



ENGR 1100




Week 01- Introduction to MATLAB
& Basic Algebra

DISPLAY FORMAT

Agenda

1. General information about MATLAB environment
2. MATLAB command windows
3. Arithmetic operations with scalars
4. MATLAB Display Format
5. MATLAB Built-in Functions
6. MATLAB Variables
7. MATLAB Scripts

Environment

- **Window Action icon** – the icon showing a down arrow with a circle around it, i.e.,  It is in the upper, right corner of most MATLAB windows
- **Help icon** – the question-mark icon  in the Resources Section of the desktop toolbar
- **Layout icon** –  in the Environment Section of the desktop toolbar



[Introduction to MATLAB Enviroment](#)

Starting MATLAB: MATLAB Windows

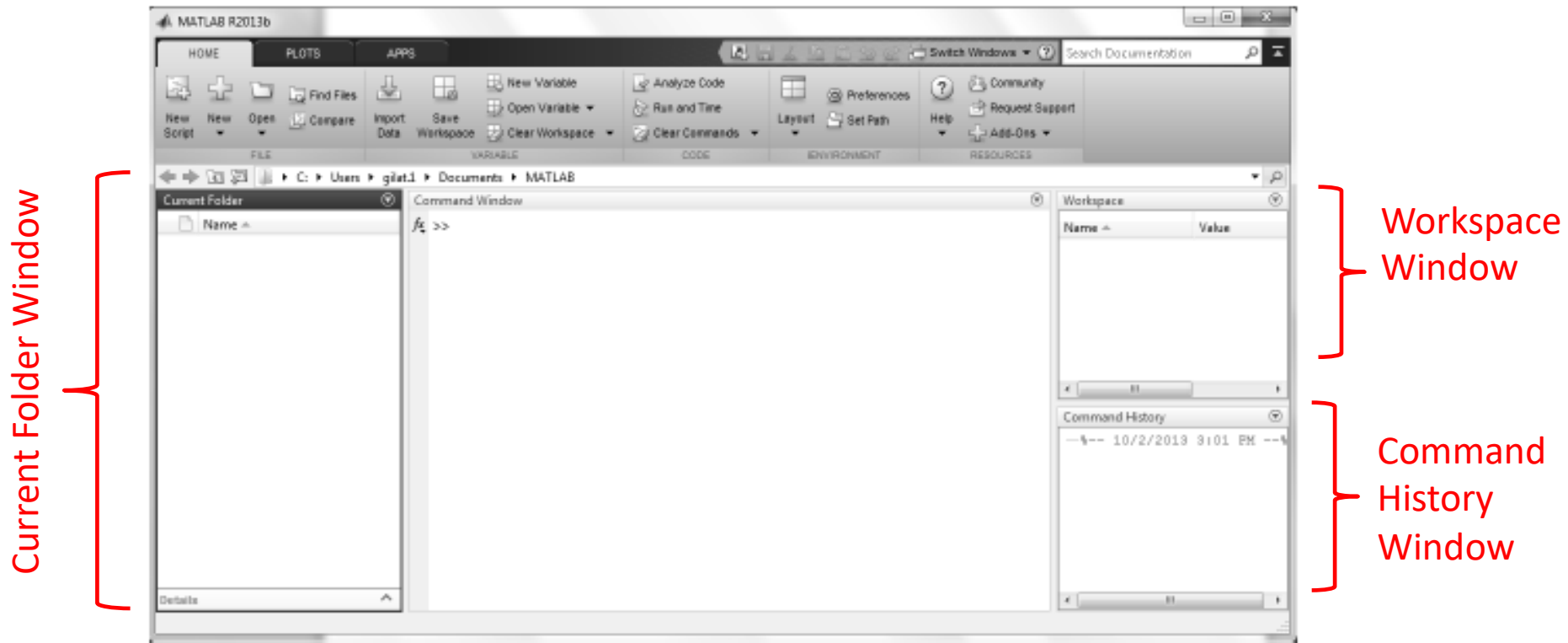


Figure 1-1: The default view of MATLAB desktop.

Command Window

Starting MATLAB: MATLAB Windows

Window	Purpose
Command Window	Main window, enters variables, runs programs.
Figure Window	Contains output from graphic commands.
Editor Window	Creates and debugs script and function files.
Help Window	Provides help information.
Command History Window	Logs commands entered in the Command Window.
Workspace Window	Provides information about the variables that are stored.
Current Folder Window	Shows the files in the current folder.

