

Mục lục

1 Chuẩn bị	1
1.1 Kiến thức giải tích	1
1.2 Sai số làm tròn và số học máy tính	3
1.3 Thuật toán và sự hội tụ	3
1.4 MATLAB: ngôn ngữ tính toán và lập trình	3
1.5 MATLAB: giải tích và đại số	5
2 Giải phương trình một biến	13
2.1 Phương pháp chia đôi	13
2.2 Phương pháp Newton và mở rộng	15
2.3 Lặp điểm bất động	21
2.4 Phân tích sai số của các phương pháp lặp	25
2.5 Tăng tốc độ hội tụ	25
2.6 Nghiệm của đa thức và phương pháp Müller	25
3 Nội suy và xấp xỉ bằng đa thức	27
3.1 Nội suy tổng quát	27
3.2 Đa thức nội suy	28
3.3 Xấp xỉ số liệu và phương pháp Neville	32
3.4 Sai phân chia	32
3.5 Nội suy Hermite	32
3.6 Nội suy Newton	32
3.7 Nội suy spline bậc ba	36
3.8 Đường cong tham số	36

Chương 1

Chuẩn bị

1.1	Kiến thức giải tích	1
1.2	Sai số làm tròn và số học máy tính	3
1.3	Thuật toán và sự hội tụ	3
1.4	MATLAB: ngôn ngữ tính toán và lập trình	3
1.5	MATLAB: giải tích và đại số	5

1.1 Kiến thức giải tích

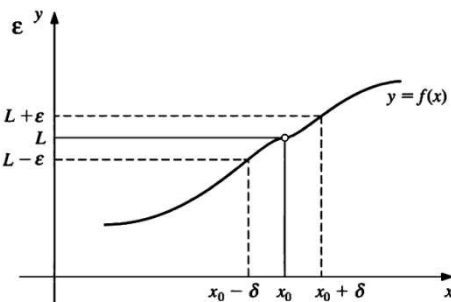
1.1.1 Giới hạn và tính liên tục

Định nghĩa 1.1. Hàm f xác định trên tập X các số thực có **giới hạn** L tại x_0 , ký hiệu

$$\lim_{x \rightarrow x_0} f(x) = L$$

nếu với mọi $\varepsilon > 0$, tồn tại $\delta > 0$ sao cho

$$|f(x) - L| < \varepsilon \quad \text{với mọi } x \in X \quad \text{và} \quad 0 < |x - x_0| < \delta.$$



Định nghĩa 1.2. Cho hàm f xác định trên tập X các số thực và $x_0 \in X$. f **liên tục** tại x_0 nếu

$$\lim_{x \rightarrow x_0} f(x) = f(x_0).$$

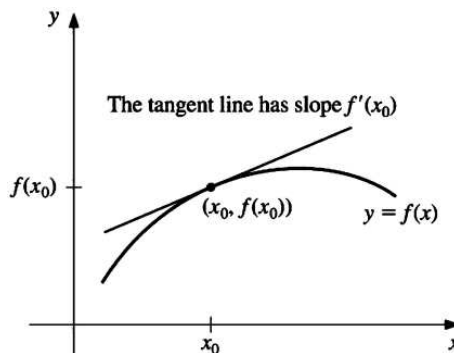
f **liên tục trên X** nếu nó liên tục tại mọi điểm của X .

1.1.2 Tính khả vi

Định nghĩa 1.3. Cho hàm f xác định trên khoảng mở chứa x_0 . f **khả vi** tại x_0 nếu tồn tại

$$f'(x_0) = \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}.$$

$f'(x_0)$ gọi là **đạo hàm** của f tại x_0 . Hàm có đạo hàm tại mọi điểm của X gọi là **khả vi trên X** .



