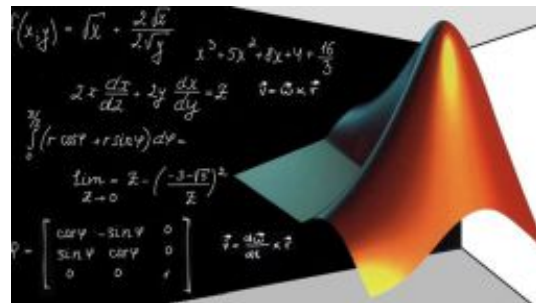


Informatics Practical Work



Dr. SALAH DJEROUNI

MOHAMED KHIDER UNIVERSITY OF BISKRA,
ALGERIA

FACULTY OF SCIENCES AND TECHNOLOGY
CIVIL ENGINEERING AND HYDRAULICS
DEPARTMENT

EMAIL : s.djerouni@univ-biskra.dz

Table of contents



I - Vectors in MATLAB	3
1. Introduction	3
2. Objectives	3
3. What is MATLAB?	3
3.1. <i>Programming environment</i>	3
3.2. <i>Applications</i>	4
4. Definition a vector	4
5. Different methods to identify a vector	4
5.1. <i>Random vector</i>	4
5.2. <i>Regular vector</i>	5
6. Types of the vectors	6
6.1. <i>Row vectors</i>	6
6.2. <i>Column vectors</i>	6
7. Vector operations	7
7.1. <i>Addition of vectors</i>	7
7.2. <i>Subtraction of vectors</i>	8
7.3. <i>Multiplication of vectors</i>	8
7.4. <i>Division of vectors</i>	9
7.5. <i>Power vectors</i>	9
7.6. <i>Transpose of vector</i>	10
8. Other useful MATLAB functions	11
9. Acknowledgement and guide in the application by MATLAB	13
10. Exercice : Acquisition test - Chapter 1	14
Abbreviation	15
Bibliography	16
Web bibliography	17

Vectors in MATLAB

I

1. Introduction

Fundamental

MATLAB allows you to address elements or subsets of vector elements simply and efficiently. This is one of the important features of **MATLAB**.

2. Objectives

- Learn the way to **define** and/or **declare row vector**, as well as **column vector**
- Learn the basics to **apply operations** in **two vectors** (+, -, x, and /)
- Learn additional commands that can help in **vectors operation**

3. What is MATLAB?

MATLAB is a **software package** designed for **mathematical** and **scientific computing**. It is also a **development environment** and a **programming language**. Its primary specialization is **efficiently handling matrix** and **vector** mathematics.

You can think of **MATLAB**^{*} as one of the **most useful programs** in the **engineering fields**. It deals with the **numbers** as a **matrices**, so, that is why the first part of its name is the **first three later** of the **matrix**, and the other part of its name is **LAP**, which is represents the short cut of **Laboratory**. It is designed to help you **manipulate very large sets of numbers quickly** and with **minimal programming**. **Operations on numbers** can be done **efficiently** by **storing them** as **matrices**. **MATLAB** is **particularly good** at doing **matrix operations** (this is the origin of its name).

3.1. Programming environment

Fig 1 shows a version of the default MATLAB programming environment. It consists of four spaces:

1. The **MATLAB Command Window** with the MATLAB prompt sign,
2. The **workspace** displaying the variables in the MATLAB memory,
3. The **current folder** box showing the folder's content, and
4. The **Command history** box showing a list of recent commands.

