



# Course N°07

## The while-end loop

### in MATLAB



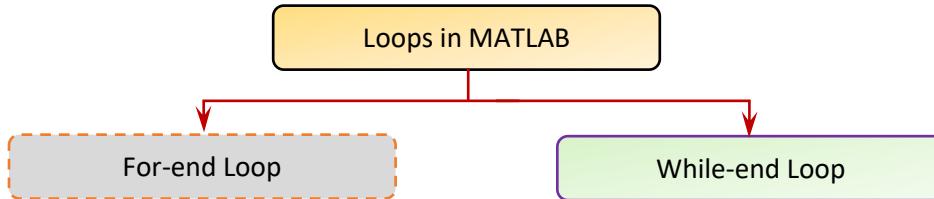
Dr. Salah Djerouni



## 1. Repetitive execution/operation

**MATLAB** features two of the most common loop structures: **counter loops (*for*)** and **conditional loops (*while*)**.

Unlike other languages, **MATLAB** only has **two types** of loop, ***for*** and ***while***. For loops should be used when the **number of iterations** is known beforehand - as in ‘Loop over these statements five times. When the required number of iterations is unknown, or may be different for each run of the program, use a **while** loop.



## 2. The *while-end* Loop

### 2.1. General loop structure

A **while loop** is used to **repeat** an **operation** until a **condition** is **met**.

An alternative to the ***for loop*** is the ***while loop***. If an index in the program is required, the use of the while loop statement (unlike the for-loop statement requires that the program generates its own index. The basic syntax for a ‘while’ loop is as follows:

```

var = start ;
while (var <= stop)
    .... Statements/operations to be repeated.....
    var = var + step ;
end
    
```

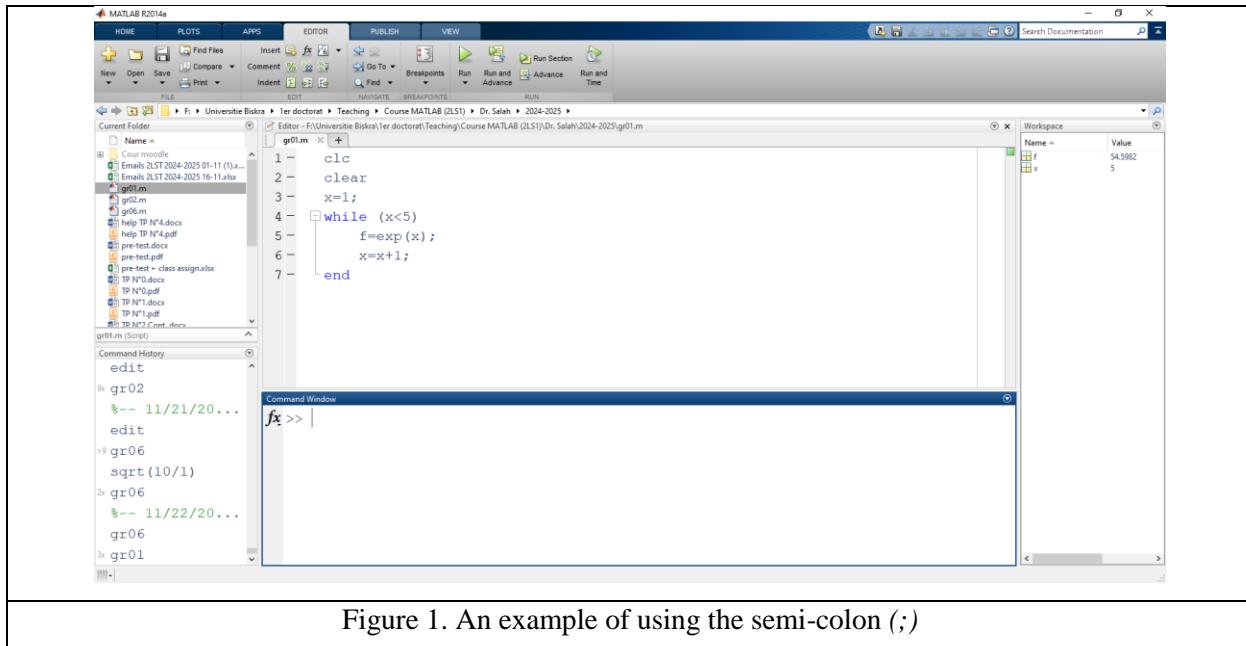
*Note.*

- The while loop is very useful, for example, to reach process convergence.



