Chapter 1

Presentation of computing programming environment: MATLAB

1. Introduction

Nowadays, using programming language to solve scientific problems is indispensable for engineers and those working in many fields such as medicine, biology, finance and others.

The main objective of this chapter is to introduce the basic programming concepts of one of the most used computing programming environment for scientists and engineers, which is MATLAB.

2. Computing programming environment

A "computing programming environment" can be defined as software and tools used to write, test, and debug computer programs. It offers an integrated set of tools and features that make the software development process more efficient and manageable. Several computing programming environments exist, and the choice of an environment often depends on the programming language and the specific needs of the project.

3. MATLAB Programming Environment

MATLAB (an abbreviation of "MATrix LABoratory") is an interactive computing environment developed by **MathWorks**, which provides a technical computing environment designed to enable numerical computation and data visualization. MATLAB can be used to solve problems ranging from the very simple to the sophisticated and complex problems. MATLAB has proven its ability to facilitate

solving problems in many fields such as applied mathematics, physics, chemistry, engineering and any application dealing with complex numerical calculations.

In addition, MATLAB has optional toolbox allowing access to symbolic computing abilities and an additional package, Simulink, adds graphical multi domain simulation and for dynamic and embedded systems.

MATLAB was implemented by mathematician and computer programmer **Cleve Moler** in 1970 and was first released as a commercial product in 1984 at the Automatic Control Conference in Las Vegas.

4. Octave programming environment

Gnu Octave is a free scientific programming software developed by John W. Eaton in collaboration with others and licensed under GNU General public License (GPL). Octave is considered as a powerful mathematics oriented syntax, which is largely compatible with MATLAB. In addition, Octave provides extensive powerful tools for solving common problems including numerical linear algebra, nonlinear equations, manipulating polynomials and many others. It runs on GNU/Linux, macOS, BSD and Windows and any one can contribute to its development by writing additional functions for it or reporting any encountered problem.

5. Scilab Programming Environment

Scilab is a free, open-source and high-level, numerically oriented programming language. Scilab can be used for signal processing, statistical analysis, image enhancement, fluid dynamics simulations and numerical optimization. The Scilab language provides an interpreted programming environment, with matrices as the main data type.

Scilab also includes a free open source package called **Xcos** for modeling and simulation of dynamical systems, including both continuous and discrete sub-systems. Xcos is equivalent to Simulink package of MATLAB. As the syntax of Scilab is similar to MATLAB, it includes a translator for supporting the conversion of code from MATLAB to Scilab.

6. Alternatives numerically oriented programming language

MATLAB has some alternative programming languages, which are showcased in what follows.

Commercial Alternatives

AMESim, GAUSS, IDL, Maple, Mathcad, Mathematica, OxMetrics, PyIMSL Studio, SAS/IML, Stata (Mata), Sysquake

Open alternatives

Octave, Scilab, FreeMat, JMathLib, R, SageMath, SciPy

7. MATLAB online version

Mathworks provides an online version of MATLAB to researchers interested in using MATLAB, Simulink and toolboxes anywhere through web browser without any downloads and installation. Two versions of access are provided, in which the basic version offers 20 hours of use per month and allows the access to 10 commonly used products. The second way is the licensed version, which provides a full access to MATLAB online in addition to desktop version.

8. Getting started with MATLAB

When MATLAB, Octave or Scilab software is started, a window opens. The main part is the Command Window, where the user can type his commands interpreted later by the software to provide directly the solution.

The main parts of MATLAB graphical users are indicated in Figure 1.

- Current folder (File browser): This section in the MATLAB desktop environment displays the current, the saved and the performed working directory;
- Workspace: This part refers to the environment where we create, store, and manage variables and data during MATLAB session. The Workspace provides a visual representation of the variables we have defined and their values;
- Command window: MATLAB's Command Window is where we can enter commands and see their results. It is the primary interface for interacting with MATLAB;
- Command history: It is a feature that allows us to view and access the history of commands we have entered during our current session

