

THỰC HÀNH MATLAB

TÀI LIỆU THỰC HÀNH LABORATORY

Nguyễn Thành Nhật
Hoàng Văn Hà
Bùi Xuân Thắng

8 - 2010

Lời mở đầu

MATLAB là một hệ tính toán, lập trình cao cấp với thư viện hàm phong phú, đa dạng. Chính vì thế, MATLAB được sử dụng là một trong những công cụ tính toán hiệu quả trong giảng dạy, học tập và nghiên cứu trong các trường đại học. Quyển sách này ra đời để phục vụ cho môn học *Thực hành Laboratory* của sinh viên Khoa Toán-Tin học Trường Đại học Khoa học Tự nhiên Đại học Quốc gia Thành phố Hồ Chí Minh. Ngoài ra, giáo trình này cũng có thể dùng làm tài liệu cho sinh viên các môn học như *Tin học tính toán* hay *MATLAB và ứng dụng* của các ngành Khoa học-Kỹ thuật.

Quyển sách này có thể dùng cho sinh viên tự học hoặc giảng viên dạy trực tiếp cho sinh viên ngay trong phòng thực hành. Nội dung gồm cả phần tóm tắt lý thuyết và hướng dẫn các bước thực hành từ cơ bản đến nâng cao. Song song đó là hệ thống các bài tập phong phú được soạn và sưu tầm bởi các giảng viên nhiều kinh nghiệm.

Xin cảm ơn TS Phạm Thế Bảo, ThS Vũ Đỗ Huy Cường, ThS Đặng Thị Bạch Tuyết, GV Bùi Ngọc Nam, GV Nguyễn Trần Lan Anh của khoa Toán-Tin học, Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh đã tham gia đóng góp soạn giáo trình này thật công phu. Tuy nhiên với phiên bản lần đầu tiên này, quyển sách không tránh khỏi những sai sót nhất định. Mong quý độc giả đóng góp các ý kiến đáng quý để các tác giả có thể hoàn thiện hơn quyển giáo trình này cho những lần tái bản sau.

Thành phố Hồ Chí Minh, tháng 09 năm 2010.

Nhóm tác giả

Mục lục

Lời mở đầu	1
1 Matlab cơ bản	5
1.1 Giới thiệu Matlab	5
1.2 Phép toán, biến, vector, ma trận	8
1.2.1 Biến	8
1.2.2 Phép toán	9
1.2.3 Vector	12
1.2.4 Ma trận	16
1.3 Biểu thức Logic	19
1.3.1 Các toán tử logic	19
1.3.2 Vectơ và biểu thức logic	20
1.3.3 Các hàm logic: All, Any và Find	20
1.4 Lệnh điều kiện và vòng lặp	21
1.4.1 Lệnh IF	21
1.4.2 Lệnh FOR	22
1.4.3 Lệnh WHILE	23
1.4.4 Lệnh SWITCH ... CASE	24
1.4.5 Script và Hàm	24
1.5 Vẽ đồ thị	28
1.5.1 Vẽ đồ thị trong 2-D	28
1.5.2 Vẽ đồ thị trong 3-D	33
2 Đại số tuyến tính	38
2.1 Các phép toán ma trận, các phép biến đổi sơ cấp	38
2.1.1 Các phép toán ma trận	38

2.1.2	Các phép biến đổi sơ cấp	42
2.2	Ma trận nghịch đảo, Phương trình ma trận và Hệ phương trình tuyến tính	44
2.2.1	Ma trận nghịch đảo	44
2.2.2	Ma trận giả nghịch đảo	46
2.2.3	Giải phương trình ma trận	49
2.3	Hệ phương trình tuyến tính	51
2.3.1	Đưa về dạng ma trận	51
2.3.2	Sử dụng tính toán symbolic	52
2.4	Định thức, giải hệ phương trình tuyến tính bằng định thức	54
2.4.1	Định thức	54
2.4.2	Giải hệ phương trình tuyến tính bằng định thức	55
2.5	Đa thức đặc trưng, trị riêng và vectơ riêng	55
3	Giải tích hàm một biến	58
3.1	Các phép toán tập hợp	58
3.1.1	Định nghĩa tập hợp và cách khai báo tập hợp trong Matlab	58
3.1.2	Các phép toán trong tập hợp	59
3.2	symbolic math cơ bản	66
3.2.1	Khái niệm chung	66
3.2.2	Khởi động TOOLBOX	66
3.3	Các bài toán dãy số và chuỗi số	71
3.3.1	Khái niệm về dãy số, chuỗi số và cách khai báo trong matlab	71
3.3.2	Một số hàm về xử lý dãy số và chuỗi số trong Matlab	72
3.4	Các bài toán giới hạn hàm số và tính liên tục của hàm số	76
3.4.1	Giới hạn của hàm số	76
3.4.2	Sự liên tục của hàm số	77
3.5	Các bài toán tích phân hàm một biến	80
3.5.1	Tích phân bất định	80
3.5.2	Tích phân xác định	80
3.5.3	Tích phân số	81
3.5.4	Các hàm trong Matlab dùng cho bài toán vi phân hàm một biến	83

4	Cơ học và Xác suất - Thống kê	88
4.1	Cơ học	88
4.1.1	Động học chất điểm	88
4.1.2	Sóng cơ học	93
4.2	Xác suất - Thống kê	97
4.2.1	Tính toán xác suất cơ bản	97
4.2.2	Thống kê mô tả	99
4.2.3	Kiểm định giả thiết: z - test và t - test	99
5	Giao diện đồ họa người dùng – GUI	102
5.1	Tổng quan về các tệp tin GUI	102
5.1.1	Tệp tin M và tệp tin FIG	102
5.1.2	Cấu trúc tệp tin M của GUI	102
5.2	Ví dụ tạo một GUI đơn giản	103
5.3	Một số component trong GUI	103

CHƯƠNG 1

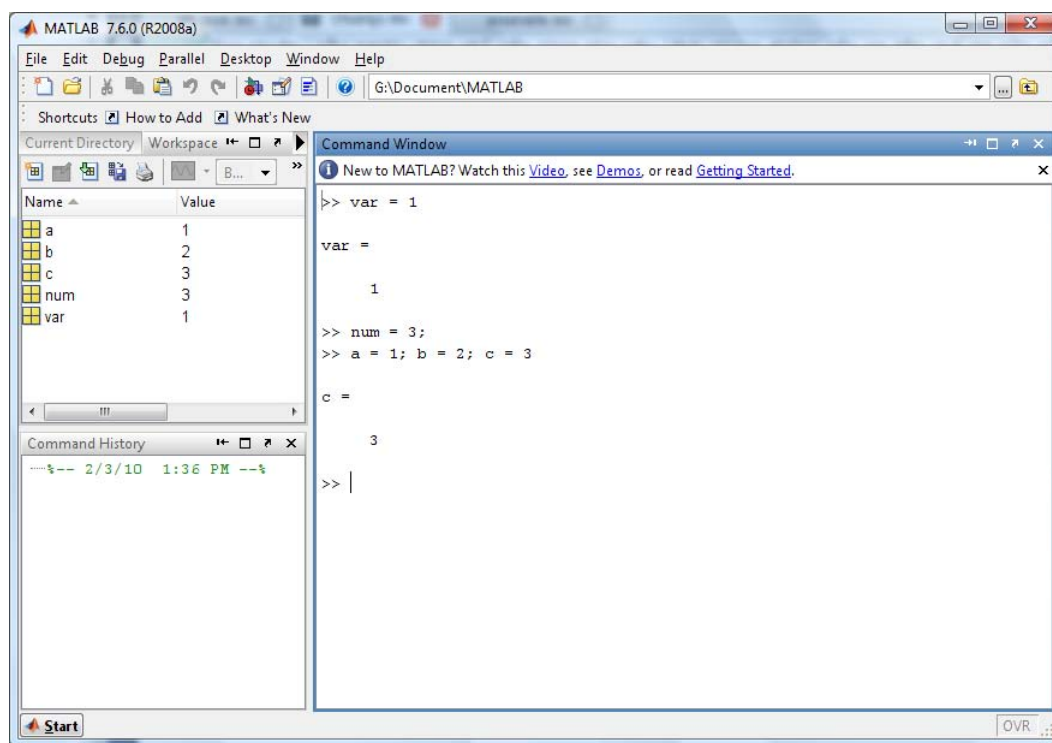
Matlab cơ bản

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,

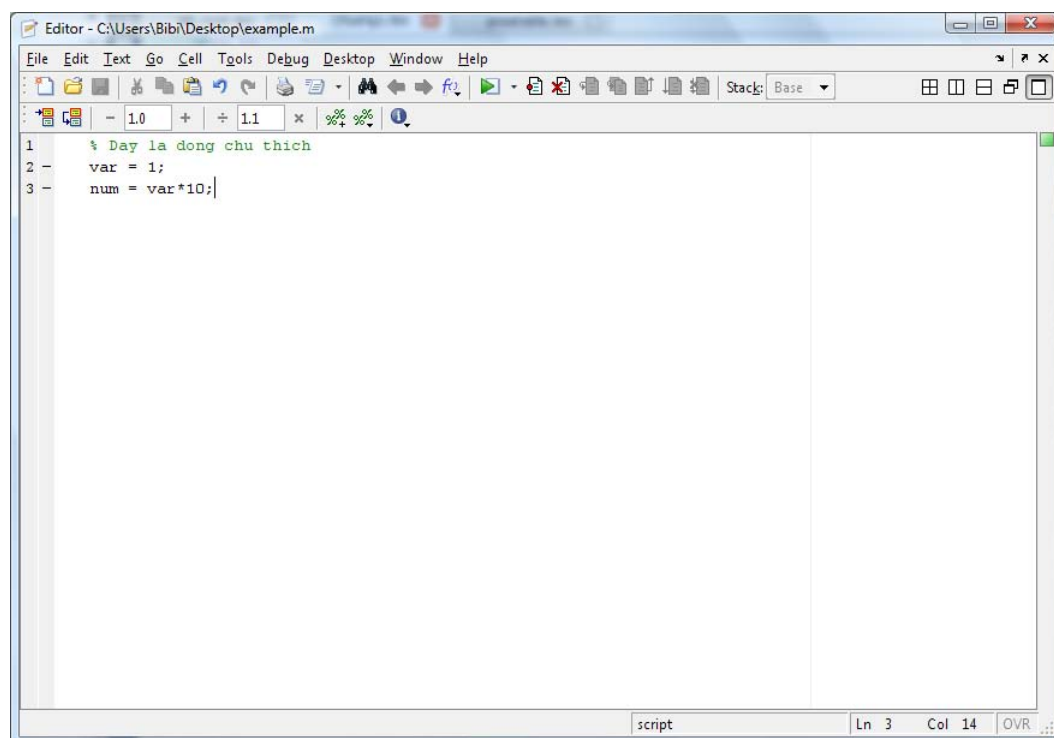
- *Current directory*: thể hiện thư mục đang làm việc. Khi muốn thực thi một tập tin .m nào đó, người dùng phải chắc chắn rằng tập tin .m phải được nhìn thấy trong tab này.
- *Workspace*: chứa danh sách các biến đã được khai báo và sử dụng trong chương trình. Ở tab này, người dùng có thể đọc được tên 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 kỳ 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	$a + b$	a+b
Trừ	$a - b$	a-b
Nhân	$a * b$	a*b
Chia phải	a/b	a/b
Chia trái	b/a	a\b
Lũy thừa	a^b	a ^ b

```
>> 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

- Tính bằng tay các biểu thức sau, rồi thử lại bằng Matlab:
 - $10 / 2 \setminus 5 - 3 + 2 * 4$
 - $3 ^ 2 / 4$
 - $3 ^ 2 ^ 2$
 - $2 + \text{round}(6 / 9 + 3 * 2) / 2 - 3$
 - $2 + \text{floor}(6 / 11) / 2 - 3$
 - $2 + \text{ceil}(-6 / 9) - 3$
 - $\text{fix}(-4/9) + \text{fix}(3*(5/6))$
- Cho $a=36$, $b=15$. Tính bằng tay các biểu thức sau, rồi thử lại bằng Matlab:
 - $\text{mod}(a,b)$
 - $\text{rem}(a,b)$
 - $\text{gcd}(a,b)$
 - $\text{lcm}(a,b)$

Thử lại với các cặp giá trị (a,b) khác.
- Dự đoán kết quả những câu sau, giải thích và kiểm tra lại bằng Matlab
 - $1 \& -1$

- b. $13 \& (-6)$
- c. $0 < -2|0$
- d. $0 \leq 0.2 \leq 0.4$
- 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 nhảy 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

1. Cho $x = [3 \ 1 \ 5 \ 7 \ 9 \ 2 \ 6]$, dự đoán kết quả các dòng lệnh sau và thử lại bằng Matlab
 - a. $x(3)$
 - b. $x(1:7)$
 - c. $x(1:end)$
 - d. $x(1:end-1)$
 - 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: $A(\text{danh sách các dòng}, \text{danh sách các cột})$
- Trích đường chéo chính của ma trận: $\text{diag}(A)$
- 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

- $[x; y']$
- $[x; y]$
- $A(:, [1 \ 4])$
- $A([2 \ 3], [3 \ 1])$
- $A(:)$
- $[A; A(end, :)]$
- $A(1:3, :)$
- $[A; A(1 : 2, :)]$

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 vectơ thì kết quả sẽ là một vectơ 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.	\wedge \cdot^{\wedge} , \cdot^{\prime}
3.	$+$ $-$ (dấu) \sim
4.	$*$ $/$ \backslash \cdot^* $\cdot^/$ \cdot^{\backslash}
5.	$+$ $-$
6.	$:$
7.	$>$ $<$ $>=$ $<=$ $==$ \sim
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 Vectơ và biểu thức logic

Biểu thức logic cho phép truy xuất một cách linh hoạt đến các thành phần của một vectơ 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 vectơ x .
3. `x(x>2 & x <=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 vectơ 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 vectơ 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 vectơ x dương hay không.

`all`: kiểm tra tất cả các phần tử của vectơ 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 vectơ 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 Dk 1 đúng

elseif Biểu thức điều kiện 2

Lệnh thực thi nếu Dk 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 vectơ chạy một vòng **for**:

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

Ví dụ 1.4.5. Tính tích tất cả các phần tử trong vectơ **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 + n2  
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 vectơ x

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

BÀI TẬP

1. Dự đoán kết quả sau và thử lại bằng Matlab