Định lý 1.1. Giả sử $f \in C^n[a, b]$, tồn tại $f^{(n+1)}$ trên $[a, b]$, và $x_0 \in [a, b]$. Khi đó, với mọi $x \in [a, b]$, tồn tại số $\xi(x)$ ở giữa x_0 và x sao cho

$$f(x) = P_n(x) + R_n(x)$$

trong đó

$$\begin{aligned} P_n(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \\ &= \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k \end{aligned}$$

và

$$R_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)^{n+1}.$$

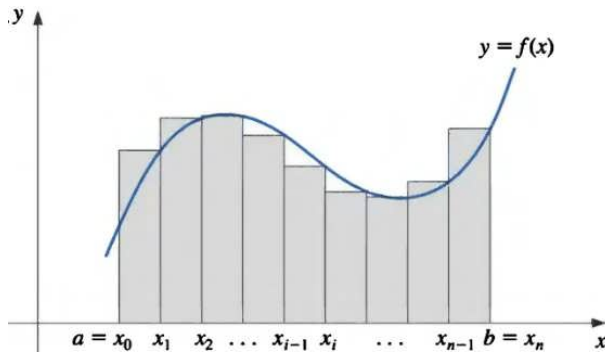
$P_n(x)$ là **đa thức Taylor bậc n** của f quanh x_0 , và $R_n(x)$ là **phần dư** (hay **sai số cắt**) của $P_n(x)$.

1.1.3 Tích phân

Định nghĩa 1.4. Tích phân Riemann của hàm f trên đoạn $[a, b]$ là giới hạn sau, miễn là nó tồn tại

$$\int_a^b f(x) dx = \lim_{\max \Delta x_i \rightarrow 0} \sum_{i=1}^n f(z_i) \Delta x_i$$

trong đó $a = x_0 \leq x_1 \leq \dots \leq x_n = b$, $\Delta x_i = x_i - x_{i-1}$ với $i = 1, 2, \dots, n$, và $z_i \in [x_{i-1}, x_i]$ tùy ý.



1.2 Sai số làm tròn và số học máy tính

1.2.1 Số nhị phân máy

1.2.2 Số thập phân máy

1.2.3 Số học hữu hạn chữ số

1.2.4 Số học lồng nhau

1.3 Thuật toán và sự hội tụ

1.4 MATLAB: ngôn ngữ tính toán và lập trình

1.4.1 Lý do chọn MATLAB

Các ngôn ngữ tính toán và lập trình mạnh và phổ biến như MATLAB, Mathematica, Python, Maple, tích hợp các lệnh, gói lệnh giải các bài toán phổ biến. Nắm vững thuật toán giúp ta triển khai giải bài toán trên các ngôn ngữ khác như C, C++, Java, FORTRAN,...

Thông tin	MATLAB	Mathematica	Python
Năm ra đời	1989	1986	1989
Tác giả, công ty	MathWorks	Wolfram Research	Guido van Rossum
Hệ điều hành: Windows (1), macOS (2), Linux (3), Android (4), iOS (5), Raspberry Pi (6)	1, 2, 3	1, 2, 3, 6	tất cả
Phiên bản năm 2021	R2021a (9.10)	12.2	3.9.5
Giá	49 – 2 150\$ không kèm Toolbox	177 – 5 780\$/năm, miễn phí trên (6)	miễn phí, mã nguồn mở
Dung lượng tải – cài đặt trên Windows	20.8 – 30.5GB	4.3 – 11.9GB	27 – 100.6MB
Độ phổ biến theo chỉ số PYPL*	1.71%		29.9%
Độ phổ biến theo chỉ số TIOBE	1.23%		11.87%

MATLAB có khả năng tính toán mạnh mẽ, ngôn ngữ dễ hiểu, dễ lập trình, được cộng đồng khoa học kỹ thuật, kinh tế,... đón nhận và sử dụng rộng rãi. Hầu hết các bài toán đề cập trong cuốn sách, với sự hỗ trợ của MATLAB, đều được giải quyết ngắn gọn, mà không đòi hỏi ta phải nhớ quá nhiều kiến thức toán học.

Có hai cách tiếp cận và sử dụng MATLAB

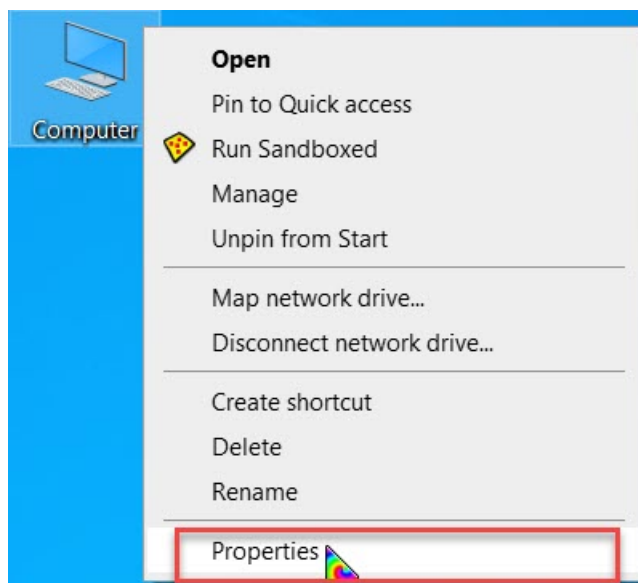
Cách 1: Đăng ký và sử dụng online tại matlab.mathworks.com. Với tài khoản chưa mua bản quyền, được dùng miễn phí 20 giờ / tháng.

