MATLAB Marina: Introduction to MATLAB

Student Learning Objectives

After completing this module, one should:

1. Be able to identify and explain the use of the main portions of the MATLAB IDE.
2. Be able to use the MATLAB Command Window to perform numeric computations.

Terms

MATLAB, IDE, main desktop, command window, command history, workspace, current folder, search path, variable

MATLAB Functions, Keywords, and Operators

help, =, ans, clear, clc, close all, who, whos

MATLAB

MATLAB refers to both the high-level programming language and the development environment. When executing statements or programs in MATLAB, each statement is interpreted and executed one at a time when encountered. This allows problem solutions to be developed quickly compared to a compiled program. Compiled programs have been converted to machine code for a particular hardware platform and the compiled program can be directly executed on the platform. Most C++ programs are compiled rather than interpreted. A compiled program typically runs more efficiently than an interpreted program. MATLAB code can also be compiled for stand-alone programs but generally MATLAB code is executed using the MATLAB environment/interpreter.

MATLAB release and feature information can be found on the MathWorks site
MATLAB Release Information

MATLAB IDE

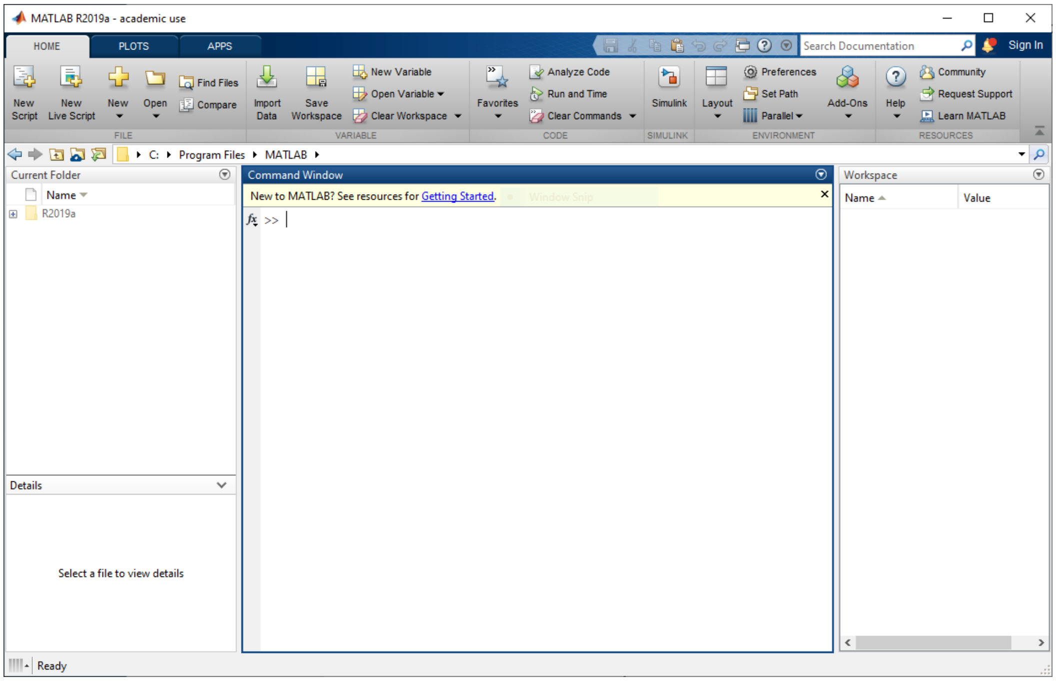
MATLAB can be launched by double clicking on the MATLAB icon on the desktop or choosing MATLAB from the Windows Start menu.

The MATLAB IDE (integrated development environment) has a main desktop with several panes. There are three main panes in the default MATLAB main desktop: the Command Window (center), Workspace (right), and Current Folder (left), see Figure 1a. Additional windows for plotting and creating scripts can be opened and panes/windows can be docked and undocked from the MATLAB main desktop. Across the top of the main desktop starting at the left side is the MATLAB Toolstrip. The Toolstrip has three tabs (HOME, PLOT, APPS) which contain icons for commands and action menus. The top right of the main desktop has the Quick Access Toolbar and the Search Bar. The Current Folder Bar is on the left of the main desktop just below the Toolstrip.