```
if n > 1
    m = n+1
else
    m = n - 1
```

với

a. $n = 7$, $m = ?$ b. $n = 0$, $m = ?$ c. $n = -10$, $m = ?$

2. if $T < 30$

```
    h = 2*T + 1
elseif T < 10
    h = T - 2
else
    h = 0
```

end

a. $T = 50$, $h = ?$ b. $T = 15$, $h = ?$ c. $T = 0$, $h = ?$

3. if $0 < x < 10$

```
    y = 4*x
elseif 10 < x < 40
    y = 10*x
else
    y = 500
```

end

a. $x = -1$, $y = ?$ b. $x = 5$, $y = ?$ c. $x = 30$, $y = ?$

4. Viết Script để tính hàm sau

$h(T) = T - 10$ khi $0 < T < 100$
 $= 0.45T + 900$ khi $T > 100$

Thử các trường hợp a. $T = 5$ b. $T = 110$.

5. Cho vectơ $x = [1 \ 8 \ 3 \ 9 \ 0 \ 1]$, viết các dòng lệnh để

- Tính tổng các phần tử. (So sánh kết quả với lệnh **sum**)
- Viết hàm tính tổng dãy con từ 1 đến j . (So sánh kết quả với lệnh **cumsum**)

6. Viết hàm tính tổ hợp n chập r theo công thức sau

$$C_n^r = \frac{n!}{r!(n-r)!}$$

7. Tạo ma trận $M \times N$ các số ngẫu nhiên (dùng lệnh `rand`). Chuyển các giá trị nhỏ hơn 0.2 thành 0, các giá trị lớn hơn hay bằng 0.2 thành 1.
8. Tạo một vec tơ ngẫu nhiên gồm 1.000.000 phần tử tính tổng bình phương các phần tử sử dụng 2 cách:
- Dùng phép toán trên ma trận.
 - Vòng lặp `for`.

Đo thời gian chạy giữa 2 cách bằng cách sử dụng cặp lệnh `tic ... toc`. Ví dụ, ta muốn đo thời gian tính hàm $x = \sin(t)$ với thì dùng: `tic x = sin(t) toc`.

9. Tạo một vec tơ có giá trị nguyên gồm 100.000 phần tử trong khoảng $[0, 100]$, tìm tất cả những phần tử chia hết cho 3 bằng 2 cách:
- Dùng phép toán trên ma trận.
 - Dùng `for` và `if`.

Đo thời gian chạy giữa 2 cách.

10. Cho $x = [4 \ 1 \ 6]$ và $y = [6 \ 2 \ 7]$, tính các mảng/vector/ma trận sau

- $a_{ij} = x_i y_j$
- $b_{ij} = x_i / y_j$
- $c_i = x_i y_i$, tính tổng các phần tử của c .
- $d_{ij} = x_i / (2 + x_i + y_j)$

11. Cho một vectơ bất kỳ, ví dụ $x = [-4 \ 0 \ 5 \ -3 \ 0 \ 3 \ 7 \ -1 \ 6]$, viết script để đếm xem trong vec tơ có bao nhiêu giá trị âm, bao nhiêu giá trị dương, và bao nhiêu giá trị bằng 0.
12. Viết script để tính trong một vector các số ngẫu nhiên (sử dụng lệnh `rand`):

- a. Có bao nhiêu phần tử đứng trước một phần tử có giá trị nằm trong khoảng 0.8 đến 0.85.
- b. Có bao nhiêu phần tử trong khoảng $[0.01, 0.5]$ đứng trước giá trị trung bình của vector đó.

(Dự đoán kết quả trước khi thực thi lệnh.)

13. Viết một script yêu cầu nhập một nhiệt độ Fahrenheit rồi chuyển sang độ Celcius tương ứng. Script vẫn chạy đến khi không nhập nhiệt độ nữa thì thôi. [Sử dụng hàm isempty.]
14. Giả sử ta đầu tư vào một quỹ tín dụng một số tiền ban đầu là a , số tiền sinh lời hàng năm là 10% số tiền vốn tích lũy. Hãy tính số năm khi mà lợi nhuận thu được gấp đôi tiền vốn ban đầu.

1.5 Vẽ đồ thị

1.5.1 Vẽ đồ thị trong 2-D

Lệnh cơ bản:

`plot(x,f(x))`

Với x : vectơ 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.


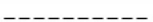


`ylabel('...')`: ghi nhãn cho trục Oy.

`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>	

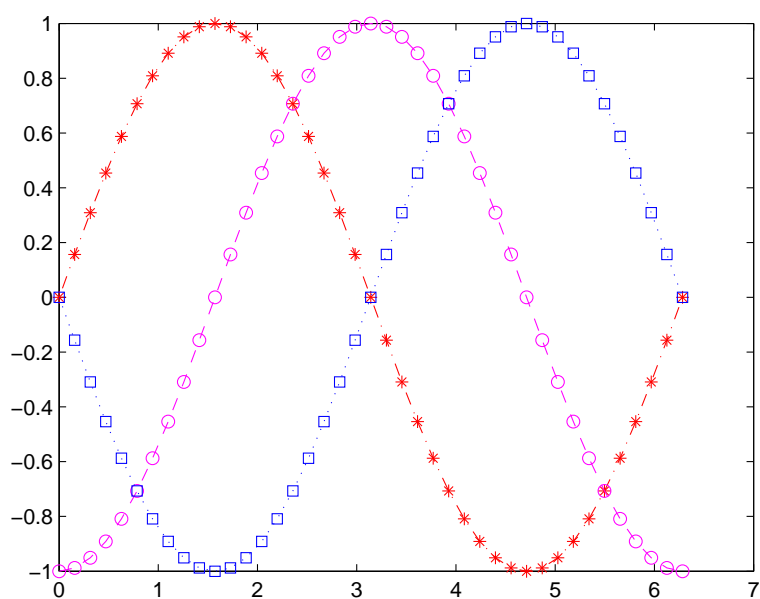
Dấu (marker):

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.

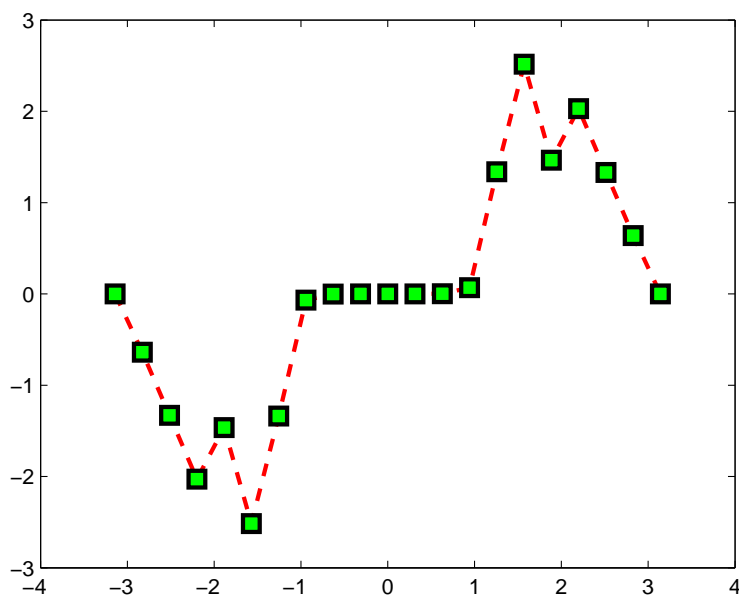
MarkerEdgecolor: màu của đường viền dấu (marker).

MarkerFacecolor: màu bên trong dấu.

Markersize: độ lớn của dấu, tính bằng pt.

Ví dụ 1.5.3.

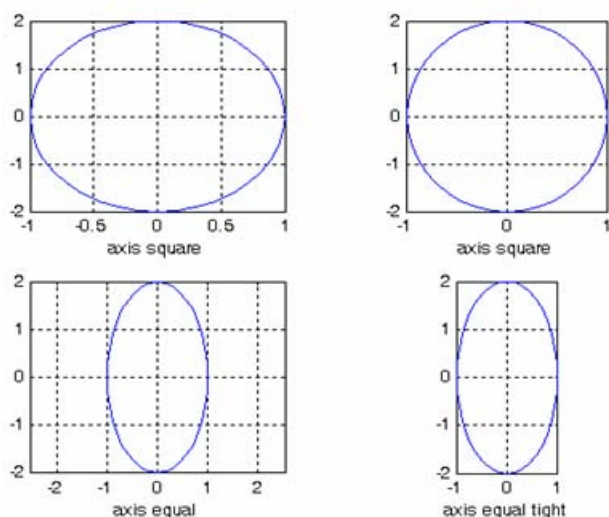
```
x = -pi:pi/10:pi;  
y = tan(sin(x)) - sin(tan(x));  
plot(x,y,'-rs','LineWidth',2,'MarkerEdgecolor','k',...  
      'MarkerFacecolor','g', 'Markersize',10)
```

**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
```

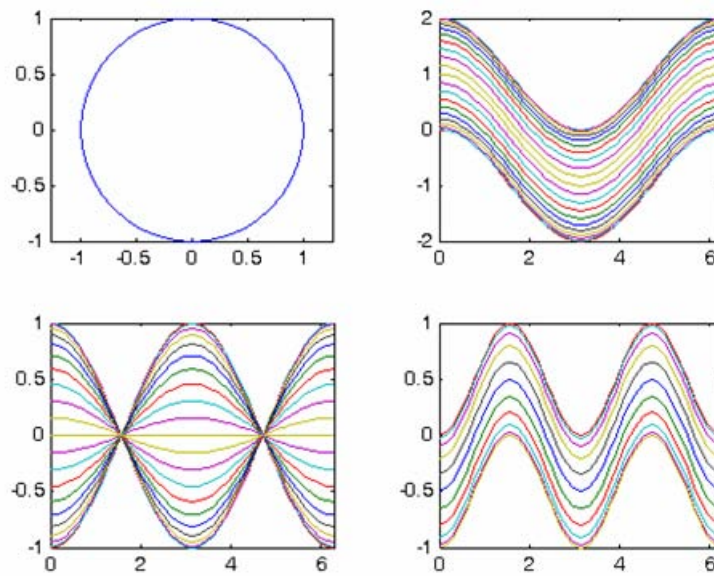
Các kiểu tùy chỉnh hệ trục tọa độ

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])
```

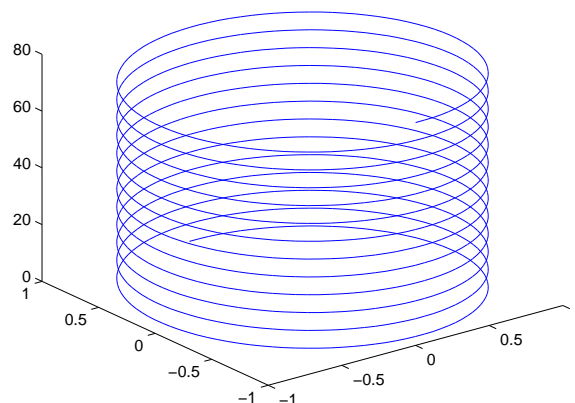


1.5.2 Vẽ đồ thị trong 3-D

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

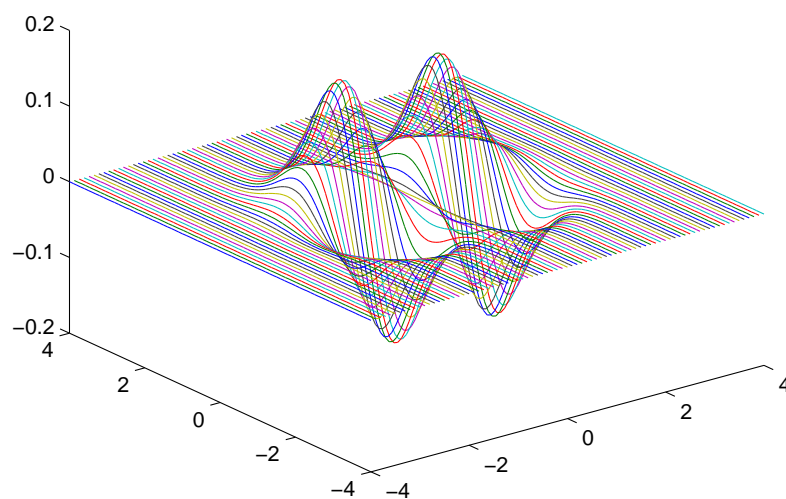
Ví dụ 1.5.5.

