Trabalho Aplicado 1

Erickson Giesel Müller

23 de Setembro de 2024

1 Código

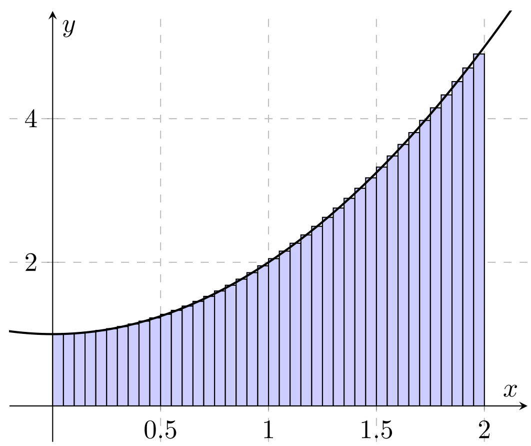
```
import sympy

def somaRiemann(funcao, a, b, n):
    x = sympy.Symbol('x')
    f = sympy.sympify(funcao)

    delta_x = (b - a) / n
    area = 0
    for i in range(n):
        x_i = a + i * delta_x + delta_x / 2
        altura = f.subs(x, x_i)
        area += altura * delta_x

    return area


funcao = input("escreva a funcao f(x): ")
a = int(input("Insira o valor de a: "))
b = int(input("Insira o valor de b: "))
n = int(input("insira o valor de n: "))

aprox_area = somaRiemann(funcao, a, b, n)
print("A area aproximada usando a Soma de Riemann
e: {:.6f} u.a".format(aprox_area))
```

Para calcular a aproximação da área abaixo da curva, o código em python usou a biblioteca *sympy*, que fornece diversas ferramentas de álgebra computacional. Para instalar a biblioteca, caso o usuário tenha o python instalado em seu computador, basta rodar no terminal o comando:

```
pip install sympy
```

2 Soma de Riemann

O programa calcula a aproximação da área fazendo a soma das áreas dos n retângulos abaixo da curva. Os retângulos tem base $\frac{b-a}{n}$ e altura f(x) no ponto médio. Semelhante à figura abaixo:



3 Como operar o programa

Para usar o programa no prompt de comando, o usuário deverá escrever:

```
python t1-20230001178.py
```

Em seguida serão solicitadas os dados de entrada:

$$y; a; b; n$$

A função é calculada no formato de string, portanto o usuário deverá escrevê-la no seguinte formato:

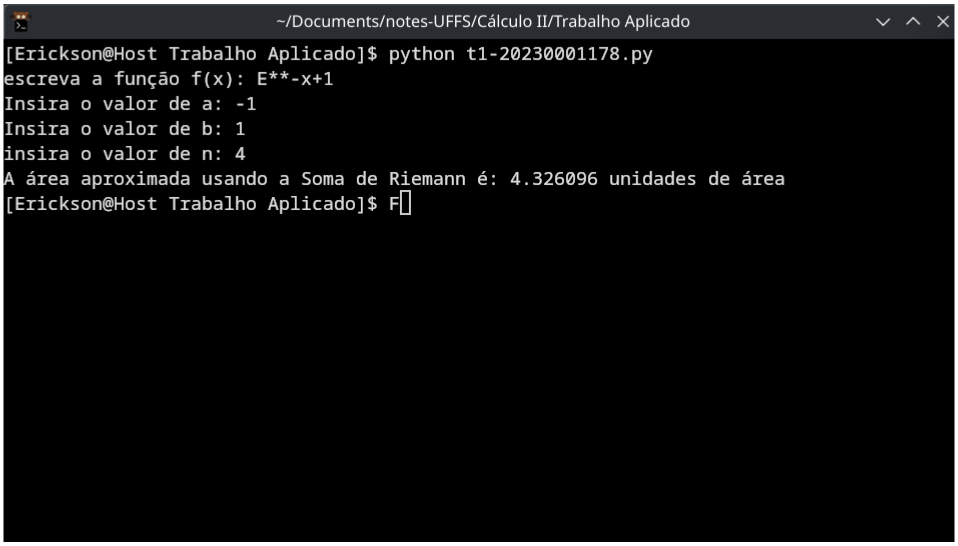
```
a*x**n+b*x**(n-1)+c*x**(n-2)...+z
```

Para representar o número e, deve-se digitar a letra maíuscula E, pois essa é a sintaxe da biblioteca *sympy*.

