi hành, có thể có nhiều lệnh, kết thúc lệnh phải có dấu ;

**d) Ví dụ:**

|  |
| --- |
|  |

Viết chương trình nhập vào 2 số và so sánh hai số đó.

a = input(‘Nhập a: ’);

b = input(‘Nhập b: ’);

if a > b

disp(‘a lớn hơn b’);

elseif a ==b

disp(‘a bằng b’);

else

disp(‘a nhỏ hơn b’);

end

nhập a: 4

nhập b: 5

a nhỏ hơn b

### Lệnh MENU

**a) Công dụng:**

Tạo menu để chọn chức năng.

**b) Cú pháp:**

tên biến = menu (‘Tên menu’,‘chức năng1’,‘chức năng2’, …. , ‘chức năng n’)

**c) Giải thích:**

tên menu: là tiêu đề của menu.

tên biến: là nơi cất giá trị nhận được sau khi chọn chức năng của menu.

Chức năng 1, 2, ….,n:khi chọn chức năng nào thì tên biến có giá trị là số thứ tự của chức năng đó.

**d) Ví dụ:**

|  |
| --- |
|  |

k = menu(‘Choose a color’, ‘Red’, ‘Blue’, ‘Green’)

---- Choose a color ----

1) Red

2) Blue

3) Green

### Lệnh PAUSE

**a) Công dụng:**

Dừng chương trình theo ý muốn.

**b) Cú pháp:**

pause on

pause off

pause (n)

**c) Giải thích:**

pause on: dừng chương trình, và chờ nhấn 1 phím bất kỳ (trừ các phím điều khiển) chương trình thực hiện tiếp.

pause off: tắt chức năng pause.

pause (n): dừng chương trình tại n giây.

**d) Ví dụ:**

|  |
| --- |
|  |

for n = 1 : 3;

disp(‘Press any key to continue…’)

pause

end

Press any key to continue…

Press any key to continue…

Press any key to continue…

### Lệnh WHILE

**a) Công dụng:**

Dùng để thực hiện 1 công việc cần lặp đi lặp lại theo một quy luật, với số bước lặp không xác định, phụ thuộc vào biểu thức luận lý.

**b) Cú pháp:**

while biểu thức luận lý

thực hiện công việc;

end

**c) Giải thích:**

Biểu thức luận lý là các phép so sánh = =, <, >, <=, >=

Công việc chính là các lệnh cần thi hành, có thể có nhiều lệnh, kết thúc lệnh phải có dấu ;

Khi thực hiện xong công việc thì quay lên kiểm tra lại biểu thức luận lý, nếu vẫn còn đúng thì tiếp tục thực hiện, nếu sai thì kết thúc.

**d) Ví dụ:**

|  |
| --- |
|  |

tính tổng A = 1+1/2+1/3+…+1/n

n = input(‘nhập vào số n ’);

a = 0; i = 1

while i <= n

a = a + 1/i

i = i + 1;

end

disp(‘ket qua’);

disp(a);

nhap vao so n 3

ket qua

1.8333

## BÀI TẬP

**B1(BT4a):** Viết chương trình nhập vào một số n(n>=0)

với các trường hợp sau:

a) Nếu n<0 thì in thông báo bạn nhập sai

b) Nếu n>0 và lẽ thì tính tổng s1=1+3+5+...+n,n là số lẽ.

c) Nếu n>0 và chẵn thì s2=2+4+6+...+n,n chẵn.

1. Nếu n=0 dừng chương trình lại.

% BT4a: Viet chuong trinh nhap vao mot so n(n>=0)

% voi cac truong hop sau:

% a) Neu n<0 thi in thong bao ban nhap sai

% b) Neu n>0 va le thi tinh tong s1=1+3+5+...+n,n la so le.

% c) Neu n>0 va chan thi s2=2+4+6+...+n,n chan.

% d) Neu n=0 dung chuong trinh lai.

n=input('nhap n= '); %nhap so n

du=rem(n,2); %kiem tra n la le hay chan

%neu n le du=1, n chan du=0

if n<0

fprintf('Ban nhap sai') %xuat ra thong bao

end

if (n>0) & (du==1) %neu n>0 va le

i=1; %gan i=1;

s1=1; %gan tong s1=1

while i<n %thuc hien vong lap

i=i+2; %tang i len 2 sau moi lan lap

s1=s1+i; %tinh tong s1 voi gia tri i moi

end

s1 %in ra ket qua sau khi ket thuc vong lap

end

if (n>0) & (du==0)

i=0;

s2=0;

while i<n

i=i+2;

s2=s2+i;

end

s2

end

if n==0 %neu n=0

break %lenh ket thuc

end

Khi chạy chương trình:

» nhap n= 5

s1 =

9

» BT4a

nhap n= 4

s2 =

6

» BT4a

nhap n= -6

Ban nhap sai» BT4a

nhap n= 0

»

## TẬP LỆNH XỬ LÝ CHUỖI

### Lệnh ABS

**a) Công dụng:**

Tạo vector đơn có giá trị của mỗi phần tử là số thứ tự tương ứng với ký tự trong bảng mả ASCII.

**Lấy trị tuyệt đối của một số âm.**

**b) Cú pháp:**

n = ABS(s)

x = ABS(a)

**c) Giải thích:**

n: tên vector.

s: chuỗi ký tự, hoặc là tên biến chứa chuỗi ký tự.

a: số âm, hoặc là tên biến chứa số âm.

x: trị tuyệt đối của a.

**d) Ví dụ:**

|  |
| --- |
|  |

» n=abs('PHAM QUOC TRUONG')

n =

Columns 1 through 12

80 72 65 77 32 81 85 79 67 32 84 82

Columns 13 through 16

85 79 78 71

» m=abs('MATLAB')

m =

77 65 84 76 65 66

» U=abs('abc')

U =

97 98 99

» T=abs(-1)

T =

1

### Lệnh BLANKS

**a) Công dụng:**

Tạo khoảng trắng giữa hai hay nhiều chuỗi ký tự theo mong muốn.

**b) Cú pháp:**

[S1 BLANKS(b1) S2 BLANKS(b2) …BLANKS(bn) Sn]

**c) Giải thích:**

S1, S2, …Sn: các chuỗi ký tự.

b1, b2: số khoảng trắng.

**d) Ví dụ:**

|  |
| --- |
|  |

In 4 chuỗi ‘Khao sat’,‘ứng dụng’, ‘MATLAB’, ‘trong điều khiển tự động’ ra màn hình với khoảng cách lần lượt giữa 4 chuỗi là: 2,4,3

» S=['Khao sat'blanks(2) 'ung dung'blanks(4) 'MATLAB'blanks(3) 'trong dieu khien tu dong']

S =

Khao sat ung dung MATLAB trong dieu khien tu dong

### Lệnh DEC2HEX

**a) Công dụng:**

Đổi con số của hệ 10 sang hệ 16.

**b) Cú pháp:**

s = dec2hex(n)

**c) Giải thích:**

s: biến chứa chuỗi ký số của hệ 16

n: con số nguyên hệ 10.

**d) Ví dụ:**

|  |
| --- |
|  |

s = dec2hex(10)

s = ‘A’

### Lệnh HEX2DEC

**a) Công dụng:**

Đổi chuỗi ký số của hệ 16 sang con số của hệ 10.

**b) Cú pháp:**

n = hex2dec(‘s’)

**c) Giải thích:**

n: con số của hệ 10.

s: chuỗi ký số hệ 16.

**d) Ví dụ:**

|  |
| --- |
|  |

n = hex2dec(‘A’)

n = 10

### Lệnh INT2STR

**a) Công dụng:**

Chuyển số nguyên sang dạng chuỗi.

Chuyển các ký tự trong một chuỗi sang số thứ tự tương ứng trong bảng mã ASCII.

**b) Cú pháp:**

kq = INT

**c) Giải thích:**

kq: biến STR(n)chứa kết quả.

n: tên biến cần chuyển.

Nếu n là số nguyên thì kq là chuỗi ký số.

Nếu n là chuỗi ký tự thì kq là số tương ứng trong bảng mã ASCII

**d) Ví dụ:**

|  |
| --- |
|  |

» n='MATLAB'

n =

MATLAB

» t=int2str(n)

t =

77 65 84 76 65 66

» n=2001

n =

2001

» t=int2str(n)

t =

2001

### Lệnh ISSTR

**a) Công dụng:**

Kiểm tra nội dung biến có phải là chuỗi ký tự không.