```
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)
```



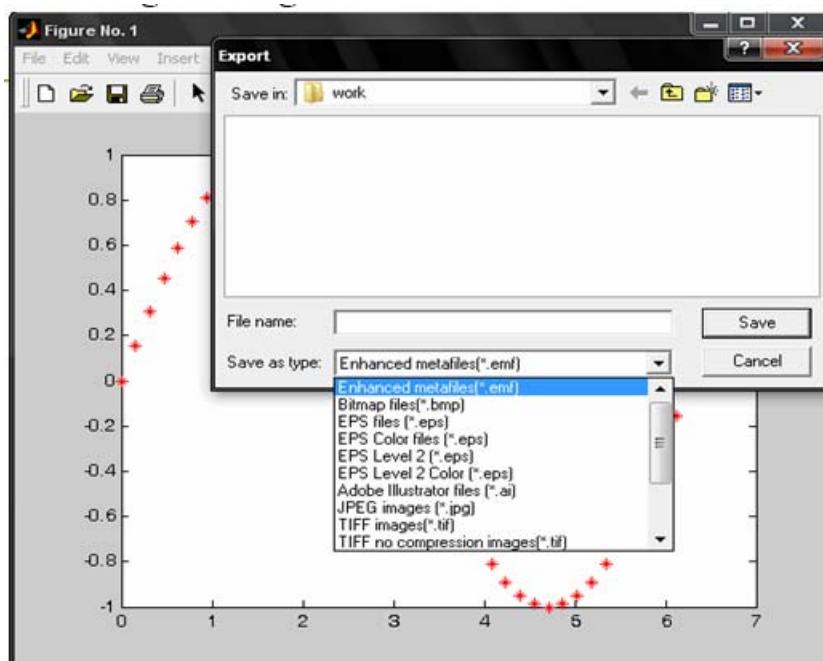
Một số lệnh vẽ đồ thị trong 3-D khác:

- `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

BÀI TẬP

1. Vẽ đồ thị hàm số x , x^3 , e^x và e^{x^2} với $0 < x < 4$.
2. Vẽ đồ thị hàm số $f(x) = \sin(1/x)$ với $0.01 < x < 0.1$.
3. Vẽ lại đồ thị hàm số e^x , ($0 < x < 4$) nhưng với đường đứt nét, màu đỏ và dấu (marker) là dấu $+$.
4. Vẽ lại đồ thị hàm số e^{x^2} ($0 < x < 4$) nhưng với đường gạch chấm, màu xanh và dấu (marker) là hình tam giác hướng lên trên. (Thử lại với các tham số khác (Màu sắc, kiểu đường, dấu))
5. Vẽ đồ thị hàm số $f(x) = \frac{x}{1+x^4}$, $-5 \leq x \leq 5$ với: Kiểu đường là nét gạch chấm, độ rộng 2pt, màu đỏ thẫm.
Dấu (marker) là hình tròn, độ rộng 6pt, màu tô của dấu là xanh, màu đường viền là đen.
6. Vẽ hai hàm $y = x^2$ và $y = \sin(3x)$ trên cùng một đồ thị.
7. Vẽ hai hàm số $f = x \sin(x)$, $0 \leq x \leq 20$ và $g = 50x$, $20 \leq x \leq 25$ trên 2 hệ trục tọa độ khác nhau.
8. Vẽ đường tròn có tâm $O(0,0)$ bán kính bằng 3. Định tọa độ cho trục Ox và Oy là từ -6 đến 6. Tạo đường lưới trên hình vẽ và thử lại với các lệnh axis normal/square/equal/tight.
9. Vẽ đồ thị hàm số $y = \cos(t)$ trong khoảng $0 < t < 2\pi$, chia lại các giá trị trên trục Ox cách đều nhau một khoảng là $\pi/4$. Ghi nhãn cho các giá trị tương ứng là $0, \pi/4, \pi/2, 3\pi/4, \dots, 2\pi$. Chia trục Oy thành 3 khoảng là $-1, 0, 1$.
10. Vẽ đồ thị hàm số với $0 < t < 30$. Ghi chú thích cho trục Ox là 't (thời gian)', Oy là 'Hàm số x(t)' và tựa đề cho đồ thị.
11. Vẽ đồ thị hàm số $x(t) = 2 + 3 \sin(\pi t + 10)e^{-0.35t}$, với $-5 < t < 5$. Định giá trị cho trục Ox là từ -20 đến 20 và Oy từ -6 đến 6. Ghi chú thích cho trục Oy là 'x(t)'; trục Ox là 'thời gian t(giay)'; tựa đề của đồ thị là 'hàm số x(t)'. Dùng lệnh text để viết phương trình của đồ thị trên một điểm của đồ thị.

12. Vẽ đồ thị hàm số x^3 , e^x và e^{x^2} với $0 < x < 4$ trong cùng một cửa sổ hình vẽ.
13. Vẽ đồ thị hàm số $z(x, y) = \frac{\sin \sqrt{x^2 + y^2 + \varepsilon}}{\sqrt{x^2 + y^2 + \varepsilon}}$, $\varepsilon = 1 \times 10^{-10}$ với $-10 \leq x \leq 10$ và $-10 \leq y \leq 10$. Ghi chú thích cho đồ thị.
14. Vẽ đồ thị hàm số sau trên miền $[-2, 2] \times [-2, 2]$, sử dụng các hàm plot3, mesh, meshc, meshz, surf, surfc, waterfall.
 - a. $f(x, y) = 3x - x^3 - 2y^2 + y^4$
 - b. $f(x, y) = \sin(\pi x) + \sin(\pi y) + \sin[(\pi x + \pi y)]$
 - c. $f(x, y) = e^x + y^4 - x^3 + 4 \cos \pi y$
 - d. $f(x, y) = xy e^{(-x^2 - y^2)}$

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. Lũy 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 4×4 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

- Gán cho vector x là dòng thứ nhất của A .
- Gán cho ma trận y là hai dòng còn lại (cuối) của A .
- Tính tổng theo dòng ma trận A .
- Tính tổng theo cột ma trận A .
- Tìm phần tử lớn nhất và phần tử nhỏ nhất của ma trận.
- 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 3x3 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:

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

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

45

```
>> 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
```

BÀI TẬP

1. Tự tạo ngẫu nhiên ma trận A là ma trận vuông 25 phần tử thuộc \mathbb{C} .
 - a. Kiểm tra ma trận A có khả nghịch hay không, nếu không thì cho lại ma trận ngẫu nhiên khác.
 - b. Xác định ma trận nghịch đảo của A bằng các phép biến đổi sơ cấp trên dòng (làm từng bước).
2. Tự tạo ma trận ngẫu nhiên A là ma trận vuông 100 phần tử thuộc \mathbb{C} .
 - a. Kiểm tra ma trận A có khả nghịch hay không, nếu không thì cho lại ma trận ngẫu nhiên khác.
 - b. Xác định ma trận nghịch đảo của A bằng các phép biến đổi sơ cấp trên dòng (không cần làm từng bước).
 - c. Dùng hàm inv hoặc phép toán mũ -1 để xác định ma trận nghịch đảo của A .
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.
4. Cho các cặp ma trận A, B theo thứ tự. Xác định $A^{-1}B^{-1}, (AB)^{-1}, B^{-1}A^{-1}$

a.

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

b.

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

2.2.2 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:

1. $ABA = A$

2. $BAB = B$
3. $(AB)^* = AB$
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 2.2.7. 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 2.2.8. 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 2.2.9. 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
```

BÀI TẬP

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. (*) 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.

2.2.3 Giải phương trình ma trận

Định nghĩa 2.2.10. 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 2.2.11. 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 `mldivide(A,b)`.

Ví dụ 2.2.12. 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 2.2.13. 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`.

Bảng 2.2.1: 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
TRANSA	Ma trận đối xứng liên hợp

BÀI TẬP

- Cho ma trận A như sau $\text{rand}(50,50)$. Tìm ma trận X thỏa :
 - $12X - 2.5A = I_{50}$
 - $XA^{50} = B$. Với B là ma trận ngẫu nhiên khác A.
- 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 $AX=B$.
- Cho ma trận $A, B \in \mathfrak{M}_{100 \times 70}(\mathbb{C})$. Tìm ma trận X thỏa:
 - $100A + 10X = B$
 - $XA = B$

- 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.

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

2.3 Hệ phương trình tuyến tính

2.3.1 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 2.3.1. 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 2.3.2. 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 2.3.3. 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 =
```

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

```
ans =
```

```
3
```

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

2.3.2 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 2.3.4. `>> solve('x^2+2*x+a','x')`

```
ans =
```

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

```
>> 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 2.3.5. `>> S=solve('x^2-3*x+y','y-2+8*x+2*a','x','y')`

```
S =
```

```
x: [2x1 sym]
y: [2x1 sym]
```

```
>> 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 =
```

$$\begin{aligned} & -2*a - 4*(8*a + 113)^{(1/2)} - 42 \\ & 4*(8*a + 113)^{(1/2)} - 2*a - 42 \end{aligned}$$

```
>> 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 =
```

$$\begin{aligned} & 2*(9 - 4*y)^{(1/2)} - y/2 - 5 \\ & - y/2 - 2*(9 - 4*y)^{(1/2)} - 5 \end{aligned}$$

```
>> S.x
```

```
ans =
```

$$\begin{aligned} & 3/2 - (9 - 4*y)^{(1/2)}/2 \\ & (9 - 4*y)^{(1/2)}/2 + 3/2 \end{aligned}$$

BÀI TẬP

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** 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. Cho hệ phương trình sau:

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

- Giải hệ phương trình trên.
- 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.
- Cho biết ý nghĩa của hàm **rcond**.

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

2.4.1 Định thức

Mệnh đề 2.4.1. 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 2.4.2. Tính định thức của ma trận sau:

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

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

```
ans =
```

```
-316
```

2.4.2 Giải hệ phương trình tuyến tính bằng định thức

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

Thuật toán 2.4.3. 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.

2.5 Đa thức đặc trưng, trị riêng và vectơ riêng

Định nghĩa 2.5.1. 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 2.5.2. *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 2.5.3. 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).

BÀI TẬP

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 .
2. 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.

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

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

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 3.1.1. tập hợp $A = [1 \ 4 \ 8 \ 9 \ 10]$ có thể khai báo như sau :

- vector dòng

```
> >A = [1, 4, 8, 9, 10]
```

```
A=
```

```
1 4 8 9 10
```

- hoặc vector cột

```
> > A = [1; 4; 8; 9; 10]
A =
     1
     4
     8
     9
    10
```

- hoặc đó là tập rỗng

```
> > A = []
A =
[]
```

- Để gọi các phần tử trong tập A

```
> > A(1)
ans =
     1
> > A(2)
ans =
     4
> > A(5)
ans =
    10
```

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 một đối số, giá trị trả về sẽ là **1** nếu A là tập rỗng, ngược lại thì giá trị trả về là **0**.

```
> > isempty(A)
ans=
    0
> > B= []
B =
    []
> > isempty(B)
ans=
    1
```

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

Hàm $length(A)$ 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.

```
> > length(A)
ans=
    5
```

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

Hàm $ismember(s, A)$ 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 $ismember(s, A)$ sẽ là **1** nếu s thuộc A , ngược lại là **0**.

```
> > ismember(2,A)
ans=
    0
> > ismember(8,A)
ans=
    1
```

Ngoài ra, đối số s ở trên có thể là một tập hợp (vector). Giá trị trả về của $ismember(s, A)$ 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).

```
> > s = [0, 3, 9]
s =
    0    3    9
> > ismember(s,A)
ans =
    0    0    1
```

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

Hàm *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.

```
> > C = nchoosek(A,4)
C =
    1    4    8    9
    1    4    8   10
    1    4    9   10
    4    8    9   10
```

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 *unique*(A). Nếu các phần tử A không trùng nhau, thì *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.

```
> > unique(A)
ans =    1    4    8    9   10
> > D = [1, 9, 4, 8, 10, 1, 8, 4, 9, 10]
D =
    1    9    4    8   10    1    8    4    9   10
> > unique(D)
    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 $\text{union}(\mathbf{A}, \mathbf{B})$ dùng để tìm hội của hai tập hợp \mathbf{A} và \mathbf{B} . Kết quả của $\text{union}(\mathbf{A}, \mathbf{B})$ là một tập hợp trong đó liệt kê các phần tử trong \mathbf{A} và \mathbf{B} mà không có sự liệt kê lặp lại. Ngoài ra kết quả trả về sẽ được sắp theo thứ tự tăng dần.

```
> >E = [2, 3, 5]
E=
    2  3  5
> > union(A,E)
ans=
    1  2  3  4  5  8  9  10
> >F = [4, 2, 5, 3, 2, 5]
F=
    4  2  5  3  2  5
> > union(A,F)
ans=
    1  2  3  4  5  8  9  10
```

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

```
> >G = [6, 7]
G=
    6  7
> > union(union(A,E),G)
ans=
    1  2  3  4  5  6  7  8  9  10
```

7. Giao giữa hai tập hợp (*intersect*)

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

```
> > intersect(A,F)
ans=
     4
> > intersect(A,E)
ans=
     []
```

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

```
> > intersect(intersect(A,F),E)
ans=
     []
```

8. Hàm hiệu giữa hai tập hợp (*setdiff*)

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.

```
> > H = [10, 5, 8, 9]
F=
     10     5     8     9
> > setdiff(A,H)
ans=
     1     4
> > K = []
K=
     []
> > setdiff(K,H)
ans=
     []
```

9. Hàm tính hội hiệu giao (*setxor*)

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 tăng dần.


```
> > setxor(A,H)
ans=
    1    4    5
```

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?
 - c. Nếu A và B là 2 tập khác trống thì tìm $A \cup B$ và $A \cap B$.
 - d. 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 $D = 3 \cdot \text{rand}(1,9)$
- Kiểm tra xem tập hợp đó có thuộc tập số nguyên hay không?
 - Liệt kê tất cả các tập con gồm 4 phần tử của tập D .
 - 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 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:
- $A \subset B$ hay $B \subset A$ hay $A = B$ không?
 - $A \setminus B; B \setminus A; A \cup B; A \cap B$.
 - $A^2; B^2; A \times B; B \times A$.
 - 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
- $(A \cup B) \cap C$.
 - $(A \cup B \cup C) \setminus (A \cap B \cap C)$.
 - $A \cap C \cup B$.
10. Cho tập A, B và C như trên.
- 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

3.2.1 Khái niệm chung

Symbolic Math Toolboxes kết hợp tính toán bằng chữ vào môi trường MATLAB. Các toolbox này bổ sung các tiện ích số và đồ thị với các kiểu tính toán toán học khác nhau.

Tiện ích	Nội dung
Calculus	đạo hàm, tích phân, giới hạn, tổng và chuỗi Taylor
Linear Algebra	nghịch đảo, định thức, giá trị riêng, phân tích và dạng chính tắc của ma trận
Simplification	phương pháp rút gọn các biểu thức đại số
Solution of Equations	giải bằng chữ và bằng số các phương trình đại số và vi phân
Variable Precision Arithmetic	đánh giá độ chính xác của các biểu thức đại số
Transform	biến đổi Laplace, Fourier
Special Mathematical Function	các hàm toán học đặc biệt của các ứng dụng toán học kinh điển

3.2.2 Khởi động TOOLBOX

1. Các đối tượng chữ:

Trong phần này chúng ta sẽ xem xét cách tạo và dùng các đối tượng chữ. Chúng ta cũng sẽ xem xét các biến chữ mặc định. **Symbolic Math Toolbox** định nghĩa một kiểu dữ liệu MATLAB mới gọi là đối tượng chữ hay sym. Bên trong, một đối tượng chữ là một cấu trúc số liệu mà nó lưu biểu diễn chuỗi các ký tự. **Symbolic Math Toolbox**

dùng các đối tượng chữ để biểu diễn các biến chữ, các biểu thức chữ, các ma trận chữ.

2. **Tạo các biến và các biểu thức chữ:** Lệnh *sym* cho phép ta xây dựng các biến và các biểu thức chữ.