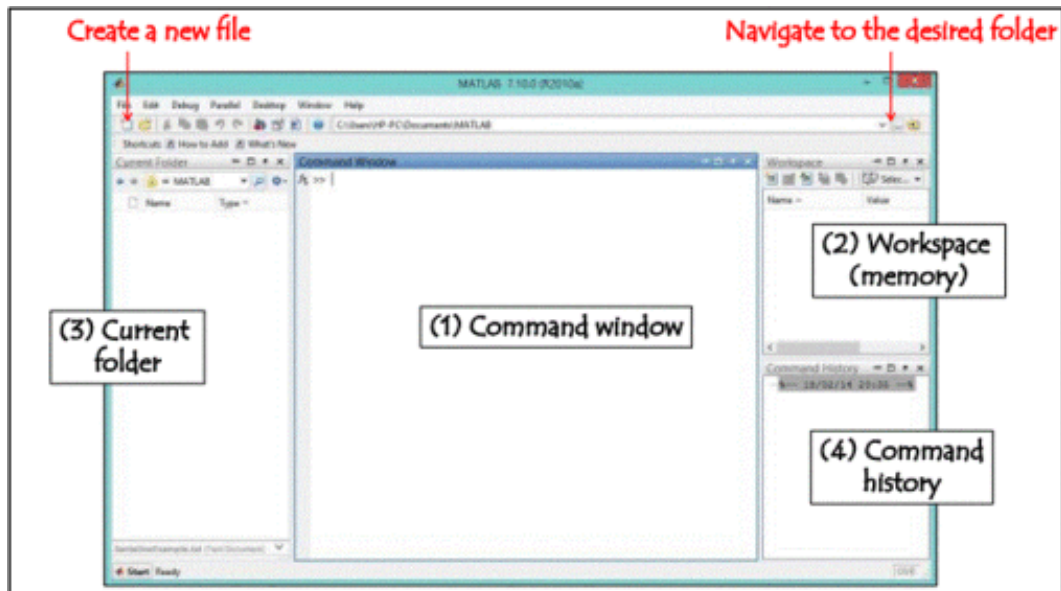


Fig 1.1 Default layout of the MATLAB Programming Environment

3.2. Applications

- **Matrices** calculations
- **Programming**
- **Graphing** (plotting of equations)
- **Numerical analysis**
- **Automatic control**
- **Statistical analysis**
- And so many applications

4. Definition a vector

Definition

By default, a ^{*}vector is a **one-dimensional array** of **numbers**. In other words, it is a **single row** with **several columns** or a **single column** with **several rows**.

5. Different methods to identify a vector

Before we identify a vector in **MATLAB**, we must know if the vector:

5.1. Random vector

MATLAB allows you to create an ^{*}arbitrary vector or called (**irregular, random**) vector in **three ways**, containing different numbers.

- Each element in the vector with the **order**
- All the elements inside a **square bracket** and between each element and the other **space**

- All the elements inside a **square bracket** and between each element and the other **comma “,”**,

Example

```

>> % first method
>> x(1)=8; x(2)=4; x(3)=4; x(5)=7; x(6)=11; x(7)=2; x(8)=0

x =

     8     4     4     1     7    11     2     0

>> % second method
>> x = [8 4 4 1 7 11 2 0]

x =

     8     4     4     1     7    11     2     0

>> % third method
>> x = [8,4,4,1,7,11,2,0]

x =

     8     4     4     1     7    11     2     0
  
```

Fig 1.2. Different methods to create a random vector

5.2. Regular vector

MATLAB allows you to create a uniformly spaced vector called a **proper (regular)** vector in **two ways**.

- Using the function/command **linspace(X1 , X2 , N)**; which generates **N** points between **X1** and **X2**.
- Using the function/command **v = [X1 : P : X2]**; which generates **v** vector, with the **first** element **X1**, **last** element **X2**, and the **difference between** elements is any real number **P**.

Example

```

>> % first method
>> v = linspace(1,10,10)

v =

     1     2     3     4     5     6     7     8     9    10

>> % second method
>> v = 1 : 1 : 10

v =

     1     2     3     4     5     6     7     8     9    10
  
```

Fig 1.3. Two methods to create a regular vector

Note

If you **don't** write the **step** or call the **difference between** elements **P** value it will take $P = 1$ **automatically**.

6. Types of the vectors

MATLAB allows you to create **two types** of **vectors** which can be stored either:

- A * row vectors and
- A * column vectors.

6.1. Row vectors

Definition

Are **created** and/or **declared** by **enclosing** the set of elements in **square brackets**, using **space** or **comma “,”** to **delimit** the **elements**, which can have any number of elements. For example, there are **two ways** to **create** a **row vector** with **five elements**.

Example

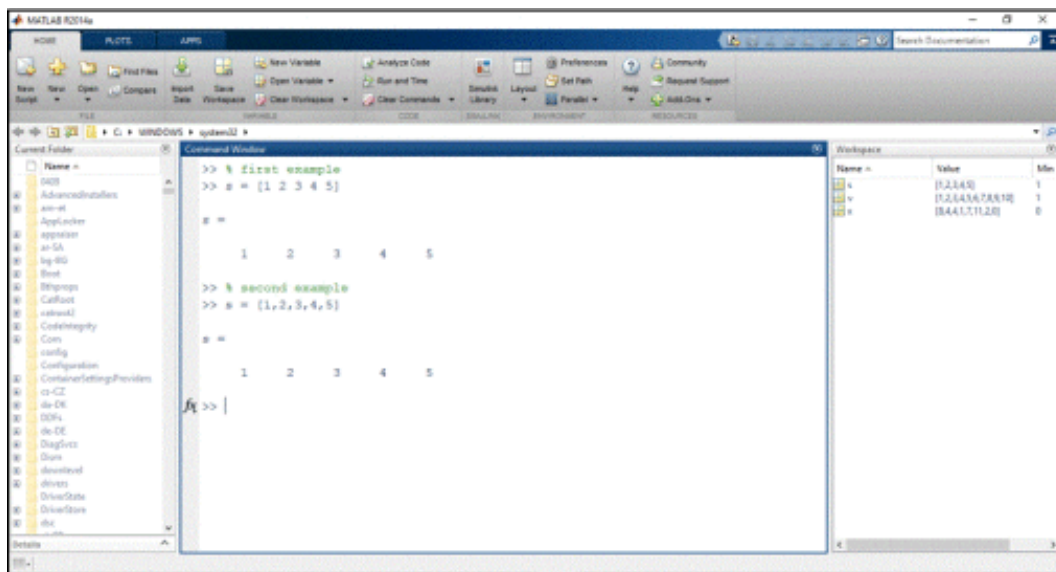


Fig 1.4. Two ways to create a row vectors

6.2. Column vectors

Definition

Are **created** and/or **declared** by **enclosing** the set of elements in **square brackets**, using a **semicolon “;”** to **delimit** the **elements**. For example, there is **one way** to create a **column vector** with **five elements**.

