THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

ENG 101

Matlab Scripts & Functions

Dr. Emil Jovanov
Electrical and Computer Engineering
University of Alabama in Huntsville
emil.jovanov@uah.edu
http://www.ece.uah.edu/~jovanov

Scripts

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

So far, have run MATLAB commands by typing in single command, pressing ENTER, getting MATLAB's result, and then repeating this process for next command

- Not practical for calculations involving more than a few commands. Can use up and down arrow keys to avoid lots of typing, but still not practical

Better way

- Save all commands in a file
- With one command in Command Window, tell MATLAB to run all commands in file

Will use script files to do this

A *script file* is a sequence of MATLAB commands, also called a program

- When a script file runs (is executed), MATLAB performs the commands in the order they are written, just as if they were typed in the Command Window
- When a script file has a command that generates an output (e.g. assignment of a value to a variable without semicolon at the end), the file displays the output in the Command Window

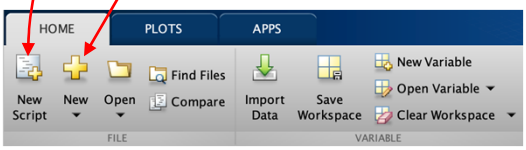
Scripts

- Using a script file is convenient because it can be edited (corrected and/or changed) and executed many times
- Script files can be typed and edited in any text editor and then pasted into the MATLAB editor
- Script files are also called *M-files* because the extension .m is used when they are saved

Use the Editor Window to work with script files

Can open window and create file two ways

1. Click on New Script icon
2. Click on New icon, select Script
3. In the Command Window, type `edit` and then press ENTER



Editor has tool strip on top with three tabs – EDITOR, PUBLISH, VIEW

- MATLAB used most often with EDITOR tab selected

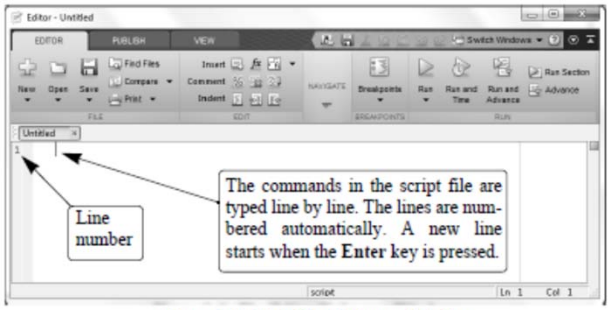


Figure 1-6: The Editor/Debugger Window.

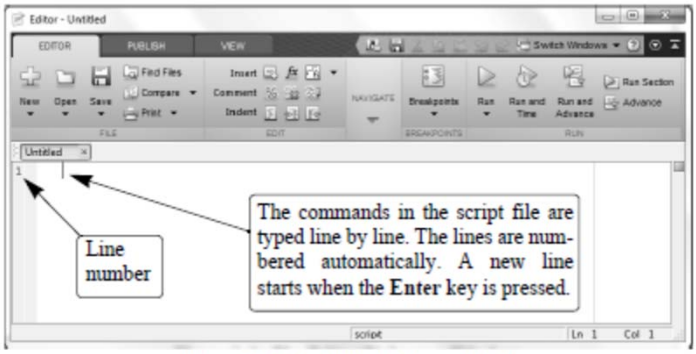


Figure 1-6: The Editor/Debugger Window.

- Type in commands line by line, pressing ENTER after each one
- MATLAB automatically numbers lines

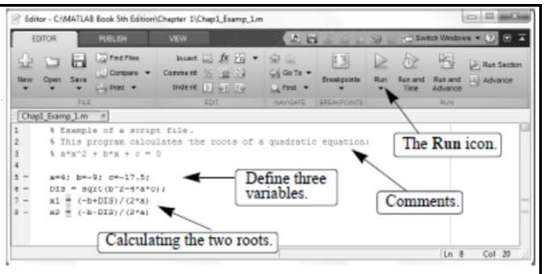


Figure 1-7: A program typed in the Editor/Debugger Window.