**b) Cú pháp:**

kq = isstr(n)

**c) Giải thích:**

kq: biến chứa kết quả.

n: tên biến cần kiểm tra.

kq = 1 nếu n là chuỗi ký tự.

0 nếu n không là chuỗi ký tự.

**d) Ví dụ:**

|  |
| --- |
|  |

» n='MATLAB';

» kq=isstr(n)

kq =

1

» m=[1 2 3 4];

» kq=isstr(m)

kq =

0

### Lệnh LOWER

**a) Công dụng:**

Cho ra chuỗi ký tự viết thường.

**b) Cú pháp:**

b = lower(s)

**c) Giải thích:**

b: biến chứa kết quả.

s: tên biến chứa chuỗi ký tự hay chuỗi ký tự.

**d) Ví dụ:**

|  |
| --- |
|  |

» a='DO AN cua pHAm quOC TRuOnG';

» b=lower(a)

b =

do an cua pham quoc truong

### Lệnh NUM2STR

**a) Công dụng:**

Chuyễn số thực sang dạng chuỗi.

Chuyển các ký tự trong một chuỗi sang số thứ tự tương ứng trong bảng mã ASCII.

**b) Cú pháp:**

kq = num2tr(n)

**c) Giải thích:**

kq: biến chứa kết quả.

n: tên biến cần chuyển.

Nếu n là số thực thì kq là số tương ứng trong bảng mã ASCII.

**d) Ví dụ:**

|  |
| --- |
|  |

» n=3.1416;

» kq=num2str(n)

kq =

3.1416

### Lệnh SETSTR

**a) Công dụng:**

Cho ra ký tự tương ứng với số thứ tự trong bảng mã ASCII.

**b) Cú pháp:**

x = Set Str(n)

**c) Giải thích:**

x: biến chứa ký tự tương ứng (thuộc bảng mã ASCII).

n: số nguyên (0 ≤ n ≤ 255).

**d) Ví dụ**

|  |
| --- |
|  |

Tìm ký tự có số thứ tự là 65 trong bảng mã ASCII.

» kt=setstr(65)

kt =

A

### Lệnh STR2MAT

**a) Công dụng:**

Tạo ma trận có các phần tử dạng chuỗi.

**b) Cú pháp:**

s = str2mat(‘s1’, ‘s2’, …)

**c) Giải thích:**

s: tên ma trận kết quả.

s1, s2: chuỗi ký tự.

**d) Ví dụ**

|  |
| --- |
|  |

s = str2mat(‘mat’, ‘lab’)

s =

mat

lab

### Lệnh STR2NUM

**a) Công dụng:**

Chuyển chuỗi (dạng số) sang số thực.

**b) Cú pháp:**

n = str2num(s)

**c) Giải thích:**

s: chuỗi dạng số.

n: số thực.

**d) Ví dụ**

|  |
| --- |
|  |

n = str2num(‘456456’)

n = 456456

### Lệnh STRCMP

**a) Công dụng:**

So sánh 2 chuỗi ký tự.

**b) Cú pháp:**

l = strcmp(s1, s2)

**c) Giải thích**:

l: biến chứa kết quả.

s1, s2: chuỗi cần so sánh.

**d) Ví dụ**

|  |
| --- |
|  |

a = ‘MatLab WoRkS’

b = ‘MatLab WoRkS’

strcmp(a,b)

ans = 1

### Lệnh UPPER

**a) Công dụng:**

Cho ra chuỗi viết hoa.

**b) Cú pháp:**

b = upper

**c) Giải thích:**

b: biến chứa kết quả.

s: tên biến chứa chuỗi ký tự.

**d) Ví dụ**

|  |
| --- |
|  |

a = ‘MaTlab WORks’

b = upper(a)

b = MATLAB

b = upper(‘MaTlab WORks’)

b= MATLAB WORKS

## CÁC HÀM GIAO TIẾP

### Lệnh FCLOSE

**a) Công dụng:**

Đóng file đang mở sau khi truy xuất xong.

**b) Cú pháp:**

fclose(fid)

**c) Giải thích:**

fid: tên biến trỏ đến file đang mở.

### Lệnh FOPEN

**a) Công dụng:**

Mở file hoặc truy xuất dữ liệu của file đang mở.

**b) Cú pháp:**

fid = fopen(‘fn’)

fid = fopen(‘fn’, ‘p’)

**c) Giải thích:**

fid: tên biến trỏ đến file đang mở.

fn: tên file (có thể đặt đường dẫn).

Tham số p có các định dạng sau:

‘r’: chỉ đọc.

‘r+’: đọc và ghi.

‘w’: xóa tất cả nội dung của file hoặc tạo 1 file mới và mở file đó để ghi.

‘w+’: xóa tất cả nội dung của file hoặc tạo 1 file mới và mở file đó để ghi và đọc.

### Lệnh FPRINTF

**a) Công dụng:**

Ghi đoạn dữ liệu thành file.

**b) Cú pháp:**

fprintf(fid, f)

**c) Giải thích:**

fid: tên biến trỏ đến file cần ghi.

f: các tham số để định dạng.

**d) Ví dụ**

|  |
| --- |
|  |

Tạo file exp.txt có nội dung:

x = 0:2:10;

y = [x, x/2];

fid = fopen(‘exp.txt’, ‘w’);

fprintf(fid, ‘%d’, [2, inf]);

Gán file exp.txt và biến a để xem nội dung:

fid = fopen(‘exp.txt’)

a = fscanf(fid, ‘%d’, [2,inf]);

disp(a);

fclose(fid);

Kết quả

0 2 4 6 8 10

0 1 2 3 4 5

### Lệnh FREAD

**a) Công dụng:**

Đọc dữ liệu dạng nhị phân từ file.

**b) Cú pháp:**

[a, c] =fscanf(fid)

[a, c] = fscanf(fid,s)

**c) Giải thích:**

a: tên biến chứa dữ liệc được đọc vào.

c: số phần tử được đọc vào.

fid: tên biến trỏ đến file cần đọc.

s: kích thước dữ liệu đọc vào.

s được định dạng bởi các thông số:

n: chỉ đọc n phần tử vào cột vector a.

inf: đọc đến hết file.

[m,n]: chỉ đọc vào m cột và n hàng, n có thể bằng inf còn m thì không.

**d) Ví dụ1**

|  |
| --- |
|  |

file vd.txt có nội dung:

A B C

1 2 3

fid = fopen(vd.txt’);

[a,c] = fread(fid);

disp(a);

disp(c);

a =

65

32

66

32

67

13

10

49

32

50

32

51

c =

12

**e) Ví dụ2**

|  |
| --- |
|  |

fid = fopen(‘vd1.txt’);

[a,c] = fread(fid, 4);

disp(a);

disp(c);

a=

65

32

66

32

c =

4

**f) Ví dụ 3**

|  |
| --- |
|  |

file vd3.txt có nội dung

ABCDE

FGHIJ

KLMNO

fid = fopen(‘vd3.txt’);

[a,c] = fread(fid, [7, inf]);

disp(a);

disp(c);

a =

65 70 75

66 71 76

67 72 76

1. 73 78
2. 74 79

13 13 13

10 10 10

c =

21

a’=

65 66 67 68 69 13 10

70 71 72 73 74 13 10

75 76 77 78 79 13 10

### Lệnh FWRITE

**a) Công dụng:**

Ghi đoạn dữ liệu dạng nhị phân thành file.

**b) Cú pháp:**

fwrite (fid,a)

**c) Giải thích:**

fid: tên biến trỏ đến file cần ghi.

a: tên biến chứa dữ liệu.

**d) Ví dụ**

|  |
| --- |
|  |

Ghi đoạn dữ liệu của biến a thành file a.txt

a = [65 66 67]

fid = fopen(‘a.txt’, ‘w’);

fwrite(fid, ‘%’);

fwite(fid,a);

Gán file a.txt vào biến b để xem nội dung

fid = fopen(‘a.txt’);

b = fscanf(fid, ‘%’);

disp(b);

fclose(fid);

Kết quả

b = ABC

### Lệnh SPRINTF

**a) Công dụng:**

Hiển thị thông tin lên màn hình.

**b) Cú pháp:**

s = sprintf(‘ts’,ds)

**c) Giải thích:**

s: biến chứa chuỗi số hiển thị trên màn hình.

ts: các tham số định dạng.

ds: danh sách các đối số.