MATLAB has a built-in editor, see Figure 1b, which can be used to create and run MATLAB scripts (programs). The editor window is docked as the top center pane in the default layout.

MATLAB scripts are sequences of statements in a text file that are executed one statement at a time.

The default layout can be restored by selecting Default from the Layout Toolstrip icon.

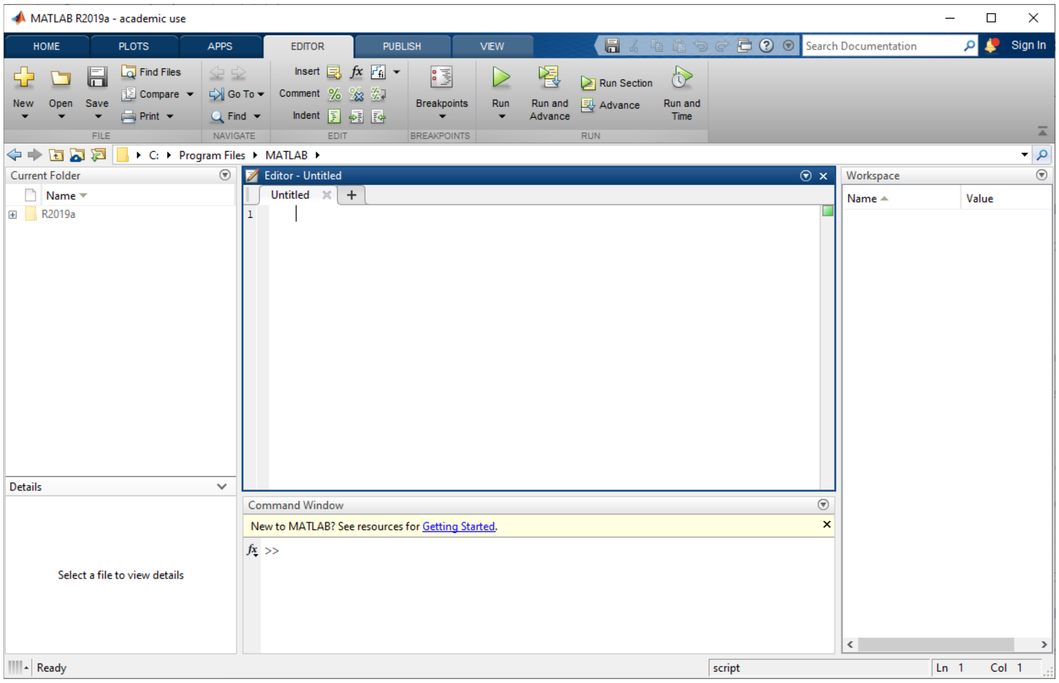


New to MATLAB? See resources for Getting Started.

```
>>
```

Select a file to view details

Figure 1a. MATLAB Main Desktop (R2019a Default Layout)



Select a file to view details

New to MATLAB? See resources for Getting Started.

```
>>
```

Figure 1b. MATLAB Editor (Docked)

MATLAB Current Folder and Workspace

The Current Folder pane shows the default folder/directory that MATLAB saves files to. The pane shows the files in the folder and allows new folders and files to be created.

The Workspace pane shows all the variables that have been defined and their values. Variables can be selected by double clicking on them which will open the Variable Editor. The Variable Editor can be used to inspect or modify the contents of a variable.

MATLAB Command Window

MATLAB statements and programs can be executed in the Command Window pane. MATLAB can be used like a calculator if commands are typed and executed in the Command Window. The basic arithmetic operations addition, subtraction, multiplication, division, and power are indicated by the symbols +, -, *, /, and ^ respectively. You have the option of using parenthesis () to indicate order of precedence. Recall, expressions are evaluated from left to right. Powers have the highest precedence, followed by multiplication and division, followed by addition and subtraction. Spaces may be used to separate things in statements to make the statement easier to read. Spaces in statements are generally ignored by MATLAB. After the enter key is hit, MATLAB returns the result of the executed statement.

