

Mục lục

I	Cơ bản về MATLAB	1
1	Không gian làm việc	1
a	Giao diện chính	1
b	Các thao tác cơ bản	2
2	Biến, hằng số, hàm cơ bản	3
3	Quản lý tập tin, M-file	5
a	Tạo thư mục lưu trữ	5
b	Scripts (Đoạn mã lệnh)	5
c	Comment	6
II	Các thao tác và phép toán với mảng	6
III	Vòng lặp điều khiển	6
IV	Đồ thị trong mặt phẳng và trong không gian	6

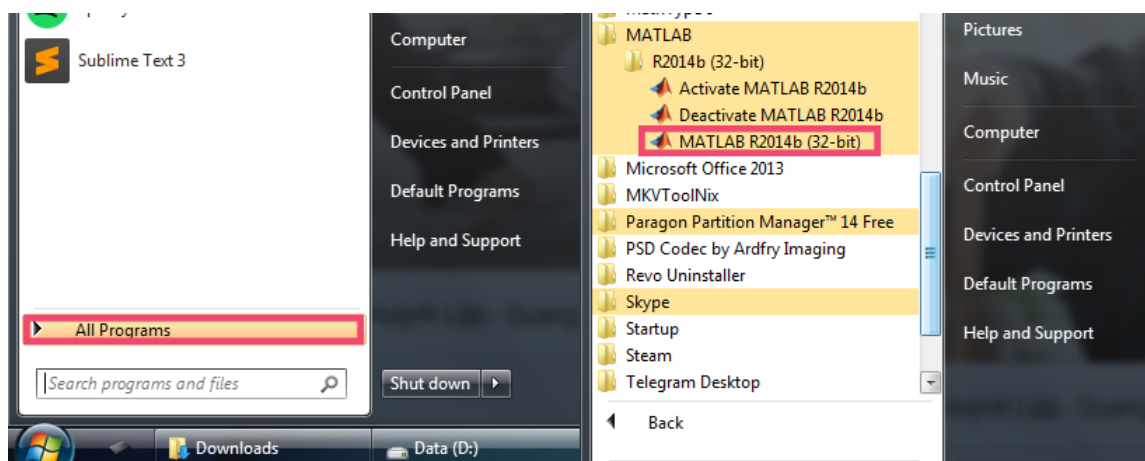
I. Cơ bản về MATLAB

1. Không gian làm việc

a. Giao diện chính

Có 3 cách khởi động chương trình Matlab từ máy tính chạy Windows:

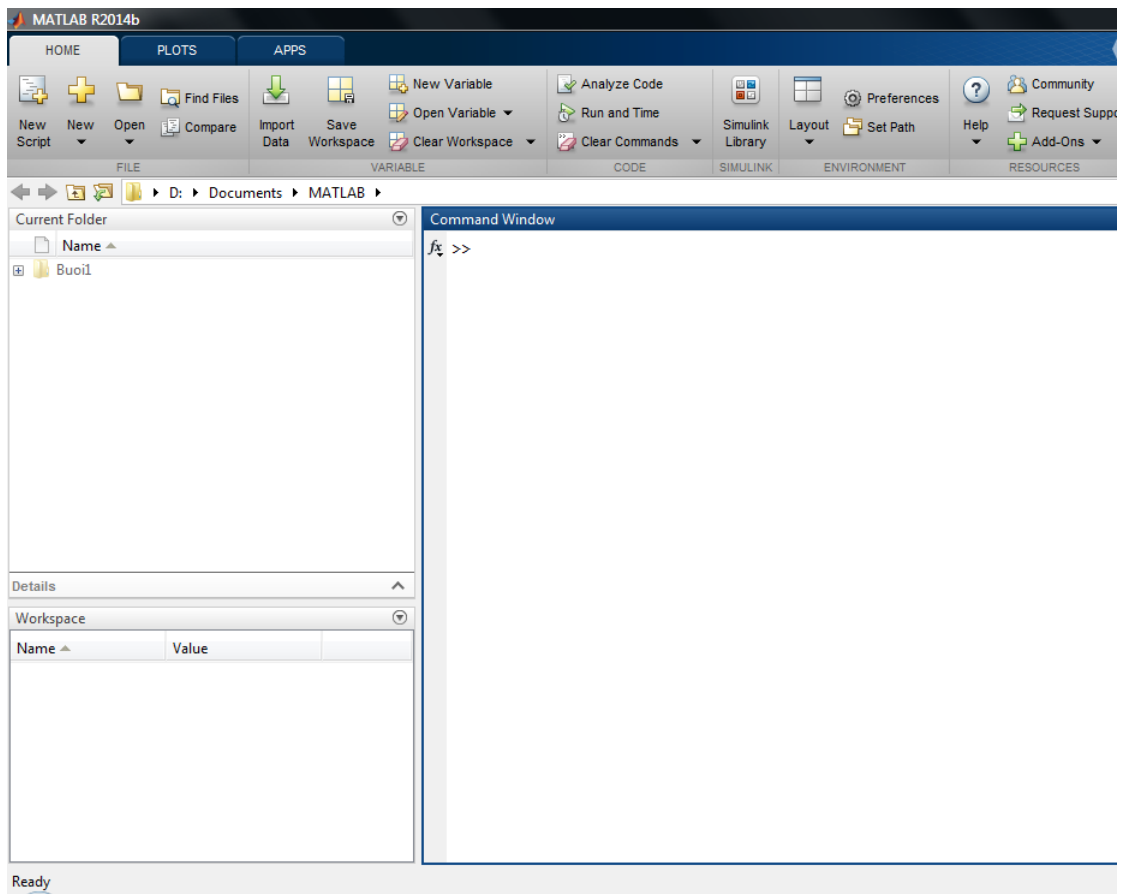
- Khởi động từ biểu tượng ngoài màn hình.
- Mở trực tiếp tập tin có đuôi mở rộng là .m, ví dụ: Bai1.m hoặc Hamso.m
- Vào biểu tượng Start > All Program > MATLAB > Matlab 2014b.



Hình 1: Thao tác mở chương trình từ Start Menu.

Giao diện chính của chương trình sau khi khởi động sẽ tương tự như hình bên dưới. Có 3 khu vực làm việc chính. Khung lớn nhất được bao viền màu xanh dương là Command Window. Khung Current Folder nằm ở phần trên, bên trái của Command Window. Phía dưới khung

Current Folder là Khung Workspace.



Hình 2: Giao diện chính của chương trình

Trong đó:

- **Command Window:** Khu vực người dùng nhập các câu lệnh điều khiển và tính toán. Mỗi câu lệnh sẽ được thực hiện khi người dùng nhấn Enter.
- **Current Folder:** Khu vực truy cập các tập tin m-file, m-hàm... được lưu trữ trên ổ đĩa máy tính.
- **Workspace:** Liệt kê các biến và dữ liệu đang được xử lý tạm thời. Ngoài ra người dùng có thể thao tác thêm bớt chỉnh sửa các giá trị của biến và tham số trực tiếp trên Workspace mà không cần dùng đến câu lệnh.

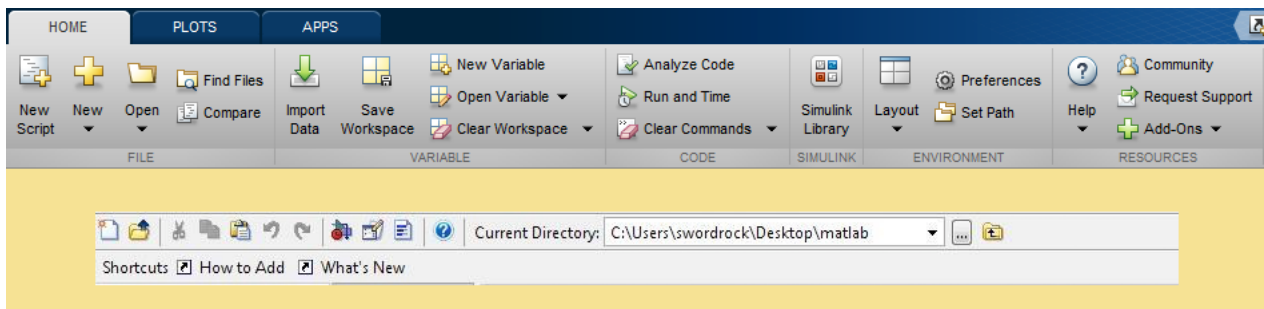
Ngoài ba khu vực chính trên, kể từ phiên bản MATLAB 2012a, thanh công cụ phía trên cùng truyền thống đã được thay thế bằng các thẻ lệnh Ribbon với biểu tượng cụ thể và rõ ràng hơn.