Tham số định dạng thuộc 1 trong 2 kiểu sau:

1. Chuỗi ký tự: chuỗi này sẽ được hiển thị lên màn hình giống hệt như được viết trong câu lệnh.

(2) Chuỗi các tham số định dạng: các chuỗi này sẽ không được hiển thị lên màn hình, nhưng tác dụng điều khiển việc chuyển đổi và cách hiển thị các đối số được đưa ra trong danh sách các đối số.

Ví dụ các tham số định dạng:

1) %d: đối số là số nguyên được viết dưới dạng thập phân.

s = sprintf(‘Đây là số: %d’,-24)

s = Đây là số: -2

2) %u: đối số là số nguyên được viết dưới dạng thập phân không dấu.

s = sprintf(‘Đây là số: %u’,24)

s = Đây là số: 24

3) %o: đối số là số nguyên được viết dưới dạng cơ số 8 không dấu.

s = sprintf(‘Đây là số: %o’,9)

s = Đây là số: 11

4) %x: đối số là số nguyên được viết dưới dạng cơ số 16.

s = sprintf(‘Đây là số: %x’,255)

s = Đây là số:ff

5) %f: đối số là số nguyên được viết dưới dạng cp số 10.

s = sprintf(‘Đây là số: %f’,2550

s = Đây là số: 255.000000

Để định dạng phần thập phân thì thêm vào con số chứa số thập phân cần lấy.

s = sprintf(‘Đây là số: %.3f’, 2.5568)

s = Đây là số: 2.557

6) %c: đối số là 1 ký tự riêng đặc biệt.

s = sprintf(‘Đây là chữ: %c’,’M’)

s = Đây là chữ: M

7)%s: đối số là chuỗi ký tự.

s = sprintf(‘Đây là chuỗi: %s’, ‘Matlab’)

s = Đây là chuỗi: Matlab

### Lệnh SSCANF

**a) Công dụng:**

Đọc chuỗi ký tự và định dạng lại chuỗi ký tự đó.

**b) Cú pháp:**

[a,count] = sscanf(s, ‘format’, size)

**c) Giải thích:**

a: tên biến chứa chuỗi ký tự sau khi được định dạng.

count: đếm số phần tử được đọc vào.

size: kích thước sẽ được đọc vào.

format: phần định dạng giống như lệnh sprintf.

**d) Ví dụ**

|  |
| --- |
|  |

s = ‘3.12 1.2 0.23 2.56’;

[a, count] = sscanf(s, ‘%f’,3)

a =

3.1200

1.2000

0.2300

count =

3

## CÁC HÀM TOÁN HỌC CƠ BẢN

### Một số hàm lượng giác

**a) Cú pháp:**

kq = hlg(x)

**b) Giải thích:**

kq: tên biến chứa kết quả.

x: đơn vị radian.

hlg: tên hàm lượng giác.

|  |  |
| --- | --- |
| Tên hàm lượng giác | Giải thích |
| sin  cos  tan  asin  atan  sinh  cosh  tanh | Tính giá trị sine  Tính giá trị cosine  Tính giá trị tangent  Nghịch đảo của sine  Nghịch đảo của tangent  Tính giá trị hyperbolic sine  Tính giá trị hyperbolic cosine  Tính gía trị hyperbolic tangent |

### Lệnh ANGLE

**a) Công dụng:**

Tính góc pha của số phức.

**b) Cú pháp:**

p = angle(z)

**c) Giải thích:**

p: tên biến chứa kết quả, đơn vị radians

z: số phức

**d) Ví dụ**

|  |
| --- |
|  |

z = i-3j

z = 0 – 2.0000i

p = angle(z)

p = -1.5708

### Lệnh CEIL

**a) Công dụng:**

Làm tròn số về phía số nguyên lớn hơn.

**b) Cú pháp**:

y = ceil(x)

**c) Giải thích:**

y: số sau khi được làm tròn.

x: số cần được làm tròn.

**d) Ví dụ**

|  |
| --- |
|  |

x = -1.9000 -0.2000 3.4000 5.6000 7.0000

y = ceil(x)

y = -1 0 4 6 7

### Lệnh CONJ

**a) Công dụng:**

Tính lượng liên hiệp của số phức.

**b) Cú pháp:**

y = conj(z)

**c) Giải thích:**

y: tên biến chứa lượng liên hiệp

z: số phức

**d) Ví dụ**

|  |
| --- |
|  |

z = -3i + 2j

z = 0 – 1.0000i

y = conj(z)

y = 0 + 1.0000i

### Lệnh EXP

**a) Công dụng:**

Tính giá trị ex.

**b) Cú pháp:**

y = exp(x)

**c) Ví dụ**

|  |
| --- |
|  |

y = exp(x)

y = 20.0855

### Lệnh FIX

**a) Công dụng:**

Làm tròn số về phía zero.

**b) Cú pháp:**

y = fix(x)

**c) Giải thích:**

y: số sau khi được làm tròn.

x: số cần được làm tròn.

**d) Ví dụ**

|  |
| --- |
|  |

x = -1.9000 -0.2000 3.4000 5.6000 7.0000

y = fix(x)

y = -1 0 3 5 7

### Lệnh FLOOR

**a) Công dụng:**

Làm tròn số về phía số nguyên nhỏ hơn.

**b) Cú pháp:**

y = floor(x)

**c) Giải thích:**

y: số sau khi được làm tròn .

x: số cần được làm tròn

**d) Ví dụ**

|  |
| --- |
|  |

x = -1.9000 -0.2000 3.4000 5.6000 7.0000

y = floor(x)

y = -2 -1 3 5 7

### Lệnh IMAG

**a) Công dụng:**

Lấy phần ảo của số phức.

**b) Cú pháp:**

y = imag(z)

**c) Ví dụ**

|  |
| --- |
|  |

y = imag(2 + 3j)

y = 3

### Lệnh LOG

**a) Công dụng:**

Tìm logarithm cơ số e.

**b) Cú pháp:**

y = log(x)

**d) Ví dụ**

|  |
| --- |
|  |

y = log(2.718)

y = 0.9999

### Lệnh LOG2

**a) Công dụng:**

Tìm logarithm cơ số 2.

**b) Cú pháp:**

y = log2(x)

**d) Ví dụ**

|  |
| --- |
|  |

y = log2(2)

y = 1

### Lệnh LOG10

**a) Công dụng:**

Tìm logarithm cơ số 10.

**b) Cú pháp:**

y = log10(x)

**d) Ví dụ**

|  |
| --- |
|  |

y = log10(10)

y = 1

### Lệnh REAL

**a) Công dụng:**

Lấy phần thực của số phức.

**b) Cú pháp:**

y = real(z)

**d) Ví dụ**

|  |
| --- |
|  |

y = real(1 + 3j)

y = 2

### Lệnh REM

**a) Công dụng:**

Cho phần dư của phép chia.

**b) Cú pháp:**

r = rem(a,b)

**c) Giải thích:**

r: biến chứa kết quả

a, b: số chia và số bị chia

**d) Ví dụ**

|  |
| --- |
|  |

r = rem(16, 3)

r = 1

### Lệnh ROUND

**a) Công dụng:**

Làm tròn số sao cho gần số nguyên nhất.

**b) Cú pháp:**

y = round(x)

**c) Ví dụ**

|  |
| --- |
|  |

x = -1.9000 -0.2000 3.4000 5.6000 7.0000

y = round(x)

y= -2 0 3 6 7

Bảng so sánh của các phép làm tròn số

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | -1.9000 | -0.2000 | 3.4000 | 5.6000 | 7.0000 |
| ceil(x) | -1 | 0 | 4 | 6 | 7 |
| floor(x) | -2 | -1 | 3 | 5 | 7 |
| fix(x) | -1 | 0 | 3 | 5 | 7 |
| round(x) | -2 | 0 | 3 | 6 | 7 |

### Lệnh SIGN

**a) Công dụng:**

Xét dấu số thực.

**b) Cú pháp:**

y = sign(x)

**c) Giải thích:**

x: số thực cần xét dấu.

y: kết quả trả về.

|  |  |
| --- | --- |
| y | x |
| 0 | số 0 |
| 1 | số dương |
| -1 | số âm |

**d) Ví dụ**

|  |
| --- |
|  |

x = 2 0 -3 0.5

y = sugn(x)

y= 1 0 -1 1

### Lệnh SQRT

**a) Công dụng:**

Tính căn bậc hai.

