

MATLAB Marina: MATLAB Editor and Scripts

Student Learning Objectives

After completing this module, one should:

1. Be able to use the MATLAB IDE and editor to create and execute scripts.

Terms

editor, variable, script, program, comment, syntax highlighting, code section

MATLAB Functions, Keywords, and Operators

=, clear, clc

MATLAB Editor

The MATLAB editor is a text editor with syntax highlighting for the MATLAB programming language. The MATLAB editor can either be docked in the main MATLAB IDE window (Figure 1a) or undocked (Figure 1b).

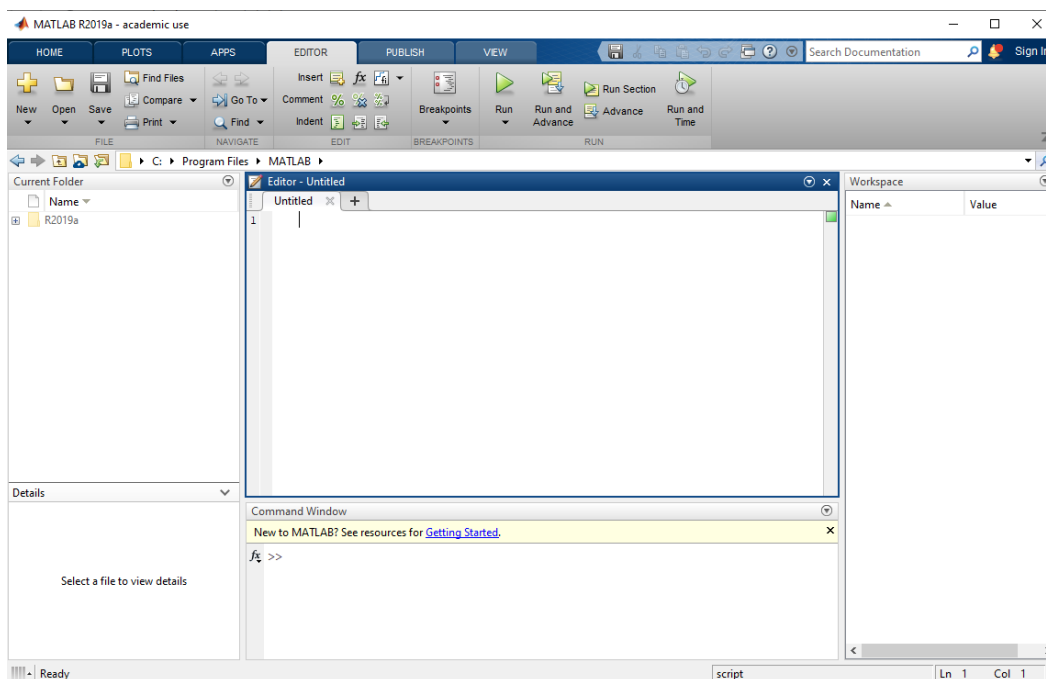


Figure 1a. MATLAB Editor (Docked)

If MATLAB is launched using an existing script file, the editor will open with the script in it. The MATLAB editor can also be opened using the New Script icon on the HOME tab of the Toolstrip or by selecting Script from the New menu on the HOME tab of the Toolstrip. The script can be saved from the Editor Toolstrip. The script will be saved in the current folder unless a different is specified. Multiple editor windows can be open at the same time.

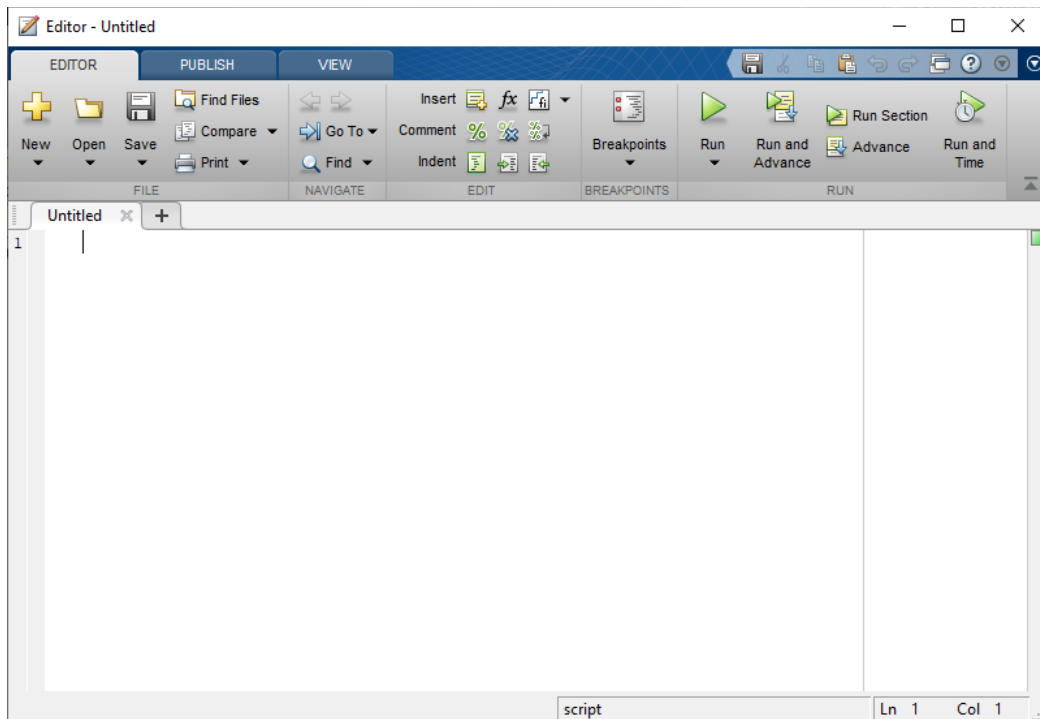


Figure 1b. MATLAB Editor (Undocked)

MATLAB Scripts

MATLAB scripts or programs are sequences of MATLAB statements stored in a text file that are executed one statement at a time. MATLAB script files must have the extension `.m`. Scripts can be created in any text editor, but the MATLAB editor has syntax highlighting for MATLAB keywords, reserved words, and many MATLAB constructs; and allows breakpoints to be set for debugging.