Example

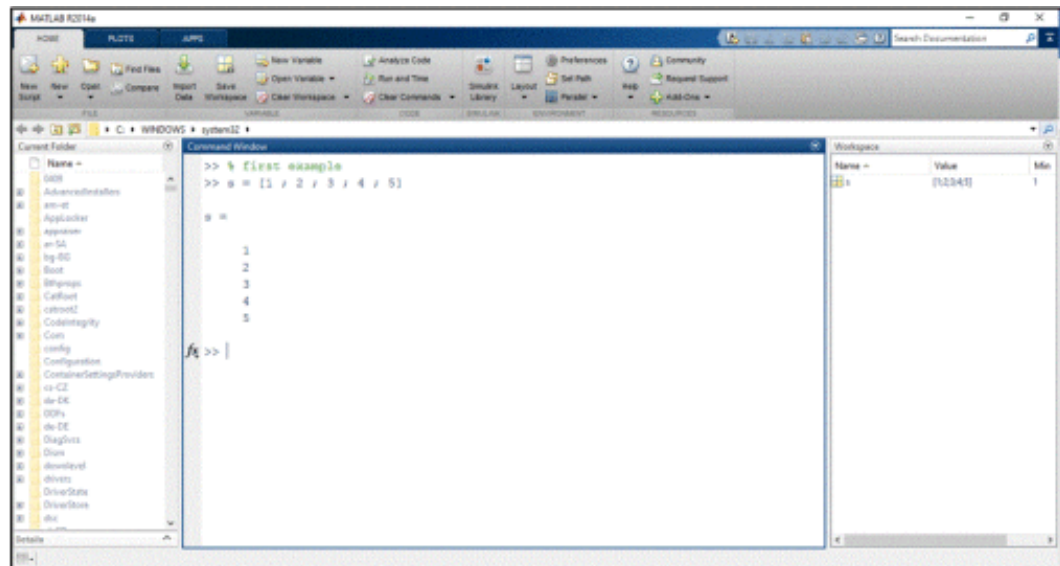


Fig 1.5. Define a column vector

7. Vector operations

In this section, let us discuss the following **vector operations**-

- Addition and subtraction of vectors
- Multiplication of vector
- *Division of vector
- *Power of vector, and
- Transpose of vector

Besides the standard **vector operations**, **MATLAB** performs an element-by-element array operations (**addition**, **subtraction**, **multiplication**, **division**, and **power**) among vectors of the **same dimensions**.

To illustrate this **special feature**, consider **two vectors**, **A** and **B**, of **n=3** elements.

7.1. Addition of vectors

You can **add two vectors**. Both the **operand vectors** must be the **same type** and have the **same number** of elements.

Example

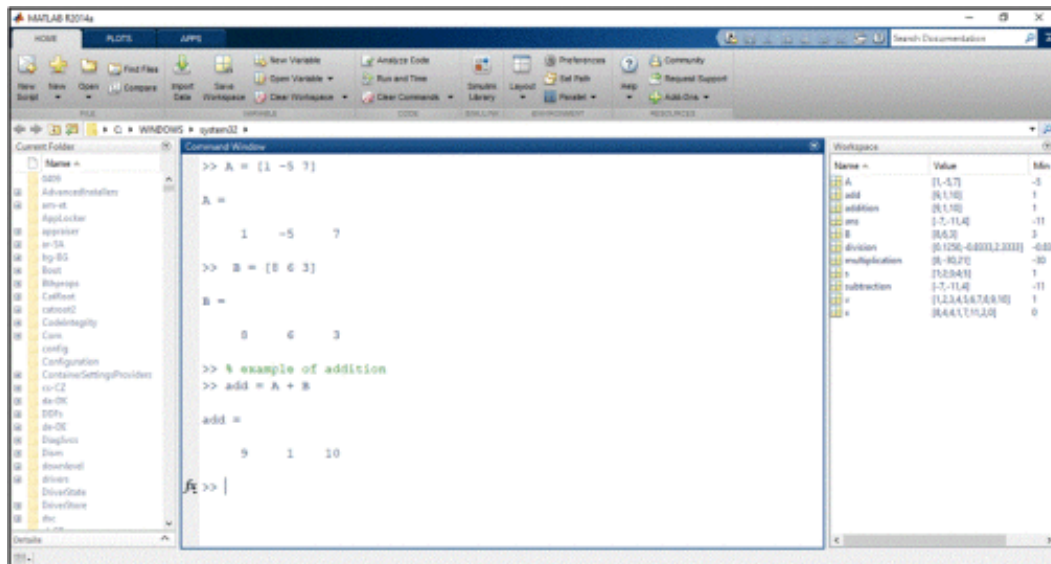


Fig 1.6. The sum of two vectors A and B

7.2. Subtraction of vectors

Is the **difference** between two given vectors, whose defined in **MATLAB** by the following way.

