

TÀI LIỆU THỰC HÀNH LABORATORY

Mục lục

1	Matlab cơ bản	3
1.1	Giới thiệu Matlab	3
1.2	Phép toán, biến, vector, ma trận	6
1.2.1	Biến	6
1.2.2	Phép toán	7
1.2.3	Vector	10
1.2.4	Ma trận	14
1.3	Biểu thức Logic	17
1.3.1	Các toán tử logic	17
1.3.2	Vec-tơ và biểu thức logic	18
1.3.3	Các hàm logic: All, Any và Find	19
1.4	Lệnh điều kiện và vòng lặp	19
1.4.1	Lệnh IF	19
1.4.2	Lệnh FOR	20
1.4.3	Lệnh WHILE	21
1.4.4	Lệnh SWITCH ... CASE	22
1.4.5	Script và Hàm	23
1.5	Vẽ đồ thị	24
1.5.1	Vẽ đồ thị trong 2-D	24
1.5.2	Vẽ đồ thị trong 3-D	29
2	Đại số tuyến tính	33
2.1	Các phép toán ma trận, các phép biến đổi sơ cấp	33
2.1.1	Các phép toán ma trận	33
2.1.2	Các phép biến đổi sơ cấp	37

2.2	Ma trận nghịch đảo, Phương trình ma trận và Hệ phương trình tuyến tính	39
2.2.1	Ma trận nghịch đảo	39
2.3	Bài tập	41
2.3.1	Ma trận giả nghịch đảo	41
2.4	Bài tập	43
2.4.1	Giải phương trình ma trận	44
2.5	Bài tập	44
2.5.1	Hệ phương trình tuyến tính	45
2.5.2	Đưa về dạng ma trận	45
2.5.3	Sử dụng tính toán symbolic	47
2.6	Bài tập	48
2.7	Định thức, giải hệ phương trình tuyến tính bằng định thức	49
2.7.1	Định thức	49
2.7.2	Giải hệ phương trình tuyến tính bằng định thức	50
2.8	Bài tập	50
2.9	Đa thức đặc trưng, trị riêng và vectơ riêng	51
2.10	Bài tập	52
3	Giải tích hàm một biến	53
3.1	Các phép toán tập hợp	53
3.1.1	Định nghĩa tập hợp và cách khai báo tập hợp trong Matlab	53
3.1.2	Các phép toán trong tập hợp	53
3.1.3	BÀI TẬP	56
3.2	symbolic math cơ bản	57
3.3	Các bài toán dãy số và chuỗi số	58
3.3.1	Khái niệm về dãy số, chuỗi số và cách khai báo trong matlab	58
3.3.2	Một số hàm về xử lý dãy số và chuỗi số trong Matlab	59
3.3.3	Bài tập	60
3.4	Các bài toán vi phân hàm một biến	61
3.4.1	Vi phân hàm một biến	61
3.5	Các bài toán giới hạn hàm số và tính liên tục của hàm số	61
3.5.1	Giới hạn của hàm số	61
3.5.2	Sự liên tục của hàm số	62
3.6	Các bài toán tích phân hàm một biến	65
3.6.1	Tích phân bất định	65

3.6.2	Tích phân xác định	65
3.6.3	Tích phân số	66
3.6.4	Các hàm trong Matlab dùng cho bài toán vi phân hàm một biến	68
3.6.5	Bài tập	68

CHƯƠNG 1

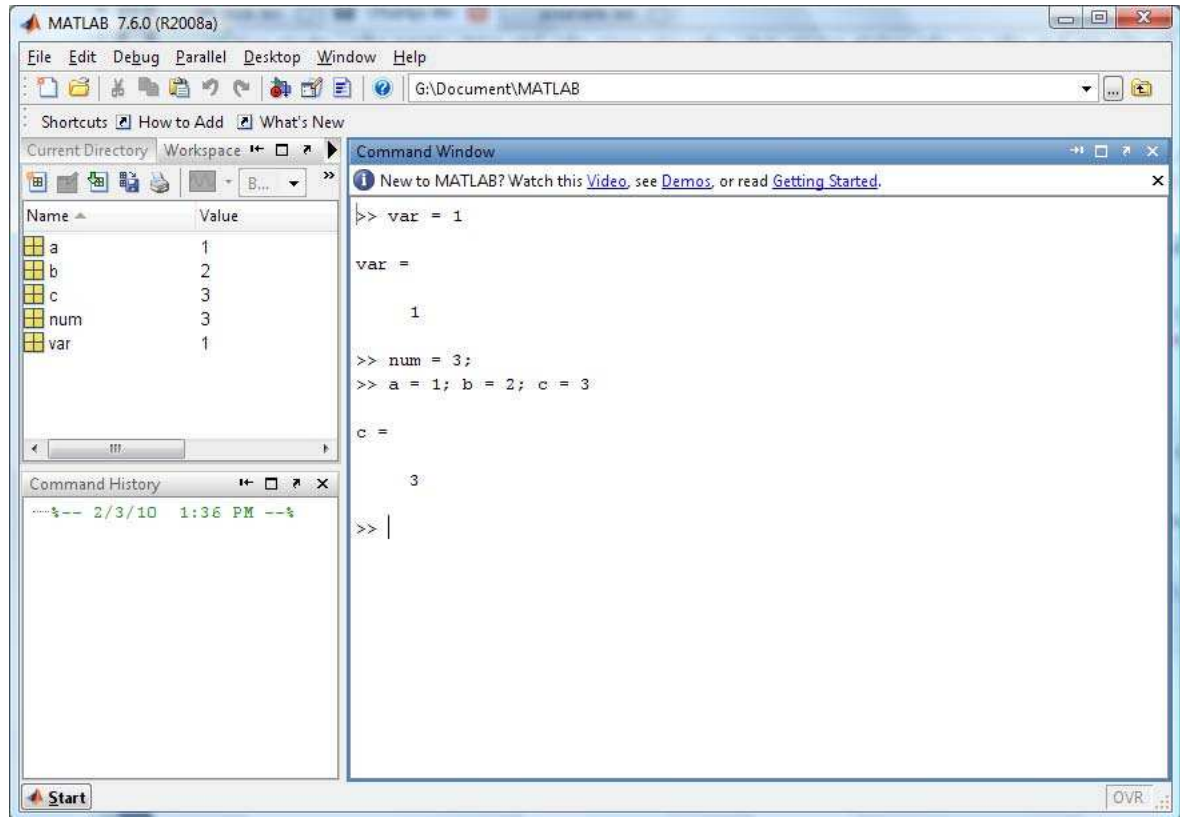
Matlab cơ bản

1.1 Giới thiệu Matlab

Matlab là một ngôn ngữ lập trình cấp cao được sử dụng rộng rãi trong môi trường học thuật và công nghệ. Matlab được xem là lựa chọn ưu tiên vì có khả năng hỗ trợ tối ưu cho việc nghiên cứu cũng như dạy học đối với các môn toán học, kỹ thuật và khoa học. Matlab được viết tắt từ **MAT**rix **LAB**oratory do mục đích ban đầu của Matlab là xây dựng nên một công cụ hỗ trợ việc tính toán các ma trận một cách dễ dàng nhất.

Một trong nhiều lý do khiến người sử dụng thích dùng Matlab chính là chế độ tương tác (interactive mode). Ở chế độ này, sau khi gõ câu lệnh và thực thi, kết quả sẽ được in ra ngay trong cửa sổ dòng lệnh (command window).

Tuy nhiên, nếu người dùng chỉ cần thực thi câu lệnh nhưng không cần in kết quả ra cửa sổ dòng lệnh, ta thêm dấu chấm phẩy (;) ngay sau câu lệnh.



Hơn nữa, ở bên trái màn hình, người dùng có thể thấy 3 tab Current directory, Workspace và Command history. Trong đó,

- *Current directory*: thể hiện thư mục đang làm việc. Khi muốn hực thi một tập tin .m nào đó, người dùng phải chắc chắn rằng tập tin .m phải được nhìn thấy trong tab này.
- *Workspace*: chứa danh sách các biến đã được khai báo và sử dụng trong chương trình. Ở tab này, người dùng có thể đọc được tên biến, giá trị, kích thước của biến,
- *Command history*: chứa danh sách các câu lệnh đã được thực thi trong cửa sổ dòng lệnh (command window). Người dùng có thể nhấp đôi vào một lệnh bất kì để chương trình thực hiện lại lệnh đó. Hoặc người dùng có thể dùng phím mũi tên lên (↑), xuống (↓) trong cửa sổ dòng lệnh để tìm lại các lệnh mà chương trình đã thực thi.

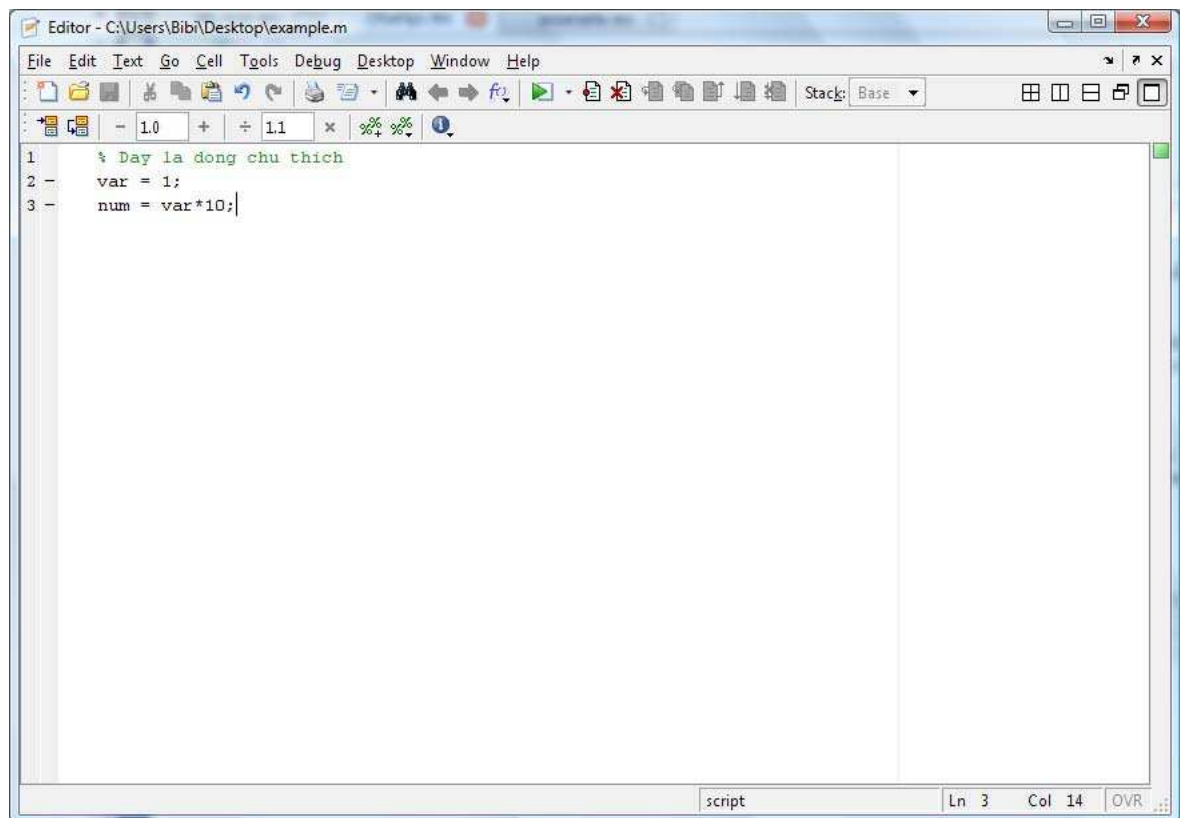
Bên cạnh đó, Matlab vẫn duy trì chế độ kịch bản (script mode) hỗ trợ cho người dùng khi lập trình các hàm hay chương trình từ đơn giản đến phức tạp. Các câu lệnh sẽ được lưu trong một tập tin có đuôi .m (ví dụ: `example.m`) và được thực thi một lần khi chương trình khởi chạy.

Để tạo một tập tin .m, người dùng vào File chọn New → M-File hay nhấn




vào biểu tượng nằm trên thanh MATLAB Toolbar.

Trong Matlab, khi muốn viết một dòng chú thích, ta đặt dấu phần trăm (%) ở đầu dòng. Tiện lợi hơn, ta có thể sử dụng phím tắt **Ctrl+R** để biến các dòng đã chọn trở thành chú thích và **Ctrl+T** để loại bỏ ký hiệu chú thích trước các dòng chú thích.





Để thực thi một tập tin .m, ta nhấn vào biểu tượng  nằm trên thanh Editor Toolbar hoặc sử dụng phím tắt **F5**. Tương tự như các ngôn ngữ lập trình khác, ở chế độ kịch bản, Matlab cũng hỗ trợ công cụ debug giúp người dùng kiểm tra chương trình của mình từng bước nhằm phát hiện lỗi sai trong quá trình viết.

1.2 Phép toán, biến, vector, ma trận

1.2.1 Biến

Trong ngôn ngữ lập trình Matlab, một *biến* (variable) được khai báo và khởi tạo thông qua câu lệnh gán.

```
>> num = 98
num =
    98
>> pi = 3.1415926535897931
pi =
    3.1416
>> msg = 'Good morning'
msg =
    Good morning
```

Tên biến bao gồm các ký tự chữ, số và ký hiệu gạch dưới (_). Tên biến phải bắt đầu bằng ký tự chữ và có độ dài tùy thích. Tuy nhiên, Matlab chỉ ghi nhớ 31 ký tự đầu tiên. Đồng thời, Matlab luôn phân biệt chữ in và chữ thường khi đặt tên biến hoặc tên chương trình.

Các kiểu tên biến hợp lệ: `arg1`, `no_name`, `vars`, `Vars`

Khi tên biến được đặt không hợp lệ, Matlab sẽ xuất hiện thông báo:

```
>> 4rum = 'Forum'
??? 4rum = 'Forum'
|
Error: Unexpected MATLAB expression.
```


Nếu tên biến chưa được khởi tạo mà xuất hiện khi chạy một dòng lệnh nào đó, Matlab sẽ xuất hiện thông báo:

```
??? Undefined function or variable ...
```

Chú ý: Trong ngôn ngữ lập trình Matlab, mỗi biến khi khởi tạo sẽ được xem như một mảng. Nếu biến có giá trị đơn thì mảng có kích thước 1x1. Nếu biến là ma trận hoặc vector thì kích thước của mảng chính là kích thước của ma trận hoặc vector đó. Đây là một điểm khác biệt của Matlab so với các ngôn ngữ lập trình khác. Để lấy kích thước của một biến, ta sử dụng hàm `size()`.

```
>> size(num)
ans =
     1     1
>> size(msg)
ans =
     1    12
```

Ngôn ngữ lập trình Matlab xem chuỗi ký tự như mảng một chiều chứa các ký tự. Do đó, kích thước của biến `msg` là 1 dòng, 12 cột.

