

MATLAB
Fundamentals
2017
Part 1



These notes are derived from several sources (see below) including MathWorks documentation which can be found at <https://uk.mathworks.com/help/matlab/>

B.C. Lowe, An Introduction to Matlab

David F. Griffiths, Dundee University, An Introduction to Matlab, <http://www.maths.dundee.ac.uk/software/MatlabNotes.pdf>

David Houcque, Northwestern University, Introduction To Matlab For Engineering Students, <https://www.mccormick.northwestern.edu/documents/students/undergraduate/introduction-to-matlab.pdf>

Other Matlab Resources:

<https://ocw.mit.edu/resources/res-18-002-introduction-to-matlab-spring-2008/lecture-notes/>

Contents

1	Introduction	4
1.1	Running a MATLAB session	4
1.2	Navigating and Mistakes	5
1.3	MATLAB as a calculator	5
1.4	Formats	6
1.5	Variables	7
1.6	Built-in Functions	8
1.7	Comments	9

1 Introduction

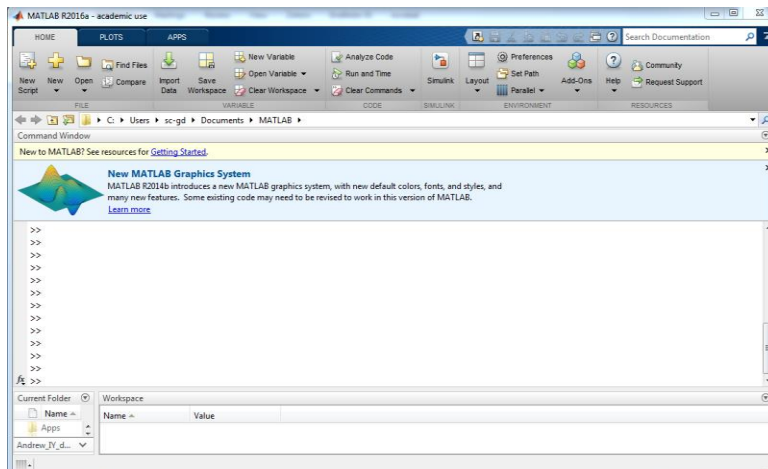
MATLAB is a software application developed by MathWorks that uses numerical computing methods to solve engineering, technical computing and scientific problems and since its initial release it has become very popular in both industry and academia. Matlab has grown enormously since its inception in 1984 and now has many add-ons called Toolboxes, some of which we will be using. These notes provide an introduction to the basic package. For a more detailed view of Matlab there are several texts available, for example, Matlab: A Practical Introduction to Programming and Problem Solving by S. Attaway.

1.1 Running a MATLAB session

Start every session by double clicking on the MATLAB icon on your desktop.



A MATLAB Command Window and two other windows appear and you can start using MATLAB immediately by typing commands at the » prompt.



1.2 Navigating and Mistakes

We all make mistakes. We can correct mistakes we make when using the command line using the cursor keys. The cursor keys $\rightarrow$ and $\leftarrow$ can be used to move backwards and forwards along a line. Delete or insert characters as required.

Previous MATLAB commands can be recalled by using the $\uparrow$ and $\downarrow$ cursor keys. Repeatedly pressing $\uparrow$ will review previous commands, starting with the most recent. To re-execute the command press the Enter key, editing it first if necessary. This is useful when you want to execute a command that is very similar to one used previously.

1.3 MATLAB as a calculator

The basic arithmetic operators are :-

addition	subtraction	multiplication	division	powers
+	-	*	/	^

and these are used with brackets if necessary, ()

Type commands after the `>>` prompt, for example,

```
>> 1+2+7
```

Matlab will print the following in response:

```
ans =  
      10.00  
Here's more:-  
>> 10+2*5  
ans =  
      20.00  
>> 24/12-4  
ans =  
      -2.00  
>> 24/(12-4)  
ans =  
       3.00
```

Note that we can do the three calculations at the same time:

```
1+2+7, 10+2*5, 24/12-4, 24/(12-4)  
ans =  
      10.00  
ans =  
      20.00  
ans =  
      -2.00  
ans =
```