```
>> x = sym('x')
>> a = sym('alpha')
```

tạo ra các biến chữ là x và a với x là x và a là *alpha*.

Giả sử ta muốn ta muốn dùng biến chữ để biểu diễn tỉ lệ vàng $\rho = \frac{1 + \sqrt{5}}{2}$ Ta dùng lệnh:

```
>> rho = sym('(1 + sqrt(5))/2')
```

Bây giờ ta có thể thực hiện các phép toán khác nhau với rho. Ví dụ:

```
>> f = rho^2 - rho -1
f=
(1/2+1/2*5^(1/2))^2-3/2-1/2*5^(1/2)
```

Ta rút gọn biểu thức:

```
>> simplify(f)
ans =
0
```

Bây giờ giả sử ta muốn giải phương trình bậc hai $f = ax^2 + bx + c$.
Phát biểu:

```
>> f = sym('a*x^2 + b*x + c')
```

gán biểu thức chữ $ax^2 + bx + c$ cho biến f . Tuy nhiên trong trường hợp này **Symbolic Math Toolbox** không tạo ra các biến tương ứng với các số hạng a, b, c và x trong biểu thức. Để thực hiện các phép toán bằng chữ (ví dụ tích phân, đạo hàm, thay thế v.v) trên f ta phải tạo các biến một cách rõ ràng, nghĩa là cần viết:

```
>> a = sym('a')
>> b = sym(b)
>> c = sym('c')
>> x = sym('x')
```

hay đơn giản là:

```
>> syms a b c x
```

Nói chung là ta có thể dùng *sym* hay *syms* để tạo các biến chữ nhưng nên dùng *syms* để tiết kiệm thời gian.

3. Biến đổi giữa số và chữ

- (a) **Tạo các biến thực và phức:** Lệnh *sym* cho phép ta mô tả các thuộc tính toán học của các biến chữ bằng cách dùng tùy chọn *real*. Phát biểu:

```
>> x = sym('x','real')
>> y = sym('y','real')
```

hay hiệu quả hơn:

```
>> syms x y real
>> z = x + i*y
```

tạo ra biến chữ x và y có thuộc tính là số thực. Đặc biệt:

```
>> f = x^2 + y^2
```

thực sự là số không âm. Như vậy z là biến phức và các lệnh:

```
>> conj(x)
ans=
    x
>> conj(z)
ans=
    x - i*y
>> expand(z*conj(z))
ans=
    x^2 + y^2
```

Lệnh *conj* là toán tử tạo số phức liên hợp. Để xóa thuộc tính *real* của x ta dùng lệnh:

```
>> syms x unreal
```

hay:

```
>> x = sym('x','unreal')
```

Lệnh *clear x* không xoá thuộc tính số *real* của x .

- (b) **Tạo các hàm trừu tượng:** Nếu ta muốn tạo một hàm trừu tượng (nghĩa là một hàm không xác định) $f(x)$ cần dùng lệnh:

```
>> f = sym('f(x)')
```

Khi này f hoạt động như là $f(x)$ và có thể xử lí bằng các lệnh **toolbox**. Ví dụ để tính vi phân bậc 1 ta viết:

```
>> df = (subs(f,'x','x+h') - f)/'h'
```

hay

```
>> syms x h
>> df = (subs(f,x,x+h)-f)/h
df =
    (f(x+h)-1f(x))/h
```

ứng dụng này của hàm *sym* sẽ rất hữu ích trong biến đổi **Fourier**, **Laplace**.

- (c) **Dùng *sym* để truy cập các hàm của Maple:** Ta có thể truy cập hàm giai thừa $k!$ của Maple khi dùng *sym*.

```
>> kfac = sym('k!')
```

Để tính $6!$ hay $k!$ ta viết:

```
>> syms k n
>> subs(kfac,k,6)
ans =
    720
>> subs(kfac,k,n)
ans =
    n!
```

- (d) **Ví dụ tạo ma trận chữ:** Một ma trận vòng là ma trận mà hàng sau có được bằng cách dịch các phần tử của hàng trước đi 1 lần. Ta tạo một ma trận vòng A bằng các phần tử a, b và c :

```
>> syms a b c
>> A = [a b c; b c a; c a b]
A=
     a     b     c
     b     c     a
     c     a     b
```

4. Tạo các hàm toán học bằng chữ:

- (a) **Dùng các biểu thức chữ:**

```
>> syms x y z
>> r = sqrt(x^2 + y^2 + z^2)
>> t = atan(y/x)
>> f = sin(x*y)/(x*y)
```

tạo ra các biểu thức chữ r, t và f . Ta có thể dùng các lệnh *diff*, *int*, *subs* hay các lệnh **Symbolic Math Toolbox** khác để xử lí các biểu thức như vậy.

- (b) **Tạo các M-file:** M-file cho phép ta dùng các hàm tổng quát hơn.

Chẳng hạn, ta muốn tạo ra hàm $\sin c = \frac{\sin(x)}{x}$ ta sẽ viết một M-file có nội dung như sau:

```
function z = sinc(x)
    if isequal(x, sym(0))
        z = 1;
    else
        z = sin(x)/x;
    end
```

Ta có thể mở rộng các ví dụ như vậy cho các hàm và biến khác nhau.

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
$\tan(x)$		$\log_{10}(x)$	logarit cơ số 10
$\arccos(x)$	$\cos^{-1}(x)$	$\text{abs}(x)$	trị tuyệt đối
$\arcsin(x)$	$\sin^{-1}(x)$	$\text{sqrt}(x)$	căn bậc hai
$\arctan(x)$	$-\pi/2 \leq \tan^{-1} \leq \pi/2$	$\text{real}(x)$	phần thực
$\text{atan2}(x)$	$-\pi \leq \tan^{-1} \leq \pi$	$\text{imag}(x)$	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
$\text{acosh}(x)$	$\cosh^{-1}(x)$	$\text{sign}(x)$	1(dương)/0/-1(âm)
$\text{asinh}(x)$	$\sinh^{-1}(x)$	$\text{mod}(y,x)$	phần dư của y/x
$\text{atanh}(x)$	$\tanh^{-1}(x)$	$\text{rem}(y,x)$	phần dư của y/x
\max	giá trị lớn nhất và chỉ số của nó	$\text{eval}(f)$	giá trị một biểu thức
\min	giá trị bé nhất và chỉ số của nó	$\text{feval}(f,a)$	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	nghiệm đa thức	poly	giá trị của hàm đa thức
i, j	$\sqrt{-1}$	π	π
break	thoát vòng lặp while/for	Inf, inf	∞
nargin	số đối số vào	nargout	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 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 $\{x_n\}_{n=1}^{\infty}$

Chẳng hạn 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 $s_n = \sum_{i=1}^n x_i$.

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 :

- $\text{limit}(x_n, n, a)$ dùng để tính giới hạn của dãy x_n khi n dần về giá trị a .

- $\text{limit}(x_n, n, a, \text{right})$; $\text{limit}(x_n, n, a, \text{left})$ dùng tính giới hạn một bên khi x tiến về a từ 2 phía.

```
>> syms n
>> xn=1/n
>> limit(xn,n,inf)
ans=
    0
>> limit(xn,n,5)
ans=
    0.2
```

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 :

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

```
>> N=1000
>> for i=1:N
        X(i)=1/i
    end
>> plot(X)
```

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. $\frac{1.2.3...(2n-1)}{(2n)^n}$.

g. $\frac{1.2.3...(2n-1)}{n!}$.

2. Tính 20 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. $a_n = \{\frac{n}{n+1}\}_{n=1}^{\infty}$.
- c. $a_n = \{\cos \frac{n\pi}{6}\}_{n=0}^{\infty}$.
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. 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.
9. Chỉ ra một giá trị r để dãy nr^n hội tụ.

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

3.4.1 Giới hạn của hàm số

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

- $\text{limit}(\mathbf{f}, \mathbf{x}, \mathbf{a})$: Tính giới hạn của hàm số \mathbf{f} khi \mathbf{x} tiến về \mathbf{a} .
- $\text{limit}(\mathbf{f}, \mathbf{x}, \mathbf{a}, 'right')$ hoặc $\text{limit}(\mathbf{f}, \mathbf{x}, \mathbf{a}, 'left')$: Tính giới hạn trái hoặc giới hạn phải của hàm số khi \mathbf{x} tiến về \mathbf{a} .

```
>> syms x
>> limit(sin(x)/x,x,0)
ans =
    1
>> limit(1/x,x,0)
ans =
    NaN
>> limit((1 + x/n)^n,n,inf)
ans =
    exp(x)
```

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

```
>> 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 3.4.1. 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.

Cho hàm số $g(x) = \cos(x)$ và $f(x) = \arctan(x)$, tìm $g'(a)$ và $f'(a)$ với $a \in \mathbb{R}$?

```
>> syms a h
>> dg = limit( (cos(a+h) - cos(a))/h, h, 0 )
dg =
    -sin(x)
>> df = limit((atan(a+h)-atan(a))/h,h,0)

df =

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

3.4.2 Sự liên tục của hàm số

Định nghĩa 3.4.2. 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 3.4.3. 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 3.4.4.

1. Tìm hiểu lệnh *subs* trong trường hợp có nhiều biến.
2. Áp dụng lệnh *limit* để tính giới hạn hàm số trong trường hợp hàm nhiều biến.

BÀI TẬP

1. Khảo sát tính liên tục của hàm số tại a . Vẽ đồ thị hàm số.

a. $f(x) = \ln |x - 2| \quad a = 2$

b. $f(x) = \begin{cases} \frac{1}{x-1} & x \neq 1 \\ 2 & x = 1 \end{cases} \quad a = 1$

c. $f(x) = \begin{cases} e^x & x < 0 \\ x^2 & x \geq 0 \end{cases} \quad a = 0$

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

e. $f(x) = \begin{cases} \cos(x) & x < 1 \\ 0 & x = 1 \\ 1 - x^2 & x > 1 \end{cases} \quad a = 0$

f. $f(x) = \begin{cases} \frac{2x^2 - 3x - 3}{x - 3} & x \neq 3 \\ 0 & x = 3 \end{cases} \quad a = 3$

2. Vẽ đồ thị hàm số và xác định các điểm bất liên tục của các hàm số.

a. $y = \frac{1}{1 + e^{1/x}}$

b. $y = \ln(\tan^2 x)$

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

a. $f(x) = \begin{cases} x^2 & x < 1 \\ \sqrt{x} & x \geq 1 \end{cases}$

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

4. Xác định $f'(0)$ có tồn tại hay không?

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

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

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

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

3.5 Các bài toán tích phân hàm một biến

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

3.5.1 Tích phân bất định

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

Ví dụ 3.5.1. 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*.

3.5.2 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 3.5.2. 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 =

$$1/1024*\pi^4*\operatorname{atan}(1/4*\pi)-1/768*\pi^3+1/16*\pi-1/4*\operatorname{atan}(1/4*\pi)$$

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

3.5.3 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 3.5.3. 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

1. 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*.

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

2. Hàm tích phân sine

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

có vai trò quan trọng trong kỹ thuật điện.

- a. Vẽ đồ thị của Si.
 - b. Tìm những điểm mà tại đó hàm này đạt cực đại địa phương.
 - c. Tìm tọa độ của điểm uốn đầu tiên phía bên phải gốc tọa độ.
 - d. Hàm số có tiệm cận ngang hay không?
3. 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.
- a. $y = x + x^2 - x^4$.
 - b. $y = 2x + 3x^4 - 2x^6$.

4. 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?
5. 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.

3.5.4 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)*.

```
>> syms a b c x
>> f = a*x^3 + x^2 - b*x - c
f =
      a*x^3 + x^2 - b*x - c
>> diff(f)
ans =
      3*a*x^2 + 2*x - b
>> diff(f,a)
ans =
      x^3
>> diff(f,2)
ans=
      6*a*x + 2
>> diff(f,a,2)
ans=
      0
```

Hàm *diff* có thể dùng đối số là ma trận. Trong trường hợp này đạo hàm được thực hiện trên từng phần tử.

```
>> syms a x
```

```
>> A = [cos(a*x), sin(a*x); sin(a*x), cos(a*x)]
A =
    cos(a*x)    sin(a*x)
    sin(a*x)    cos(a*x)
>> dy = diff(A)
dy =
    sin(a*x)*a    cos(a*x)*a
    cos(a*x)*a    sin(a*x)*a
```

Ta khảo sát biến đổi từ toạ độ Euclide (x, y, z) sang toạ độ cầu (r, λ, ϕ) thực hiện bằng các công thức:

$$\begin{aligned}x &= r \cos \lambda \cos \phi \\y &= r \cos \lambda \sin \phi \\z &= r \sin \lambda\end{aligned}$$

Để tính ma trận **Jacobi** J của phép biến đổi này ta dùng hàm *jacobian*. Định nghĩa toán học của J là:

$$J(x, y, z) = \frac{\partial(x, y, z)}{\partial(r, \lambda, \phi)}$$

Để dễ viết ta dùng kí tự l thay cho λ và f thay cho ϕ . Các lệnh

```
>> syms r l f
>> x = r*cos(l)*cos(f);
>> y = r*cos(l)*sin(f);
>> z = r*sin(l);
>> J = jacobian([x; y; z], [r l f])
J =
    cos(l)*cos(f)    -r*sin(l)*cos(f)    -r*cos(l)*sin(f)
    cos(l)*sin(f)    -r*sin(l)*sin(f)    r*cos(l)*cos(f)
    sin(l)           r*cos(l)           0
>> detJ = simple(det(J))
detJ =
    -cos(l)*r^2
```

Chú ý là đối số thứ nhất của hàm *jacobian* phải là **vec tơ cột** và đối số thứ hai là **vec tơ hàng**. Hơn nữa do định thức của ma trận **Jacobian** là biểu thức lượng giác khá phức tạp nên ta dùng lệnh *simple* để thay thế và rút gọn.

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 .

Đôi khi MATLAB trả lại một biểu thức đặc trưng quá khó để có thể đọc. Một số công cụ có sẵn trợ giúp làm cho biểu thức dễ đọc hơn. Trước tiên đó là hàm *pretty*. Lệnh này hiển thị biểu thức đặc trưng theo một khuôn mẫu tương tự như kiểu toán học.