b. Các thao tác cơ bản

MATLAB thực hiện các phép cộng, trừ, nhân, chia như một chiếc máy tính bình thường. Xét các ví dụ đơn giản sau:

Ví dụ 1: Tính $3 + 2 + 6$; $4 \times 6 \times 25 + 32 - 9 : 3$.

1 >> 3 + 2 + 6 =



Hình 3: Thẻ Home (hình trên) ở giao diện mới và thanh công cụ truyền thống (hình dưới)

```

2  ans =
3      12
4  >> 4 * 6 * 25 + 32 - 9 / 3 =
5  ans =
6      629

```

Lưu ý: MATLAB không quan tâm đến khoảng trắng giữa các dấu và luôn ưu tiên phép nhân rồi mới đến phép cộng. Kí hiệu `ans` là viết tắt của từ "answer" có nghĩa là kết quả của phép tính.

Ví dụ 2: Tính $\sin\frac{\pi}{2} + \cos\frac{\pi}{2}$

```

1  >> sin(\pi/2)+cos(\pi/2)=
2  ans =
3      1

```

2. Biến, hằng số, hàm cơ bản

Trong MATLAB, ta có thể gán giá trị cho biến với tên gọi bất kỳ, điều này giúp cho việc tính toán rõ ràng và dễ dàng hơn. Đặt biệt với những bài tính toán với các biến số, ta dễ dàng thay đổi giá trị của biến số để có được các kết quả phù hợp với yêu cầu tính toán.

Yêu cầu khi đặt tên biến: Tên biến phải bắt đầu bằng chữ cái, các ký tự sau ký tự đầu tiên có thể là số, chữ hoặc ký tự "_". Ngoài ra, MATLAB phân biệt ký tự hoa sẽ khác với ký tự thường. Tức là biến "var1" sẽ khác với biến "Var1".

Một số biến đã được định nghĩa sẵn (một số còn được gọi là hằng số):

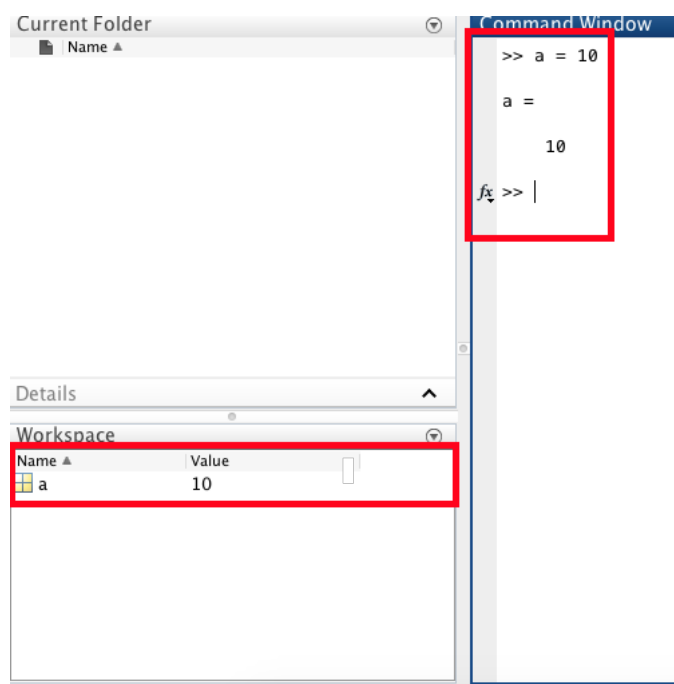
- Ký tự `i` và `j` được dùng cho đơn vị phức.
- Biến `pi` dùng để chỉ số pi (π), với giá trị là 3.1415926...
- Biến `ans` dùng để lưu kết quả vừa tính toán trước đó.
- Biến `Inf` và `-Inf` (lưu ý ký tự `I` đứng đầu được viết hoa) dùng để biểu diễn dương và âm vô cực.
- Ký tự `NaN` thể hiện cho "Not a Number", không phải là một số.