- Semi colon (;) after each command/operation do **not show or display the execution** of that command/operation in command window (see figure 1)
- To specify the results that need to show in command window just type the command ***disp( )*** and the name of the variable between parenthesis ( ),(see figure 2)
- If you want to show **more than single variable** use **brackets inside parenthesis** ***disp([ ])*** and separate them with space or comma (,),,(see figure 3)
- The command ***disp(.)*** allow you to classify the result as in table, (see figure 4)

<	Smaller
>	Greater
<=	Smaller than or equal to
>=	Greater than or equal to
&&	And operator
~=	Not equal
==	Equal to





```

MATLAB R2014a
HOME PLOTS APPS
FILE EDIT PUBLISH VIEW
Current Folder Editor: F:\Université Biskra\1er doctorat\Teaching\Course MATLAB (2LS1)\Dr. Salah\2024-2025\gr01.m
gr01.m
1 - clc
2 - clear
3 - x=1;
4 - while (x<5)
5 - f=exp (x);
6 - disp (f)
7 - x=x+1;
8 - end
2.72
7.39
20.09
54.60
fx >> |

```

Figure 2. An example of using the command *disp(.)* to show the result of single variable

```

MATLAB R2014a
HOME PLOTS APPS
FILE EDIT PUBLISH VIEW
Current Folder Editor: F:\Université Biskra\1er doctorat\Teaching\Course MATLAB (2LS1)\Dr. Salah\2024-2025\gr01.m
gr01.m
1 - clc
2 - clear
3 - x=1;
4 - while (x<5)
5 - f=exp (x);
6 - disp([x, f])
7 - x=x+1;
8 - end
1.00 2.72
2.00 7.39
3.00 20.09
4.00 54.60
fx >> |

```

Figure 3. An example of using the command *disp(.)* to show the result of multiple variables



```

MATLAB R2014a
HOME PLOTS APPS EDIT PUBLISH VIEW
FILE Current Folder Editor - F:\Université Biskra\1er doctorat\Teaching\Course MATLAB (2LS1)\Dr. Salah\2024-2025\gr01.m
gr01.m
1 clc
2 clear
3 x=1;
4 while (x<5)
5 f=exp(x);
6 g=log(x);
7 h=sqrt(x);
8 disp([x,f,g,h])
9 x=x+1;
end

```

```

Command History
edit
% gr02
%-- 11/21/20...
edit
>9 gr06
sqrt(10/1)
2x gr06
%-- 11/22/20...
gr06
7x gr01

```

```

Command Window
1.00 2.72 0 1.00
2.00 7.39 0.69 1.41
3.00 20.09 1.10 1.73
4.00 54.60 1.39 2.00
fx>> |

```

```

Workspace
Name Value
f 54.5982
g 1.3863
h 2
x 5
y 5

```

Figure 4. An example of using the command *disp(.)* to show the result of multiple functions

## 2.2. How many loops you should selected ?

A number of variables are only the variable that control the number of loops used, which means **number of variables** should be **equal number of loops**.

```

MATLAB R2014a
HOME PLOTS APPS EDIT PUBLISH VIEW
FILE Current Folder Editor - D:\Université Biskra\1er doctorat\Teaching\Course MATLAB (2LS1)\Dr. Salah\2023-2024\LM07\tp_07.m
tp_07.m
1 clc;clear
2 a = 1 ; b = 2 ;
3 while (a<=5)
4 while (b<=10)
5 s = a*b ;
6 p = (a+b)*2 ;
7 disp([s p])
8 a = a + 1 ;
9 b = b + 2 ;
10 end
11 end

```

```

Command Window
2 6
8 12
18 18
32 24
fx>> |

```

```

Workspace
Name Value
a 6
b 12
p 30
s 50

```

Figure 5. An example of using two-loops

**Note.** Rather than using many loops you can use one loop and the variables can be declare in one line all together and connect them with *and* see figure 6 .



```

1- clc;clear
2- a = 1 ; b = 2 ;
3- while (a<=5) && (b<=10)
4-     s = a*b ;
5-     p = (a+b)^2 ;
6-     disp([s p])
7-     a = a + 1 ;
8-     b = b + 2 ;
9- end

```

Figure 6. An example of using one loop with many variables

### 3. 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.