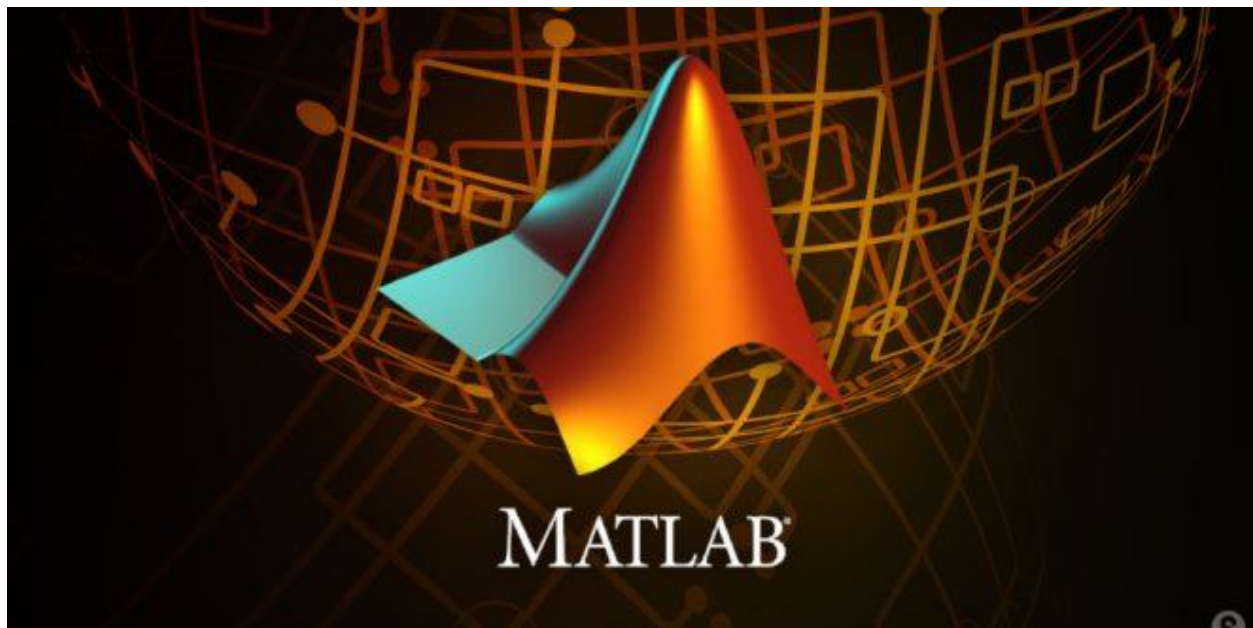


# Course N°04

## Matrix in MATLAB



Dr. Salah Djerouni



## 1. Definition a matrix

A matrix is a surrounded by **brackets** and may have an **arbitrary number** of **rows** and **columns**; for example, the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad (1)$$

To create a such **matrix** in **MATLAB**, the following basic conventions must be followed:

- ✓ **Separate** the elements of a row with **spaces** or **commas** ","
- ✓ Use a **semicolon** ";" to indicate the end of each **line** or **row**
- ✓ **Surround** the entire list of items with **square brackets** [ ].

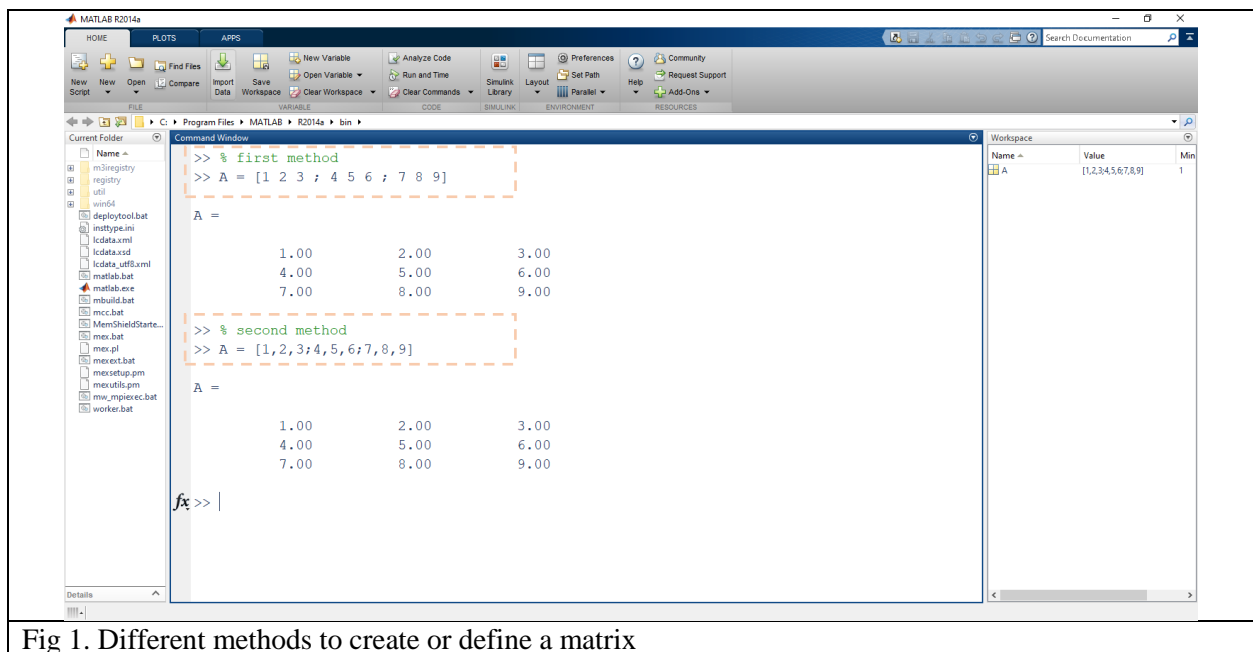
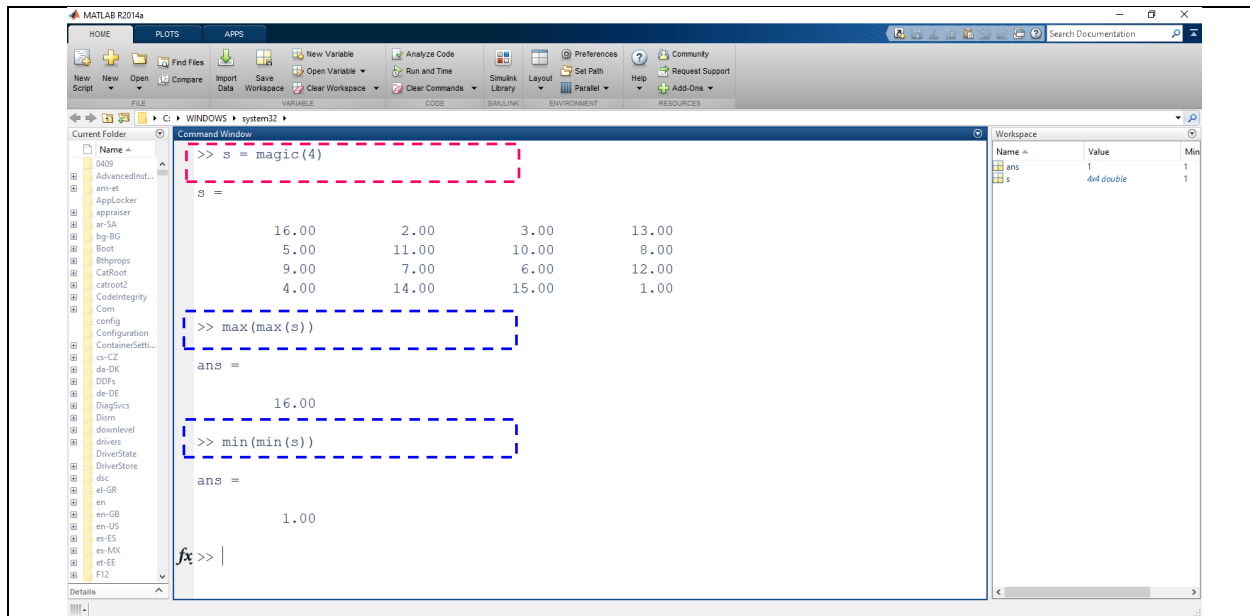


