## What is programming?

Programming the computer is the art of producing code in a standard computer language, such as Fortran, BASIC, Java, etc., to control the computer for calculations, data processing, or controlling a process. For details on programming structures, use of flowcharts and pseudo code, and some simple coding in MATLAB, Scilab, and Visual Basic.

## What do I need to program the computer?

To type code you need an editor, which is simply a program where you can edit text files. Modern programming environments such as Matlab, Scilab, or Visual Basic include an editor for typing code.

Once the code is typed and the corresponding file saved, you need to have either a compiler (as in the case of Fortran) or an interpreter as in the case of Visual Basic (although, compilation is also allowed). A compiler translates the contents of a program's code into machine language, thus making the operation of the program faster than interpreting it. An interpreter collects as much code as necessary to activate an operation and performs that portion of the code before seeking additional code in the source file.

## Introduction to the MATLAB

**MATLAB is:** higher program language oriented in solving technical problems with integrated interactive developing environment. Programming language in MATLAB is called M-code.

## Short history of MATLAB

MATLAB has grown due to needs of technical computing in the year 1980, when the university teacher of mathematics Cleve Moler from Stanford University wrote the basics of MATLAB for linear algebra – that is how to calculate with matrices (adding, subtracting, multiplying, dividing… ) and how to solve linear systems. The Figure 1.1 shows the example of magic matrix with dimensions 5 x 5, where the sum of rows, columns and diagonals are equal. Check it, if it is true!

17 24 1 8 15

23 5 7 14 16

4 6 13 20 22

10 12 19 21 3

11 18 25 2 9

Figure 1: Magic matrix with dimensions 5 x 5

Cleve Moler discovered that program languages are not user friendly for his needs to teach mathematics, soon. If students would use the program languages FORTRAN, Pascal, C, etc, they would lose a lot of time programming and learning these languages. This is the reason, why he had started to develop MATLAB, as interactive calculator, without possibility to program, just to calculate matrices etc.

The first version of MATLAB was written in FORTRAN and it used the libraries LINPACK and EISPACK to calculate matrices. The next stage in the development of MATLAB was the year 1983, when Jack Little and Steve Bangert joined Cleve Moler and programmed the newer version of MATLAB in C language. They also added the possibility of interpreter programming in so called M-code and some other characteristics. They called this version of MATLAB – MATLAB 1.0 and was published on the market in the year 1984, when the company MATHWORKS was established. The company MATHWORKS has developed several newer versions from year to year. They introduced first version of SIMULINK, which offer also graphical programming in year 1990. The usability of MATLAB and SIMULINK is even improved by so called Toolboxes and Block sets, which have been available and improved from year 1990.

## The characteristics of MATLAB are:

Program language MATLAB has dynamic memory management, therefore there is no need to define in advance the names and types of variables, like in C.

- MATLAB development environment allows users interactive experimenting with the problem to be solved, as well as writing a longer program in M-code.
- An interpreter of M-code is not a classical translator, as case of C-language. The interpreter translates every single command (row) and executes it immediately after was confirmed by type of Enter on a keyboard. In case of a classical translator, the user has to write the complete program and after that the program is translated and then in the second stage executed. Therefore the MATLAB interpreter allows an interactive work.
- The time needed for developing program in M-code is pretty lower than in case of programming in Pascal, FORTRAN or in C-language.
- Every program written in M-code is interpreted and not translated in classical way, so the execution is slow.
- Of course, the MATLAB developing environment supports not only interpreter mode, but also a classical two stage translating. The classical translating is only used in the case, when the development of the program in M-code is completely finished. Then the execution of the program becomes faster.
- The drawback of the classical translated M-code is that this translated code is not transferrable to other microprocessors (PIC, ATMEL, etc), as it is possible in case of translated C-code.

### The MATLAB Desktop

When you start MATLAB especial window called MATLAB desktop appears. The default configuration of the MATLAB desktop is shown in the figure 1.