Starting MATLAB: MATLAB Windows

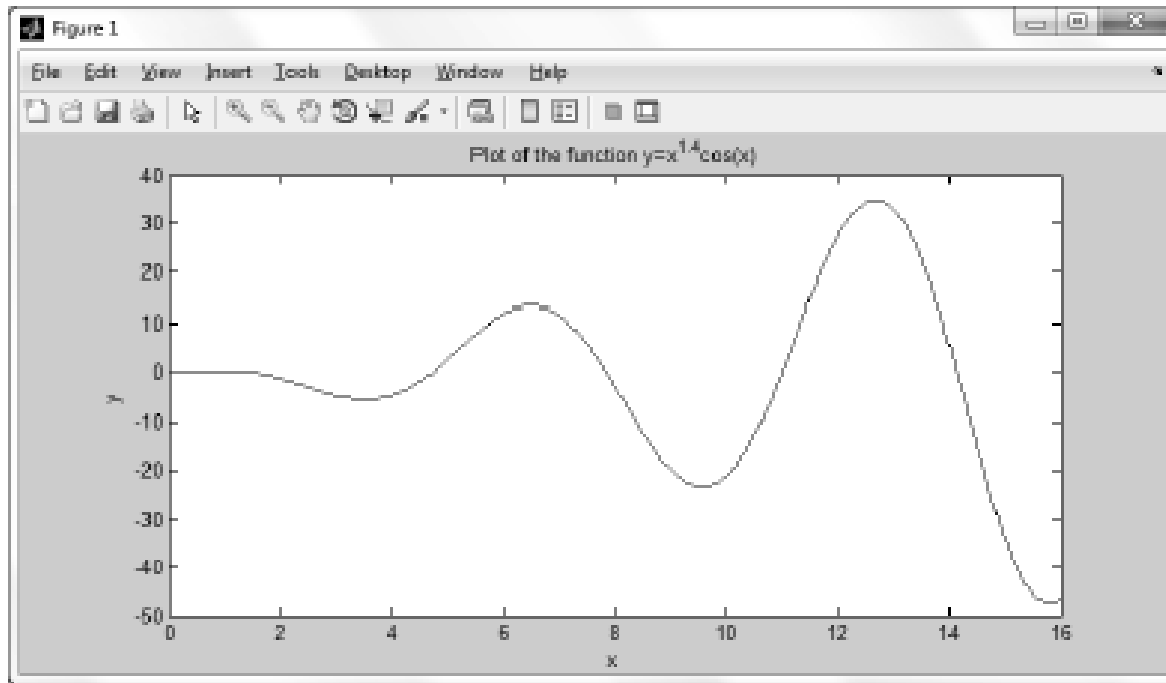


Figure 1-2: Example of a Figure Window.

Figure Window opens automatically after any command that draws a graph



If you don't see a figure window open up, look on the task bar for a black, program bar and click it