Biến thông thường: (hay còn được gọi là biến vô hướng) là biến được người dùng định nghĩa bằng cách gán một giá trị cụ thể nào đó trực tiếp thông qua câu lệnh trong Command Windows.

Ví dụ 1: Tạo biến tên `a` chứa giá trị là số 10. Ta sẽ nhập lệnh vào Command Windows như sau:

```
1 >> a = 10
2 a =
3     10
```

Ngay lập tức biến tên a sẽ được liệt kê trong khu vực Workspace. Người dùng có thể truy cập hoặc điều chỉnh giá trị biến ngay trong cửa sổ Workspace.



Hình 4: Tạo biến ở khu vực Command Window và quản lý biến tại khu vực Workspace.

Ví dụ 2: Tạo biến c dựa vào biến a có sẵn biết rằng c được tính qua biểu thức $c = a^2 + a^3$. Ta tiếp tục nhập lệnh sau vào khu vực Command Window từ ví dụ trước đó:

```
1 >> c = a^2 + a^3
2 c =
3     1100
```

Qua ví dụ 2 ta có thể tạo biến dựa vào giá trị của một biến khác cho trước. Ngoài ra, ta có thể ẩn đi kết quả tính toán của câu lệnh bằng cách thêm dấu ";" vào cuối câu lệnh. Ví dụ sau sẽ giúp ta dễ hình dung:

```
1 >> b = 6 + c;
2 >> b = 6 + c
3 b =
4     1106
```

3. Quản lý tập tin, M-file

a. Tạo thư mục lưu trữ

Trong MATLAB, ta nên tập thói quen dùng các thư mục được sắp xếp gọn gàng để phân loại dữ liệu đang xử lý. Những thao tác cơ bản gồm:

- Để tạo thư mục mới, chọn nút "Browse for folder" ở gần khu vực Current Folder. Sau đó ta tạo thư mục mới trong cửa sổ vừa hiện ra, đặt tên cho thư mục (lưu ý rằng tên thư mục không được phép chứa ký tự khoảng trắng).
- Chọn thư mục vừa tạo và nhấn nút "Open" để mở.
- Lúc này, thư mục hiện hành chính là thư mục bạn vừa mới tạo.
- Không chỉ cho phép người dùng tạo thư mục thông qua cửa sổ "Browse for folder", người dùng còn có thể xoá, sửa hay di chuyển thư mục trong cửa sổ đó.

Nâng cao: Với những người dùng thường xuyên lưu trữ chương trình hoặc các tập tin m-file ở các thư mục nằm nhiều nơi trên ổ đĩa, người dùng có thể thêm đường dẫn thư mục đó vào cửa sổ **Set Path (trong phần Enviroment)** để có thể chạy lệnh tính toán mà không cần quan tâm vị trí lưu trữ thư mục chứa các tập tin đó.

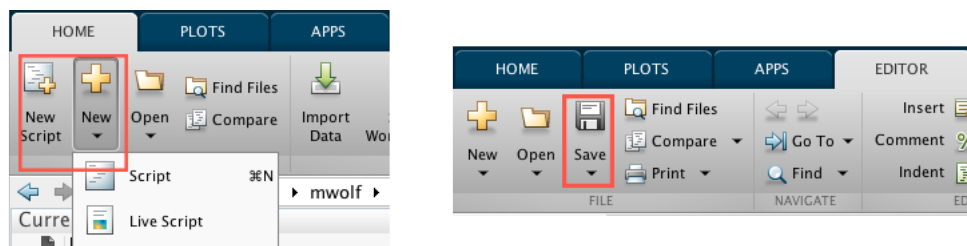
b. Scripts (Đoạn mã lệnh)

Scripts (còn được gọi là đoạn mã lệnh) là một tập tin chứa các câu lệnh chạy trong môi trường MATLAB. Các câu lệnh được lưu trữ trong tập tin này sẽ được thực thi theo trình tự.