Fig 1. Different methods to create or define a matrix

## 2. Other useful MATLAB functions

For **matrices**, to **find** or **extract** the **highest** and/or **lowest value** or **number** in the **whole matrix**, we use the **command/function** **max(.)** and/or **min(.)** two times because these are matrix with **two dimensions** not one dimension. Otherwise, will indicate the highest values or the lowest values for each column in the matrix.



```
>> s = magic(4)

s =

    16.00    2.00    3.00   13.00
     5.00   11.00   10.00    8.00
     9.00    7.00    6.00   12.00
     4.00   14.00   15.00    1.00

>> max(max(s))

ans =

    16.00

>> min(min(s))

ans =

     1.00
```

Fig 2. Find the **highest** and **lowest** value in the whole matrix

Again, in **order** to **find** or **evaluate** the **summation** and/or **production** of such element in the matrix, we use the **command/function** **sum(.)** and **prod(.)**

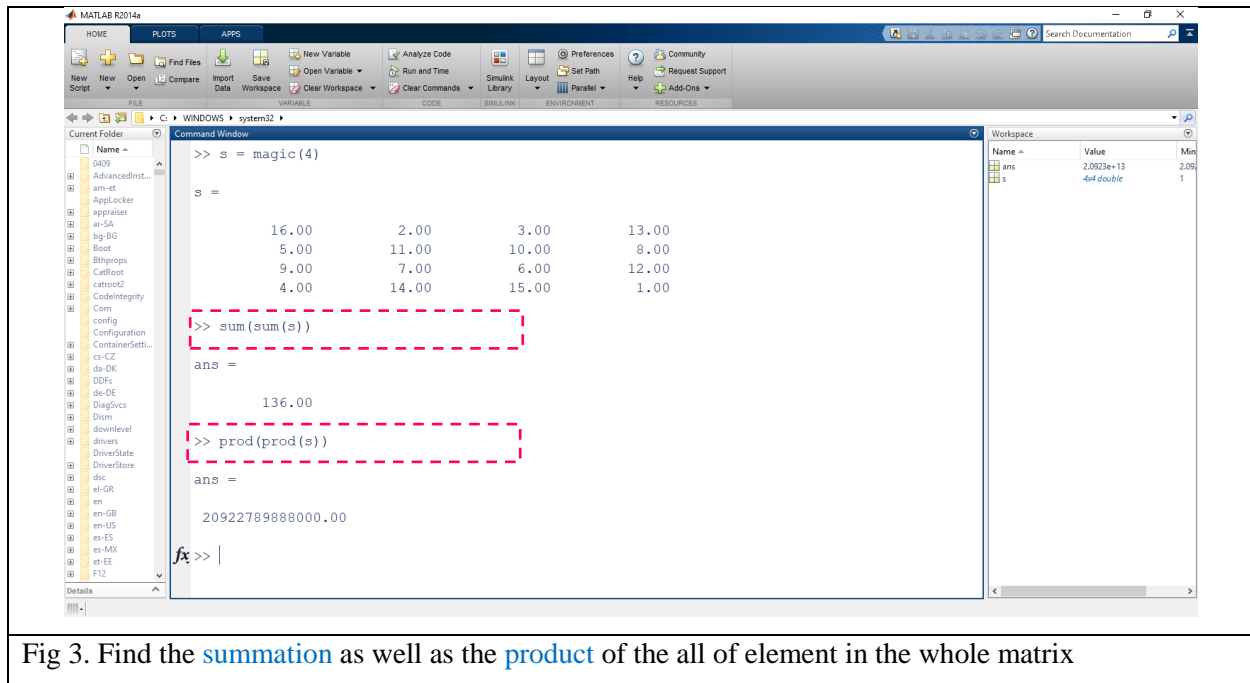


Fig 3. Find the **summation** as well as the **product** of the all of element in the whole matrix

The **mean** of a **matrix**, also known as the **average** which equal the **sum** of the **numbers** in **each row** in the matrix **divided** by the **number** of **rows** and then the result will **sum** again and **divided** on the **number** of **columns**, using the command/function `mean(.)`

