

Phần A: CÁC KHÁI NIỆM CƠ BẢN VỀ MATLAB

Chương 1

CƠ SỞ VỀ MATLAB

1.1. Khởi động MATLAB

Để khởi động làm việc với MATLAB, ta nhấp đúp vào biểu tượng của MATLAB. Màn hình nền xuất hiện bao gồm các cửa sổ:

Cửa sổ lệnh **Command Windows**: Đây là cửa sổ chính của MATLAB. Tại đây ta thực hiện toàn bộ việc nhập dữ liệu và xuất kết quả tính toán. Dấu nhắc `>>` để gõ các lệnh.

Có một số lệnh tổng quát liên quan đến cửa sổ này như sau:

clc: xóa cửa sổ lệnh.

home: di chuyển con chạy lên góc trên trái của cửa sổ (khi chạy chương trình).

help: trợ giúp thông tin về một mục nào đó

echo, echo on/off: tắt mở hiển thị các dòng của file m khi chạy chương trình.

edit: gọi chương trình soạn file *.m.

type tên_file: đọc nội dung file *.m.

demo: gọi chương trình demo.

quit, exit: thoát chương trình MATLAB.

Ngoài ra còn có một số phím hỗ trợ thao tác:

Phím	Chức năng	Phím	Chức năng
↑	Gọi lại hàng trước	End	Về cuối hàng
↓	Gọi lại hàng kế	Esc	Xóa hàng
←	Lùi lại một ký tự	Delete	Xóa ký tự ở con chạy
→	Tới một ký tự	Backspace	Xóa ký tự trước con chạy
Ctrl →	Qua phải một từ	Ctrl K	Xóa đến cuối hàng
Ctrl ←	Qua trái một từ	Shift home	Đánh dấu đến đầu hàng
Home	Về đầu hàng	Shift end	Đánh dấu đến cuối hàng

Cửa sổ lịch sử lệnh Command History: liệt kê các lệnh đã sử dụng trước đó kèm thời gian bắt đầu. Có thể lặp lại lệnh cũ bằng cách nhấp chuột kép vào lệnh đó. Cũng có thể cắt, sao hoặc xóa cả nhóm lệnh hoặc từng nhóm lệnh riêng rẽ.

Cửa sổ không gian làm việc Workspace Browser: cho biết các biến sử dụng trong chương trình.

Cửa sổ Launch Pad: cho phép người sử dụng truy cập nhanh các công cụ, tài liệu của MATLAB.

Cửa sổ thư mục hiện tại Current Directory Browser: cho biết thư mục hiện tại đang sử dụng. Người sử dụng có thể nhanh chóng nhận biết, chuyển đổi thư mục hiện tại của môi trường công tác, mở File, tạo thư mục mới.

1.2. Sử dụng lệnh trực tiếp với MATLAB

Những lệnh của MATLAB có thể được sử dụng trực tiếp hoặc chạy từ các file lệnh, nhưng nói chung, chúng ta thường sử dụng cả hai phương pháp trên.

Ví dụ: Giải phương trình bậc hai $ax^2 + bx + c = 0$

Các nghiệm của phương trình có dạng:
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Vì MATLAB là một chương trình tính toán số nên chúng ta phải xác định các giá trị của a, b, c. Dấu = được sử dụng để gán một giá trị nào đó cho một biến. Ta nhập các giá trị của a, b và c như sau (gõ phím Enter ở cuối mỗi hàng)

```
>>a=2
```

```
a=
```

```
2
```

```
>>b=5
```

```
b=
```

```
5
```

```
>>c=-3
```

```
c=
```

```
-3
```

Nếu có sử dụng dấu ; ở cuối dòng thì MATLAB sẽ không hiển thị lại giá trị vừa nhập.

Ta hướng dẫn MATLAB tính nghiệm thứ nhất bằng cách gõ tiếp:

```
x1=(-b+sqrt(b^2-4*a*c))/(2*a)
```

```
x1=
```

```
0.5000
```

Tính nghiệm thứ hai bằng cách gọi lại lệnh trước và thay dấu + bằng dấu -

```
x2=(-b-sqrt(b^2-4*a*c))/(2*a)
```

$x2 =$
 -3

Các phép toán cơ bản để tạo nên biểu thức trong MATLAB:

Phép tính	Ký hiệu	Ví dụ
Phép cộng $a + b$	+	$5 + 3$
Phép trừ $a - b$	-	$7 - 4$
Phép nhân $a . b$	*	$18 * 24$
Phép chia $a \div b$	/ hoặc \	$56/8 = 8 \backslash 56$
Phép lũy thừa a^b	^	$5 ^ 2$

1.3. Sử dụng lệnh từ các file lệnh

Những lệnh của MATLAB có thể được đưa vào một file. Sau đó hướng dẫn MATLAB làm việc với các lệnh đó. Tên của file phải được bắt đầu bằng một ký tự và có phần mở rộng là .m.

