>>Metodo da Bissecção

import math

def calcular\_funcao(expressao, x):

# irá susbstituir o x pelo intervalo

expressao = expressao.replace('x', str(x))

# Eval tem como objetivo calcular a função

return eval(expressao)

def metodo\_da\_bisseccao(expressao, a, b):

if calcular\_funcao(expressao, a) \* calcular\_funcao(expressao, b) > 0:

print("Não tem numeros para onde essa função seja 0")

return None

else:

ponto\_medio = (a + b) / 2

while calcular\_funcao(expressao, ponto\_medio) > 10\*\*-5:

ponto\_medio = (a + b) / 2

if calcular\_funcao(expressao, ponto\_medio) \* calcular\_funcao(expressao, a) < 0:

b = ponto\_medio

else:

a = ponto\_medio

return ponto\_medio

def main():

expressao = input("Digite a expressao da funcao (por exemplo, x\*\*3+x\*\*2-x+2): ")

a = float(input("Digite o valor de a: "))

b = float(input("Digite o valor de b: "))

resultado = metodo\_da\_bisseccao(expressao, a, b)

if resultado is not None:

print("O resultado aproximado da raiz é:", resultado)

else:

print("Não foi possível encontrar a raiz dentro da precisão desejada.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

######OUTPUT########

Digite a expressao da funcao (por exemplo, x\*\*3+x\*\*2-x+2): x\*\*2-4

Digite o valor de a: 1

Digite o valor de b: 3

O resultado aproximado da raiz é: 2.0

Digite a expressao da funcao (por exemplo, x\*\*3+x\*\*2-x+2): x\*\*3-x-1

Digite o valor de a: 1

Digite o valor de b: 2

O resultado aproximado da raiz é: 1.25

Digite a expressao da funcao (por exemplo, x\*\*3+x\*\*2-x+2): x\*\*4+x\*\*3-25

Digite o valor de a: 1

Digite o valor de b: 55

O resultado aproximado da raiz é: 1.84375

>>Metodo de Newton

import math

#calcula f(x)

def calcular\_funcao(expressao, x):

expressao = expressao.replace('x', str(x))

return eval(expressao)

#calcula a derivada

def calcular\_derivada(expressao, x):

h = 10\*\*-8

derivada =(calcular\_funcao(expressao, x+h) - calcular\_funcao(expressao, x))/h

return derivada

def metodo\_newton(expressao, chute):

k=0

while(calcular\_funcao(expressao, chute) > 10\*\*-5):

chute = chute - (calcular\_funcao(expressao, chute)/calcular\_derivada(expressao, chute))

k+=1

print(f"Numero de interações foi: {k}")

return chute

def main():

expressao = input("Digite a expressão da função (por exemplo, x\*\*2):")

chute = int(input("Digite o seu chute:"))

resultado = metodo\_newton(expressao, chute)

if(resultado < chute):

print("Chutou alto...")

else:

print("Chutou baixo\n")

print(">>> O resultado aproximado da raiz é:{:2f}".format(resultado))

if \_\_name\_\_ == "\_\_main\_\_":

main()

**######OUTPUT########**

Digite a expressão da função (por exemplo, x\*\*2):x\*\*2-4

Digite o seu chute:5

Numero de interações foi: 5, Chutou alto...

>>> O resultado aproximado da raiz é:2.000000

Digite a expressão da função (por exemplo, x\*\*2):x\*\*2

Digite o seu chute:3

Numero de interações foi: 10, Chutou alto...

>>> O resultado aproximado da raiz é:0.002930

Digite a expressão da função (por exemplo, x\*\*2):x\*\*4+x\*\*3-25

Digite o seu chute:55

Numero de interações foi: 15, Chutou alto...

>>> O resultado aproximado da raiz é:2.022383