Cách 2: Tải về từ mathworks.com/products/get-matlab.html?s_tid=gn_getml. Trước khi tải, cần kiểm tra tính tương thích của phiên bản MATLAB với hệ điều hành tại

mathworks.com/support/requirements/matlab-system-requirements.html

Để tìm phiên bản của hệ điều hành, nhấp chuột phải vào biểu tượng Computer

*Số liệu từ <https://statisticstimes.com/>



View basic information about your computer

Windows edition

Windows 10 Pro
© Microsoft
Corporation. All rights
reserved.



Windows 10

System

Processor:	Intel(R) Core(TM) i5 CPU M 520 @ 2.40GHz 2.40 GHz
Installed memory (RAM):	4.00 GB (3.87 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

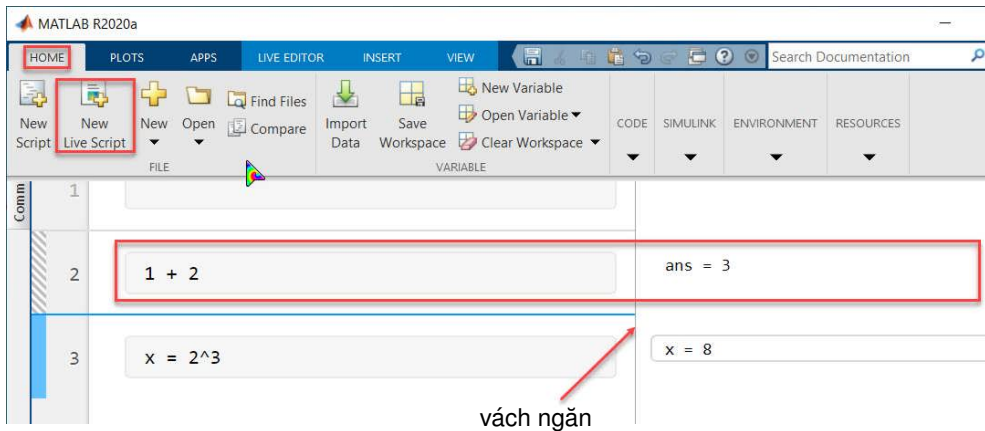
Computer name, domain and workgroup settings

Computer name:	NDT
Full computer name:	NDT
Computer description:	Thinh's Computer
Workgroup:	WORKGROUP

 [Change settings](#)

1.5 MATLAB: giải tích và đại số

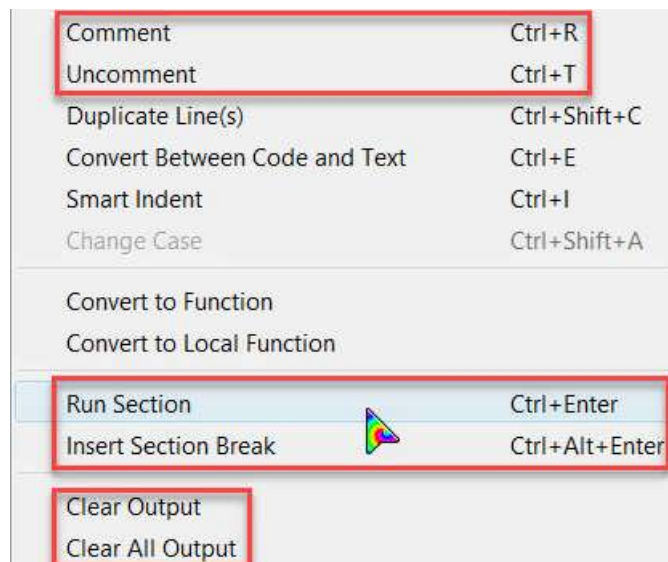
Chọn [Menu]HOME → *New Live Script* để tạo sổ tay *.mlx.



