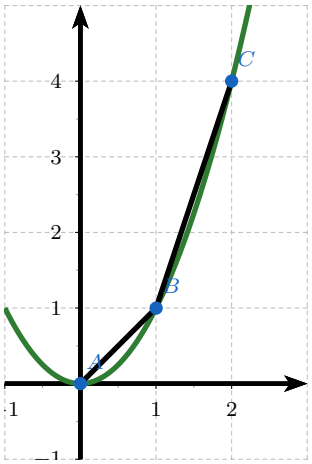
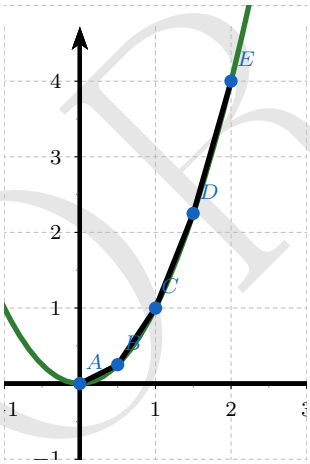
1 Donner la valeur approchée de la longueur d'une courbe sur un intervalle donné

Soit la fonction $f(x) = x^2$. Donner une valeur approchée de la longueur de la courbe de f sur [0 ;2]

1.1 L'idée mathématique

On va choisir des points de la courbe espacés régulièrement puis on va calculer et ajouter les longueurs des segments obtenus avec ces points . Plus les points sont nombreux et proches , et plus l'approximation est précise .





1.2 La mise en algorithme

Variables
x1 , y1 , x2 , y2 , L , a , b réels
i , p : entiers
Début de l'algorithme
Saisir p , a , b
$x1 \leftarrow a$
$x2 \leftarrow x1 + p$
$L \leftarrow 0$
Tant que $x1 < b$ **Faire**
  $y1 \leftarrow x1^2$
  $y2 \leftarrow x2^2$
  $L \leftarrow L + \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$
  $x1 \leftarrow x2$
  $x2 \leftarrow x1 + p$
FinTantque
Sorties :
Afficher L

```
from math import*
def longsegment(xA,yA,xB,yB):
    l=sqrt((xB-xA)**2+(yB-yA)**2)
    return l
def longcourbe(p,a,b):
    x1=a
    x2=a+p
    L=0
    while x1<b:
        y1=x1**2
        y2=x2**2
        L=L+longsegment(x1,y1,x2,y2)
        x1=x2
        x2=x1+p
    return L
```

Compléter le tableau ci-dessous en faisant tourner l'algorithme dans le cas où on l'applique avec $a = 0$, $b = 2$ et $p = 1$

	y1	y2	x1	x2	L	segment
Avant boucle						
passage 1						
passage 2						
Sortie						

2 Donner une valeur approchée d'un extremum par balayage

Déterminer une valeur approchée au centième du minimum de $f(x) = 3x^3-21x^2+22x+26$ sur [-1 ;5]

2.1 L'idée mathématique

f admet un minimum en a si pour tout x, $f(x) > f(a)$. On doit donc tester les images pour x dans [-1 ;5] et affecter à a les valeurs les plus petites .

Au départ , $a = -1$, on prend $x = a + 0.1 = -0.9$. On calcule $f(-1)$ et $f(-0.9)$. On constate que $f(-1) < f(-0.9)$ donc a reste égal à -1 .

Mais quand on arrive à 3.1 , a est toujours égal à -1 mais , $f(a) = -20$ et $f(3.2) = -20.33$ donc à cette étape , a devient 3.2 .

2.2 La mise en algorithme

Variables
a , x , p , m : réels
i : entier
Début de l'algorithme
Saisir a , b , n
$m \leftarrow a$
$x \leftarrow a$
Pour *i allant de 1 à n* **Faire**
  $p \leftarrow 10^{-i}$
  Tant que $x < b$ **Faire**
    $x \leftarrow x + p$
    Si $f(m) > f(x)$ **Alors**
      $m = x$
    Finsi
  FinTantque
FinPour
Sorties :
Afficher m

```
1  def f(x):
2      f=3*x**3-21*x**2+22*x+26
3      return f
4  def extremumbalayage(a,b,n):
5      m=a
6      x=a
7      for k in range (1,n+1):
8          p=10**(-k)
9          while x<b:
10             x=x+p
11             if  f(m)>f(x):
12                 m=x
13     return (m)
```

3 Donner une valeur approchée d'un extremum par dichotomie

Déterminer une valeur approchée au centième du minimum de $f(x) = 3x^3 - 21x^2 + 22x + 26$ sur [-1 ;5]

3.1 L'idée mathématique

Soit f une fonction décroissante sur [a ;M] puis croissante sur [M ;b] .

On appelle m le milieu de [a ;b] , c le milieu de [a ;m] et d le milieu de [m ;b] .

Le principe est de diviser par 2 l'intervalle de travail .

Si $f(c) < f(m)$, la fonction f est croissante et donc le minimum est entre a et m . On choisit donc de travailler sur [a ;m]

Si $f(d) < f(m)$, puisque $m < d$, la fonction est décroissante et donc le minimum est entre m et b .

Dans les autres cas ,le minimum est entre c et d .

3.2 La mise en algorithme

Variables
a , b , c , d , m : réels
k : entier
Début de l'algorithme
$a \leftarrow 2$
$b \leftarrow 5$
Pour *k allant de 1 à 10* **Faire**
  $m \leftarrow \dfrac{a+b}{2}$
  $c \leftarrow \dfrac{a+m}{2}$
  $d \leftarrow \dfrac{b+m}{2}$
  Si $f(c) < f(m)$ **Alors**
    $b \leftarrow m$
  Sinon
    Si $f(d) < f(m)$ **Alors**
      $a \leftarrow m$
    Sinon
      $a \leftarrow c$
      $b \leftarrow d$
    Finsi
  Finsi
FinPour
Sorties :
Afficher $\dfrac{a+b}{2}$

```
1  def f(x):
2      f=3*x**3-21*x**2+22*x+26
3      return f
4  def extremumdichotomie():
5      a=2
6      b=5
7      for k in range (1,11):
8          m=(a+b)/2
9          c=(a+m)/2
10         d=(b+m)/2
11         if  f(c)<f(m):
12             b=m
13         else:
14             if  f(d)<f(m):
15                 a=m
16             else:
17                 a=c
18                 b=d
19     return ((a+b)/2)
```