Example

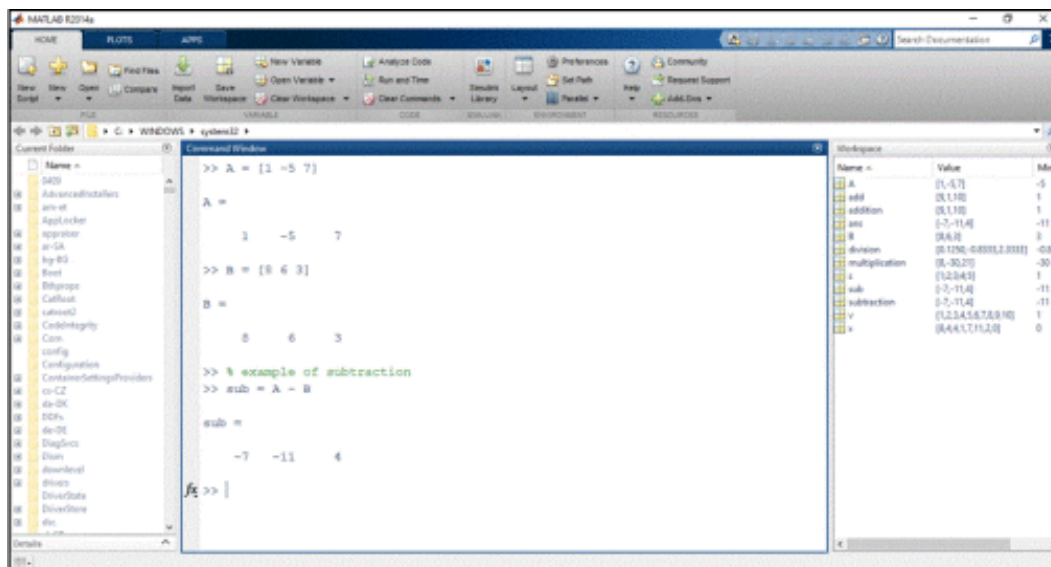


Fig 1.7. The difference between two vectors, A and B

7.3. Multiplication of vectors

Is the **multiplying** of two given vectors **element-by-element**, whose defined in **MATLAB** by the following way.

Example

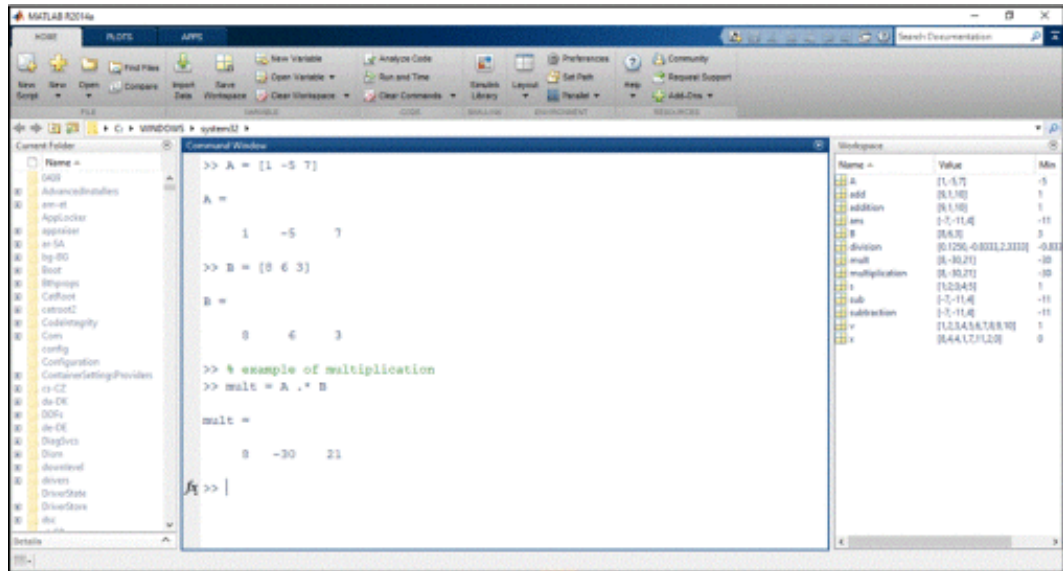


Fig 1.8. The multiplication of two vectors, A and B

7.4. Division of vectors

Is the **division** of **two given vectors element-by-element**, whose defined in **MATLAB** by the following way