**b) Cú pháp:**

y = sqrt(x)

**c) Ví dụ**

|  |
| --- |
|  |

x = 4

y = sqrt(x)

y = 2

## TẬP LỆNH THAO TÁC TRÊN MA TRẬN

### Cộng, trừ, nhân, chia từng phần tử của ma trận với hằng số

**a) Cú pháp:**

Ma trận kết quả = ma trận [+] [-] [.] [/] hằng số.

**b) Ví dụ**

|  |
| --- |
|  |

a =

1 2 3

4 5 6

7 8 9

Cộng ma trận a với 2 kết quả là ma trận b

b = a + 2

b =

3 4 5

6 7 8

9 10 11

tương tự cho các phép tính trừ, nhân và chia.

### Lệnh DET

**a) Công dụng:**

Dùng để tính định thức của ma trận.

**b) Ví dụ**

|  |
| --- |
|  |

Tính định thức của ma trận a

a =

1. 4

5 6

det(a)

ans = -8

### Lệnh DIAG

**a) Công dụng:**

Tạo ma trận mới và xử lý đường chéo theo quy ước.

**b) Cú pháp:**

v = diag(x)

v = diag(x,k)

**c) Giải thích:**

x: là vector có n phần tử.

v: là ma trận được tạo ra từ x theo quy tắc: số hàng bằng số cột và các phần tử của x nằm trên đường chéo của v.

k: tham số định dạng cho v, số hàng và cột của v = n + abs(k).

Nếu k = 0 đường chéo của v chính là các phần tử của x

Nếu k > 0 các phần tử của x nằm phía trên đường chéo v

Nếu k < 0 các phần tử của x nằm phía dưới đường chéo v

**d) Ví dụ**

|  |
| --- |
|  |

x = 2 1 9 5 4

v = diag(x)

v =

2 0 0 0 0

0 1 0 0 0

0 0 9 0 0

0 0 0 0 4

v = diag(x,2)

v =

0 0 2 0 0 0 0

0 0 0 1 0 0 0

0 0 0 0 9 0 0

0 0 0 0 0 5 0

0 0 0 0 0 0 4

0 0 0 0 0 0 0

0 0 0 0 0 0 0

v = diag(x,0)

v =

2 0 0 0 0

0 1 0 0 0

0 0 9 0 0

0 0 0 5 0

0 0 0 0 4

v = diag(x,-2)

v =

0 0 0 0 0 0 0

0 0 0 0 0 0 0

2 0 0 0 0 0 0

0 1 0 0 0 0 0

0 0 9 0 0 0 0

0 0 0 5 0 0 0

0 0 0 0 4 0 0

### Lệnh EYE

**a) Công dụng:**

Tạo ma trận đơn vị.

**b) Cú pháp:**

y = eye(n)

y = eye(n,m)

**c) Giải thích:**

n: tạo ma trận có n hàng, n cột.

m, n: tạo ma trận có m hàng, n cột.

**d) Ví dụ**

|  |
| --- |
|  |

y = eye(3)

y =

1 0 0

0 1 0

0 0 1

y = eye(3,5)

y =

1 0 0 0 0

0 1 0 0 0

0 0 1 0 0

### Lệnh FLIPLR

**a) Công dụng:**

Chuyển các phần tử của các ma trận theo thứ tự cột ngược lại.

**b) Cú pháp:**

b = fliplr(a)

**c) Giải thích:**

b: tên ma trận được chuyển đổi.

a: tên ma trận cần chuyển đổi.

**d) Ví dụ**

|  |
| --- |
|  |

a =

0 1 2 3 4

5 6 7 8 9

b = fliplr(a)

4 3 2 1 0

9 8 7 6 5

### Lệnh FLIPUD

**a) Công dụng:**

Chuyển các phần tử của ma trận theo thứ tự hàng ngược lại.

**b) Cú pháp:**

b = flipud(a)

**c) Giải thích:**

b: tên ma trận được chuyển đổi.

a: tên ma trận cần chuyển đổi.

**d) Ví dụ**

|  |
| --- |
|  |

a =

1. 4
2. 5
3. 6

b = flipud(a)

b =

3 6

2 5

1 4

### Lệnh INV

**a) Công dụng:**

Tìm ma trận nghịch đảo.

**b) Cú pháp:**

Ma trận nghịch đảo = inv (ma trận)

**c) Ví dụ**

|  |
| --- |
|  |

Tìm ma trận nghịch đảo của a.

a =

1 2 0

2 5 -1

4 10 -1

b = inv(a)

b =

5 2 -2

-2 -1 1

0 -2 1

### Lệnh tạo ma trận

**a) Công dụng:**

Dùng để tạo 1 ma trận gồm có n hàng và m cột.

**b) Cú pháp:**

Tên ma trận = [a11 a12…a1m ; a21 a22… a2m ;…;…]

**c) Giải thích:**

a11, a12, a1m là các giá trị tại hàng 1 cột 1 đến các giá trị tại hàng 1 cột m, có n dấu (;) là có n hàng.

**d) Ví dụ**

|  |
| --- |
|  |

Tạo ma trận gồm 3 hàng và 3 cột với giá trị là

1 2 3

4 5 6

1 0 0

a = [1 2 3; 4 5 6; 1 0 0]

a = 1 2 3

4 5 6

1 0 0

### Lệnh tạo vector đơn

**a) Công dụng**:

Lệnh này dùng để tạo 1 vector đơn gồm có n phần tử.

**b) Cú pháp 1:**

Tên vector = [pt1 pt2 pt3 …ptn]

**c) Giải thích:**

pt1 pt2 …ptn: là các số thực.

**d) Ví dụ**

|  |
| --- |
|  |

Tạo vector a gồm có 4 phần tử, với các giá trị là:1, 3, 7, 4

a = [1 3 7 4]

a =

1 3 7 4

**e) Cú pháp 2:**

Tên vector = gtđ:csc:gtkt

**f) Giải thích:**

gtđ: là giá trị bắt đầu của vector.

csc: cấp số cộng.

gtkt: giá trị kết thúc.

**g) Ví dụ**

|  |
| --- |
|  |

Tạo vector a có giá trị bắt đầu 0.2, giá trị kết thúc pi/2

(= 1.5708), cấp số cộng 0,3.

a = 0.2;0.3;pi/2

a =

0.2000 0.5000 0.8000 1.1000 1.4000

### Lệnh LINSPACE

**a) Công dụng:**

Tạo vector có giá trị ngẫu nhiên giới hạn trong khoảng định trước.

**b) Cú pháp:**

y = linspace(x1, x2)

y = linspace(x1, x2, n)

**c) Giải thích:**

y: tên của vector.

x1, x2: giới hạn giá trị lớn nhất và nhỏ nhất của vector y.

n: số phần tử của vector y.

Nếu không có giá trị n thì mặc định n = 100.

**d) Ví dụ**

|  |
| --- |
|  |

y = linspace(1, 10, 7)

y = 1.0000 2.5000 4.0000 5.5000 7.0000 8.5000 10.0000

### Ma trận chuyển vị

**a) Công dụng:**

Ma trận chuyển vị = ma trận đang có.

**b) Cú pháp:**

Tạo 1 ma trận chuyển vị từ 1 ma trận đang có.

**c) Ví dụ**

|  |
| --- |
|  |

a =

1 2 3

4 5 6

7 8 9

ma trận chuyển vị b

b = a’

b =

1. 7
2. 8
3. 9

### Lệnh MAGIC

**a) Công dụng:**

Tạo 1 ma trận vuông có tổng của các phần tử trong 1 hàng, 1 cột hoặc trên đường chéo bằng nhau.

**b) Cú pháp:**

Tên ma trận = magic(n)

**c) Giải thích:**

n: kích thước ma trận.

Giá trị của mỗi phần tử trong ma trận là một dãy số nguyên liên tục từ 1 đến 2n.

Tổng các hàng, cột và các đường chéo đều bằng nhau.

**d) Ví dụ**

|  |
| --- |
|  |

tmt = magic(3)

tmt =

8 1 6

3 5 7

4 9 2

### Nhân ma trận

**a) Công dụng:**

Ma trận kết quả = ma trận 1\* ma trận 2.

**b) Ví dụ**

|  |
| --- |
|  |

Ta có 2 ma trận a và b như trên và c là ma trận kết quả

c = a\*b

c =

14 32 50

32 77 122

50 122 194

### Lệnh ONES

**a) Công dụng:**

Tạo ma trận mà giá trị của các phần tử là 1.