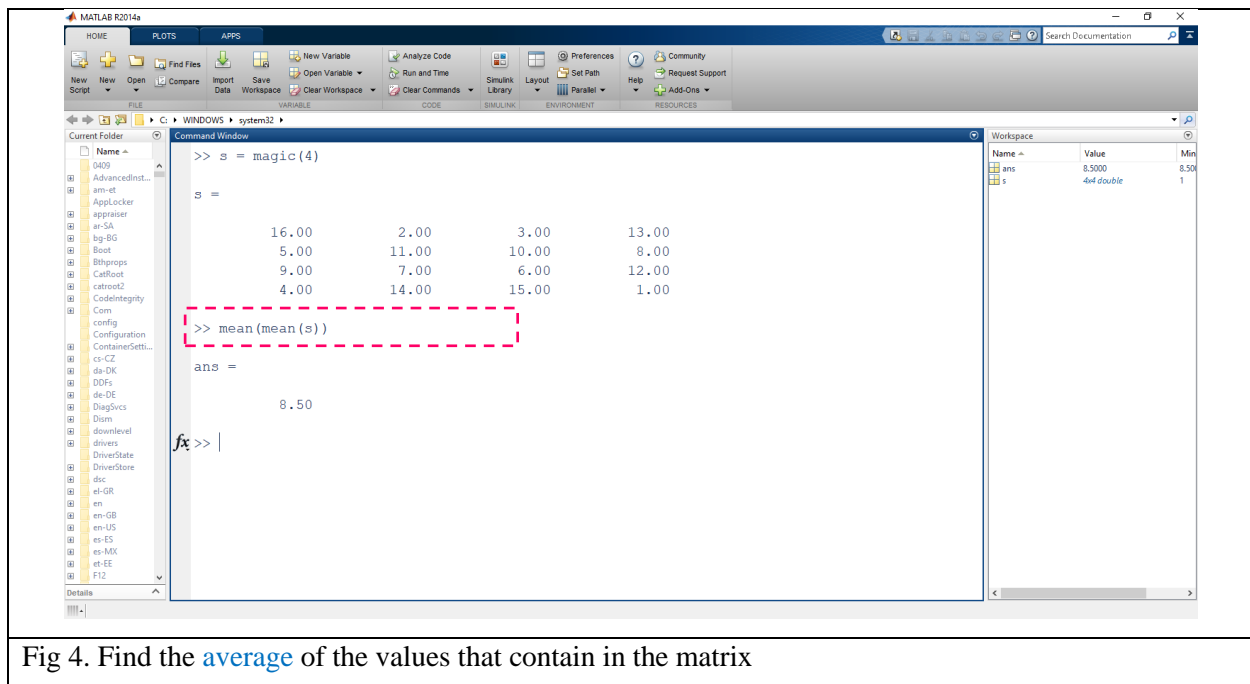


Fig 4. Find the **average** of the values that contain in the matrix

### 3. MATLAB output

#### 3.1. The *diag* command

Displaying or to generate a **diagonal** of a **matrix**, using the command/function **diag(.)**

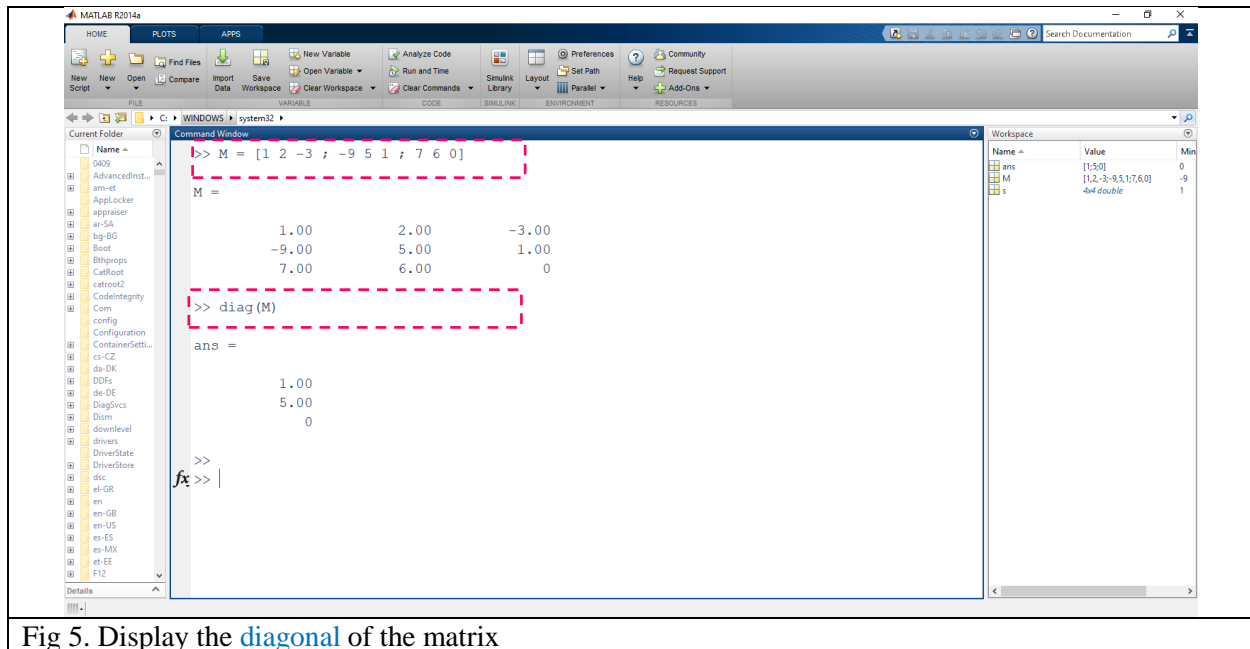
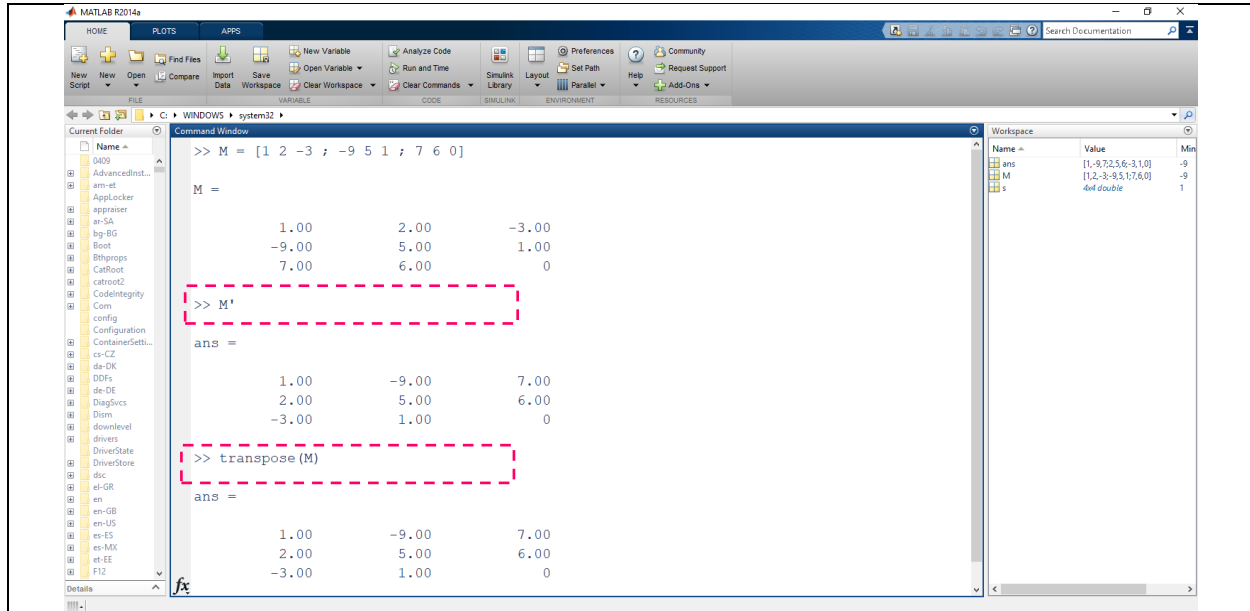


