

# Where are we?

- Linux?



- Python?



- Matlab?



# Course Description

- Learning to program is similar to learning a spoken language:
  - is hard to do
  - is easier for some people than others
  - is difficult to learn how on your own by just reading a book
  - requires practice, lots of practice

# Matlab

Matlab (MATrix LABoratory)-

- commercial product built initially to simplify learning Fortran and later rebuilt using C
- Don't have to declare variable type, handles matrix math handily, etc.
- Matlab has tool packages to purchase, but don't import such tools, already loaded
- Let's get going!
  - Find matlab on your mac

## 1.1 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.

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
  - ver- version. What version of Matlab is installed and toolboxes available

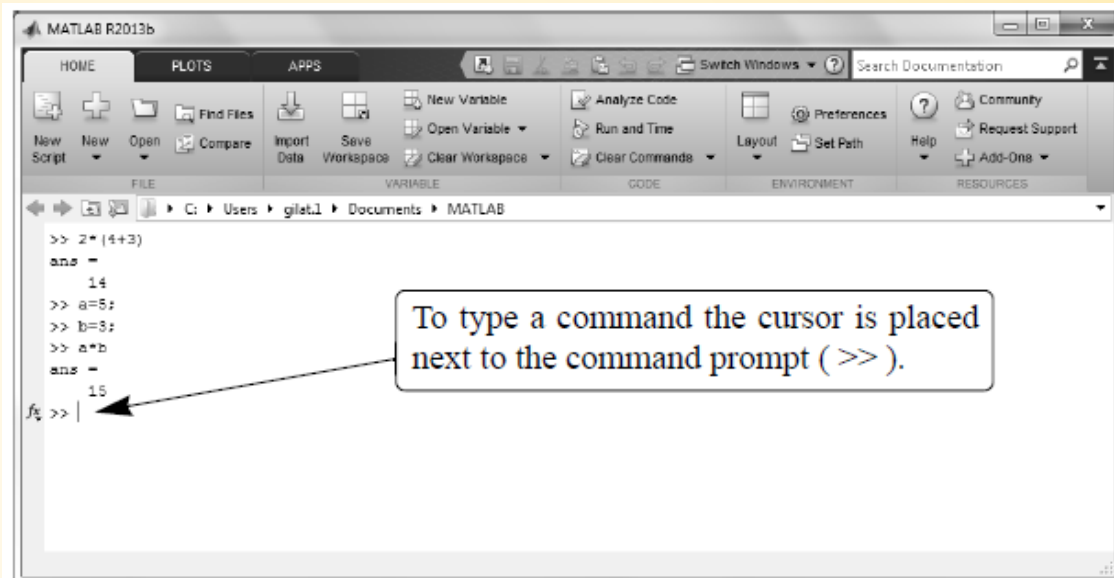


Figure 1-5: The Command Window.

### 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

## Notes on Command Window

- To start a command, make sure cursor is next to prompt
- MATLAB won't respond until you press ENTER
  - It then executes only last command
  - Commands before last one may still be visible, but MATLAB doesn't execute them

- Can type several commands in same line by putting a comma between commands
  - Hard to read, so don't do this often
- If command too long to fit on line, can continue to next line by typing ellipsis (3 periods, i.e., ... ) and then pressing ENTER



When cursor is in bottom command line:

- ← key moves cursor one character to left
- → key moves cursor one character to right
- ↑ key recalls preceding command
- ↓ key recalls command that follows one being displayed, i.e., undoes ↑

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 History Window

- Shows previous commands, including ones from previous MATLAB sessions
- Double-clicking on command puts it in Command Window and executes it
- Can drag command to Command Window, make changes in command, then execute it
- To clear one or more commands, select the lines to delete, right click, choose Delete Selection
- To clear entire history, right click, select Clear Command History

Symbols for arithmetic are:

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

What's different in python?