Example

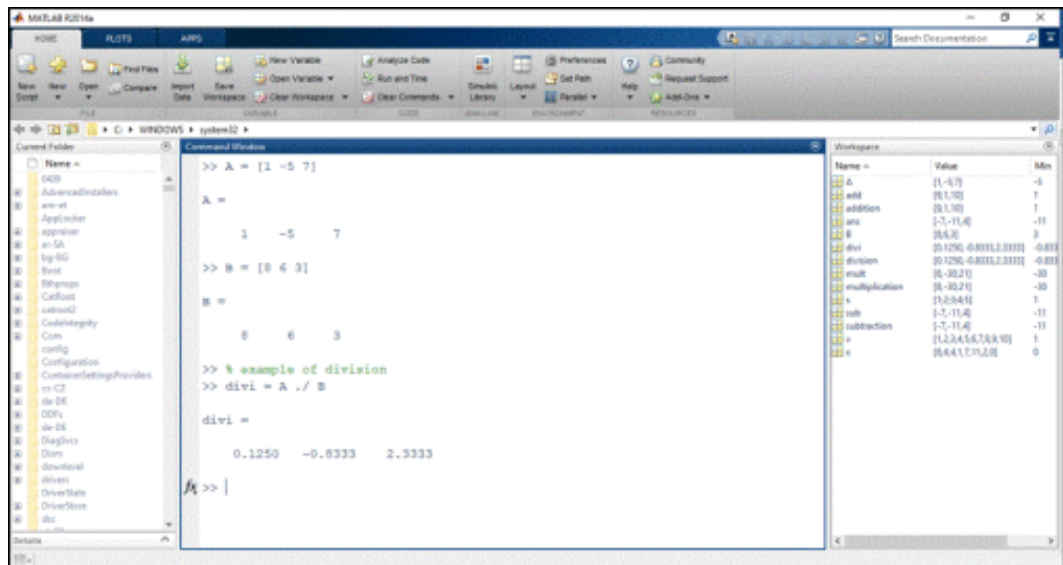


Fig 1.9. The division of two vectors, A and B

7.5. Power vectors

Is the **power** of **two given vectors**, whose defined in **MATLAB** by the following way.

Example

```

>> A = [1 -5 7]
A =
     1     -5     7

>> B = [8 6 3]
B =
     8     6     3

>> % example of power
>> pow = A.*B

pow =
     1 15625    343

fx >>

```

Fig 1.10. The power of two vectors, A and B

7.6. Transpose of vector

Is an **operation** that **changes** a **row vector** into a **column vector**, and vice-versa using the **transpose commands** or the **apostrophe** (**'**).

Example

```

>> A = [-1 -5 7]
A =
    -1     -5     7

>> transpose(A)
ans =
    -1     -5     7

>> A'
ans =
    -1     -5     7

fx >>

```

Fig 1.11. Transpose vectors with two different ways

Note

- In **MATLAB**, **row** and **column** numbers always **begin** with **1**, **not** zeros, as in other **programming languages**. **Zero**, as it is the case in other **programming languages**.
- The **dimension** of the **two vectors** must be the **same** for the **operation in the vectors**, which means the **number of elements** in **each vector** must be the **same/equal**.

- Here, operator **.*** (dot multiplication), **./** (dot division), and **.^** (dot power) are used for **element-by-element** array operations. These operators make programming for computation compact and efficient.

8. Other useful MATLAB functions

For vectors, to find the **maximum** and **minimum** values of the vector **x**, we use the command/function **max(.)** and **min(.)**

