

Feuille de TP N°01 – Quelques premiers pas à Matlab

Le but de ce premier TP est de familiariser les étudiants avec l'usage de Matlab. Pour effectuer ce TP il faut avoir installé Matlab (les étudiants sont libres à utiliser tout logiciel de calcul numérique compatible avec la syntaxe Matlab comme : Octave, Scilab ... etc.).

Exercice 01 : un premier pas

1. Identifier les différents composants de l'interface Matlab.
Les composants les plus importants à reconnaître sont : la fenêtre de commandes, le workspace et l'historique de commandes.
2. Quel est le rôle de chaque composant ?
La fenêtre de commandes : sert à taper et exécuter des commandes et voir leurs résultats.
Le workspace : affiche l'ensemble des variables déclarées avec leurs types, valeurs ... etc.
L'historique de commande : voir et réutiliser les commandes que vous avez tapées précédemment.
3. Taper sur la fenêtre de commandes la ligne suivante : `help sin`.
 - a. Interpréter le résultat obtenu.
Matlab affiche les entrées et les sorties de la fonction sin et explique son fonctionnement.
 - b. Utiliser la commande `help` pour lire la documentation d'une autre fonction Matlab de votre choix.

Exercice 02 : Matlab peut être utilisé comme une calculatrice ?

Utiliser la fenêtre de commandes pour effectuer les calculs suivants :

- $1 + 2 * 3$.
`>> 1 + 2 * 3`
- $(1 + 2) * 3$.
`>> (1 + 2) * 3`
- $1 + 2 * 3^4$.
`>> 1 + 2 * 3 ^ 4`
- $1 + (2 * 3)^4$.
`>> 1 + (2 * 3) ^ 4`
- $(1 + 2 * 3)^4$.
`>> (1 + 2 * 3) ^ 4`

Que signifie la variable `ans` affichée sur la fenêtre de commandes après chaque opération ?
Comment peut-on faire pour réutiliser les résultats des opérations précédentes ?

La variable `ans` est utilisée par Matlab par défaut pour affecter les résultats qui ne sont pas explicitement affectés à une autre variable.

Pour réutiliser ces résultats il faut les affecter à d'autres variables.

Utiliser des commandes Matlab pour initialiser les variables suivantes (l'expression est juste si Matlab répond `var = 50`) :

- $a = e^{\ln 50}$.
`>> a = exp(log(50))`
- $b = 100 \sin \frac{\pi}{6}$.
`>> b = 100 * sin(pi / 6)`
- $c = \lfloor e^\pi \rfloor + \lfloor \pi^e \rfloor + \lfloor \pi \rfloor + \lfloor e \rfloor$.
`>> c = floor(exp(pi)) + floor(pi ^ exp(1)) + floor(pi) + floor(exp(1))`
- $d = \frac{2^{\frac{(2*2)!}{2+2}} + 22 + 2^{2^2} - \sqrt{2^2}}{2^{2-\frac{2}{2}}}$.
`>> d = (2 ^ (factorial(2 + 2) / (2 + 2)) + 22 + 2 ^ 2 ^ 2 - sqrt(2 ^ 2)) / (2 ^ (2 - 2 / 2))`
- $e = \left(\left(\frac{5^5 - 5}{5} + 5^0 \right) * (5 - 5^0) \right)^{0.5}$.
`>> e = ((5 ^ 5 - 5) / 5 + 5 ^ 0) * (5 - 5 ^ 0) ^ 0.5`
- $f = 0.5 * (5 + 5)^{\frac{5^0}{0.5}}$.
`>> f = 0.5 * (5 + 5) ^ (5 ^ 0 / 0.5)`
- $g = 5 * \left(\frac{5}{0.5} + 5^0 \right) - 5$.
`>> g = 5 * (5 / 0.5 + 5 ^ 0) - 5`
- $k = \frac{3^{3!} - 3^{3-3^0}}{3^{3-3^0}} - 30$.
`>> k = (3 ^ factorial(3) - 3 ^ (3 - 3 ^ 0)) / (3 ^ (3 - 3 ^ 0)) - 30`
- $l = \frac{(10i)^2 \ln i^i}{\pi}$. Tel que : $i = \sqrt{-1}$.
`>> l = (10i ^ 2 * log(i ^ i)) / pi`

Exercice 03 : types de données

1. Citer quelques types de données supportés par Matlab.
Entier, Flottant, Complexe, Caractère, Chaîne de caractères, Tableaux ... etc.
2. Taper les commandes suivantes sur la fenêtre de commandes :
`>> a = 1`
`>> b = int32(1)`
`>> c = logical(1)`

```
>> d = 1 + 2i
```

```
>> e = 'chaine'
```

- a. Observer le type de chaque variable sur le workspace.

```
a -> double.
```

```
b -> int32.
```

```
c -> logical.
```

```
d -> double (complex).
```

```
e -> char.
```

- b. Que peut-on déduire ?

Le type utilisé par default par Matlab pour les variables contenant des valeurs numérique est `double` (nombre flottant en double précision) même si la valeur donnée est une valeur entière.

Matlab définit le type utilisé automatiquement selon la valeur affectée à la variable.

Les chaîne de caractères sont considérées comme des tableaux de caractères.

Il est possible de forcer un type de données en l'utilisant en tant que constructeur.

- c. Extraire la partie réel et la partie imaginaire de la variable `d` puis calculer son module et son argument.

Respectivement :

```
>> real(d)
```

```
>> imag(d)
```

```
>> abs(d)
```

```
>> arg(d)
```

Exercice 04 : construire des vecteurs et des matrices

Utiliser la fenêtre de commandes pour initialiser les tableaux suivants.

- $A = (1 \ 2 \ 3); B = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}; C = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}.$

```
>> A = [1 2 3]
```

```
>> B = [1 ; 2 ; 3]
```

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

- $D =$ un vecteur ligne qui comporte les nombres pairs strictement positifs et inférieurs à 1000.

```
>> D = 2 : 2 : 1000
```

- $E =$ un vecteur colonne qui comporte les nombres impairs positifs inférieurs à 1000.

```
>> E = 1 : 2 : 1000
```

- $F =$ un vecteur ligne qui comporte 100 nombres répartis uniformément sur $[3 ; 15]$.

```
>> F = linspace(3, 15, 100)
```

- G = une matrice magique d'ordre 3.
`>> G = magic(3)`
- H = une matrice identité d'ordre 4.
`>> H = eye(4)`
- I = une matrice aléatoire de 2 lignes et 3 colonnes.
`>> I = rand(2, 3)`