Dùng lệnh edit hoặc vào menu File - New - M_file môi trường soạn thảo Editor/Debugger sẽ xuất hiện. Trên màn hình soạn thảo, ta gõ các lệnh MATLAB:

```
a=2;
b=5;
c=-3;
x1=(-b+sqrt(b^2-4*a*c))/(2*a)
x2=(-b-sqrt(b^2-4*a*c))/(2*a)
```

sau khi gõ xong lệnh, ta lưu tập tin với tên vidu.m rồi thoát khỏi môi trường soạn thảo để trở về Command Window. Để chạy các lệnh trong tập tin vidu.m ta chỉ cần gõ tên tập tin:

```
>>vidu
x1 =
    0.5000
x2 =
   -3
```

1.4. Dòng nhắc gán giá trị các biến

Để thay đổi các giá trị a, b, c ta phải soạn thảo lại file vidu.m rồi chạy lại. Ta có thể sửa lại chương trình để có dòng nhắc nhập a, b, c với các lần lượt chạy chương trình khác nhau.

```
a=input('nhap vao gia tri a = ');
b=input('nhap vao gia tri b = ');
c=input('nhap vao gia tri c = ');
x1=(-b+sqrt(b^2-4*a*c))/(2*a)
```

$$x2 = (-b - \sqrt{b^2 - 4*a*c}) / (2*a)$$

Khi chạy chương trình ta sẽ được kết quả như sau:

>> vidu

nhap vao gia tri a = 1

nhap vao gia tri b = 0

nhap vao gia tri c = -2

x1 =

1.4142

x2 =

-1.4142

1.5. Cách tạo một hàm

Mỗi một file hàm của MATLAB (M_file) đều được bắt đầu với khai báo như sau:

Function [tên kết quả] = tên hàm (danh sách các biến)

File.m thường lấy tên là tên của hàm.

Phần thân của chương trình trong hàm là các lệnh của MATLAB thực hiện việc tính toán giá trị của đại lượng được nêu trong phần tên kết quả theo các biến được nêu trong phần danh sách các biến. Các biến chỉ có tác dụng nội trong hàm vừa được khai báo. Tên của các biến được ngăn cách nhau bằng dấu phẩy.

Ví dụ: tạo hàm giải phương trình bậc hai, tên tập tin hàm được đặt là *bachai.m*

function [x1,x2]=bachai(a,b,c)

*delta=b^2-4*a*c;*

*x1=(-b+sqrt(delta))/(2*a);*

*x2=(-b-sqrt(delta))/(2*a);*

Giải phương trình bậc hai với a = 1, b = 0, c = -2, ta gõ:

>> [x1,x2]=bachai(1,0,-2)

x1 =

1.4142

x2 =

-1.4142

1.6. Sử dụng hàm có sẵn

MATLAB có thư viện hàm rất lớn, muốn biết cách sử dụng các hàm như thế nào trong màn hình Command Window ta gõ lệnh:

Help + tên hàm

1.7. Vẽ các hàm

Khi muốn vẽ một hàm nào đó, phải xác định hàm đó trong một file.m, sau đó sử dụng lệnh:

fplot('tên file',[khoảng vẽ])

Ví dụ: Vẽ hàm bậc hai $4x^2 + 6x - 7$ trong đoạn $[-6,6]$.

+ Tạo file bachai.m:

```
function y=bachai(x)
```

```
a=4;
```

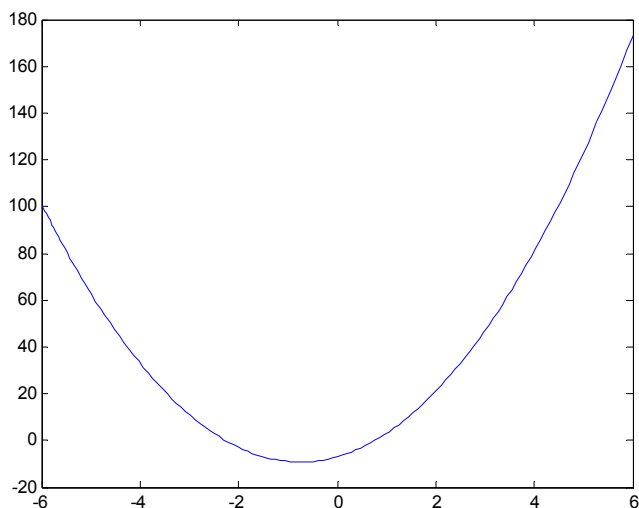
```
b=6;
```

```
c=-7;
```

```
y=a*x^2+b*x+c;
```

+ Vẽ hàm:

```
fplot('bachai',[-6,6])
```



1.8. Lưu và lấy dữ liệu

Ta có thể tạo một file dữ liệu, sau đó khi cần dùng thì lấy ra.