 *Example*

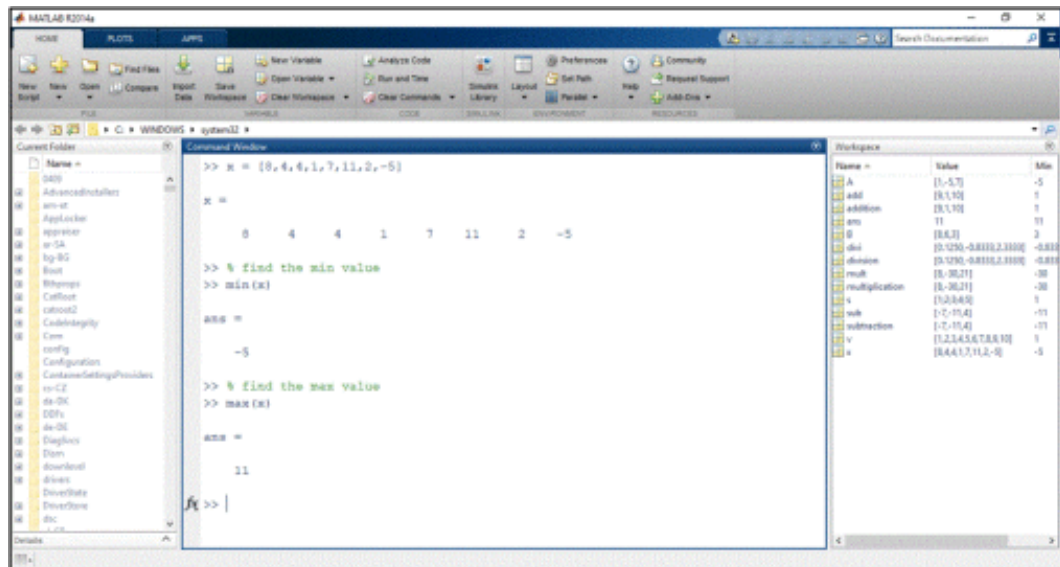


Fig 1.12. The minimum and maximum values from a vector *x*

To find the **summation** and the **production** values of the vector **x**, we use the command/function **sum(.)** and **prod(.)**

 *Example*

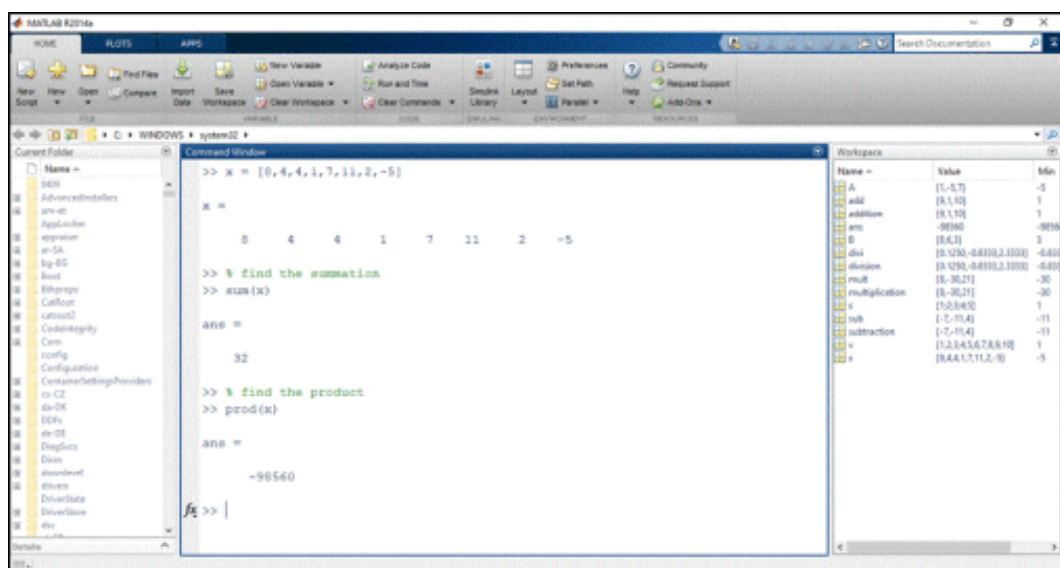


Fig 1.13. The sum and the product of the element in a vector *x*

For vectors, to **sorts the elements** of a given vector **x** in **ascending order arrange without** or **with reputation**, we use the command/function **unique(.)** and **sort(.)**

Example

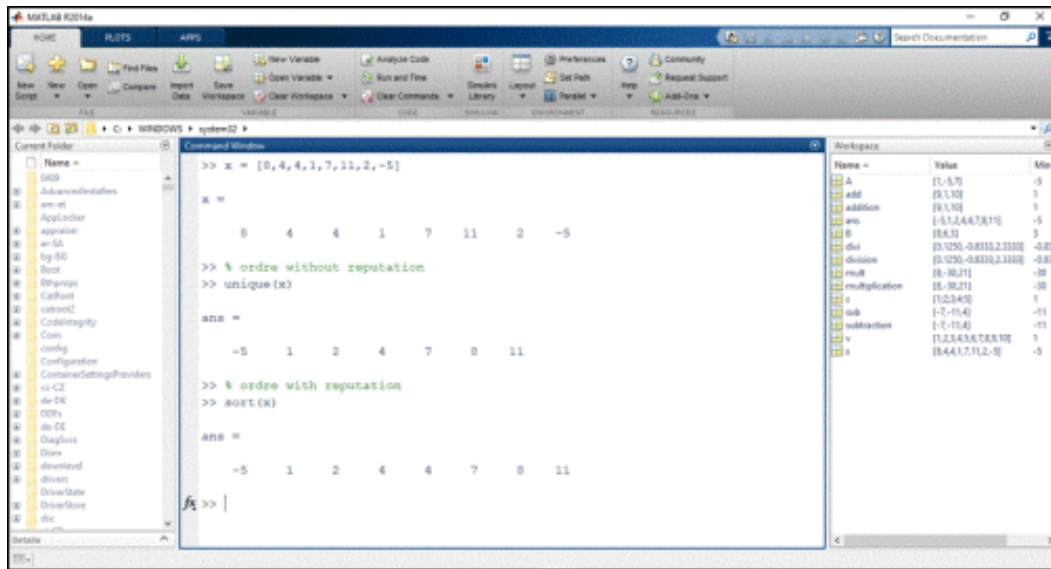


Fig 1.14. Arranging the vector's value from smallest to largest with or without reputation

