

Chapitre 4

LES BOUCLES

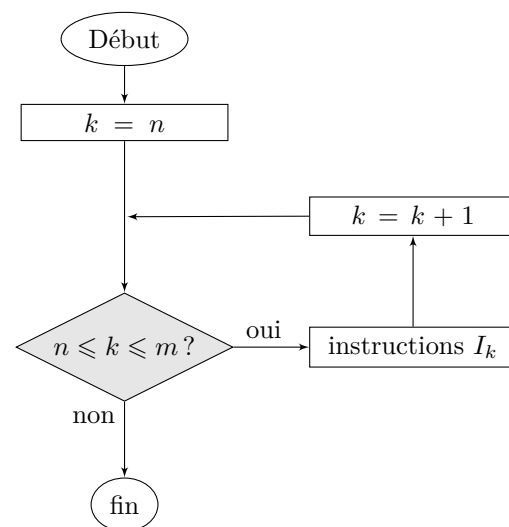
Dans ce chapitre, nous allons voir comment écrire un programme comportant une boucle.

I Les boucles for

I.1 Syntaxe

On souhaite traduire un algorithme de ce type :

Pour k variant de n à m ,
Réaliser les instructions I_k
Fin



La syntaxe Scilab pour écrire ce type d'instruction est la suivante :

```

for k = n : m
  instructions Ik
end
  
```

Remarque.

- Le nom de la variable de boucle (ici k) peut bien sûr être changé.
- L'initialisation $k = n$ et l'incrémement $k = k + 1$ sont faites automatiquement, il n'y a pas à les écrire explicitement.

Exemple. Afficher $1^2, 2^2, 3^2, \dots, 100^2$.

Programme Scilab

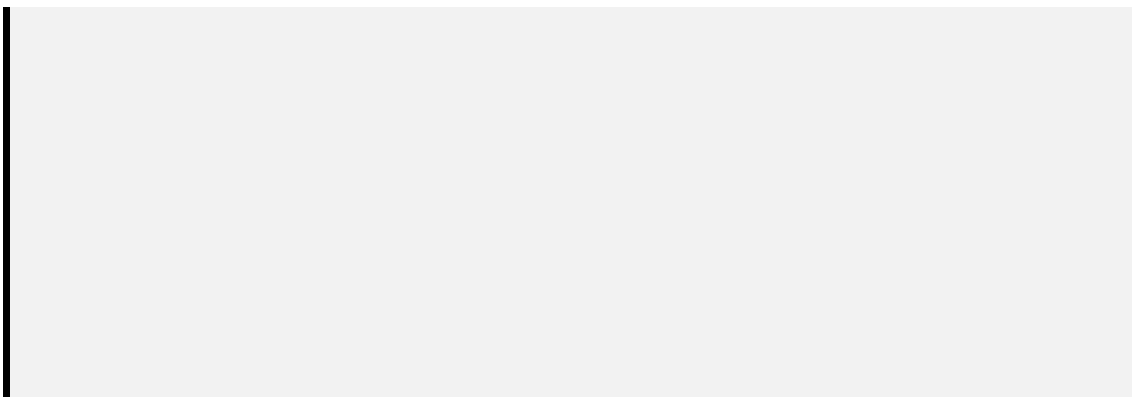
I.2 Exemple classique : calculer le terme de rang n d'une suite récurrente

Soit $(u_n)_{n \in \mathbb{N}}$ la suite définie par
$$\begin{cases} u_0 = 0 \\ \forall n \in \mathbb{N}, u_{n+1} = \sqrt{u_n + 2n} \end{cases}$$

1. Compléter le tableau suivant donnant les calculs à faire pour obtenir u_3 .

| Étape | Calcul à faire | Instruction Scilab |
|-------|----------------|--------------------|
| | | |
| | | |
| | | |
| | | |
| | | |

2. Écrire un programme Scilab demandant à l'utilisateur de saisir un entier n puis calculant et affichant u_n .



I.3 Exemple classique : calculer une somme ou un produit

Exemple 1 : Pour $n \geq 2$, on définit : $S_n = \sum_{i=2}^n i^4$.

1. Compléter le tableau suivant donnant les calculs à faire pour obtenir S_3 .

| Étape | Calcul à faire | Instruction Scilab |
|-------|----------------|--------------------|
| | | |
| | | |
| | | |
| | | |

2. Écrire un programme Scilab demandant à l'utilisateur de saisir un entier n puis calculant et affichant S_n .

Exemple 2 : Pour $n \in \mathbb{N}^*$, on définit : $P_n = n! = \prod_{k=1}^n k$. Écrire un programme Scilab demandant à l'utilisateur de saisir un entier n puis calculant et affichant P_n .