Starting MATLAB: MATLAB Windows

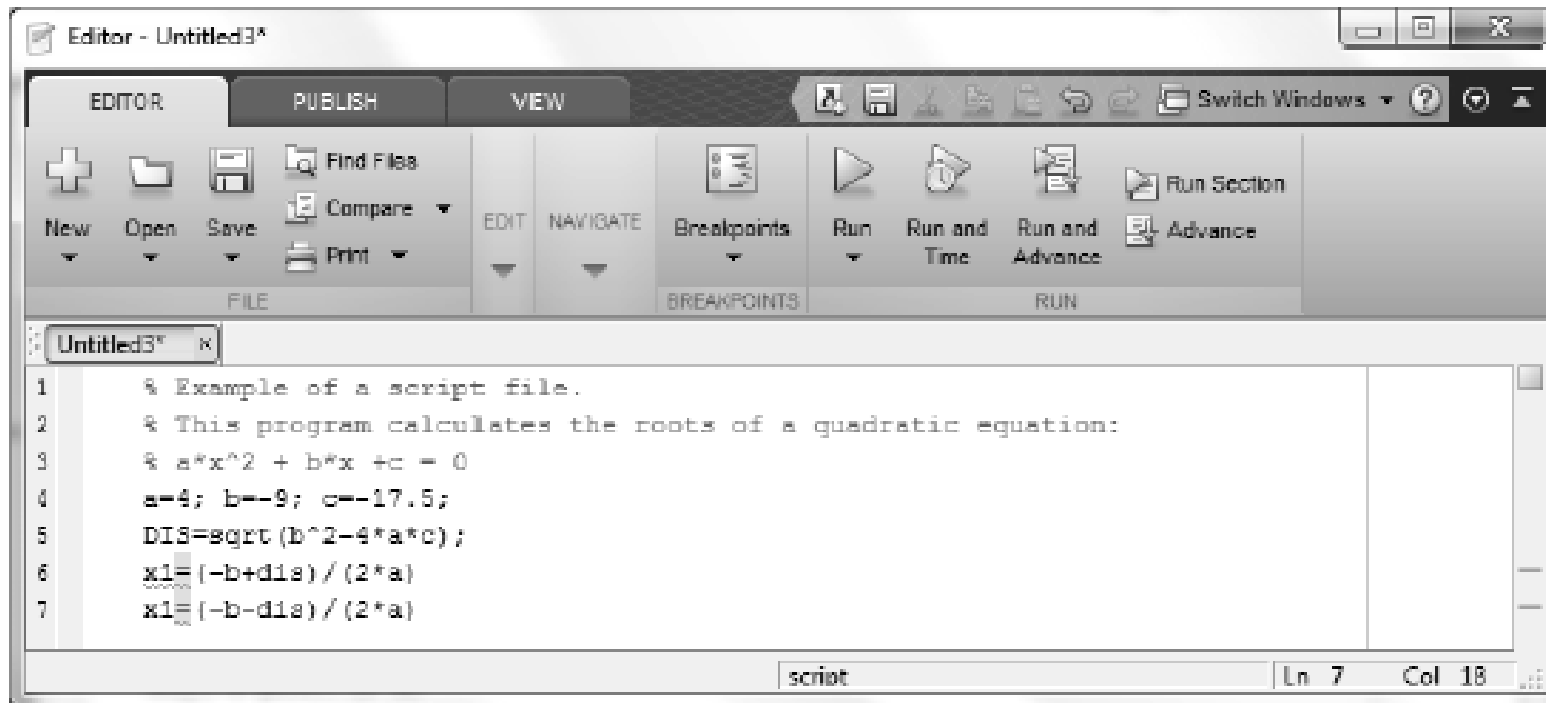
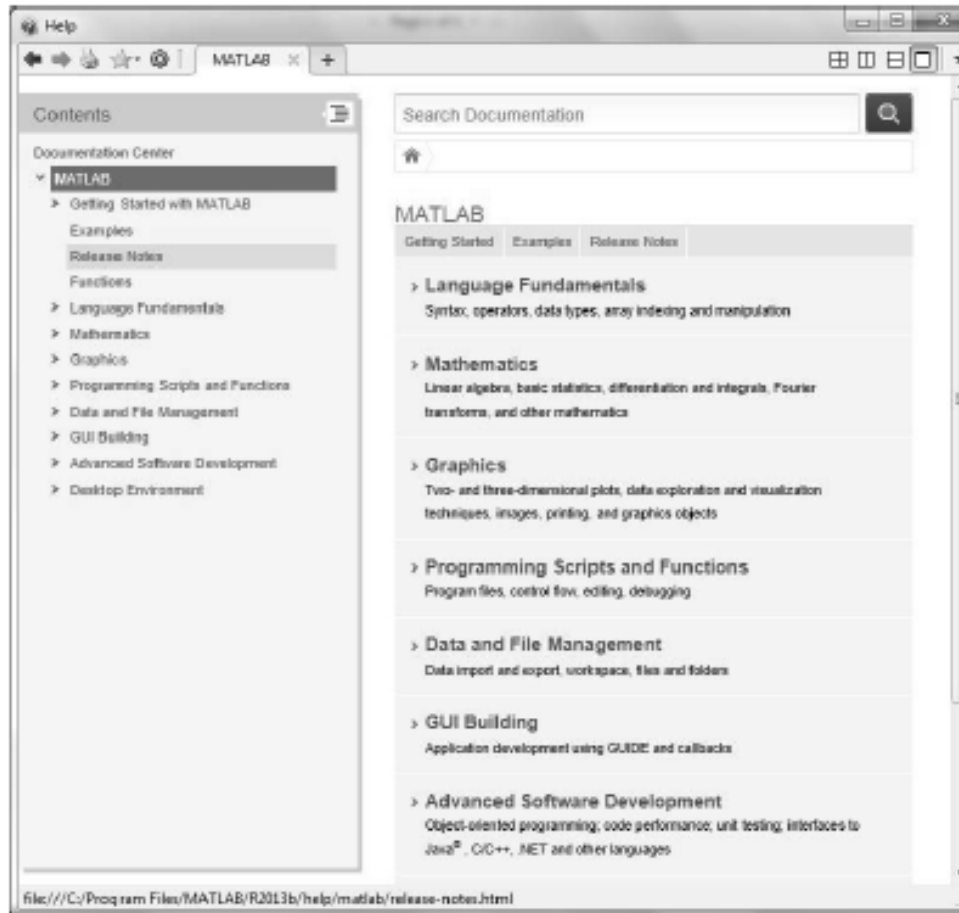
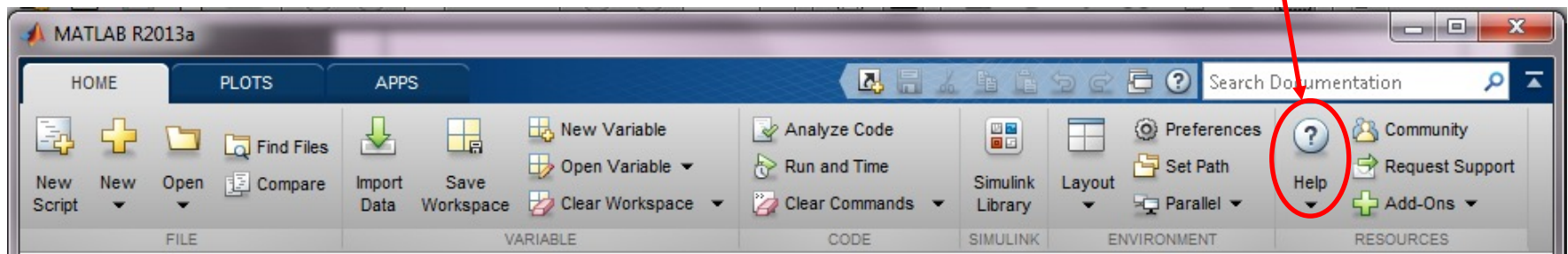


