

Outils de Programmation 2

Chapitre 4 : Éléments de programmation Scilab

1 Les scripts

- L'environnement Scilab contient un éditeur de texte intégré, cet éditeur peut être lancé par l'instruction :
`-->scinotes`
directement depuis la console
- Un **programme Scilab** est une suite d'instructions, ces instructions peuvent être placées dans un fichier se terminant par « `.sce` » appelé **script**.
- Dans l'éditeur du texte, l'apparition des parenthèses ouvrantes et fermantes et des commandes de fin boucle, de fonction et test est automatique.
- On peut ce pendant désactiver ces deux fonctionnalités dans le menu option → compilation automatique en cliquant sur les deux entrées activées par défaut.
- On peut exécuter un script dans la console avec l'instruction
`exec('mon_fichier.sce')`
- Pour ajouter des commentaires à un script, on utilise le symbole « `//` » par exemple
`-->A=B+C; //donner à A la valeur de B+C`

1.1 Conseils de mise en forme des scripts

- Écrire une seule instruction par ligne.
- Une instruction trop longue sur une seule ligne peut se continuer sur une nouvelle ligne, il suffit de taper ... à la fin d'une ligne pour indiquer que l'instruction se poursuit sur la ligne suivante.
- Donner à votre script un nom significatif pour repérer rapidement vos programmes.
- Mettez des commentaires aux passages compliqués pour vous y retourner par la suite, si vous devez reprendre votre script lors d'une autre séance.
- Sauter des lignes blanches dans les boucles, structures conditionnelles, ect , surtout lorsqu'elles s'imbriquent afin de bien faire ressortir la structure de votre programme.

2 Structures de contrôle

Les structures de contrôles sont des instructions permettant de définir et manipuler l'ordre d'exécution des tâches dans un programme. Elles offrent la possibilité de réaliser des traitements différents selon l'état des données du programme ou de réaliser des boucles répétitives pour un processus donné.

2.1 L'instruction if

- L'instruction `if` est la plus simple et la plus utilisée dans les structures de contrôle, lorsqu'une certaine condition est vérifiée et une autre dans le cas contraire. Elle permet d'orienter l'exécution du programme en fonction de la valeur logique de la condition.
- La syntaxe de l'instruction conditionnelle `if` est la suivante :

```
if condition then
    instructions
end
ou
if condition then
    instructions1
else
    instructions2
end
```

Les instructions ne sont exécutées que si l'expression logique "`condition`" est vraie.

Exemple.

```
a=0.99;
if a>1 then
    c=log(a-1)
end
-->c
!--error 4
Variable non définie : c
-->a=3.5;
-->c
c =
    0.9162907
```

Exemple. Utilisation de `if`, `then` et `else` : les "`instructions1`" sont exécutées si l'expression logique "`condition`" est vraie, sinon les conditions "`instructions2`" sont exécutées.

```
a=0.9;
if a>0 then
    if a>1 then
        f=1;
    else
        f=a*a;
    end
else
    f=0;
end
-->f
f =
    0.81
```

- Quand else est suivi d'un if, on peut utiliser la forme

```

if condition1 then
    instructions
elseif condition2 then
    instructions
...
else
    instructions
end

```

2.2 Boucle for

- L'instruction for est la façon la plus courante d'effectuer des boucles, cette instruction répète l'exécution d'un groupe d'instructions un nombre déterminé de fois.

- La syntaxe d'une boucle for est la suivante :

```

for var = debut:pas:fin
    instructions
end

```

- La variable var prend la valeur debut, puis toutes les valeurs de la forme debut+k*pas sans dépasser la valeur fin.

Exemple.

```

for i=1:5:20
    i*i
end
ans =
    1.
ans =
    36.
ans =
    121.
ans =
    256.

```

- En fait debut:pas:fin désigne un vecteur et l'instruction var = debut:pas:fin peut être remplacée par var = v où v désigne un vecteur, la variable var décrit alors les composante du vecteur v.

Exemple.

```

v=[1 3 7 8];
for i=v
    i/2
end
ans =
    0.5
ans =
    1.5
ans =
    3.5
ans =
    4.

```

- On peut utiliser la boucle `for` pour définir une matrice.

Exemple.

```
n=4, a=zeros(n,n);
for i=1:n-1
    a(i,i)=2; a(i,i+1)=-1;a(i+1,i)=-1;
end
a(n,n)=2;
-->a
a =
    2.  -1.   0.   0.
   -1.   2.  -1.   0.
    0.  -1.   2.  -1.
    0.   0.  -1.   2.
```

- On peut sortir d'une boucle `for` par la commande `break`.

Exemple.

```
for i=0:0.01:1
    test=i-exp(-i);
    if (test>0) then
        i
        break
    end
end
i =
    0.57
```

2.3 Boucle while

- L'instruction `while` répète l'exécution d'un groupe d'instructions un nombre indéterminé de fois selon la valeur d'une condition logique.
- La syntaxe d'une boucle `while` est la suivante :

```
while expression do
    instructions
end
```

Les instructions sont exécutées tant que le booléen "expression" est vrai.

Exemple. Le coefficient binomial $C_n^k = \frac{n}{k} C_{n-1}^{k-1}$, pour avoir C_{13}^7 on peut procéder ainsi :

// on calcule 7 parmi 13

```
C=1,n=13,k=7;
while (n>0)&(k>0) then
    C=C*n/k
    n=n-1;
    k=k-1;
end
C
```