

16/05/2022

TP 0 : Prise en main du Logiciel Matlab

MATLAB®

PRÉPARÉE PAR
TAFFAH ACHRAF

PROFESSEUR
RIAD Manal



Table des matières

Introduction.....	3
Partie pratique	4
Quelques exemples élémentaires à tester :	4
Opérations numériques :	4
Comment déclarer des variables (Signe =)	4
Les variables peuvent s'afficher sous différents formats.....	4
Matrices, vecteurs et tableaux :	4
Calcule matriciel.....	10
Autre opération de base :	14

Introduction

Matlab est un logiciel de calcul numérique commercialisé par la société Math-Works¹. Il a été initialement développé à la fin des années 70 par Cleve Moler, professeur de mathématique à l'université du Nouveau-Mexique puis à Stanford, pour permettre aux étudiants de travailler à partir d'un outil de programmation de haut niveau et sans apprendre le Fortran ou le C. Matlab signifie Matrix laboratory. Il est un langage pour le calcul scientifique, l'analyse de données, leur visualisation, le développement d'algorithmes. Son interface propose, d'une part, une fenêtre interactive type console pour l'exécution de commandes, et d'autre part, un environnement de développement intégré (IDE) pour la programmation d'applications. Matlab trouve ses applications dans de nombreuses disciplines. Il constitue un outil numérique puissant pour la modélisation de systèmes physiques, la simulation de modèles mathématiques, la conception et la validation (tests en simulation et expérimentation) d'applications. Le logiciel de base peut être complété par de multiples toolboxes, c'est-à-dire des boîtes à outils. Celles-ci sont des bibliothèques de fonctions dédiées à des domaines particuliers. Nous pouvons citer par exemple : l'Automatique, le traitement du signal, l'analyse statistique, l'optimisation...

Partie pratique

Quelques exemples élémentaires à tester :

Opérations numériques :

```
>> 5*6, 2^5
```

```
ans =
```

```
30
```

```
ans =
```

```
32
```

```
>> 3+5*2^5
```

```
ans =
```

```
163
```

Comment déclarer des variables (Signe =)

```
>> x=2
```

```
x =
```

```
2
```

```
>> y=x^5
```

```
y =
```

```
32
```

```
>> y/x
```

```
ans =
```

```
16
```

Les variables peuvent s'afficher sous différents formats

La fonction **sqrt** permet de calculer la racine carrée d'un nombre donné.

```
>> a=sqrt(3)
```

```
a =
```

```
1.7321
```

```
>> format long, b=sqrt(3)
```

```
b =
```

```
1.732050807568877
```

```
>> format short
```

Matrices, vecteurs et tableaux :

Who permet d'afficher toutes les variables utilisées.

```
>> who  
  
Your variables are:  
  
a    ans  b    x    y
```

La commande **Clear** permet de supprimer toutes les variables de Workspace.

```
>> clear
```

La déclaration des matrices avec plusieurs méthodes.

```
>> a=[1,2,3;4 5 6]
```

a =

1	2	3
4	5	6

```
>> a(1,2), a(2,3)
```

ans =

2

ans =

6

A' permet de d'afficher le transposé d'une matrice nommé a.

```
>> a(2,3)=10
```

a =

1	2	3
4	5	10

```
>> a'
```

ans =

1	4
2	5
3	10

Rand renvoie un nombre avec une rand probabilité uniforme

```
>> rand(1,3), rand(2)
|
ans =

    0.8147    0.9058    0.1270

ans =

    0.9134    0.0975
    0.6324    0.2785
```

Zeros Renvoie une matrice nulle

```
>> zeros(3), ones(3,2)

ans =

    0    0    0
    0    0    0
    0    0    0

ans =

    1    1
    1    1
    1    1
```

Eye retourne la matrice carrée identité

```
>> eye(3), eye(2,3)

ans =

    1    0    0
    0    1    0
    0    0    1

ans =

    1    0    0
    0    1    0
```

Magic retourne une matrice correspondante au carré magique

```
>> magic(3)

ans =

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

Diag est une instruction permet de construire une matrice particulière.