3.00

MATLAB evaluates expressions using the usual BODMAS rules, i.e. it uses the following order of precedence:

- | | | |
|----|---------------------------------|-----------------------|
| 1. | quantities in brackets | $3*(10-3) = 3*7 = 21$ |
| 2. | powers | $3+6^2 = 3 + 36 = 39$ |
| 3. | *, / working from left to right | $12/6*2 = 2*2 = 4$ |
| 4. | +, - working from left to right | $1+7 - 3 = 8 - 3 = 5$ |

Examples:

```
>> 5+3^2*5, (5+3^2)*5, 5+4/2+1, (5+4)/(2+1)
```

```
ans =
```

```
50
```

```
ans =
```

```
70
```

```
ans =
```

```
8
```

```
ans =
```

```
3
```

1.4 Formats

Matlab uses the [IEEE 754](#) floating point standard in calculations but the default is to print numbers between 0.001 and 1000 correct to 4 decimal places and other numbers in exponential form, so

```
» 200/3, 20000/3, 0.0004/3
```

```
ans =
```

```
66.6667
```

```
ans =
```

```
6.6667e+003
```

this means $6.6667 \times 10^3 = 6666.7$

```
ans =
```

```
1.3333e-004
```

this means $1.3333 \times 10^{-4} = 0.00013333$

The way numbers are presented can be controlled by the **format** command.

Type **» help format** to see a list of format commands.

Obtain the following – remember to type *format* commands in lower case.

```
» format bank
```

```
» 200/3
```

```
ans =
```

```
66.67
```

```
» format long
```

```
» 200/3
```

```
ans =  
    66.66666666666667
```

format compact reduces the number of blank lines.

To return to the default format:

```
» format
```

If you require higher precision, use the `vpa` and `digits` function in Symbolic Math Toolbox.

1.5 Variables

```
» 8+2^3  
ans =  
    16  
» 3*ans  
ans =  
    48
```

The result of the first calculation is called *ans* and is used in the second calculation, after which its value is updated to 48. It could be used in further calculations. *ans* is an example of a variable.

We can use our own names to store numbers. For example:

```
» a = 3;b = 5;  
» x = 3*a+2*b,y = a^2 + b^2  
x =  
    19  
y =  
    34
```

(Note that the use of a semi-colon ; in the first line stops the values of *a* and *b* being printed in the succeeding lines)

The values of *a*, *b*, *x* and *y* can be used in subsequent calculations.

They are called **variables** and the expressions such as *a* = 3 are examples of **assignment statements**.

Variable names can be any combination of letters and digits, starting with a letter. MATLAB distinguishes between upper and lower case, so *n* and *N* are different variables.

The following are valid variable names: *a*, *x*, *y1*, *max*, *Temp*, *cost*, *R2D2*

But the following are **not** allowable: *2x*, *Max-temp*, *#ab*, *a%*, *(ABC)*

Use meaning names for variables. For example, V for voltage, t or *time* for time, and f or *freq* for frequency.

You should avoid certain names as they are already in use. For example, π is used for $\pi = 3.14159\dots$ and i and j for $\sqrt{-1}$. Also, do not use *ans*.

If you start a new section or problem it is a good idea to start with

```
» clear all
» format short
```

This clears all variables, preventing you from using a previous value erroneously. For example:

```
» qq=24
qq =
    24
» clear all
» qq
??? Undefined function or variable 'qq'.
```

1.6 Built-in Functions

Some useful functions are:

Function	calculates	condition/comment
$\text{sqrt}(x)$	$\sqrt{x}$	$x \geq 0$
$\log(x)$	$\ln x = \log_e x$	$x > 0$
$\log_{10}(x)$	$\log_{10} x$	$x > 0$
$\exp(x)$	e^x	
$\text{round}(x)$		rounds to nearest integer
$\sin(x)$	$\sin x$	x in radians
$\cos(x)$	$\cos x$	x in radians
$\tan(x)$	$\tan x$	x in radians
$\text{asin}(x)$	$\sin^{-1} x$	returns value in range $-\frac{\pi}{2}$ to $\frac{\pi}{2}$
$\text{acos}(x)$	$\cos^{-1} x$	returns value in range 0 to π
$\text{atan}(x)$	$\tan^{-1} x$	returns value in range $-\frac{\pi}{2}$ to $\frac{\pi}{2}$