Fig 5. Display the **diagonal** of the matrix

### 3.2.The *transpose* and/or '*'* and/or *transp* command

Transpose a matrix, rows become a columns and columns become rows.



```

>> M = [1 2 -3 ; -9 5 1 ; 7 6 0]

M =

     1.00     2.00    -3.00
    -9.00     5.00     1.00
     7.00     6.00     0.00

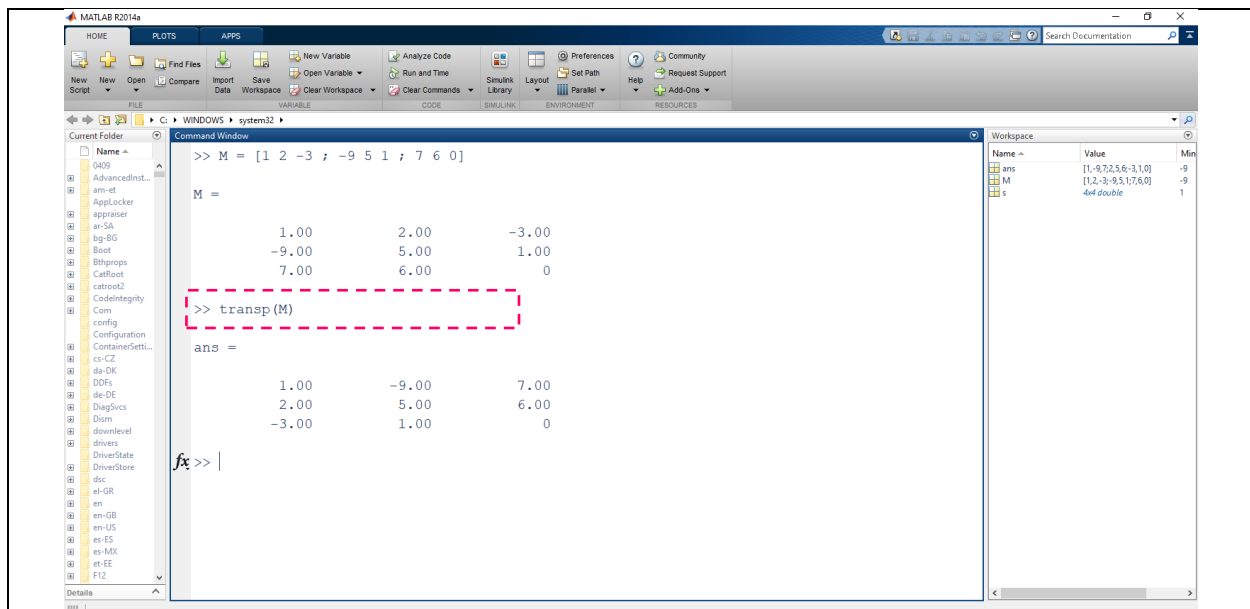
>> M'

ans =

     1.00    -9.00     7.00
     2.00     5.00     6.00
    -3.00     1.00     0.00
    
```

Fig 6. Display the *transpose* of the matrix

Or



```

>> M = [1 2 -3 ; -9 5 1 ; 7 6 0]

M =

     1.00     2.00    -3.00
    -9.00     5.00     1.00
     7.00     6.00     0.00

>> transp(M)

ans =

     1.00    -9.00     7.00
     2.00     5.00     6.00
    -3.00     1.00     0.00
    
```

Fig 7. Display the *transpose* of the matrix another command

### 3.3.The *inv* command

To display the inverse of such matrix just type the command/function `inv(.)` and the inverse of the matrix will be printed in the screen

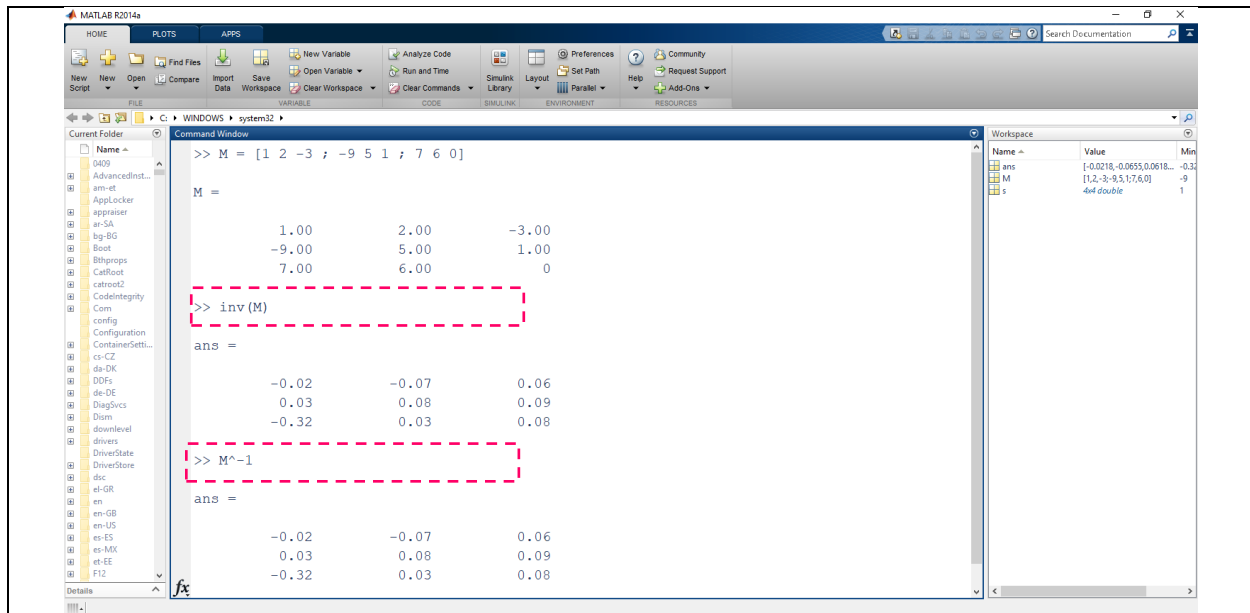


Fig 8. Display the **inverse** of the matrix

### 3.4.The *size* command

To display the dimension of a matrix, just type the command/function `size(.)` and the dimension will be printed in the screen.

