Islamic University of Technology (IUT)

Organization of Islamic Cooperation (OIC)

Department of Electrical and Electronic Engineering (EEE)

COURSE NO	:	EEE 4416
LAB NO	:	01 (Part – B)
TOPIC	:	FAMILIARIZATION WITH THE MATLAB ENVIRONMENT

Variables

A variable is a name that represents a value stored in memory. You can create and use variables to store data such as numbers, arrays, strings, or more complex data types like structures and cell arrays. When a new variable is defined, MATLAB allocates appropriate memory space for that variable.

Once a variable is assigned a numerical value, it can be used in mathematical expressions, in functions, and in any MATLAB statements and commands. Any variable created will appear in MATLAB 'Workspace'. You can track all the created variables through there.

Creating Variables

You can assign a value to a variable using the = operator. Create the following variables using the command window.

```
x = 5

name = 'Xavier'

A = [1, 2, 3; 4, 5, 6]
```

Variable Naming Rules

- Must start with a letter.
- Can contain letters, digits, and underscores.
- Cannot use reserved keywords (like for, if, etc.). Reserved keywords will appear in purple in MATLAB.
- MATLAB is case-sensitive (Age and age are different).

Check Variable Info

- whos: Lists all variables with size and type.
- class(x): Returns the class/type of the variable.

Prepared by: Asif Newaz, A. K. M. Rakib
Lecturer, EEE, IUT

Variable Types

MATLAB variables can be categorized into 3 different types:

- **Global:** available everywhere (shown in Workspace)
- **Local:** only inside a user-defined function (not shown in Workspace)
- **Persistent:** used for caching. It's a type of local variable inside a function that remembers its value between function calls**.** Unlike normal local variables, which get reinitialized every time the function is called, a persistent variable keeps its last value in memory.

Predefined Variables and Keywords

Variable	Description	Value / Notes
ans	Stores the result of the most recent expression that wasn't assigned to a variable	Changes each time you enter an unassigned expression.
pi	Ratio of a circle's circumference to its diameter	3.141592653589793…
eps	Floating-point relative accuracy: the smallest increment between two distinct numbers	$2^{(-52)} \approx 2.2204\times10^{-16}$
inf	Represents positive infinity (e.g., result of 1/0)	Inf
i	Imaginary unit	0 + 1.0000i
j	Same as i (second name for the imaginary unit)	0 + 1.0000i
NaN	"Not-a-Number"; used when a result is undefined or unrepresentable (e.g., 0/0)	NaN

Numeric Types

MATLAB supports different numeric types. Each type has specific precision, storage size, and range. The default is 'double', and it is used in general. In certain applications, you may need to use different numeric types. For instance, while representing an image, uint8 is used.

Type	Description	Size (Bytes)	Approximate Value Range
double	Double-precision float (default)	8	$\pm1.7 \times 10^{308}$ (15 digits precision)
single	Single-precision float	4	$\pm3.4 \times 10^{38}$ (7 digits precision)
int8	8-bit signed integer	1	-128 to 127
uint8	8-bit unsigned integer	1	0 to 255
int16	16-bit signed integer	2	-32,768 to 32,767
uint16	16-bit unsigned integer	2	0 to 65,535
int32	32-bit signed integer	4	-2,147,483,648 to 2,147,483,647
uint32	32-bit unsigned integer	4	0 to 4,294,967,295
int64	64-bit signed integer	8	$\pm9.2 \times 10^{18}$
uint64	64-bit unsigned integer	8	0 to 1.8×10^{19}

Prepared by: Asif Newaz, A. K. M. Rakib
Lecturer, EEE, IUT

```
x = 10;

y = int32(x);  % Convert to 32-bit signed integer

z = single(x); % Convert to single-precision float
```

MATLAB also supports complex numbers:

```
z = 3 + 4i;          % Complex number

realPart = real(z);   % 3

imagPart = imag(z);   % 4

magnitude = abs(z);   % 5

angleRad = angle(z);  % Phase angle
```

Arithmetic Operations

You can perform all arithmetic operations using MATLAB.

