



# TP 5

## Instructions de contrôle (Boucles for et While, Instructions if et switch) (Matlab)



## Boucles et instructions de contrôle

Une structure répétitive, également appelée structure itérative ou encore boucle, permet de répéter plusieurs fois l'exécution d'une ou plusieurs instructions.

On identifie en règle générale deux types de répétitive :

- for
- while

### 1. boucle for

L'instruction for permet d'effectuer des boucles, c'est-à-dire d'effectuer une action donnée plusieurs fois.

La plupart du temps, une boucle est effectuée sur des valeurs entières, qui vont d'une valeur d'index de départ à une valeur d'index de fin.

#### a. syntaxe

```
for compteur=debut:pas:fin
```

```
    instructions.
```

```
end
```

Dans le script Matlab suivant, nous affichons la valeur de i, de 1 à 5.

```
for i = 1 : 5
```

```
    disp(i)
```

```
end
```

Nous pouvons utiliser une forme plus complète afin d'afficher les entiers impairs de 1 à 5. Pour ce faire, nous définissons le **pas**

```
for i = 1 : 2 : 5
```

```
    disp(i)
```

```
end
```

#### b. Exemple 1 :

Ecrire un programme pour calculer et afficher la somme de 100 premiers nombres entiers ?

```
s=0
```

```
for val=1:100
```

```
    s=s+val
```

```
end
```

```
disp(s," la somme est : ")
```

#### c. Exemple 2 :

Ecrire un programme permettant de lire un nombre entier N puis calcule son factoriel.

```
N=input("saisir un nombre")
```

```
f=1
```

```
for i=2:N
```

```
    f=f*i
```

```
end
```

```
disp (f," le factoriel est :")
```

## 2. boucle while

L'instruction while permet d'effectuer une boucle tant qu'une expression booléenne est vraie. Au début de la boucle, si l'expression est vraie, les instructions dans le corps de la boucle sont exécutées. Lorsque l'expression devient fausse (un événement qui doit survenir à un certain moment), la boucle est terminée.

### a. syntaxe

```
while(condition)
```

```
    //instructions si la condition est vraie
```

```
end
```

Dans le script suivant, nous calculons la somme des nombres de 1 à 10 avec une instruction while.

```
s=0
```

```
i=1
```

```
while (i<= 10)
```

```
    s=s+i
```

```
    i=i+1
```

```
end
```

### b. Exemple 1 :

Ecrire un programme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne.

```
n=input("saisir un nombre ")
```

```
while (n<1 | n>3)
```

```
    n=input("saisir un nombre")
```

```
end
```

```
disp("bravo !")
```

### c. Exemple 2 :

Ecrire un programme permettant de lire un nombre entier N puis calcule son factoriel.

```
N=input("saisir un nombre ")
```

```
i=1
```

```
f=1
```

```
while(i<N)
```

```
    i=i+1
```

```
    f=f*i
```

```
end
```

```
disp(f, "le factoriel est : ")
```

## 3. Instructions conditionnelle if

L'instruction conditionnelle if permet d'effectuer une suite d'instructions si une condition est vraie, et d'effectuer une autre suite d'instructions si la condition est fausse.

La syntaxe la suivante :

```
if condition then
instructions1(Code à exécuter si la condition est vraie)
else
instructions2 (Code à exécuter si la condition est fausse)
end
```

#### **Exemple 1 :**

Vérifiez si un nombre est positif ou négatif

```
a =input ("Donner la valeur de a:");
```

```
if a > 0
    disp("Le nombre est positif.")
else
    disp("Le nombre est non-positif.")
end
```

Exemple 2 : Vérifier si un nombre est compris entre 0 et 10

```
c = input ("Donner la Valeur de c:");
```

```
if c > 0 && c < 10
    disp("Le nombre est compris entre 0 et 10.")
else
    disp("Le nombre est en dehors de la plage.")
end
```

#### **4. Instructions break et continue**

L'instruction **break** permet d'interrompre une boucle. Habituellement, nous utilisons cette instruction dans les boucles où, une fois que certaines conditions sont satisfaites, les boucles ne doivent pas être continuées.

L'instruction **continue** permet de passer à la boucle suivante, de sorte que les instructions dans le corps de la boucle ne sont pas exécutées cette fois. Lorsque l'instruction continue est exécutée, Matlab ignore les autres instructions et passe directement l'instruction while ou for et évalue la boucle suivante.

##### **a. Exemple 1 :**

```
disp("début")
```

```
for i=1:10
    if(i==5)then
        break
    end
    disp(i)
end
disp("fin")
le programme affichera :
début
```

1  
2  
3  
4  
fin

**b. Exemple 2 :**

```
disp("début")
for i=1:10
    if(i==5)then
        continue
    end
    disp(i)
end
disp("fin")
le programme affichera :
début
```

1  
2  
3  
4  
6  
7  
8  
9  
10  
fin

### ***L'instruction switch***

Sa syntaxe est:

**Switch** var

**case** cst-1

*Séquence d'instructions-1*

**case** cst-2

*Séquence d'instructions-2*

.

.

.

**case** cst-N

*Séquence d'instructions-N*

**otherwise**

*Séquence d'instructions par défaut*

**End**

```
jj=input('donner le jour 1 :7 ');
switch jj
case 1
disp('samedi')
case 2
disp('dimanche')
case 3
disp('lundi')
case 4
disp('mardi')
case 5
disp('mercredi')
case 6
disp('jeudi')
case 7
disp('vendredi')
end
```