**b) Cú pháp:**

y = ones(n)

y = ones(m,n)

**c) Giải thích:**

y = tên ma trận.

n: tạo ma trận có n hàng

m, n: tạo ma trận có m hàng, n cột.

**d) Ví dụ**

|  |
| --- |
|  |

y = ones(3)

y =

1 1 1

1 1 1

1 1 1

y = ones(3,5)

y =

1 1 1 1 1

1 1 1 1 1

1 1 1 1 1

### Lệnh PASCAL

**a) Công dụng:**

Tạo ma trận theo quy luận tam giác Pascal.

**b) Cú pháp:**

pascal (n)

**c) Giải thích:**

n: là số hàng (cột)

**d) Ví dụ**

|  |
| --- |
|  |

pascal(4)

ans =

1 1 1 1

1 2 3 4

1 3 6 10

1 4 10 20

### Lệnh RAND

**a) Công dụng:**

Tạo ma trận mà kết mà giá trị của các phần tử là ngẫu nhiên.

**b) Cú pháp:**

y = rand(n)   
 y = rand(m,n)

**c) Giải thích:**

y: tên ma trận.

n: tạo ma trận có n hàng, n cột.

m, n: tạo ma trận có m hàng, n cột.

Giá trị của các phần tử nằm trong khoảng [0 1]

**d) Ví dụ**

|  |
| --- |
|  |

y = rand(3)

y =

0.9340 0.0920 0.7012

0.8462 0.6539 0.7622

0.5269 0.4160 0.7622

y = rand(3,5)

y =

0.2625 0.3282 0.9910 0.9826 0.6515

0.0475 0.6326 0.3653 0.7227 0.0727

0.7361 0.7564 0.2470 0.7534 0.6316

### Lệnh RESHAPE

**a) Công dụng:**

Định dạng lại kích thước ma trận.

**b) Cú pháp:**

b = reshape(a,m,n)

**c) Giải thích:**

b: ma trận được định dạng lại.

a: ma trận cần được định dạng.

m, n: số hàng và số cột của b.

Ma trận a phải có số phần tử là: m\*n.

**d) Ví dụ**

|  |
| --- |
|  |

a =

1 4 7 10

2 5 8 11

3 6 9 12

b = reshape(a,2,6)

b =

1 3 5 7 9 11

2 4 6 8 10 12

### Lệnh ROT90

**a) Công dụng:**

Xoay ma trận 900.

**b) Cú pháp**:

b = rot90(a)

**c) Giải thích:**

b: ma trận đã được xoay 900

a: ma trận cần xoay.

**d) Ví dụ**

|  |
| --- |
|  |

a =

1 2 3

4 5 6

7 8 9

b = rot90(a)

b =

3 6 9

2 5 8

1 4 7

### Lệnh TRACE

**a) Công dụng:**

Tính tổng các phần tử của đường chéo ma trận.

**b) Cú pháp:**

d = trace(a)

**c) Giải thích:**

d: biến chứa kết quả.

a: tên ma trận.

**d) Ví dụ**

|  |
| --- |
|  |

a =

2 8 3

4 7 1

6 9 2

d = trace(a)

d = 11

### Lệnh TRIL

**a) Công dụng:**

Lấy phân nửa dưới ma trận theo hình.

**b) Cú pháp:**

I = tril(x)

I = tril(x,k)

**c) Giải thích:**

I: tên ma trận kết quả.

k: tham số.

Nếu k = o lấy từ đường chéo trở xuống.

Nếu k = n lấy từ đường chéo trở lên n đơn vị.

Nếu k = -n lấy từ đường chéo trở xuống n đơn vị.

**d) Ví dụ**

|  |
| --- |
|  |

a =

5 9 13

6 10 14

7 11 15

8 12 16

i = tril(a)

i =

1 0 0 0

2 6 0 0

3 7 11 15

4 8 12 16

i = tril(a,0)

i =

1 0 0 0

2 6 0 0

3 7 11 0

4 8 12 16

i = tril(a,1)

i =

1 5 0 0

2 6 10 0

3 7 11 15

4 8 12 16

i = tril(a,-1)

i =

0 0 0 0

2 0 0 0

3 7 0 0

4 8 12 0

### Lệnh TRIU

**a) Công dụng:**

Lấy phân nửa trên ma trận theo hình tam giác.

**b) Cú pháp:**

I = triu(x)

I = triu(x,k)

**c) Giải thích:**

I: tên ma trận kết qủa.

k: tham số

Nếu k = 0 lấy từ đường chéo trở lên.

Nếu k = n lấy từ đường chéo trở xuống n đơn vị.

Nếu k = -n lấy từ đường chéo trở lên n đơn vị.

**d) Ví dụ**

|  |
| --- |
|  |

a =

1 5 9 13

2 6 10 14

3 7 11 15

48 12 16

I = triu(a)

I =

1 5 9 13

0 6 10 14

0 0 11 15

0 0 0 16

I = triu(a,0)

I =

1 5 9 13

0 6 10 14

0 0 11 15

0 0 0 16

I = triu(a,-1)

I =

1 5 9 13

2 6 10 14

0 7 11 15

0 0 12 16

I = triu(a,1)

I = 0 5 9 13

0 0 10 14

0 0 0 15

0 0 0 0

### Lệnh ZEROS

**a) Công dụng:**

Tạo ma trận mà giá trị của các phần tử

**b) Cú pháp:**

y = zeros(n)

y = zeros(m,n)

**c) Giải thích:**

y: tên ma trận.

n: tạo ma trận có n hàng và n cột.

m, n: tạo ma trận có m hàng, n cột.

**d) Ví dụ**

|  |
| --- |
|  |

y = zeros(3)

y =

0 0 0

0 0 0

0 0 0

y = zeros(3,7)

y =

0 0 0 0 0 0 0

0 0 0 0 0 0 0

1. 0 0 0 0 0 0

## CÁC PHÉP TÍNH ĐẠI SỐ

### Lệnh CONV

**a) Công dụng:**

Nhân hai đa thức.

**b) Cú pháp:**

c = conv(a,b)

**c) Giải thích:**

a,b: đa thức

c: tích số của a,b

Cách khai báo: sắp xếp biến theo thứ tự giảm dần của lũy thừa.

**d) Ví dụ**

|  |
| --- |
|  |

Nhân hai đa thức (3x2+4x+5).(2x3-3x2+2)

a = [0 3 4 5]

a = 0 3 4 5

b = [2 -3 0 2]

b =2 -3 0 2

c = conv(a,b)

c = 0 6 -1 -2 -9 8 10

### Lệnh CUMPROD

**a) Công dụng:**

Nhân dồn các phần tử.

**b) Cú pháp:**

cp = cumprod (a)

**c) Giải thích:**

cp: biến chứa kết qủa

a: tên của ma trận hay vector.

**d) Ví dụ**

|  |
| --- |
|  |

b = 1 9 3 4

cp =cumprod(b)

cp = 1 9 27 108

a =

1 3 5

9 1 2

4 2 1

cp = cumprod(a)

cp = 1 3 5

9 3 10

36 6 10

### Lệnh CUMSUM

**a) Công dụng:**

Cộng dồn các phần tử.

**b) Cú pháp**:

cs = cumprod(a)

**c) Giải thích:**

cs: biến chứa kết quả.

a: là tên của ma trận hay vector.

**d) Ví dụ**

|  |
| --- |
|  |

b = 1 10 1 2 5

cs = cumsum(b)

cs =1 11 12 14 19

a=

1 3 5

9 1 2

4 2 1

cs = cumsum(a)

cs =

1 3 5

10 4 7

14 6 8

### Lệnh DECONV

**a) Công dụng:**

Chia hai đa thức.

**b) Cú pháp:**

[q,r] =deconv(a,b)

**c) Giải thích:**

a,b: đa thức.

q: thương số của a, b.

r: số dư.

Cách khai báo: sắp xếp biến theo thứ tự giảm dần của lũy thừa.

**d) Ví dụ**

|  |
| --- |
|  |

Chia 2 đa thức (2x2+3x+6)/(2x+3)

a = [2 3 6]

b = [2 3]

[q,r] = deconv (a,b)

q = 1 0

r = 0 0 6

### Lệnh EXPM

**a) Công dụng:**

Tính ex

**b) Cú pháp:**

kq = expm(x)

**c) Giải thích:**

kq: biếnchứa kết qủa.

**d) Ví dụ**

|  |
| --- |
|  |

kq = expm(3)

kq = 20.0855

### Lệnh FMIN

**a) Công dụng:**

Tìm giá trị nhỏ nhất của hàm số.