```
>> b=[1 4 5], diag(b)

b =

     1     4     5

ans =

     1     0     0
     0     4     0
     0     0     5
```

Whos liste et décrit les variables en mémoire

```
>> whos

Name      Size      Bytes  Class  Attributes

a         2x3         48  double

ans       3x3         72  double

b         1x3         24  double
```

La création d'une matrice de deux lignes et trois colonnes à l'aide de la commande **zeros**

```
>> s1=zeros(20,25) ; s2=zeros(2,3)

s2 =

     0     0     0
     0     0     0
```

L'opérateur « : » est très utile pour construire un tableau ou en extraire une partie, une ligne ou une colonne

Dans cet exemple nous allons créer un tableau qui contient des éléments de -3 jusqu'à 3.

```
>> -3:3  
  
ans =  
  
-3    -2    -1     0     1     2     3
```

Dans cet exemple nous allons créer deux tableaux.

```
>> x=-3:.3:3  
  
x =  
|  
Columns 1 through 11  
-3.0000 -2.7000 -2.4000 -2.1000 -1.8000 -1.5000 -1.2000 -0.9000 -0.6000 -0.3000 0  
Columns 12 through 21  
0.3000 0.6000 0.9000 1.2000 1.5000 1.8000 2.1000 2.4000 2.7000 3.0000
```

Dans cet exemple nous allons afficher les 10 éléments du tableau.

```
>> x(2:12)  
  
ans =  
  
-2.7000 -2.4000 -2.1000 -1.8000 -1.5000 -1.2000 -0.9000 -0.6000 -0.3000 0 0.3000
```

Dans cet exemple nous allons afficher les 5 éléments du tableau.

```
>> x(9:-2:1)  
  
ans =  
  
-0.6000 -1.2000 -1.8000 -2.4000 -3.0000
```

Dans cet exemple nous allons afficher 11 et 19.


```
>> x=10:100; x(2), x(10)

ans =

    11

ans =

    19
```

Dans cet exemple nous allons afficher les 5 éléments existents dans le tableau entre 40 et 60.

```
>> x(40:5:60)

ans =

    49    54    59    64    69
```

Dans cet exemple nous allons créer une matrice qui contient trois lignes et les colonnes du chaque faire appartient à un intervalle donné.

```
>> a=[1:6 ; 2:7 ; 4:9]

a =

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

Dans cet exemple nous allons créer une matrice avec trois lignes et des colonnes ayant des valeurs appartenant d'un intervalle précis.

```
>> a, a(1, :), a(:, 2)

a =

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

ans =

     1     2     3     4     5     6

ans =

     2
     3
     5
```

Rand renvoie un nombre avec une rand probabilité uniforme

```
>> s=rand(20,5); s(6:7, 2:4)

ans =

     0.2769     0.6551     0.8143
     0.0462     0.1626     0.2435
```

Calcule matriciel

Après avoir nous llons entré les données, nous aessayant d'effectuer les opérations ci-dessous.

La création de la matrice a avec trois lignes et trois colonnes, et le vecteur b qui contient trois nombres.

```
>> a=[1 2 3; 4 5 6; 7 8 10] , b=[1 1 1]'
```

a =

1	2	3
4	5	6
7	8	10

b =

1
1
1

La multiplication de la matrice a par 2, et la division de cette matrice par 4

```
>> 2*a, a/4
```

ans =

2	4	6
8	10	12
14	16	20

ans =

0.2500	0.5000	0.7500
1.0000	1.2500	1.5000
1.7500	2.0000	2.5000

L'addition de la matrice a par une autre matrice, cette dernière a été former le vecteur b.

```
>> a+[b,b,b]
```

ans =

2	3	4
5	6	7
8	9	11

La multiplication de la matrice a par le vecteur b, ce qui permet d'afficher un vecteur de trois éléments.

```
>> a*b  
  
ans =  
  
     6  
    15  
    25
```

Nous ne pouvons pas multiplier un vecteur avec une matrice, l'inverse est vrai.

```
>> b*a  
Error using *  
Incorrect dimensions for matrix multiplication. Check that the number of columns in the first matrix matches the number of rows in the second matrix. To operate on each element of the matrix individually, use TIMES (.* ) for elementwise multiplication.
```