# 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

## Using MATLAB as a calculator

1. Type in mathematical expression
2. Press **Enter** key
3. MATLAB displays answer in Command Window as `ans =` followed by the result

Your display may appear on more than one line and have blank lines between text

# 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



# Some types of formatting

Table 1-2: Display formats		
Command	Description	Example
<code>format short</code>	Fixed-point with 4 decimal digits for: $0.001 \leq number \leq 1000$ Otherwise display format short e.	<pre>&gt;&gt; 290/7 ans =     41.4286</pre>
<code>format long</code>	Fixed-point with 15 decimal digits for: $0.001 \leq number \leq 100$ Otherwise display format long e.	<pre>&gt;&gt; 290/7 ans =     41.428571428571431</pre>
<code>format short e</code>	Scientific notation with 4 decimal digits.	<pre>&gt;&gt; 290/7 ans =     4.1429e+001</pre>
<code>format long e</code>	Scientific notation with 15 decimal digits.	<pre>&gt;&gt; 290/7 ans =     4.142857142857143e+001</pre>
<code>format short g</code>	Best of 5-digit fixed or floating point.	<pre>&gt;&gt; 290/7 ans =     41.429</pre>
<code>format long g</code>	Best of 15-digit fixed or floating point.	<pre>&gt;&gt; 290/7 ans =     41.4285714285714</pre>
<code>format bank</code>	Two decimal digits.	<pre>&gt;&gt; 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).	

# MATLAB expressions can include functions

*A function* takes inputs, does some computations with them, and produces outputs



## A function

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

`y = sqrt ( x )`  
output      name      argument

# A function's arguments can be

- Numbers
- Variables
- 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)

# 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)` –  $x!$

See Table 1-3 for details

# Trigonometric functions

- `sin(x)` – sine (x in radians)
- `sind(x)` – sine (x in degrees)
- `cos(x)` – cosine (x in radians)
- `cosd(x)` – cosine (x in degrees)
- `tan(x)` – tangent (x in radians)
- `tand(x)` – tangent (x in degrees)
- `cot(x)` – cotangent (x in radians)
- `cotd(x)` – cotangent (x in degrees)

See Table 1-4 for details

# Inverse trigonometric functions

- $\text{asin}(x)$  ,  $\text{acos}(x)$  ,  $\text{atan}(x)$  ,  
 $\text{acot}(x)$   
( $x$  in radians)
- $\text{asind}(x)$  ,  $\text{acosd}(x)$  ,  $\text{atand}(x)$  ,  
 $\text{acotd}(x)$  ( $x$  in degrees)

# Hyperbolic trigonometric functions

- $\cosh(x) = (e^x + e^{-x})/2$
- $\sinh(x) = (e^x - e^{-x})/2$
- $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$
- $\coth(x) = (e^x + e^{-x})/(e^x - e^{-x})$

