

# EG-062: Development of Key Skills for Engineers – MATLAB Getting started

Dr Alison Williams

[alison.j.williams@swansea.ac.uk](mailto:alison.j.williams@swansea.ac.uk)

# Getting started with MATLAB

- MATLAB can be accessed through **Common Apps** on SU Unified Desktop



MATLAB  
icon

# MATLAB Environment



The screenshot shows the MATLAB R2013a interface. The top menu bar includes HOME, PLOTS, and APPS. Below it is a ribbon with various toolboxes like FILE, VARIABLE, CODE, SIMULINK, ENVIRONMENT, and RESOURCES. The main workspace is divided into several panes:

- Current Folder:** Shows a list of files in the current directory: forced.m, roots.m, test1.m, test2.m, test3.m, test4.m, and Untitled.m. A callout box explains its function.
- Command Window:** Displays the MATLAB prompt `>>` and a message about the Classroom License. A callout box explains its location and function.
- Command History:** Lists previously executed commands, including mathematical expressions and plotting functions. A callout box explains its purpose.

Arrows from the callout boxes point to the corresponding panes in the screenshot.

**Current folder**

- Indicates the folder that you are working in and files that it contains

**Command window**

- Located within the desktop
- Displays the prompt `>>`
- Displays a blinking cursor to the right of the prompt when the window is active
- Numerical results appear within this window

Can access **help** here or by typing **help** in the Command Window

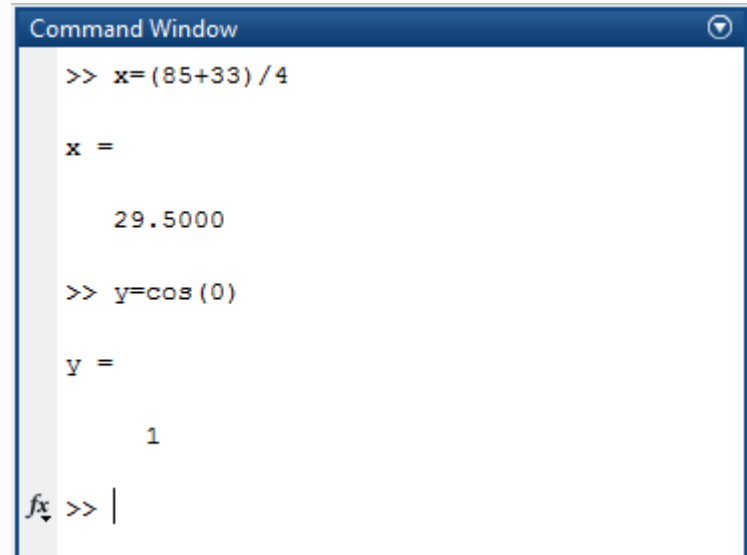
**Command history**

- Contains a list of all the commands that have been used

```
Command History
y=(10-x).^1/5)
y=(10-x).^1/5)-2
y=((10-x).^1/5)-2)
y=((10-x).^1/5)-2)./((+x.^2).^1/2))
plot(x,y)
clf
x=0:0.02:10
y=(x.^2+3)*sin(pi*x.^2)
y=(x.^2+3)*sin(pi*x.^2)
y=(x.^2+3).*sin(pi*x.^2)
z=(sin(pi
z=((sin(pi*x).^2)
z=((sin(pi*x).^2)./(x.^(-2)+3)
w=y.*z
plot(x,w)
%-- 14/11/2013 11:58 --%
%-- 14/11/2013 12:00 --%
%-- 05/09/2014 16:06 --%
%-- 12/09/2014 11:40 --%
```

# MATLAB as a calculator

- In the command window
  - MATLAB may be used as a calculator
  - Perform basic mathematical calculations by typing MATLAB statements
  - MATLAB statement consists of
    - Mathematical operators
    - Numbers
    - Mathematical functions
    - Variable names (perhaps)



```
Command Window
>> x=(85+33)/4

x =

    29.5000

>> y=cos(0)

y =

     1

fx >> |
```



# Arithmetic Operations

- Mathematical symbols are defined as shown below

Operation		Symbol	Example
Division:	$a \div b$	/	$1/10 = 0.1$
Multiplication:	$a \times b$	*	$2 * \text{pi} = 6.2832$
Addition:	$a + b$	+	$4 + 9 = 13$
Subtraction:	$a - b$	-	$6 - 4 = 2$
Exponentiation:	$a^b$	^	$3^4 = 81$

- **Note:** Different operators may be necessary for matrix/array calculations
  - These will be defined later



# Example 1 - MATLAB as calculator

```
Command Window
>> 4+5
ans =
    9
>> 6*9
ans =
   54
>> 50/6
ans =
  8.3333
>> 2^4
ans =
   16
fx >>
```

MATLAB calculates the expression and **assigns** the calculated value to a variable called "ans"

- The equals sign "=" means **"assigned to"**

In these examples the variable "ans" is reassigned during each calculation

# Arithmetic Operations and BODMAS

- MATLAB uses the BODMAS rule to work out the order in which a more complicated mathematical expression should be calculated
  - (B)rackets, (O)rder, (D)ivision, (M)ultiplication, (A)ddition, (S)ubtraction
- Example, what is the value of the following expression?

