



Partie A : Tracer des courbes avec Python

Syntaxe avec Python



- ✓ **pylab** : Module à importer qui permet de tracer des courbes.
- ✓ **plot(x, y, 'r')** : affiche les points de coordonnées (x, y) en rouge.
- ✓ **plot(x, y, 'g')** : affiche les points de coordonnées (x, y) en vert.
- ✓ **Show()** : affiche la figure à l'écran.
- ✓ **Boucle non bornée** : dans une boucle, le traitement est répété tant que la condition est vraie. Lorsque la condition est fausse, on sort de la boucle et la suite des instructions s'applique.
- ✓ En Python, la fin de l'indentation marque la fin de la boucle.

Algorithme :

Tant que **condition**
 Traitement
Fin Tant que

Python :

While **condition** :
 ...
...

Voici un programme écrit dans un langage Python.

```
from pylab import*  
  
def h(x) :  
    return -4 * x ** 2 + 4 * x + 3  
  
x = -3  
while x <= 3 :  
    plot (x, h(x), 'g.')  
    x = x + 0.01  
  
show ()
```

- 1) Quelle est la fonction dont on veut afficher la courbe ?
 $h(x) = -4x^2 + 4x + 3$
- 2) Sur quel intervalle la courbe est-elle tracée ?
 h est définie sur $[-3 ; 3]$.
- 3) Quel est le pas entre chaque point tracé ?
Le pas est de 0,01.
- 4) Saisir ce programme à l'ordinateur et l'exécuter.
- 5) Quel est l'allure de la courbe C_h sur l'intervalle $[-4 ; 4]$?
La courbe est une parabole.

Attention : Python ne travaille pas avec la valeur exacte de 0,01 ce qui entraîne des erreurs d'arrondi. Ceci peut poser problème si on prend une fonction comme $\sqrt{3 - x}$.

Partie B : Résolution par le calcul

Un plongeur saute d'un plongoir. Sa hauteur, en mètres, repérée par rapport au niveau de l'eau, est exprimée en fonction du temps écoulé, en secondes, depuis le départ du saut par :

$$h(t) = 4 - (2t - 1)^2$$

1) Prouver que : $h(t) = -4t^2 + 4t + 3$.

$$h(t) = 4 - (2t - 1)^2 = 4 - (4t^2 - 4t + 1) = -4t^2 + 4t + 3$$

2) Montrer que : $h(t) = (3 - 2t)(1 + 2t)$.

$$(3 - 2t)(1 + 2t) = 3 + 6t - 2t - 4t^2 = -4t^2 + 4t + 3 = h(t)$$

3) Utiliser la forme la mieux adaptée pour répondre aux questions suivantes, justifier soigneusement chaque réponse en détaillant votre raisonnement :

a) À quelle hauteur se trouve le plongoir ? $h(0) = 3$ m de hauteur

b) Quelle est l'altitude maximale du plongeur ? $h(0,5) = 4$ m de hauteur

c) Au bout de combien de temps le plongeur arrive-t-il dans l'eau ?

$$h(1,5) = 0. \text{ Au bout de 1,5 secondes.}$$

Partie C : Résolution avec Python

Syntaxe avec Python



- ✓ **Boucle bornée** : permet de répéter plusieurs fois de suite un même traitement.
- ✓ On utilise un compteur initialisé à 1 et qui s'incrémente de 1 à chaque itération jusqu'à la valeur n .
- ✓ Pour construire une liste de nombre allant par exemple de 0 à 3 avec un pas de 0.1 vous pouvez passer par numpy et la fonction arange:

✓

Algorithme :

Pour i allant de 1 à n
 Traitement
Fin Pour

Python :

For i in range (1, $n+1$) :
 ...

Python :

From numpy import *
For i in arange (0, 3.1, 0.1) :
 ...

1) Modifier le programme de la partie A pour faire apparaître en rouge les solutions de l'inéquation $h(x) \geq 0$ sur l'intervalle $[0, 3]$.
Interpréter ce résultat.

$$h(x) \geq 0 \text{ sur } [0 ; 1,5].$$

```
from pylab import *
from numpy import *

def h(x) :
    y=-4*x**2+4*x+3
    return y

x=0
while x<=3 :
    y=h(x)
    if y>=0:
        plot (x, y, 'r.')
    else :
        plot (x, y, 'g.')
    x=x+0.01
show()
```

2) On veut modifier le programme de la partie A pour afficher les coordonnées du maximum de la fonction h sur l'intervalle $[0, 3]$.

- a) Compléter l'algorithme ci-contre pour qu'après exécution, la variable $yMax$ contienne le maximum de h et la variable $xMax$ contienne l'abscisse de ce maximum.

```
 $yMax \leftarrow 3$   
 $xMax \leftarrow 0$   
Pour  $i$  variant de 0 à 3 avec un pas de 0,1  
     $H \leftarrow -4i^2 + 4i + 3$   
    Si  $H > yMax$  alors  
         $yMax \leftarrow H$   
         $xMax \leftarrow i$   
    Fin Si  
Fin Pour
```

- b) Programmer une fonction, en langage Python, sans argument, qui retourne les variables($xMax, yMax$) .

```
def maximumh():  
    yMax=3; xMax=0  
    for i in arange (0, 3.1, 0.1) :  
        H=-4*i**2+4*i+3  
        print (i,H)  
        if H>yMax:  
            yMax=H; xMax=i  
    return (xMax, yMax)
```

- c) Préciser ses valeurs. Interpréter ce résultat.

Le maximum a pour coordonnées (0.5, 4.0), ce qui signifie que la hauteur maximale atteinte par le plongeur est de 4 m, atteint au bout de 0,5 secondes.