Comment lines

- Lines that start with percent sign (%)
- Common for first few lines to be comments and to briefly explain what commands in file do
- Editor Window shows comment lines in green

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Before MATLAB can run commands in file, you must save file

- If you haven't named file yet, click on Save icon, MATLAB brings up Save As dialog box
- If you've already named and saved file, just click on Save icon
- If you don't add an extension (.xxx) to the file name, MATLAB adds ".m"
- Rules for file names are same as rules for function names
- Don't use names of your variables, predefined variables, MATLAB commands, or MATLAB functions

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

To *execute* a script file means to run all of the commands in it.

You can execute a file by

- Pressing the Run icon (a green arrow)
- Typing the file name in the Command Window and pressing ENTER

MATLAB will execute file if it is in MATLAB's current folder or if the file's folder is in the search path (explained next)

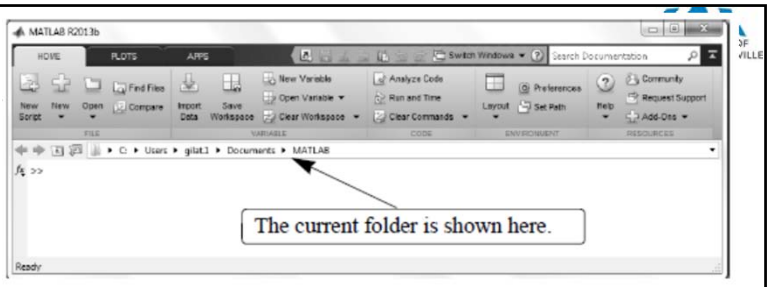


Figure 1-8: The Current folder field in the Command Window.

The *current folder* is the folder that MATLAB checks first when looking for your script file

- Can see current folder in desktop toolbar
- Can also display current folder by issuing MATLAB command pwd

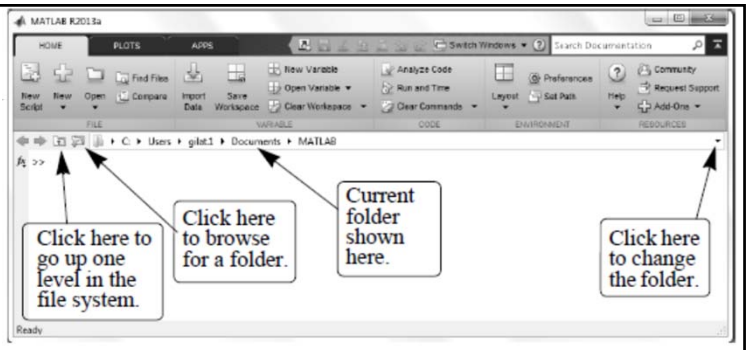


Figure 1-10: The Current Folder Window.

Can change current folder in Current Folder Window

- To show Current Folder Window, click on Layout icon in desktop, then select Current Folder

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Can change current folder from command line using `cd` command, space, new folder name in single quote marks, ENTER, i.e.,

```
>> cd 'new folder'
```

For example,

```
>> cd 'F:\slides\Chapter 1'
```


Variables

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

A *variable* is a name that is assigned a numerical value

- Once assigned, can use variable in expressions, functions, and MATLAB statements and commands
- Can *read* the variable (get its value)
- Can *write to* the variable (set its value)

= (equals sign) is MATLAB's *assignment operator*. It evaluates the expression on its right side and stores the resulting value in the variable on its left side

```
>> a = 3
a =
     3
```

Create the variable called "a" and store the value 3 in it
MATLAB acknowledges that it has created "a" and set it to 3

EXAMPLE

```
>> a = 3
a =
     3
>> b = 10*a + 5
b =
     35
```

Make a variable and store a number in it

Make a variable and store the value of an expression made up of a variable, numbers, and addition and multiplication

Think of = as meaning "assign to" or "store in" but not meaning "equals"!