Figure 1-3: Example of an Editor Window.

Use Editor Window to write and debug MATLAB scripts.
Open with `edit` command



Get Help Window by clicking on Help icon (question mark) in tool strip



COMMAND WINDOWS

Command Window

Command Window is MATLAB's main window.
Use it to:

- Execute commands
- Open other windows
- Run programs that you've written
- Manage the MATLAB software

Command Window

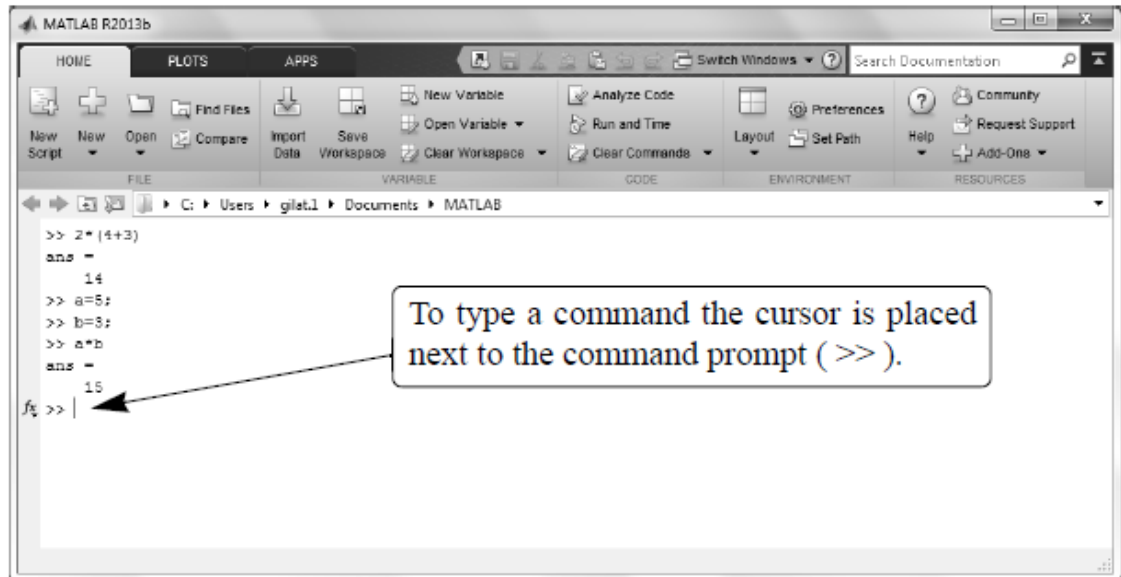


Figure 1-5: The Command Window.

Basic procedure

1. At prompt (`>>`), type in MATLAB command
2. Press ENTER key
3. MATLAB displays result in Command Window, followed by a prompt
4. Repeat from step 1

Command Window

To quickly execute a previous command but with minor changes

1. Recall command with up- and down-arrow keys
2. Use left- and right-arrow keys to move to characters to be altered
3. Use BACKSPACE or DELETE to remove old character, then type new character
4. Press ENTER to execute modified command

Command Window

Semicolon (;)

- When typed at end of command, suppresses output. (Only prompt displayed at next line)
 - Useful for preventing display of large outputs
 - Used much more in scripts

Percent sign(%)

- When typed at beginning of line, MATLAB treats line as a *comment* and doesn't execute line
 - Used much more in scripts

ARITHMETIC WITH SCALARS

Arithmetic Operations with Scalars

In this section, we will only discuss arithmetic with *scalars* (single numbers)

- Can do arithmetic directly on numbers (like a calculator)
- Can store numbers in variables



[MATLAB: Arithmetic](#)

Arithmetic Operations with Scalars

Symbols for arithmetic are:

Operation	Symbol	Example
Addition	+	$5 + 3$
Subtraction	-	$5 - 3$
Multiplication	*	$5 * 3$
Right division	/	$5 / 3$
Left division	\	$5 \setminus 3 = 3 / 5$
Exponentiation	^	$5 \wedge 3$ (means $5^3 = 125$)

****Left division rarely used with scalars**

Order of Precedence

Order in which MATLAB does arithmetic

Precedence	Mathematical Operation
First	Parentheses. For nested parentheses, the innermost are executed first.
Second	Exponentiation.
Third	Multiplication, division (equal precedence).
Fourth	Addition and subtraction.