Statements in the MATLAB script file are executed in sequential order just as if all the statements were typed in the MATLAB Command Window. Statements in the script, when executed, have access to all current workspace variables, and anything created or modified by statements in the script will be saved in the workspace. Normally, the MATLAB statements in the script file are not displayed as they are executed. The results of these commands are displayed though. Semicolons placed after MATLAB statements have the effect of suppressing the result of the statements.

Scripts can be run two ways: either by typing the script name in the Command Window or by selecting Run from the Editor tab of the Editor Toolstrip. MATLAB expects scripts to be in the current folder or in one of the default search paths. MATLAB's processing of a script can be interrupted at any time by hitting `Ctrl-c` (control c).

Figure 2b shows MATLAB a script that solves for the voltages and currents in the series resistive circuit of Figure 2a. For conciseness, the program header describing what the script does has been omitted.

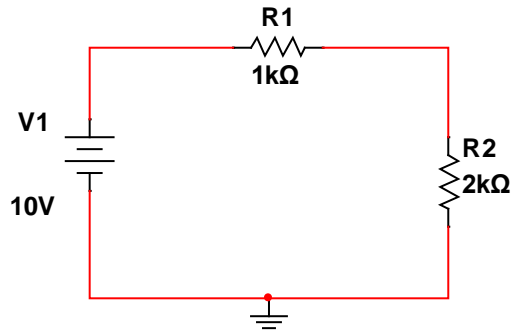


Figure 2a. Series Resistive Circuit

```

%% determine currents and voltages
clear; clc;
% source voltage (V) and resistances (Ohms)
Vs = 10;
R1 = 1000;
R2 = 2000;

% compute current in loop
I = Vs/(R1 + R2);
% compute voltage across each resistor
VR1 = I * R1;
VR2 = I * R2;

% display the current and voltage
fprintf("Current in the circuit is %0.2gA\n", I);
fprintf("Voltage across R1 is %0.2fV\n", VR1);
fprintf("Voltage across R2 is %0.2fV\n", VR2);

```

Figure 2b. Circuit Analysis Script

Notice that sections of the script are separated using blank lines and comments. Comments are denoted in MATLAB using the percent sign (%) and show up in green. Everything after the percent sign is a comment and is ignored by the MATLAB interpreter. Multiple statements can be entered in the command line if they are separated by commas or semicolons.

MATLAB also allows one to break scripts up into sections that can be run separately. MATLAB code sections are denoted by double percent signs. One % denotes a comment and two %% denote a new code section. The line the %% is on also serves as a comment line. To execute the statements in a code section, select the code section and choose Run section from the Editor tab toolbar. Choosing Run from the Editor tab toolbar will execute the entire script. The script of Figure 2b has two code sections; one for the computations and one for the display. Code sections are normally only used in longer scripts.

There are MATLAB statements that are rarely used in the Command Window but are commonly used in scripts; two of these are `input` and `fprintf`. The `input` function displays a string

as a user prompt and reads in a line from the user. Instead of setting the source voltage to a constant 10 Volts in the script of Figure 2b, the source voltage could be read from the user using the statement:

```
V1 = input("Enter the source Voltage in Volts: ");
```

The `fprintf` function displays a formatted string and is used for program output.

Figure 3 shows a script that determines that initial velocity of a water balloon thrown straight up for it to reach one mile. The program header and the clearing of the command window and workspace variables is omitted for conciseness.

```
% acceleration due to gravity in ft/s^2
a = -32.14;
% initial and final position in ft
x0 = 0.0;
x = 5280.0;    % one mile
% final velocity in ft/s
v = 0.0;

% initial velocity to reach one mile
v0 = sqrt(v^2 - 2*a*(x-x0));
fprintf("Initial velocity is %0.2f ft/s\n", v0);
```


Figure 3. Balloon Height Script

The balloon height script was developed as follows:

- Problem is described: two dimensional motion can be described using kinematic equations relating position to velocity, acceleration, and time $x - x_0 = v_0 t + \frac{1}{2} a t^2$ and relating velocity to position and acceleration $v^2 = v_0^2 + 2a(x - x_0)$; where x is position (x_0 is the initial position), v is velocity (v_0 is the initial velocity), a is acceleration, and t is time.
- Problem is solved analytically: only the second equation relating velocity to position and acceleration is needed, solving for the initial velocity the equation $v_0^2 = v^2 - 2a(x - x_0)$ is obtained, and the initial velocity is the square root of v_0^2 , $v_0 = \sqrt{v^2 - 2a(x - x_0)}$
- Inputs and outputs of the script were determined: output is the initial velocity, there are no inputs (although a more general problem might have the altitude to reach as the input) but the initial position, final position, and acceleration due to gravity need to be specified
- Sequence of operations is determined and operations are specified using one or more MATLAB statements: define acceleration due to gravity, define initial and final position, define final velocity, compute initial velocity using equation, display initial velocity

Take advantage of the syntax highlighting when writing and debugging programs using the MATLAB editor. Keywords show up in blue, comments show up in green, unterminated strings show up in maroon and completed strings in purple. The rest of the program text will be black. Syntax errors are indicated by red squiggly lines under the text and warnings are indicated by highlighted text with orange squiggly lines underneath.

Last modified Friday, September 18, 2020

 [MATLAB Marina](#) is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.