# 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

See Table 1-5 for details



*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
```

Create the variable called "a" and store the value 3 in it

## EXAMPLE

>> a = 3    Make a variable and store a number in it

a =

3

>> b = 10\*a + 5    Make a variable and store the value of an  
expression made up of a variable,  
b =    numbers, and addition and multiplication

35

= means “assign to” or “store in”  
but not “equals”!

`x = x + 6` means “take  
whatever is in `x`, add 6 to that and  
store the result back into `x`”

*Is python any different?*

## EXAMPLE

>> x = 3;  ; at end prevents MATLAB from displaying value of x

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

x =

9

now x's value is 9

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

x =

18

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, MT<sub>V</sub>, mTV, and m<sub>t</sub>v are four different variable names

*Is python any different?*

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*!)

MATLAB has pre-defined variables for some common quantities; avoid using as variables

`pi` the number  $\pi$

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

`inf` or `Inf` infinity

`i`  $\sqrt{-1}$

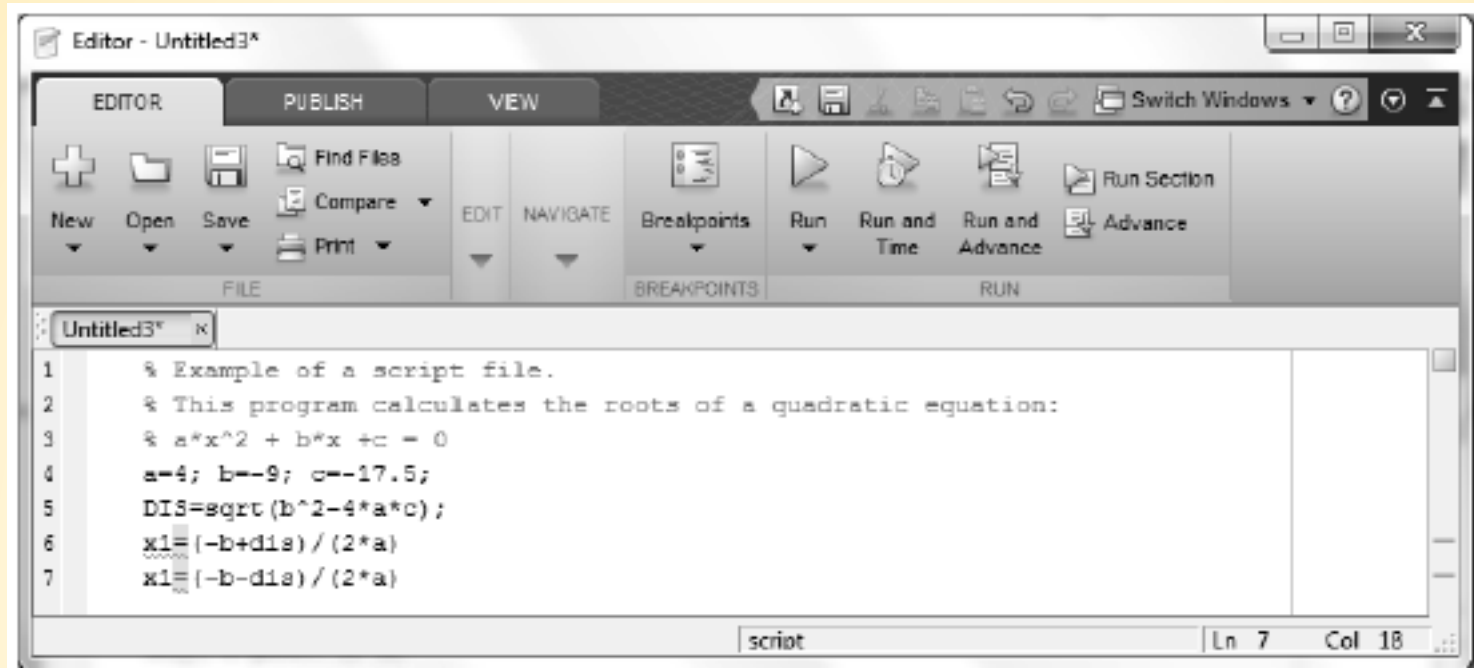
`ans` the value of the last expression that was not assigned to a variable

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

# Some commands for managing variables

Command	Outcome
<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 (see Section 4.1)

## 1.1 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



- Program or Script file: save all commands in a file rather than typing at command line