Mỗi sổ tay có thể chia thành nhiều phần, mỗi phần chia thành hai cột: ô lệnh bên trái và kết quả bên phải, được điều chỉnh độ rộng bằng vách ngăn. Các sổ tay là liên thông, tức là tức là đối tượng nhận giá trị nào trên sổ tay này, thì cũng nhận giá trị đó trên các sổ tay khác, hay sổ tay này có thể gọi đối tượng trong sổ tay khác*.

- Để chạy lệnh trong một ô, (1) nhấn *Chuột phải* → *Run Section*, hoặc (2) nhấn phím tắt tương ứng *Ctrl Enter*.

Các ô nhớ sẽ lưu giá trị của đối tượng theo thứ tự được chạy của ô, chứ không phải theo thứ tự từ trên xuống.



Để chạy toàn bộ sổ tay, (1) chọn *[Menu]LIVE EDITOR* → *Run*, hoặc (2) nhấn *F5*.

* Các sổ tay trong Mathematica cũng liên thông giống MATLAB, nhưng trong Python, các sổ tay là độc lập

- Để ngắt sang ô mới
 - Đưa con trỏ tới vị trí cần ngắt, thường là tại cuối ô
 - (1) Nhấn *Chuột phải* → *Insert Section Break*, hoặc (2) *Ctrl + Alt + Enter*

Mỗi lệnh nên đặt trong một ô, trừ khi các lệnh tham gia vào việc thay đổi giá trị của biến.

Trong phần này, các cú pháp trong MATLAB được diễn đạt tổng quát, hoặc minh họa bằng ví dụ cụ thể.

1.5.1 Hằng số / logic

Hằng	MATLAB
e	<code>exp(1)</code>
π	<code>pi</code>
∞	<code>inf</code>

TRUE, 1	<code>true</code>
FALSE, 0	<code>false</code>

1.5.2 Phép toán số học / logic / so sánh

Phép toán	MATLAB	Phép so sánh	MATLAB
$a + b$	<code>a + b</code>	$a < b$	<code>a < b</code>
$a - b$	<code>a - b</code>	$a > b$	<code>a > b</code>
ab	<code>a * b</code>	$a \leq b$	<code>a <= b</code>
$\frac{a}{b}$	<code>a / b</code>	$a \geq b$	<code>a >= b</code>
a^b	<code>a ^ b</code>	$a = b$	<code>a == b</code>
-----		$a \neq b$	<code>a ~= b</code>
$\neg p$	<code>~p</code>		
$p \wedge q$	<code>p & q</code>		
$p \vee q$	<code>p q</code>		

theo thứ tự ưu tiên $\wedge \rightarrow *, / \rightarrow +, -, \text{ và } \neg, \wedge, \vee$. Ngoài ra, với biểu thức phức tạp, ta dùng dấu nhóm biểu thức ().

1.5.3 Hàm sơ cấp

Hàm	MATLAB
\sqrt{x}	<code>sqrt(x)</code>

$\sqrt[n]{x^m}$	$x \wedge (m/n)$
$\sin x, \cos x, \tan x, \cot x$	$\sin(x), \cos(x), \tan(x), \cot(x)$
$\arcsin x, \arccos x, \arctan x, \operatorname{arccot} x$	$\operatorname{asin}(x), \operatorname{acos}(x), \operatorname{atan}(x), \operatorname{acot}(x)$
$\ln x, \log_2 x, \lg x$	$\log(x), \log2(x), \log10(x)$
$\log_a x \rightarrow$ đổi cơ số	$\log(x) / \log(a)$

1.5.4 **Lệnh thường dùng**

Hàm	Kết quả	Mô tả
<code>round(7.019, 2)</code>	7.0200	làm tròn lấy 2 chữ số sau dấu phẩy
<code>abs(-3)</code>	$ -3 = 3$	
<code>min(1, -2)</code>	-2	số nhỏ nhất trong hai số
<code>min([3, 1, -2])</code>	-2	số nhỏ nhất của dãy (véc tơ). Tương tự với <code>max</code>
<code>min([3; 1; -2])</code>	-2	
<code>sum([0, 4, 1])</code>	5	tổng của dãy
<code>factorial(3)</code>	$3! = 6$	

1.5.5 **Biến**

Cú pháp khai báo biến

```
var = expr
```

trong đó `expr` có thể là biểu thức toán học, xâu,... Chẳng hạn, lệnh

```
1 x = 1
2 x = x + 2
```

cho kết quả của biến `x` là 3.