1.2.2 Phép toán

Matlab cung cấp các phép toán số học cơ bản như sau

Phép toán	Dạng đại số	Matlab
Cộng	$a + b$	<code>a+b</code>
Trừ	$a - b$	<code>a-b</code>
Nhân	$a * b$	<code>a*b</code>
Chia phải	a/b	<code>a/b</code>
Chia trái	b/a	<code>a\b</code>
Lũy thừa	a^b	<code>a ^ b</code>

```
>> 1+2; 5-3; 2*4; 5 ^ 2*(10-4);  
>> 9/3  
ans =  
    3  
>> 9\3  
ans =  
    0.3333
```

Hơn nữa, Matlab còn hỗ trợ một số hàm số học đơn giản như hàm làm tròn *round()*, làm tròn lên *ceil()*, làm tròn xuống *floor()*, lấy phần dư *mod()*, tìm ước chung lớn nhất *gcd()*, tìm bội chung nhỏ nhất *lcm()*, và hàm lấy căn *sqrt()*.

```
>> round(1.6)  
ans =  
    2  
>> floor(10.8)  
ans =  
   10  
>> mod(10,8)  
ans =  
    2  
>> gcd(45,30)  
ans =  
   15  
>> lcm(45,30)  
ans =  
   90  
>> sqrt(9)  
ans =  
    3
```

Ngoài ra còn có các phép toán so sánh như bằng (*==*), khác (*~=*), lớn hơn (*>*), nhỏ hơn (*<*), lớn hơn hoặc bằng (*>=*), và nhỏ hơn hoặc bằng (*<=*). Giá trị trả về của biểu thức so sánh sẽ bằng 1 nếu biểu thức đúng và bằng 0 nếu biểu thức sai.

```
>> 1 ~= 2
ans =
    1
>> 5 == 10
ans =
    0
```

Cuối cùng là các phép toán luận lý bao gồm *and* (&&), *or* (||) và *not* (!). Một số khác 0 được xem là một giá trị đúng trong các phép toán luận lý của Matlab. Giá trị trả về của các biểu thức luận lý tương tự như biểu thức so sánh.

```
>> n = 15
>> mod(n,2)==0 && mod(n,3)==0
ans =
    0
>> mod(n,2)==0 || mod(n,3)==0
ans =
    1
```

BÀI TẬP

1. Tính bằng tay các biểu thức sau, rồi thử lại bằng Matlab:

- (a) $10 / 2 \setminus 5 - 3 + 2 * 4$
- (b) $3 ^ 2 / 4$
- (c) $3 ^ 2 ^ 2$
- (d) $2 + \text{round}(6 / 9 + 3 * 2) / 2 - 3$
- (e) $2 + \text{floor}(6 / 11) / 2 - 3$
- (f) $2 + \text{ceil}(-6 / 9) - 3$
- (g) $\text{fix}(-4/9) + \text{fix}(3*(5/6))$

2. Cho a=36, b=15. Tính bằng tay các biểu thức sau, rồi thử lại bằng Matlab:

- (a) $\text{mod}(a,b)$

- (b) `rem(a,b)`
- (c) `gcd(a,b)`
- (d) `lcm(a,b)`

Thử lại với các cặp giá trị (a,b) khác.

3. Dự đoán kết quả những câu sau, giải thích và kiểm tra lại bằng Matlab

- (a) `1 & -1`
- (b) `13 & (-6)`
- (c) `0 < -2|0`
- (d) `0 <= 0.2 <= 0.4`
- (e) `5 > 4 > 3`
- (f) `2 > 3 & 1`

1.2.3 Vector

Vector là một dạng đặc biệt của ma trận có một dòng hoặc một cột. Trong các ngôn ngữ lập trình khác, sinh viên đã được làm quen với vector thông qua tên gọi *danh sách* (list) hoặc *mảng một chiều* (1-D array).

Để khởi tạo vector dòng chứa các giá trị rời rạc, các phần tử trong vector phải nằm trong cặp ngoặc vuông (`[]`) và được ngăn cách nhau bởi khoảng trắng hoặc dấu phẩy (`,`).

```
>> arr1 = [1 2 3]
arr1 =
     1     2     3
>> arr2 = [0,-5]
arr2 =
     0    -5
>> arr3 = [arr1 arr2]
arr3 =
     1     2     3     0    -5
```

Để khởi tạo vector dòng chứa các giá trị liên tục (mặc định trong Matlab

là 1) hoặc cách nhau một khoảng giá trị nhất định (còn gọi là bước nhảy), Matlab sử dụng dấu hai chấm (:). Đồng thời, giá trị đầu và cuối của vector không cần thiết đặt trong cặp dấu ngoặc vuông ([]).

```
>> arr1 = 1:5
arr1 =
     1     2     3     4     5
>> arr2 = [1:0.5:2]
arr2 =
     1.0000     1.5000     2.0000
>> arr3 = 10:-1:6
arr3 =
    10     9     8     7     6
```

Hơn nữa, để tạo một vector rỗng - vector không chứa giá trị - trong Matlab, chúng ta khai báo như sau:

```
>> emp_vect = []
emp_vect =
     []
```

Ngược lại, để tạo ra vector cột, chúng ta cần nghịch đảo vector cột bằng cách sử dụng dấu nháy đơn (') hoặc sử dụng dấu chấm phẩy (;) để ngăn cách giữa các phần tử.

```
>> col_arr = [1:3]'
>> col_arr = [1;2;3]
col_arr =
     1
     2
     3
```

Giá trị của một phần tử tại một vị trí bất kỳ trong vector được truy xuất thông qua chỉ số. Trong Matlab, chỉ số luôn bắt đầu từ 1 và có thể là một giá trị đơn hoặc một mảng.

- Trích phần tử thứ i: $X(i)$

- Trích nhiều phần tử: `X([danh sách vị trí])`

```
>> arr = 10:-1:0
arr =
    10     9     8     7     6     5     4     3     2     1     0
>> arr(5)
ans =
     6
>> arr(1:3)
ans =
    10     9     8
>> arr([10:-2:6])
ans =
     1     3     5
>> arr([7,8,11])
ans =
     4     3     0
```

Để xóa một phần tử trong vector, chúng ta sẽ gán phần tử đó với vector rỗng.

```
>> arr([2 5]) = []
arr =
    10     8     7     5     4     3     2     1     0
>> size(arr)
ans =
     1     9
```

BÀI TẬP

- Cho $x = [3 \ 1 \ 5 \ 7 \ 9 \ 2 \ 6]$, dự đoán kết quả các dòng lệnh sau và thử lại bằng Matlab
 - $x(3)$
 - $x(1:7)$
 - $x(1:end)$
 - $x(1:end-1)$

- (e) $x(6:-2:1)$
- (f) $x([1\ 6\ 2\ 1\ 1])$
- (g) $\text{sum}(x)$

2. Cho $x = [1\ 5\ 2\ 8\ 9\ 0\ 1]$ và $y = [5\ 2\ 2\ 6\ 0\ 0\ 2]$, giải thích kết quả các dòng lệnh sau

- (a) $x > y$
- (b) $y < x$
- (c) $x == y$
- (d) $x \leq y$
- (e) $y \geq x$
- (f) $x \mid y$
- (g) $x \& y$
- (h) $x \& (-y)$
- (i) $(x > y) \mid (y < x)$
- (j) $(x > y) \& (y < x)$

3. Cho 2 vectơ $a = [1\ 0\ 2]$ và $b = [0\ 2\ 2]$, xác định giá trị các biểu thức sau. Kiểm tra lại bằng Matlab

- (a) $a = b$
- (b) $a < b$
- (c) $a < b < a$
- (d) $a < b < b$
- (e) $a \mid (a)$
- (f) $b \& (b)$
- (g) $a ((b))$
- (h) $a=b==a$ (Xác định giá trị cuối của a)

4. Cho $x = 1:10$ và $y = [3\ 1\ 5\ 6\ 8\ 2\ 9\ 4\ 7\ 0]$, dự đoán kết quả, giải thích và thử lại bằng Matlab

- (a) $(x > 3) \& (x < 8)$

- (b) $x(x > 5)$
- (c) $y(x \leq 4)$
- (d) $x((x < 2) \mid (x \geq 8))$
- (e) $y((x < 2) \mid (x \geq 8))$
- (f) $x(y < 0)$

1.2.4 Ma trận

Trong Matlab, *ma trận* đại diện cho mảng nhiều chiều có nhiều dòng và nhiều cột. Phương thức khai báo và khởi tạo ma trận tương tự như vector. Tuy nhiên, để kết thúc một dòng trong ma trận, chúng ta sử dụng dấu chấm phẩy (;).

```
>> mat = [1 2 3 ; 4,5,6 ; 7:2:11]
mat =
     1     2     3
     4     5     6
     7     9    11
```

Đồng thời, Matlab cũng hỗ trợ một số hàm cụ thể để khởi tạo các ma trận đặc biệt như sau

- Ma trận không: `zeros(số dòng, số cột)`
- Ma trận vuông không cấp n: `zeros(n)`
- Ma trận đơn vị: `eye(n)`
- Ma trận đường chéo: `diag([các phần tử trên đường chéo chính])`
- Ma trận thực ngẫu nhiên trong khoảng $[0,1]$: `rand(số dòng, số cột)` hoặc `rand(n)` (ma trận vuông cấp n)
- Ma trận toàn số một: `ones(số dòng, số cột)`
- Ma trận vuông một cấp n: `ones(n)`


```
>> zeros(1,2)
ans =
    0    0
>> eye(2)
ans =
    1    0
    0    1
>> diag([3 4 5])
ans =
    3    0    0
    0    4    0
    0    0    5
>> rand(2,1)
ans =
    0.8147
    0.9058
>> ones(2)
ans =
    1    1
    1    1
```

Tương tự như vector, giá trị của một phần tử tại một vị trí bất kỳ trong ma trận được truy xuất thông qua chỉ số dòng và chỉ số cột.

- Trích phần tử tại dòng i cột j : $A(i,j)$
- Trích nhiều phần tử: $A([danh\ sách\ các\ dòng,\ danh\ sách\ các\ cột])$
- Trích đường chéo chính của ma trận: $diag(A)$
- Trích tất cả phần tử của ma trận: $A(:)$
- Trích tất cả phần tử tại cột i : $A(:,i)$
- Trích tất cả phần tử tại dòng j : $A(j,:)$

```
>> mat = rand(2,3)
ans =
    0.1270    0.6324    0.2785
    0.9134    0.0975    0.5469
>> mat(1,3)
ans =
    0.2785
>> mat([2],[1 3])
ans =
    0.9134    0.5469
>> mat(:,2)
ans =
    0.6324
    0.0975
```

Chú ý: Trong Matlab, chỉ số cuối cùng của dòng hay cột của ma trận hoặc vector có thể được thay thế bởi chữ *end*.

BÀI TẬP

Cho $x = [1 \ 4 \ 8]$, $y = [2 \ 1 \ 5]$, và $A = [2 \ 7 \ 9 \ 7 ; 3 \ 1 \ 5 \ 6 ; 8 \ 1 \ 2 \ 5]$. Xét xem dòng lệnh nào hợp lệ, dự đoán kết quả, giải thích và thử lại bằng Matlab

1. $[x; y']$
2. $[x; y]$
3. $A(:, [1 \ 4])$
4. $A([2 \ 3], [3 \ 1])$
5. $A(:)$
6. $[A; A(end, :)]$
7. $A(1:3, :)$
8. $[A; A(1 : 2, :)]$

1.3 Biểu thức Logic

1.3.1 Các toán tử logic

Một biểu thức logic trong Matlab được xây dựng từ 6 toán tử quan hệ là: $>$, $<$, $>=$, $<=$, $==$ (so sánh bằng), $\sim=$ (không bằng) và 3 toán tử logic: \sim (NOT), $\&$ (AND) và $|$ (OR).

Kết quả của một biểu thức logic trả về 1 nếu biểu thức đúng và 0 nếu biểu thức sai. Nếu phần tử so sánh là vec-tơ thì kết quả sẽ là một vec-tơ chứa các giá trị 0 hoặc 1.

Ví dụ 1.3.1. Thực hiện các câu lệnh sau và nhận xét kết quả

1. `r=1;`
`r<=0.5`
2. `r=1:5;`
`r<=3`
3. `a=1:5;`
`b=[0 2 4 5 6];`
`a==b`

Độ ưu tiên của các toán tử

Độ ưu tiên	Toán tử
1.	()
2.	^ .^ , .'
3.	+ - (dấu) ~
4.	* / \ .* ./ .\
5.	+ -
6.	:
7.	> < >= <= == ~=
8.	& (AND)
9.	(OR)

Ví dụ 1.3.2. Thực hiện các câu lệnh sau và nhận xét kết quả

1. `a = 1; b=2; c=3;`
`(b*b==4*a*c) & (a~=0)`
2. `a=1:10`
`(a>=20) & (a<=20)`
3. `a=1;b=2;`
`(a~=0) | (b~=0) | ((a==0) & (b==0))`

Để một biểu thức logic trong sáng và tránh nhầm lẫn, ta nên sử dụng các dấu (,). Thực hiện các câu lệnh sau và so sánh

```
r = 0.5;
0 < r < 1
(0 < r) & (r < 1)
```

1.3.2 Vec-tơ và biểu thức logic

Biểu thức logic cho phép truy xuất một cách linh hoạt đến các thành phần của một vec-tơ hay ma trận.

Ví dụ 1.3.3.

1. `x = [-1 0 2 3 5 6 7 4 9]`

2. $x(x>0)$: xuất những giá trị dương của vec-tơ x .
3. $x(x>2 \ \& \ x \leq 5)$: xuất những giá trị thỏa $2 < x \leq 5$.
4. Lệnh $x>2$ cho kết quả là $0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1$ là vec-tơ chứa kết quả so sánh từng phần tử tương ứng của x với 2 và $x(x>2)$ sẽ xuất ra các giá trị lớn hơn 2.

1.3.3 Các hàm logic: All, Any và Find

Một số hàm logic thông dụng là: **any**, **all**, **find**

any: Kiểm tra xem có tồn tại một phần tử nào của vec-tơ thỏa điều kiện không? Nếu có thì cho kết quả là 1, ngược lại là 0. Ví dụ, **any**($x>0$): kiểm tra xem có tồn tại phần tử nào của vec-tơ x dương hay không.

all: kiểm tra tất cả các phần tử của vec-tơ có thỏa điều kiện không? Kết quả là 1 nếu đúng. Ví dụ, **all**($x < 5$): tất cả các phần tử của x có < 5 hay không?

Lưu ý: kết quả của **all** và **any** là vô hướng (1 hoặc 0).

find: trả về các chỉ số của một vec-tơ thỏa điều kiện logic nào đó. Ví dụ,

$A = [1 \ 2 \ 4; 4 \ 5 \ 6]$

find(**isprime**(A)): xuất ra các vị trí của ma trận A có giá trị là số nguyên tố.