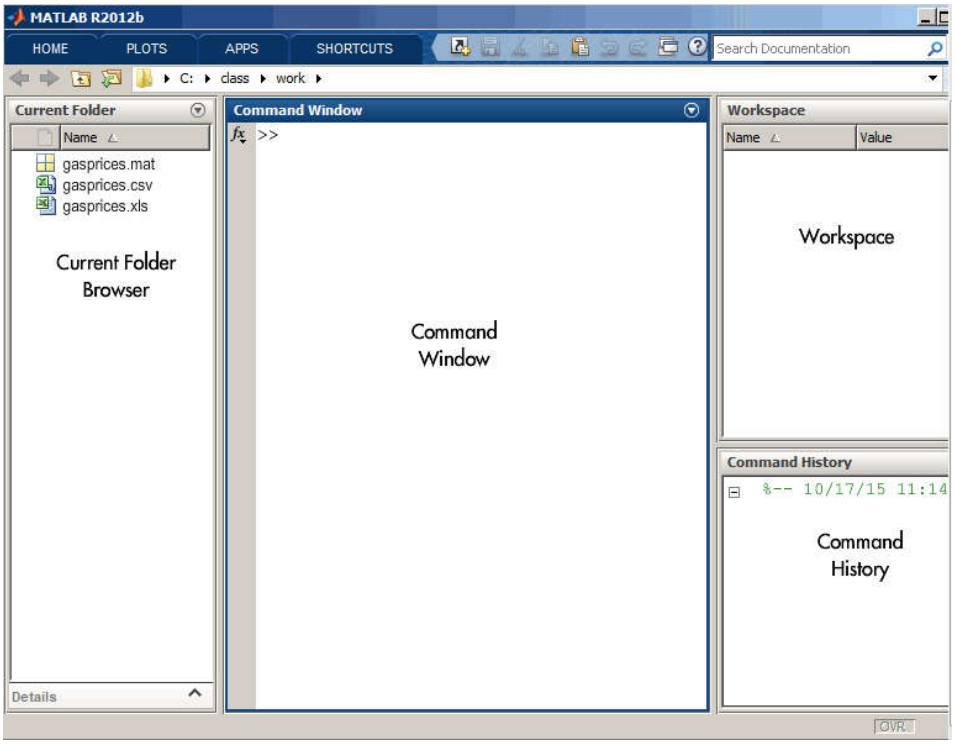


Figure 1. the MATLAB desktop. The exact appearance of the window may differ slightly on the different types of computers.

Generally the desktop consist of main four main windows:

1- The Command window
2- The Commend history window
3- The work space window
4- The current folder browser window

## Command Window

The right side of the MATLAB desktop contains the commend window. It used to enter commends and instructions to the MATLAB. In the Command Window, you should see:

>>

The >> is called the prompt. In the Command Window, MATLAB can be used interactively. At the prompt, any MATLAB command or expression can be entered, and MATLAB will immediately respond with the result. For example we can calculate the sum of tow numbers

```
>> 5 + 2
ans =
     7
fx >>
```

It is also possible to write programs in MATLAB that are contained in script files or M-files. Programs will be introduced later.
You can clear commend window by using clc (clear commend).

```
>> 5+2
ans =
     7
fx >> clc
```

## Commend History Window

the commend history window records the commends you issued in the command window in chronological sequence. When you exit MATLAB or when you issue the (clc) instruction, the commend listed in the command window are cleared. However , the command history window retain a list of all the commands you issued. You can clear the command history by using Edit menu if you need, by selecting edit and then clear command window.

### The Work Space Window

The work space is the collection of all variables and arrays that can be used by MATLAB when a particular command, M- file or function is executed. If you don't creat any variable yet, the work space window just show one variable, ans. As we see in the figure 2, The columns in the window display the name of variable, its current value, and variable type indicate in the class.

