

## TP 3 : Boucle for et tracé de fonction

**EXERCICE 1.** On considère la suite  $(u_n)_{n \geq 0}$  définie par :

$$u_0 = 1 \text{ et } \forall n \in \mathbb{N}, u_{n+1} = \sqrt{u_n} + 1$$

Lequel des programmes suivant permet de calculer  $u_n$  en fonction de l'entier  $n$  entré par l'utilisateur ? Attention seul un des programmes convient. Pour chaque mauvais programme, indiquez l'erreur commise. Que vaut  $u_{100}$  ?

```
n=input("entrer un entier n")
U=1
for i=1 : n
    U=sqrt(U)+1
end
disp(U)
```

```
n=input"entrer un entier n"
U=1
for i=1 : n
    U=sqrt(U)+1
end
disp(U)
```

```
n=input("entrer un entier n")
U=1
for i=1 : n
    U=sqrt(U+1)
end
disp(U)
```

Le programme recherché est le premier. Le deuxième renvoie un message d'erreur car il manque les parenthèses dans input, le troisième calcule des valeurs de la suite :

$$u_0 = 1 \text{ et } \forall n \in \mathbb{N}, u_{n+1} = \sqrt{u_n + 1}$$

On trouve alors  $u_{100} = 2.618034$

**EXERCICE 2.** On considère le programme suivant :

```
clf
x=[0 :3]
y=x.^2+1
plot2d(x,y)
```

À quoi sert la commande `clf` ?

Cette commande permet d'effacer tous les graphes qui ont été tracés précédemment et de pouvoir ensuite tracer un graphe sur un repère vierge. Il vaut mieux prendre l'habitude de mettre cette commande en premier avant de tracer un graphe.

Que sont  $x$  et  $y$  ? Que valent ils ?

$x$  vaut  $\{0; 1; 2; 3\}$  et  $y$  vaut  $\{1; 2; 5; 10\}$

Qu'est ce que Scilab vous affiche en exécutant ce programme ? (faire un dessin)

Je vous laisse taper ça dans Scilab, on obtient un 'graphe' de la fonction obtenu avec 4 points que l'on relie entre eux.

Que se passe-t-il, si au lieu de `x=[0 :3]` on prend `x=[0 :0.1 :3]` ? Et `x=[0 :0.01 :3]` ?

Le graphe obtenue se lisse de plus en plus et devient quasiment le graphe de la fonction  $x \mapsto x^2 + 1$  sur  $[0; 3]$ .

**EXERCICE 3.**

Taper `x=[1 :0.5 :3]`.

Quelle commande doit on écrire pour affecter à la variable  $y$  le vecteur  $(1e^1; 1.5e^{1.5}; 2e^2; 2.5e^{2.5}; 3e^3)$  ?

```
y=x.*exp(x)
```

**EXERCICE 4.**

Écrire un programme permettant de tracer avec Scilab le graphe de la fonction :  $x \mapsto \exp(x)$  sur l'intervalle  $[-10; 10]$ .

```
clf
x=[-10 :0.1 :10]
y=exp(x)
plot2d(x,y)
```

**EXERCICE 5.**

Écrire deux programmes qui demandent à l'utilisateur d'entrer un entier  $n$ , qui calculent et enfin affichent  $\sum_{k=1}^n e^k$ .

L'un en utilisant une boucle for :

```
n=input("Entrer un entier n")
U=exp(1)
for i=2 : n
    U=U+exp(k)
end
disp(U)
```

L'autre en utilisant la fonction sum :

```
n=input("Entrer une valeur pour n")
sum(exp([1 :n]))
```

**EXERCICE 6.** Tracer avec l'aide de Scilab les graphes de fonctions suivantes :

$x \mapsto \sqrt{x-1}$  sur  $[1; 5]$

$x \mapsto x^3 - 2$  sur  $[-2; 3]$

$x \mapsto \frac{1}{6}x(x+1)$  sur  $[-100; 100]$

(difficile)  $x \mapsto \frac{1}{x^2}$  sur  $[-1; 1] \setminus \{0\}$

```
clf
x=[1 :0.1 :5]
y=sqrt(x-1)
plot2d(x,y)
```

```
clf
x=[-2 :0.1 :3]
y=x.^3-2*ones(x)
plot2d(x,y)
```

```
clf
x=[-100 :100]
y=x.*(x+ones(x))./6
plot2d(x,y)
```

```
clf
x=[-1 :0.01 :-0.01]
y=1./x
plot2d(x,y)
x=[0.01 :0.01 :1]
y=1./x
plot2d(x,y)
```