Why?

x = *x* + *6* has no meaning in math because it implies that *0* = *6*

x = x + 6 is perfectly fine in MATLAB because it means "take whatever is in x, add 6 to that and store the result back into x"

EXAMPLE

```
>> x = 3;
>> x = x + 6
x =
     9
>> x = 2 * x
x =
    18
```

; at end prevents MATLAB from displaying value of x

takes what's in x (3), adds 6 to it to get 9, then stores 9 back into x

now x's value is 9

takes what's in x (9), multiplies it by 2 to get 18, then stores 18 back into x

now x's value is 18

A variable must have a value before you use it in an expression

```
>> x = 3;
>> x+2
ans =
     5
>> x + y % assume y undefined
??? Undefined function or variable
'y'
```


To find out the value of a variable, just type it and press ENTER

```
>> x = 3;
>> y = 10 * x;
>> z = y ^ 2;
>> y
y =
     30
>> z
z =
     900
```

Can do multiple assignments on one line by separating with a comma or semicolon. If semicolon, no display for that assignment

```
>> a=12, B=4; C=(a-B)+40-a/B*10
a =
     12
C =
     18
```


To change the value of a variable, just assign it the new value

```
>> ABB=72;
>> ABB=9;
>> ABB
ABB =
      9
```

You must define a variable (give it a value) before you can use it in an argument of a function

```
>> sqrt( x ) % assume x undefined
??? Undefined function or variable 'x'
>> x = 144;
>> sqrt( x )
x =
     12
```


A variable name

- Must begin with a letter
- Can be up to 63 characters long
- Can contain letters, digits, and underscores (_)
- Can't contain punctuation, e.g., period, comma, semicolon

Avoid using the name of a built-in function as the name of a variable, e.g., don't call a variable `exp` or `sqrt`

MATLAB is *case-sensitive*, and does not consider an upper-case letter in a variable name to be the same as its lower-case counterpart, e.g., `MTV`, `MTv`, `mTV`, and `mtv` are four different variable names

A variable name cannot contain a space. Two common alternatives:

1. Use an underscore in place of a space, e.g., `speed_of_light`
2. Capitalize the first letter of every other word, e.g., `speedOfLight` (This is known as *camel case*!)

A *keyword* is a word that has special meaning to MATLAB

- There are 20 keywords (see book)
- Appear in blue when typed in the Editor Window
- Can't be used as variable names

MATLAB has pre-defined variables for some common quantities

`pi` the number π

`eps` the smallest difference between any two numbers in MATLAB

`inf` or `Inf` infinity

`i` $\sqrt{-1}$

`j` $\sqrt{-1}$ (same as `i`) but commonly used instead of `i` in electrical engineering

More pre-defined variables

`ans` the value of the last expression that was not assigned to a variable

`NaN` or `nan` not-a-number. Used to express mathematically undefined values, such as `0/0`

Some commands for managing variables

Command	**Outcome**
`clear`	Removes all variables from memory
`clear x y z`	Removes only variables `x`, `y`, and `z` from memory
`who`	Displays a list of the variables currently in memory
`whos`	Displays a list of the variables currently in memory and their size, together with information about their bytes and class (see Section 4.1)

Scripts

So far, have run MATLAB commands by typing in single command, pressing ENTER, getting MATLAB's result, and then repeating this process for next command

- Not practical for calculations involving more than a few commands. Can use up and down arrow keys to avoid lots of typing, but still not practical

Better way

- Save all commands in a file
- With one command in Command Window, tell MATLAB to run all commands in file

Will use script files to do this

A *script file* is a sequence of MATLAB commands, also called a program

- When a script file runs (is executed), MATLAB performs the commands in the order they are written, just as if they were typed in the Command Window
- When a script file has a command that generates an output (e.g. assignment of a value to a variable without semicolon at the end), the file displays the output in the Command Window

