|  |
| --- |
| THINK AND INK EDUCATION AND RESEARCH FOUNDATION  **THINK...**  **INK.**  **LINK.**  **Workshop on Matlab**  **at**  **ReVA UNIVERSITY**  **ECE 2nd semester**  **DIRECTOR**  **Prof. B. A. Patil**  **Shwetha A**  **Senior Research Engineer**  **Thinksoft Research and Information Technologies** |
| |  | | --- | | curiecvramanIrène Joliot-CurieImage result for Nikola Tesla  **Think & Ink Education & Research Foundation, # 180, CQAL Layout, Sahakarnagar,**  **Bengaluru- 560092**  **Email:**[**info@thinksciences.org**](http://thinksciences.org/team.html)**, Website: www.thinksciences.or** | |
|  |
| **Think & Ink Education & Research Foundation**  **Thinksoft Research and Information Technologies** |
|  |
| |  |  | | --- | --- | | **Material** | **Matlab Course Material** | | **Course ID** | **Thinksoft/Reva/1year/ECE-Matlab-2019-101** | | **University** | **Reva University, \BANGALORE** | | **Director** | **Prof.B.A.Patil** | | **Prepared By** | **Shwetha A** | |

**Module 1 Day1 Course Material**

**Introduction to MATLAB**

What is MATLAB?

The name MATLAB stands for MATrix LABoratory. MATLAB was written originally

to provide easy access to matrix software developed by the LINPACK (linear system package)

and EISPACK (Eigen system package) projects.

What are the application of MATLAB?

* MATLAB is a high-performance language for technical computing.
* It integrates *computation*, *visualization*, and *programming* environment.
* It has sophisticated *data structures*, contains built-in editing and *debugging tools*,
* MATLAB supports *object-oriented programming*.

These factors make MATLAB an excellent tool for teaching and research.

Advantages of MATLAB

MATLAB has many advantages compared to conventional computer languages (e.g.,

C, FORTRAN) for solving technical problems. MATLAB is an interactive system whose

basic data element is an *array* that does not require dimensioning.

It has powerful *built-in* routines that enable a very wide variety of computations. It

also has easy to use graphics commands that make the visualization of results immediately

available. Specific applications are collected in packages referred to as *toolbox*. There are

toolboxes for signal processing, symbolic computation, control theory, simulation, optimiza-

tion, and several other fields of applied science and engineering.

Basic features

As we mentioned earlier, the following tutorial lessons are designed to get you started

quickly in MATLAB. The lessons are intended to make you familiar with the basics of

MATLAB. We urge you to complete the exercises given at the end of each lesson.

A minimum MATLAB session

The goal of this *minimum* session (also called *starting* and *exiting* sessions) is to learn the

* first steps:
* How to log on
* Invoke MATLAB
* Do a few simple calculations
* How to quit MATLAB

1.3.1 Starting MATLAB

After logging into your account, you can enter MATLAB by double-clicking on the MATLAB

shortcut *icon* (MATLAB 7.0.4) on your Windows desktop. When you start MATLAB, a

special window called the MATLAB desktop appears. The desktop is a window that contains

*other* windows. The major tools within or accessible from the desktop are:

* The Command Window
* The Command History
* The Workspace
* The Current Directory
* The Help Browser
* The Start button

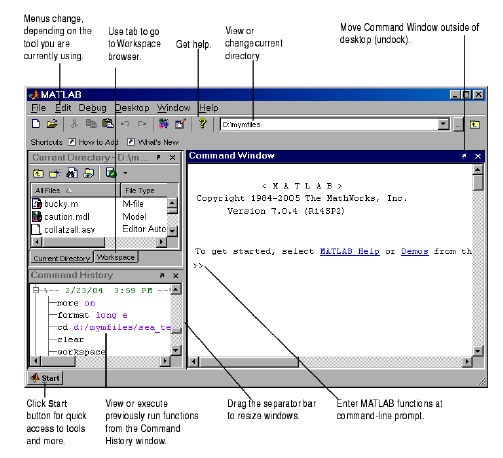


Figure 1.1: The graphical interface to the MATLAB workspace

When MATLAB is started for the first time, the screen looks like the one that shown

in the Figure 1.1. This illustration also shows the default con¯guration of the MATLAB

desktop. You can customize the arrangement of tools and documents to suit your needs.