Figure 2a shows examples of evaluating the expressions $\left(8+2\times\left(-1.5\right)\right)\div 4$ and $\left(81^{0.5}+2\times\left(-3\right)\right)/6$ in the Command Window (>> is the Command Window line prompt).

```
>> (8 + (2 * -1.5))/4
ans = 1.2500
>> result = (81^0.5 + (2 * -3))/6
result = 0.5000
```

Figure 2a. Evaluating Expressions using MATLAB Command Window

Variables are names referring to data stored in a computer’s memory. The equal sign (=) allows one to assign the result of the right-hand side of a statement to a variable on the left-hand side of the statement. The right-hand side of a statement is evaluated according to operator precedence and the resulting value is assigned to the variable on the left-hand side. The left-hand side of the assignment operator (=) can only have a single variable. If no variable is given for the result, MATLAB places the result in the default variable `ans`. The result of a statement is also displayed in the Command Window. The result displayed in the command window can be suppressed by ending commands with a semicolon. Once a variable has a value, it can then be used in other statements. Variables can be redefined by assigning a new value(s) to them. Figure 2b shows an example of assigning values to two variables and then using the variables in a statement.

```
>> len = 5.0;
>> ht = 2.7;
>> rectangleArea = len * ht
rectangleArea = 13.5000
```

Figure 2b. Use of Variables in Statements

Variables can be cleared from the workspace using the `clear` statement. Individual variables are deleted by `clear` followed by the variable name. The entire workspace (all variables) is deleted by using `clear` with no argument. The Command Window can be cleared and the cursor homed using the `clc` statement. The `clc` statement cleans up the Command Window but it does not delete the workspace variables.

The MATLAB statement `who` returns a list of all currently defined variables. The statement `whos` returns a list of all currently defined variables, variable sizes, numbers of bytes used, and class. The currently defined variables and their values can also be seen in the Workspace pane. The MATLAB workspace (all currently defined variables) can be saved to a data file using the `save` function. The statement `save data1;` will save the workspace to the file `data1.mat`. This file is a binary file that can be loaded into MATLAB using the `load` function.

```
>> a = 10;
>> b = 5;
>> c = 7;
>> who
Your variables are:  a  b  c
>> clear b
>> whos
  Name      Size           Bytes      Class     Attributes
  a         1x1            8          double
  c         1x1            8          double
>> clear all
>> who
>>
```

Figure 2c. Use of clear, who, and whos

MATLAB Pop Up Command History

The command history shows statements that have been executed in the Command Window in chronological order. The command history is accessed via a pop-up window that is accessed by hitting the up arrow in the Command Window.

Previously executed statements can be accessed from the pop-up command history using the up arrow ↑ key and the down arrow ↓ key (this allows one to scroll though past statements) or the mouse. The statement can then be edited or executed in the Command Window.

MATLAB Functions

Functions are program modules that are passed arguments and return a result. MATLAB has built in functions for most mathematical, engineering, and science functions that are commonly needed. MATLAB functions are called by their name. The variables or data inside the parenthesis of the call are the arguments for the function and the result from a function is saved by assigning it to a variable. Pay attention to the domain, the range, and the units of the MATLAB functions. For instance, the built-in cosine function, `cos`, takes an angle in radians. MATLAB also has a built-in function, `cosd`, that takes an angle in degrees.

Figure 3 shows an example of using the MATLAB square root and cosine functions to evaluate the law of cosines. Notice that functions can be called from within statements and the results of functions can either be immediately used or saved in a variable.

```
>> sideA = 4;
>> sideB = 6;
>> thetaC = pi/3;
>> sideC = sqrt(sideA^2+sideB^2-2*sideA*sideB*cos(thetaC))
sideC = 5.2915
```

Figure 3. Use of MATLAB Square Root and Cosine Functions

Some commonly used MATLAB functions are:

Function Name	**Operation**
`cos, acos`	cosine, inverse cosine
`sin, asin`	sine, inverse sine
`tan, atan`	tangent, inverse tangent
`log, log10, log2`	natural log, common log (base 10), base 2 log
`exp`	exponential function
`sqrt`	Square root
`abs`	absolute value (magnitude of complex number)
`angle`	phase angle
`real`	real part of a complex number
`imag`	imaginary part of a complex number
`conj`	complex conjugate

Table 1. Some Commonly Used Built in MATLAB Functions

Last modified Friday, September 18, 2020