1.5.6 **Khai báo hàm**

Khai báo hàm giúp làm việc với hàm đó đơn giản, ngắn gọn, dễ hiểu hơn, mang tính tự động hóa nhiều hơn.

- Hàm toán học thường khai báo tại bất cứ chỗ nào bằng từ khóa `@`
- Hàm có tính toán phức tạp trước khi trả về kết quả được khai báo bằng từ khóa `function`, và đặt tại
 - phần cuối của số tay*, hoặc
 - trong tệp `*.m`. Mục này sẽ được trình bày chi tiết ở dưới.

*không cần chạy ô này, lệnh gọi hàm ở ô trên sẽ tự tìm

Ví dụ 1.1. $f(x) = x^2$.

Cách 1:

```
1 f = @(x) x^2
```

Cách 2:

```
1 function y = f(x)
2 y = x^2;
3 end
```

Khi đó lệnh

```
1 f(-3)
```

cho kết quả là 9.

Ví dụ 1.2. Hàm nhiều biến $f(x, y) = x + y$.

Cách 1:

```
1 f = @(x, y) x + y
```

Cách 2:

```
1 function z = f(x, y)
2 z = x + y;
3 end
```

Khi đó lệnh

```
1 f(1, 2)
```

cho kết quả là 3

Ví dụ 1.3. Hàm có đối số là vectơ $f(x_1, x_2) = x_1 - x_2$.

Cách 1:

```
1 f = @(x) x(1) - x(2)
```

Cách 2:

```
1 function y = f(x)
2 y = x(1) - x(2);
3 end
```

Khi đó, hai lệnh

```
1 f([1, 2])
2 f([1; 2])
```

đều cho kết quả là -1 .

Ví dụ 1.4. Hàm véc tơ $f(x, y) = (x + y, x - y)$.

Cách 1:

```
1 f = @(x, y) [x + y, x - y]
2 f(1, 2) % → (3, -1)
```

Cách 2:

```
1 function [cong, tru] = g(x, y)
2 cong = x + y;
3 tru = x - y;
4 end
```

Khi đó

```
1 [a, b] = f(1, 2)
```

cho kết quả $a = 3, b = -1$, còn

```
1 a = f(1, 2)
```

cho kết quả $a = 3$, ứng với thành phần đầu tiên cong của hàm.

Cách 3:

```
1 function z = f(x, y)
2 z(1) = x + y;
3 z(2) = x - y;
4 end
```

Khi đó lệnh

```
1 f(1, 2)
```

cho kết quả $(3, -1)$.

Khai báo hàm trong m-file

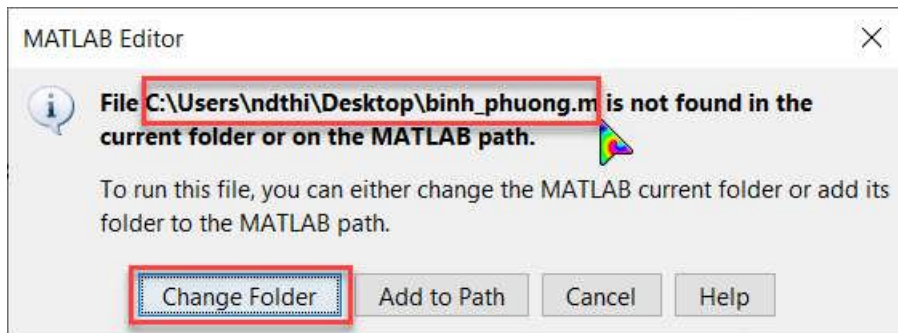
1. Vào *[Menu]HOME* → *New Script*, và lưu thành file *binh_phuong.m*, và soạn mã

```

1 function y = f(x)
2 y = x^2;
3 end

```

2. Nhấn *F5* để chạy file này. Hộp thoại sau yêu cầu thay đổi hoặc nạp đường dẫn của thư mục chứa file vào môi trường thực thi → chọn *Change Folder* hoặc *Add to Path*.