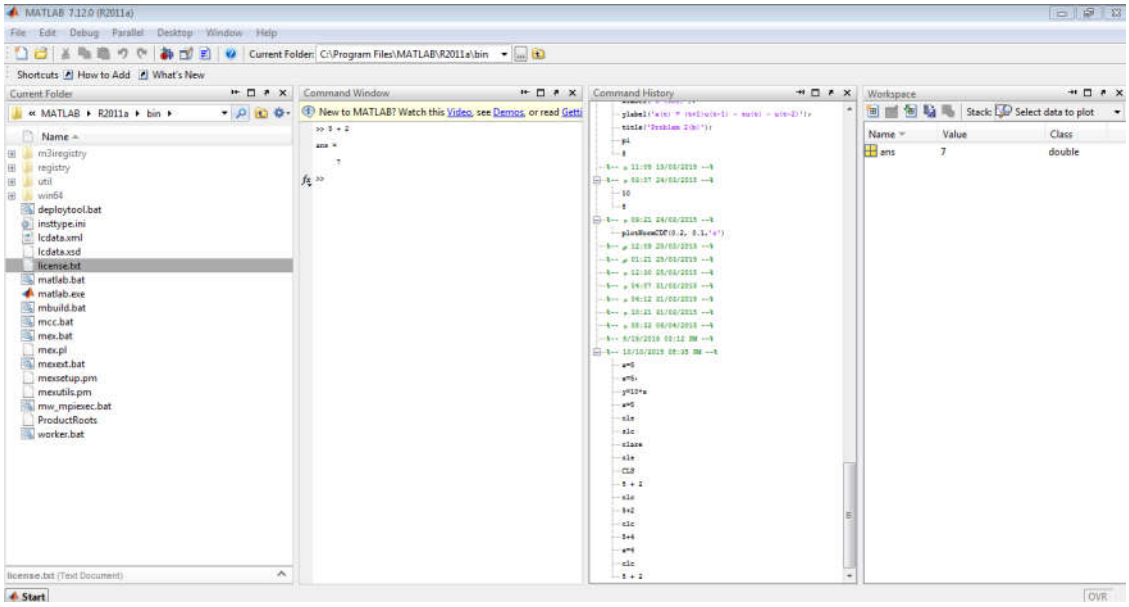

Figure 2

## The Current Folder Browser Window

When MATLAB accesses files from and saves information to your hard drive, it uses the current directory. The default for current directory depends on your version of the software and how it was installed. The current directory as we see in the figure 3, is listed at the top of main window. It can changed by selecting another directory from the drop- down list

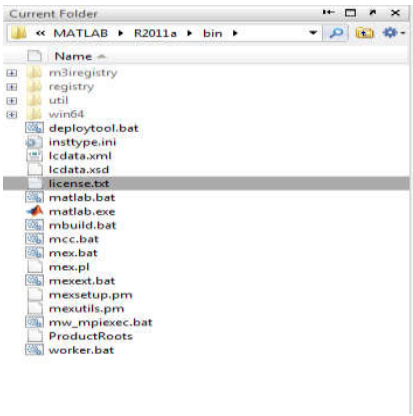


Figure 3

## Solving problems in MATLAB

The command window environment is a powerful tool for solving engineering problems. To use is effectively, you will need to understand more about how MATLAB works.

### 1- Variable

In MATLAB we assign names to the scalar, vectors and matrices we use. The following rules apply to these variable names:

- Variable names must start with letter.
- Variable names are case sensitive. The names **time**, **Time** and **TIME** all represent different variables.
- Other than the first letter, variable names can contain letters, digits, and the underscore ( _ ) character. To test whether a name is a legitimate variable name, use the **isvarname** command. The answer 1 mean true, and answer 0 means false. For example,

```
>>isvarname vector


Ans =

      1
```

- Variable name can be any length, but only the first N characters are used by MATLAB. The value of N varies, depending on the version of MATLAB that are using. For version 7, the value of N is 63. You can see the value of N on your system by typing

```
namelengthmax


ans =


   63
```

- Variable cannot have the same name as any MATLAB's keywords. To see a list of all MATLAB's keywords, type

```
>>iskeyword
ans =


   'break'
   'case'
   'catch'
```

'classdef'
'continue'
'else'
'elseif'
'end'
'for'
'function'
'global'
'if'
'otherwise'
'parfor'
'persistent'
'return'
'spmd'
'switch'
'try'
'while'

- MATLAB allows you to use the names of its built-in function as variable names. This is a dangerous practice, since you can overwrite the meaning of a function, such as **sin.** To check whether a name is a built-in function, use the **which** command. For example, typing

  **which sin**

  returns

  **built-in**

**1- Working with Matrices**

When solving engineering problems, it is important to visualize the data related to the problem .some time the data just a single number, such as the radius of a circle. Other times, the data may be a coordinate on a plane that can be represented as  a pair of numbers, with one

number representing the x-coordinate and the other number representing the y-coordinate. In another problem, we might have a set of four x-y-z coordinates that represent the four vertices of a pyramid with triangular base in three- dimensions space. We can represent all of these examples using a special type of data structures called matrix. A matrix is a set of numbers arranged in rectangular grid of rows and columns. Thus, a single point can be considered a matrix with one row and one column often referred to as a scalar. An x-y coordinate can be considered a matrix with one row and two columns, and often called vector. A set of four x-y-z coordinates can be considered a matrix with four rows and three columns:

A=[3.5]

B=[1.5 3.1]

Not that the data within a matrix are written in side brackets.