- Suffix for a script file is “.m”
- With one command in Command Window, tell MATLAB to run all commands in file
- *What is the suffix for a program file in python?*

*A script file/program/m file* is a sequence of MATLAB commands

- When a program 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

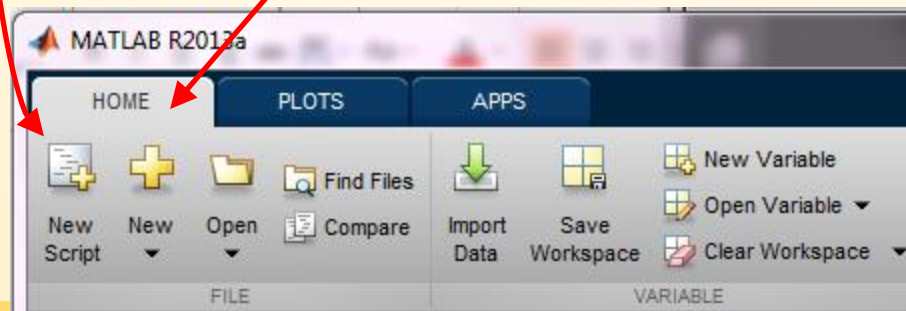
## **projectile.m and projectile\_function.m**

- script file and a function file
- compare to the python code
- differences:
  - can compute trig functions in degrees
  - not rigid on indenting
  - have to be careful to use array elementwise multiplication and exponentiation (discussed later)

# 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

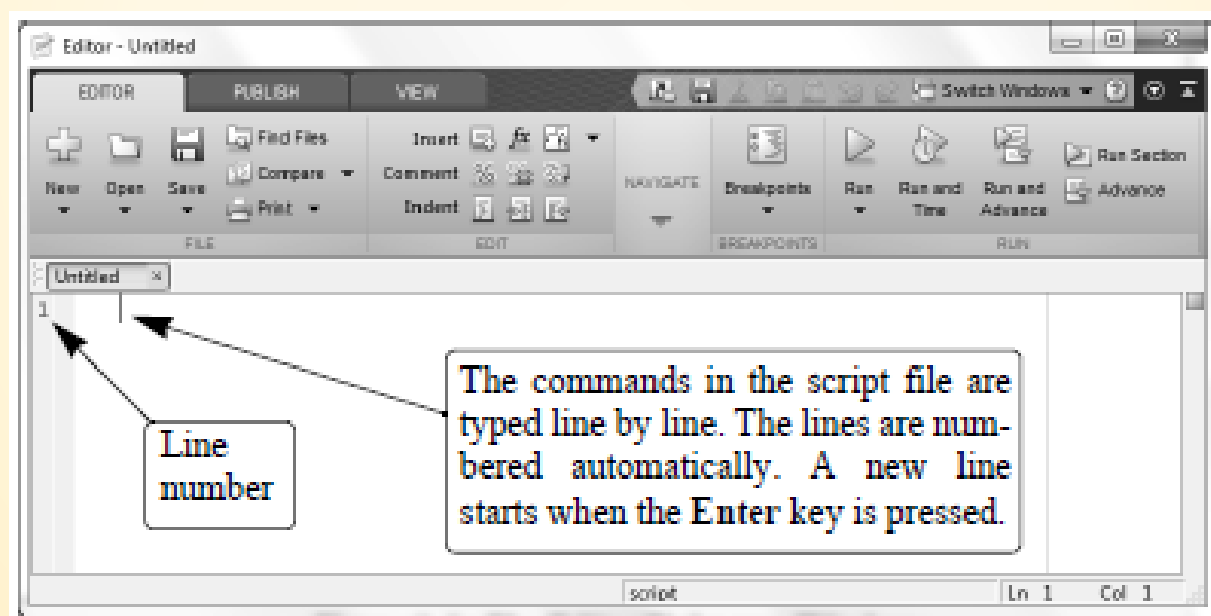


Figure 1-6: The Editor/Debugger Window.