3. Khi đó, trong sổ tay, lệnh

```

1 binh_phuong(-3) % tên hàm trùng tên file

```

cho kết quả là 9.

Với cách này, mỗi tệp chỉ nạp được một hàm. Tệp *.m chứa nhiều hàm được viết mã kiểu hướng đối tượng. Tên lớp trùng với tên tệp. Chẳng hạn, với tệp *my_functions.m* có mã

```

1 classdef my_functions
2     methods(Static)
3         function z = cong(x, y)
4             z = x + y;
5         end
6
7         function z = tru(x, y)
8             z = x - y;
9         end
10    end

```

Khi đó, trong sổ tay, lệnh

```

1 my_functions.cong(1, 2)

```

cho kết quả là 3.

1.5.7 MATLAB: tính toán trong giải tích

Biến, biểu thức symbolic

Đặc điểm nổi bật của các ngôn ngữ lập trình tính toán là khả năng làm việc với biến và biểu thức **symbolic**. Không như biến thông thường trong các ngôn ngữ lập trình khác, biến symbolic không có giá trị cụ thể, ta có thể gọi là biến bất định. Biểu thức toán học chứa biến symbolic gọi là biểu thức symbolic. Để làm việc với biểu thức symbolic, trước hết cần khai báo các biến symbolic có trong biểu thức đó.

- Khai báo một hoặc vài biến cụ thể

Cách 1:

```
1 syms x
2 syms x y % các biến cách nhau dấu cách
```

Cách 2: Hiển thị bằng tên hay ký hiệu khác

```
1 l = sym('lambda') % λ
```

- Khai báo vectơ hàng x chứa các biến symbolic $x = (x_1, x_2, x_3)$

```
1 x = sym('x', [1, 3]) % 1 hàng, 3 cột
```

- Khai báo một ma trận symbolic A cỡ 2×3

```
1 A = sym('a', [2, 3]) % A =  $\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix}$ 
2 A(1, 2) %  $a_{1,2}$ 
3 A(5) %  $a_{1,3}$ , duyệt theo cột
```

- Khai báo hàm bất định f

```
1 syms f(x)
2 syms f(x) g(x, y) % tự động khai báo biến symbolic x, y
```

Xử lý biểu thức

MATLAB	Kết quả	Mô tả
<code>vpa(x/2, 6)</code>	$0.5x$	đưa các số trong biểu thức về số thập phân với 6 chữ số có nghĩa
<code>simplify(sin(x) * cos(x))</code>	$\frac{\sin 2x}{2}$	rút gọn
<code>factor(x^2 - 3*x + 2)</code>	$(x - 1)(x - 2)$	phân tích thành nhân tử
<code>expand((x + 1)^2)</code>	$x^2 + 2x + 1$	khai triển đa thức

Tính giá trị của biểu thức

MATLAB	Kết quả
<code>subs(x^3, x, 2)</code>	$2^3 = 8$
<code>subs(x^3, x, [1, 2])</code>	$(1, 8)$
<code>subs(x + y, {x, y}, {1, -2})</code>	$1 + (-2) = -1$
<code>subs(x + y, [x, y], [1, -2])</code>	
<code>subs(x*y, {x, y}, {[1, 2], [3, 4]})</code>	$(1 \times 3, 2 \times 4) = (3, 8)$

Khi biểu thức đã được khai báo bởi hàm, xem lại phần 1.5.6.

Đạo hàm

MATLAB	Kết quả
<code>diff(x^3)</code>	$\frac{\partial}{\partial x} (x^3) = 3x^2$
<code>diff(x^3, 2)</code>	$\frac{\partial^2}{\partial x^2} (x^3) = 6x$
<code>diff(x^3, x, 2)</code>	

Tích phân

Python	Kết quả
<code>int(x^2, 0, 1)</code>	$\int_0^1 x^2 dx = \frac{1}{3}$
<code>int(x^2, x, 0, 1)</code>	