Again, to **sorts the elements** of a given vector **x** in **descending order**, we use the command/function **sort** (**','descend'**)

Example

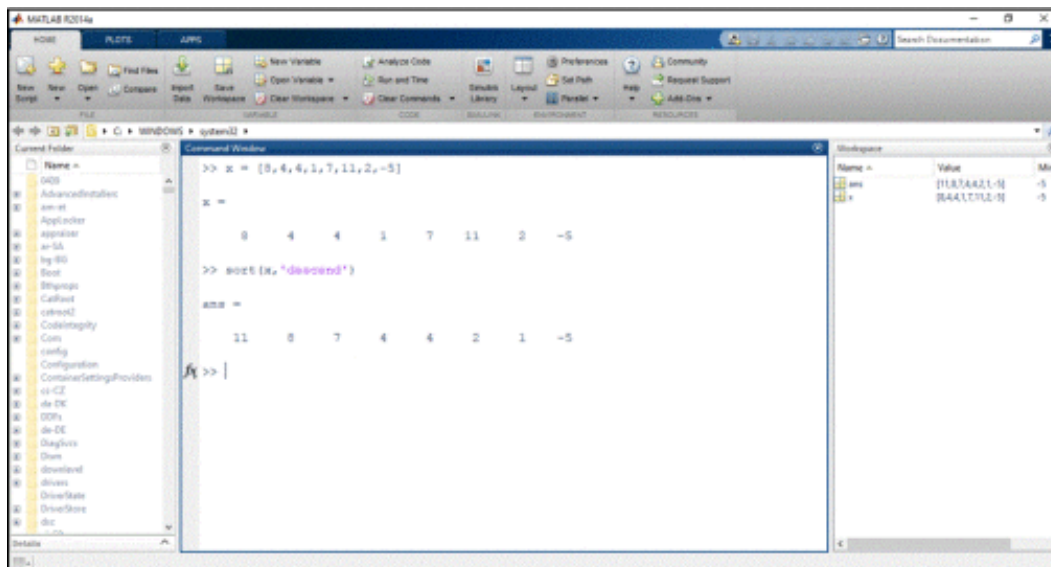


Fig 1.15. Arranging the vector's value from largest to smallest

The **mean** of a vector, also **known as the average equals the sum of the vector elements divided by the number of elements in the vector**, we use the command/function **mean(.)**

Example

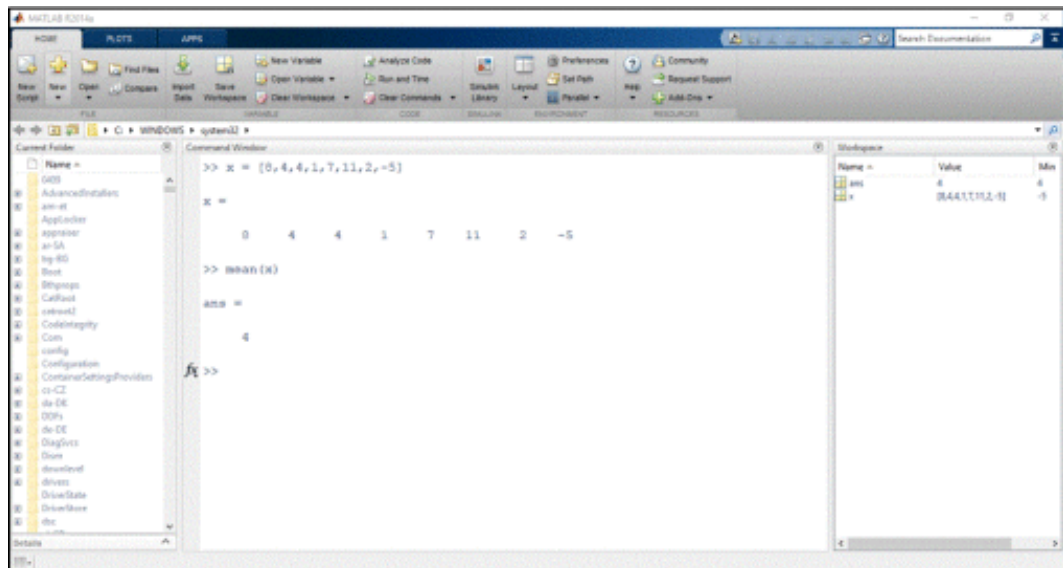


Fig 1.16. The mean of the vector x

Note

You can **get information** about the **dimension** of a **vector** using command/function **length(.)**