Nous allons multiplier le transpose de du vecteur b avec la matrice a.

```
>> b'*a  
  
ans =  
  
    12    15    19
```

Nous allons multiplier le transpose du vecteur b avec le vecteur b, puis nous allons inverser l'opération.

```
>> b'*b , b*b'  
  
ans =  
  
     3  
  
ans =  
  
     1     1     1  
     1     1     1  
     1     1     1
```

Nous allons afficher la puissance carrée de la matrice a.

```
>> a^2  
  
ans =  
  
    30    36    45  
    66    81   102  
   109   134   169
```

Nous allons affecter 1 a la matrice a, et 2 au vecteur b.

```
>> a+1, b+2  
  
ans =  
|  
    2    3    4  
    5    6    7  
    8    9   11  
  
ans =  
  
    3  
    3  
    3
```

Nous allons exécuter l'opération terme à terme dans cet exemple.

```
>> a.^2, a.*a  
  
ans =  
  
    1    4    9  
   16   25   36  
   49   64  100  
  
ans =  
  
    1    4    9  
   16   25   36  
   49   64  100
```

Nous allons exécuter l'opération terme à terme par la matrice a et le vecteur b.

```
>> a.*b  
  
ans =  
  
    1     2     3  
    4     5     6  
    7     8    10
```

Nous allons diviser 1 par a, et 1 par a puissance 2.

```
>> 1./a, 1./a.^2  
  
ans =  
  
    1.0000    0.5000    0.3333  
    0.2500    0.2000    0.1667  
    0.1429    0.1250    0.1000  
  
ans =  
  
    1.0000    0.2500    0.1111  
    0.0625    0.0400    0.0278  
    0.0204    0.0156    0.0100
```

Autre opération de base :

```
>> m1 = [1 2 3 ; 4 5 6 ; 7 8 9]  
  
m1 =  
  
    1     2     3  
    4     5     6  
    7     8     9
```

```
>> m2 = [1:1:3 ; 11:1:13]  
  
m2 =  
  
     1     2     3  
    11    12    13
```

```
>> m2'
```

```
ans =
```

```
1    11
2    12
3    13
```

```
>> m2'*10
```

```
ans =
```

```
10    110
20    120
30    130
```

```
>> m2^2
```

```
Error using ^
```

```
Incorrect dimensions for raising a matrix to a power. Check that the matrix is square and the power is a scalar.  
operate on each element of the matrix individually, use POWER (.^) for elementwise power.
```

```
>> m2 .^2
```

```
Unrecognized function or variable 'm2'.
```

```
>> det(m1)
```

```
ans =
```

```
-9.5162e-16
```

```
>> m1^(-1)
```

```
ans =
```

```
1.0e+16 *  
0.3153 -0.6305 0.3153  
-0.6305 1.2610 -0.6305  
0.3153 -0.6305 0.3153
```

```
ans =
```

```
1.0e+16 *  
0.3153 -0.6305 0.3153  
-0.6305 1.2610 -0.6305  
0.3153 -0.6305 0.3153
```

```
>> inv(m1)
```

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 2.202823e-18.
```

```
>> pinv(m1)
```

```
ans =
```

```
-0.6389   -0.1667    0.3056  
-0.0556    0.0000    0.0556  
 0.5278    0.1667   -0.1944
```

```
>> ones(3)
```

```
ans =
```

```
 1    1    1  
 1    1    1  
 1    1    1
```

```
>> m1 * m2'
```

```
ans =
```

```
14    74  
32   182  
50   290
```

```
>> eye(4)
```

```
ans =
```

```
 1    0    0    0  
 0    1    0    0  
 0    0    1    0  
 0    0    0    1
```

```
>> zeros(2,5)
```

```
ans =
```

```
 0    0    0    0    0  
 0    0    0    0    0
```

```
>> diag([1 : 4])
```

```
ans =
```

```
 1    0    0    0  
 0    2    0    0  
 0    0    3    0  
 0    0    0    4
```

