Introduction to Matlab

Table of Contents


```
clear variables % clean all variables
close all       % close all windows
clc             % clear the "command Window"
```

1 - Global presentation

- Matlab
- Simulink

```
% simulink
```

- Applications (Apps)
- Matlab Help

```
help % or select a function and then F1
```

```
New to MATLAB? See resources for Getting Started.

To view the documentation, open the Help browser.
```

- Matlab central : https://fr.mathworks.com/matlabcentral/
- Getting Started : https://fr.mathworks.com/videos.html

2 - The basics of programming

Vectors, matrices, structures, cells

- Variables

```
a = 2
b = 3;  % the ; remove the display of the performed operation
c = "bleu"
```

Variable names should have a signification and not be too long. Exemple : programmedTrajectory => cmdTraj

- Vectors

```
x = [1, 2, 3, 4, 5] % The "," are not necessary for a line vector
```

```
x = 1×5
     1     2     3     4     5
```

```
y = [6; 7; 8; 9; 10]
```

```
y = 5×1
     6
     7
     8
     9
    10
```

- struc

```
s.x = [1, 2, 3];
s.y = [4, 5, 6];
s.t = "t"
```

```
s = struct with fields:
    x: [1 2 3]
    y: [4 5 6]
    t: "t"
```

```
s % display the content of the variable "s"
```

```
s = struct with fields:
    x: [1 2 3]
    y: [4 5 6]
    t: "t"
```

- cells

```
C = {"A",  "réponse", "10"; ...
     "B", "Question", 20}
```

```
C = 2×3 cell
```

	1	2	3
1	"A"	"réponse"	"10"
2	"B"	"Question"	20

- matrices

```
M = [ 1,  2,  3,  4,  5,  6; ... % line breaks are not required
      6,  7,  8,  9, 10, 11; ... % but they improve readability
     12, 13, 14, 15, 16, 17]
```

```
M = 3×6
     1     2     3     4     5     6
     6     7     8     9    10    11
    12    13    14    15    16    17
```

Go to the workspace to fully understand the nature of these different elements.

Handling of the various classes

- Retrieve an element

```
M(3,2) % (row, column)
```

```
ans = 13
```

```
M(3,end) % "end" for the last element
```

```
ans = 17
```

```
s.y(1)
```

```
ans = 4
```

- Operator ":"

```
v = 0:9 % can also be written v = 0:9 (by default interval is 1)
```

```
v = 1×10
     0     1     2     3     4     5     6     7     8     9
```

```
w = 0:2:9 % be careful about what happens at the end
```

```
w = 1×5
```

```
     0     2     4     6     8
```

- Retrieve multiple elements

```
M(1,1:4)
```

```
ans = 1×4
     1     2     3     4
```

```
M(:,end)
```

```
ans = 3×1
     6
    11
    17
```

```
M(2,2:5)
```

```
ans = 1×4
     7     8     9    10
```

- To go further : https://fr.mathworks.com/company/newsletters/articles/matrix-indexing-in-matlab.html

Logic operators

```
clear variables

M = [ 1,  2,  3,  4,  5,  6; ...
      6,  7,  8,  9, 10, 11; ...
     12, 13, 14, 15, 16, 17]
```

```
M = 3×6
     1     2     3     4     5     6
     6     7     8     9    10    11
    12    13    14    15    16    17
```

- greater or equal **>=** / less or equal **<=**

```
Ms = (M>=8)
```

```
Ms = 3×6 logical array
   0   0   0   0   0   0
   0   0   1   1   1   1
   1   1   1   1   1   1
```

```
Mi = (M<3)
```

```
Mi = 3×6 logical array
   1   1   0   0   0   0
   0   0   0   0   0   0
   0   0   0   0   0   0
```

- and **&**

```
Me = (~Ms & ~Mi)
```

```
Me = 3×6 logical array
   0   0   1   1   1   1
   1   1   0   0   0   0
   0   0   0   0   0   0
```

- Or **|**

```
Mo = (Ms | Mi)
```

```
Mo = 3×6 logical array
   1   1   0   0   0   0
   0   0   1   1   1   1
   1   1   1   1   1   1
```

- Not a Number **~** or **not()**

```
M(3,4) = NaN
```

```
M = 3×6
     1     2     3     4     5     6
     6     7     8     9    10    11
    12    13    14   NaN    16    17
```

```
isnan(M)
```

```
ans = 3×6 logical array
   0   0   0   0   0   0
   0   0   0   0   0   0
   0   0   0   1   0   0
```

```
~any(isnan(M), 1)
```

```
ans = 1×6 logical array
   1   1   1   0   1   1
```

```
E= M(:, ~any(isnan(M), 1)) % remove the columns where there is a NaN
```

```
E = 3×5
     1     2     3     5     6
     6     7     8    10    11
    12    13    14    16    17
```

Algorithmic structure

- *if, while, for, switch* ...
- more advanced function : *parfor* ...

