Tarefa Análise de Erros na Integração Numérica

Matheus Coelho Frank - 00221444

IF - UFRGS

Resumo

Este trabalho tem por objetivo analisar brevemente os erros de Integração dos métodos do Trapézio e de Simpson, provenientes do Cálculo Numérico . Para isso foi utilizada um programa de computador escrito na linguagem Python. Após a análise, foi possível concluir que o método de Simpson possui vantagens em relação ao método do Trapézio, quando a integração numérica é requirida.

1 Método

A fim de analisar os erros atribuídos aos dois métodos de Integração Numérica, o Método do Trapézio e o Método de Simpson, foi escolhida como exemplo a função

$$\int_{-3}^{3} \frac{1}{x^2+1} dx \tag{1}$$

que tem como resultado análitico

$$2arctan(3) \tag{2}$$

e fazendo uso da linguagem de Programação Python e de seus módulos Numpy e Matplotlib, escrevemos o seguinte programa:

```
import numpy as np
import matplotlib.pyplot as plt

#Funções
int_analitica = 2*np.arctan(3)

def f(x):
  return 1/(x**2 + 1)

def dx(a, b, n):
  return (b - a) / n

#Trapezio

def int_trapezio(f,a,b,N):
  integral = 0
  dx = (b - a) / N
  for i in range(N):
    x = a + i*dx
    trapezio = ((f(x)+f(x+dx))/2)*dx
    integral = integral + trapezio
  return integral

def erro_trapezio(int_analitica, int_trapezio):
  return abs(int_analitica - int_trapezio)

#Simpson

def int_simpson(f, a, b, n):
    if n & 1:
```

```
        print ("Erro: n não é par.")
        return 0
    dx = (b - a) / n
    integral = 0
    x = a
    n_div = int(n / 2)
    for i in range(0, n_div):
        integral += f(x) + (2 * f(x + dx))
        x += 2 * dx
    integral = (2 * integral) - f(a) + f(b)
    integral = dx * integral / 3
    return integral

def erro_simpson(int_analitica, int_simpson):
  return abs(int_analitica - int_simpson)

#Listas
lista_erros_trapezio = []
lista_dx_trapezio = []
lista_erros_simpson = []
lista_dx_simpson = []

for n in 2**np.arange(2,22,2):
  lista_erros_simpson.append(erro_simpson(int_analitica, int_simpson(f, -3, 3, n)))
  lista_dx_simpson.append(dx(-3,3,n))
  lista_erros_trapezio.append(erro_trapezio(int_analitica, int_trapezio(f, -3, 3, n)))
  lista_dx_trapezio.append(dx(-3,3,n))

array_erros_trapezio = np.array(lista_erros_trapezio)
array_dx_trapezio = np.array(lista_dx_trapezio)
array_erros_simpson = np.array(lista_erros_simpson)
array_dx_simpson = np.array(lista_dx_simpson)

#Gráficos
figura = plt.figure(figsize=(15, 5))
plt.subplot(121)
plt.scatter(np.log10(array_dx_trapezio),np.log10(array_erros_trapezio))
plt.plot(np.log10(array_dx_trapezio),np.log10(array_erros_trapezio))
plt.ylabel('Log Erro')
plt.xlabel('dx')
plt.grid()
plt.title('Erro Método Trapézio')

plt.subplot(122)
plt.scatter(np.log10(array_dx_simpson),np.log10(array_erros_simpson))
plt.plot(np.log10(array_dx_simpson),np.log10(array_erros_simpson))
plt.ylabel('Log Erro')
plt.xlabel('dx')
plt.grid()
plt.title('Erro Método de Simpson')

plt.savefig('grafico.png')

#Inclinação
a1=(np.log10(array_erros_trapezio[5])-np.log10(array_erros_trapezio[0]))/(np.log10(array_dx_trapezio[5])-np.l
print(f'Inclinação Trapezio {a1:.2}')
a=(np.log10(array_erros_simpson[5])-np.log10(array_erros_simpson[0]))/(np.log10(array_dx_simpson[5])-np.log10
print(f'Inclinação  Simpson {a:.2}')
```

2 Resultados do Método

Os resultados obtidos pelo programa podem ser conferidos a seguir:

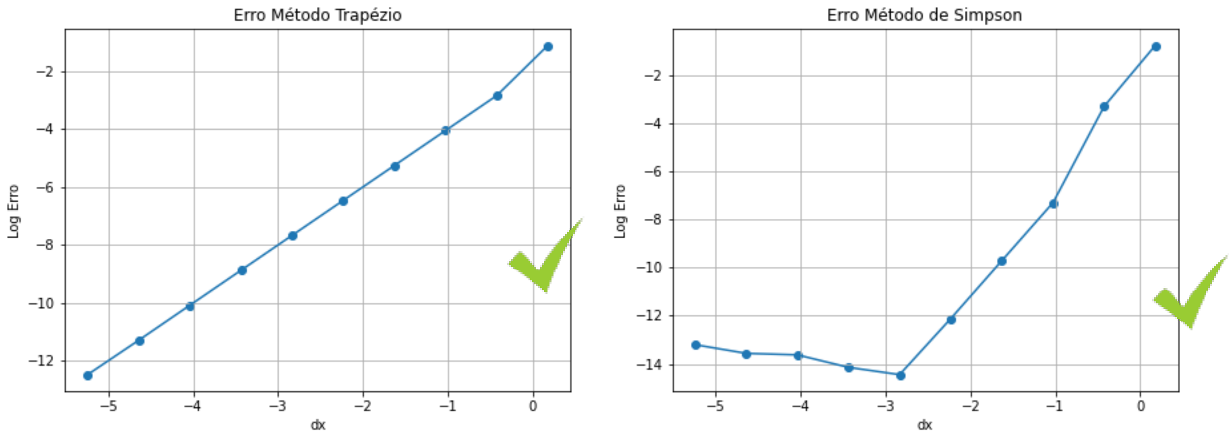

Figura 1: Gráficos dos Logaritmos dos Erros para cada método *versus* dx.

- Coeficiente de Inclinação para o Método dos Trapézios: 2.2
- Coeficiente de Inclinação para o Método dos Trapézios: 4.5

3 Conclusão

Após comparar os dois métodos em estudo, que o Método de Simpson atinge um platô de erro muito mais rapidamente que o método dos Trapézios, que neste caso, devido a precisão de 15 casas decimais envolvida da função $2arctan(3)$ (resultado analítico da integral (1)) não chega a atingir o limite de truncamento. Nesse sentido, é possível concluir que o método mais vantajoso é sem sombra de dúvidas, o Método de Simpson, para integrações numéricas.

É possível notar também que os erros se aproximam de suas ordens de grandeza teóricas, apesar de notadamente possuírem erros significativos frente aos valores esperados.