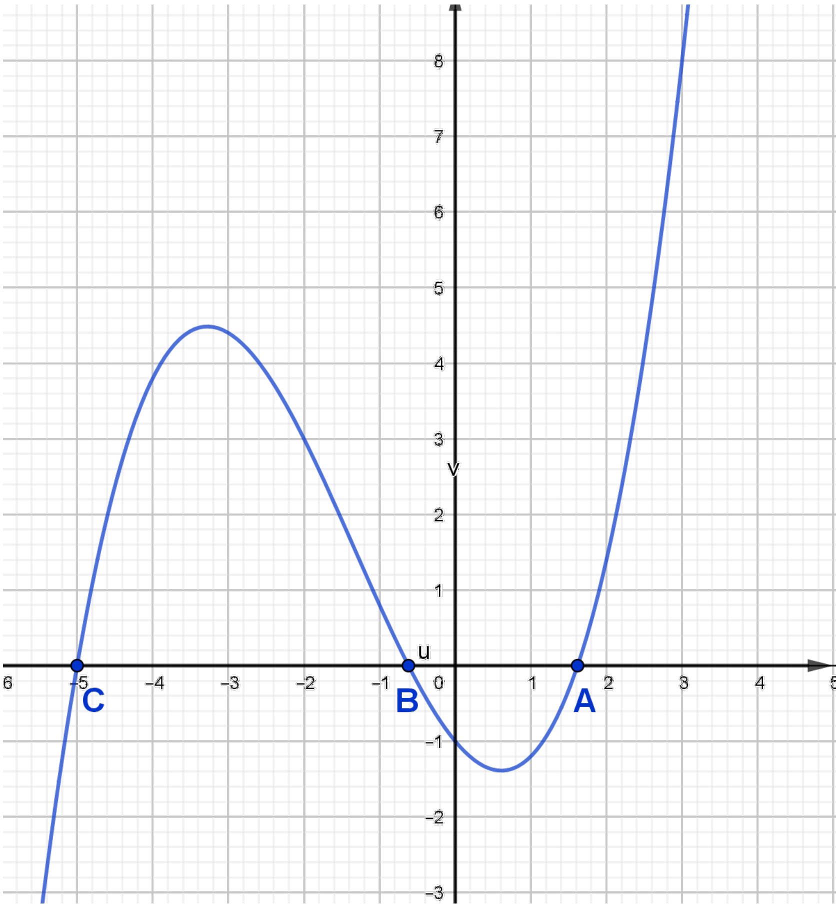
TP 3

1. Préambule

On se propose de donner un exemple de détermination d'une valeur approchée d'une solution d'une équation en utilisant la méthode de dichotomie.

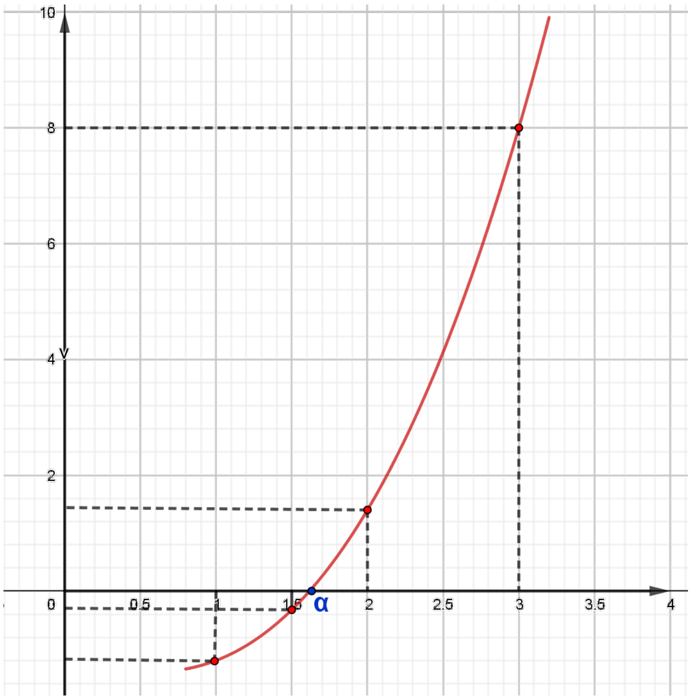
2. Énoncé

On considère la fonction f définie sur l'intervalle [-5,5;3,1] par $f(x)=0,2x^3+0,8x^2-1,2x-1$.
Soit $\mathscr{C}$ sa courbe représentative dans un repère orthonormé.



La courbe $\mathscr{C}$ coupe l'axe des abscisses en trois points : A, B et C.
Par lecture graphique, on détermine des valeurs approchées des abscisses de A, B et C, on obtient A(1,6), B(-0,6) et C(-5).
C'est à dire:1,6, -0,6 et -5 sont des valeurs approchées des solutions de l'équation :
$0,2x^3+0,8x^2-1,2x-1=0$.

On désire déterminer une valeur approchée à 0,001 de la solution α positive.
On représente la fonction f sur un intervalle contenant l'intervalle [1;3] dans un repère non orthonormé.



Par lecture graphique, $f(1)<0$ et $f(3)>0$ et $f(\alpha)=0$, on a : $1\leqslant\alpha\leqslant3$.
On considère le centre de l'intervalle [1;3]: 2 et on constate que $f(2)>0$ donc $1\leqslant\alpha\leqslant2$.
On considère le centre de l'intervalle [1;2] : 1,5 et on constate que $f(1{,}5)<0$ donc $1{,}5\leqslant\alpha\leqslant2$
On divise par 2 l'amplitude de l'intervalle contenant α à chaque opération, on peut donc obtenir des intervalles d'amplitude inférieure à 0,1 ou 0,01 ou 0,001 . . .