1.4 Lệnh điều kiện và vòng lặp

1.4.1 Lệnh IF

Cú pháp:

if *Biểu thức điều kiện*

Lệnh thực thi

end

```

hoặc
if Biểu thức điều kiện 1
    Lệnh thực thi nếu Đk 1 đúng
elseif Biểu thức điều kiện 2
    Lệnh thực thi nếu Đk 2 đúng
    :
end

```

Ví dụ 1.4.1. Hoán vị x và y nếu $x > y$

```

if x > y
    temp = y;
    y = x ;
    x = temp;
end

```

Ví dụ 1.4.2. Biện luận số nghiệm của phương trình $ax^2 + bx + c = 0$, ($a \neq 0$)

```

delt = b2 - 4*a*c;
if delt < 0
    disp('PT Vo nghiem');
elseif delt == 0
    disp('PT co nghiem kep');
else
    disp('PT co 2 nghiem phan biet');
end

```

Lưu ý: Nếu câu lệnh f được đặt trên 1 dòng thì phải dùng dấu `","` sau `if`. Ví dụ, `if x > 0, x = sqrt(x); end`.

1.4.2 Lệnh FOR

Cú pháp:

```

for var = start : step : end
    Lệnh thực thi
end

```

Vòng lặp với biến chạy là `var` bắt đầu từ `start` và kết thúc là `end`, mỗi bước lặp sẽ tăng một bước nhảy là `step`. `step` có thể là dương (vòng lặp tăng dần) hoặc âm (giảm dần). Nếu không chỉ ra `step`, bước nhảy mặc định là 1.

Ví dụ 1.4.3. Xuất các giá trị từ 1 đến 5 ra màn hình

```
for i=1:5, disp(i), end
```

Xuất các giá trị chẵn từ 1 đến 10 theo thứ tự ngược

```
for i=10:-2:1, disp(i), end
```

Ví dụ 1.4.4. Tính giai thừa $n! = 1 \times 2 \times \dots \times (n - 1) \times n$

```
n = 10;
```

```
fact = 1;
```

```
for k=1:n
```

```
    fact = k*fact;
```

```
end
```

Ta có thể thay thế dùng một vec-tơ chạy một vòng for:

```
for var = v, ..., end.
```

Ví dụ 1.4.5. Tính tích tất cả các phần tử trong vec-tơ $a = [1 \ 2 \ 3 \ -2 \ 4]$

```
prod = 1;
```

```
for i = v, prod = prod*i, end
```

1.4.3 Lệnh WHILE

Cú pháp:

```
while Biểu thức điều kiện
```

```
    Các lệnh thực thi
```

```
end
```

Các lệnh sẽ được thực hiện khi nào biểu thức điều kiện còn đúng.

Ví dụ 1.4.6. Tìm giá trị lớn nhất của n để tổng $1^2 + 2^2 + \dots + n^2$ bé hơn 100.

```
S = 1; n=1;
```

```
while (S + (n+1) ^ 2) < 100
```

```
    n = n + 1;
```

```
    S = S + n ^ 2
```

```
end
```

```
[n, S]
```

Ví dụ 1.4.7. $x = 1$;

```
while 1
    xmin = x;
    x = x/2;
    if x == 0, break, end
end
xmin
```

Lệnh `while 1, ...` sẽ tạo ra vòng lặp vô hạn, để ngắt, ta dùng lệnh `break`.

Sử dụng Break và Continue:

Break: Dùng trong vòng lặp `for` hoặc `while`, dùng để thoát khỏi vòng lặp tại vị trí nó xuất hiện. Nếu có nhiều vòng lặp lồng nhau thì lệnh `break` có tác dụng nhảy sang vòng lặp kế tiếp.

Continue: Dùng trong vòng lặp `for` hoặc `while`, lệnh `continue` cho phép bỏ qua các câu lệnh đứng sau nó và chuyển sang bước lặp kế tiếp trong cùng một vòng lặp.

1.4.4 Lệnh SWITCH ... CASE

Cú pháp:

```
switch Biểu thức điều kiện
case Giá trị thử 1
    Khối lệnh 1
case Giá trị thử 2
    Khối lệnh 2
:
otherwise
    Khối lệnh n
end
```

Lệnh `switch` sẽ lần lượt thực hiện các khối lệnh tương ứng với từng giá trị thử trong biểu thức điều kiện. Biểu thức điều kiện phải có dạng số hoặc chuỗi.

Ví dụ 1.4.8. Tính chuẩn p của vec-tơ x

```
switch p
    case 1
        y = sum(abs(x));
    case 2
        y = sqrt(x'*x);
    case inf
        y = max(abs(x));
otherwise
    error('p phải là 1,2 hoặc inf');
end
```

1.4.5 Script và Hàm

Script: Là các dòng lệnh Matlab được chứa trong một file có phần mở rộng .m; file script có thể được soạn thảo bằng Matlab Editor hoặc các chương trình soạn thảo khác. Để thực thi script chỉ cần gọi tên file trong cửa sổ dòng lệnh của Matlab.

Hàm: Cũng là các đoạn lệnh Matlab được soạn thảo trong file .m, hàm nhận các tham số truyền vào, xử lý và xuất ra giá trị. Tên của hàm phải giống như tên của file .m, tránh đặt tên hàm trùng với các hàm có sẵn của Matlab. Dòng đầu tiên của hàm (trừ phần chú thích) phải có dạng như sau

```
function [Các giá trị xuất] = Tên_hàm(Các giá trị nhập)
```

Sau dòng này, các dòng chú thích bắt đầu bằng dấu % sẽ xuất hiện khi gọi lệnh `help tên_hàm`.

Ví dụ 1.4.9. Hàm tính giá trị thứ n của dãy Fibonnaci

```
function f = Fib(n)
F = zeros(1,n+1);
F(2) = 1;
for i = 3:n+1
    F(i) = F(i-1) + F(i-2);
end
f = F(n);
```

Lưu tên file *Fib.m*.

Ví dụ 1.4.10. Hàm trả về nhiều giá trị

```
function [a b] = maxmin(x)
```

```
% Ham tra ve gia tri lon nhat va nho nhat cua mang x
```

```
a = max(x);
```

```
b = min(x);
```

Lưu tên file *maxmin.m*. Hai giá trị trả về chứa trong mảng [a b].

Ví dụ 1.4.11. Hàm không có giá trị trả về.

```
function circle(a,b,r)
```

```
% Ham ve duong tron tam (a,b), ban kinh r
```

```
t = 0:pi/20:2*pi;
```

```
x = a + r*sin(t);
```

```
y = b + r*cos(t);
```

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

```
axis equal;
```

1.5 Vẽ đồ thị

1.5.1 Vẽ đồ thị trong 2-D

Lệnh cơ bản:

```
plot(x,f(x))
```

Với x : vec-tơ chứa miền giá trị của hàm f . $f(x)$: các giá trị của f ứng với x .

Ví dụ 1.5.1. Vẽ đồ thị $y = \sin(x)$ từ $[0, 2\pi]$

```
x = 0 : pi/100 : 2*pi;
```

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

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

Chú thích trên đồ thị:

`text(x, y, '...')`: đặt một chú thích (trong dấu ' ') lên đồ thị tại tọa độ (x, y) .

`gtext('...')`: đặt chú thích lên đồ thị, vị trí được xác định bởi click chuột.

`title('...')`: tựa đề của đồ thị.

`xlabel('...')`: ghi nhãn cho trục Ox.


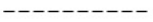


`ylabel('...')`: ghi nhãn cho trục Oy.

`hold on/off`: bật / tắt chế độ cho phép vẽ nhiều đồ thị trong cùng một hệ trục tọa độ.













Các tùy chỉnh về nét vẽ, dấu và màu sắc:

Lệnh: `plot(x,y,'Nét vẽ_Dấu_Màu sắc')`

Nét vẽ:

Nét vẽ	
<code>_ /</code>	
<code>_ _ \</code>	
<code>\: /</code>	
<code>_ . /</code>	

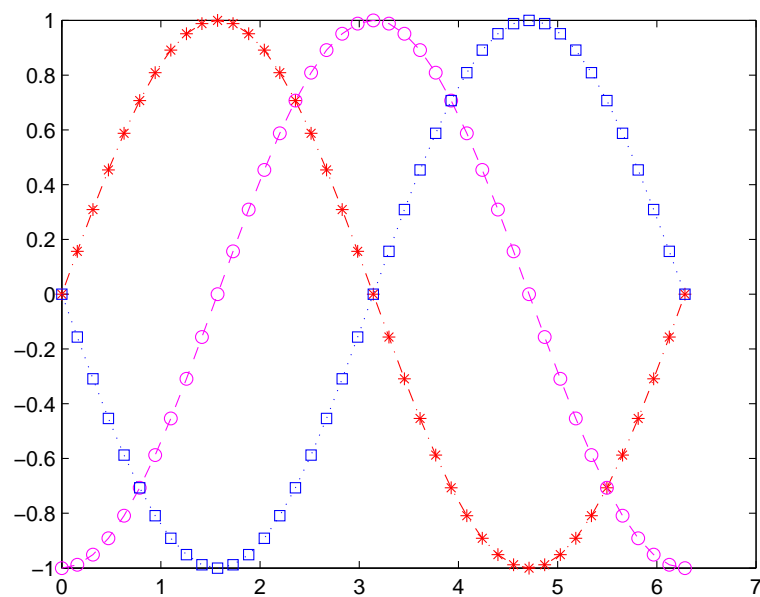
Dấu (marker):

Dấu			
<code>\+ /</code>		<code>\h' / 'hexagram'</code>	
<code>\o /</code>		<code>\^</code>	
<code>* /</code>		<code>\v</code>	
<code>_ . /</code>		<code>\></code>	
<code>\s' / 'square'</code>		<code>\<</code>	
<code>\d' / 'diamond'</code>			
<code>\p' / 'pentagram'</code>			

Màu sắc: gồm 8 tùy chọn là 'r' - đỏ, 'k' - đen, 'w' - trắng, 'y' - vàng, 'c' - cyan, 'b' - xanh nước biển, 'g' - xanh lá cây, 'm' - tím.

Ví dụ 1.5.2.

```
x = 0:pi/20:2*pi;
plot(x, sin(x), '-.*r');
hold on
plot(x, sin(x - pi/2), '-om');
plot(x, sin(x - pi), ':bs');
hold off
```



Tùy chỉnh màu sắc và độ lớn của nét vẽ:

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

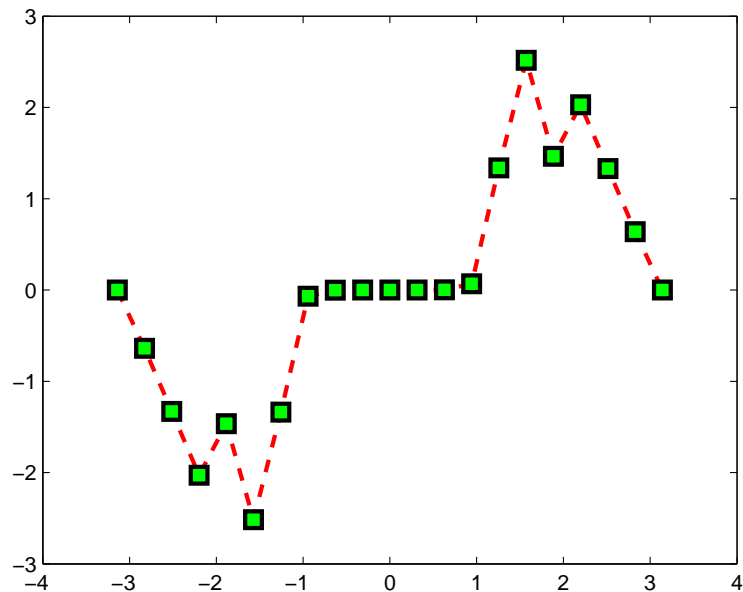
MarkerEdgecolor: màu của đường viền dấu (marker).

MarkerFacecolor: màu bên trong dấu.

Markersize: độ lớn của dấu, tính bằng pt.

Ví dụ 1.5.3.

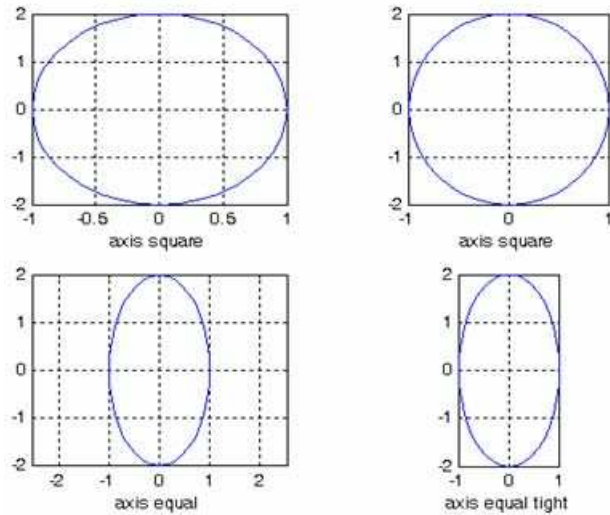
```
x = -pi:pi/10:pi;  
y = tan(sin(x)) - sin(tan(x));  
plot(x,y,'-rs','LineWidth',2,'MarkerEdgecolor','k',...  
'MarkerFacecolor','g', 'Markersize',10)
```

**Xác định tọa độ:**

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

Tùy chỉnh các kiểu trục tọa độ:

```
axis on/off/auto  
axis normal/square/equal/tight  
axis ij/xy  
grid on/off
```



Các kiểu tùy chỉnh hệ trục tọa độ

subplot - Vẽ nhiều đồ thị trong cùng một cửa sổ:

`subplot(m, n, p)`: tạo ra một 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.