## 1.8.2 Creating and Saving a Script File

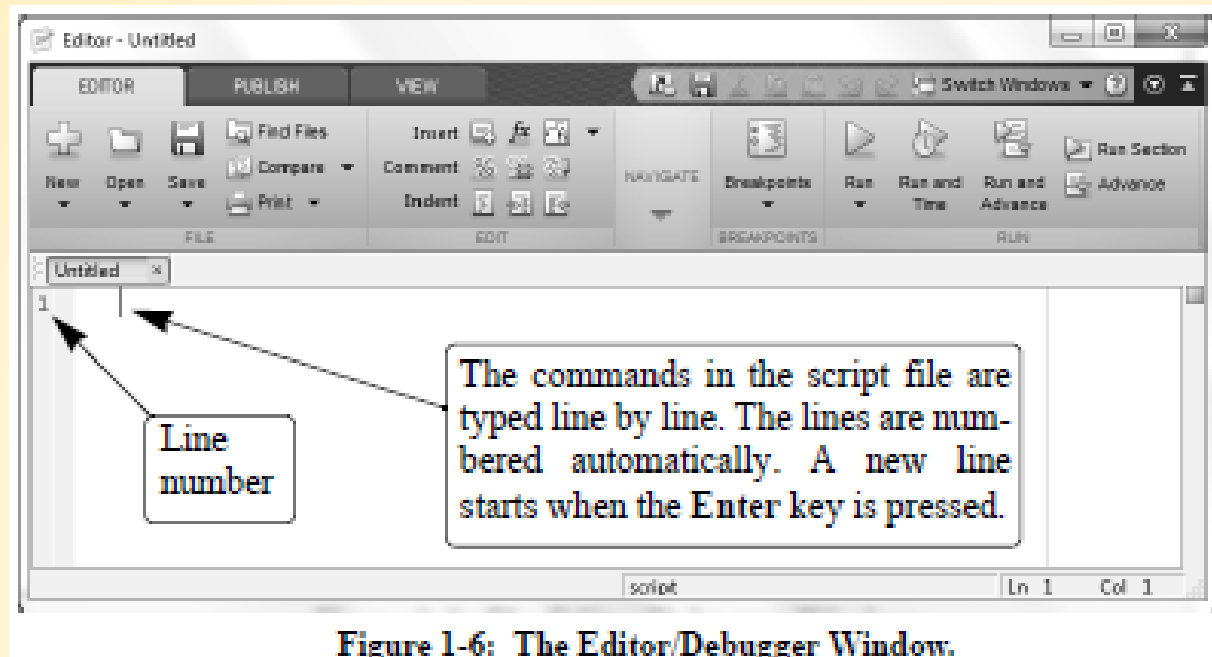


Figure 1-6: The Editor/Debugger Window.

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

## 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
- *What is the comment character in python?*



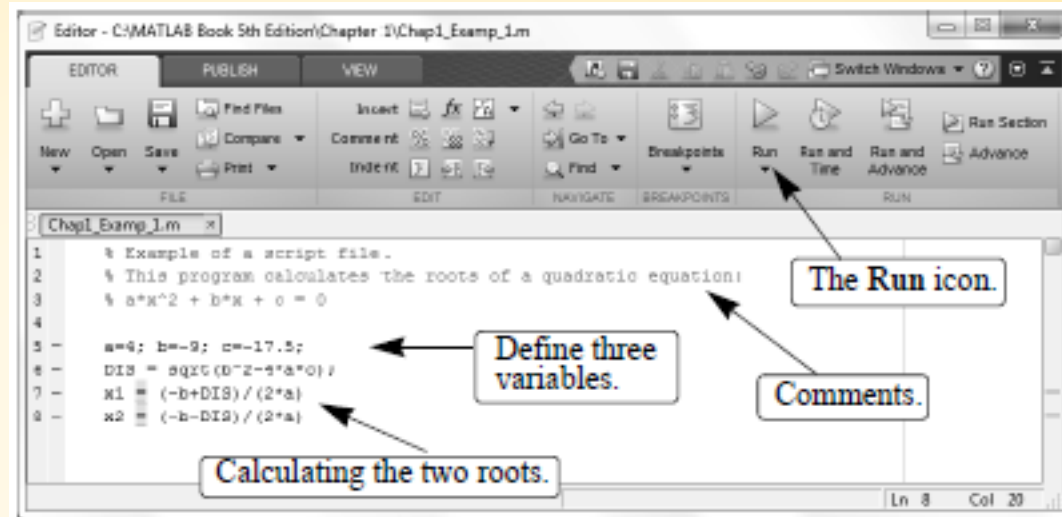


Figure 1-7: A program typed in the Editor/Debugger Window.

## 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

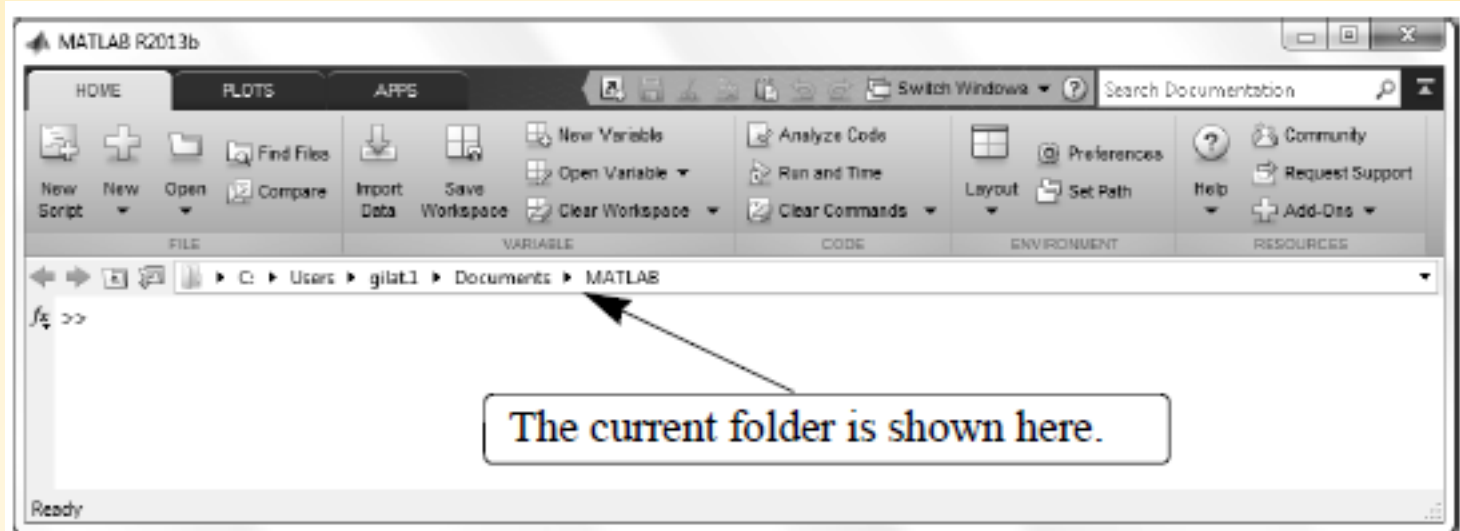
Before MATLAB can run commands in file, you must save file

- 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

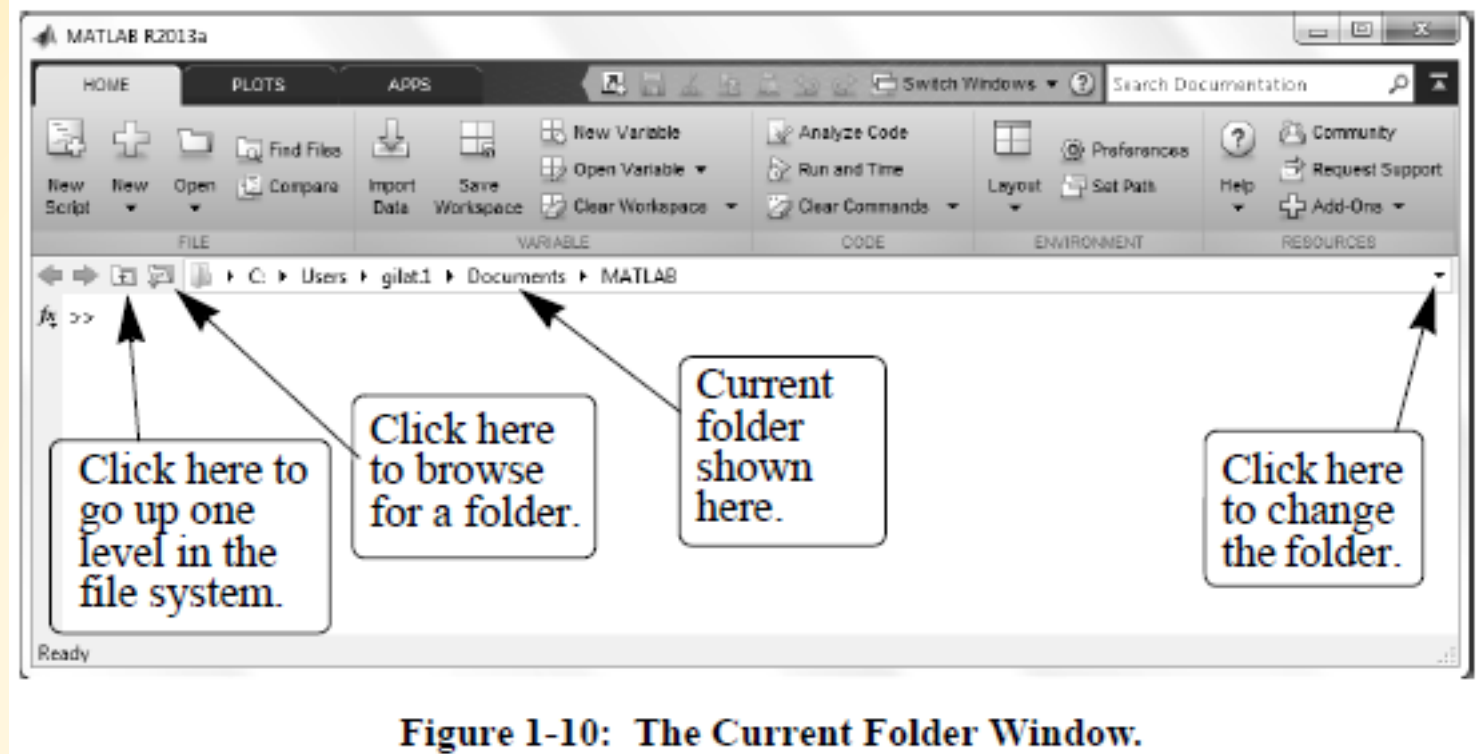
MATLAB will execute file if it is in MATLAB's current folder or if the file's folder is in the search path



**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

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'
```