Precedence order

- Same as most calculators
- Same as doing arithmetic by hand
- For multiple operations of same precedence, MATLAB goes left to right
- Can change order by using parentheses

Display Formats

Can control display of numbers with `format` command

- Once enter command, format stays the same until another `format` command
- **Default format** is fixed point with four digits to right of decimal point
 - *fixed-point* means decimal point always between one's-digit and one-tenth's digit
- Format only affects display of numbers. MATLAB always computes and saves numbers in full precision

Display Format for Numeric Values

Table 1-2: Display formats

Command	Description	Example
<code>format short</code>	Fixed-point with 4 decimal digits for: $0.001 \leq \text{number} \leq 1000$ Otherwise display format short e.	<pre>>> 290/7 ans = 41.4286</pre>
<code>format long</code>	Fixed-point with 15 decimal digits for: $0.001 \leq \text{number} \leq 100$ Otherwise display format long e.	<pre>>> 290/7 ans = 41.428571428571431</pre>
<code>format short e</code>	Scientific notation with 4 decimal digits.	<pre>>> 290/7 ans = 4.1429e+001</pre>
<code>format long e</code>	Scientific notation with 15 decimal digits.	<pre>>> 290/7 ans = 4.142857142857143e+001</pre>
<code>format short g</code>	Best of 5-digit fixed or floating point.	<pre>>> 290/7 ans = 41.429</pre>
<code>format long g</code>	Best of 15-digit fixed or floating point.	<pre>>> 290/7 ans = 41.4285714285714</pre>
<code>format bank</code>	Two decimal digits.	<pre>>> 290/7 ans = 41.43</pre>
<code>format compact</code>	Eliminates empty lines to allow more lines with information displayed on the screen.	
<code>format loose</code>	Adds empty lines (opposite of compact).	



[Mathworks Link](#)

BUILT-IN FUNCTIONS

Built-in Functions

MATLAB expressions can include functions. You can think of a *function* as a black box that, in general, takes inputs, does some computations with them, and produces outputs.



Built-in Functions

A function

- Has a name
- Can have zero or more *arguments* (inputs)
- Can produce zero or more outputs

```
y = sqrt ( x )
```

output

name

argument

Built-in Functions

A function's arguments can be

- Numbers
- Variables (explained in next section)
- Expressions involving numbers, variables, or functions

`sqrt(64)` Argument is a number

`sqrt(a)` Argument is the variable "a"

`atan(y/sqrt(3^2+y^2))`

Argument to arctan function is an expression that has a number (3), a variable (y), and a function (sqrt)



[Mathworks Link for Mathematical Functions](#)

Built-in Functions

Elementary math functions

- `sqrt(x)` – square root
- `nthroot(x, n)` – nth real root
- `exp(x)` – e^x
- `abs(x)` – absolute value
- `log(x)` – natural log (base e)
- `log10(x)` – log base 10
- `factorial(x)` – $x!$

Built-in Functions

Rounding functions

- `round(x)` – round to nearest integer
- `fix(x)` – round toward zero
- `ceil(x)` – round toward infinity
- `floor(x)` – round toward minus infinity
- `rem(x, y)` – remainder after `x` is divided by `y` (also called modulus)
- `sign(x)` – returns 1 if `x` is positive, -1 if `x` is negative, zero if `x` is zero