Ví dụ 1.5.4. Vẽ 4 đồ thị trong cùng 1 cửa sổ

```
t = 0:pi/20:2*pi; [x,y] = meshgrid(t);
```

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

```
plot(sin(t),cos(t))
```

```
axis equal
```

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

```
z = sin(x)+cos(y);
```

```
plot(t,z)
```

```
axis([0 2*pi -2 2])
```

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

```
z = sin(x).*cos(y);
```

```
plot(t,z)
```

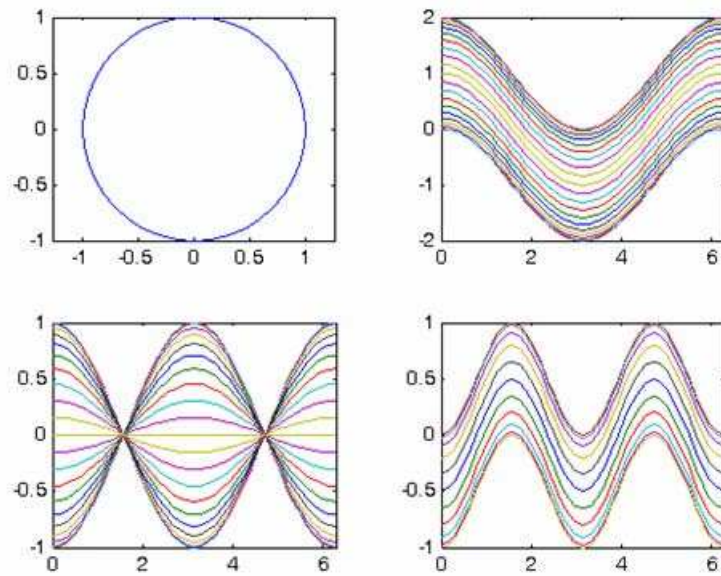
```
axis([0 2*pi -1 1])
```

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

```
z = (sin(x). 2)-(cos(y). 2);
```

```
plot(t,z)
```

```
axis([0 2*pi -1 1])
```

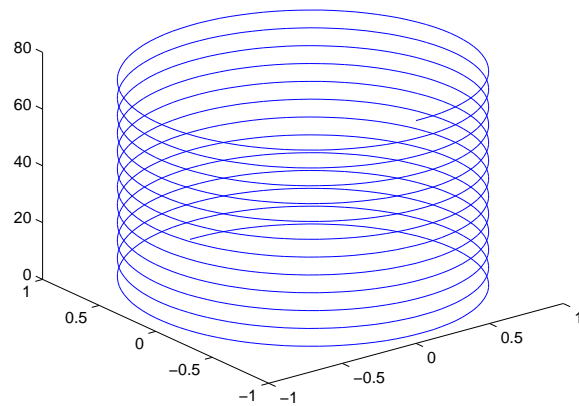


1.5.2 Vẽ đồ thị trong 3-D

Lệnh cơ bản: `plot3(x, y, z)` Trong `plot3`, ta cần xác định các vec-tơ (x, y, z) . Để vẽ mặt $(x, y, z = f(x, y))$, sử dụng lệnh `meshgrid(x, y)`.

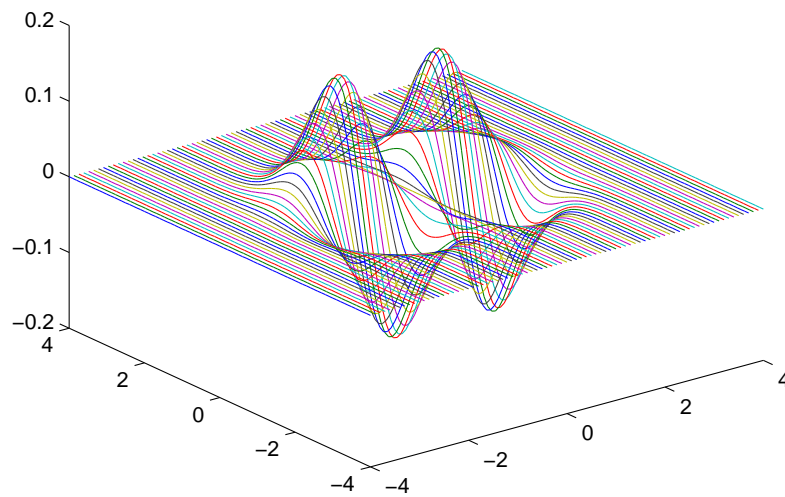
Ví dụ 1.5.5.

```
t = 0:0.02*pi:25*pi;
x = sin(t); y = cos(t);
z = t;
plot3(x,y,z);
```



Ví dụ 1.5.6. Vẽ mặt $z(x, y) = x^2 y e^{-x^2 - y^2}$ với $-4 \leq x \leq 4$ và $-4 \leq y \leq 4$.

```
[x,y]=meshgrid([-4:0.1:4]);  
z=x.*x.*y.*exp(-x.^2-y.^2);  
plot3(x,y,z)
```



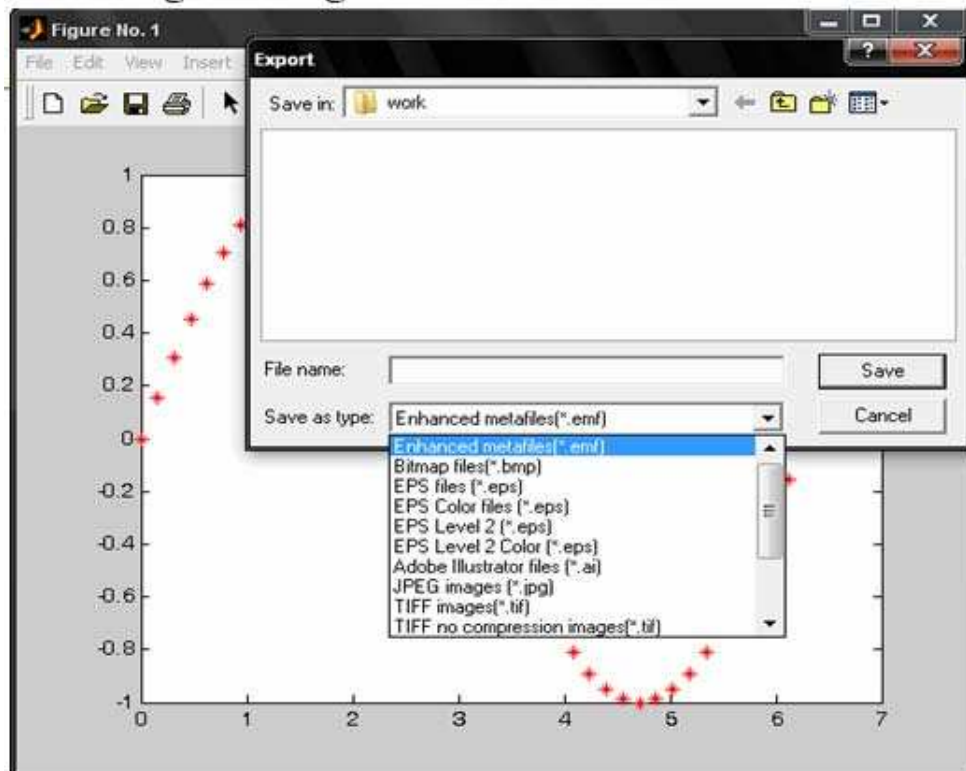
Một số lệnh vẽ đồ thị trong 3-D khác:

- `contour / contourf / contour3`
- `mesh / meshc / meshz`
- `surf / surfc`
- `waterfall`
- `bar3 / bar3h`
- `pie3 / fill3`
- `comet3 / scatter3 / stem3`

(Xem hướng dẫn chi tiết trong `help`)

In và xuất đồ thị:

- Dùng lệnh
`print -dtiff -r200 mygraph.tiff print -deps2 mygraph.eps`
- Sử dụng Plotting Tools



Xuất đồ thị ra file sử dụng Plotting Tools

CHƯƠNG 2

Đại số tuyến tính

2.1 Các phép toán ma trận, các phép biến đổi sơ cấp

2.1.1 Các phép toán ma trận

Trong Matlab, các phép toán ma trận như cộng, trừ, nhân, và lũy thừa được viết tương tự như các phép toán cơ bản đã trình bày ở chương 1.

- Cộng: $A+B$
- Trừ: $A-B$
- Nhân: $A*B$
- Chia phải: A/B tương đương với $A*inv(B)$
- Chia trái: $A\backslash B$ tương đương với $inv(A)*B$
- Lũy thừa: A^n
- Nhân với một số: $A*n$

```
>> A = [1,2;3,4]
>> B = [5,6;7,8]
>> A+B
ans =
     6     8
    10    12
>> A*B
ans =
    19    22
    43    50
>> A/B
ans =
     3    -2
     2    -1
>> A\B
ans =
    -3    -4
     4     5
```

Bên cạnh đó, Matlab còn hỗ trợ một số hàm sơ cấp liên quan đến ma trận mà người dùng không phải viết lại chương trình để tính toán như các ngôn ngữ lập trình khác.

- Chuyển vị: `A'`
- Định thức: `det(A)`
- Ma trận nghịch đảo: `A ^ (-1)` hoặc `inv(A)`
- Hạng của ma trận: `rank(A)`
- Vết của ma trận: `trace(A)`
- Tìm dạng bậc thang rút gọn theo phương pháp Gauss Jordan: `rref(A)`
- Tính tổng các phần tử của ma trận theo cột: `sum(A)`

```
>> A = [2 4 1 ; 6 7 2 ; 3 5 9]
>> det(A)
ans =
    -77
>> rank(A)
ans =
     3
>> trace(A)
ans =
    18
>> sum(A)
ans =
    11    16    12
```

Đặc biệt hơn, để thực hiện các phép toán nhân, chia giữa phần tử với phần tử của hai ma trận, hay lũy thừa từng phần tử của ma trận, chúng ta thêm dấu chấm (.) vào trước phép toán cần tính.

```
>> A = [1,2;3,4]
>> B = [5,6;7,8]
>> A.*B
ans =
     5     12
    21     32
>> A./B
ans =
    0.2000    0.3333
    0.4286    0.5000
>> A.\B
ans =
    5.0000    3.0000
    2.3333    2.0000
>> A.^3
ans =
     1     8
    27    64
```

BÀI TẬP

1. Cho $x = [2 \ 5 \ 1 \ 6]$.
 - (a) Cộng thêm 16 vào tất cả các phần tử.
 - (b) Cộng thêm 3 vào các phần tử ở vị trí lẻ.
 - (c) Lấy căn bậc 2 tất cả các phần tử.
 - (d) Bình phương tất cả các phần tử.
2. Cho x, y lần lượt là các vector cột. $x = [3 \ 2 \ 6 \ 8]'$, $y = [4 \ 1 \ 3 \ 5]'$.
 - (a) Lấy tổng các phần tử của x cộng thêm vào từng phần tử của y .
 - (b) Luỹ thừa mỗi phần tử của x với số mũ tương ứng là các phần tử của y .
 - (c) Chia các phần tử của y với các phần tử tương ứng của x .
 - (d) Nhân các phần tử của x với các phần tử tương ứng của y , đặt trong vector z .
 - (e) Tính tổng các phần tử của z , gán cho w .
 - (f) Tính $x.*y - w$.
 - (g) Tích vô hướng của x và y
3. Cho $x = [1 \ 4 \ 8]$, $y = [2 \ 1 \ 5]$ và $A = [3 \ 1 \ 6 ; 5 \ 2 \ 7]$. Xét xem dòng lệnh nào hợp lệ, dự đoán kết quả, giải thích; rồi thử lại bằng Matlab :
 - (a) $x + y$
 - (b) $x + A$
 - (c) $x' + y$
 - (d) $A - [x' \ y']$
 - (e) $A - 3$
4. Cho $A = [2 \ 7 \ 9 \ 7 ; 3 \ 1 \ 5 \ 6 ; 8 \ 1 \ 2 \ 5]$, dự đoán kết quả, giải thích; rồi thử lại bằng Matlab:
 - (a) A'
 - (b) $\text{sum}(A)$

- (c) $\text{sum}(A')$
- (d) $\text{sum}(A, 2)$
- (e) $[[A; \text{sum}(A)] [\text{sum}(A, 2); \text{sum}(A(:))]]$

5. Hãy tạo ra ma trận 4x4 có giá trị nguyên nằm trong khoảng $[-10, 10]$, Sau đó:

- (a) Cộng mỗi phần tử của ma trận cho 15
- (b) Bình phương mỗi phần tử của ma trận
- (c) Cộng thêm 10 vào các phần tử ở dòng 1 và dòng 2
- (d) Cộng thêm 10 vào các phần tử ở cột 1 và cột 4
- (e) Tính nghịch đảo mọi phần tử
- (f) Lấy căn bậc hai mọi phần tử

2.1.2 Các phép biến đổi sơ cấp

Sau đây, chúng ta sẽ làm quen với các phép biến đổi sơ cấp trên dòng và cột của ma trận. Đối với phép biến đổi sơ cấp trên dòng, để

- Biến dòng i thành α lần dòng i : $A(i, :) = A(i, :) * \alpha$
- Biến dòng i thành dòng i cộng α lần dòng j : $A(i, :) = A(i, :) + A(j, :) * \alpha$
- Hoán vị dòng i và dòng j : $A = A([\text{thứ tự dòng}], :)$

```

>> A = [2 4; 3 8; 6 7]
>> A(1,:) = A(1,:)*10
A =
    20    40
     3     8
     6     7
>> A(2,:) = A(2,:) + 3*A(3,:)
A =
    20    40
    21    29
     6     7
>> A = A([3 1 2],:)
A =
     6     7
    20    40
    21    29

```

Tương tự, ta có các phép biến đổi sơ cấp trên cột là

- Biến cột i thành α lần cột i : $A(:,i) = A(:,i)*\alpha$
- Biến cột i thành cột i cộng α lần cột j : $A(:,i) = A(:,i) + A(:,j)*\alpha$
- Hoán vị cột i và cột j : $A = A(:, [\text{thứ tự cột}])$

BÀI TẬP

Cho ma trận $A = [2 \ 4 \ 1 ; 6 \ 7 \ 2 ; 3 \ 5 \ 9]$, viết lệnh Matlab để

1. Gán cho vector x là dòng thứ nhất của A .
2. Gán cho ma trận y là hai dòng còn lại (cuối) của A .
3. Tính tổng theo dòng ma trận A .
4. Tính tổng theo cột ma trận A .
5. Tìm phần tử lớn nhất và phần tử nhỏ nhất của ma trận.
6. Tính tổng các phần tử của A

2.2 Ma trận nghịch đảo, Phương trình ma trận và Hệ phương trình tuyến tính

2.2.1 Ma trận nghịch đảo

Định nghĩa 2.2.1. Ma trận đơn vị cấp n là ma trận vuông cấp n có các phần tử trên đường chéo chính là 1, các phần tử còn lại là 0. Ví dụ:

$$\begin{bmatrix} 1 & 0 & . & . & . & 0 \\ 0 & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & 0 \\ 0 & . & . & . & 0 & 1 \end{bmatrix}$$

Trong Matlab, chúng ta sử dụng hàm **eye(n)** để tạo ma trận đơn vị cấp n .

Ví dụ 2.2.2. ta tạo ma trận đơn vị 3×3 bằng code matlab như sau.

```
>> eye(3)
```

```
ans =
```

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

Định nghĩa 2.2.3. Cho A là ma trận vuông cấp n . Ma trận A được gọi là ma trận khả nghịch nếu như tồn tại ma trận B vuông cấp n sao cho $A.B = B.A = I_n$. Ta sử dụng hàm **inv(A)** hoặc sử dụng phép toán A^{-1} để tính toán ma trận nghịch đảo.