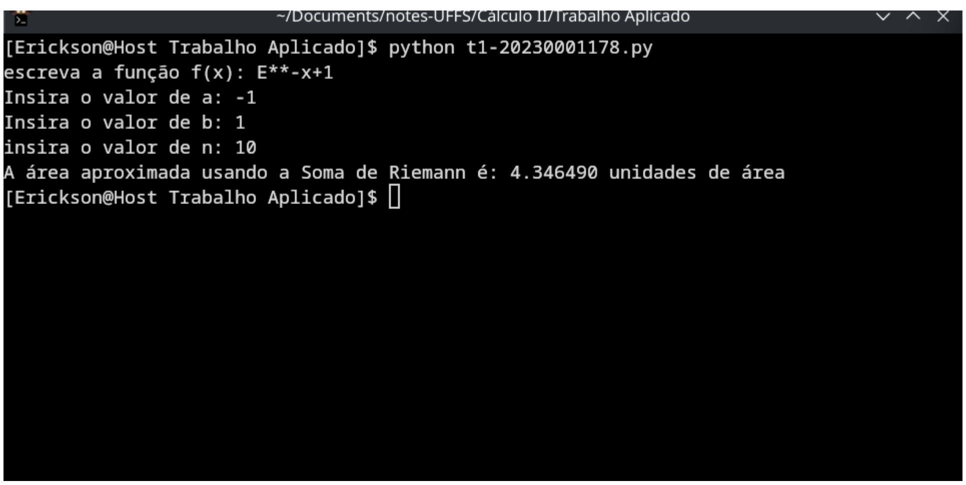
4 Resolução dos problemas:

Considerando a função $f(x) = e^{-x} + 1$, calcular as aproximações das áreas no intervalo $[-1, 1]$ com n subintervalos:

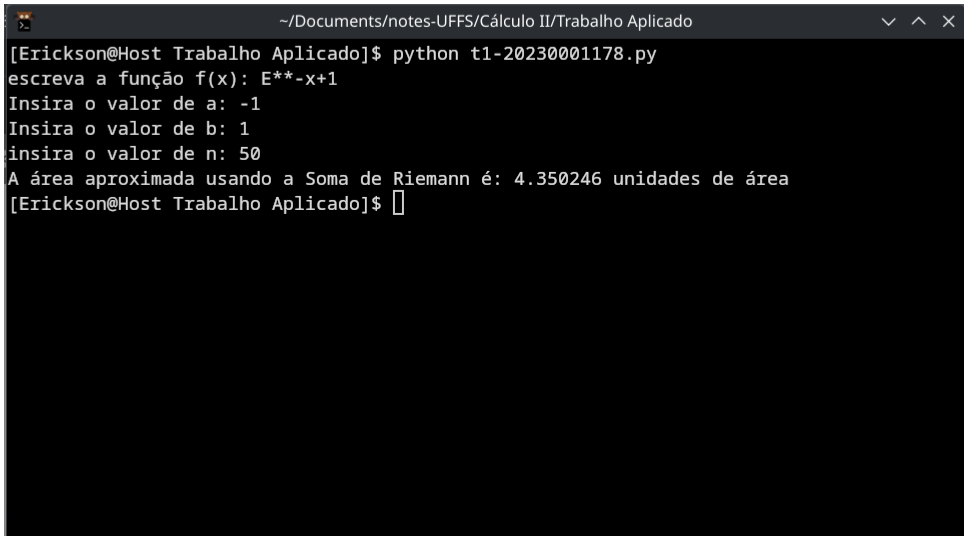
4.1 n=4


~/Documents/notes-UFFS/Cálculo II/Trabalho Aplicado
[Erickson@Host Trabalho Aplicado]$ python t1-20230001178.py
escreva a função f(x): E**-x+1
Insira o valor de a: -1
Insira o valor de b: 1
insira o valor de n: 4
A área aproximada usando a Soma de Riemann é: 4.326096 unidades de área
[Erickson@Host Trabalho Aplicado]$ F

