

Université Mohamed Boudiaf, M'sila
Faculté de technologie
Socle commun

Module : TP-informatique 3
2^{ième} année ST

TP5 – Les Structures et les Boucles sous MATLAB

1. Rappels sur les opérateurs logiques

Comme rappels voici la liste des opérateurs logiques servant entre autres à écrire les conditions :

==	égal à
~=	différent de
<	inférieur à
<=	inférieur ou égal à
>	supérieur à
>=	supérieur ou égal à
&	et
	ou
~	non

Exemple : Dans Matlab commande on fait les opérations suivante, conclure ?

```
>> a=[1 2 ;1 0] , b=[2 1 ; 1 2]
>> a == b
>> a > b
>> a & b
>> a | b
```

2. Structures conditionnelles

Il existe deux structures possibles permettant de réaliser des tests sur les données.

- La structure **if** permet de tester la valeur d'une variable et d'effectuer différents traitements suivant les cas testés.
- La structure **switch** permet de choisir entre différents cas, et de faire correspondre un traitement adapté à chacun des cas reconnus

a) Structure **if ... else ... end**

La structure **if** est indispensable à la programmation. Elle est présente dans tous les langages, mais utilise des syntaxes un peu différentes.

En langage Matlab, elle est déclarée de la manière suivante :

```
If condition 1
    Commandes
Elseif condition 2
    Commandes
Elseif condition 3
    Commandes
```

```

:
else
    Commandes
end

```

Seul le **if ... end** est obligatoire. Mais **Else** et **elseif** sont facultatifs et permettent d'effectuer des tests supplémentaires.

Exemple1 : Le script suivant vous donne un exemple d'utilisation de la fonction if sous Matlab. Tapez ce script dans un fichier "encadre.m" et testez les différents cas possibles.

```

% Exemple d'utilisation de if, elseif, ... end
nb = input('Rentrez un nombre : ');
if (nb < 100)
    disp('votre nombre est < 100')
elseif (nb == 100)
    disp('Votre nombre est 100')
else
    disp('votre nombre est > 100') ;
end

```

- Modifiez ensuite ce script pour qu'il indique en plus si le nombre est compris entre 100 et 200.

Exemple2 : Essayez et commentez l'exemple ci-dessous

```

>> a = zeros(1,10) ;
>> if a == 0, a(5) = 1, end
???.
>> if a == 0, a(5) = 0, end
???.

```

b) Structure Switch ... case ... end

La deuxième structure permet le choix entre différents cas. Le seul test effectué dans cette structure est donc un test d'égalité.

Cette structure est déclarée par :

```

Switch variable_testée
Case choix 1 (valeur prise par variable-testée)
    Commandes
Case choix 2
    Commandes
:
otherwise
    Commandes
end

```

Otherwise : est une alternative lorsque la variable testée n'a pas d'égaux parmi les différents cas.

Exemple1 : Le script suivant vous donne un exemple d'utilisation de la fonction **switch** sous Matlab.

Tapez ce script dans le fichier "semaine.m" et testez les différents cas possibles.

```

% Un exemple d'utilisation de switch
jour = input('Quel jour sommes nous ? ', 's ');
switch jour
case ' lundi '
    jr = 5;
case ' mardi '

```

```
jr = 4;
case ' mercredi '
jr = 3;
case ' jeudi '
jr = 2;
case ' vendredi '
jr = 1;
otherwise
jr = 0;
end
disp ([ 'Encore ' num2str(jr) ' jours avant le weekend' ])
```

Exemple2 : Modifiez ce script pour qu'il indique "c'est le week end" si le jour est samedi ou dimanche et qu'il indique "ce n'est pas un jour de la semaine" si la chaîne rentrée par l'utilisateur ne correspond pas à un jour de la semaine.

3. Structures itératives

Il existe deux grands types de boucles : les boucles de type **for** dont on connaît le nombre d'itérations et les boucles de type **while** dont le nombre d'itérations peut évoluer à l'intérieur de la boucle.

a) Structure for ... end

La structure **for** se présente comme suit :

```
For vecteur des valeurs
  Commandes
End
```

Cette structure est à éviter autant que possible sous Matlab car elle est très coûteuse en temps de calcul par rapport au calcul matriciel.

Dans de nombreux cas, il est possible de faire des traitements en blocs, par des opérations sur les matrices, et il n'est pas nécessaire d'utiliser les boucles for.

Exemple1. Définissez la fonction fibo.m qui prend un argument n et renvoie la valeur du n^{ième} terme de la suite de Fibonacci vue au cours (La suite de Fibonacci u_n est décrite par l'équation récurrente suivante : $u_{n+2} = u_{n+1} + u_n$ avec $u_0=0$ et $u_1=1$).

Réponse :

```
function y = fibo(n)
fib = [1, 2];
for i = 3:n
fib(i) = fib(i-1) + fib(i-2);
end
y = fib(n);
```

Question : Dans Matlab commande calculer Fibonacci par exemple pour n=10 ?

Exemple2. Si vous êtes motivé, et que vous avez bien compris le fonctionnement des deux boucles imbriquées, essayez d'écrire un script qui calcule le produit matriciel de deux matrices carrées A et B de dimension ($n \times m$).

Réponse :

```
for i = 1:n
  for j = 1:n
    x = 0
```

```

    for k = 1:n
        x = x + A(i,k)*B(k,j);
    end
    C(i,j) = x
end
end

```

Exemple2. Voici un exemple d'utilisation de la boucle **for** pour calculer **n!**

```

>> n = 4;
>> nfac = 1;
>> for k = 1:n
        nfac = nfac*k;
    end
>> nfac
nfac =
    24
>>

```

Question : Créer un script (M-file) pour l'exemple précédent ?

b) Structure **while ... end**

La structure **while** se présente comme suit :

```

While  test sur une variable
      Commandes
End

```

La boucle continue donc *tant que* le test renvoie une valeur de 1. Ce test est aussi appelé condition d'arrêt.

La variable de test doit donc être actualisée pendant l'exécution de la boucle afin que celle-ci s'arrête. Elle doit aussi être initialisée avant la première exécution de sorte que le test renvoie 1, et que la boucle soit effectuée au moins une fois.

Il peut aussi arriver volontairement dans le déroulement d'un programme qu'une boucle **while** ne soit jamais exécutée.

Exemple1. Dans les programmes précédents, on ne vérifie pas si le nombre passé en argument était bien un entier. On va donc redemander d'entrer un nombre tant que celui-ci ne sera pas entier.

```

n = input('Entrez un entier: ');
while n-floor(n) ~= 0
    n = input('On a dit un entier: ');
end

```

Exemple2. Voici comment calculer **n!** avec une boucle **while** :

```

>> n = 4;
>> k = 1; nfac = 1;
>> while k <= n
        nfac = nfac*k;
        k = k+1;
    end
>> nfac
nfac =
    24
>>

```

Question : Créer un script (M-file) pour l'exemple précédent ?