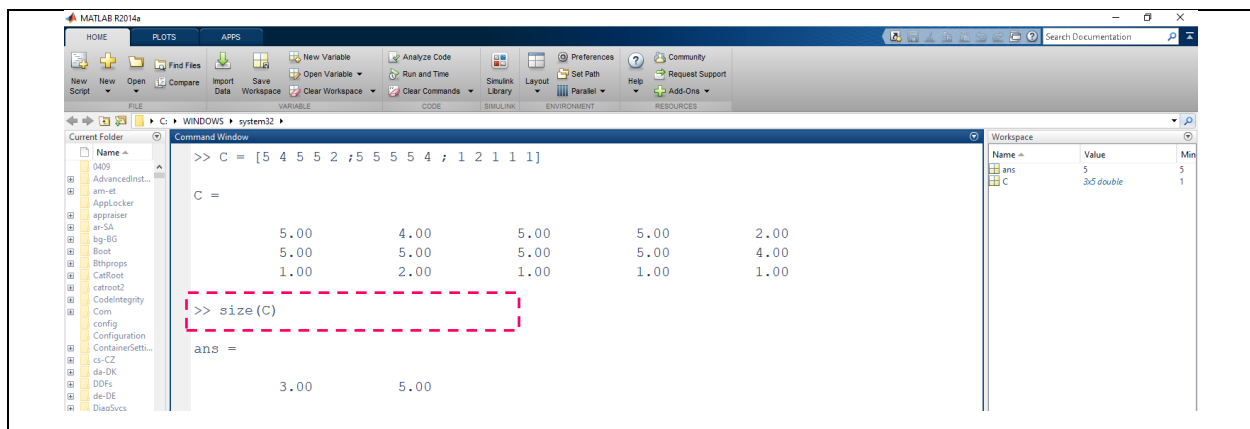
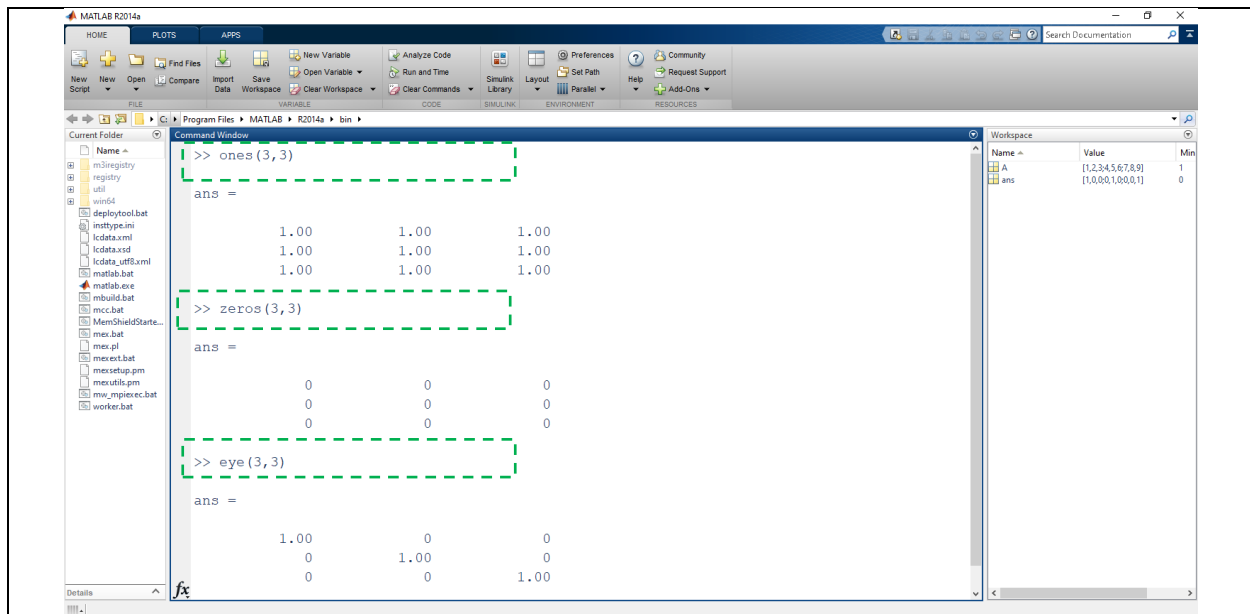


Fig 9. Display the **dimension** of the matrix

#### 4.Special or particular matrices

In MATLAB, there are functions that automatically generate specific matrices for example containing all ones, zero elements, ones in the diagonal only, arbitrary number.



```
>> ones(3,3)

ans =

    1.00    1.00    1.00
    1.00    1.00    1.00
    1.00    1.00    1.00

>> zeros(3,3)

ans =

     0     0     0
     0     0     0
     0     0     0

>> eye(3,3)

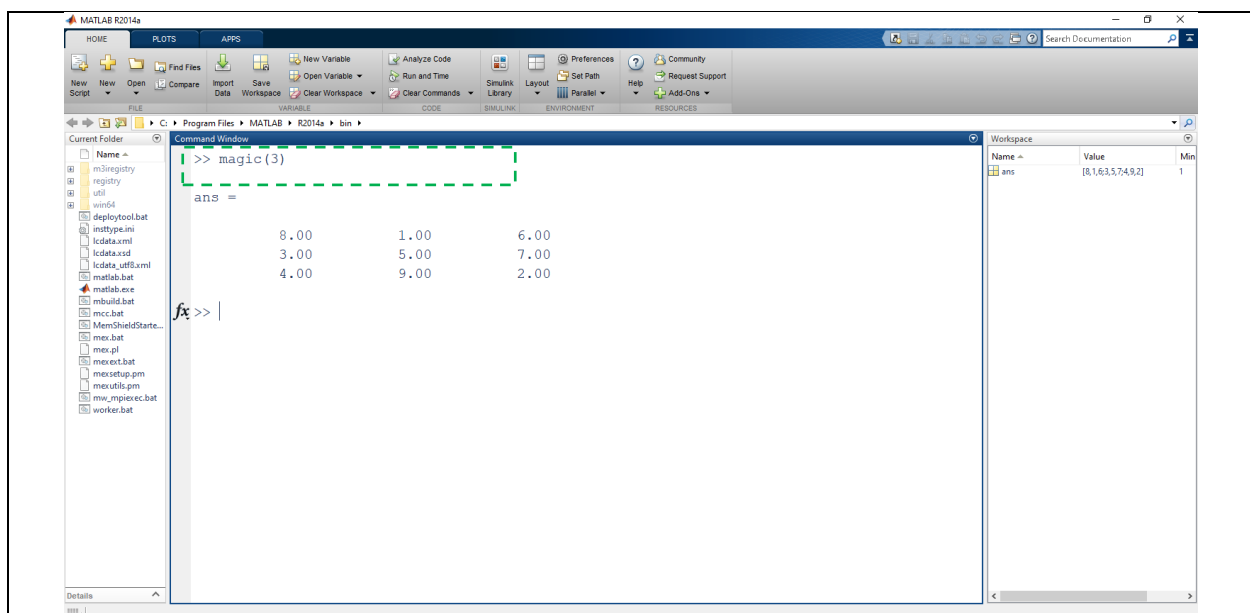
ans =

    1.00     0     0
         0    1.00     0
         0     0    1.00
```

The figure shows the MATLAB R2014a interface. The Command Window displays the execution of three commands: `ones(3,3)`, `zeros(3,3)`, and `eye(3,3)`. Each command is highlighted with a green dashed box. The output for each command is a 3x3 matrix. The `ones` matrix contains all 1s, the `zeros` matrix contains all 0s, and the `eye` matrix contains 1s on the diagonal and 0s elsewhere. The Workspace window on the right shows the variables `A` and `ans` with their respective values.

Fig 10. Create a matrix contain number « 1 » or « 0 » in all the column and row and number « 1 » in diagonal only

And



```
>> magic(3)

ans =

     8.00     1.00     6.00
     3.00     5.00     7.00
     4.00     9.00     2.00
```

The figure shows the MATLAB R2014a interface. The Command Window displays the execution of the `magic(3)` command, which is highlighted with a green dashed box. The output is a 3x3 magic matrix. The Workspace window on the right shows the variable `ans` with its value.