```
>> x = sym('x');
>> f = taylor(log(x+1)/(x-5))
f =
-1/5*x+3/50*x^2-41/750*x^3+293/7500*x^4-1207/37500*x^5
>> pretty(f)
ans=
          2    41    3    293    4    1207    5
-1/5 x + 3/50 x - --- x + ---- x - ---- x
          750          7500          37500
```

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. Cố 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. $\frac{x}{4+x^2}$

CHƯƠNG 4

Cơ học và Xác suất - Thống kê

4.1 Cơ học

4.1.1 Động học chất điểm

Động học chất điểm: là môn học nghiên cứu chuyển động của vật thể mà kích thước của nó quá nhỏ so với quỹ đạo. Các tác nhân như khối lượng, lực, môment không được đề cập đến trong môn học này.

Động học chất điểm quan tâm đến các đại lượng: (1) tọa độ (vị trí) $\mathbf{x} = (x_1, \dots, x_n)$, (2) vận tốc $\mathbf{v} = (v_1, \dots, v_n)$ và (3) gia tốc $\mathbf{a} = (a_1, \dots, a_n)$ với n là chiều không gian đang xét. Quan hệ giữa các đại lượng này như sau

$$\mathbf{v} = \frac{d\mathbf{x}}{dt} = \dot{\mathbf{x}}, \quad \mathbf{a} = \frac{d\mathbf{v}}{dt} = \ddot{\mathbf{x}} \quad (4.1)$$

Phương trình chuyển động: Trong không gian 1 chiều, gọi t_0 là thời điểm bắt đầu khảo sát chuyển động, x_0, v_0, a_0 lần lượt là vị trí, vận tốc và gia tốc tại thời điểm đó. Từ phương trình (4.1) ta tìm được

$$x(t) = \int_{t_0}^t v(\tau) d\tau + x_0, \quad v(t) = \int_{t_0}^t a(\tau) d\tau + v_0 \quad (4.2)$$

Trong học trình này, các thành phần của vectơ chỉ phụ thuộc vào thời gian mà không phụ thuộc vào các thành phần khác của vectơ đó. Vì vậy trong

không gian có số chiều lớn hơn 1, chúng ta cũng thu được các phương trình chuyển động tương đương với trong không gian 1 chiều.

$$x_i(t) = \int_{t_0}^t v_i(\tau) d\tau + x_i(t_0), \quad v_i(t) = \int_{t_0}^t a_i(\tau) d\tau + v_i(t_0) \quad (4.3)$$

Ví dụ 4.1.1. Trong không gian 1 chiều, giả sử gia tốc của vật thể là hằng số theo thời gian $a = k$. Dựa theo phương trình (4.2), tìm biểu diễn của vận tốc và tọa độ của vật thể theo thời gian và các giá trị đầu

Tìm vận tốc tức thời	» syms x0 v0 k t0 t tt
Đổi biến từ t sang tt	» v = int(k,tt,t0,t)+v0
Tìm tọa độ tại thời điểm t	» v = subs(v,t,tt);
	» x = int(v,tt,t0,t) + x0

Ví dụ 4.1.2. Trong không gian 2 chiều, bài toán ném xiên được phát biểu như sau: một vật được ném lên trời với vận tốc v_0 theo phương hợp với mặt đất góc α . vật chịu gia tốc trọng trường theo phương thẳng đứng $a_y = g = -10m/s^2$. Dựa theo phương trình (4.3), tìm biểu diễn của vận tốc và tọa độ của vật thể theo thời gian và các giá trị đầu.

Các vận tốc đầu	» syms x0 y0 v0 g al t0 t tt
Vận tốc tức thời v_x	» v0x=v0*cos(al); v0y=v0*sin(al)
Tọa độ x theo thời gian	» vx = v0x
Vận tốc tức thời v_y	» x = int(vx,tt,t0,t) +x0
Đổi biến từ t sang tt	» vy = int(g,tt,t0,t) +v0y
Tọa độ y theo thời gian	» vy = subs(vy,t,tt);
	» y = int(vy,tt,t0,t) +y0

Phương trình giao điểm: Trong quá trình chuyển động, khi 2 vật A và B gặp nhau, ta thu được hệ phương trình sau

$$x_i^A(t) = x_i^B(t) \forall i \quad (4.4)$$

Giải hệ phương trình trên, ta xác định được thời điểm 2 vật gặp nhau, sau đó tìm lại các giá trị tọa độ, vận tốc.

Ví dụ 4.1.3. Xe 1 chạy từ thành phố A đến thành phố B với vận tốc đều 30km/h . Cùng lúc với xe 1, xe 2 chạy từ B đến A với vận tốc đầu là 20km/h , gia tốc 10km/h^2 . Quãng đường AB dài 280 cây số, hỏi khi nào 2 xe gặp nhau? Vị trí gặp nhau cách A bao xa?

Các giá trị cho trước	» syms t
PTCĐ của xe 1	» v1=30; v20=20; a2=10;sAB=280;
PTCĐ của xe 2	» x1 = v1*t
Thời điểm 2 xe gặp nhau	» x2 = -a2*t^2/2-v20*t+sAB
Chọn giá trị hợp lý	» ts = solve('30*t=- 5*t^2 - 20*t + 280',t))
Khoảng cách từ đó đến A	» ts = 4
	» x1 = v1*ts

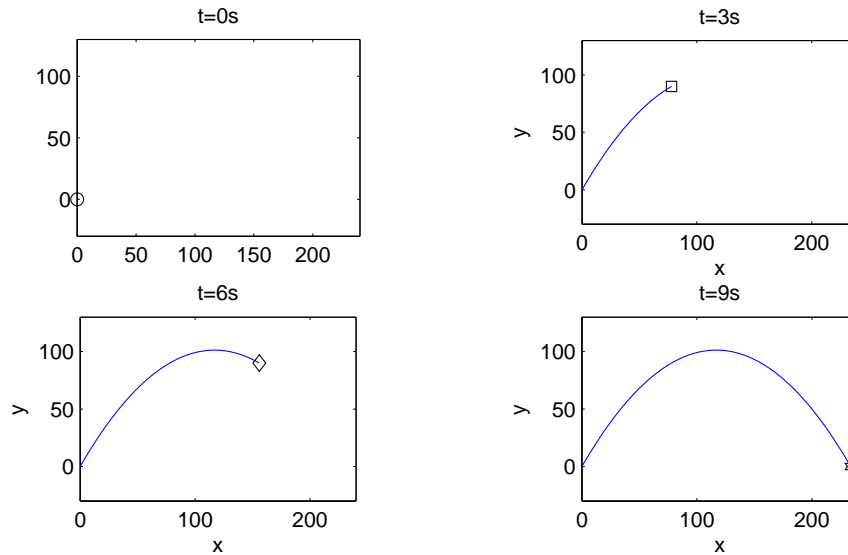
Vận tốc trung bình: Đôi khi, để giảm bớt sự phức tạp, người ta không quan tâm đến vận tốc tức thời mà chỉ chú ý đến vận tốc trung bình. Khi chất điểm M di chuyển trong khoảng thời gian từ t_1 đến t_2 , thì vận tốc trung bình của nó là

$$v_{tb} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} v(\tau) d\tau \quad (4.5)$$

với $v(\tau)$ là vận tốc tức thời được tính bởi công thức $\left(\sum_i^n v_i^2\right)^{\frac{1}{2}}$.

Ví dụ 4.1.4. Trong bài toán ném nghiêng, vật được ném với vận tốc $30\sqrt{3}\text{m/s}$ theo phương hợp với mặt đất 60° . Tính vận tốc trung bình của vật trong mỗi nửa giây từ lúc được ném cho đến khi chạm đất. Vẽ đồ thị vận tốc trung bình theo thời gian.

Các giá trị cho trước	» syms t tt
Các vận tốc đầu	» v0=30*sqrt(3); al=pi/3; g=-10;
Vận tốc tức thời v_x	» v0x=v0*cos(al), v0y=v0*sin(al)
Vận tốc tức thời v_y	» vx = v0x
Vận tốc tức thời v	» vy = int(g,tt,0,t)+v0y;
Đổi biến từ t sang tt	» v = sqrt(vx^2 + vy^2)
Tọa độ y theo thời gian	» vy = subs(vy,t,tt)
Tìm thời điểm y bằng 0	» y = int(vy,tt,0,t)
Chọn giá trị hợp lý	» [ts]=solve('(-5)*t*(t - 9)=0')
Tính nguyên hàm của v	» ts = 9;
Chia khoảng thời gian	» S = int(v,t)
Tính vận tốc trung bình	» T1=[0:0.5:ts-0.5], T2=[0.5:0.5:ts]
Vẽ đồ thị vận tốc tb	» Vtb=(subs(S,t,T2)-subs(S,t,T1))/0.5
	» stairs([T1 ts],[Vtb Vtb(end)])



Hình 4.1.1: Mô phỏng vật ném xiên

BÀI TẬP

- Một chất điểm P di chuyển dọc theo đường thẳng theo phương trình $x = 4t^3 + 2t + 5$.
 - Xác định tọa độ, vận tốc và gia tốc khi $t = 3$.
 - Tìm vận tốc trung bình và gia tốc trung bình của P trong khoảng thời gian từ giây thứ 3 đến giây thứ 4.
- Một chất điểm di chuyển dọc theo đường thẳng theo phương trình $x = 8t^2 + 2t$. Vẽ các đồ thị tọa độ, vận tốc và gia tốc theo thời gian.
- Thành phần x và y của tọa độ chất điểm M được cho bởi phương trình $x = 10t^2 + 2t$, $y = t^3 + 5$.
 - Vẽ đồ thị thể hiện quỹ đạo của M.
 - Xác định vận tốc và gia tốc của M tại thời điểm $t = 2$.
- Tại thời điểm đầu, chất điểm N có tọa độ $x = 5$, $y = -15$. Thành phần vận tốc của nó được cho bởi phương trình $v_x = 20t + 5$, $v_y = t^2 - 20$. Xác định tọa độ, vận tốc và gia tốc vật thể tại $t = 2$.
- Một chất điểm di chuyển dọc theo đường thẳng với gia tốc $a = 2\sqrt{v}$. tại $t = 2$ tọa độ nó là $x = 64/3$ và vận tốc là $v = 16$. Xác định tọa độ, vận tốc và gia tốc vật thể tại $t = 3$.
- Xe A xuất phát tại thành phố C với vận tốc đều $v^A = 60km/h$. Xe B cũng xuất phát từ thành phố C với vận tốc đều $v_0^B = 20km/h$ và gia tốc đều $a^B = 10km/h^2$.
 - Biểu diễn quan hệ vị trí - thời gian của 2 xe trên cùng 1 đồ thị trong từng trường hợp 2 xe chạy cùng chiều, ngược chiều.
 - Tìm phương trình biểu diễn khoảng cách của 2 xe trong trường hợp 2 xe chạy cùng chiều, ngược chiều.
 - 2 xe chạy cùng chiều. Xác định thời điểm 2 xe gặp nhau. Khoảng cách từ điểm gặp nhau đến thành phố C là bao xa?
 - 2 xe chạy ngược chiều. Khi hai xe có cùng độ lớn vận tốc thì khoảng cách 2 xe là bao nhiêu.

7. Giải lại bài tập 6 trong trường hợp xe B xuất phát trễ hơn xe A 1 giờ đồng hồ.
8. Giải lại bài tập 6 trong trường hợp xe B xuất phát trễ hơn xe A 1 giờ đồng hồ và xuất phát tại vị trí cách thành phố C 20km ngược hướng chuyển động của xe A.
9. Tại tọa độ $x = 0m$, độ cao $10m$, lúc $t = 0s$ có 1 con chim bay với vận tốc đều $v_c = 2m/s$. Một người thợ săn đứng tại tọa độ $x = -2m$ nhắm bắn con chim. Tên bay ra lúc $t = 1s$ ở độ cao $h = 1.5m$ với vận tốc $v_0 = 25m/s$ hợp với phương nằm ngang 45° .
 - a. Người thợ săn có bắn trúng con chim hay không?
 - b. Nếu không bắn trúng, anh ta phải chỉnh góc bắn là bao nhiêu để bắn trúng?
10. Một con thỏ chạy dọc theo phương Ox với vận tốc không đổi $v_t = 4m/s$. Một con cáo đứng cách thỏ $10m$ theo phương Oy cũng bắt đầu chạy hướng theo con thỏ với vận tốc $v_c = 6m/s$. Tìm quỹ đạo của con cáo. Lúc nào thì con cáo bắt được con thỏ?

4.1.2 Sóng cơ học

Sóng cơ học: là những dao động cơ học lan truyền theo thời gian trong một môi trường vật chất. Ở đây chỉ có trạng thái dao động được truyền đi, còn bản thân các phần tử vật chất chỉ dao động tại chỗ. Các hiện tượng sóng cơ học thường gặp trong đời sống là: sự dao động của mặt nước khi có vật thể rơi xuống, sự lan truyền âm thanh trong không khí, sự dao động của các bộ phận xe máy khi đang chạy.

Như đã nói ở trên, ta cần phân biệt rõ 2 chuyển động của sóng cơ học: (1) sự dao động và (2) sự truyền sóng. Trong giáo trình này, chúng ta chỉ đề cập đến sóng ngang là loại sóng có phương dao động vuông góc với phương truyền sóng. Nếu xét trên trục Ox , thì vật chất dao động tại chỗ theo trục Oy và lan truyền sự dao động theo trục Ox . Ví dụ đơn giản và dễ hiểu nhất là hình ảnh "sóng người" trên sân vận động trong các trận bóng đá.

Sự dao động: Một chất điểm M được gọi là dao động nếu tọa độ của nó tuân theo quy tắc: $y_M = A_M \sin(\omega t + \varphi)$. Trong đó A được gọi là biên độ, là độ dịch chuyển lớn nhất mà chất điểm có thể đạt được; ω được gọi là vận tốc góc, cho biết sự dao động nhanh hay chậm của sóng và φ được gọi là pha ban đầu, chỉ trạng thái ban đầu của sự dao động.

Sự truyền sóng: được đặt trưng bởi vận tốc truyền sóng v . Theo đó phương trình dao động tại điểm C do nguồn dao động từ M truyền đến có dạng $y_C = y_{CM} = A_M \sin(\omega t + \varphi - \omega d/v)$ với d là khoảng cách MN.

Ví dụ 4.1.5. Tại thời điểm $t = 0$, một nguồn dao động tác động vào điểm M ($x_M = 0$) trên mặt hồ đang nằm yên khiến cho điểm M dao động theo biểu thức $y = 3\sin(10t)$. Biết sóng truyền theo hướng Ox với vận tốc $v = 2$.

- (a) Tại giây thứ 3, điểm M đã truyền sóng đến vị trí nào của hồ? Xác định dao động M lúc đó.
- (b) Tại giây thứ 5, xác định những điểm nào có cùng dao động như M.
- (c) Tại giây thứ 7, xác định những điểm nào có cùng đang ở đỉnh sóng (dao động bằng biên độ).
- (d) Vẽ đồ thị thể hiện mặt hồ tại thời điểm $t = 10$.

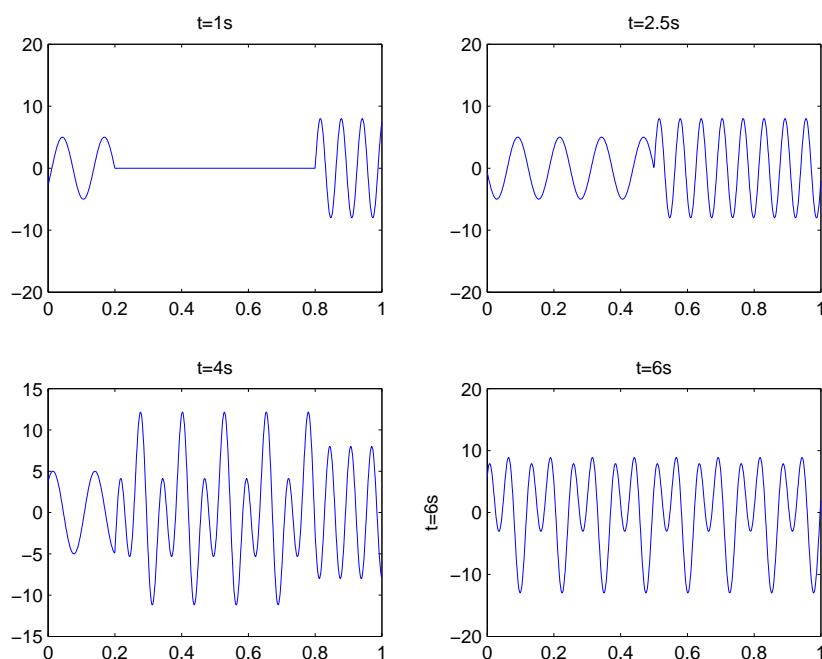
Các giá trị đầu khác bằng 0	» syms t d k2p
Các giá trị cho trước	» y=3*sin(10*t); v=2;
a) Vị trí sóng lan đến	» ta = 3; xa = v*ta
Dao động của M lúc đó	» yM=subs(y,t,ta)
b) $\omega d/v = k2\pi$	» dD=solve('10*d/2=k2p',d)
Chọn tùy ý hệ số k	» k=-50:50
Các điểm tương ứng	» dD=k*2*3.14/5
Các điểm thỏa yêu cầu	» dD=dD(dD<10); dD=dD(dD>0)
c) $\omega t - \omega d/v = \pi/2 + k2\pi$	» dD=solve('10*7-10*d/2=k2p+3.14/2',d)
Chọn tùy ý hệ số k	» k=-50:50
Các điểm tương ứng	» dD=1.3686-0.2*k*2*3.14;
Các điểm thỏa yêu cầu	» dD=dD(dD<14); dD=dD(dD>0)
d) Toạ độ sóng lan tới	» td=10; xd=v*td
Tạo các nút trên mặt hồ	» X=linspace(0,xd,1000)
Dao động của các nút	» Y=3*sin(10*td-10*X/v)
Vẽ đồ thị mặt hồ	» plot(X,Y)

Sự chồng chất dao động: Chúng ta sẽ nghiên cứu sự dao động của các vật chất chịu tác động của nhiều nguồn truyền cùng một lúc. Giả sử có 2 nguồn M và N , phương trình dao động của vật chất C là:

$$y_C = y_{CM} + y_{CN} \quad (4.6)$$

Ví dụ 4.1.6. Tại thời điểm $t = 0$, 2 nguồn M và N ở cách nhau khoảng $MN = 1m$ bắt đầu tạo dao động với biên độ $A_M = 5cm, A_N = 8cm$, vận tốc góc $\omega_M = 10rad/s, \omega_N = 20rad/s$, pha ban đầu $\varphi_M = \varphi_N = 0$. Cả 2 đều truyền dao động với vận tốc $v = 0.2m/s$. Tìm dao động của các vật chất nằm giữa đoạn MN tại giây thứ 10. (Lúc này dao động từ M đã truyền sang N và dao động từ N đã truyền sang M .)

Các giá trị cho trước	» t=10;v=0.2;MN=1;
Các giá trị cho trước	» AM=5; wM=10; AN=8; wN=20
Tạo các nút trên MN	» X=linspace(0,MN,1000)
Dao động do M truyền	» YCM=AM*sin(wM*t-wM*X/v)
Dao động do N truyền	» YCN=AN*sin(wN*t-wN*X/v)
Xếp xếp đúng chiều	» YCN=YCN(end:-1:1)
Dao động tổng hợp	» YC=YCM+YCN;
Vẽ đồ thị mặt hồ	» plot(X,YC);



Hình 4.1.2: Mô phỏng dao động mặt hồ

BÀI TẬP

- Trên mặt hồ nước rất rộng, một nguồn dao động được đặt tại $O(0,0)$ có biên độ $A = 3\text{cm}$, vận tốc góc $\omega = 5\text{rad/s}$, pha ban đầu $\varphi = 0$. Vận tốc truyền sóng là $v = 0.3\text{m/s}$. Mô phỏng sóng trên mặt hồ tại $t = 10\text{s}$.
- Tại thời điểm $t = 0$, nguồn M bắt đầu tạo dao động với biên độ $A_M = 5\text{cm}$, vận tốc góc $\omega_M = 10\text{rad/s}$, pha ban đầu $\varphi_M = 0$. Tại $t = 2$, nguồn N bắt đầu tạo dao động với biên độ $A_N = 8\text{cm}$, vận tốc góc $\omega_N = 20\text{rad/s}$. Cả 2 đều truyền dao động với vận tốc $v = 0.2\text{m/s}$. Cho biết $MN = 1\text{m}$.
 - Tìm dao động của các vật chất nằm giữa đoạn MN tại $t = 2$.
 - Tìm dao động của các vật chất nằm giữa đoạn MN khi 2 sóng gặp nhau.

- c. Tìm dao động của các vật chất nằm giữa đoạn MN khi sóng từ M truyền đến N .
 - d. Tìm dao động của các vật chất nằm giữa đoạn MN khi sóng từ N truyền đến M .
3. Giải lại bài toán 2, nhưng không tìm dao động trên đoạn MN mà tìm dao động trên cả mặt hồ.
 4. Giải lại bài toán 1 biết rằng không phải có 1 nguồn dao động mà có 3 nguồn dao động đặt trên đỉnh tam giác đều cạnh $l = 1.5m$.

4.2 Xác suất - Thống kê

4.2.1 Tính toán xác suất cơ bản

Bảng tóm tắt các hàm matlab liên quan đến các phân phối thường gặp

Phân phối	Hàm mật độ (PDF)	Hàm phân phối (CDF)	Tạo số ngẫu nhiên
Chuẩn	<code>normpdf(X, μ, σ)</code>	<code>normcdf(X, μ, σ)</code>	<code>normrnd(μ, σ, m, n)</code>
Đều	<code>unifpdf(X, a, b)</code>	<code>unifcdf(X, a, b)</code>	<code>unifrnd(a, b, m, n)</code>
Mũ	<code>expdpdf(X, μ)</code>	<code>expcdf(X, μ)</code>	<code>exprnd(μ, m, n)</code>
Nhị thức	<code>binopdf(X, N, P)</code>	<code>binocdf(X, N, P)</code>	<code>binornd(N, P, m, n)</code>
Poisson	<code>poisspdf(X, λ)</code>	<code>poisscdf(X, λ)</code>	<code>poissrnd(λ, m, n)</code>

Ví dụ 4.2.1.

- Tạo 100 số ngẫu nhiên có phân phối chuẩn với $\mu = 100$ và $\sigma = 15$
`x = normrnd(100,15,100,1)`
- Tạo 50 số ngẫu nhiên có phân phối nhị thức $B(20, 0.7)$
`y = binornd(20,15,50,1)`

Ví dụ 4.2.2. Mô tả phân phối chuẩn