Try the following and check with your calculator:

```
» sqrt(7), log(10), exp(1.52), sin(0.841), acos(1), round(200/3)
ans =
    2.6458
ans =
    2.3026
ans =
    4.5722
```

```
ans =
    0.7453
ans =
    3.1416
ans =
    67
```

Trigonometric functions use radians as the units for angles. To work in degrees rather than radians in the trigonometric functions the argument must be converted using π radians = 180° . Type `pi` to obtain π . For example, to find
(i) $\cos 43^\circ$ and (ii) $\tan^{-1} 1.59$ in degrees:

```
» cos(43*pi/180), atan(1.59)*180/pi
ans =
    0.7314
ans =
    57.8329
```

Functions can be nested, so to find $\sin \sqrt{0.63}$ and $\sqrt{1 + e^{\cos^{-1} 0.25}}$:

```
» x = sin(sqrt(0.63)), y = sqrt(1+exp(acos(0.25)))
x =
    0.7130
y =
    2.1763
```

Note You can see a list of available functions in Help. Click on the help icon on the toolbar and use the search facility there.

1.7 Comments

Comments are preceded with % and MATLAB ignores anything on the line after %. You use comments to help describe what is going on to anyone who is looking at your work. **Always** comment your Matlab script and function files (see later). Examples are.

```
» a = base*height/2      % find the area of a triangle
» % draw the graphs of sin x and cos x
» % x takes values from 1 to 5 in steps of 0.01
```

These will be included where appropriate from here on.

Exercise 1

1. Evaluate each of the following using MATLAB. Try to predict the answer first.
 $1 + 2 + 6$, $6 - (4 - 2)$, $6 \times 5 - 12 / 2$, $(2 + 3)^2 - 4 \times 2$
2. Predict the results of the following, and check.
 $24 / (2 \times 3)$, $24 / 2 \times 3$, $24 / 2 / 3$.
What is the value of $1000 / 2 / 4 / 5$? How else could it be written?
3. Calculate $\frac{10}{7}$ correct to (i) 2D (ii) 4D (iii) 15D.
4. Use the default format to find (i) π (ii) 1000π (iii) $\frac{\pi}{100}$ (iv) $\frac{\pi}{10000}$.
(Type pi to obtain π)
5. Find $\sqrt{174}$, $\ln 30$, $\log_{10} 21.6$, $e^{2.07}$ correct to 4D.
Round the first two of these to the nearest integer.
6. (i) Find $\sin 0.8$, $\cos 1.26$, $\tan 2.35$, $\sin^{-1} 0.763$, $\cos^{-1}(-0.47)$, $\tan^{-1}(-3.17)$,
where all angles are in radians.
(ii) Find $\sin 27^\circ$, $\cos 138.5^\circ$, $\tan(-67^\circ)$, and find, to the nearest degree,
 $\sin^{-1} 0.721$, $\cos^{-1}(-0.0488)$, $\tan^{-1} 1.233$.
7. Find $\frac{1}{\sqrt{31.493}}$, $e^{\sqrt{1+\ln 4}}$, $\sin(2 \tan^{-1} 1)$, $(\log_{10}(3 \sin^{-1} 0.7))^2$.
Show that $e^{\sqrt{\ln 162 \times \cos 54^\circ}} = 6$, to the nearest integer.
8. Which of the following are valid variable names?
 p , $5q$, PQr , *henry*, *jan20*, *jan_20*, *20jan*, *Area*, *£pounds*, *Minimum*
9. Set variables $p = 5$, $q = 3.7$. Calculate $r = 3p + 5q$, $s = \sqrt{p^2 + q^2}$,
 $t = \ln(rs)$, $u = \sqrt{p + q}$, $v = \frac{u^3}{t}$. You should obtain $v = 4.8061$.