Ví dụ 2.2.4. Cho ma trận A như sau:

$$\begin{bmatrix} -3 & 4 & 6 \\ 0 & 1 & 1 \\ 2 & -3 & 4 \end{bmatrix}$$

```
>> inv(A)
```

```
ans =
```

```
-4.0000    -3.0000     2.0000
-1.0000    -1.0000         0
 6.0000     4.0000    -3.0000
```

Ví dụ 2.2.5. Cho ma trận A như sau:

$$\begin{bmatrix} 8+3i & 9-i & 3+2i \\ 9-3i & 6 & 5-2i \\ 1+2i & 1-2i & 10+i \end{bmatrix}$$

Ma trận nghịch đảo của A là

```
>> inv(A)
```

```
ans =
```

```
-0.0126 - 0.0995i    0.0515 + 0.1266i    -0.0686 - 0.0138i
 0.0990 + 0.1053i    -0.0075 - 0.1357i     0.0235 + 0.0126i
-0.0469 + 0.0264i    0.0465 - 0.0155i     0.1001 + 0.0085i
```

Ví dụ 2.2.6. Cho ma trận B như sau, B không khả nghịch khi đó hàm inv sẽ trả ra các giá trị như sau:

$$\begin{bmatrix} 3 & 4 & 6 \\ -5 & 2 & 16 \\ 2 & 1 & -1 \end{bmatrix}$$

```
inv(B)
```

```
Warning: Matrix is singular to working precision.
```

```
ans =
```

```
Inf     Inf     Inf
Inf     Inf     Inf
Inf     Inf     Inf
```

2.3 Bài tập

- 2.1 Tự tạo ngẫu nhiên ma trận A là ma trận vuông 25 phần tử thuộc \mathbb{C} .
- Kiểm tra ma trận A có khả nghịch hay không, nếu không thì cho lại ma trận ngẫu nhiên khác.
 - Xác định ma trận nghịch đảo của A bằng các phép biến đổi sơ cấp trên dòng (làm từng bước).

- 2.2 Tự tạo ma trận ngẫu nhiên A là ma trận vuông 100 phần tử thuộc \mathbb{C} .
- Kiểm tra ma trận A có khả nghịch hay không, nếu không thì cho lại ma trận ngẫu nhiên khác.
 - Xác định ma trận nghịch đảo của A bằng các phép biến đổi sơ cấp trên dòng (không cần làm từng bước).
 - Dùng hàm inv hoặc phép toán mũ -1 để xác định ma trận nghịch đảo của A .

- 2.3 Cho ma trận bất kì không khả nghịch 100 phần tử. Dùng các phép biến đổi sơ cấp trên dòng để kiểm tra tính không khả nghịch của ma trận đó.

- 2.4 Cho các cặp ma trận A, B theo thứ tự. Xác định $A^{-1}B^{-1}, (AB)^{-1}, B^{-1}A^{-1}$

$$(a) \quad \begin{bmatrix} 1 & 2 \\ 4 & -1 \end{bmatrix}, \begin{bmatrix} 2 & -3 \\ -4 & 1 \end{bmatrix}$$

$$(b) \quad \begin{bmatrix} 2 & 3-i & 1 \\ i-3 & 1 & 0 \\ 1 & 2 & i-1 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 2i+1 \\ 0 & 1 & 2 \\ 3 & 1 & i+1 \end{bmatrix}$$

2.3.1 Ma trận giả nghịch đảo

Cho ma trận A kích thước $m \times n$. Ma trận B là ma trận giả đảo của A là ma trận thỏa 4 tính chất sau:

- $ABA = A$
- $BAB = B$
- $(AB)^* = AB$

$$4.(BA)^* = BA$$

Trong đó ma trận AB và BA là ma trận Hermit, tức là ma trận phức có tính đối xứng liên hợp.

Ghi chú 2.3.1. Trường hợp đặt biệt:

1. Nếu A^*A khả nghịch thì ma trận $B = (A^*A)^{-1}A^*$ là ma trận giả đảo trái của A .
2. Nếu AA^* khả nghịch thì ma trận $B = A^*(AA^*)^{-1}$ là ma trận giả đảo phải của A .

Ví dụ 2.3.2. Ma trận Hermit Cho ma trận A như sau

$$\begin{bmatrix} 7 & 5+i \\ 5-i & 9 \end{bmatrix}$$

Ma trận chuyển vị của A là

```
>> A'
```

```
ans =
```

```
7.0000          5.0000 + 1.0000i
5.0000 - 1.0000i    9.0000
```

Ta nhận thấy rằng ma trận chuyển vị của A cũng bằng chính A do vậy ma trận A chính là ma trận Hermit. Tiếp theo, ta sẽ dùng Matlab để tính toán ma trận giả đảo của A . Ta sử dụng hàm **pinv(A)**

Ví dụ 2.3.3. Cho ma trận A như sau

$$\begin{bmatrix} 2 & 4 & 3 \\ 5 & 1 & 7 \end{bmatrix}$$

Sau đó dùng hàm **pinv** để tính ma trận giả đảo của A và kiểm tra dựa vào định nghĩa.

```
pinv(A)
```

```
ans =
```

```
-0.0263    0.0789
 0.2789   -0.1168
-0.0211    0.1032
```

```
>> A*pinv(A)*A
```

```
ans =
```

```
2.0000    4.0000    3.0000
5.0000    1.0000    7.0000
```

2.4 Bài tập

2.1 Cho ma trận A như sau:

$$\begin{bmatrix} 81 + 7i & 10 + 8i & 16 + 7i & 14 + 8i \\ 91 & 28 + 7i & 97 & 42 + 7i \\ 13 + 8i & 55 + 4i & 96 + 3i & 92 + 3i \\ 91 + 9i & 96 + 7i & 49 & 79 + 10i \\ 63 + 7i & 96 + 2i & 80 + i & 96 \end{bmatrix}$$

Xác định ma trận giả đảo trái hoặc giả đảo phải (nếu có) của A theo 2 cách:

- _ Sử dụng hàm `pinv`. Cho biết ma trận giả đảo trên là ma trận giả đảo trái hay phải (nếu có).
- _ Không sử dụng hàm `pinv`.

2.2 (*) Cho ma trận $A \in \mathfrak{M}_{4 \times 7}(\mathbb{R})$ với các phần tử bất kì.

- (a) Kiểm tra tính nghịch đảo của $B = AA^T$ và $C = A^T A$.
- (b) Nếu B không khả nghịch tìm ma trận giả đảo của B (nếu có). (Nếu B khả nghịch ta làm tương tự với C).
- (c) (**) Nếu sử dụng hàm `pinv` thì cho biết ma trận đó là giả đảo trái hay giả đảo phải.
- (d) Thực hành lại cái câu (a), (b), (c) với tính toán symbolic.

2.4.1 Giải phương trình ma trận

Định nghĩa 2.4.1. Cho phương trình $AX=B$. Trong đó A là ma trận $m \times n$, B là ma trận $m \times k$ và X là ẩn. Ta cần tìm X .

Ghi chú 2.4.2. Nếu A là ma trận vuông tức $m=n$ và khả nghịch. Ta dùng nghịch đảo của A .

Trong Matlab ta sử dụng phép toán \backslash gọi là phép chia trái hoặc hàm `mrdivide(A,b)`.

Ví dụ 2.4.3. Cho phương trình ma trận như trên.

```
A=[2 4 3;5 1 7;2 3 5];b=[2 5;4 1;3 7];
>> X=A\b
```

X =

```
0.0270    -2.2703
0.0811     1.1892
0.5405     1.5946
```

Đối với phương trình $XA=b$, ta dùng phép toán chia phải / hoặc hàm `mrdivide(b,A)`.

Để giải phương trình ma trận trong Matlab, ta chỉ sử dụng hàm `linsolve(A,b,opts)`. Trong đó `opts` là tham số chỉ tính chất của ma trận A . Điều này rất quan trọng vì nó ảnh hưởng đến tốc độ tính toán của hàm.

Ví dụ 2.4.4. Nếu A là ma trận đối xứng, ta nhập vào câu lệnh như sau `linsolve(A,b,SYM)`. Sau đây là bảng các giá trị của tham số `opts`.

2.5 Bài tập

2.1 Cho ma trận A như sau `rand(50,50)`. Tìm ma trận X thỏa :

(a) $12X - 2.5A = I_{50}$

(b) $XA^{50} = B$. Với B là ma trận ngẫu nhiên khác A .

Bảng 2.4.1: Giá trị của tham số opts

Giá trị tham số	Thuộc tính của ma trận A
LT	Ma trận tam giác dưới
UT	Ma trận tam giác trên
UHESS	Ma trận Hessenberg trên
SYM	Ma trận đối xứng
POSDEF	Ma trận xác định dương
RECT	Ma trận chữ nhật
TRANS	Ma trận đối xứng liên hợp

2.2 Cho ma trận $A \in \mathfrak{M}_{2 \times 3}(\mathbb{C})$, ma trận $B \in \mathfrak{M}_{2 \times 2}(\mathbb{C})$. Tìm ma trận X thỏa $AX=B$.

2.3 Cho ma trận $A, B \in \mathfrak{M}_{100 \times 70}(\mathbb{C})$. Tìm ma trận X thỏa:

(a) $100A + 10X = B$

(b) $XA = B$

2.4 Giải phương trình ma trận sau:

(a)

$$X \begin{bmatrix} 13 & -8 & -12 \\ 12 & -7 & -12 \\ 6 & -4 & -5 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 1-i & 2+i & 108-i \\ 12-7i & 9 & 22-5i \\ 3i & 5+10i & -25 \end{bmatrix} X \begin{bmatrix} 1 & -1 & -1 \\ 1 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

2.5.1 Hệ phương trình tuyến tính

2.5.2 Đưa về dạng ma trận

Để giải hệ phương trình tuyến tính, ta đưa về dạng ma trận với b là ma trận $m \times 1$. Ngoài các phương pháp trên, ta có thể sử dụng hàm `rref([A b])` để đưa ma trận về dạng bậc thang rút gọn.

Ví dụ 2.5.1. Giải hệ phương trình tuyến tính sau:

$$\begin{cases} x_1 + 7x_2 - 2x_3 = 21 \\ 2x_1 + 3x_2 + 7x_3 = 2 \\ x_1 + 8x_2 + 2x_3 = -1 \end{cases}$$

Ta đưa phương trình trên về ma trận A, vec tơ x và b có dạng:

$$A = \begin{bmatrix} 1 & 7 & -2 \\ 2 & 3 & 7 \\ 1 & 8 & 2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, b = \begin{bmatrix} 21 \\ 2 \\ -1 \end{bmatrix}$$

Sau đó giải phương trình ma trận : $Ax=b$.

Ghi chú 2.5.2. Trước khi giải phương trình trên, ta cần xem xét phương trình này có nghiệm hay không bằng cách so sánh hạng của ma trận A và $A'=[A \ B]$.

Ví dụ 2.5.3. Cho phương trình sau:

$$\begin{bmatrix} 2 & 4 & 3 \\ 5 & 1 & 7 \\ 2 & 3 & 5 \end{bmatrix} X = \begin{bmatrix} 2 & 5 \\ 4 & 1 \\ 3 & 7 \end{bmatrix}$$

```
>> A=[2 4 3;5 1 7;2 3 5];b=[2;4;3];
```

```
>> rank(A)
```

```
ans =
```

```
3
```

```
>> rank([A b])
```

```
ans =
```

```
3
```

Ta nhận thấy $\text{rank}(A)=\text{rank}(A')=m$ do vậy phương trình trên có nghiệm duy nhất.

2.5.3 Sử dụng tính toán symbolic

Ngoài phương pháp đưa về dạng ma trận, ta có thể nhập trực tiếp hệ phương trình và sử dụng hàm **solve**.

Dạng 1: `solve('phương trình','biến')`

Ví dụ 2.5.4. `>> solve('x^2+2*x+a','x')`

`ans =`

$$\begin{aligned} & - (1 - a)^{(1/2)} - 1 \\ & (1 - a)^{(1/2)} - 1 \end{aligned}$$

`>> solve('x^2+2*x+a','a')`

`ans =`

$$- x^2 - 2*x$$

Dạng 2: `solve('phương trình thứ 1',..., 'phương trình thứ n', 'biến thứ 1',..., 'biến thứ n')`

Ví dụ 2.5.5. `>> S=solve('x^2-3*x+y','y-2+8*x+2*a','x','y')`

`S =`

$$\begin{aligned} x: & [2 \times 1 \text{ sym}] \\ y: & [2 \times 1 \text{ sym}] \end{aligned}$$

`>> S.x`

`ans =`

$$\begin{aligned} & (8*a + 113)^{(1/2)}/2 + 11/2 \\ & 11/2 - (8*a + 113)^{(1/2)}/2 \end{aligned}$$

`>> S.y`

`ans =`

```

- 2*a - 4*(8*a + 113)^(1/2) - 42
  4*(8*a + 113)^(1/2) - 2*a - 42

>> S=solve('x^2-3*x+y','y-2+8*x+2*a','x','a')

S =

    a: [2x1 sym]
    x: [2x1 sym]

>> S.a

ans =

    2*(9 - 4*y)^(1/2) - y/2 - 5
    - y/2 - 2*(9 - 4*y)^(1/2) - 5

>> S.x

ans =

    3/2 - (9 - 4*y)^(1/2)/2
    (9 - 4*y)^(1/2)/2 + 3/2

>>

```

2.6 Bài tập

2.1 Giải các phương trình và hệ phương trình sau đây:

(a) $x^3 - 5x^2 + \frac{3}{2}x + 10 = 22$

(b) Giải hệ phương trình sau theo hai cách dùng ma trận hoặc dùng hàm **solve**

2.7 Định thức, giải hệ phương trình tuyến tính bằng định thức 50

và rút ra nhận xét về hai phương pháp trên.

$$\begin{cases} x_1 + 3x_2 + 5x_3 - 4x_4 = 1 \\ x_1 + 8x_2 - 17x_3 + 9x_4 = 20 \\ \frac{2}{3}x_1 + \frac{7}{8}x_2 - 6x_3 - 4x_4 = 20 \\ 12x_1 - 7x_2 + 22x_3 - x_4 = -1 \end{cases}$$

2.2 Cho hệ phương trình sau:

$$\begin{cases} x + 5.000y = 17.0 \\ 15x + 7.501y = 25.503 \end{cases}$$

- (a) Giải hệ phương trình trên.
- (b) Thay giá trị 25.503 bằng các giá trị sau 25.504; 25.505; 25.501 và cho biết nghiệm của nó thay đổi như thế nào so với phương trình ban đầu.
- (c) Cho biết ý nghĩa của hàm `rcond`.

2.7 Định thức, giải hệ phương trình tuyến tính bằng định thức

2.7.1 Định thức

Mệnh đề 2.7.1. Cho $A \in \mathfrak{M}_n(\mathbb{F})$. Ta có:

a/ A khả nghịch $\iff |A| \neq 0$.

b/ A không khả nghịch $\iff |A| = 0$.