Example

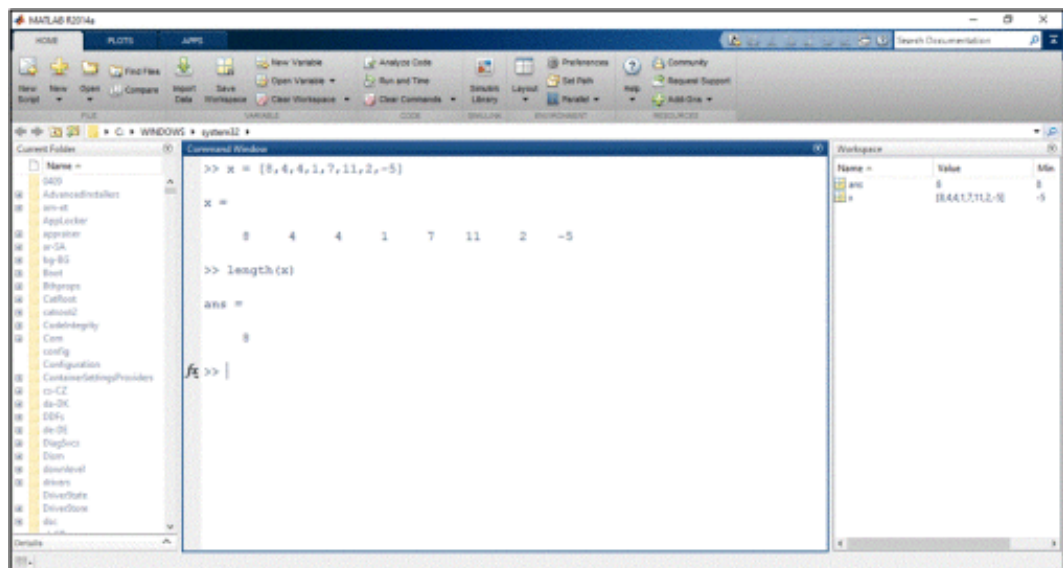


Fig 1.17. The dimension of the vector x

9. Acknowledgement and guide in the application by MATLAB

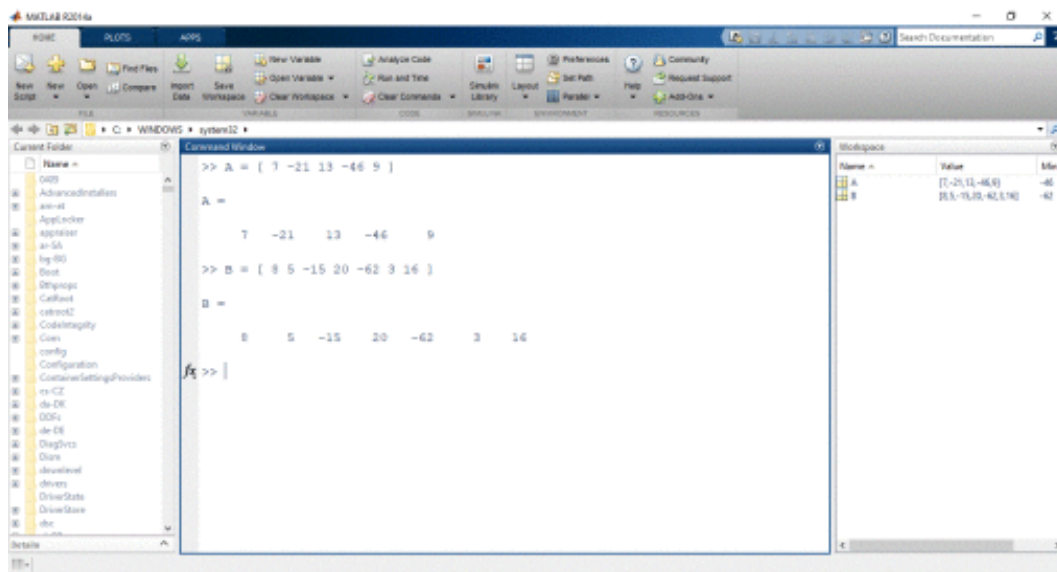
 **Complement:** For more information about the practical application on vectors in MATLAB, you can access to the link of the video below

See "Vectors in MATLAB"

10. Exercise : Acquisition test - Chapter 1

Exercise : Assignment N°1

In MATLAB, is it possible to calculate the division element-by-element of the following two vectors ?



☐ Yes

☐ No

Exercise : Assignment N°2

What's the types of vectors?

☐ Row vectors

☐ Column vectors

☐ Row and column vectors

☐ Column and row vectors

Abbreviation

MATLAB: MATrix LABoratory

Bibliography



MATLAB A PRACTICAL INTRODUCTION TO PROGRAMMING AND PROBLEM SOLVING

MATLAB A SELF-TEACHING GUIDE

MATLAB for Beginners

Web bibliography

https://www.tutorialspoint.com/matlab/matlab_vectors.htm