Fig 11. Create a random matrix

## 5. Matrix and submatrix manipulations

### 5.1. Pick and/or replace command

To show or **peak** or **extract** only **one value** from **matrix already** written by MATLAB, we need to know

- ✓ The **name** of the **matrix first**,
- ✓ Second, the **position** of that **element** in **row** and **column** of the **matrix**

Similarly, to **replace one** value **from matrix**, we need to know

- ✓ The **name** of the **matrix**
- ✓ Second, specify the **position** of the element by number of **row** and **column** in the matrix
- ✓ Third, give the **new number** or **value**

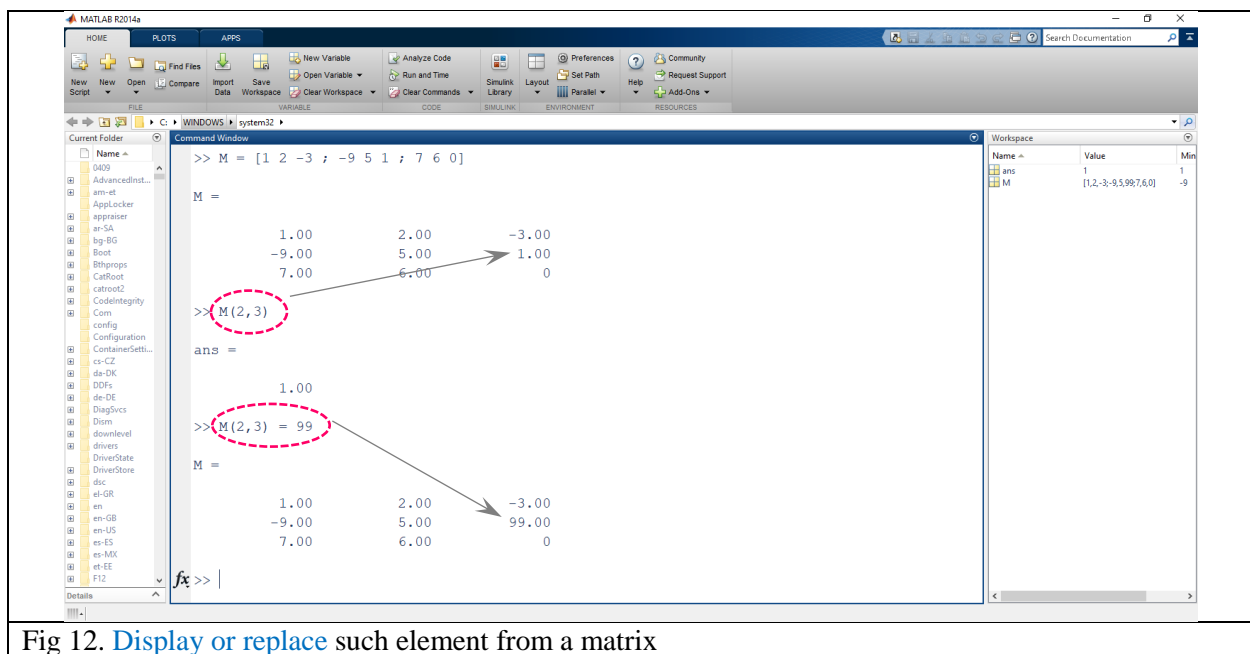


Fig 12. Display or replace such element from a matrix

### 5.2. Remove a row from a matrix

To **remove** a **row** from a matrix already written by MATLAB, we need to know the **position** or the **order** of this **row** in that matrix

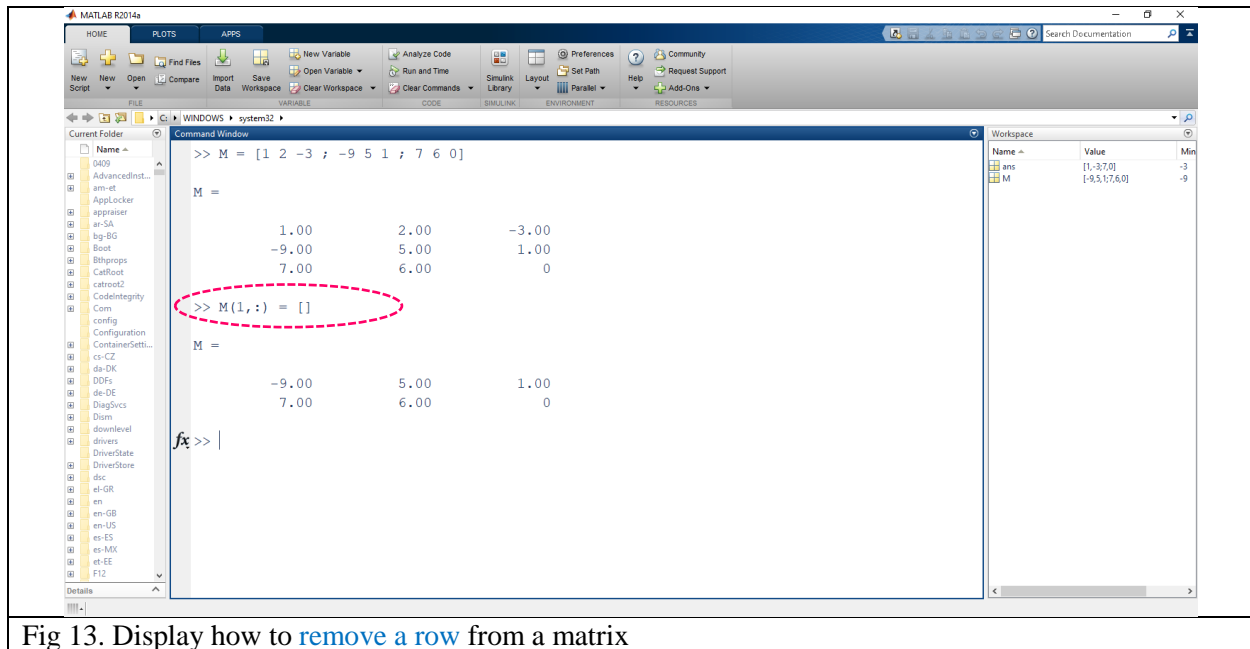


Fig 13. Display how to remove a row from a matrix

### 5.3.Remove column from a matrix

To remove a column from a matrix already written by MATLAB, we need to know the position or the order of this column in that matrix