*Ta dùng hàm **det** để tính định thức của ma trận vuông.*

Ví dụ 2.7.2. Tính định thức của ma trận sau:

$$\begin{bmatrix} 8 & 9 & 3 \\ 9 & 6 & 5 \\ 1 & 1 & 10 \end{bmatrix}$$

```
>> det(A)
```

```
ans =
```

-316

2.7.2 Giải hệ phương trình tuyến tính bằng định thức

Dựa phương trình về dạng ma trận $AX=b$. Sau đó sử dụng qui tắc Cramer sau đây để giải.

Thuật toán 2.7.3. Qui tắc Cramer:

Kí hiệu:

$$\Delta = |A|$$

$\Delta_j = |A_j|$ (với A_j là ma trận A xóa cột j và thay bằng cột b).

Trường hợp 1: Nếu $\Delta \neq 0$ thì hệ có nghiệm duy nhất: $x_1 = \frac{\Delta_1}{\Delta}, x_2 = \frac{\Delta_2}{\Delta}, \dots, x_n = \frac{\Delta_n}{\Delta}$.

Trường hợp 2: Nếu $\Delta = 0$ và $\exists j \in 1, 2, \dots, n : \Delta_j \neq 0$ thì hệ vô nghiệm.

Trường hợp 3: Nếu $\Delta = 0$ và $\forall j \in 1, 2, \dots, n : \Delta_j = 0$ thì hệ có vô số nghiệm hoặc vô nghiệm.

2.8 Bài tập

2.1 Tính định thức của các ma trận sau đây:

(a)

$$\begin{bmatrix} 2-5i & 23.2 & 3-6i \\ 21.2i & 3-7i & 11 \\ 40-8i & 2+i & 12-23i \end{bmatrix}$$

(b)

$$\begin{bmatrix} 12222.555 & -50000 & 28123445 \\ 197772.22345 & -596959394 & 32344634596.232 \\ -19284785763.2315 & -23454564 & -235856394.111223 \end{bmatrix}$$

Kiểm tra lại bằng tính toán symbolic.

2.2 Viết chương trình biện luận hệ phương trình tuyến tính và thực hiện với hệ phương trình sau:

$$\begin{bmatrix} 2 & 8+m & 11 \\ 3+m & 7 & 2 \\ 9 & 1 & 12-m \end{bmatrix}$$

2.9 Đa thức đặc trưng, trị riêng và vectơ riêng

Định nghĩa 2.9.1. Cho $A \in \mathfrak{M}_n(\mathbb{F})$. Giá trị $\lambda \in \mathbb{F}$ được gọi là giá trị riêng của ma trận A nếu tồn tại vectơ $u \neq 0 \in \mathbb{F}^n$ sao cho $Au = \lambda u$. Khi đó vectơ u được gọi là vectơ riêng của ma trận A ứng với trị riêng λ .

Mệnh đề 2.9.2. Giá trị riêng của ma trận A là nghiệm của phương trình $\det(A - \lambda I) = 0$, gọi là phương trình đặt trưng của A . Ta sử dụng hàm **poly** để tính phương trình đặt trưng. Hàm trên sẽ trả ra một vectơ dòng mà mỗi phần tử là hệ số tương ứng theo thứ tự của đa thức đặc trưng.

Ví dụ 2.9.3. Cho ma trận A như sau:

$$\begin{bmatrix} 3 & -2 & 7 \\ -5 & 9 & 21 \\ 12 & 4 & -3 \end{bmatrix}$$

Tính đa thức đặc trưng của ma trận A .

```
>> p=poly(A)
```

```
p =
```

```
1.0e+003 *
```

```
0.0010    -0.0090    -0.1870    1.7030
```

Đa thức đặc trưng của A là: $\lambda^3 - 9\lambda^2 - 187\lambda + 1703$.

Sau đó ta có thể dùng hàm **roots(p)** để tính nghiệm của đa thức trên.

Hàm **[V,D]=eig(A)** cho ta giá trị riêng và vectơ riêng tương ứng (vectơ riêng trong ma trận V là vectơ cột).

```
>> [V,D]=eig(A)
```

V =

```
    0.3521    0.3915   -0.0549
    0.6761   -0.9163    0.9788
   -0.6472    0.0847    0.1972
```

D =

```
  -13.7070         0         0
         0     9.1952         0
         0         0    13.5117
```

Với trị riêng là -13.707 ta có vectơ riêng tương ứng là (0.3521,0.6761,-0.6472).

2.10 Bài tập

- 2.1 Cho ma trận $A \in \mathfrak{M}_{50}(\mathbb{F})$. Xác định các vectơ x nào khi thực hiện tích Ax thì không làm thay đổi hướng của vectơ x .

CHƯƠNG 3

Giải tích hàm một biến

3.1 Các phép toán tập hợp

3.1.1 Định nghĩa tập hợp và cách khai báo tập hợp trong Matlab

Tập hợp

Đó là những đối tượng được nhóm theo một tính chất nào đó.

Khai báo tập hợp trong Matlab

Một tập hợp trong Matlab được khai báo bằng cách liệt kê dưới dạng một vector (vector dòng hoặc vector cột).

Ví dụ : tập hợp a gồm 5 phần tử có thể khai báo như sau : $A = [1 \ 4 \ 8 \ 9 \ 10]$; $A = [1, 4, 8, 9, 10]$ - vector dòng hoặc $A = [1; 4; 8; 9; 10]$ - vector cột hoặc đó là tập rỗng thì $A = []$.

Để gọi các phần tử trong tập A, ta có : $1 = A(1)$, $4 = A(2)$, $8 = A(3)$, $9 = A(4)$, $10 = A(5)$.

3.1.2 Các phép toán trong tập hợp

Đối với một tập hợp cho trước, những thông tin về nó bao gồm :

1. Biết tập có phải tập rỗng hay không (isempty)

Hàm `isempty(A)` trong Matlab có 1 đối số, giá trị trả về sẽ là 1 nếu A là tập

rỗng, ngược lại thì giá trị trả về là 0.

Ví dụ : Nếu $A = [1 \ 4 \ 8 \ 9 \ 10]$, $\text{isempty}(A)$ sẽ trả về là 0. Nếu $A = []$, $\text{isempty}(A)$ sẽ trả về là 1.

2. Tìm số phần tử trong tập hợp (length)

Hàm $\text{length}(A)$ trong Matlab để cho biết chiều dài của vector A , nó cũng chính là số phần tử của tập hợp A mà ta liệt kê.

Ví dụ : $A = [1 \ 4 \ 8 \ 9 \ 10]$, $\text{length}(A)$ sẽ bằng 5.

3. Biết phần tử cho trước có thuộc tập hợp đã cho hay không (ismember)

Hàm $\text{ismember}(s, A)$ gồm 2 đối số. A là tập hợp cho trước và s là phần tử ta cần xác định có thuộc tập hợp A hay không. Giá trị của $\text{ismember}(s, A)$ sẽ là 1 nếu s thuộc A , ngược lại là 0.

Ví dụ : $A = [1 \ 4 \ 8 \ 9 \ 10]$, $s = 2$. $\text{ismember}(s, A) = 0$. Nếu $s = 8$ thì $\text{ismember}(s, A) = 1$.

Ngoài ra, đối số s ở trên có thể là một tập hợp (vector). Giá trị trả về của $\text{ismember}(s, A)$ sẽ là một vector có chiều dài bằng đúng chiều dài của mảng s và có giá trị tương ứng là 0 (nếu phần tử của s tại vị trí đó không thuộc A) hoặc 1 (nếu phần tử của s tại vị trí đó thuộc A).

Ví dụ : $A = [1 \ 4 \ 8 \ 9 \ 10]$, $s = [0 \ 3 \ 9]$ thì $\text{ismember}(s, A) = [0 \ 0 \ 1]$.

4. Liệt kê tất cả các tập con có k phần tử của tập hợp cho trước (nchoosek)

Hàm $\text{nchoosek}(A, k)$ trong Matlab dùng để tìm tất cả các tập con k phần tử của tập A , tập A gồm có n phần tử. Giá trị trả về là một ma trận gồm có C_n^k dòng và k cột. Mỗi dòng của ma trận kết quả là một tập con gồm k phần tử.

Ví dụ : $A = [1 \ 4 \ 8 \ 9 \ 10]$, $k = 4$ thì

$$\text{nchoosek}(A, k) = \begin{pmatrix} 1 & 4 & 8 & 9 \\ 1 & 4 & 8 & 10 \\ 1 & 4 & 9 & 10 \\ 1 & 8 & 9 & 10 \\ 4 & 8 & 9 & 10 \end{pmatrix}$$

5. Loại bỏ những phần tử trùng nhau trong tập hợp (unique)

Cho một tập hợp A cho trước, có thể trong tập hợp đó có chứa những phần tử được liệt kê nhiều lần. Để loại những phần tử trùng nhau trong tập hợp A , ta dùng hàm $\text{unique}(A)$. Nếu các phần tử A không trùng nhau, thì $\text{unique}(A)$ cho kết quả là A . Ngược lại, kết quả trả về sẽ là một tập hợp (vector) có số

phần tử ít hơn số phần tử của A và được sắp theo thứ tự tăng dần.

Ví dụ : $A = [1\ 4\ 8\ 9\ 10]$, $\text{unique}(A)$ cho kết quả là A. Nếu $A = [1\ 9\ 4\ 8\ 10\ 1\ 8\ 4\ 9\ 10]$, thì $\text{unique}(A)$ sẽ cho kết quả là $[1\ 4\ 8\ 9\ 10]$.

Khi có hai tập hợp cho trước, thì các phép toán giữa chúng gồm có phép hội, giao và hiệu.

6. Hội giữa hai tập hợp (union)

Hàm $\text{union}(A, B)$ dùng để tìm hội của hai tập hợp A và B. Kết quả của $\text{union}(A, B)$ là một tập hợp trong đó liệt kê các phần tử trong A, B mà không có sự liệt kê lặp lại. Ngoài ra kết quả trả về sẽ được sắp theo thứ tự tăng dần.

Ví dụ : $A = [1\ 4\ 8\ 9\ 10]$, $B = [2\ 3\ 5]$ thì $\text{union}(A, B) = [1\ 2\ 3\ 4\ 5\ 8\ 9\ 10]$.

Nếu $A = [1\ 4\ 8\ 9\ 10]$, $B = [4\ 2\ 5\ 3\ 2\ 5]$ thì $\text{union}(A, B) = [1\ 2\ 3\ 4\ 5\ 8\ 9\ 10]$.

Dĩ nhiên, ta có thể áp dụng hàm union để tìm hội của nhiều hơn 2 tập hợp bằng cách lồng các hàm union vào với nhau

Ví dụ : Để tìm hội của 3 tập A, B và C, ta dùng $\text{union}(\text{union}(A,B),C)$

7. Giao giữa hai tập hợp (intersect)

Hàm $\text{intersect}(A, B)$ dùng để tìm phần giao của hai tập A và B. Kết quả của $\text{intersect}(A, B)$ là một tập hợp liệt kê những phần tử nào thuộc cả hai tập A và B, và tập giao này cũng sẽ được sắp xếp theo thứ tự tăng dần.

Ví dụ ; $A = [1\ 4\ 8\ 9\ 10]$, $B = [10\ 5\ 8\ 9]$ thì $\text{intersect}(A, B) = [8\ 9]$. Nếu $A = [1\ 4\ 8\ 9\ 10]$, $B = [5\ 3]$ thì $\text{intersect}(A, B) = []$.

Tương tự như trong hàm union, ta có thể dùng hàm intersect để tìm giao của nhiều hơn hai tập hợp bằng cách lồng các hàm intersect vào với nhau

Ví dụ : Để tìm giao của 3 tập A, B và C, ta dùng $\text{intersect}(\text{intersect}(A,B), C)$.

8. Hàm hiệu giữa hai tập hợp (setdiff) $\text{setdiff}(A, B)$ sẽ trả về tập $A \setminus B$ để tìm những phần tử chỉ chứa trong A mà không chứa trong B. Kết quả cũng được sắp theo thứ tự tăng dần như hai hàm trên.

Ví dụ : $A = [1\ 4\ 8\ 9\ 10]$, $B = [10\ 5\ 8\ 9]$ thì $\text{setdiff}(A, B) = [1\ 4]$. Nếu $A = [10\ 5\ 8\ 9]$, $B = [10\ 5\ 8\ 9]$ thì $\text{setdiff}(A, B) = []$.

9. Hàm tính hội hiệu giao (setxor)

$\text{setxor}(A, B)$ dùng để tính $(A \cup B) \setminus (A \cap B)$, kết quả là một tập hợp chỉ chứa các phần tử có trong A, B mà không chứa trong phần giao của A và B. Cũng giống như trên, kết quả trả về sẽ là một tập được sắp theo thứ tự

Ví dụ : $A = [1\ 4\ 8\ 9\ 10]$, $B = [10\ 5\ 8\ 9]$ thì $\text{setxor}(A, B) = [1\ 4\ 5]$

3.1.3 BÀI TẬP

1. Cho tập $A = \{2, 4, 6, 8, 10, 1, 0\}$
 - a. Cho biết tập A có phải rỗng hay không ?
 - b. Tính tổng tất cả các phần tử trong tập A .
 - c. Xem phần tử 0 có trong tập A hay không?
2. Cho tập A, B gồm có các phần tử sau đây : $A = \{1, 5, 9, 4, 3, 2, 0, 3, 6\}$ và $B = \{3, 4, 8, 9, 10, 4, 6\}$
 - a. Tập A gồm có bao nhiêu phần tử ?
 - b. Dùng lệnh isempty để biết xem A và B có phải là tập trống hay không?
 - b. Nếu A và B là 2 tập khác trống thì tìm $A \cup B$ và $A \cap B$.
 - c. Xem tập A có chứa phần tử trùng nhau hay không ?
3. Cho hai tập A và B như trên bài 2.
 - a. Xem tập $\{0, 2, 4\}$ có thuộc phần giao của hai tập A và B hay không?
 - b. Tìm phần tử lớn nhất và bé nhất của cả hai tập A và B
 - c. Kiểm tra xem tập A có phải là tập con của tập B hay không? Và B có phải là tập con của A hay không?
4. Phát sinh ngẫu nhiên một tập hợp C gồm 100 phần tử bằng hàm rand của Matlab
 - a. Kiểm tra xem tập hợp C có chứa những phần tử nào trùng nhau hay không?
 - b. Kiểm tra xem giá trị lớn nhất của A có lớn hơn 1 hay không? và giá trị bé nhất của A có nhỏ hơn 0 hay không?
 - c. Phát sinh ngẫu nhiên 1 tập B gồm 10 phần tử. Tính $(A \cup B) \setminus (A \cap B)$.
5. Phát sinh ngẫu nhiên 1 tập D gồm 9 phần tử bằng lệnh $D = 3 - 3 * \text{rand}(1, 9)$
 - a. Kiểm tra xem tập hợp đó có thuộc tập số nguyên hay không?
 - b. Liệt kê tất cả các tập con gồm 4 phần tử của tập D .
 - c. Liệt kê tất cả các tập con của D .
6. Mỗi sinh viên nam khoa Toán-Tin học đều chơi bóng đá hoặc bóng chuyền. Biết rằng có 350 bạn chơi bóng đá, 260 bạn chơi bóng chuyền và 70 bạn chơi cả hai môn thể thao này.
Hỏi khoa Toán-Tin học có bao nhiêu sinh viên nam (dùng các phép toán tập hợp)?
7. Cho $x = 1 : 180$, $y = 1 : 150$. Dùng chỉ số logic và hàm mod, liệt kê các phần tử của tập hợp A = các ước số nguyên dương của 180 và của tập hợp B = các ước số nguyên dương của 150.
Xác định các tập hợp $A \cup B$, $A \cap B$, $A \setminus B$, $B \setminus A$

8. Phát sinh ngẫu nhiên hai số nguyên dương m và n không vượt quá 20. Phát sinh tập hợp A gồm m phần tử nguyên ngẫu nhiên trong khoảng $[-10, 10]$ và B gồm n phần tử ngẫu nhiên trong khoảng $[-20, 20]$.