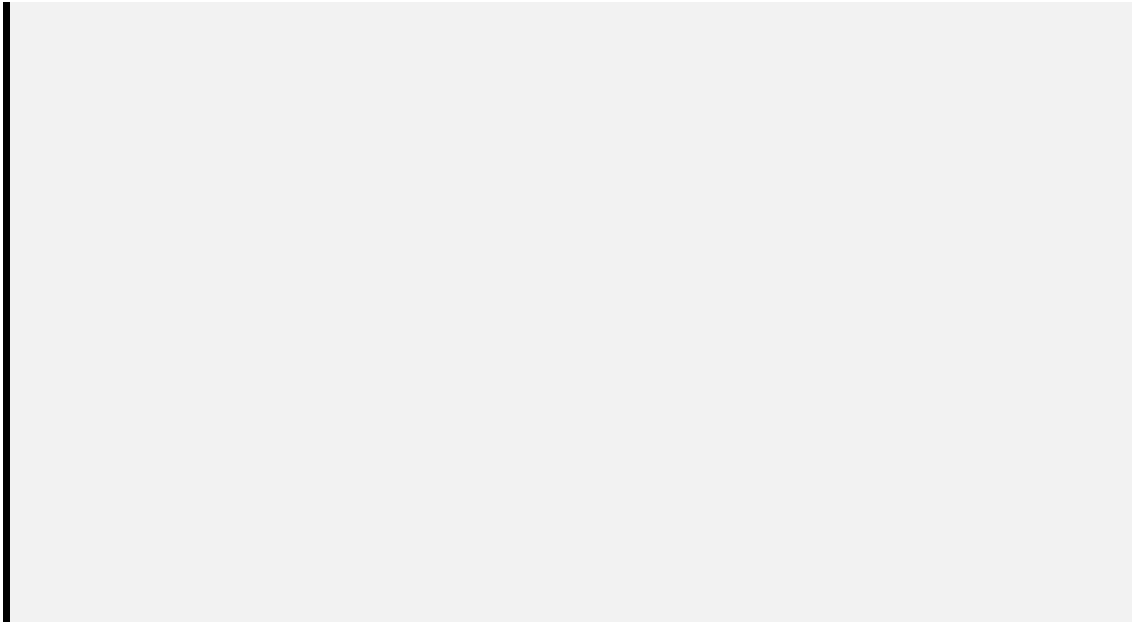
I.4 Exemple classique : les compteurs

Demander à Scilab de choisir 50 nombres aléatoires de l'intervalle $[0, 10]$ et déterminer combien sont inférieurs ou égaux à 4.

La commande `rand()`

La commande `rand()` permet d'obtenir un nombre aléatoire de l'intervalle $[0, 1]$.

Ainsi, la commande `10 * rand()` permet d'obtenir un nombre aléatoire de l'intervalle $[0, 10]$.

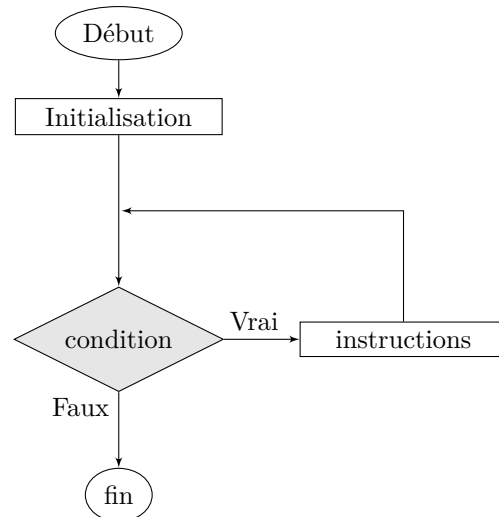


II Les boucles while

II.1 Syntaxe

On souhaite traduire un algorithme de ce type :

Initialisation
Tant que la condition est vraie,
 Réaliser les instructions
Fin



La syntaxe Scilab pour écrire ce type d'instruction est la suivante :

```

initialisation
while condition
    instructions
end
  
```

Remarque.

- La *condition* est un booléen (comme pour les if).
- Avec les boucles while, il y a un risque de créer une *boucle infinie*, ce qui bloque Scilab. Pour éviter cela, on respectera les règles suivantes.

La condition doit :

être initialisée : on doit pouvoir dire dès le départ si elle est vraie ou fausse. Les variables de la condition doivent donc être apparues avant le while.

évoluer : au moins une des variables de la condition doit changer de valeur à chaque tour de boucle.

devenir fausse à un moment.