Giải phương trình, hệ phương trình

Ví dụ 1.5. a) Phương trình $x^3 - 3x + 2 = 0$ có hai nghiệm là $x_1 = 1$ (nghiệm kép) và $x_2 = -2$.

```
1 solve( x^3 - 3*x + 2 == 0) %  $\begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}$ 
```

b) Hệ phương trình $\begin{cases} 3x - y^2 = 5 \\ 2x - y^2 = 2 \end{cases}$ có nghiệm $(x_1 = 3, y_1 = -2)$ và $(x_2 = 3, y_2 = 2)$.

```
1 sol = solve(3*x - y^2 == 5, 2*x - y^2 == 2)
```

cho thông báo

```
1 sol = struct with fields:
2   x: [2x1 sym]
3   y: [2x1 sym]
```

nói rằng hệ có hai nghiệm. Ta dùng tiếp lệnh

```
1 sol.x %  $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$ 
2 sol.y %  $\begin{bmatrix} -2 \\ 2 \end{bmatrix}$ 
```

để thu được nghiệm của hệ.

Giải phương trình vi phân

Ví dụ 1.6. Bài toán giá trị ban đầu $y' = y - x$, $y(0) = 2$ có nghiệm $y = e^x + x + 1$.

```
1 syms y(x)
2 dsolve(diff(y) == y - x , y(0) == 2)
```

Vẽ đồ thị

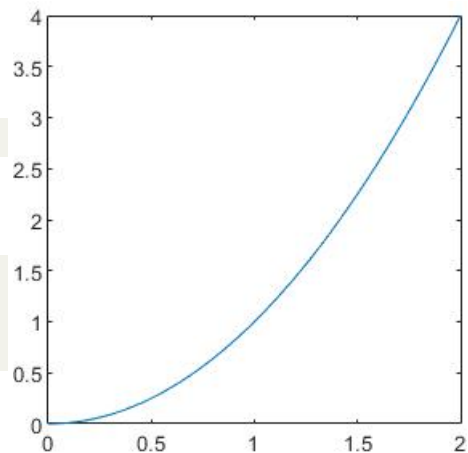
Ví dụ 1.7. Vẽ đồ thị hàm số $y = x^2$ trên đoạn $[0, 2]$.

Cách 1:

```
1 fplot(x^2, [0, 2])
```

Cách 2:

```
1 X = 0:0.1:2
2 Y = subs(x^2, X)
3 plot(X, Y)
```



1.5.8 MATLAB: tính toán trong đại số

Véc tơ

MATLAB	Kết quả	Mô tả
$x = [4, 1, -2]$		véc tơ hàng $x = (4, 1, -2)$
$x = [4; 1; -2]$		véc tơ cột $x = (4, 1, -2)^T = \begin{bmatrix} 4 \\ 1 \\ -2 \end{bmatrix}$

<code>length(x)</code>	3	số phần tử, hay độ dài, cỡ của x
<code>x(1)</code>	4	phần tử đầu của x , có chỉ số là 1
<code>x'</code>	x^T	chuyển vị của x
<code>x(:)</code>		véc tơ cột ứng với x
<hr/>		
<code>i = 1:5</code>		$i = (1, 2, 3, 4, 5)$
<code>1:2:10</code>		
<code>2*i - 1</code>	[1, 3, 5, 7, 9]	
<code>i.^2</code>	[1, 4, 9, 16, 25]	

Xếp hai véc tơ thành ma trận

```
1 X = [1, 2, 3]
2 Y = [4, 5, 6]
3 [X; Y] %  $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ 
```

và

```
1 X = [1; 2; 3]
2 Y = [4; 5; 6]
3 [X, Y] %  $\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$ 
```

Nếu véc tơ a độ dài n , gồm các phần tử a_1, a_2, \dots, a_n , thì với $1 \leq i \leq j \leq n$, lệnh

$$a(i:j)$$

là véc tơ gồm a_i, a_{i+1}, \dots, a_j , cùng kiểu hàng / cột với a .