Xác định: a. $A \subset B$ hay $B \subset A$ hay $A=B$ không?

b. $A \setminus B$; $B \setminus A$; $A \cup B$; $A \cap B$; c. A^2 ; B^2 ; $A \times B$; $B \times A$; d. Nhập một số nguyên x .

Kiểm tra $x \in A$ hay $x \in B$ không? 9. Cho $A = \{1, 5, 9, 4, 3, 2, 0, 3, 6\}$ và $B = \{3, 4, 8, 9, 10, 4, 6\}$ và $C = \{0, 2, 4\}$. Tìm

a. $(A \cup B) \cap C$ b. $(A \cup B \cup C) \setminus (A \cap B \cap C)$ c. $A \cap C \cup B$ 10. Cho tập A , B và C như trên.

a. Tập nào có nhiều phần tử nhất.

b. Tìm giá trị lớn nhất trong cả 3 tập hợp trên.

c. Liệt kê tất cả các tập con của tập C . Xem tập con nào của tập C là tập con của A .

3.2 symbolic math cơ bản

Matlab cũng như một số phần mềm như Maple, Mathematica, Maxima... có thể hỗ trợ người dùng về khả năng tính toán hình thức. Chúng ta có thể khai báo một biến hình thức trong Matlab bằng lệnh SYM hoặc khai báo cùng lúc nhiều biến bằng lệnh SYMS. Ngoài ra, chúng ta có thể khai báo một hàm theo các biến bằng số lệnh INLINE.

```
Ví dụ 3.2.1. >> sym x;
>> syms y z t;
>> f=inline(sin(x*y*z))
```

f =

```
Inline function:
f(x,y,z) = sin(x.*y.*z)
```

Một số hàm và biến/hằng toán học trong Matlab

Tên hàm	Ghi chú	Tên hàm	Ghi chú
cos(x)		exp(x)	hàm e mũ
sin(x)		log(x)	logarit tự nhiên

tan(x)		log10(x)	logarit cơ số 10
acos(x)	$\cos^{-1}(x)$	abs(x)	trị tuyệt đối
asin(x)	$\sin^{-1}(x)$	sqrt(x)	căn bậc hai
atan(x)	$-\pi/2 \leq \tan^{-1} \leq \pi/2$	real(x)	phần thực
atan2(x)	$-\pi \leq \tan^{-1} \leq \pi$	imag(x)	phần ảo
cosh(x)	$(e^x + e^{-x})/2$	conj	số phức liên hợp
sinh(x)	$(e^x - e^{-x})/2$	round	làm tròn
tanh(x)	$(e^x - e^{-x})/(e^x + e^{-x})$	fix	làm tròn
acosh(x)	$\cosh^{-1}(x)$	sign(x)	1(dương)/0/-1(âm)
asinh(x)	$\sinh^{-1}(x)$	mod(y,x)	phần dư của y/x
atanh(x)	$\tanh^{-1}(x)$	rem(y,x)	phần dư của y/x
max	giá trị lớn nhất và chỉ số của nó	eval(f)	giá trị một biểu thức
min	giá trị bé nhất và chỉ số của nó	feval(f,a)	giá trị của hàm
sum	tổng	polyval	giá trị của hàm đa thức
sum	tổng	polyval	giá trị của hàm đa thức
prod	tích	poly	giá trị của hàm đa thức
norm	chuẩn vector hoặc ma trận	sort	sắp xếp theo thứ tự tăng
roots	ng nghiệm đa thức	poly	giá trị của hàm đa thức
i,j	$\sqrt{-1}$	pi	π
break	thoát vòng lặp while/for	Inf, inf	∞
nargin	số đối số vào	nargout	Không-là-một-số (không xác định)
varargin	danh sách các đối số vào	varargout	danh sách đối số ra

3.3 Các bài toán dãy số và chuỗi số

3.3.1 Khái niệm về dãy số, chuỗi số và cách khai báo trong matlab

Dãy số và chuỗi số Dãy số có thể xem là một danh sách các số có thứ tự $x_1, x_2, x_3, \dots, x_n, \dots$

Trong đó a_1 là giá trị đầu tiên, a_2 là giá trị thứ 2. Tổng quát a_n là giá trị thứ n. Đối với dãy vô hạn, giá trị tiếp theo sau x_n luôn tồn tại là x_{n+1} . Người ta kí hiệu như sau $\{x_n\}$ hoặc $\{x_n\}_{n=1}^{\infty}$

Ví dụ : Ta có dãy số $\{x_n\} = \frac{1}{n}$. Trong đó $x_1 = 1, x_2 = \frac{1}{2}, \dots, x_k = \frac{1}{k}, x_{k+1} = \frac{1}{k+1}, \dots$

Chuỗi số có thể được định nghĩa thông qua dãy số, trong đó phần tử thứ n là $s_n = \sum_{i=1}^n x_i$.

Ví dụ : Chuỗi $\sum_{n=1}^{\infty} \frac{1}{n}$ được xem như một dãy số có $s_1 = \sum_{n=1}^1 \frac{1}{n} = 1, s_2 = \sum_{n=1}^2 \frac{1}{n} = 1 + \frac{1}{2}, \dots, s_k = \sum_{n=1}^k \frac{1}{n} = 1 + \frac{1}{2} + \dots + \frac{1}{k}, s_{k+1} = s_k + \frac{1}{k+1}$

Cách khai báo dãy số và chuỗi số trong Matlab

Để khai báo chuỗi trong matlab ta làm như sau : ví dụ trong trường hợp chuỗi $x_n = \frac{1}{n}$ ta làm như sau :

`syms n; xn = 1/n ;` Tương tự như thế đối với chuỗi số, chuỗi s_n được khai báo như sau :

`syms k n; sn = symsum(1/k, 1, n) ;`

và đối với chuỗi hàm $\sum_{k=1}^{\infty} x^k$ ta khai báo như sau :

`syms x, k, n; sn=symsum(x^k, k, 1, n)`

3.3.2 Một số hàm về xử lý dãy số và chuỗi số trong Matlab

1. Hàm tính giới hạn (limit)

Hàm limit trong Matlab dùng để tính giới hạn của dãy số theo nhiều cách sau đây :

`limit(xn,n,a)` dùng để tính giới hạn của dãy x_n khi n dần về giá trị a

`limit(xn,n,a,right)` ; `limit(xn,n,a,left)` dùng tính giới hạn một bên khi x tiến về a từ 2 phía.

Ví dụ : Cho dãy $x_n = 1/n$

Giới hạn của x_n khi $n \rightarrow \infty$ được tính bằng Matlab `limit(xn, n, inf) = 0`, khi $n \rightarrow 5$, `limit(xn, n, 5) = 1/5`.

2. Hàm tính tổng theo biến (symsum)

Như trong phần ví dụ trong chuỗi, hàm `symsum` được dùng để tính tổng theo một biến. Cú pháp của hàm `symsum` trong Matlab như sau :

`symsum(S,v,a,b)`, trong đó S là biểu thức phụ thuộc vào v ($S = \frac{1}{v}$), hoặc S là một hàm số phụ thuộc vào v ($S = x^v$, x là biến). v là chỉ số ta muốn tính tổng theo của S theo v từ a đến b .

3. Biểu diễn dãy (plot)

Dãy số hay chuỗi số có thể gồm vô hạn phần tử, tuy nhiên để minh họa trong máy, ta sẽ chọn đến phần tử thứ N nào đó của dãy hay chuỗi để minh họa.

Ví dụ : Cho $N = 1000$, ta làm như sau : `for i = 1 : N`

`X(i) = 1/i; end; plot(X);`

3.3.3 Bài tập

1 Dùng đồ thị mô tả các dãy/chuỗi số sau để xem chúng hội tụ hay phân kỳ (cho trước n đủ lớn). Nếu chúng hội tụ, ước lượng giá trị hội tụ.

a. $a_n = (-1)^n \frac{n+1}{n}$ b. $a_n = 2 + \left\{\frac{-2}{\pi}\right\}^n$ c. $a_n = \frac{\sin(n)}{\sqrt{n}}$ d. $a_n = \frac{n^3}{n!}$ e. $a_n = (3^n + 5^n)^{1/n}$ f. $\frac{1.2.3...\cdot(2n-1)}{(2n)^n}$ g. $\frac{1.2.3...\cdot(2n-1)}{n!}$

2 Tính tổng riêng đầu tiên của các chuỗi sau. Vẽ trên cùng hệ trục dãy số hạng tử của chuỗi và dãy giá trị các tổng riêng của chuỗi. Xét xem chúng hội tụ hay phân kỳ. Nếu hội tụ thì tính giá trị hội tụ. Nếu phân kỳ thì giải thích tại sao.

a. $\sum_{n=1}^{\infty} \frac{12}{(-5)^n}$ b. $\sum_{n=1}^{\infty} \frac{2n^2-1}{n^2+1}$ c. $\sum_{n=1}^{\infty} \tan(n)$ d. $\sum_{n=1}^{\infty} (0.6)^{n-1}$ e. $\sum_{n=1}^{\infty} \frac{1}{15^{1.5}} - \frac{1}{(n+1)^{1.5}}$ f. $\sum_{n=2}^{\infty} \frac{1}{n(n-1)}$ g. $\sum_{n=1}^{\infty} \frac{2n}{3n+1}$

3 Xét xem các chuỗi sau hội tụ hay phân kỳ bằng định nghĩa dùng vòng lặp while khi tăng n với epsilon đủ nhỏ cho trước. Nếu chúng hội tụ, tính giá trị hội tụ.

a. $3 + 2 + \frac{4}{3} + \frac{8}{9} + \dots$ b. $\frac{1}{8} - \frac{1}{4} + \frac{1}{2} - 1 + \dots$ c. $\sum_{n=1}^{\infty} \frac{(-6)^{n-1}}{5^{n-1}}$ d. $\sum_{n=0}^{\infty} \frac{\pi^n}{3^{n+1}}$ e. $\sum_{n=1}^{\infty} \frac{(3^n+2^n)}{6^n}$ f. $\sum_{n=1}^{\infty} (0.8^{n-1} - 0.3^n)$

4. Tìm giá trị n vừa đủ để chuỗi hội tụ (dùng vòng lặp while để tăng n).

a. $\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^4}$ (sai số nhỏ hơn 0.001).

b. $\sum_{n=1}^{\infty} \frac{(-2)^n}{n!}$ (sai số nhỏ hơn 0.01).

c. $\sum_{n=1}^{\infty} \frac{(-1)^n n}{4^n}$ (sai số nhỏ hơn 0.002).

5. Ước lượng giá trị của tổng chuỗi đến 4 chữ số thập phân (theo tư tưởng của bài trên)

a. $\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^5}$ b. $\sum_{n=1}^{\infty} \frac{(-1)^n n}{8^n}$ c. $\sum_{n=1}^{\infty} \frac{(-1)^n}{3^n n!}$

6. Tìm giới hạn của dãy số sau :

a. $a_n = \{\sqrt{n-3}\}_{n=3}^{\infty}$ b. $a_n = \left\{\frac{n}{n+1}\right\}_{n=1}^{\infty}$ c. $a_n = \left\{\cos \frac{n\pi}{6}\right\}_{n=0}^{\infty}$

7. Liệt kê 5 phần tử đầu tiên của dãy số sau :

a. $a_n = 1 - (0.2)^n$ b. $a_n = \frac{n+1}{3n-1}$ c. $a_1 = 3, a_{n+1} = 2a_n - 1$

8. Liệt kê 6 phần tử đầu tiên của dãy số sau đây và cho nó có tồn tại giới hạn hay không. Nếu dãy số sau tồn tại giới hạn thì giới hạn đó.

$a_n = n2n + 1$

9. Nếu 1000 đôla được đầu tư với lợi nhuận 0.6%. Sau n năm thì vốn đầu tư sẽ lên đến $1000(1.6)^n$

a. Liệt kê 5 phần tử đầu tiên của dãy số trên.

b. Dãy số trên có hội tụ hay không? Giải thích.

10. Chỉ ra một giá trị r để dãy nr^n hội tụ.

3.4 Các bài toán vi phân hàm một biến

3.4.1 Vi phân hàm một biến

Cho hàm số thực $f(x)$ trên khoảng mở (a, b) và $x \in (a, b)$. Ta nói f khả vi tại x nếu và chỉ nếu $\lim_{h \rightarrow 0} \frac{f(x+h)-f(x)}{h}$ tồn tại và là số thực. Khi đó ta ký hiệu giới hạn này là $f'(x)$ và gọi là đạo hàm của f tại x .

3.5 Các bài toán giới hạn hàm số và tính liên tục của hàm số

3.5.1 Giới hạn của hàm số

Trong matlab ta dùng lệnh `limit` để tính giới hạn của hàm số. Cụ thể:

`LIMIT(f,x,a)`: Tính giới hạn của hàm số f khi x tiến về a .

`LIMIT(f,x,a,'right')` hoặc `LIMIT(f,x,a,'left')`: Tính giới hạn trái hoặc giới hạn phải của hàm số khi x tiến về a .

Ví dụ 3.5.1. Cho $f(x) = \frac{\sin(x)}{x}$, tìm giới hạn của f khi $x \rightarrow 0$

Trong matlab ta có thể làm như sau:

```
>> syms x
```

```
>> limit(sin(x)/x,x,0)
```

```
ans =
```

```
1
```

Ngoài ra matlab còn có thể tính giới hạn trái và giới hạn phải của một hàm số.

