Universidade Federal da Fronteira Sul - UFFS

Curso: Ciência da Computação (3ª Fase)

Disciplina: Cálculo 2

Professor: Milton Kist

Alunos: Luan Bortoli / Yuri Luis Malinski Lanzini

Para execução dos programas, acessar o link:

https://colab.research.google.com/drive/1A_M8OqdYRRdwd2N2DDeFurHKS0jQED6o?usp=sharing

Dada a função f:[a,b] -> (0, ∞) definido por y=f(x) continua. Determine uma aproximação para área da região plano limitada pelas retas x=a, x=b, y=0 e a curva y=f(x).

Subintervalos: n=4, n=10, n=50 e n=1000

Problema 1: f(x) = e^(-x) + 1 a = 0 b = 2

Sendo o intervalo de [0,2], e os subintervalos chamados de n no inicio do exercicio.

Utilizando a soma de Riemman

```
import math

#Definição da função f(x)
def f(x):
  return math.exp(-x) + 1

#Definição do cálculo da soma de Riemann
def soma_riemann(a, b, n):
  delta_x = (b - a) / n
  soma = 0

  for i in range(1, n + 1):
    x_i = a + i * delta_x
    soma += f(x_i) * delta_x
  return soma

#Definição dos intervalos
a = 0
b = 2

#Definição dos subintervalor
n = [4, 10, 50, 1000]

#Cálculo e exibição da soma de Riemann para cada valor do subintervalo
for n_i in n:
```

```
  area = soma_riemann(a, b, n_i)
  print(f"Área para o subintervalos n = {n_i} é: {area:.6f}")
```

```
Área para o subintervalos n = 4 é: 2.666438
Área para o subintervalos n = 10 é: 2.781079
Área para o subintervalos n = 50 é: 2.847487
Área para o subintervalos n = 1000 é: 2.863800
```

Problema 2: f(x) = x^2 + 1 a = 0 b = 2

Sendo o intervalo de [0,2], e os subintervalos chamados de n no inicio do exercicio.

Utilizando a soma de Riemman

```
import math

#Definição da função f(x)
def f(x):
  return x**2 + 1

#Definição do cálculo da soma de Riemann
def soma_riemann(a, b, n):
  delta_x = (b - a) / n
  soma = 0

  for i in range(1, n + 1):
    x_i = a + i * delta_x
    soma += f(x_i) * delta_x
  return soma

#Definição dos intervalos
a = 0
b = 2

#Definição dos subintervalor
n = [4, 10, 50, 1000]

#Cálculo e exibição da soma de Riemann para cada valor do subintervalo
for n_i in n:
  area = soma_riemann(a, b, n_i)
  print(f"Área para o subintervalos n = {n_i} é: {area:.6f}")
```

```
Área para o subintervalos n = 4 é: 5.750000
Área para o subintervalos n = 10 é: 5.080000
Área para o subintervalos n = 50 é: 4.747200
Área para o subintervalos n = 1000 é: 4.670668
```

Percepções sobre os resultados

Função (f(x) = e^(-x) + 1):

Observa-se uma leve variação na aproximação da área conforme aumentamos o número de subintervalos. Com (n=4), a soma de Riemann é (2.666438), e com (n=1000) é (2.781079). O comportamento da função (e^(-x) + 1) decresce rapidamente, tornando as aproximações para valores baixos de (n) mais precisas desde o início, já que a função tende rapidamente a se estabilizar. Apesar de uma leve diferença com valores pequenos de (n), a área converge para valores semelhantes quando (n) aumenta, com as aproximações de (n=50) e (n=1000) sendo bastante próximas.

Função (f(x) = x^2 + 1):

A área aproximada diminui à medida que aumentamos o número de subintervalos (n). Para (n=4), a soma de Riemann é de (5.750000), enquanto para (n=1000) é de (4.670668). Essa convergência para um valor menor reflete que, com menos subintervalos, a aproximação da área considera uma superestimação, especialmente em funções crescentes como (x^2 + 1), onde os retângulos sobreestimam a área sob a curva. À medida que o valor de (n) aumenta, os retângulos tornam-se mais estreitos, melhorando a precisão da aproximação. O valor para (n=1000) está mais próximo da verdadeira área sob a curva.

Comparação entre as funções:

A função (x^2 + 1), sendo quadrática, cresce rapidamente, tornando a soma de Riemann menos precisa para valores pequenos de (n), enquanto a função (e^(-x) + 1), que decresce rapidamente, fornece aproximações razoáveis mesmo para (n) pequenos. Para funções com crescimento mais rápido, como (x^2 + 1), o número de subintervalos (n) tem um impacto mais significativo na precisão da área aproximada.

Essa análise evidencia como o comportamento da função influencia a precisão da soma de Riemann e a importância de escolher um número adequado de subintervalos.