Using MATLAB as a calculator

As an example of a simple interactive calculation, just type the expression you want to

evaluate. Let's start at the very beginning. For example, let's suppose you want to calculate

the expression, 1 + 2 *£* 3. You type it at the prompt command (>>) as follows,

>> 1+2\*3

ans =

7

You will have noticed that if you do not specify an output variable, MATLAB uses a

default variable ans, short for answer, to store the results of the current calculation. Note

that the variable ans is created (or overwritten, if it is already existed). To avoid this, you

may assign a value to a variable or output argument name. For example,

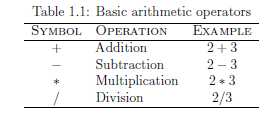
>> x = 1+2\*3

x =

7

Before we conclude this minimum session, Table 1.1 gives the partial list of arithmetic

operators.



Getting started

After learning the minimum MATLAB session, we will now learn to use some additional

operations.

1.4.1 Creating MATLAB variables

MATLAB variables are created with an assignment statement. The syntax of variable as-

signment is

variable name = a value (or an expression)

For example,

>> x = expression

where expression is a combination of numerical values, mathematical operators, variables,

and function calls. On other words, expression can involve:

* manual entry
* built-in functions
* user-defined functions

Overwriting variable

Once a variable has been created, it can be reassigned. In addition, if you do not wish to

see the intermediate results, you can suppress the numerical output by putting a semicolon

(;) at the end of the line. Then the sequence of commands looks like this:

>> t = 5;

>> t = t+1

t =

6

Error messages

If we enter an expression incorrectly, MATLAB will return an error message. For example,

in the following, we left out the multiplication sign, \*, in the following expression

>> x = 10;

>> 5x

??? 5x

|

Error: Unexpected MATLAB expression.

**Module 2**

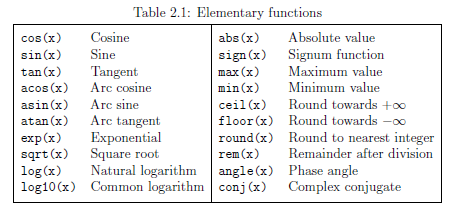
**Mathematical functions**

MATLAB offers many predefined mathematical functions for technical computing which

contains a large set of mathematical functions. Typing help elfun and help specfun calls up full lists of *elementary* and *special* functions respectively. There is a long list of mathematical functions that are *built* into MATLAB. These functions are called *built-ins*. Many standard mathematical functions, such as sin(*x*), cos(*x*), tan(*x*), *ex*, *ln*(*x*), are evaluated by the functions sin, cos, tan, exp, and log respectively in MATLAB.

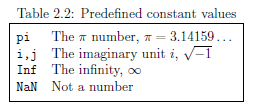
Table 2.1 lists some commonly used functions, where variables x and y can be numbers,

vectors, or matrices.



In addition to the elementary functions, MATLAB includes a number of predefined

constant values. A list of the most common values is given in Table 2.2.



Examples

We illustrate here some typical examples which related to the elementary functions previously defined.

As a first example, the value of the expression y = e¡a sin(x) + 10p y, for a = 5, x = 2, and

y = 8 is computed by

>> a = 5; x = 2; y = 8;

>> y = exp(-a)\*sin(x)+10\*sqrt(y)

y =

28.2904

The subsequent examples are

>> log(142)

ans =

4.9558

>> log10(142)

ans =

2.1523

Note the difference between the natural logarithm log(x) and the decimal logarithm (base 10) log10(x).

To calculate sin(¼=4) and e10, we enter the following commands in MATLAB,

>> sin(pi/4)

ans =

0.7071

>> exp(10)

ans =

.2026e+004

Notes:

* Only use built-in functions on the right hand side of an expression. Reassigning the

value to a built-in function can create problems.

* There are some exceptions. For example, i and j are pre-assigned to *p¡*1. However,

one or both of i or j are often used as loop indices.

* To avoid any possible confusion, it is suggested to use instead ii or jj as loop indices.