Ví dụ 3.5.2. Cho $f(x) = \frac{1}{x}$, tìm giới hạn của f khi $x \rightarrow 0^+$ và $x \rightarrow 0^-$

Trong matlab ta có thể làm như sau:

```
>> syms x
>> limit(1/x,x,0,'right')
```

```
ans =
```

```
inf
```

```
>> limit(1/x,x,0,'left')
```

```
ans =
```

```
-inf
```

Ngoài ra chúng ta có thể áp dụng hàm LIMIT để tính đạo hàm của một hàm số bằng định nghĩa của đạo hàm.

Định nghĩa 3.5.3. Đạo hàm của một hàm số f tại a , ký hiệu là $f'(a)$ là

$$f'(a) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

nếu giới hạn này tồn tại.

Ví dụ 3.5.4. Cho hàm số $f(x) = \arctan(x)$, tìm $f'(a)$ với $a \in \mathbf{R}$?

```
>> syms a
>> limit((atan(a+h)-atan(a))/h,h,0)
```

```
ans =
```

```
1/(1+a^2)
```

3.5.2 Sự liên tục của hàm số

Định nghĩa 3.5.5. Hàm số f liên tục tại a nếu

$$\lim_{x \rightarrow a} f(x) = f(a).$$

Như vậy để một hàm số liên tục tại một điểm thì hàm số đó phải thỏa ba điều kiện sau:

1. $f(a)$ xác định với a là một phần tử trong tập xác định,
2. $\lim_{x \rightarrow a} f(x)$ tồn tại,
3. $\lim_{x \rightarrow a} f(x) = f(a)$.

Dựa vào định nghĩa hàm số liên tục tại một điểm, chúng ta có thể sử dụng các câu lệnh trong matlab như sau:

1. Tính giới hạn của hàm số f khi $x \rightarrow a$ bằng lệnh LIMIT
2. Tính giá trị hàm số tại a bằng lệnh SUBS

Ví dụ 3.5.6. Cho

$$f(x) = \begin{cases} \frac{x^3 - 2x^2 - x + 2}{x - 2} & x \neq 2 \\ 2 & x = 2 \end{cases}$$

```
>> syms x
>> limit((x^3-2*x^2-x+2)/(x-2), x, 2)
```

ans =

3

Vì $\lim_{x \rightarrow 2} f(x) = 3 \neq 2 = f(2)$ nên f không liên tục tại $x = 2$. Ngược lại, f liên tục tại tất các điểm $x \neq 2$. Cụ thể, xét sự liên tục của f tại $x = 0$:

```
>> syms x
>> limit((x^3-2*x^2-x+2)/(x-2), x, 0)
```

ans =

-1

Bài toán 3.5.7. Tìm hiểu lệnh SUBS trong trường hợp có nhiều biến.

Áp dụng lệnh LIMIT để tính giới hạn hàm số trong trường hợp hàm nhiều biến.

Bài tập 3.5.8. Khảo sát tính liên tục của hàm số tại a . Vẽ đồ thị hàm số.

1. $f(x) = \ln|x - 2| \quad a = 2$
2. $f(x) = \begin{cases} \frac{1}{x-1} & x \neq 1 \\ 2 & x = 1 \end{cases} \quad a = 1$
3. $f(x) = \begin{cases} e^x & x < 0 \\ x^2 & x \geq 1 \end{cases} \quad a = 0$
4. $f(x) = \begin{cases} \frac{x^2 - x}{x^2 - 1} & x \neq 1 \\ 1 & x = 1 \end{cases} \quad a = 1$
5. $f(x) = \begin{cases} \cos(x) & x < 1 \\ 0 & x = 1 \\ 1 - x^2 & x > 0 \end{cases} \quad a = 0$
6. $f(x) = \begin{cases} \frac{2x^2 - 3x - 3}{x - 3} & x \neq 3 \\ 0 & x = 3 \end{cases} \quad a = 3$

Bài tập 3.5.9. Vẽ đồ thị hàm số và xác định các điểm bất liên tục của các hàm số.

1. $y = \frac{1}{1 + e^{1/x}}$
2. $y = \ln(\tan^2 x)$

Bài tập 3.5.10. Sử dụng matlab chứng minh các hàm số sau liên tục trên \mathbf{R} ?

1. $f(x) = \begin{cases} x^2 & x < 1 \\ \sqrt{x} & x \geq 1 \end{cases}$
2. $f(x) = \begin{cases} \sin(x) & x < \pi/4 \\ \cos(x) & x \geq \pi/4 \end{cases}$

Bài tập 3.5.11. Xác định $f'(0)$ có tồn tại hay không?

1. $f(x) = \begin{cases} x \sin \frac{1}{x} & x \neq 0 \\ 0 & x = 0 \end{cases}$

$$2. \quad f(x) = \begin{cases} x^2 \sin \frac{1}{x} & x \neq 0 \\ 0 & x = 0 \end{cases}$$

$$3. \quad f(x) = \arctan \left(\frac{a^2 - x^2}{a^2 + x^2} \right)$$

$$4. \quad f(x) = \frac{1}{x} \arctan \left(\ln \frac{1}{x^2} \right)$$

3.6 Các bài toán tích phân hàm một biến

Trong matlab, để tính tích phân hàm một biến chúng ta dùng lệnh INT.

3.6.1 Tích phân bất định

INT(f,x): Tính tích phân bất định của hàm f theo biến x .

Ví dụ 3.6.1. Tính tích phân bất định của hàm số $f(x) = x^3 \arctan(x)$?

```
>> syms x
>> int(x^3*atan(x),x)
```

ans =

```
1/4*x^4*atan(x)-1/12*x^3+1/4*x-1/4*atan(x)
```

Chúng ta có thể rút gọn kết quả tính hình thức bằng hàm SIMPLE hoặc SIMPLIFY.

3.6.2 Tích phân xác định

INT(f,x,a,b): Tính tích phân xác định của hàm f theo biến x với cận lấy tích phân từ a đến b .

Ví dụ 3.6.2. Tính tích phân xác định

$$I = \int_0^{\pi/4} x^3 \arctan(x) dx$$

```
>> syms x
>> I=int(x^3*atan(x),x,0,pi/4)
```

```
I =
```

```
1/1024*pi^4*atan(1/4*pi)-1/768*pi^3+1/16*pi-1/4*atan(1/4*pi)
```

Kết quả ở trên cho thấy matlab hiểu π như là một biến hình thức. Do đó để biểu diễn kết quả dưới dạng số thực ta dùng lệnh EVAL như sau:

```
>> I=eval(I)
```

```
I =
```

```
0.0529
```

3.6.3 Tích phân số

Trong thực tế, nhiều tích phân không thể tính nguyên hàm được. Trong trường hợp đó, chúng ta sử dụng tích phân số để tính tích phân xác định. Matlab cung cấp cho chúng ta hàm tính tích phân số: QUAD. Hàm QUAD dùng để tính tích phân số bằng phương pháp cầu phương. Sinh viên có thể tìm hiểu phương pháp tích phân cầu phương trong các giáo trình Giải tích số.

Ví dụ 3.6.3. Tính tích phân xác định sau bằng phương pháp tích phân cầu phương gần đúng

$$I = \int_0^1 \frac{e^x \arctan(x^2)}{\cos(x)} dx.$$

```
>> F = inline('exp(x).*atan(x.^2)./cos(x)');
>> Q=quad(F,0,1)
```

```
Q =
```

```
0.9230
```

Bài tập 3.6.4. Viết một hàm hoặc đoạn chương trình tính xấp xỉ tích phân xác định bằng phương pháp điểm giữa sau:

$$\int_a^b f(x)dx \approx \sum_{i=1}^n f(\bar{x}_i)\Delta x,$$

trong đó $\Delta x = \frac{b-a}{n}$, và $\bar{x}_i = \frac{1}{2}(x_{i-1} + x_i)$. Áp dụng tính các tích phân xác định trong khoảng (a, b) chính xác đến tám chữ số thập phân. So sánh kết quả của phương pháp này với kết quả bằng lệnh QUAD.

1. $\int_0^1 \frac{e^{x^2} dx}{1 + e^{2x}}$
2. $\int_2^{10} \sqrt{x^5 + 1} dx$
3. $\int_0^{\pi/2} \tan^4 x dx$
4. $\int_0^1 \cos x^2 dx$
5. $\int_1^5 x^2 e^{-x^2} dx$

Bài tập 3.6.5. Hàm tích phân sine

$$\text{Si}(x) = \int_0^x \frac{\sin t}{t} dx$$

có vai trò quan trọng trong kỹ thuật điện.

1. Vẽ đồ thị của Si.
2. Tìm những điểm mà tại đó hàm này đạt cực đại địa phương.
3. Tìm tọa độ của điểm uốn đầu tiên phía bên phải gốc tọa độ.
4. Hàm số có tiệm cận ngang hay không?

Bài tập 3.6.6. Sử dụng đồ thị ước lượng giao điểm của hàm số với trục hoành Ox và tính xấp xỉ diện tích nằm bên dưới đường cong và bên trên trục Ox của các hàm số bên dưới.

1. $y = x + x^2 - x^4$.

2. $y = 2x + 3x^4 - 2x^6$.

Bài tập 3.6.7. Cho một vật thể có biên giới hạn bởi trục Oy , đường thẳng $y = 1$, và đường cong $y = \sqrt[4]{x}$. Tính diện tích của vật thể?

Bài tập 3.6.8. Cho đường cong có phương trình $y^2 = x^2(x + 3)$. Đồ thị của đường cong này có một phần tạo hình một hình vòng cung. Hãy vẽ đồ thị và tính diện tích của hình tạo bởi hình vòng cung đó.

3.6.4 Các hàm trong Matlab dùng cho bài toán vi phân hàm một biến

1. Đạo hàm cấp k theo một biến (diff)

Hàm diff dùng để tìm đạo hàm cấp k của hàm số $f(x, y)$ theo biến x theo cú pháp diff(f, x, k) hay theo biến y theo cú pháp diff(f, y, k). Nhưng khi hàm số chỉ phụ thuộc vào duy nhất một biến x thì ta có diff(f, k).

Ví dụ : Tìm đạo hàm cấp 1 của hàm số $f(x) = x^2 - \cos(x)$, ta làm như sau :
syms x; $f = x^2 - \cos(x)$; diff(f) = 2*x - sin(x)

Tìm đạo hàm cấp 3 của hàm số $f(x, y) = x^2 - \sin(x) - x * y^2$ theo biến y, ta làm như sau :
syms x y; $f = x^2 - \sin(x) - x * y^2$; diff(f, y, 3) = 0;

2. Khai triển Taylor Khai triển Taylor dùng để xấp xỉ một hàm số có đạo hàm ở mọi cấp thành một đa thức bậc n trong lân cận một điểm cho trước, với sai số cho phép. Hàm Taylor trong Matlab taylor có những cú pháp sau đây :

taylor(f(x)) khai triển Taylor hàm f(x) đến bậc 5, trong vùng lân cận 0.

taylor(f(x),n) khai triển Taylor hàm f(x) đến bậc n-1, trong vùng lân cận 0.

taylor(f(x),a) khai triển Taylor hàm f(x) đến bậc 5, trong vùng lân cận a.

taylor(f(x),a,n) khai triển Taylor hàm f(x) đến bậc n-1, trong vùng lân cận a.

3.6.5 Bài tập

1. Tìm đạo hàm cấp 1 của các hàm số sau đây :

a. $f(x) = x^6$ b. $f(x) = \sqrt{x}$ c. $f(x) = x\sqrt{x}$

2. Tìm đạo hàm cấp 1 và cấp 2 của các hàm số sau đây :

a. $f(x) = x^4 - 3x^3 - 16x$ b. $f(x) = \sqrt{x} + x^{\frac{1}{3}}$ c. $f(x) = \sin(x)x + x^4$

3. Phương trình chuyển động của một chất điểm có dạng $s = t^3 - 3t$. Trong

đó s có đơn vị là mét và t có đơn vị là giây. Tìm

a. Vận tốc và gia tốc của chuyển động.

b. Gia tốc chuyển động sau 2 giây.

c. Gia tốc chuyển động khi vận tốc bằng 0.

4. Một chất điểm chuyển động có dạng phương trình $s = 2t^3 - 7t^2 + 4t + 1$.

Trong đó s có đơn vị là mét và t có đơn vị là giây. Tìm

a. Vận tốc và gia tốc của chuyển động.

b. Gia tốc chuyển động sau 1 giây.

c. Vẽ đồ thị của chuyển động, cùng với vận tốc và gia tốc.

5. Tìm trên đường cong $y = 2x^3 + 3x^2 - 12x + 1$ điểm mà tiếp tuyến với đường cong tại điểm đó song song với trục hoành. 6. Phương trình $y'' + y' - 2y = x^2$ được gọi là phương trình vi phân vì nó chứa hàm số chưa biết $y(x)$, đạo hàm cấp 1 và cấp 2 của nó. Tìm 3 hệ số A, B và C để hàm số $y = Ax^2 + Bx + C$ là nghiệm của phương trình vi phân trên.

7. Với những giá trị nào của x thì đồ thị hàm số $f(x) = x^3 + 3x^2 + x + 3$ có tiếp tuyến song song với trục hoành.

8. So sánh đạo hàm của 2 hàm số $f(x) = e^x$ và $g(x) = x^e$. Hàm số nào sẽ tăng nhanh hơn khi x càng lớn?

9. Tìm đạo hàm cấp 1 và cấp 2 của hàm số $f(x) = 2x - 5x^{3/4}$ và nhận xét.

10. Tìm vị trí trên đường cong $f(x) = 1 + 2e^x - 3x$ sao cho tiếp tuyến tại đó song song với đường thẳng $3x - y = 5$. Vẽ trên cùng đồ thị hai đường hàm số trên bằng hàm ezplot của Matlab.

11. Cho hàm số $f(x) = \frac{x^2}{x+1}$. Tìm $f''(1)$.

12. Một nhà máy sản xuất những bó sợi với chiều rộng cố định. Cổ lượng sợi q (đơn vị yards) được bán là hàm của giá bán p (đơn vị đôla), có thể biểu diễn dưới dạng $q = f(p)$. Tổng thu nhập với giá bán cố định p là $R(p) = pf(p)$.

a. Có ý nghĩa gì khi nói là $f(20) = 20000$ và $f'(20) = -350$

b. Dùng câu a, tính $R'(20)$

13. Khai triển Taylor hàm số $f(x)$ trong lân cận 0 (bậc 5).

a. $f(x) = e^x$ b. $f(x) = \sin(x)$ c. $f(x) = \cos(x)$ d. $f(x) = \ln(x)$

Sau đó so sánh giá trị xấp xỉ và giá trị đúng tại các điểm 0.4 và 0.1.

14. Khai triển Taylor hàm số $f(x)$ trong lân cận 1 đến cấp 9

a. xe^x b. $\cosh(x)$ c. $\ln(1+x)$ d. $\frac{x}{4+x^2}$