```
>> length(m2)
```

```
ans =
```

```
3
```

```
>> t=0 : 0.1 :2
```

```
t =
```

```
Columns 1 through 11
```

```
 0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000    0.7000    0.8000    0.9000    1.0000
```

```
Columns 12 through 21
```

```
1.1000    1.2000    1.3000    1.4000    1.5000    1.6000    1.7000    1.8000    1.9000    2.0000
```



```
>> x=exp(t)

x =

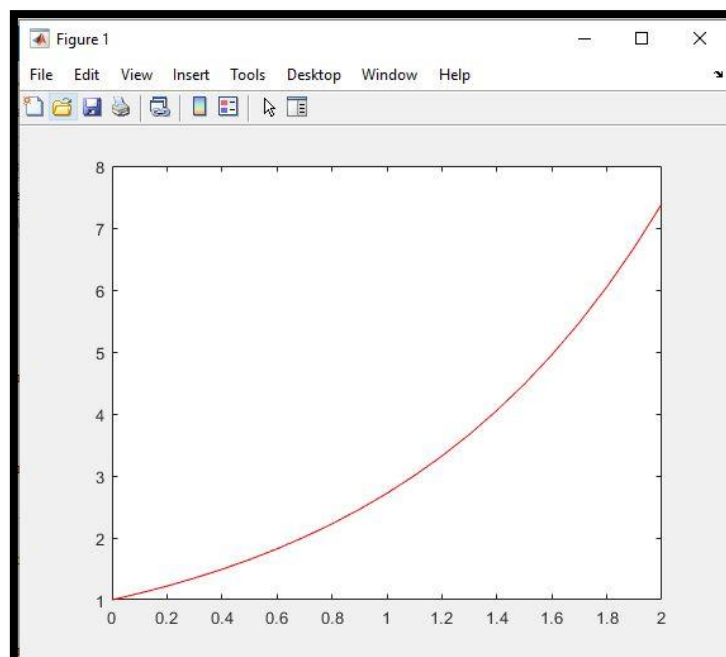
Columns 1 through 11

    1.0000    1.1052    1.2214    1.3499    1.4918    1.6487    1.8221    2.0138    2.2255    2.4596    2.7183