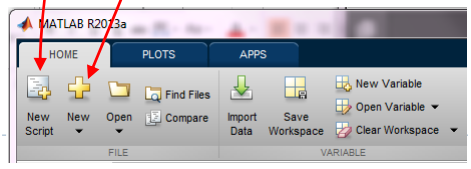
Scripts

- Using a script file is convenient because it can be edited (corrected and/or changed) and executed many times
- Script files can be typed and edited in any text editor and then pasted into the MATLAB editor
- Script files are also called *M-files* because the extension .m is used when they are saved

Use the Editor Window to work with script files

Can open window and create file two ways

1. Click on New Script icon
2. Click on New icon, select Script
3. In the Command Window, type `edit` and then press ENTER



Editor has tool strip on top with three tabs – EDITOR, PUBLISH, VIEW

- MATLAB used most often with EDITOR tab selected

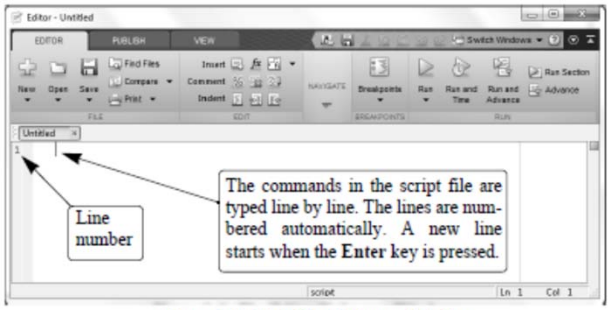


Figure 1-6: The Editor/Debugger Window.

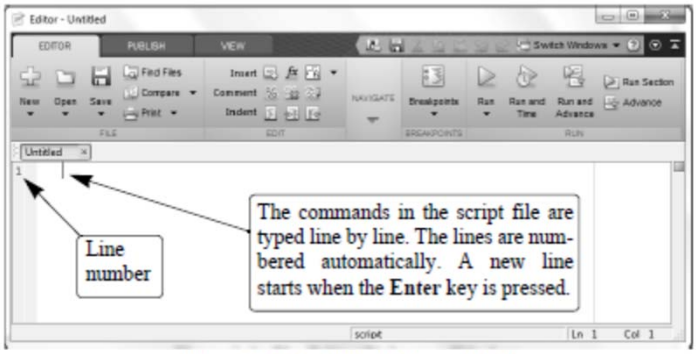


Figure 1-6: The Editor/Debugger Window.

- Type in commands line by line, pressing ENTER after each one
- MATLAB automatically numbers lines

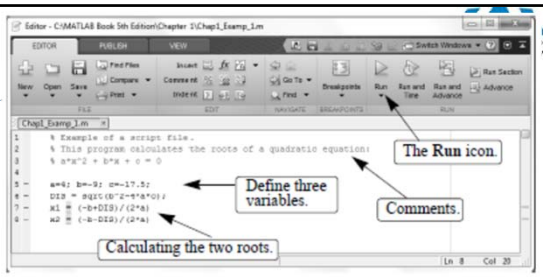


Figure 1-7: A program typed in the Editor/Debugger Window.

Comment lines

- Lines that start with percent sign (%)
- Common for first few lines to be comments and to briefly explain what commands in file do
- Editor Window shows comment lines in green

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

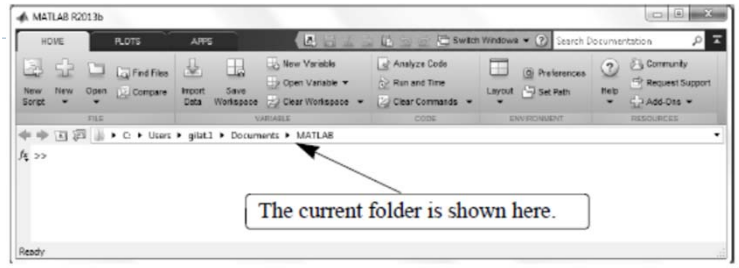
Before MATLAB can run commands in file, you must save file

- If you haven't named file yet, click on Save icon, MATLAB brings up Save As dialog box
- If you've already named and saved file, just click on Save icon
- If you don't add an extension (.xxx) to the file name, MATLAB adds ".m"
- Rules for file names are same as rules for function names
- Don't use names of your variables, predefined variables, MATLAB commands, or MATLAB functions

