Fonction et dichotomie

Renaud Costadoat
Lycée Dorian

Exemple de fonction

Une *fonction* est un algorithme qui prend des arguments en entrée, effectue une séquence d'instructions et renvoie un résultat. Elle est considérée comme un objet de type `function`.

Un premier exemple consiste à programmer la fonction qui retourne la norme d'un vecteur à 3 coordonnées $\overrightarrow{V} = a.\overrightarrow{x} + b.\overrightarrow{y} + c.\overrightarrow{z}$ $\sqrt{a^2 + b^2 + c^2}$.

```
def norme_3(a,b,c):
    n = (a**2 + b**2 + c**2)**(1/2)
    return n
```

Attention: Il faut toujours **appeler la fonction** ensuite afin de voir le résultat obtenu.

```
>>> norme_3(3,4,5)
7.0710678118645755
>>> norme_3(4,5,6)
8.774964387392123
```

Forme générale

```
def <nom_de_la_fonction>(<argument_1>,<argument_2>,...,<argument_n>):
    "Ce commentaire explique à quoi sert cette fonction"
    <instructions>
```

On n'oubliera pas les **:** et l'*indentation* qui sont obligatoires en Python !

Remarques:

- les arguments entre parenthèses sont des paramètres formels,
- Les instructions qui forment le corps de la fonction commencent sur la ligne suivante, indentée par quatre espaces (ou une tabulation),
- La première instruction du corps de la fonction peut être un texte dans une chaîne de caractères, cette chaîne est la chaîne de documentation de la fonction et on peut la visualiser dans un terminal en tapant l'instruction `help(nom_de_la_fonction)` ,
- Le retour à la ligne signale la fin de la fonction.

Dès que l'instruction return est exécutée (si elle est présente), l'exécution de **la fonction se termine**.

Erreur de programmation

La fonction suivante est sensée retourner la somme des nombres pairs inférieurs à un nombre donné `a` .

```
def somme_pairs(a) :
    i=1
    n=0
    while(i<=a):
        if i%2==0:
            n=n+i
        i=i+1
        return n
```

```
>>> somme_pairs(5)
0
>>> somme_pairs(3)
0
```

Question 1: Expliquer pourquoi ce programme ne donne pas le résultat attendu.

Question 2: Proposer une modification afin que ce soit le cas.

Dichotomie

Voici le processus complet :

- Au rang 0,
  - soient $a_0 = a$, $b_0 = b$. Il existe une solution x_0 de l'équation $(f(x) = 0)$ dans l'intervalle $[a_0, b_0]$.
- Au rang 1,
  - si $f(a_0).f(\frac{a_0+b_0}{2}) \leq 0$, alors on pose $a_1 = a_0$, $b_1 = \frac{a_0+b_0}{2}$,
  - sinon on pose $a_1 = \frac{a_0+b_0}{2}$ et $b_1 = b$.
  - dans les deux cas, il existe une solution x_1 de l'équation $(f(x) = 0)$ dans l'intervalle $[a_1, b_1]$.
- Au rang n, supposons construit un intervalle $[a_n, b_n]$, de longueur $\frac{b-a}{2^n}$, et contenant une solution x_n de l'équation $(f(x) = 0)$. Alors:
  - si $f(a_n).f(\frac{a_n+b_n}{2}) \leq 0$, alors on pose $a_{n+1} = a_n$, $b_{n+1} = \frac{a_n+b_n}{2}$,
  - sinon on pose $a_{n+1} = \frac{a_n+b_n}{2}$ et $b_{n+1} = b_n$,
  - dans les deux cas, il existe une solution x_{n+1} de l'équation $(f(x) = 0)$ dans l'intervalle $[a_{n+1}, b_{n+1}]$.

A chaque étape, on a $a_n \leq x_n \leq b_n$, on arrête le processus dès que $b_n - a_n = \frac{b-a}{2^n}$ est inférieur à la précision souhaitée.

Exemple dichotomie

Question 1: Coder la fonction `f(x)` qui à la valeur x revoie la valeur $f(x) = x^3 + 2.x^2 + 3.x - 4$.

Question 2: Coder la fonction `g(x)` qui à la valeur x revoie la valeur $g(x) = x^3 + 2.x^2 + 3.x - 6$.

Question 3: Coder la fonction `dichotomie(f,p)` qui à la fonction f revoie la valeur de la racine du polynôme avec une précision minimale p. La fonction devra retourner la valeur de a_n ainsi que l'erreur finale $b_n - a_n$.

Précision:

- Il n'existe qu'une seule racine pour les fonctions $f(x)$ et $g(x)$ et elles sont comprises entre -10 et 10.