Scripts, functions, workspace

- Script : Scripts are simply files containing a sequence of MATLAB statements.
- functions : Functions make use of their own local variables and accept input arguments.
- Workspace : The workspace consists of the set of variables built up during a session of using the MATLAB software and stored in memory. You add variables to the workspace by using functions, running MATLAB code, and loading saved workspaces.

mode debug

Ctrc+C in command Window allows ending a task

3 - Some good practices to make a Matlab code

- **Read the matlabs-guidelines**
- Always ask yourself if the function already exists either in matlab or in the community.
- Beware of matlab warnings.

Interactive example n°1

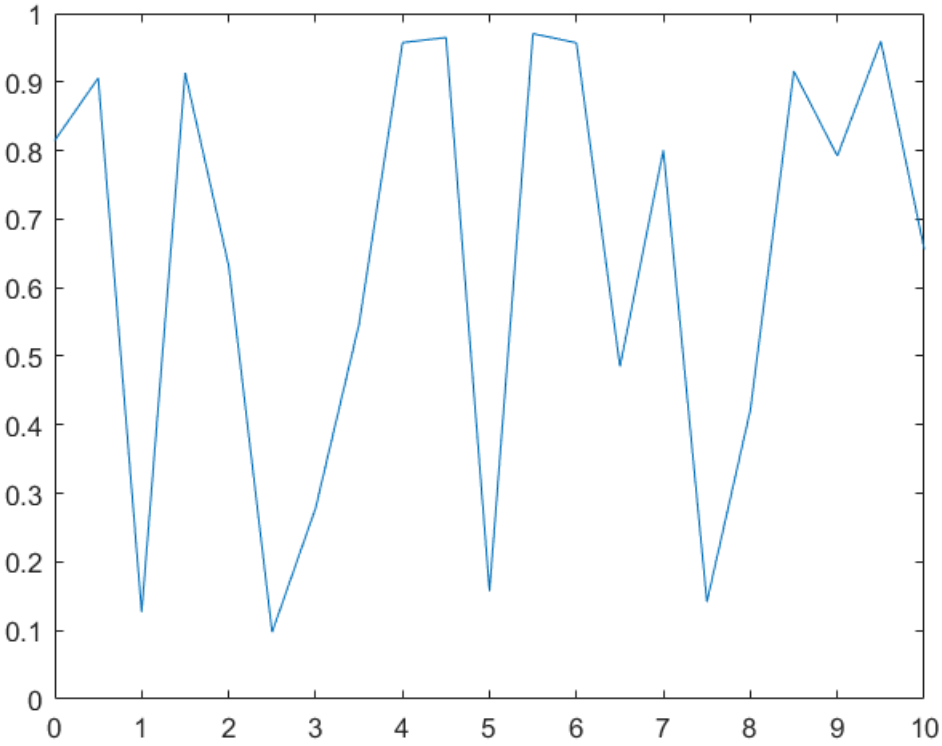
```
open("example_1.mlx")
```

4 - Figure

Creation

```
t = 0:0.5:10;
x = rand(length(t),1);

h = figure('Color',[1 1 1]);
plot(t,x)
```