To *execute* a script file means to run all of the commands in it. You can execute a file by

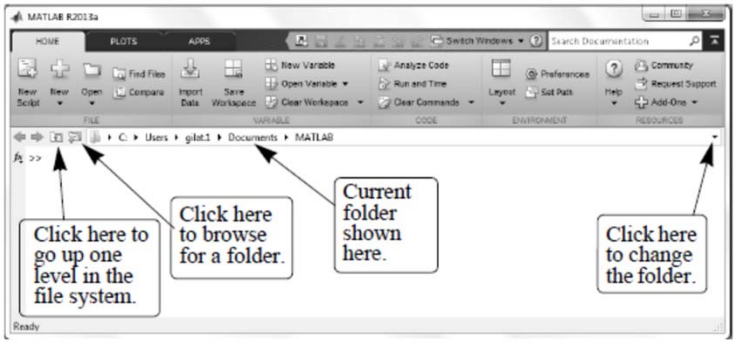
- Pressing the Run icon (a green arrow)
- Typing the file name in the Command Window and pressing ENTER

MATLAB will execute file if it is in MATLAB's current folder or if the file's folder is in the search path (explained next)



The *current folder* is the folder that MATLAB checks first when looking for your script file

- Can see current folder in desktop toolbar
- Can also display current folder by issuing MATLAB command pwd



Can change current folder in Current Folder Window

- To show Current Folder Window, click on Layout icon in desktop, then select Current Folder

Can change current folder from command line using `cd` command, space, new folder name in single quote marks, ENTER, i.e.,

```
>> cd 'new folder'
```

For example,

```
>> cd 'F:\slides\Chapter 1'
```

Setting up an m-file

- The first lines in an m-file are all comments and should consist of information:
- **%% Assignment Title – use a heading here to get a table of contents**
- **% Your Name**
- **% ENG 101**
- **% Due Date 8/21/2020**
- **%**
- %% Section Heading goes here for the first section of the file. i.e. Assignment 1, Part 1

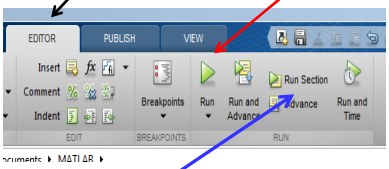
- **Note: Section headings start with %% and there must be a space between the %% and the first letter**
- **Note: You can run/execute an individual section or the entire m-file**

Example:

- Click on the new script button in the upper left part of the MATLAB window (Home tab selected). If Editor tab is selected, click on New and select script
- Add the required header (see previous slide) and save the new script as TestScript.m
- Put in a section break titled Radius
- *radius = 30;*
- Put in a section break titled Circumference
- *Circum = 2*pi*radius*
- Put in a section break titled Area
- *Area = pi*radius*radius*

Running an m-script

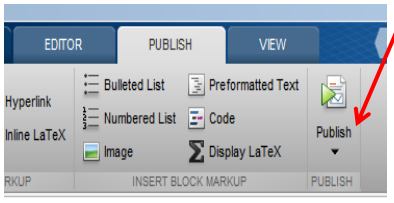
- To run the entire m-script file, click on the **run icon** in the ribbon when the Editor tab is selected.



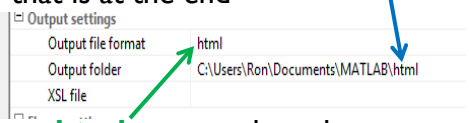
- To run a particular section of the m-script file, click on any code in that section to highlight the section. Then click on **run section**

To publish a file

- First click on the Publish tab next to the Editor Tab, and then click on the Publish **down arrow** **and select Edit Publishing Options**



in the window that appears, change the output folder name to whatever you want – for now just remove the **html folder** that is at the end



- For the output file format, click on **html** to get a drop down menu and select **pdf** from the menu. Once that is done, click on publish at the bottom of the pane.