Use Variables in Built-in Functions

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
```

VARIABLES

Rules About Variable Names

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

Rules About 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 (`break`, `case`, `catch`, `classdef`, `continue`, `else`, `elseif`, `end`, `for`, `function`, `global`, `if`, `otherwise`, `parfor`, `persistent`, `return`, `spmd`, `switch`, `try`, `while`)
- Appear in **blue** when typed in the Editor Window
- Can't be used as variable names

Predefined Variables and Keywords

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

Predefined Variables and Keywords

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 $0/0$

You can redefine (change) the values of the predefined variables, but
don't. You'll cause confusion

- Exceptions are `i` and `j`, which are often used as loop variables
(see MATLAB programming module)

Useful Commands for Managing Variables

Some commands for managing variables

Command	Outcome
<code>clear</code>	Removes all variables from memory
<code>clear x y z</code>	Removes only variables <code>x</code> , <code>y</code> , and <code>z</code> from memory
<code>who</code>	Displays a list of the variables currently in memory
<code>whos</code>	Displays a list of the variables currently in memory and their size, together with information about their bytes and class

SCRIPTS

Script Files

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



[Mathworks Link on script](#)

Notes About Script Files

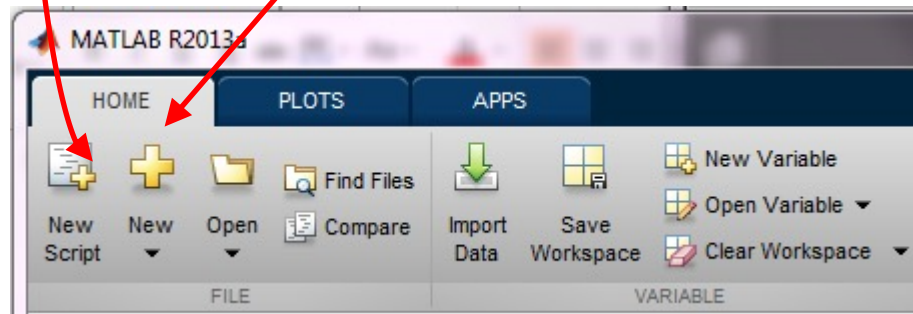
A **script file** is a sequence of MATLAB commands, also called a program

- 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
- 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

Creating and Saving a Script File

- 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

In the Command Window, type `edit` and then press ENTER



Creating and Saving a Script File

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

- MATLAB used most often with EDITOR tab selected

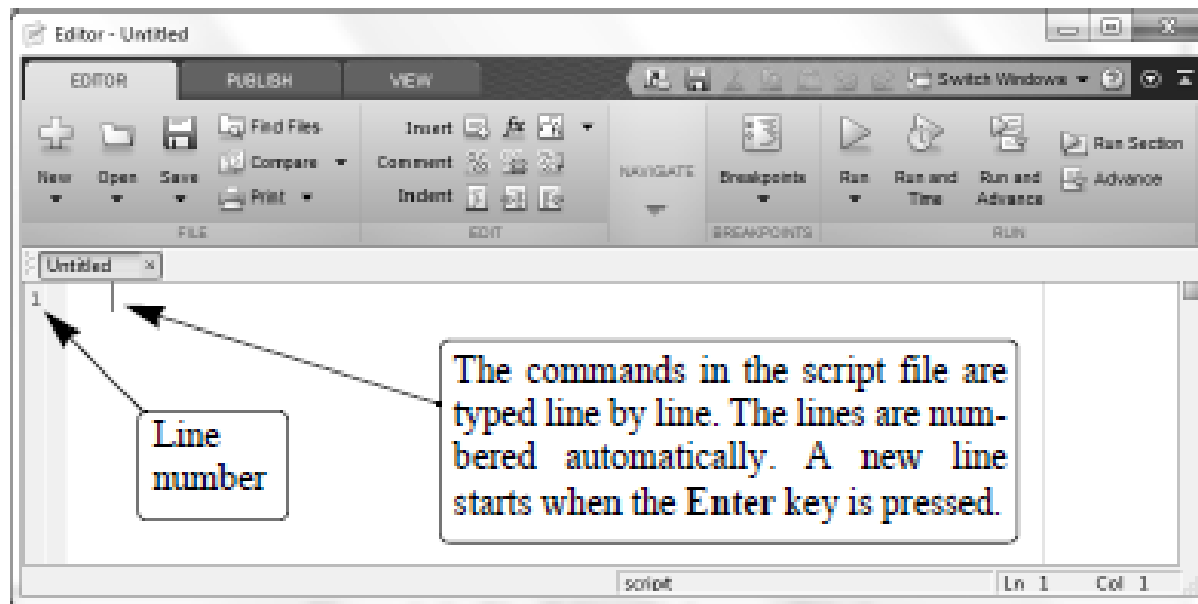


Figure 1-6: The Editor/Debugger Window.

Creating and Saving a Script File