```
mu = 100; sigma = 15; xmin = 70; xmax = 130; n = 100; k = 10000;
x = linspace(xmin, xmax, n);
p = normpdf(x, mu, sigma);
c = normcdf(x, mu, sigma);
```

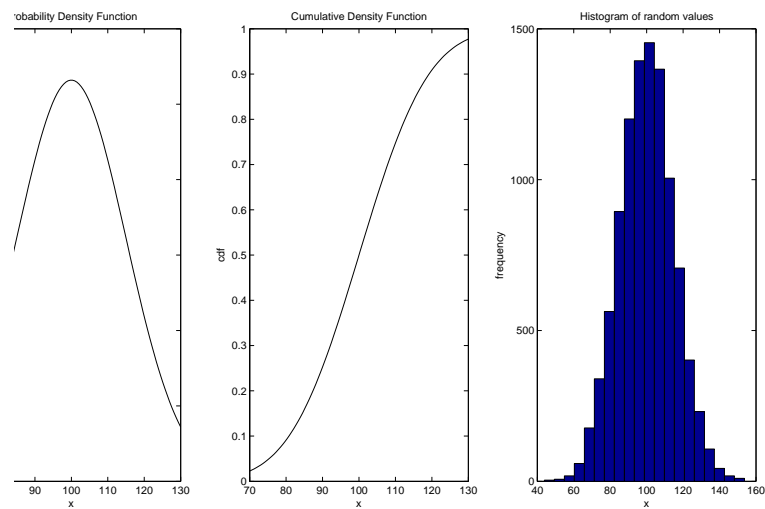
```

% Ve do thi ham mat do
subplot(1,3,1);
plot(x, p, 'k-');
xlabel('x'); ylabel('pdf'); title('Probability Density Function');

%Ve do thi ham phan phoi
subplot(1,3,2);
plot(x, c, 'k-');
xlabel('x'); ylabel('cdf'); title('Cumulative Density Function');

% Tao k = 10000 so ngau nhien co phan phoi chuan N(mu, sigma^2)
y = normrnd(mu, sigma, k, 1);
subplot(1,3,3);
hist(y, 20); % Ve do thi Histogram
xlabel( 'x' ); ylabel( 'frequency' ); title( 'Histogram of random
values' );

```



4.2.2 Thống kê mô tả

Bảng tóm tắt các hàm thống kê

Tên hàm	Mô tả
max	Giá trị lớn nhất
mean	Trung bình
median	Trung vị
min	Giá trị nhỏ nhất
mode	Giá trị có tần số xuất hiện lớn nhất
std	Độ lệch tiêu chuẩn
var	Phương sai
boxplot	Vẽ đồ thị boxplot
hist	Vẽ đồ thị histogram
qqplot	Vẽ đồ thị Q-Q Plot

Ví dụ 4.2.3.

```
load count.dat % Lay data mau
X = count(:,1);
% Tinh cac gia tri thong ke mo ta
mean(X); var(X); median(X);
% Ve do thi
boxplot(X); hist(X)
qqplot(X) % Kiem xem X co tuan theo phan phoi chuan
```

4.2.3 Kiểm định giả thiết: z - test và t - test

Giả định: số liệu là mẫu độc lập chọn từ phân phối chuẩn.

z - test: Giả sử đã biết phương sai; t - test: phương sai chưa biết.

Giả thiết $H_0: \mu = \mu_0$ (Đối thiết $H_A: \mu \neq \mu_0$)

Giá trị thống kê z và t :

$$z = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}}$$

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

Z-test: `[h, p, ci, zval] = ztest(x, m, sigma, alpha)`, trong đó

- `x`: vectơ số liệu; `m`: giá trị trung bình cần kiểm định; `sigma`: độ lệch tiêu chuẩn σ ; `alpha`: mức ý nghĩa α .
- `h` = 0 hoặc 1; `h` = 1: bác bỏ H_0 .
- `p`: P - value; `ci`: khoảng tin cậy $(1 - \alpha)\%$.
- `zval`: giá trị thống kê Z .

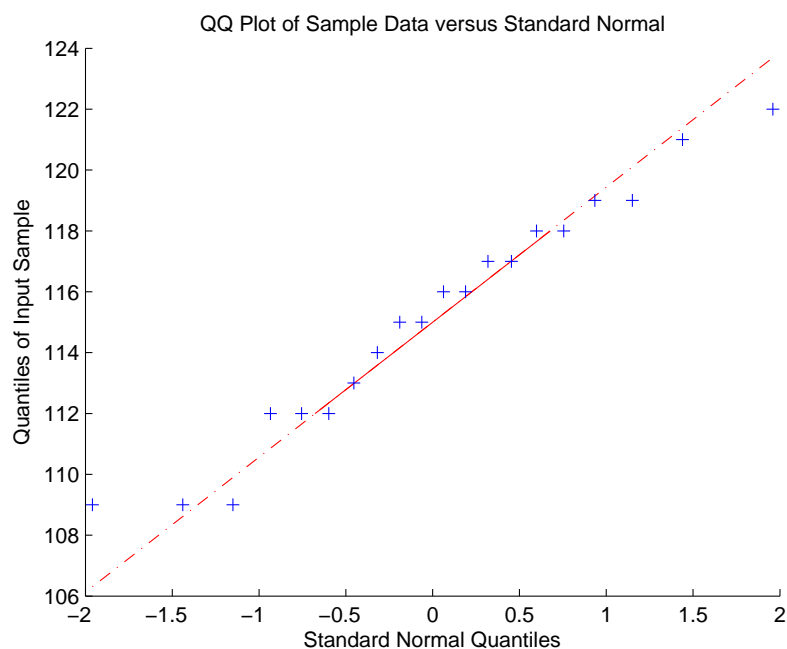
T-test: `[h, p, ci, stats] = ttest(x, m, sigma, alpha)`, trong đó

- `h`, `p`, `ci`: tương tự như hàm `ztest`.
- `stats`: xuất các giá trị sau: `tstat` - giá trị thống kê t ; `df`: bậc tự do; `sd`: độ lệch tiêu chuẩn ước lượng.

Ví dụ 4.2.4.

`load gas % File data chua so lieu ve gia gas gom price1 va price2`

`qqplot(price1) % Kiem tra gia dinh ve phan phoi chuan doi voi price1`



% So sánh giá gas trung bình với 115, giả sử đã biết phương sai $\sigma^2 = 4^2$

