

>>Metodo da Bisseccão

```
import math
def calcular_funcao(expressao, x):
    # irá substituir 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
```