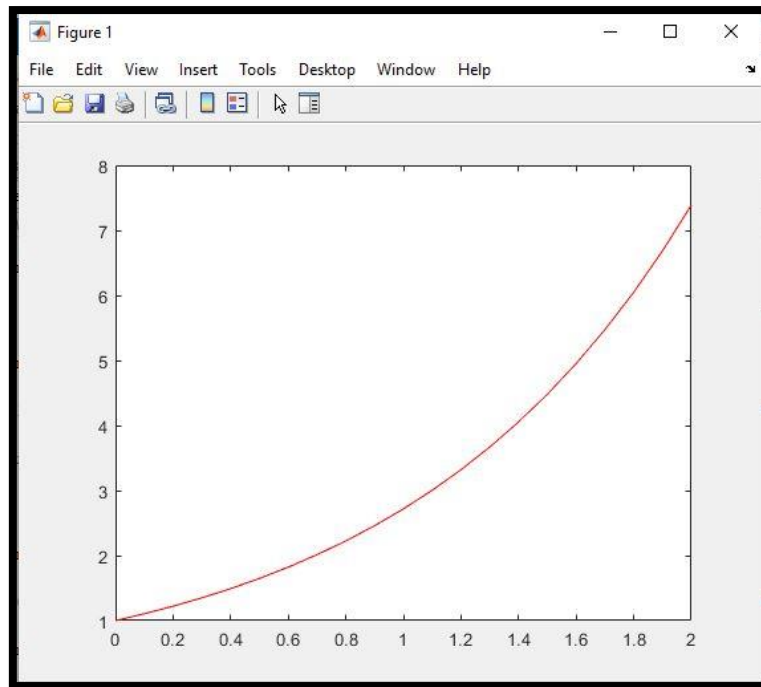
Columns 12 through 21

    3.0042    3.3201    3.6693    4.0552    4.4817    4.9530    5.4739    6.0496    6.6859    7.3891
```

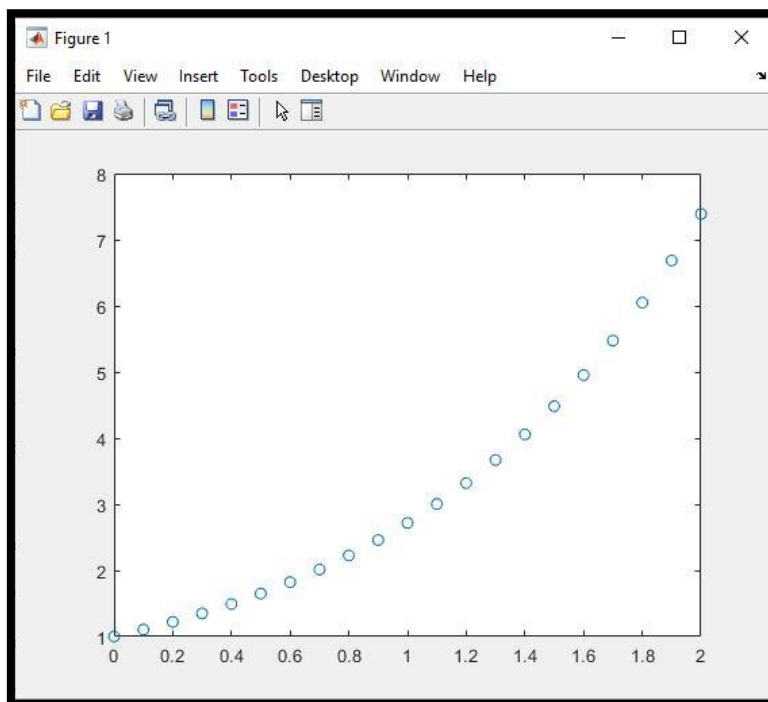
```
>> plot(t,x)
```



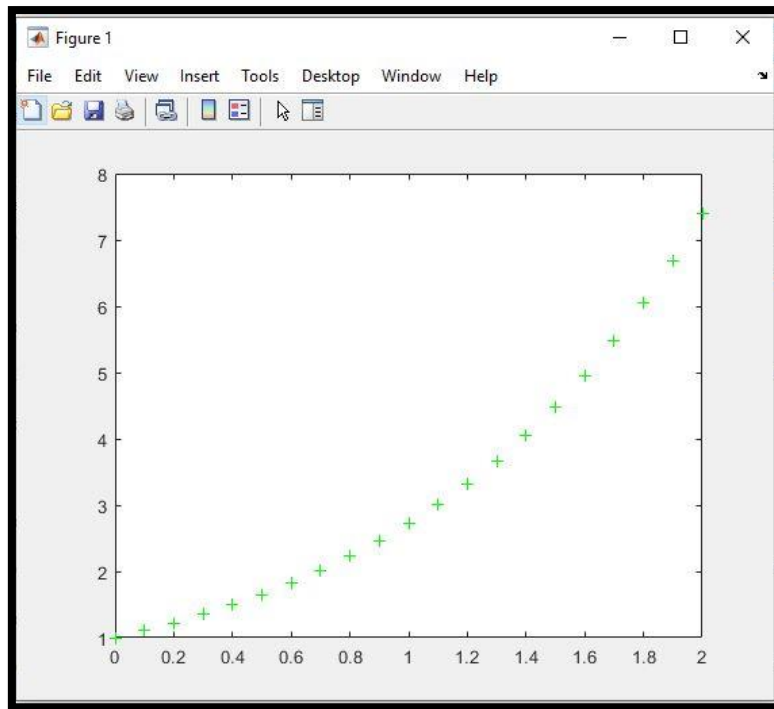
```
>> plot(t,x,'r')
```



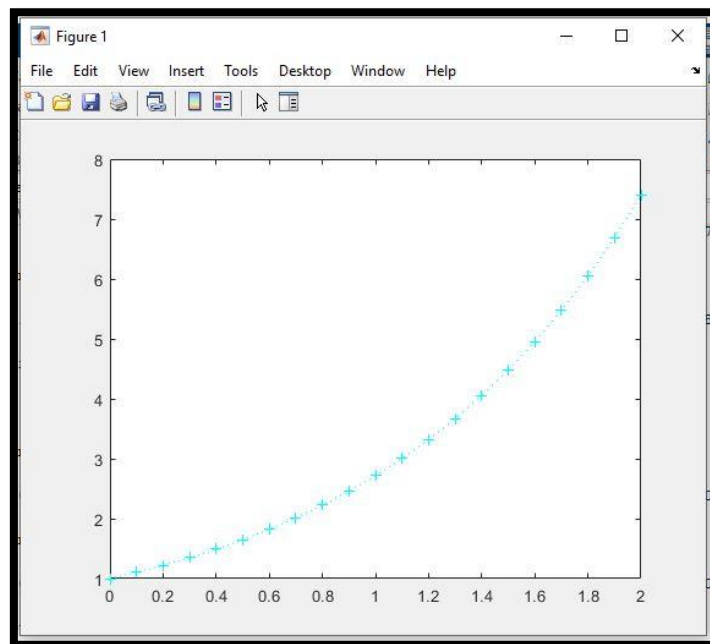
```
>> plot(t,x,'o')
```