```
[h, pvalue, ci] = ztest(price1, 115, 4)
```

```
h = 0
```

```
pvalue = 0.8668
```

```
ci =
```

```
    113.40
```

```
    116.90
```

BÀI TẬP

1. Thực hiện tương tự như ví dụ 4.2.1, hãy mô phỏng phân phối mũ với tham số $\lambda = 2$.

2. Như bài tập 1, viết script mô phỏng phân phối nhị thức $B(10, 0.7)$.

3. Giả sử chỉ số IQ tuân theo phân phối chuẩn với trung bình là 100 và độ lệch tiêu chuẩn 15. Chọn ngẫu nhiên một người, tính xác suất để chọn được

a. Người có chỉ số IQ từ 110 đến 130.

b. Người có chỉ số IQ trên 120.

4. Đo chiều dài đuôi của 15 con thằn lằn, (Đv: cm)

18.4, 22.2, 24.5, 26.4, 27.5, 28.7, 30.6, 32.9

32.9, 34.0, 34.8, 37.5, 42.1, 45.5, 45.5

a. Tính các giá trị thống kê mô tả.

b. Vẽ đồ thị boxplot, histogram.

5. Một nhà máy sản xuất thức ăn đóng hộp tuyên bố rằng trọng lượng trung bình các hộp thức ăn xuất xưởng là 140 (g). Lấy mẫu 12 hộp thức ăn

138, 125, 155, 130, 140, 150

130, 120, 125, 134, 160, 135

Dùng kiểm định z - test hoặc t- test để kiểm chứng khẳng định của nhà máy với mức ý nghĩa $\alpha = 5\%$.

CHƯƠNG 5

Giao diện đồ họa người dùng – GUI

5.1 Tổng quan về các tệp tin GUI

5.1.1 Tệp tin M và tệp tin FIG

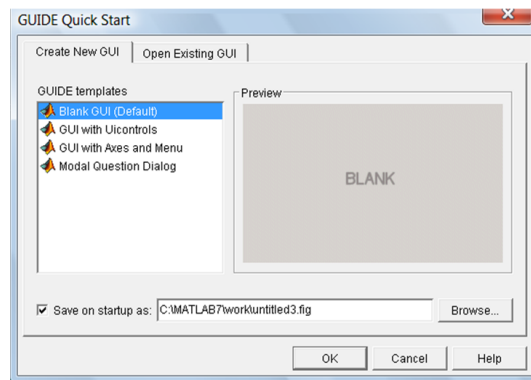
Tạo tệp tin GUI bằng câu lệnh GUIDE. Một tệp tin GUI sẽ được tạo ra và lưu trữ trong hai tệp tin:

- Tệp tin FIG, đuôi mở rộng .fig, chứa cách bố trí (layout) GUI và các thành phần của GUI như là: push buttons, axes, panels, menus v.v...
- Tệp tin M, đuôi mở rộng .m, chứa các đoạn mã khởi tạo ban đầu và các thủ tục (template) cho các hàm gọi lại (callback). Ngoài ra, phải thêm các callback khi viết các thành phần của GUI.

Tệp tin FIG và tệp tin M phải cùng tên và thường có chung đường dẫn. Thực thi: Tạo GUI → tổ chức GUI trong Layout Editor (công việc được lưu trong tệp tin FIG) → thực thi GUI (công việc được lưu trong tệp tin M).

5.1.2 Cấu trúc tệp tin M của GUI

Tệp tin M của GUI được GUIDE tạo ra là một tệp tin hàm, tên hàm chính trùng với tên tệp tin. Các callback trong tệp tin là các hàm con của hàm chính.



Hình 5.2.1: Tạo và đặt tên cho một GUI mới

Khi lệnh GUIDE tạo ra tệp tin M, nó tự động đưa vào các thủ tục của các callback thường được sử dụng nhất. Tệp tin M chứa đoạn mã khởi tạo, và ta phải thêm các đoạn mã vào các callback thành phần trong GUI theo mục đích của công việc.

5.2 Ví dụ tạo một GUI đơn giản

Ví dụ 5.2.1. Tạo một GUI có một nút bấm, khi bấm vào sẽ hiện giờ hiện hành

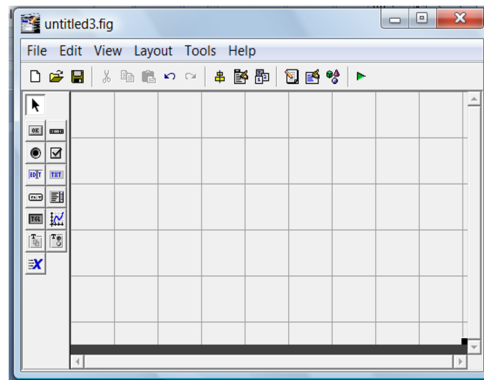
- Bước 1: Tạo GUI bằng lệnh GUIDE, hình vẽ [5.2.1](#)
- Bước 2: Chọn và đặt pushbutton, hình vẽ [5.2.3](#)
- Bước 3: Chỉnh thuộc tính component, hình vẽ [5.2.4](#) – [5.2.8](#)
- Đặt tên cho chương trình: Chọn Property Inspector → Chỉnh Name: Time, hình vẽ [5.2.9](#)

Ví dụ 5.2.2. Tạo một GUI vẽ đồ thị.

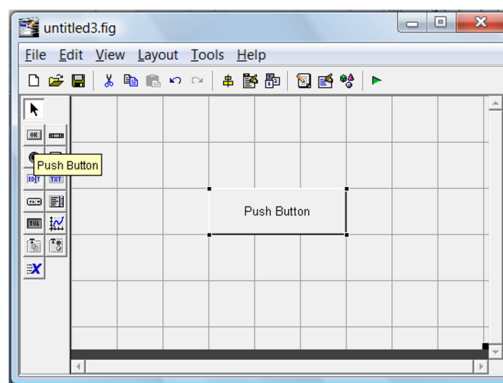
5.3 Một số component trong GUI

Push Button(Nút nhấn)

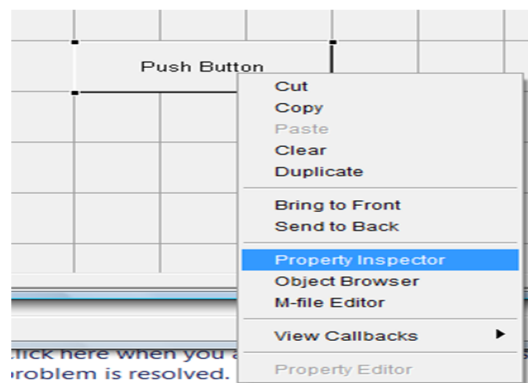
Khi nhấp chuột vào một nút, thì hàm sau sẽ được gọi



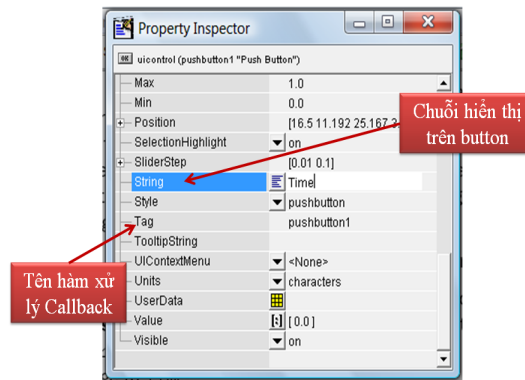
Hình 5.2.2: Layout editor



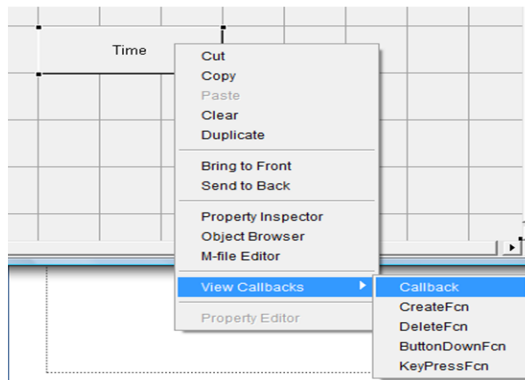
Hình 5.2.3: Chọn và đặt pushbutton



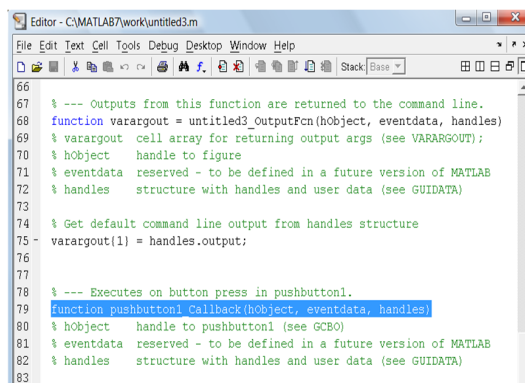
Hình 5.2.4: Chỉnh thuộc tính component



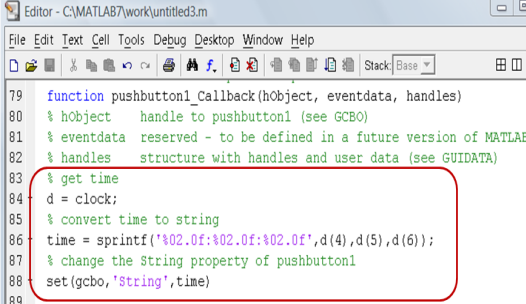
Hình 5.2.5: Thay đổi giá trị thuộc tính



Hình 5.2.6: Cài đặt Callback cho component



Hình 5.2.7: Tập tin M của GUI

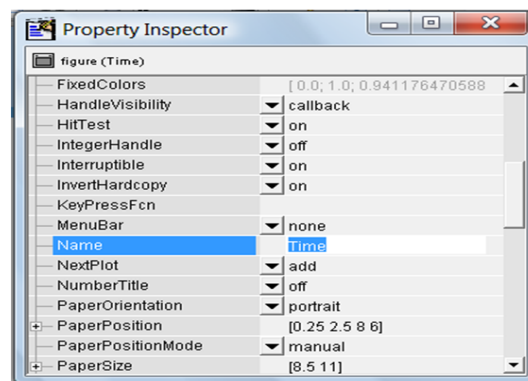