### 1- Scalar operations

The arithmetic operation between two scalars are shown in the table below. They include additional, subtraction, multiplication, and exponentiation. The command

```
a = 1 + 2
b=5
```

then

```
x = a + b
```

will return the following result:

```
x =
    8
```

| Operation | Algebraic form | MATLAB form |
|---|---|---|
| Addition | a + b | a + b |
| Subtraction | a – b | a – b |
| Multiplication | a x b | a* b |
| Division | $\frac{a}{b}$ | $a/b$ |
| Exponential | $\boldsymbol{a^b}$ | $\boldsymbol{a \wedge b}$ |

Table 1 arithmetic operations between two scalar

A single equals sign ( = ) is known in MATLAB as assignment operator. The assignment operator causes the result of your calculation to be stored in the computer memory location. In the example above, x is assigned a value of eight, and is stored in the computer memory. If you enter the variable name

```
X
```

In to MATLAB, you get the result

```
x =
    8
```

Which should be read as "x is assigned a value of 8". If we interpret assignment statement in this way, we are not disturbed by valid MATLAB statement

```
x = x + 1
```

which, since the value stored in the x was originally returns

```
x =
   9
```

Indicating that the value stored in the memory location named x has been changed to the 9. clearly, this statement is not valid algebraic statement, but is understandable when viewed as an assignment rather than as statement of equality. The assignment statement is similar to the familiar process saving file. When you first time save word document, you assign it a name. subsequently, when you have made changes, you resave your file, but still assign it the same name. the first and second versions are not equal, you have just assigned a new version of your document to an existing memory location.

**1- Precedence of arithmetic operations is**

Because several operations can be combined in the single arithmetic expression, it is important to know the order in which operations are performed. Table 2 contains the precedence of arithmetic operations performed in MATLAB. Not that this precedence follows the rules of stander algebraic precedence.

Assume that we want to calculate the area of trapezoid, where the base is horizontal and tow edge are vertical. We know that the equation for the area of this trapezoid is

```
Area = 0.5 * base * (height_1 + height_2)
```

There are two equally valid approaches to solving this problem. First assume some values for base, height_1, and height_2, and enter them into MATLAB:

```
>>Base = 5;
>>height_1;
>>height_2;
```

now you can solve the problem by entering the equation for area in the MATLAB:

```
>>Area = 0.5 * base * (height_1 + height_2)
```

| precedence | Operation |
|---|---|
| 1 | Parentheses, innermost |
| 2 | Exponentiation, left to right |
| 3 | Multiplication and division , left to right |
| 4 | Addition and subtraction , left to right |

Table 2

This equation returns

```
Area =
     45
```

Understanding the order of operation is important. MATALB will first add the tow height values together

```
height_1 + height_2
```

then the program will perform the multiplication operations, starting from the left. An alternative approach would be to simply enter the numerical values directly in to the formula:

```
area = 0.5 * 5* (12 + 6)
```

this return

```
area = 45
```

in either case, neglecting the parentheses will result in the wrong answer. For example

```
area = 0.5 * base * height_1 + height_2
```

gives

```
area = 36
```

in this case, MATLAB will first perform the multiplication, working from left to right

```
0.5 * base * height_1
```

Then it will add the result to the height_2 . clearly, it is important to be very careful when converting equations in to MATLAB statements. Adding extra parentheses is an easy way to ensure that computations are performed in the order you want.

If an expression is long, break it into multiple statements. For example, consider the equation

$$f = \frac{x^3 - 2x^2 + x - 6.3}{x^2 + 0.05005x - 3.14}$$

The value of $f$ could be computed with the following MATLAB statement :

```
numerator = x^3 - 2*x^2 + x - 6.3;

denominator = x^2 + 0.05005*x -3.14;
```

it is better to use several statements that are easy to understand than to use one statement that requires careful thought to figure out the order of operations

**Homework**

wind tunnel are used to evaluate high performance aircraft. (See the figure ) to interpret wind tunnel data, the engineer needs to understand how gases behave . the basic equation describing gas properties is the ideal gas law,

$$pv = nRT$$

Where

$p$=pressure, KPa

$v$= volume, $m^3$

$n$= number of kmoles of gas in the sample

$R$= ideal gas constant, 8.314 KPa $m^3$ kmoles K

$T$=temperature, expressed on an absolute scale

In addition, we know that the number of kmoles of gas is equal to the mass of gas divided by the molar mass (also known as molecular weight ), that is,

$$n = {}^{m}/_{MW}$$

where

$$m = mass, kg$$

$$MW = molar\ mass, kg/\text{kmole}$$

Different unites can be used in the equations, if the value of R is changed accordingly. Assume that the volume of the air in the wind tunnel is $1000m^3$ . before the wind tunnel is turned on , the temperature is 300 K and the pressure is 100 KPa. The molar mass (molecular weight ) of air is approximately 29 kg/kmole. Find the mass of the air in the wind tunnel