Exemple. Réécrire le programme suivant (qui utilise une boucle for) avec une boucle while.

```

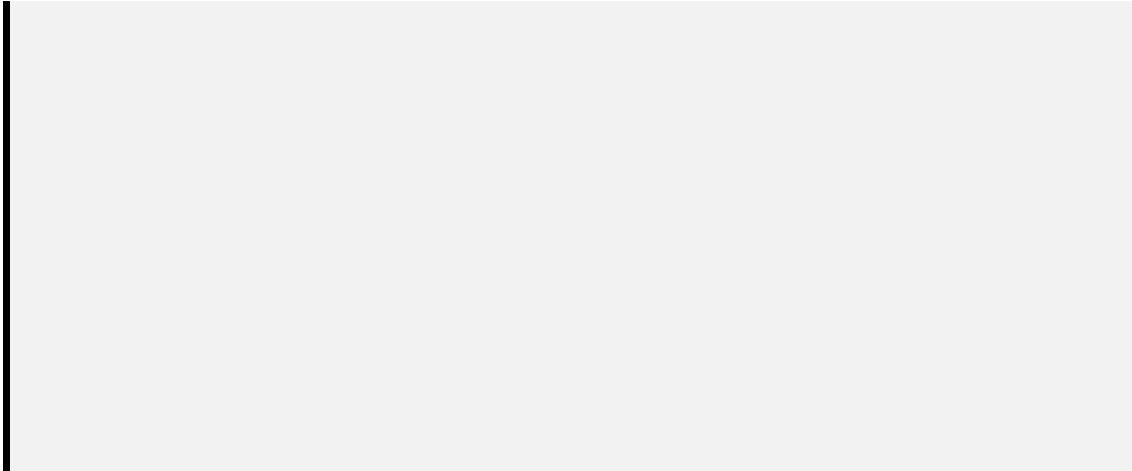
for k = 1 : 10
    disp( k^2 )
end
  
```

II.2 Exemple classique : "déterminer le plus petit n tel que ..."

On considère la suite $(u_n)_{n \in \mathbb{N}}$ définie par
$$\begin{cases} u_0 = 0 \\ \forall n \in \mathbb{N}, u_{n+1} = u_n - 3n \end{cases}$$

On peut montrer que $(u_n)_{n \in \mathbb{N}}$ est décroissante et tend vers $-\infty$ (bon exercice).

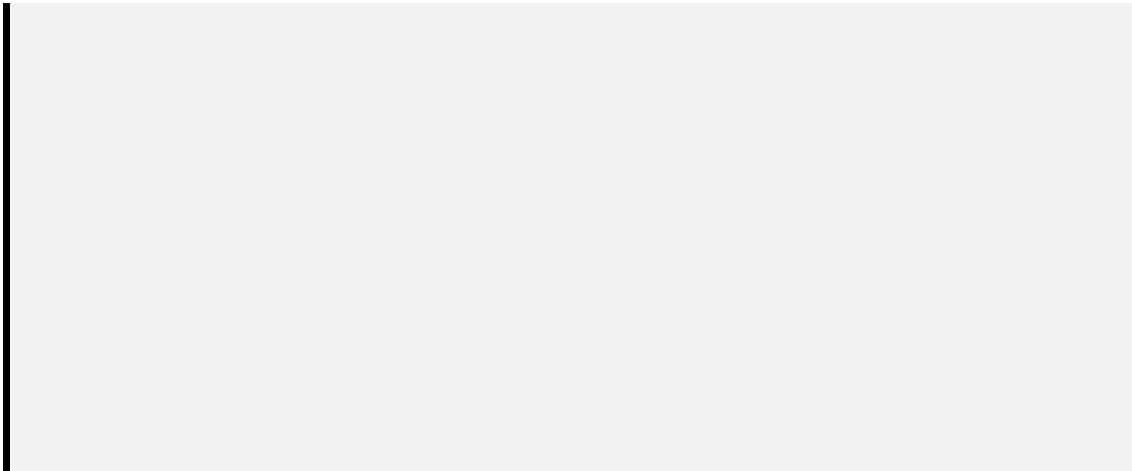
Déterminer le plus petit entier $n \in \mathbb{N}$ tel que $u_n \leq -10000$



| u | n | Condition |
|-----|-----|-----------|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

II.3 Exemple classique : "déterminer le plus grand n tel que ..."

Déterminer le plus grand $n \in \mathbb{N}$ tel que $2^n \leq 5000$.



| n | Condition |
|-----|-----------|
| | |
| | |
| | |
| | |
| | |