Ví dụ: Tạo một ma trận A, sau đó lưu file này với tên là dulieu

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

```
>>Save dulieu
```

Khi cần sử dụng dữ liệu này, ta gõ lệnh:

```
>>load dulieu
```

```
>> A
```

```
A =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

1.9. Các toán tử logic và quan hệ

Đối với các số thì trong toán tử và quan hệ quy định các số khác không là True còn số không là False. Kết quả của phép toán logic và quan hệ đưa ra là 1 cho True, 0 cho False.

1.9.1. Toán tử quan hệ

Toán tử quan hệ MATLAB bao gồm các phép so sánh:

Toán tử quan hệ	Ý nghĩa
<	nhỏ hơn
<=	nhỏ hơn hoặc bằng
>	lớn hơn
>=	lớn hơn hoặc bằng
==	bằng
~=	gần bằng

Ví dụ:

```
>> 12.2 > 12
```

```
ans =
```

```
1
```

```
>> 1 ~ 1
```

```
ans =
```

```
0
```

1.9.2. Toán tử logic

Toán tử logic cung cấp một cách diễn đạt mối quan hệ phủ định hay tổ hợp:

Toán tử logic	Ý nghĩa
&	AND
	OR
~	NOT

Biểu thức logic với các toán tử tổ hợp được xác định theo qui luật sau:

A	B	~A	A B	A&B
False	False	True	False	False
False	True	True	True	False
True	False	False	True	False
True	True	False	True	True

Đôi khi gặp những biểu thức phức tạp chứa cả toán tử & lẫn toán tử |, MATLAB sẽ có qui luật về trình tự xử lý như sau:

Đầu tiên là xử lý các tính toán số học. Sau đó, các toán tử logic được xem xét từ trái sang phải. Những biểu thức con có trong biểu thức cần xử lý (có các toán tử & và |) cũng được xem xét từ trái sang phải với các toán tử & được xử lý trước các toán tử |

Ví dụ:

```
>> A = 1:5;
```

```
>> B = (A > 2) & (A < 5)
```

```
B =
```

```
0 0 1 1 0
```

1.10. Các câu lệnh điều kiện

Dạng đơn giản nhất của lệnh điều kiện trong MATLAB là:

if biểu thức điều kiện

Khối các lệnh được thực hiện

end

Nếu biểu thức điều kiện là true thì thực hiện các lệnh tiếp theo, nếu không thì thực hiện các lệnh tiếp sau end.

Trong cấu trúc lệnh điều kiện, ta có thể sử dụng cấu trúc if với nhiều phần thân cấu trúc như sau:

if biểu thức điều kiện 1

Khối các lệnh được thực hiện nếu điều kiện 1 là đúng

elseif biểu thức điều kiện 2

Khối các lệnh được thực hiện nếu điều kiện 2 là đúng

else

Khối các lệnh được thực hiện nếu không có điều kiện nào đúng

end

Ví dụ: Căn cứ vào số điểm để xét kết quả học tập của một học sinh.

```
diem=input('nhap vao so diem = ');
```

```
if (diem>=0)&(diem<=10)
```

```
    if diem<=4
```

```
        disp('loai yeu')
```

```
    elseif (diem>=5)&(diem<=6)
```

```
        disp('loai trung binh')
```

```
    elseif (diem>=7)&(diem<=8)
```

```
        disp('loai gioi')
```

```
    elseif (diem>=9)&(diem<=10)
```

```
        disp('loai xuat sac')
```

```
    end
```

```
else ('so lieu khong hop le')
```

```
end
```

1.11. Vòng lặp

1.11.1. Vòng lặp *for*

Vòng lặp for cho phép một nhóm lệnh thực hiện lặp lại một số lần cố định. Cú pháp của vòng lặp for như sau:

```
for i=i1:Δi:i2
```

```
    Khối các lệnh
```

```
end
```

Trong vòng lặp trên i1 và i2 lần lượt là các giá trị bắt đầu và kết thúc của i, Δi là bước nhảy (nếu không đặt bước nhảy thì giá trị mặc định của bước nhảy bằng 1). Mỗi lần lặp sẽ kết thúc khi gặp từ khóa end. Đầu tiên MATLAB sẽ xác định có bao nhiêu lần lặp và sau đó mới tiến hành thực thi các lệnh trong vòng lặp với các giá trị của i. Vì vậy, nếu như $i2 < i1$ với $\Delta i > 0$ thì MATLAB sẽ không thực thi vòng lặp.

Ví dụ:

```

for i=1:10
    x(i)=sin(i*pi/10);
end
>> x
x =
    Columns 1 through 8
    0.3090    0.5878    0.8090    0.9511    1.0000    0.9511    0.8090
    0.5878
    Columns 9 through 10
    0.3090    0.0000

```

1.11.2. Vòng lặp While

Vòng lặp While thực hiện lặp lại một số lần cố định, nhưng không biết trước được số lần lặp lại. Cú pháp của vòng lặp While như sau:

```

while biểu thức điều kiện
    Khối các lệnh
end

```

Ví dụ: Tính giá trị đặc biệt của eps trong MATLAB

```

num=0;
>> EPS=1;
>> while (EPS+1)>1
    EPS=EPS/2;
    num=num+1;
end
>> num
num =
    53
>> EPS=EPS*2
EPS =
    2.2204e-016

```

1.12. Các hàm toán học thông thường

Tương tự như hầu hết các máy tính kỹ thuật, MATLAB có thể đưa ra rất nhiều các hàm toán học, kỹ thuật thông dụng, ngoài ra MATLAB còn cung cấp hàng trăm các hàm đặc biệt và thuật toán, nó rất hữu ích để giải quyết các vấn đề khoa học. Tất cả các hàm được liệt kê trong menu help. Ở đây chỉ đề cập đến các hàm thông dụng nhất.

1.12.1. Hàm toán học cơ bản

abs(x)	Hàm tính giá trị tuyệt đối của x
sqrt(x)	Hàm tính căn bậc hai của x
round(x)	Làm tròn x về số nguyên gần nhất

<i>fix(x)</i>	Làm tròn số x về 0
<i>floor(x)</i>	Làm tròn về phía $-\infty$
<i>ceil(x)</i>	làm tròn về phía ∞
<i>sign(x)</i>	Hàm cho giá trị là -1 nếu x nhỏ hơn 0, giá trị bằng 0 nếu x bằng 0, có giá trị là 1 nếu x lớn hơn 0
<i>rem(x,y)</i>	Hàm trả lại số dư của phép chia x cho y
<i>exp(x)</i>	Hàm tính giá trị của e^x
<i>log(x)</i>	Hàm tính giá trị $\ln(x)$
<i>log10(x)</i>	Hàm tính giá trị $\log_{10}(x)$

1.12.2. Hàm lượng giác cơ bản

Đơn vị của các tham số phải là radian

<i>sin(x)</i>	Tính sin của góc x
<i>cos(x)</i>	Tính cos của góc x
<i>tan(x)</i>	Tính tan của góc x
<i>asin(x)</i>	Tính arcsin của x, khi x nằm trong khoảng $[-1,1]$, hàm trả lại góc có giá trị radian trong khoảng $-\pi/2$ đến $\pi/2$
<i>acos(x)</i>	Tính arccos của x, khi x nằm trong khoảng $[-1,1]$, hàm trả lại góc có giá trị radian trong khoảng 0 đến π
<i>atan(x)</i>	Tính arctang của x trong khoảng $-\pi/2$ đến $\pi/2$
<i>atan2(x,y)</i>	Tính arctang của y/x trong khoảng $-\pi$ đến π , tùy thuộc vào dấu của x và y

1.12.3. Các hàm hyperbolic

<i>sinh(x)</i>	Hàm tính hyperbolic sin của x
<i>cosh(x)</i>	Hàm tính hyperbolic cos của x
<i>asinh(x)</i>	Hàm tính nghịch đảo của hyperbolic sin của x
<i>acosh(x)</i>	Hàm tính nghịch đảo của hyperbolic cos của x
<i>atanh(x)</i>	Hàm tính nghịch đảo của hyperbolic tang của x

1.13. Định dạng số

Các phép tính trong MATLAB được thực hiện với độ chính xác cao, ta có thể định dạng cho các số xuất ra màn hình tùy từng yêu cầu cụ thể.

Ta lấy ví dụ với số $a = -2/3$

Lệnh của MATLAB	a	Chú thích
format short	-0.6667	5 số
format long	-0.666666666666667	15 số
format short e	-6.6667e-001	5 số với số mũ
format long e	-6.666666666666666e-001	15 số với số mũ
format short g	-0.66667	Chính xác hơn format short hoặc

		format short e
format long g	-0.666666666666667	Chính xác hơn format long hoặc format long e
format hex	bfe5555555555555	Hệ cơ số 16
format bank	-0.67	Hai số hệ 10
format +	-	Dương, âm hoặc bằng không
format rat	-2/3	Dạng phân số

Một chú ý quan trọng là MATLAB không thay đổi số khi định lại khuôn dạng hiển thị được chọn, mà chỉ thay đổi màn hình thay đổi.