2.1. On propose l'algorithme suivant :

Variables : a, b et c sont des nombres réels
Initialisation : a=1
b=3
Traitement : Tant que $b-a\geqslant0{,}001$
$c=\dfrac{a+b}{2}$
Si $f(c)\geqslant0$
b=c
Sinon
a=c
Fin tant que
Sortie : Afficher a et b

. Programmer cet algorithme en Python et l'exécuter.

2.2. Donner un algorithme permettant de déterminer une valeur approchée à 0,001 près de la plus grande solution négative β de l'équation initialement proposée et le programmer en Python et l'exécuter.

3. Correction

3.1. Programme en Python

print('Début de programme')
a,b=1,3
while(b-a)>=0,001)
$c=\dfrac{a+b}{2}$ # centre de l'intervalle

```
    if  0,2*c**3+0,8*c**2-1,2*c-1>=0 : # condition f(c)>=0
        b=c
    else :
        a=c
print(« a= »+str(a), « b= »+str(b))
print('Fin de programme')
```

. <u>On obtient</u>

```
File   Edit   Format   Run   Options   Window   Help
print('Début de progrmme')
a,b=1,3
while(b-a>=0.001):
    c=(a+b)/2                          # centre de l'intervalle
    if 0.2*c**3+0.8*c**2-1.2*c-1>=0: # condition f'c)>=0
        b=c
    else :
        a=c
print("a="+str(a),"b="+str(b))
print('Fin de progamme')
```

. *<u>Remarques</u>*
*Pour noter c au cube on note c**3.*
On note aussi if pour Si et else pour Sinon

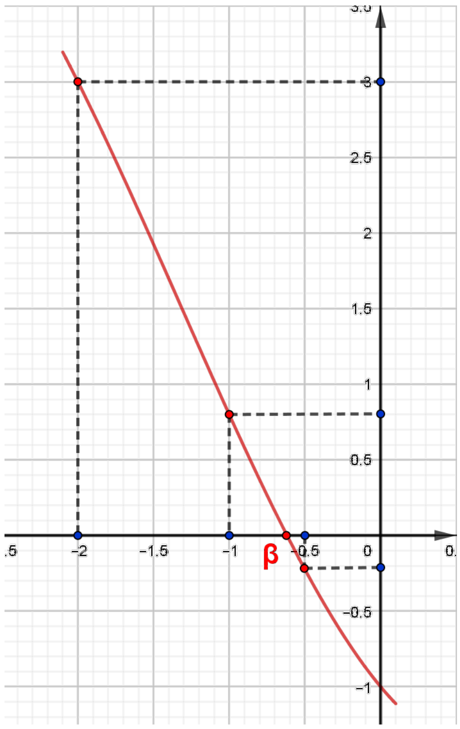
. <u>On exécute le programme et on obtient</u>

Début de programme
a=1,6171875   b=1,6181640625
Fin de programme

```
Python 3.6.4 Shell
File   Edit   Shell   Debug   Options   Window   Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)]
 on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\Serge\Documents\Programmes-Python\tp3-1.py ========
Début de progrmme
a=1.6171875 b=1.6181640625
Fin de progamme
```

. <u>Conclusion</u>
1,618 est une approchée de α **à 0,001 près.**

3.2. On représente la courbe représentative de f sur l'intervalle [-2,1;01].



La différence avec la question précédente est que la fonctionf est décroissante sur l'intervalle [-2;0] donc si $-2\leqslant x\leqslant\beta$ alors $f(x)\geqslant f(\beta)=0$ et si $\beta<x\leqslant 0$ alors $0=f(\beta)>f(x)$.

On constate que $f(-2)>0$ et $f(0)<0$ donc $-2\leqslant\beta\leqslant 0$

$f(-1)>0$ et $f(0)<0$ donc $-1\leqslant\beta\leqslant 0$

$f(-1)>0$ et $f(-0,5)<0$ donc $-1\leqslant\beta\leqslant -0,5$

- Algorithme

Variables : a, b et c sont des nombres réels

Initialisation : a=-2

b=0

Traitement : Tant que $a-b\geqslant 0{,}001$

$c=\dfrac{a+b}{2}$

Si $f(c)\geqslant 0$

a=c

Sinon

b=c

Fin de Tant que

Sortie : Afficher a et b

- Programme en Python

```
print('Début de programme')
a,b=-2,0
while(b-a>=0,001) :
   c=(a+b)/2                              # centre de l'intervalle
   if  0,2*c**3+0,8*c**2-1,2*c-1>=0 :   # condition f(c)>=0
      a=c
   else :
      b=c
print(« a= »+str(a), « b= »+str(b))
```


print('Fin de programme')

- <u>On obtient</u>

```
File  Edit  Format  Run  Options  Window  Help
print('Début de programme')
a,b=-2,0
while(b-a)>=0.001:
    c=(a+b)/2
    if 0.2*c**3+0.8*c**2-1.2*c-1>=0:
        a=c
    else:
        b=c
print("a="+str(a),"b="+str(b))
print('fin de programme')
```

- <u>On exécute le programme et on obtient</u>

Début de programme
a= -0,6181640625   b= -0,6171875
Fin de programme

```
Python 3.6.4 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)]
 on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\Serge\Documents\Programmes-Python\tp3-2.py ========
Début de programme
a=-0.6181640625 b=-0.6171875
fin de programme
>>>
```

- <u>Conclusion</u>
  -0,618 est une valeur approchée de β à 0,001 près.