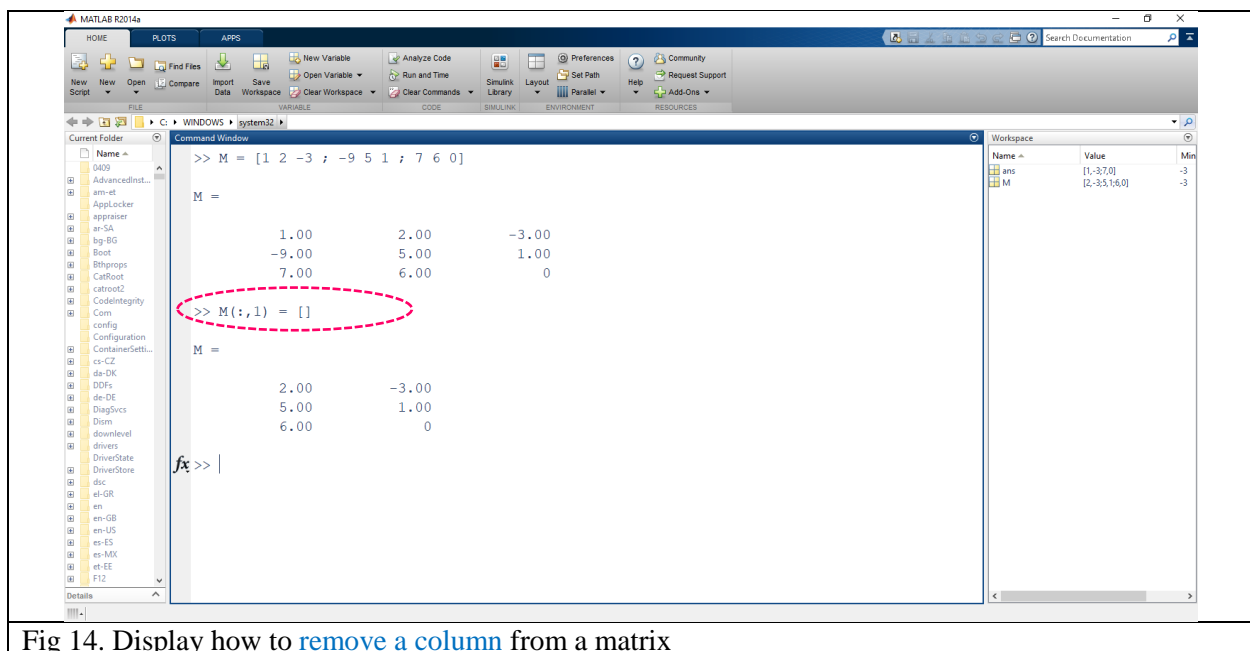
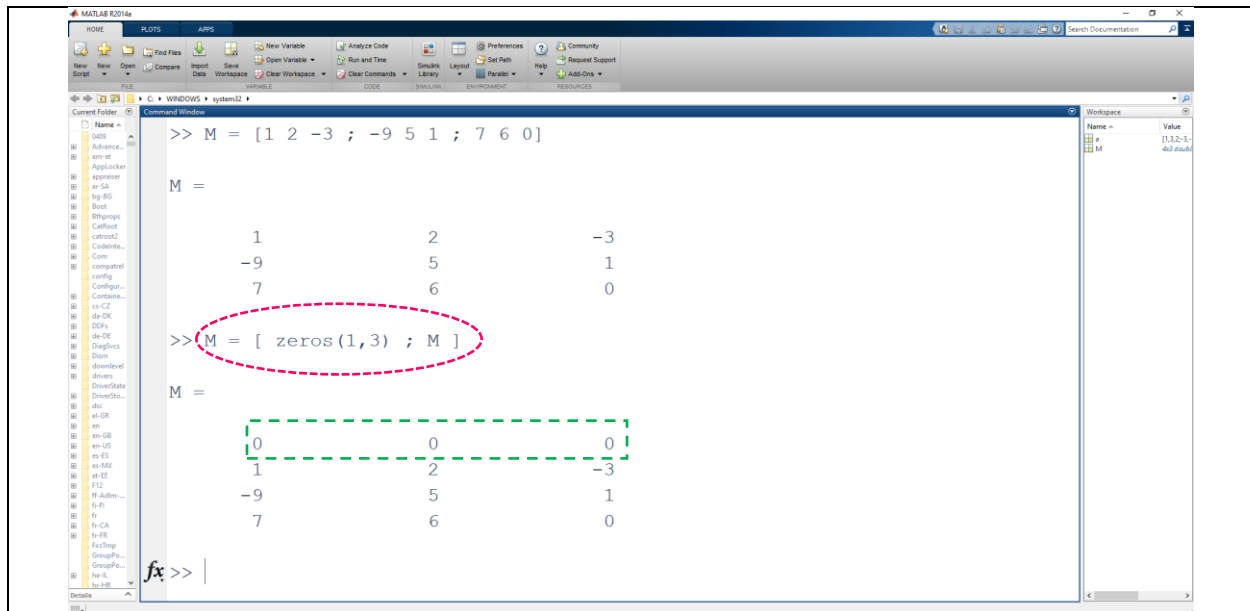


Fig 14. Display how to remove a column from a matrix

#### 5.4. Add a row to a matrix

To add a **row** from a **matrix** already written by **MATLAB**, we need to respect the dimension of the row in the matrix



```

>> M = [1 2 -3 ; -9 5 1 ; 7 6 0]

M =

     1     2    -3
    -9     5     1
     7     6     0

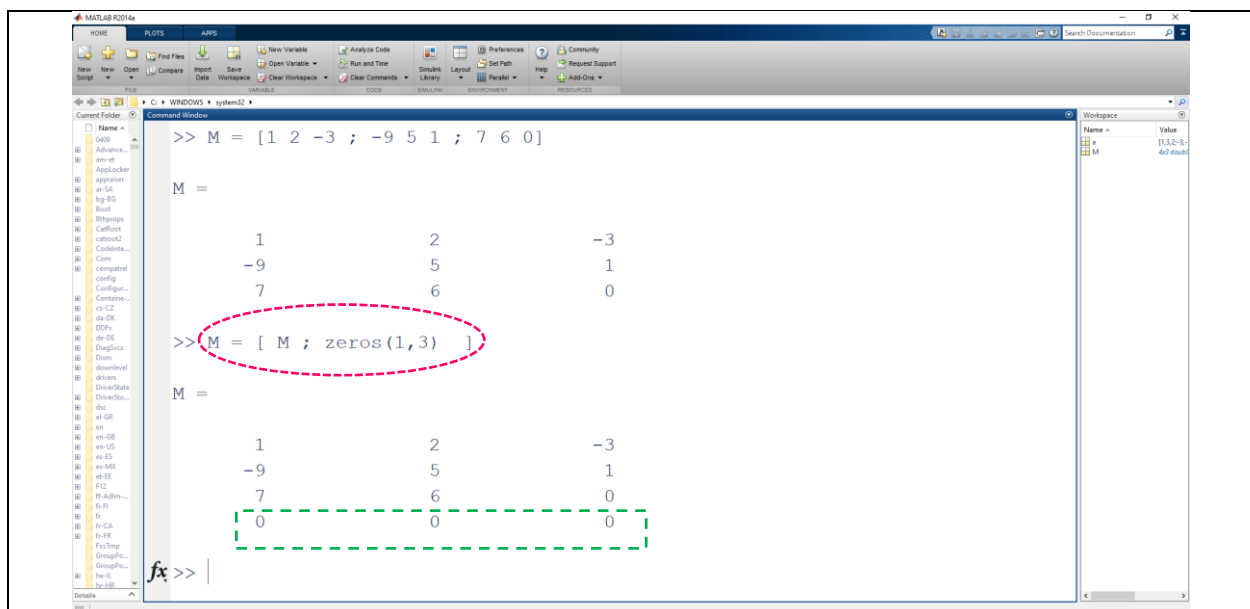
>> M = [ zeros(1,3) ; M ]

M =

     0     0     0
     1     2    -3
    -9     5     1
     7     6     0
    
```