$$5 + 80/10 \times (2 + 3)^2 - 6$$

- BODMAS Rule
- (B)rackets
- (O)rder
- (D)ivision
- (M)ultiplication
- (A)ddition
- (S)ubtraction

$$(2 + 3)$$

$$(2 + 3)^2$$

$$80/10$$

$$80/10 \times (2 + 3)^2$$

$$5 + 80/10 \times (2 + 3)^2$$

$$5 + 80/10 \times (2 + 3)^2 - 6$$

This is the order in which the above expression is calculated (red denotes that it is the current operation)



# Arithmetic Operations and BODMAS (cont'd)

- $5 + 80/10 \times (2 + 3)^2 - 6$  should produce 199

```
Command Window
>> 5+80/10*(2+3)^2-6

ans =

    199

fx >>
```





# Variables

- A **variable** is something that can vary
- A value can be **assigned** to a variable using the equals symbol “=”
- For example,  **$v=60$**  means that  **$v$  is assigned a value of 60**
- Variables can be read or used by other variables
- Example:  **$s=v*t$**  means “**read  $v$  and  $t$ , multiply their values and assign the calculated value to  $s$** ”



## Example 2 - Variables

- Calculate the average of three module marks which are 45%, 60% and 65%

Here the average is calculated in two stages

1. Firstly, the sum of the marks is calculated
2. The sum is divided by the number of marks to calculate the average mark

Two variables are created and have values assigned to them

- “**sum**” and “**average**”

```
Command Window
>> sum=45+60+65

sum =

    170

>> average=sum/3

average =

    56.6667

fx >> |
```



## Example 2 - Variables (cont'd)

- This example could have been completed in one step by using one variable (or no variable at all)
  - $\text{average}=(45+60+65)/3$
- It could also have been completed by assigning each mark to a different variable and then calculating the average (see image)
  - This might be useful if you need to use the individual marks to calculate something else too.

```
Command Window
>> mark1=45

mark1 =

    45

>> mark2=60

mark2 =

    60

>> mark3=65

mark3 =

    65

>> average=(mark1+mark2+mark3)/3

average =

    56.6667

fx >> |
```



# Variables (2)

- Variable names must start with a letter
- Variables may contain letters, numbers and the underscore character ‘\_’
- Variable names are case sensitive
  - ‘A’ and ‘a’ are different variables
- There are certain keywords you can’t use for variable names
  - Typing **iskeyword** in the command window will show the list of keywords



# Built-in functions

- MATLAB provides a large number of mathematical functions

Function	MATLAB form
$\sin x$	sin(x)
$\cos x$	cos(x)
$\tan x$	tan(x)
$ x $	abs(x)
$\log_{10} x$	log10(x)
$\log_e x$	log(x)
$\log_2 x$	log2(x)
$e^x$	exp(x)
$\sqrt{x}$	sqrt(x)

- The argument of sin, cos, tan must be expressed in radians

# Be careful with built-in functions!

```
Command Window
>> sin=5
sin =
    5
>> sin(pi)
Subscript indices must either be real positive integers or
logicals.
>> sin=1:10
sin =
    1     2     3     4     5     6     7     8     9    10
>> sin(5)
ans =
    5
>> clear sin
>> sin(pi)
ans =
    1.2246e-16
fx >> |
```

Assigning a value of 5 to **sin**

Since **sin** has been assigned a constant value it is not possible to calculate **sin(pi)**

Assigning a list (array) of integers between 1 and 10 to **sin**

Now **sin(5)** will read the value of the 5<sup>th</sup> entry in the list (array) – in this case 5

**clear sin** removes the settings from memory

Now **sin** acts as a trig function and returns the standard value for **sin(pi)**. Note, it is not exactly zero due to numerical rounding.

# Statements

- Several statements may be included on the same line but must be separated by commas or semicolons
- If a statement is followed by a semicolon, the display to the screen is suppressed but the assignment is still performed
- Nothing is printed to the screen with the second statement but, for future use, the value of 1 has been stored in a, and the value of 3 has been stored in b.

```
Command Window
>> a=0.5*2,b=a*3

a =

     1

b =

     3

>> a=0.5*2;b=a*3;
fx >> |
```



# Useful commands

- To get help on a command, type **help** plus the command name
- General MATLAB help
  - **MATLAB help** on the pull down menu under **help** on the toolbar
- **who** provides a list of variables that have been created in the command window
- **clear** removes all variables from the memory
- **clc** clears the command window
- **clf** clears the graphics window
- **save filename** save all variables in current workspace into filename.mat
- **load filename** loads all variables from filename.mat into current workspace
- **Ctrl** key + **c** key interrupts the MATLAB statement execution





# Command line recall and editing

- Up/down arrows may be used to scroll through your list of previous commands
  - Allows a previous command line to be recalled, edited and a revised command submitted
- To edit a command line
  - Position cursor with left/right arrows
  - Backspace/Delete can be used to delete characters
  - Insert appropriate characters
- To submit a command line press **Return** key (↵)