Other important functions : *hold on*

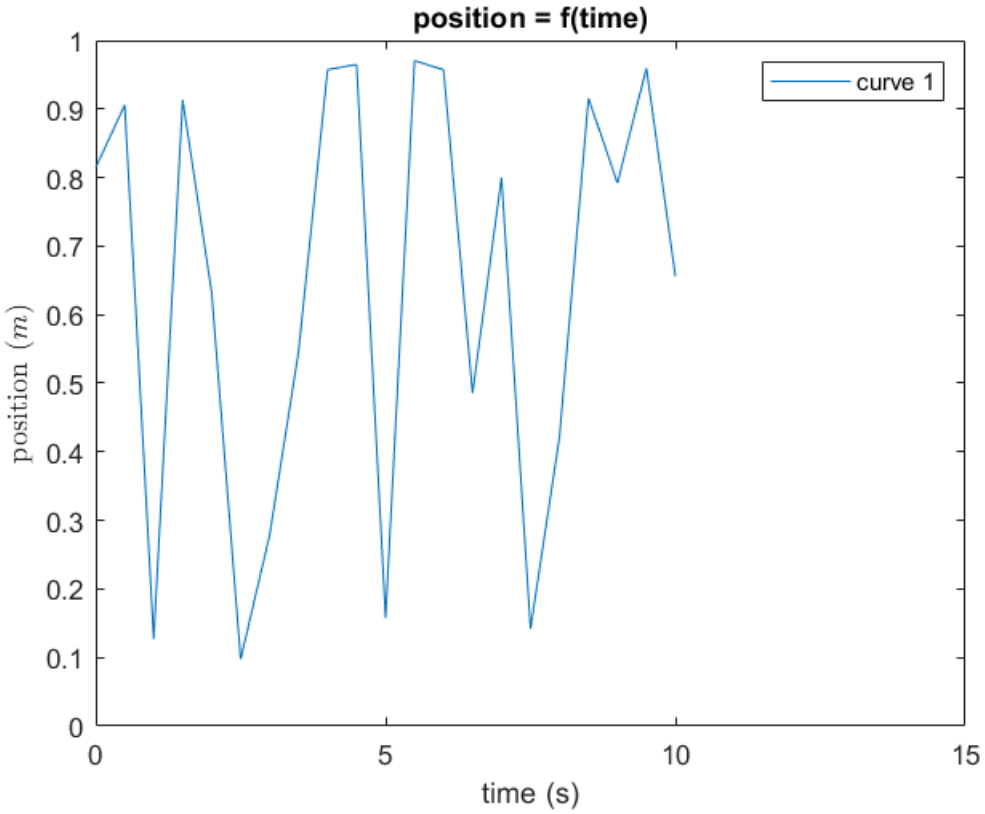
Formatting

```
title('position = f(time)');

xlabel('time (s)')
ylabel('position ($m$)','Interpreter',"latex")

xlim([0 15])
ylim([0 1])

legend('curve 1')
```



possible modification with the mouse

automatic code generation

Publication

- Matlab tools

```
% doesn't work on live editor

%saveas(h,'1_figure_test.png')
%saveas(h,'1_figure_test.pdf')

open('figureMatlab.m')
```

- package : *export_fig*

https://fr.mathworks.com/matlabcentral/fileexchange/23629-export_fig

Download

Unzip

Place in the desired location

add to path : explanation

install ghostscript

https://www.ghostscript.com/download/gsdnld.html

```
% doesn't work on live editor

%export_fig 2_figure_test -pdf
```

Various functions

- *plot3, surface, colorbar,*
- *subplot*

AppDesigner

- Tools for creating graphical user interfaces

```
appdesigner
```

5 - Exercises done in class

```
clear variables
```

Exercise n°1

- Create a function that takes as input the trajectory and returns a vector containing the distance between each point.

The trajectory is represented as a list of points with coordinates in columns.

```
Traj = [0 2 3 5 7 8 5 1 2; ... % coordonnée x
        0 5 6 4 5 7 8 9 1; ... % coordonnée y
        0 1 5 6 8 4 5 6 2];    % coordonnée z
```

- Plotting the curvilinear abscissa as a function of time

```
t = [0 1 2 3 4 5 6 7 8];   % vecteur temps
```

Definition of the curvilinear abscissa: length of an arc from its origin.

function that can be used in exercise n°1 : diff, sum, cumsum, sqrt, plot,

To go further :

- add verification of the input parameter

Exercise n°2

- Write a function that takes as input a vector of 6 components #1,#2 ... #6 and two variables a and v and outputs the following string : "movej(p[#1,#2,#3,#4,#5,#6],a,v)"

Example :

Inputs

```
a = 1.5;
v = 2;
po = [1, 2, 1, 3, 1, 4];
```

Output

```
"movej(p[1, 2, 1, 3, 1, 4], 1.5, 2)"
```

```
ans =
"movej(p[1, 2, 1, 3, 1, 4], 1.5, 2)"
```

- Add in the previous function a check of the input dimensions and return an error message if their dimensions are wrong.

function that can be used in exercise n°2 : num2str, size, warning, disp, isnumeric, varargin

References

- https://www.gipsa-lab.grenoble-inp.fr/~coriandre.vilain/mes_documents/Cours/Formation_Matlab_GIPSA_2013.pdf
- https://homepages.laas.fr/yariba/enseignement/slides-matlab.pdf

Example of well-written code :

- https://fr.mathworks.com/matlabcentral/fileexchange/34869-distance2curve?s_tid=prof_contriblnk
- https://fr.mathworks.com/matlabcentral/fileexchange/34874-interparc?s_tid=prof_contriblnk
- https://fr.mathworks.com/matlabcentral/fileexchange/34871-arclength?s_tid=prof_contriblnk