

Exercice 1.

a) Tracé de la courbe :

```

from matplotlib.pyplot import * # Import du module

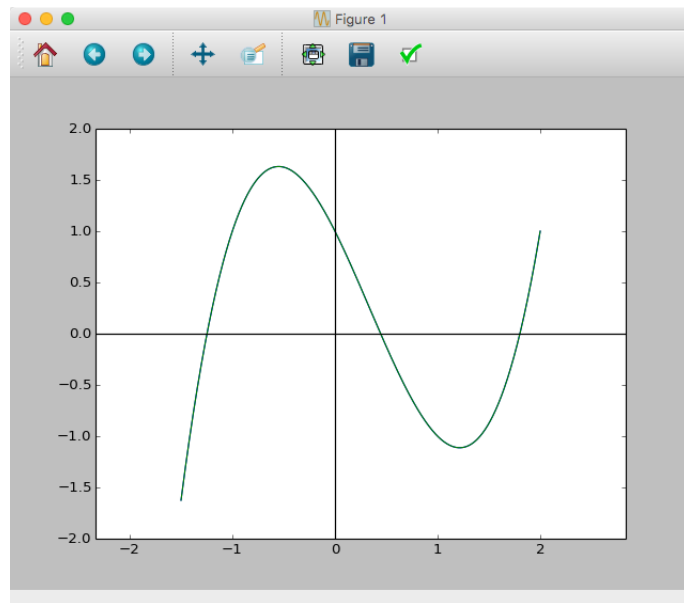
f = lambda x: x**3-x**2-2*x+1 # definition de f

n = 1000 # Nbre points
a = -1.5 # borne inf
b = 2 # borne sup
X = [a + k*(b-a)/(n-1) for k in range(n)] # abscisses
Y = [f(x) for x in X] # ordonnees

plot(X,Y) # tracé
axhline(color="black") # axe X
axvline(color="black") # axe Y
axis("equal") # repere orthonormé
show() # affichage

```

On obtient le graphique suivant :



b) On voit que le polynôme admet trois racines sur $[-1, 5; 2]$. Comme il est de degré 3 ce sont ses seules racines. On en détermine des encadrements à 10^{-3} près par dichotomie; il s'agira de bien choisir les 3 intervalles où appliquer la recherche par dichotomie. On prend : $[-1, 5; -1]$, $[0; 1]$ et $[1; 2]$ par exemple.

```

def dichotomie(f,a,b,e):

```

```

    while b-a > e:
        m = (a+b)/2
        if f(a)*f(m) <= 0:
            b = m
        else:
            a = m
    return [a,b]

f = lambda x: x**3-x**2-2*x+1 # definition de f
e = 1E-3 # tolerance
[a1,b1] = dichotomie(f,-1.5,-1,e)
[a2,b2] = dichotomie(f,0,1,e)
[a3,b3] = dichotomie(f,1,2,e)
print("Racines")
print(a1,"< r1 <",b1) # 1ere racine
print(a2,"< r2 <",b2) # 2eme racine
print(a3,"< r3 <",b3) # 3eme racine

```

On obtient pour résultats (affichés dans la console) :

```

Racines
-1.2470703125 < r1 < -1.24609375
0.4443359375 < r2 < 0.4453125
1.8017578125 < r3 < 1.802734375

```