**b) Cú pháp:**

x = fmin(‘fuction’,x1,x2)

**c) Giải thích:**

x: biến chứa kết quả.

fuction: tên hàm số.

x1, x2: khoảng khảo sát.

**d) Ví dụ**

|  |
| --- |
|  |

Tìm giá trị nhỏ nhất của hàm số: x3-2x-5 trong khoảng [0 2]

x =fmin(‘x.^3-2\*x-5’,0,2);

x = 0.8165

y = f(x)

y = -6.0887

### Lệnh FPLOT

**a) Công dụng:**

Vẽ đồ thị của hàm số.

**b) Cú pháp:**

fplot(‘fun’,[xmin,xmax]

**c) Giải thích:**

fun: tên hàm số.

xmin, xmax: xác định khoảng cần vẽ.

**d) Ví dụ**

|  |
| --- |
|  |

fplot(‘x.^3-2\*x-5’,[0,2]);

grid;

### Lệnh FZERO

**a) Công dụng:**

Tìm điểm 0 của hàm số.

**b) Cú pháp:**

fzero(‘fun’,x0)

**c) Giải thích:**

Điểm 0 của hàm số là điểm (0,x), đây cũng chính là nghiệm của hàm số. Nếu hàm số có nhiều nghiệm thì sẽ tìm được nghiệm gần giá trị x0.

fun: tên hàm số.

**c) Ví dụ**

|  |
| --- |
|  |

Tìm giá trị 0 của hàm số: x2-5x+3.

Trước tiên ta khai báo hàm số f trong tập tin f.m: (xem thêm lệnh function)

function y = f(x);

y = x.^2-5\*x+3;

Sau đó, tạo tập tin gt0.m:

x = 0:10;

% Giá trị x0 = 0

z = fzero(‘f’,0);

sprinf(‘z = %3f’,z)

z = 0.382

% Giá trị x0 = 2

z = fzero(‘f’,2);

sprintf(‘z = %.3f’,z)

z = 2.618

% Vẽ đồ thị hàm số minh họa:

z = fzero(‘f’,0);

fplot(‘f’,[0,5];

grid;

hold on;

plot(z,0,‘o’);

hold off

### Lệnh MAX

**a) Công dụng:**

Tìm giá trị lớn nhất.

**b) Cú pháp:**

m = max(x)

[m,i] = max(x) v = max(x,y)

**c) Giải thích:**

x,y,v:tên vector.

m: giá trị lớn nhất.

i: vị trí của m.

Nếu x là ma trận tìm ra giá trị lớn nhất của mỗi cột.

**d) Ví dụ**

|  |
| --- |
|  |

x = 3 5 2 1 4

m= max(x)

m = 5

[m,i] = max(x)

m =5

i =2

y = 1 6 8 -5 3

v =max(x,y)

v = 3 6 8 1 4

b =

3 6 2

1 7 9

2 8 1

m = max(b)

m = 3 8 9

[m,i] = max(b)

m= 3 8 9

i = 1 3 2

a =

0 3 6

7 1 1

4 6 8

v = max(a,b)

v =

3 6 6

7 7 9

4 8 8

### Lệnh MEAN

**a) Công dụng:**

Tìm giá trị trung bình.

**b) Cú pháp:**

Mô hình = mean(a)

**c) Giải thích:**

m: biến chứa kết qủa.

a: tên vector hay ma trận cần tính giá trị trung bình.

Nếu a là ma trận thì tính giá trị trung bình của mỗi cột.

**d) Ví dụ**

|  |
| --- |
|  |

b = 1 10 1 2 5

m = mean(b)

m = 3.8000

a =

1 3 5

9 1 2

4 2 1

m = mean(a)

m = 4.6667 2.0000 2.6667

### Lệnh MIN

**a) Công dụng:**

Tím giá trị nhỏ nhất

**b) Cú pháp:**

m = min(x)

[m,i] = min(x)

v = min(x,y)

c) Giải thích:

x,y,v: tên vector.

m: là giá trị lớn nhất.

i: là vị trí của m.

Nêú x là ma trận tìm ra giá trị nhỏ nhất trong mỗi cột.

**d) Ví dụ**

|  |
| --- |
|  |

x = 3 5 2 1 4

m = min(x)

m = 1

i =4

y =1 6 8 -5 3

v = min(x,y)

v = 1 5 2 -5 3

b =

3 6 2

1 7 9

2 8 1

m = min(b)

m = 1 6 1

i = 2 1 3

a =

0 3 6

7 1 1

4 6 8

v = min(a,b)

v =

0 3 2

1 1 1

2 6 1

### Lệnh PROD

**a) Công dụng:**

Nhân các phần tử.

**b) Cú pháp:**

p = prod(x)

**c) Giải thích:**

p: biến chứa kết quả.

x: tên ma trận hay dãy số.

Nếu là ma trận nhân từng phần tử cuả mỗi cột.

**d) Ví dụ**

|  |
| --- |
|  |

a = 2 3 4 5

p = prod(a)

p = 20

b =

2 2 3

5 6 4

7 5 4

p =prot(b)

p =70 60 48

### Lệnh ROOTS

**a) Công dụng:**

Tìm nghiệm của đa thức.

**b) Cú pháp:**

r = roots(p)

**c) Giải thích:**

r: biến chứa kết quả.

p: tên biểu thức.

**d) Ví dụ**

|  |
| --- |
|  |

Tìm nghiệm cuả phương trình: x2-1 =0

p = [1 0 -1]

r = roots(p);

disp(r)

-1.0000

1.0000

### Lệnh SORT

**a) Công dụng:**

Sắp xếp mảng hay ma trận theo thứ tự tăng dần.

**b) Cú pháp:**

kq = sort(x)

[kq,i] = sort(x)

**c) Giải thích:**

kq: biến chưá kết quả.

i: số thứ tự cuả phần tử trước khi sắp xếp.

Nếu x là ma trận thì sắp xếp theo thứ tự tăng dần của từng cột.

**d) Ví dụ**

|  |
| --- |
|  |

a = 2 8 5 6 -3 9

kq = sort(a)

kq = -3 2 5 6 8 9

[kq,i] = sort(a)

kq = -3 2 5 6 8 9

i = 5 1 3 4 2 6

b =

3 4 -4

2 -3 5

1 6 2

kq =sort(b)

kq =

1 -3 -4

2 4 2

3 6 5

[kq,i] = sort(b)

kq =

1 -3 -4

2 1 2

3 6 5

i =

3 2 1

2 1 3

1 3 2

### Lệnh SUM

**a) Công dụng:**

Tính tổng của các phần tử.

**b) Cú pháp:**

s = sum(x)

**c) Giải thích:**

s: là biến chứa kết quả.

x: là tên ma trận.

Nếu x là ma trận thì s là tổng của các cột.

**d) Ví dụ**

|  |
| --- |
|  |

a = 2 8 5 6 -3 9

s = sum(a)

s = 27

b =

3 4 -4

2 -3 5

1 6 2

s = sum(b)

s = 6 7 3

## TẬP LỆNH ĐỒ HỌA

### 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’, [2 5])

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

\*/ Ví dụ:

axes(‘ylim’, [2 5])

định giá trị trên cả hai trục

axes(‘xlim’, [min,max], ‘ylim’,[min,max])

### 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.

### 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;

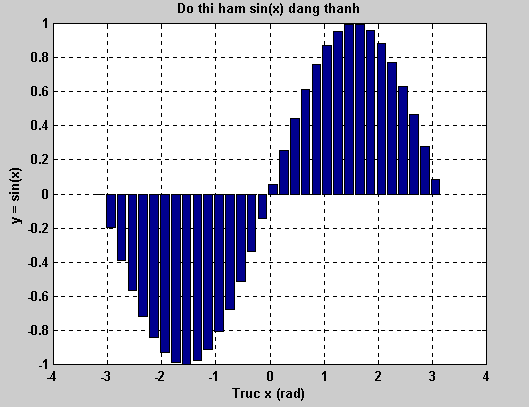
bar(x,sin(x));

grid on

title(‘Do thi ham sin(x) dang thanh’)

xlabel(‘truc x (rad)’)

ylabel(‘y = sin(x)’)



