

TP 3 : Lecture, affichage et sauvegarde des données

Comme tout langage de programmation MATLAB permet de définir des données variables. Une variable est désignée par un identificateur qui est formé d'une combinaison de lettres et de chiffres. Le premier caractère de l'identificateur doit nécessairement être une lettre. Attention, MATLAB différencie majuscules et minuscules! Ainsi **X33** et **x33** désignent deux variables distinctes. Les variables sont définies au fur et à mesure que l'on donne leurs noms (identificateur) et leurs valeurs numériques ou leurs expressions mathématiques. L'utilisation de variables avec MATLAB ne nécessite pas de déclaration de type ou de dimension. Le type et la dimension d'une variable sont déterminés de manière automatique à partir de l'expression mathématique ou de la valeur affectée à la variable.

1. Les 4 types de données

Les 3 principaux types de variables utilisés par MATLAB sont : les réels, les complexes et les chaînes de caractères. Le type logique est associé au résultat de certaines fonctions.

Pour MATLAB toute variable est considérée comme étant un tableau d'éléments d'un type donné.

MATLAB différencie trois formes particulières de tableaux. Les **scalaires** qui sont des tableaux à une ligne et une colonne. Les **vecteurs** qui sont des tableaux à une ligne ou à une colonne. Les **matrices** qui sont des tableaux ayant plusieurs lignes et colonnes. Une variable MATLAB est donc toujours un tableau que l'on appelle variable scalaire, vecteur ou matrice suivant la forme du tableau.

1.1 Le type complexe

L'unité imaginaire est désignée par i ou j (se transforme en i). Les nombres complexes peuvent être écrits sous forme cartésienne $a+ib$ ou sous forme polaire re^{it} .

Les différentes écritures possibles sont :

$a+i*b$ (ou $a+b*i$) et $r*\exp(i*t)$ (ou $r*\exp(t*i)$)

Avec a,b,r et t des variables de type réel et $i = \text{sqrt}(-1)$.

Mais on peut écrire :

```
>> z = 2+i*5 >> z = 2+5*i
```

```
>> z = 2+5i (mais z = 2+i5 error! i5 = ???)
```

```
>> z = 7*exp(i*3)
```

```
>> z = 7*exp(3*i)
```

```
>> z = 7*exp(3i)
```

Voici quelque commande concernant les nombres complexes : si Z est de type complexe

```
>> imag(Z) (retourne la partie imaginaire de Z)
```

```
>> real(Z) (retourne la partie réelle de Z)
```

```
>> abs(Z) (retourne le module de Z)
```

```
>> angle(Z) (retourne la partie imaginaire de Z)
```

```
>> conj(Z) (retourne le conjugué de Z (Z*))
```

1.2 Le type chaîne de caractères

Une chaîne de caractères est un tableau de caractères. Une donnée de type chaîne de caractère (char) est représentée sous la forme d'une suite de caractères encadrée d'apostrophes simples (').

Exemples:

```
>> ch1='bon'
ch1 =
bon
>> ch2='jour'
ch2 =
jour
>> whos
Name Size Bytes Class
ch1 1x3 6 char array
ch2 1x4 8 char array
```

Si une chaîne de caractères doit contenir le caractère apostrophe (') celui-ci doit être double dans la chaîne.

Exemple

```
>> rep='aujourd'hui'
      rep='aujourd'hui'
      |
Error: Missing MATLAB operator.
>> rep='aujourd"hui'
rep =
aujourd'hui
>> apos=""
apos =
'
```

La chaîne de caractères vide s'obtient par 2 apostrophes''.

1.3 Le type logique

Le type logique possède 2 formes : 0 pour faux et 1 pour vrai

Un résultat de type logique est retourné par certaines fonctions ou dans le cas de certains tests.