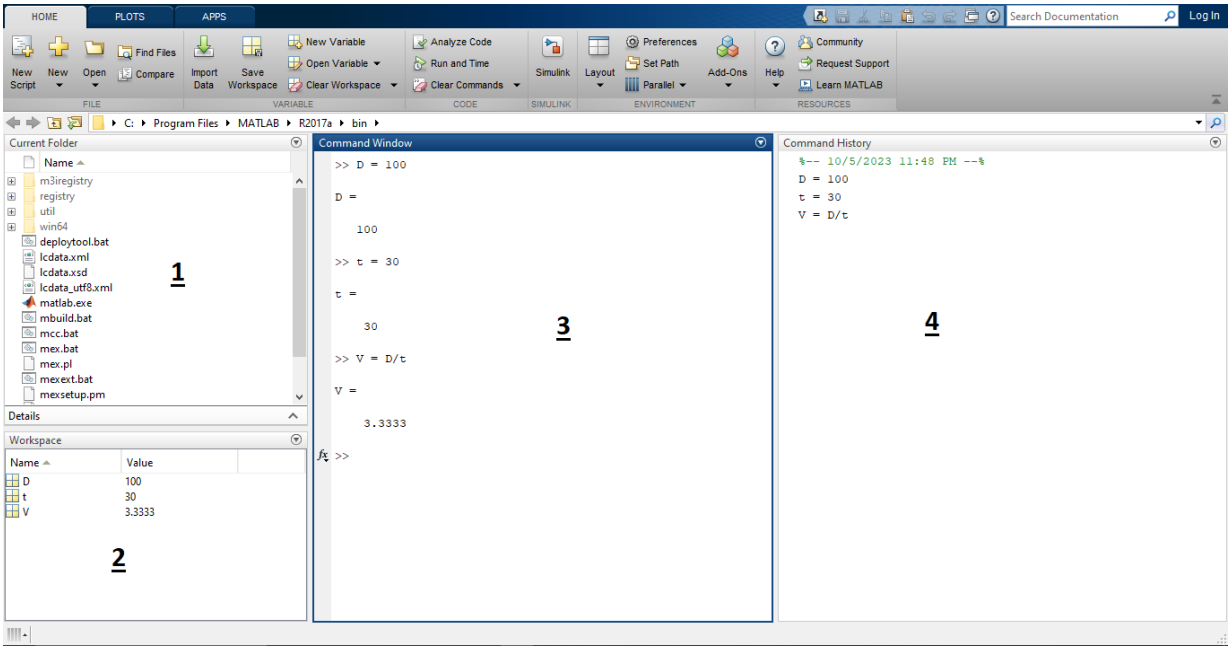

Figure.1 MATLAB User interface: 1. Current folder (File browser), 2. Work space, 3. Command window, 4. Command history.

9. Basic arithmetic

Matlab uses the standard arithmetic operators, addition, subtruction, multiplication, division and power as indicated bellow where the text before the '#' key is considered as a commant.

```
>> % Addition
>> 5+3

ans =

     8

>> % Subtraction
>> 5-3

ans =

     2
```

```
>> % Multiplication
>> 5*3

ans =

    15

>> % Division
>> 5/3

ans =

    1.6667

>> 5\3

ans =

    0.6000

>> % Puissance
>> 5^3

ans =

   125
```

10. Priority of commands

The arithmetic operators have the following priority:

1. **Brackets** first;
2. **Power** next;
3. **Multiplication** and **division** next. **Left to right** in case of competition;
4. **Addition** and **subtraction** next. **Left to right** in case of competition.

Next command lines show an example of such priority.

```
>> 1 + 4 / 2 * 3 ^ 2

ans =

    19

>> (1 + 4 / 2 * 3) ^ 2

ans =

    49

>> (1 + 4) / 2 * 3 ^ 2

ans =

   22.5000

>> (1 + 4) / (2 * 3) ^ 2

ans =

    0.1389

>> (1 + 4 / 2) * 3 ^ 2

ans =

    27
```

11. Variables

Variables in MATLAB are defined by an identifier name, which must respect the following rules:

1. The variable name must begin with a letter and can contain letters, digits and underscore character (_);
2. MATLAB is case sensitive. So, variable named **Var_1** is different from variable named **var_1**;
3. MATLAB has reserved keywords, which cannot be used as an identifier name (**sqrt, exp**, …).

Next window illustrates some operations using variables.

```
>> A = 5

A =

     5

>> Var_1 = A + 10

Var_1 =

    15

>> Var_1 = Var_1 * A

Var_1 =

    75

>> var_1
Unrecognized function or variable 'var_1'.

Did you mean:
>> Var_1
```

12. Some reserved keywords and commands

MATLAB has a list of reserved words and commands. Some of such keywords are indicated in what follow.

```
>> A = 4;
>> % sin, cos, arcsin ...
>> B = sin(A)

B =

   -0.7568

>> % Root: sqrt, Exponential: exp, ...
>> sqrt(A)

ans =

     2

>> % Constants: pi, i, j, inf, NaN
>> pi

ans =

    3.1416

>> i

ans =

   0.0000 + 1.0000i

>> % who and whos commands
>> who
```

```
Your variables are:

A      B      ans

>> whos
  Name      Size            Bytes  Class     Attributes

  A         1x1                 8  double
  B         1x1                 8  double
  ans       1x1                16  double    complex
```

13. Reusing previous commands

To rerun previous commands do one of the following tasks:

- Press the up arrow key (↑) until the command you want appears at the prompt, and then press **Enter;**
- Double-click an entry or entries in the Command History window, or select an entry and press **Enter**;
- To extend the selection to include multiple commands, press **Shift**+↑.

14. Clearing screen and variables and quitting MATLAB

To remove items from workspace and freeing up system memory, the following commands are used

`clc` : clear command window.

`clear`: removes all variables from the workspace.

`clear VARIABLES`: does the same thing as the previous command.

`clear GLOBAL`: removes all global variables.

`clear all`: removes all variables

`clear` VAR1 VAR2: clears the specified variables.

To close MATLAB using the command window we type `quit` or `exit`.

15. Practical work 1

1. Using variables, calculate the following functions.

$$a = 3,\ \ b = 5,\ \ c = 0.4, \alpha = 0.6\ , \beta = 2\pi/10$$

$$eq_1 = a + b,\ \ eq_2 = a - b,\ \ eq_3 = b \times c, eq_4 = a/c$$

$$eq_5 = \frac{a}{5 - b}$$

$$eq_6 = \frac{a + 20}{b}$$

$$eq_7 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$eq_8 = \frac{eq_1 \times e^{-3ab+50}}{5}$$

$$eq_9 = e^{\sigma+\sqrt{a-1}} + \frac{\sin\beta}{\sqrt{|-c \times b| - 1}}$$

2. Delete some variables and clear the command window.
3. Close MATLAB using the command window.