Fig 15. Display how to **add a row** to a matrix **at begin**

And



```

>> M = [1 2 -3 ; -9 5 1 ; 7 6 0]

M =

     1     2    -3
    -9     5     1
     7     6     0

>> M = [ M ; zeros(1,3) ]

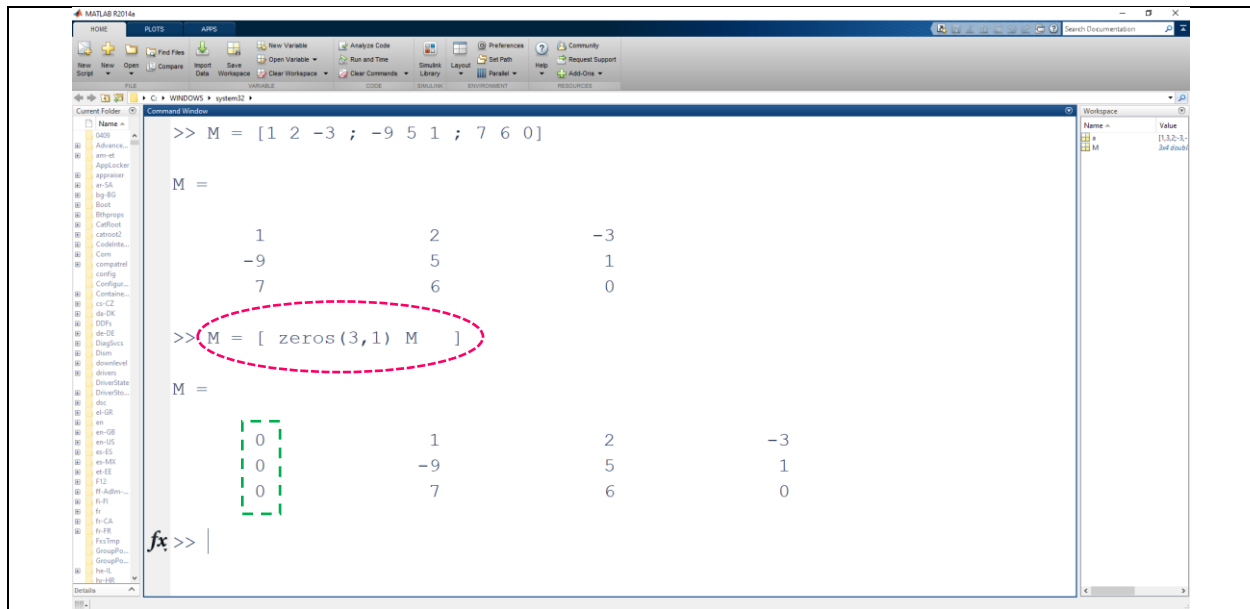
M =

     1     2    -3
    -9     5     1
     7     6     0
     0     0     0
    
```

Fig 16. Display how to **add a row** to a matrix **at end**

### 5.3. Add a column to a matrix

To add a **column** from a **matrix** already written by **MATLAB**, we need to respect the dimension of the column in the matrix



```

>> M = [1 2 -3 ; -9 5 1 ; 7 6 0]

M =

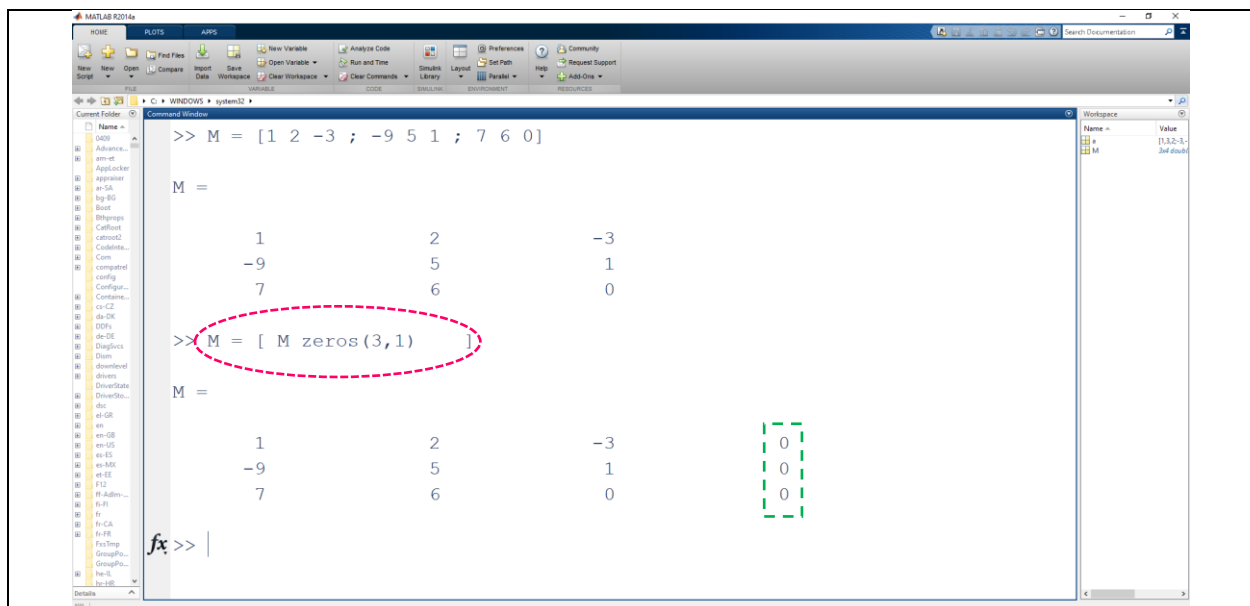
     1     2    -3
    -9     5     1
     7     6     0

>> M = [ zeros(3,1) M ]

M =

     0     1     2    -3
     0    -9     5     1
     0     7     6     0
    
```

Fig 17. Display how to **add a column** to a matrix **at begin**



```

>> M = [1 2 -3 ; -9 5 1 ; 7 6 0]

M =

     1     2    -3
    -9     5     1
     7     6     0

>> M = [ M zeros(3,1) ]

M =

     1     2    -3     0
    -9     5     1     0
     7     6     0     0
    
```

Fig 18. Display how to **add a column** to a matrix **at end**



## 6. List of References

Kattan, Peter Issa. Matlab for Beginners: A gentle approach. Petra books, 2008.

Etter, Delores M., David C. Kuncicky, and Douglas W. Hull. Introduction to MATLAB. Vol.4. Hoboken, NJ, USA: Prentice Hall, 2002.

Attaway, Stormy. Matlab: a practical introduction to programming and problem solving. Butterworth-Heinemann, 2013.

Driscoll, Tobin A. Learning Matlab. Society for Industrial and Applied Mathematics, 2009.

Butt, Rizwan. Introduction to numerical analysis using MATLAB. Laxmi Publications, Ltd., 2008.

Sigmon, Kermit. Matlab: aide-mémoire. Springer Science & Business Media, 1999.

Chapman, Stephen J. Essentials of MATLAB programming. Cengage Learning, 2016.