### 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

### Lệnh CLF

**a) Công dụng:**

Xóa hình ảnh (đồ thị) hiện tại.  
**b) Cú pháp:**

clf

### Lệnh CLOSE

**a) Công dụng:**

Đóng hình ảnh (đồ thị) hiện tại.  
**b) Cú pháp:**

close

### 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 |

### Lệnh FIGURE

**a) Công dụng:**

Tạo mới hình ảnh (đồ thị).  
**b) Cú pháp:**

figure

### 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 cga.

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. |

### 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.

### Lệnh PLOT

**a) Công dụng:**

Vẽ đồ thị tuyến tính trong không gian 2 chiều.  
**b) Cú pháp:**

plot(x,y)

plot(x,y,’linetype’)

**c) Giải thích:**

x,y: vẽ giá trị x theo giá trị y.

linetype: kiểu phần tử tạo nên nét vẽ bao gồm 3 thành phần:

- Thành phần thứ nhất là các ký tự chỉ màu sắc:

|  |  |
| --- | --- |
| 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 π/8

x = 0:pi/8:2\*pi;

y = sin(x);

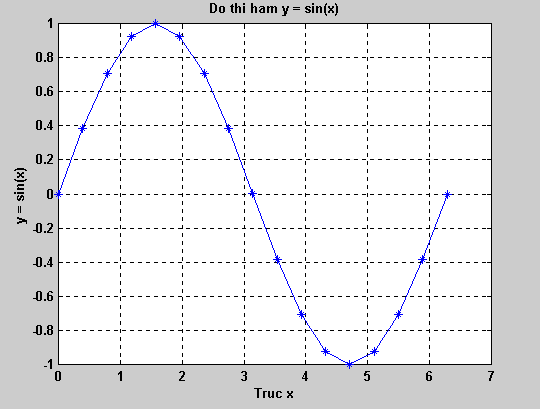
plot(x,y, ‘b-\* ’)

ylabel(‘y = sin(x)’)

xlabel(‘Truc x’)

title(‘Do thi ham y = sin(x)’)

grid on



### Lệnh SUBPLOT

**a) Công dụng:**

Tạo các trục trong một phần của cửa sổ đồ họa.  
**b) Cú pháp:**

subplot(m,n,p)

subplot(mnp)

**c) Giải thích:**

subplot(m,n,p) hoặc subplot(mnp)thành cửa sổ đồ họa thành m×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×n thì sẽ xuất hiện một thông báo lỗi.

**d) Ví dụ:**

Chia cửa sổ đồ họa thành 2×3 vùng và hiển thị trục của cả 6 vùng.

subplot(231)

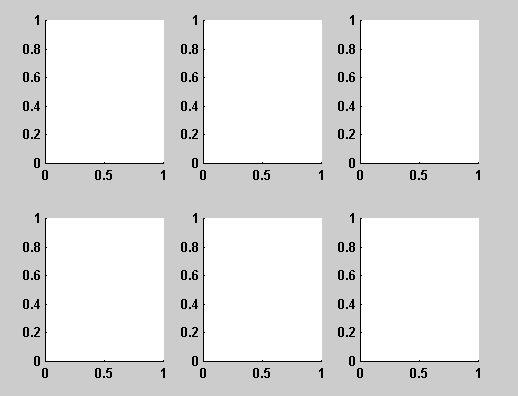
subplot(232)

subplot(233)

subplot(234)

subplot(235)

subplot(236)



### 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:**

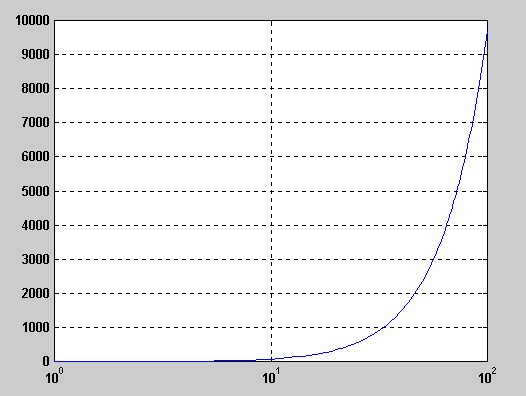
Vẽ đồ thị hàm y = x2 – 3x + 2 theo trục logarith của x.

x = 0:100;

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

semylogx(x,y,’b’)

grid on



### Lệnh POLAR

**a) Công dụng:**

Vẽ đồ thị trong hệ trục tọa độ cực.  
**b) Cú pháp:**

polar(theta,rno)

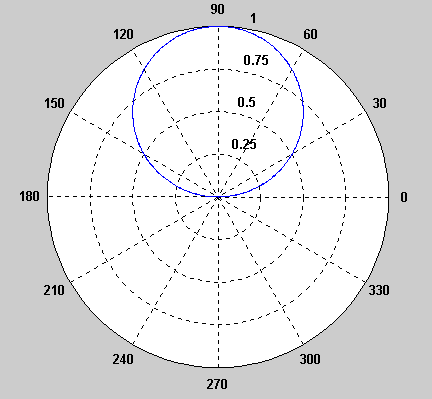
**c) Giải thích:**

Vẽ giá trị x theo giá trị y.

**d) Ví dụ:**

t = -pi:0.01:pi;

polar(t, sin(t))



### 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:

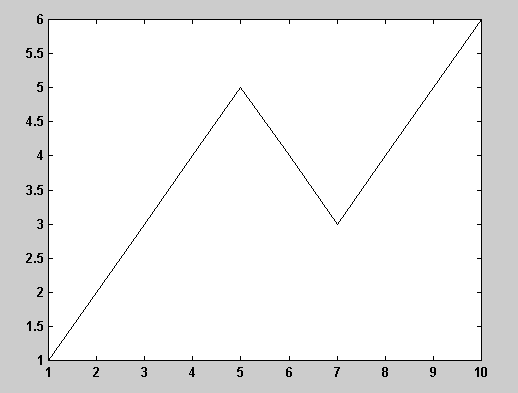
|  |  |  |  |
| --- | --- | --- | --- |
| Cú pháp | PropertyName | PropertyValue | Giải thích |
| Set(h,’Marker’,’+’) | Marker | -, --, :, -. , o, ×, +, \* | Chọn kiểu phần tử |
| Set(h,’LineWidth’,1) | LineWidth | 1, 2, 3,… | Độ dày nét vẽ |
| Set(h,’MarkerSize’,9) | MarkerSize | 1, 2, 3,… | Kích thước 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’)



### 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:**

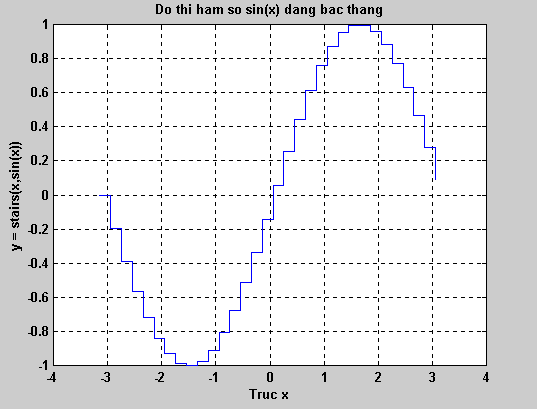
x = -pi:0.2:pi;

stairs(x,sin(x))

xlabeL(‘Truc x’)

ylabel(‘y = stairs(x,sin(x)’)

grid on



### Lệnh TITLE

**a) Công dụng:**

Đặt tiêu đề cho đồ thị.  
**b) Cú pháp:**

title(‘text’)

**c) Giải thích:**

text: tên tiêu đề.

### 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

### 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 cửa sổ đồ họa giữa trắng và đen.

whitebg(‘color’) chuyển màu nền 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).