```

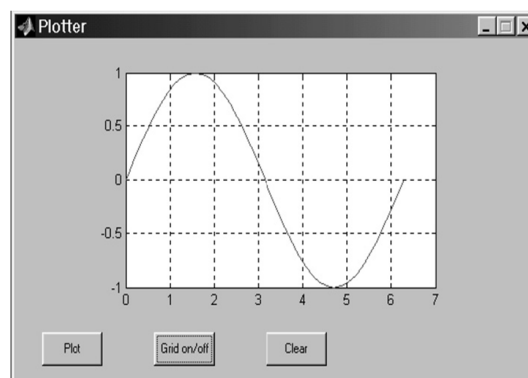
79 function pushbutton1_Callback(hObject, eventdata, handles)
80 % hObject    handle to pushbutton1 (see GCBO)
81 % eventdata  reserved - to be defined in a future version of MATLAB
82 % handles    structure with handles and user data (see GUIDATA)
83 % get time
84 d = clock;
85 % convert time to string
86 time = sprintf('%02.0f:%02.0f:%02.0f',d(4),d(5),d(6));
87 % change the String property of pushbutton1
88 set(gcbo,'String',time)
89

```

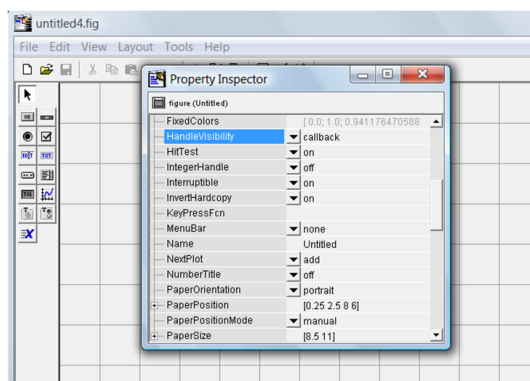
Hình 5.2.8: Thêm đoạn mã xử lý



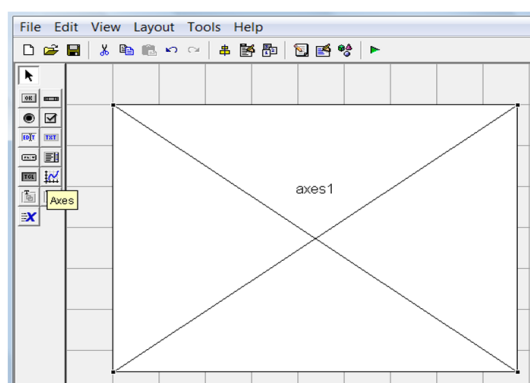
Hình 5.2.9: Đặt tên cho chương trình là Time



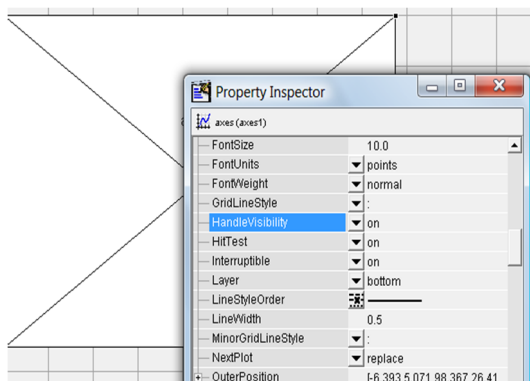
Hình 5.2.10: GUI vẽ đồ thị



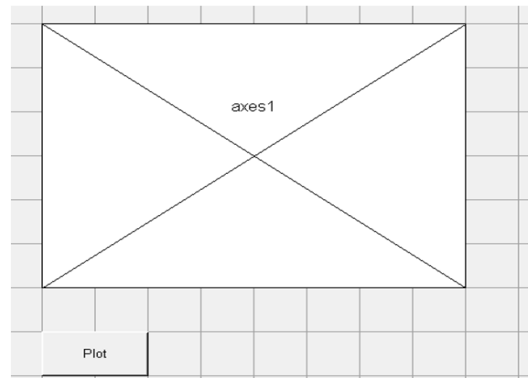
Hình 5.2.11: Handle Visibility → callback



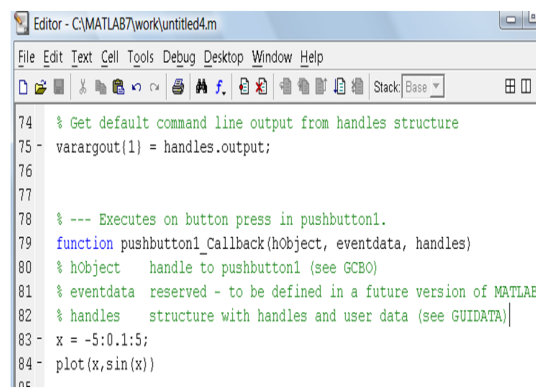
Hình 5.2.12: Chọn và đặt Axes



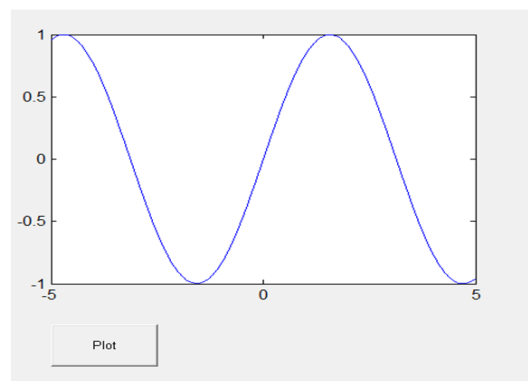
Hình 5.2.13: Handle Visibility của Axes → On



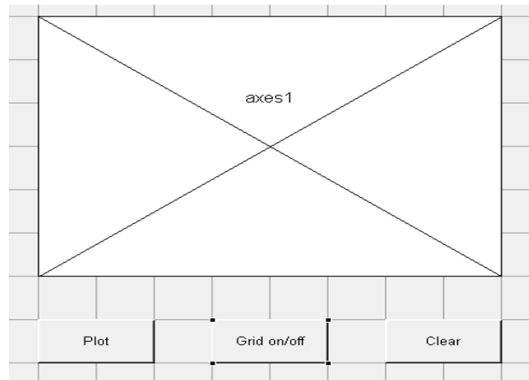
Hình 5.2.14: Thêm Button Plot



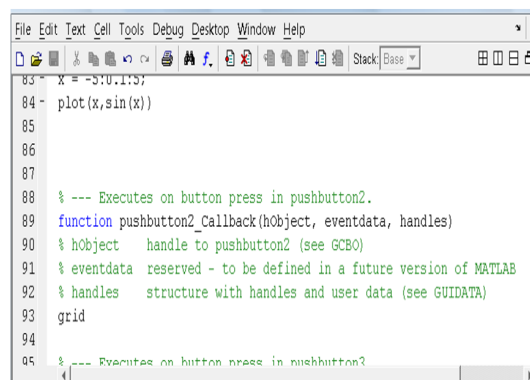
Hình 5.2.15: Cài đặt Callback cho Button Plot



Hình 5.2.16: Thêm các button grid on/off và clear



Hình 5.2.17: Cài đặt callback cho Grid on/off



Hình 5.2.18: Cài đặt callback cho Clear

```
function pushbutton1_Callback(hObject, eventdata, handles)
... ..
```

Ví dụ 5.3.1. Nhấn nút thì thay tên của nút (thuộc tính ‘String’ của nút)

```
function pushbutton1_Callback(hObject, eventdata, handles)
set(gcbo,'String','ten gi do');
```

Một số thuộc tính quan trọng của Button:

- Enable: cho phép nhấn (on/off)
- String: chuỗi hiển thị trên nút
- Tag: tên đối tượng nút phân biệt với các đối tượng khác trong chương trình
- Value: giá trị của nút.

Toggle Button, Radio Button và Check Box (Nút bật tắt, Nút chọn và Ô check)

- Thuộc tính của Toggle Button: Thuộc tính Value sẽ bằng thuộc tính Max nếu nút được nhấn hoặc được chọn, ngược lại sẽ bằng thuộc tính Min.

```
function togglebutton1_Callback(hObject, eventdata, handles)
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
% Toggle button được nhấn
...
elseif button_state == get(hObject,'Min')
% Toggle button không được nhấn
...
end
```

- Mặc định Max=1, Min=0. Cũng có thể thay đổi trạng thái cho các Toggle Button, Radio Button và Check Box bằng cách đặt lại các thuộc tính Value, Max, Min.

```
set(handles.togglebutton1,'Value',get(handles.togglebutton1,'Max'))
```

đặt nút togglebutton1 sang trạng thái được nhấn

```
set(handles.radiobutton1,'Value',get(handles.radiobutton1,'Min'))
```

đặt nút radiobutton1 sang trạng thái bỏ chọn Ô nhập văn bản: Edit Text

- Thuộc tính String chứa chuỗi mà người dùng nhập vào

```
function edittext1_Callback(hObject, eventdata, handles)
user_string = get(hObject,'String');
...
```

- Nếu thuộc tính Max, Min được đặt sao cho $\text{Max} - \text{Min} > 1$ thì người dùng có thể nhập nhiều hàng.
- Callback được gọi khi đang thay đổi nội dung của Edit Text mà nhấp chuột ra bên ngoài (nhưng vẫn bên trong GUI), hoặc khi người dùng nhấn Enter (với Edit Text một hàng), Ctrl+Enter (với Edit Text nhiều hàng).
- Thuộc tính String trả về một chuỗi. Muốn nhận dữ liệu số thì sử dụng hàm str2double. Nếu người dùng không nhập số thì str2double trả về NaN.

```
function edittext1_Callback(hObject, eventdata, handles)
user_entry = str2double(get(hObject,'string'));
if isnan(user_entry)
errordlg('You must enter a numeric value','Bad Input','modal')
return
end
% Proceed with callback...
```

Thanh trượt (Slider)

- Thuộc tính Value xác định trạng thái của Slider.
- Thuộc tính Max và Min xác định cực đại và cực tiểu cho thuộc tính Value của Slider.


```
function slider1_Callback(hObject, eventdata, handles)
slider_value = get(hObject,'Value');
% Proceed with callback...
```

List box và Pop-up menu

- Thuộc tính Value chứa chỉ số của mục được chọn. Chỉ số của mục đầu tiên là 1.
- Thuộc tính String là mảng cell chứa danh sách các mục chọn của List Box và Pop-up Menu.
- Ví dụ dưới đây cho phép lấy giá trị của chuỗi được chọn trong List Box.

```
function listbox1_Callback(hObject, eventdata, handles)
index_selected = get(hObject,'Value');
list = get(hObject,'String');
item_selected = list{index_selected};
% Convert from cell array to string
```

- Callback của List Box được gọi sau sự kiện nhả chuột hoặc nhấn một số phím đặc biệt.
- Khi các phím mũi tên thay đổi thuộc tính Value của List Box, callback Callback được gọi đồng thời đặt thuộc tính SelectionType của figure thành normal.
- Phím Enter và Space không làm thay đổi thuộc tính Value của List Box, nhưng Callback vẫn được gọi, đồng thời đặt thuộc tính SelectionType của figure thành open.
- Nếu người dùng nhấp đúp chuột thì Callback được gọi sau mỗi lần nhấp. MatLab đặt thuộc tính SelectionType của figure thành normal ở lần thứ nhất và thành open ở lần thứ hai.
- Do đó ta có thể biết được người dùng nhấp đúp chuột hay không dựa vào việc truy vấn thuộc tính SelectionType của figure.

Button Group

- Button Group chứa các thành phần GUI, đồng thời giúp ta quản lý việc lựa chọn của Radio Button và Toggle Button.
- Nếu Button Group chứa Radio Button, Toggle Button hoặc cả hai thì Button Group chỉ cho phép một trong số chúng được chọn. Nếu một cái được chọn thì những cái khác sẽ được bỏ chọn.
- Callback SelectionChangeFcn của Button Group được gọi khi có sự thay đổi chọn.
- Tham số đầu vào hObject trở tới Radio Button hoặc Toggle Button được chọn.
- Các callback của các thành phần khác trong GUI có thể truy xuất tới đối tượng được chọn thông qua thuộc tính SelectedObject của Button Group.

Axes (Hệ trục tọa độ)

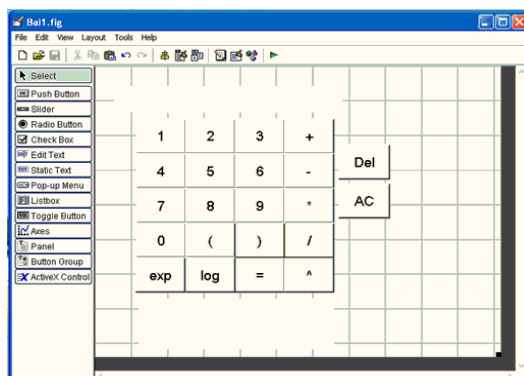
- Hệ trục tọa độ dùng để vẽ đồ thị hoặc hiển thị hình ảnh.
- Muốn vẽ đồ thị lên một hệ trục tọa độ, ta thêm handles tới hệ trục tọa độ đó vào hàm muốn sử dụng.

Ví dụ 5.3.2. vẽ đồ thị lên hệ trục axes1

```
[X,Y] = meshgrid(-3:.125:3);  
Z = peaks(X,Y);  
meshc(handles.axes1,X,Y,Z);  
axis([-3 3 -3 3 -10 5]);
```

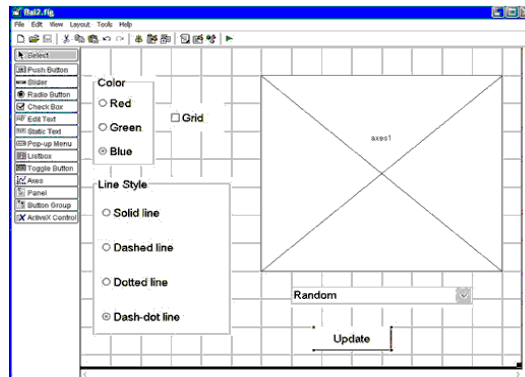
BÀI TẬP

1. Tạo một giao diện người dùng như hình vẽ 5.3.1. Chương trình mô phỏng máy tính bỏ túi dùng để tính các phép toán thông thường. Trong đó, nút dấu bằng là togglebutton, các nút còn lại đều là pushbutton. Các tag cụ thể:

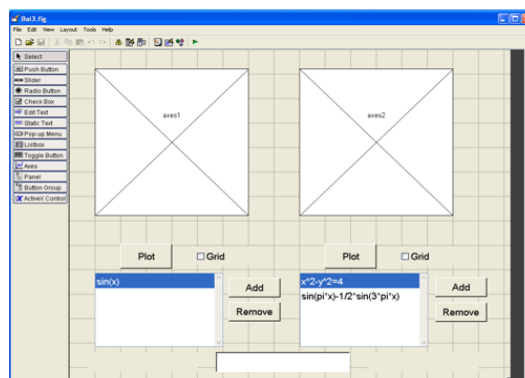


Hình 5.3.1: Máy tính với các chức năng thông thường

- Dấu bằng: **togglebutton_bang**. Nó ở trạng thái được nhấn khi người dùng nhập xong một chuỗi và nhấn vào nó. Nút này ở trạng thái không nhấn khi người dùng nhấn vào các nút khác để bắt đầu nhập chuỗi mới.
 - Màn hình hiển thị gồm: **text_bieuthuc**, hiển thị biểu thức người dùng nhập và kết quả biểu thức. Và **text_canhbao**, hiển thị thông báo khi nhập sai.
 - Nút Del và AC dùng để xóa: **pushbutton_del**, **pushbutton_AC**, nút Del xóa một ký tự cuối, nút AC xóa toàn bộ chuỗi.
 - Các nút nhập biểu thức: **pushbutton_1**, **pushbutton_2**, **pushbutton_cong**, ... để nhập biểu thức cần tính.
2. Thiết kế GUI như hình vẽ 5.3.2. Chương trình sẽ vẽ đồ thị hàm số được chọn trong **popupmenu1** lên hệ trục tọa độ khi nhấn vào nút **pushbutton_Update** với các tham số đường nét, màu sắc được chọn trong hai **buttongroup**, đó là **uipanel_Color** và **uipanel_LineStyle**. Ngoài ra đồ thị sẽ có thêm các ô lưới nếu checkbox Grid được chọn, ngược lại đồ thị sẽ không có các ô lưới.
 3. Thiết kế GUI như hình vẽ 5.3.3. Bao gồm:
 - Hai hệ trục tọa độ **axes1** và **axes2** dùng để hiển thị đồ thị.
 - Các nút vẽ và ô check để hiển thị lưới trên đồ thị: **pushbutton_Plot1**, **pushbutton_Plot2**, **checkbox_Grid1**, và **checkbox_Grid2**.



Hình 5.3.2: GUI vẽ đồ thị 2D



Hình 5.3.3: GUI vẽ đồ thị 2D với các tính năng cập nhật các hàm cần vẽ

- Hai danh sách các hàm số cần vẽ và các nút thêm, xóa hàm số: **listbox1**, **listbox2**, **pushbutton_Add1**, **pushbutton_Remove1**.
- Ô edit text dùng để nhập thêm hàm số và hai static text dùng để hiển thị thông báo lỗi: **edit1**, **text1**, **text2**.
- Khi người dùng nhấn các nút Plot, đồ thị của hàm số được chọn trong list box sẽ được hiển thị trên các hệ trục tọa độ tương ứng.
- Khi người dùng nhập hàm số vào text box và nhấn nút Add thì hàm số sẽ được thêm vào list box tương ứng.
- Khi người dùng nhấn nút Remove thì hàm số được chọn sẽ bị xóa khỏi list box tương ứng.