EP_-_02

September 11, 2019

0.1 Eric Satoshi Suzuki Kishimoto

RA: 233974

```
In [15]: #importação de pacotes
         from scipy import integrate
         from math import sqrt
         from numpy import arange
         from matplotlib import pyplot
```

```
In [16]: # funcao para calcular a integral da função pelo método do trapézio
         # f: função a ser integrada
         # a: limite inferior da integral
         # b: limite superior da inetegral
         # n: números de intervalos que o graficos será dividido(qtd de trapézio a
         # serem calculados)
         def integral(f, x, a, b, n):
             h = (b - a)/n
             integral = 0
             # somatória de trapézios
             for i in arange(a+h, b, h):
                 integral += (h/2)*f(x, i)
             integral *= 2
             integral += (h/2)*f(x, a)
             integral += (h/2)*f(x, b)
             return integral
```

```
In [17]: x0 = 0.005
         yinf = 0.650
         # definição da função
         g = lambda x, y: y**2/sqrt(x**2 + y**2)
         f = lambda x: x - x0 - 3 * (integral(g, x, x0, yinf, 100))
```

```
In [18]: # método da bissecção
         # f: função passada
         # a: primeiro chute
         # b: segundo chute
         # e: taxa de erro
         def bisseccao(f, a, b, e):
```

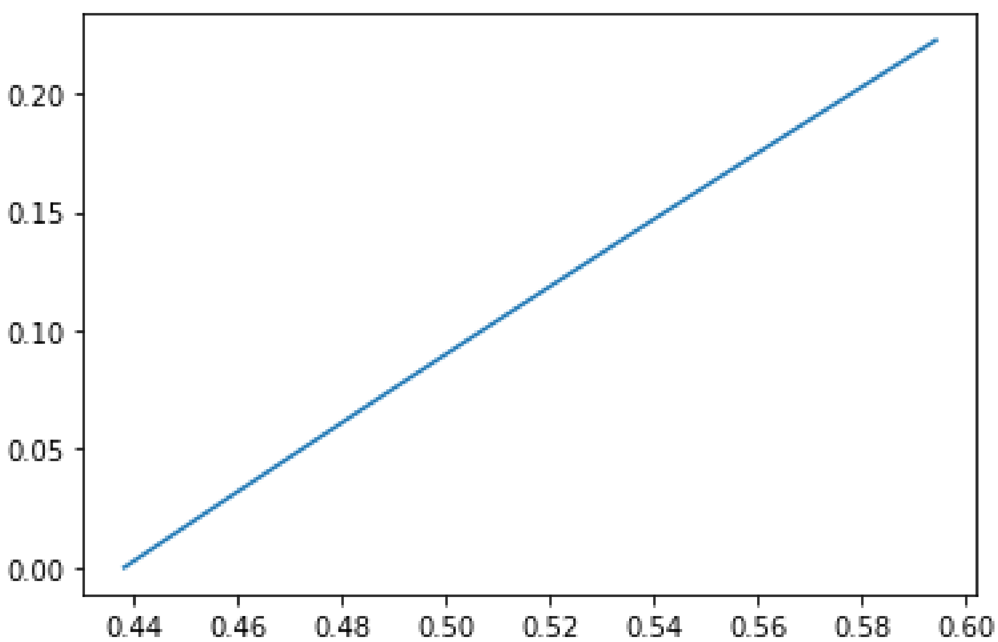
```
    isRoot = False
    while isRoot == False:
        ptoMedio = (a + b)/2
        isRoot = abs(f(ptoMedio)) <= e
        if f(a)*f(ptoMedio) < 0:
            b = ptoMedio
        else:
            a = ptoMedio
    return ptoMedio
```

In [19]:
```
bisseccao(f, 0.2, 0.6, 1*10**-8)
```

Out[19]: 0.4382679462432862

In [20]:
```
variavel = []
funcao = []
for j in arange(0.4382679462432862, 0.6, 0.0065):
    variavel.append(j)
    funcao.append(f(j))
pyplot.plot(variavel, funcao)
```

Out[20]: [<matplotlib.lines.Line2D at 0x1a164a62c18>]



Pelo método da bissecção, determinou-se que a raíz da função é igual a 0.4382679462432862