Exemple :

```
>> a=1; b=2;
>> test_E=(a==b)
test_E =
0
>> test_S=(a>b)
test_S =
0
>> test_I=(a<b)
test_I =
1
>> V=true
V =
1
>> F=false
F =
0
```

```
>> whos
Name Size Bytes Class
F 1x1 1 logical array
V 1x1 1 logical array
a 1x1 8 double array
b 1x1 8 double array
test_E 1x1 1 logical array
test_I 1x1 1 logical array
test_S 1x1 1 logical array
```

```
>> ch=[ch1,ch2]
ch =
bonjour
>> ch(1)
ans =
b
>> ch(7)
ans =
r
>> ch(1:3)
ans =
bon
>> ch3='soi' ;
>> ch=[ch(1:3),ch3,ch(7)]
ch =
bonsoir
```

1.4 Le type vecteur

1.5 Le type matrice

Variables spéciales

Ces noms de variables sont utilisés par Matlab : *pi ans inf i or j realmin realmax eps ans* etc.

```
>> pi
ans =
3.1416
>> ans
ans =
3.1416
>> eps
ans =
2.2204e-016
>> realmax
ans =
1.7977e+308
```

```
>> realmin
ans =
2.2251e-308
>> inf
ans =
Inf
>> nan
ans =
NaN
```

2. Lecture des données

Pour lire une variable simple on utilise la fonction *input* comme suite :

```
>> x=input('introduire la valeur de x=')
introduire la valeur de x=4
x =
4
```

3. Affichage des données

Pour afficher les données il suffit d'écrire le nom de la variable la commande **disp** permet d'afficher un tableau de valeurs numérique ou de caractères

Exemple

```
>> x=6;
>> disp(x)
6
>> disp('bonjour')
bonjour
>> a='bon'
a =
bon
>> disp(['le mot est : ',a])
le mot est : bon
>> disp(['la valeur de x est: ',num2str(x)])
la valeur de x est: 6
```

num2str : Convertie un nombre en caractère.

Les formats d'affichage des réels

MATLAB dispose de plusieurs formats d'affichage des réels. Par défaut le format est le format court à 5

chiffres. Les autres principaux formats sont:

format long : format long à 15 chiffres.

format short e : format court à 5 chiffres avec notation en virgule flottante.

format long e : format long à 15 chiffres avec notation en virgule flottante.

MATLAB dispose également des formats **format short g** et **format long g** qui utilise la << meilleure >> des deux écritures à virgule fixe ou à virgule flottante. On impose un format d'affichage en tapant l'instruction de format correspondante dans la fenêtre de contrôle, par exemple format long. Pour revenir au format par défaut on utilise la commande format ou format short.

Exemple :

```
>> pi
ans =
3.1416
>> format long
>> pi
ans =
3.14159265358979
>> format short e
>> pi^3
ans =
3.1006e+01
>> format short g
>> pi^3
ans =
31.006
```

```
x = [4/3 1.2345e-6]
format short
1.3333 0.0000
format short e
1.3333e+000 1.2345e-006
format short g
1.3333 1.2345e-006
format long
1.33333333333333 0.00000123450000
format long e
1.33333333333333e+000 1.23450000000000e-006
format long g
1.33333333333333 1.2345e-006
format bank
1.33 0.00
format rat
4/3 1/810045
format hex
3ff55555555555 3eb4b6231abfd271
```

2.5 Sauvegarde des données

Il est possible de sauvegarder une session MATLAB dans un fichier pour une utilisation ultérieure.

L'instruction **save nom-fic** enregistre toutes les variables de l'espace de travail dans le fichier nom-fic.mat.

Si aucun nom de fichier n'est précisé, le fichier par défaut est matlab.mat. Il est possible de ne sauver qu'une partie des variables (par exemple seulement la variable contenant le résultat d'un calcul) en utilisant l'instruction **save nom-fic nom-var** où nom-var est le nom de la (ou des) variable(s) à sauvegarder. Attention, seul le contenu des variables est sauvegardé et non pas l'ensemble des instructions effectuées durant la session. Pour ramener dans l'espace de travail les variables sauvegardées dans le fichier nom-fic.mat, taper **load nom-fic**.

Exemple :

```
>> x=2*pi/3, y=sin(x), z=cos(x)
x =
2.0944
y =
0.8660
z =
-0.5000
>> save data
>> save toto y z
>> who
Your variables are:
x y z
>> clear all
>> who
>> % vide
```

```
>> load toto
>> who
Your variables are:
y z
>> y
y =
0.8660
>> z
z =
-0.5000
>> x
??? Undefined function or variable 'x'.
>> load data
>> who
Your variables are:
x y z
```