Scripts được viết trong cửa sổ MATLAB Editor tích hợp sẵn trong MATLAB và được lưu trên ổ đĩa dưới dạng các tập tin có đuôi mở rộng là .m, dung lượng thường không quá 100MB. Ngoài ra, ta hoàn toàn có thể dùng một công cụ soạn thảo cơ bản như **Notepad** mặc định của Windows để tạo ra một tập tin đuôi .m nhưng yêu cầu các câu lệnh trong đó phải đúng và thực thi được trong môi trường MATLAB.

Các cách tạo một tập tin MATLAB (đuôi .m):

- Từ Command Windows gõ câu lệnh với cú pháp `edit tenfile.m`, trong đó ta thay thế `tenfile.m` bằng tên tập tin muốn tạo. Sau đó chọn OK trong cửa sổ xác nhận tạo tập tin tenfile.m.
- Chọn vào biểu tượng **New Script** trong khu vực **File** hoặc chọn **File >> New >> Script** đều được. Sau khi cửa sổ soạn thảo đã mở, ta chọn biểu tượng **Save** trong khu vực File và tiến hành đặt tên cho tập tin cùng với việc chọn khu vực lưu trữ.



Hình 5: Khu vực tạo Script mới (hình bên trái) và khu vực lưu Script (hình bên phải).

- Ngoài hai cách trên, ta còn có thể dùng tổ hợp phím **ctrl + N** (đối với Windows) hoặc **⌘ + N** (đối với MacOS) để mở cửa sổ MATLAB Editor và tạo tập tin mới.

Lưu ý: Nội dung ban đầu của script thường là những câu lệnh, nhưng nếu trong script có yêu cầu input (nhập dữ liệu vào) và có lệnh output (xuất dữ liệu ra) thì nó sẽ trở thành một m-hàm.

c. Comment

Trong lúc soạn thảo các câu lệnh ở cửa sổ MATLAB Editor, người dùng có thể thêm những dòng chú thích vào dòng nào đó, qua đó giúp người dùng có thể đánh dấu, ghi nhớ hoặc tra cứu lại ý nghĩa của đoạn mã mà họ sử dụng. Các phần chú thích đó còn được gọi là những Comment (bình luận) và có hai cách thực hiện tạo Comment:

- Đối với nội dung ghi chú ngắn, có thể chỉ 1 dòng thì chỉ cần thêm ký tự "%" vào trước câu ghi chú đó là được. Ví dụ: `%Day la cau ghi chu.` Thường những câu ghi chú sẽ được quy định chữ màu xanh lá để giúp người dùng dễ dàng phân biệt.
- Đối với những phần ghi chú có nhiều dòng, nội dung dài. Người dùng sẽ dùng cú pháp như dưới đây:

```
1  %{
2  Noi dung ghi chu, dong 1.
3  Noi dung ghi chu, dong 2.
4  ...
5  Noi dung ghi chu, dong n.
6  %}
```

II. Các thao tác và phép toán với mảng

III. Vòng lặp điều khiển

IV. Đồ thị trong mặt phẳng và trong không gian

Tiểu luận môn học Phần mềm Toán học

“Một số kiến thức về phần mềm MATLAB”

Chương 1: Cơ bản về MATLAB

1.1 Không gian làm việc

1.2 Biến, câu giải thích, chấm câu

1.3 Các hằng số, các phép toán cơ bản, các hàm toán học.

1.4 Quản lý tập, M-file, M-hàm

(C1 → C4)

Chương 2: Các thao tác và phép toán với mảng

(C6 và C7)

Chương 3: Vòng lặp điều khiển (C11)

Chương 4: Đồ thị trong mặt phẳng và trong không gian

(Bổ sung nhiều ví dụ và bài tập)

```
1 // Hello.java
2 for (i:1);
3 import javax.swing.JApplet;
4 import java.awt.Graphics;
5
6 public class Hello extends JApplet {
7     public void paintComponent(Graphics g) {
8         g.drawString("Hello, world!", 65, 95);
9     }
10 }
```