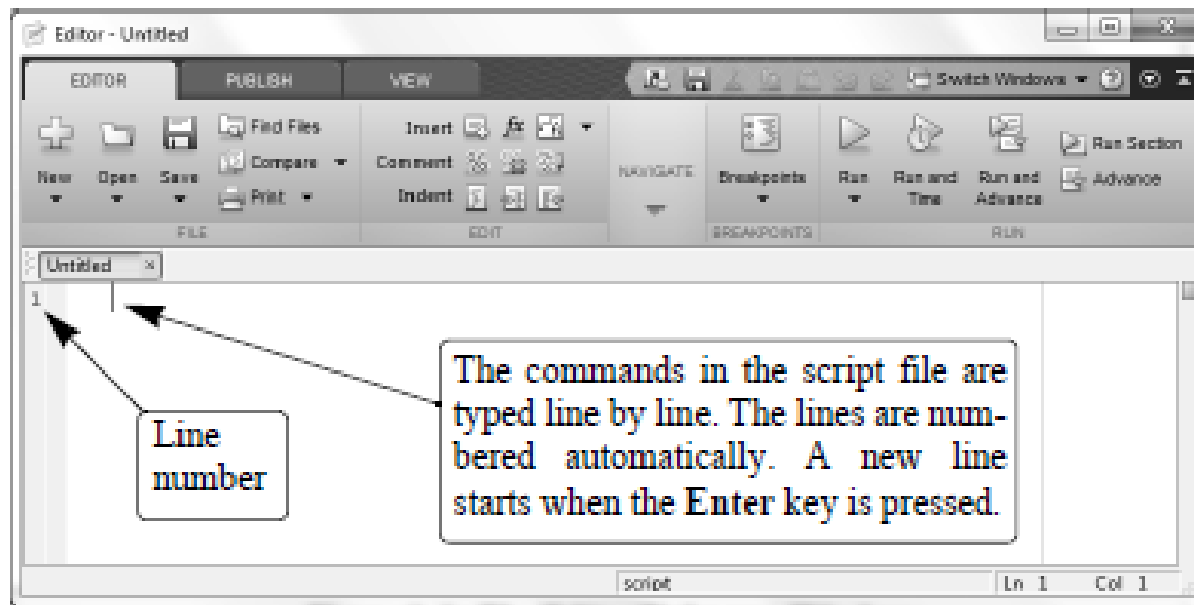


Figure 1-6: The Editor/Debugger Window.

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

Creating and Saving a Script File

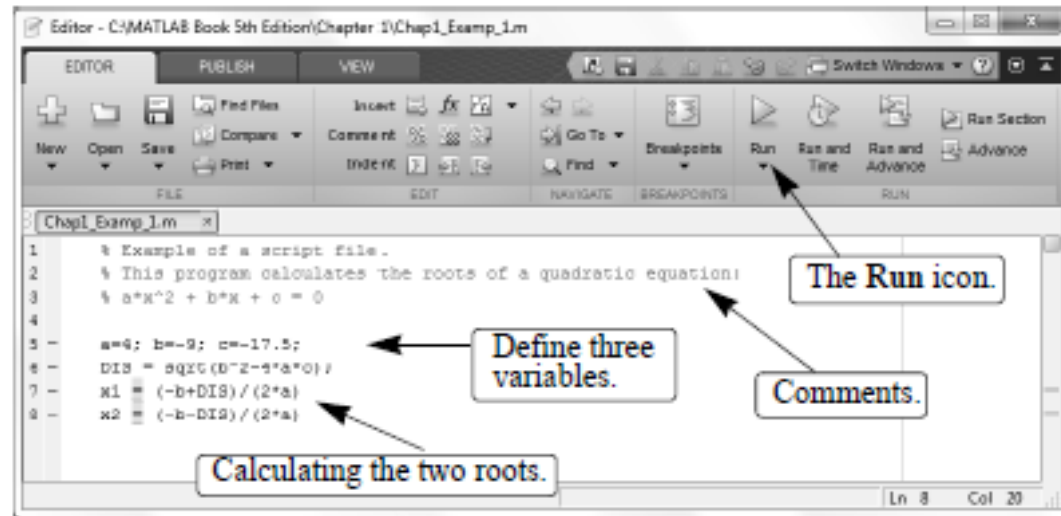


Figure 1-7: A program typed in the Editor/Debugger Window.

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

Creating and Saving a Script File

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

Running (Executing) a Script File

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

You can execute a file by

- a) Pressing the Run icon (a green arrow)
- b) 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)

Current Folder

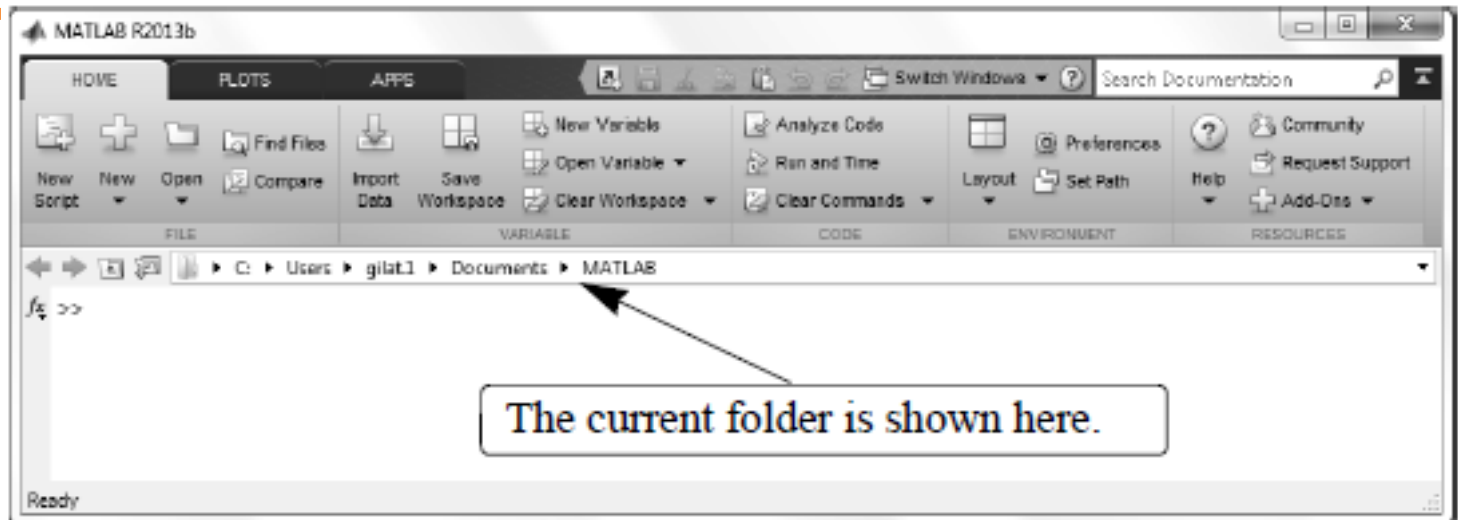


Figure 1-8: The Current folder field in the Command Window.

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`

Current Folder

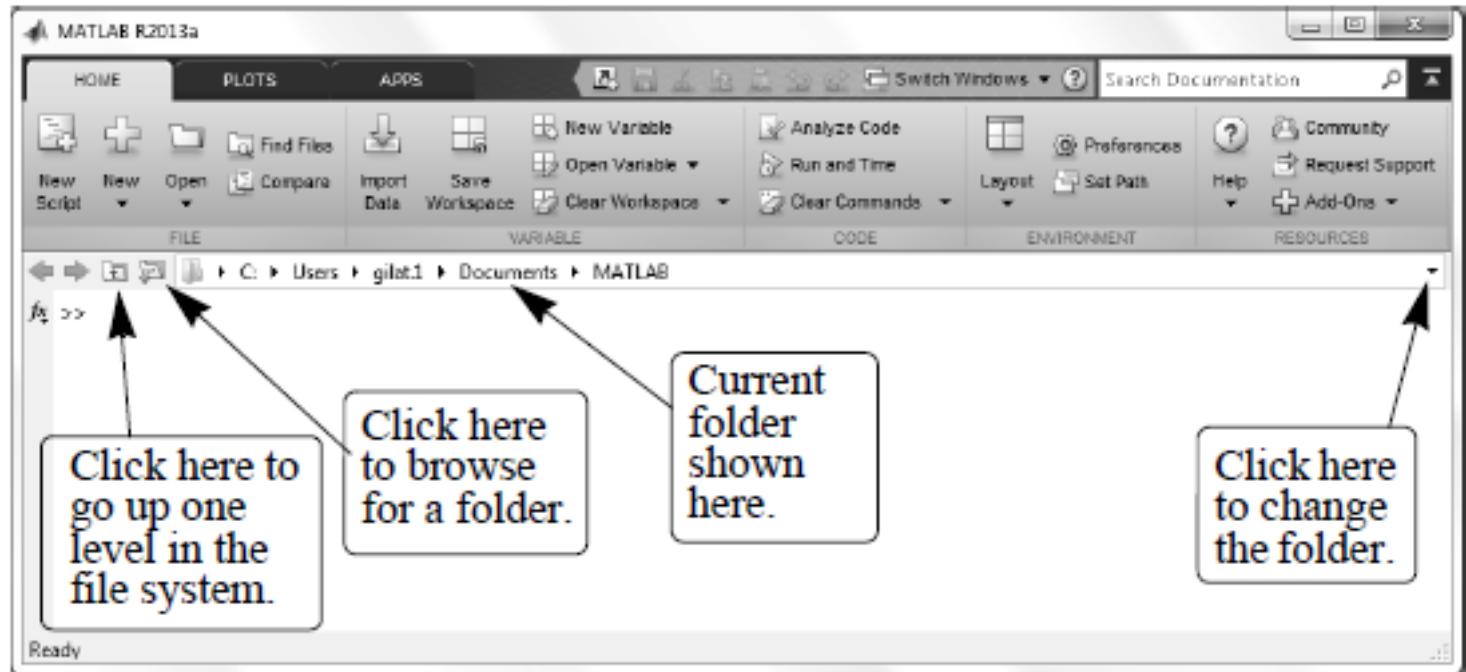


Figure 1-10: The Current Folder Window.

Can change current folder in Current Folder Window

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

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'
```