**BT3c:** được viết trong **BT3c.m**. Bài tập này tổng hợp từ các sách **‘The Student Edition of MATLAB’, ‘The MATLAB 5. Handboox’, ‘Ứng dụng MATLAB trong điều khiển tự động’**

%BT3c: VE QUA DIA CAU

[x,y]=meshgrid(-3:0.1:3);

z=peaks(x,y);

meshc(x,y,z)

pause

k=5;

n=2^k-1;

[x,y,z]=sphere(n);

c=hadamard(2^k);

surf(x,y,z,c);

colormap([1 1 0;0 1 1])

pause

t=0:pi/10:2\*pi;

[x,y,z]=cylinder(2+cos(t));

surf(x,y,z)

pause

[x,y,z]=cylinder(1:10);

surfnorm(x,y,z)

pause

[x,y,z]=meshgrid(-2:.2:2,-2:.2:2,-2:.2:2);

v=x.\*exp(-x.^2-y.^2-z.^2);

slice(v,[5 15 21],21,[1 10],21)

pause

[X,Y]=meshgrid(-3:.5:3);

Z=peaks(X,Y);

[XI,YI]=meshgrid(-3:.25:3);

ZI=interp2(X,Y,Z,XI,YI);

mesh(X,Y,Z), hold, mesh(XI,YI,ZI+15)

hold off

axis([-3 3 -3 3 -5 20])

pause

syms x y

ezsurf(real(atan(x+i\*y)))

[x,y]=meshdom(-12:.6:12,-12:.6:12);

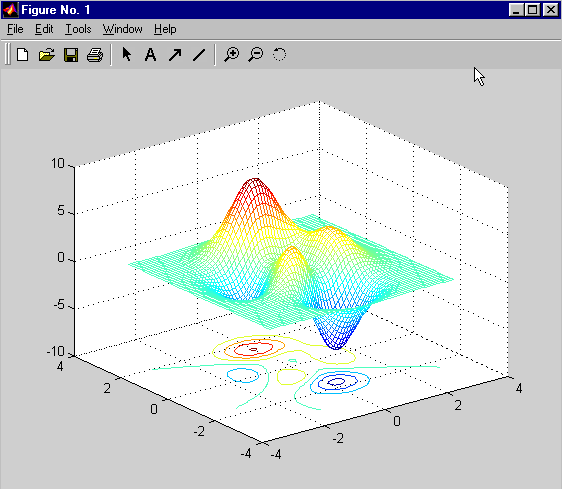
r=sqrt(x.^2+y.^2);

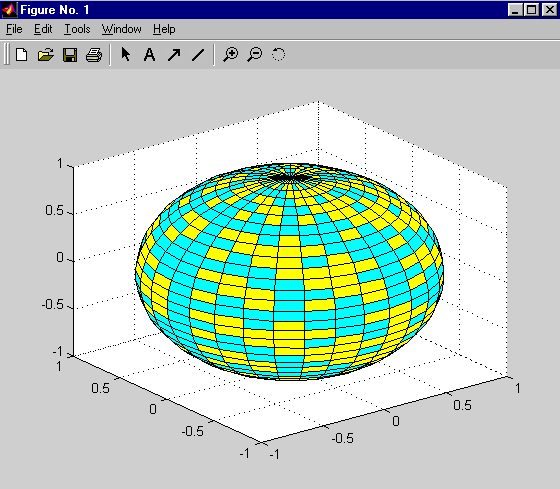
z=bessel(0,r);

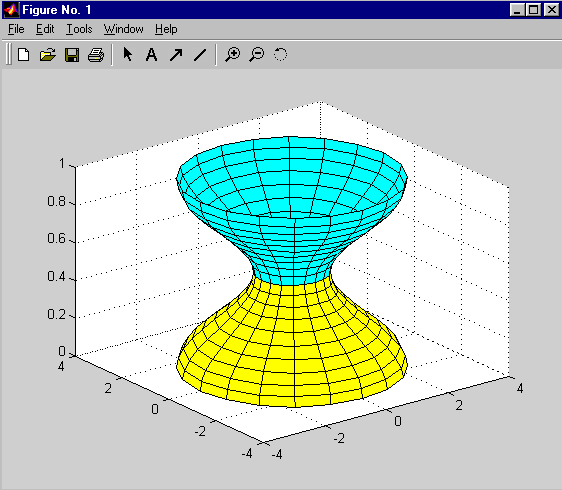
m=[-45 60];

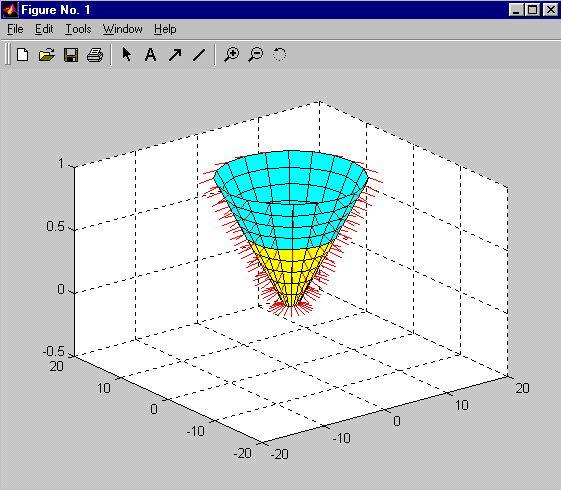
mesh(z,m)

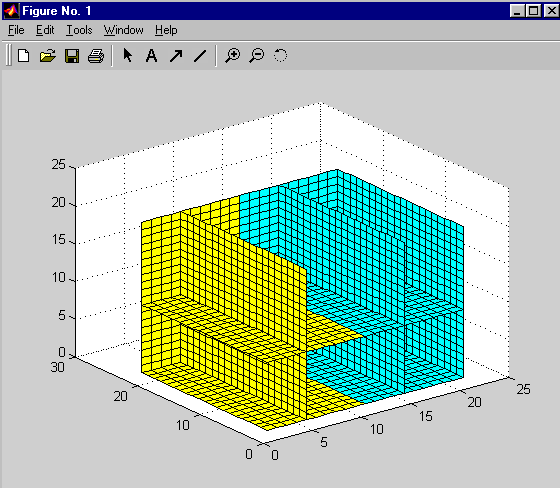
Khi chạy chương trình ta lầ lượt có kết quả:

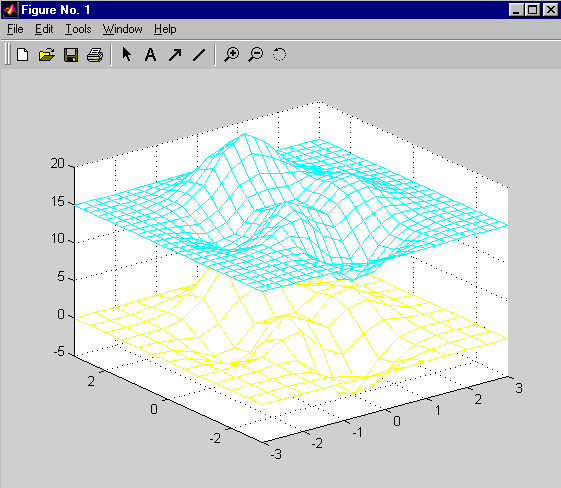


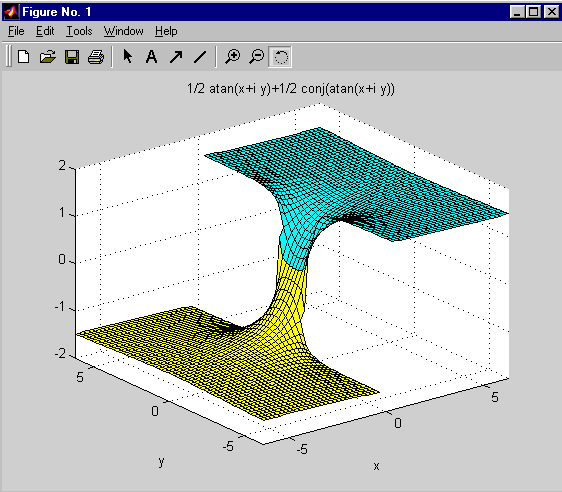


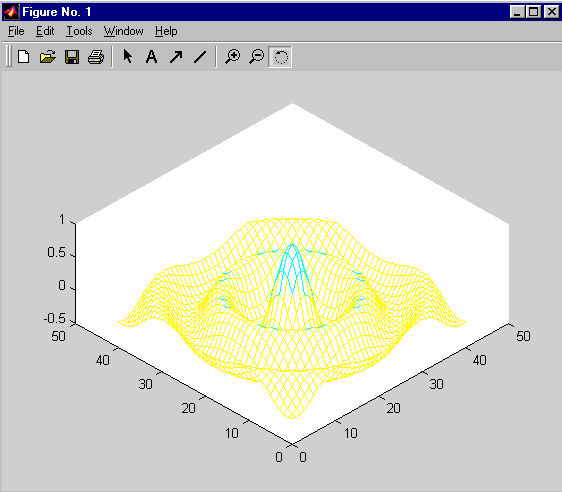












Cũng là hàm **bessel** nhưng ta khảo sát riêng 1 bài:

%hm bessel

[x,y]=meshdom(-12:.6:12,-12:.6:12);

r=sqrt(x.^2+y.^2);

z=bessel(0,r);

m=[-45 60];

mesh(z,m)