```
>> plot(t,x,'+g')
```



```
>> plot(t,x,'c+:')
```



```
>> A=magic(5)
```

```
A =
```

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

```
>> [m n]=size(A)
```

```
m =
```

```
5
```

```
n =
```

```
5
```

```
>> x=A(:,1);y=A(:,2);a=A(:,3)
```

```
a =
```

```
1  
7  
13  
19  
25
```

```
>> triu(A); tril(A)
```

```
ans =
```

17	0	0	0	0
23	5	0	0	0
4	6	13	0	0
10	12	19	21	0
11	18	25	2	9

```
>> B = rot90(A)
```

```
B =
```

15	16	22	3	9
8	14	20	21	2
1	7	13	19	25
24	5	6	12	18
17	23	4	10	11

```
>> diag(A)
```

```
ans =
```

```
17  
5  
13  
21  
9
```

```
>> diag(B)
```

```
ans =
```

```
15  
14  
13  
12  
11
```

```
>> trace(A) , trace(B)
```

```
ans =
```

```
65
```

```
ans =
```

```
65
```

```
>> det(A), det(B)
```

```
ans =  
  
5.0700e+06
```

```
ans =  
  
5.0700e+06
```

```
>> C = inv(A)
```

```
C =  
  
-0.0049    0.0512   -0.0354    0.0012    0.0034  
0.0431   -0.0373   -0.0046    0.0127    0.0015  
-0.0303    0.0031    0.0031    0.0031    0.0364  
0.0047   -0.0065    0.0108    0.0435   -0.0370  
0.0028    0.0050    0.0415   -0.0450    0.0111
```

```
>> C*A
```

```
ans =  
  
1.0000   -0.0000   -0.0000   -0.0000   -0.0000  
-0.0000    1.0000   -0.0000    0.0000    0.0000  
-0.0000    0.0000    1.0000    0.0000   -0.0000  
0.0000    0.0000   -0.0000    1.0000    0.0000  
-0.0000   -0.0000    0.0000    0.0000    1.0000
```

```
ans =
```

```
1.0000   -0.0000   -0.0000   -0.0000   -0.0000  
-0.0000    1.0000   -0.0000    0.0000    0.0000  
-0.0000    0.0000    1.0000    0.0000   -0.0000  
0.0000    0.0000   -0.0000    1.0000    0.0000  
-0.0000   -0.0000    0.0000    0.0000    1.0000
```

```
>> D = rref(A),
```

```
D =  
  
1    0    0    0    0  
0    1    0    0    0  
0    0    1    0    0  
0    0    0    1    0  
0    0    0    0    1
```

Conclusion

Pourquoi MATLAB ! Le succès actuel de MATLAB vient de sa simplicité de prise en main et d'utilisation. De plus, il existe des boîtes à outils (toolbox) optionnelles mais très utiles dans certains domaines comme l'optimisation, le traitement du signal et de l'image, l'apprentissage (réseaux de neurones. . .), l'automatique (Simulink), etc. Ce logiciel est de plus très utilisé tant dans le monde industriel que dans le monde universitaire.