Với hai véc tơ $x = (x_1, x_2, \dots, x_n)^T, y = (y_1, y_2, \dots, y_n)^T \in \mathbb{R}^n$

Phép toán	MATLAB	Ghi chú
$x + y = (x_1 + y_1, x_2 + y_2, \dots, x_n + y_n)^T$	<code>x + y</code>	
$x - y = (x_1 - y_1, x_2 - y_2, \dots, x_n - y_n)^T$	<code>x - y</code>	
$kx = (kx_1, kx_2, \dots, kx_n)^T$	<code>k * x</code>	$k \in \mathbb{R}$
$\langle x, y \rangle = x_1y_1 + x_2y_2 + \dots + x_ny_n$	<code>dot(x, y)</code>	tích vô hướng

Ma trận

Đối với ma trận $A = \begin{bmatrix} 3 & -1 & 2 \\ 0 & 4 & 1 \end{bmatrix}$:

MATLAB	Kết quả	Ghi chú
<code>A = [3, -1, 2; 0, 4, 1]</code>		kết thúc hàng bằng dấu ;

<code>size(A)</code>	$(2, 3)$	số hàng và số cột của A
<code>A(1, 2)</code>	-1	hàng 1 cột 2 của A
<code>length(A)</code>	3	số hàng của A
<code>A(5)</code>	2	phần tử thứ 5 của A , duyệt theo cột
<code>A'</code>	A^T	chuyển vị của A
<hr/>		
<code>eye(n)</code>	I_n	ma trận đơn vị cấp n
<code>zeros(m, n)</code>		ma trận không cỡ $m \times n$

Với hai ma trận A, B , và cỡ tương thích, tức là, cỡ phù hợp để thực hiện được phép toán

Phép toán	MATLAB	Ghi chú
$A + B$	<code>A + B</code>	
$A - B$	<code>A - B</code>	
kA	<code>k * A</code>	$k \in \mathbb{R}$
AB	<code>A*B</code>	
A^{-1}	<code>A^-1</code> hoặc <code>inv(A)</code>	ma trận nghịch đảo của A

Giải hệ phương trình tuyến tính

Ví dụ 1.8. Hệ phương trình

$$\begin{cases} 3x - y = 5 \\ 2x - y = 2 \end{cases}$$

ma trận

với nghiệm $x = 3, y = 4$, có thể viết dưới dạng

$$\begin{bmatrix} 3 & -1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

```
1 A = [3, -1; 2, -1]
2 b = [5; 2]
3 linsolve(A, b) %  $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$ 
```

1.5.9 Cấu trúc điều khiển

if: rẽ nhánh

Trường hợp 1:

```
1 if 1 < 2 % điều kiện
2     'đúng' % thực hiện lệnh nếu điều kiện đúng
3 end
```

Trường hợp 2:

```

1 if 1 < 2
2     'đúng'
3 else
4     'sai' % thực hiện nếu điều kiện sai
5 end

```

for: lặp xác định

Ví dụ 1.9. In ra các số nguyên từ 0 tới 9, mỗi số một dòng.

Giải.

```

1 for i = 0:9
2     i
3 end

```

□

Ví dụ 1.10. In ra các biểu thức x^5, x^6, \dots, x^{10} , mỗi biểu thức một dòng.

Giải.

```

1 syms x
2 for i = 5:10
3     x^i
4 end

```

□

while: lặp không xác định

Ví dụ 1.11. Viết đoạn mã trong [Ví dụ 1.9](#) dưới dạng vòng lặp không xác định.

Giải.

```

1 i = 0
2 while i < 10 % lặp khi điều kiện còn đúng
3     i = i + 1 % có chức năng thay đổi giá trị của điều kiện
4 end

```

□

Tóm tắt MATLAB

Dấu . , ; : ([' %

Từ khóa	classdef	function	else	end	Static
	for	if	elseif	methods	while

Hàm / lệnh	abs	dot	int	plot	subs
	acos	dsolve	inv	round	sum
	acot	exp	length	simplify	sym
	asin	eye	linsolve	sin	syms
	atan	expand	log	size	tan
	cos	factor	max	simplify	vpa
	cot	factorial	min	solve	zeros
	diff	fplot	pi	sqrt	

Tài liệu tham khảo

- [1] Phạm Kỳ Anh. *Giải tích số*. Đại học Quốc gia Hà Nội, 2002. 284 trang.
- [2] Richard L. Burden, Douglas J. Faires **and** Annette M. Burden. *Numerical Analysis*. phiên bản 10. Cengage Learning, 2016. 918 trang.
- [3] Phan Văn Hạp **and** Lê Đình Thịnh. *Phương pháp tính và các thuật toán*. Nhà xuất bản Giáo dục, 2000. 400 trang.
- [4] Doãn Tam Hòe. *Toán học tính toán*. Đại học Quốc gia Hà Nội, 2009. 240 trang.