Operation	Symbol	Example
Addition	+	5+3
Subtraction	-	5-3
Multiplication	*	5*3
Right Division	/	5 / 3
Left Division	\	5 \ 3 = 3 / 5
Exponentiation	^	5 ^ 3 =125
Natural logarithm (base e)	log(x)	log(3)
Common logarithm (base 10)	log10(x)	log10(3)
Binary logarithm (base 2)	log2(x)	log2(3)
exponential function (e^x)	exp	exp(5) = 148.4
Square root	sqrt	sqrt(4) = 2
Trigonometric function	sin, cos, …	sin(pi)

MATLAB offers some simple built-in functions to work with decimal point (floating-point) numbers. These functions help in rounding, truncating, and formatting decimal values.

Function	Description	Example
`round(x)`	Rounds to the nearest integer	`round(3.6)` → `4`
`floor(x)`	Rounds down to nearest integer	`floor(3.6)` → `3`
`ceil(x)`	Rounds up to nearest integer	`ceil(3.2)` → `4`
`fix(x)`	Rounds toward zero	`fix(-3.8)` → `-3`
`mod(x, y)`	Modulus (remainder after division)	`mod(10.5, 3)` → `1.5`
`rem(x, y)`	Remainder after division (same sign as dividend)	`rem(10.5, 3)` → `1.5`
`abs(x)`	Absolute value	`abs(-4.7)` → `4.7`
`sign(x)`	Sign of number: -1, 0, or 1	`sign(-3.2)` → `-1`

- ⇨ Round (pi) = 3
- ⇨ Round (pi, 4) = 3.1416
- ⇨ Round (pi, 3) = 3.142

Display Formats

Display formats control how numeric values are shown in the Command Window, but they do not affect the actual value stored in variables — only how the values are displayed.

Command	Description	Example (with `pi` and `1234567`)
`format short`	Fixed-point, 4 decimal digits	`pi` → `3.14161234567` → `1234567.0000`
`format long`	Fixed-point, 15 decimal digits	`pi` → `3.141592653589793`
`format short e`	Scientific notation, 4 digits	`pi` → `3.1416e+001234567` → `1.2346e+06`
`format long e`	Scientific notation, 15 digits	`pi` → `3.141592653589793e+00`
`format short g`	General format: up to 5 significant digits, auto-switches format	`pi` → `3.14161234567` → `1.2346e+06`
`format long g`	General format: up to 15 significant digits, adaptive	`pi` → `3.141592653589791234567` → `1.234567e+06`
`format bank`	Fixed-point with 2 decimal places (for currency)	`pi` → `3.14`
`format rat`	Rational approximation	`pi` → `355/113`
`format +`	Displays +, -, or blank for sign	`pi` → `+`

Prepared by: Asif Newaz, A. K. M. Rakib
Lecturer, EEE, IUT

Homework

A series RLC circuit is driven by a sinusoidal source at a frequency f = 1 kHz. The circuit parameters are:

- Resistance: R=50 Ω
- Inductance: L=100 mH
- Capacitance: C =10 μF

i. Compute the angular frequency
ii. Compute the inductive reactance
iii. Compute the capacitive reactance
iv. Determine the complex impedance
v. Find the magnitude of the impedance
vi. Determine the phase angle (in degrees)
vii. Calculate the resonant frequency ($\frac{1}{2\pi\sqrt{LC}}$)

Miscellaneous

- **The semicolon:** If a semicolon (;) is typed at the end of a command, the output of the command is not displayed in the command window.
- **Commenting using %:** When the symbol % (percent) is typed at the beginning of a line, the line is designated as a comment. The output will be suppressed.
- **Removing variable:** You can remove a variable from the workspace using the clear function.

```
a = 2*pi;

clear a   % this removes the 'a' variable
```

Prepared by: Asif Newaz, A. K. M. Rakib
Lecturer, EEE, IUT