4.2 n=10

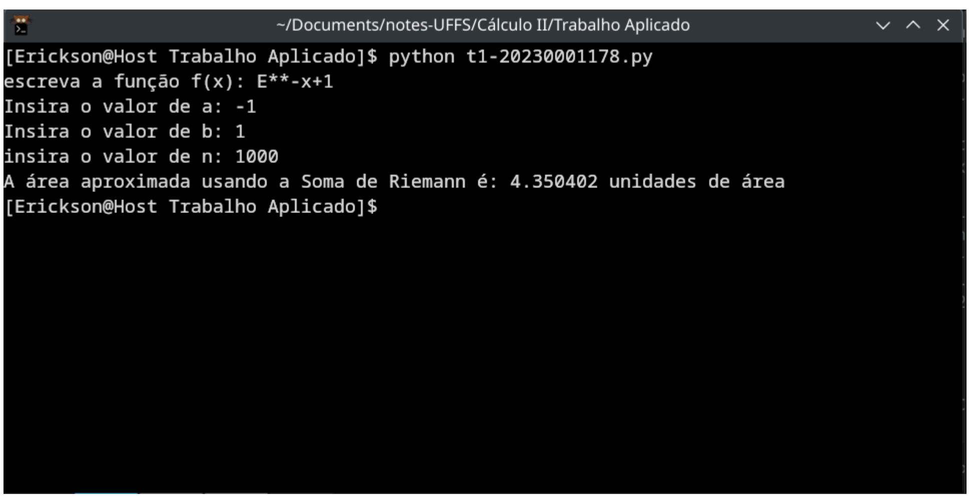

~/Documents/notes-UFFS/Cálculo II/Trabalho Aplicado
[Erickson@Host Trabalho Aplicado]$ python t1-20230001178.py
escreva a função f(x): E**-x+1
Insira o valor de a: -1
Insira o valor de b: 1
insira o valor de n: 10
A área aproximada usando a Soma de Riemann é: 4.346490 unidades de área
[Erickson@Host Trabalho Aplicado]$

4.3 n=50



```
~/Documents/notes-UFFS/Cálculo II/Trabalho Aplicado
[Erickson@Host Trabalho Aplicado]$ python t1-20230001178.py
escreva a função f(x): E**-x+1
Insira o valor de a: -1
Insira o valor de b: 1
insira o valor de n: 50
A área aproximada usando a Soma de Riemann é: 4.350246 unidades de área
[Erickson@Host Trabalho Aplicado]$
```

4.4 n=1000



```
~/Documents/notes-UFFS/Cálculo II/Trabalho Aplicado
[Erickson@Host Trabalho Aplicado]$ python t1-20230001178.py
escreva a função f(x): E**-x+1
Insira o valor de a: -1
Insira o valor de b: 1
insira o valor de n: 1000
A área aproximada usando a Soma de Riemann é: 4.350402 unidades de área
[Erickson@Host Trabalho Aplicado]$
```

5 Considerações Finais

Analisando os resultados, percebemos que quanto maior o número de subintervalos, mais preciso será o resultado da aproximação em relação à integral. Isso acontece porque na àrea dos retângulos que está sendo calculada, há uma área de imprecisão das partes dos retângulos que estão à esquerda e à direita do ponto médio da base. Essa imprecisão é reduzida conforme a largura da base dos retângulos diminui, e estes pontos ficam cada vez mais próximos da curva.

Conforme a fórmula da Soma de Riemann, percebe-se que o resultado fornecido pelo programa fica cada vez mais próximo do valor exato da integral conforme o número de subintervalos aumentam.Também, quanto maior for número de subintervalos, mais sutil será a diferença se comparado com o resultado de um valor n próximo, por esse motivo o resultado é entregue com 6 casas decimais.

$$\int_a^b f(x).dx = \lim_{x \to +\infty} \sum_{i=1}^{h} f(C_i).\Delta x$$

Utilizando os valores do exemplo acima:

$$y = e^{-x} + 1$$

$$Area = \int_{-1}^{1} [e^{-x} + 1].dx = [x - e^{-x}]_{-1}^{+1}$$

$$[1 - e^{-1}] - [-1 - e^{1}] = 2 + e - e^{-1}$$

$$Area \approx 4.35040238729u.a$$

Portanto, para essa função, sob essas condições, o programa conseguiu atingir um resultado preciso com 1000 subintervalos.