Erros na Integração Numérica

Leandra W. Rodrigues Machado

November 2021

1 Introdução

A atividade ”Erros na Integração numérica” tinha como objetivo o estudo do cálculo da integração por duas diferentes fórmulas. A primeira conhecida como ”Método do Trapézio” e a segunda ”Método de Simpson”.

Após o cálculo da integral da função dada no exercício, foi calculado o erro. $erro = |Iexata - I(N)|$. Onde $Iexata$ é o resultado da integral analítica e $I(N)$ é o valor da integral obtido dividindo o intervalo de integração em N partes.

2 Métodos

A função dada no enunciado foi :

$$\frac{1}{x^2+1}$$

O método da integração pela regra do Trapézio segue a seguinte fórmula:

$$\int_a^b f(x)\,dx \;=\; \sum_{n=1}^{\infty} \frac{f(xi)+f(xi-1)}{2}\Delta(x)$$

O código implementado em python para o cálculo da integral de $f(x)$ pela regra do trapézio foi o seguinte, em seguida também foi calculado o erro:

$$\int_a^b f(x)\,dx = \frac{1}{2}h\sum_{n=1}^{N}\left[f(xk-1)+f(xk)\right] + \frac{1}{12}h^2\left[f'(a)-f'(b)\right] + O\left(h^4\right)$$

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

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

#metodo trapezio:

def trapezio(f,a,b):
  N = np.logspace(2 , 20, num = 10, base = 2, dtype = int)

  erro = []
  for i in range(10):
    dx=((b-a)/N[i])
    integral = dx * ((f(a) + f(b))/2)

    for j in range(1, N[i]):
      x = a + j*dx
      integral = integral + (dx*f(x))

    erro_trap=(abs( 2.498091544796508851659 - integral ))
```

```
    erro.append(erro_trap)
  return erro

erro_trapezio = trapezio(f, -3, 3)
```

E o método da integração pela regra de Simpson segue a regra:

$$\int_a^b f\left(x\right)dx \;=\; \frac{h}{3}\left[f\left(a\right)+f\left(b\right)+4\sum_{n=1}^{\frac{N}{2}}\;f\left(a+\left(2i-1\right)h\right)+2\sum_{n=1}^{\frac{N}{2-1}}f\left(a+2ih\right)\right]$$

Após o cálculo da função pela regra de simpson, foi necessário também calcular o erro:

$$erro \;=\; \frac{1}{90}h^4\left[f'''\left(a\right)-f'''\left(b\right)\right]$$

o código implementado para o cálculo do erro e da integral foi:

```
  #metodo simpson integral:

def simpson(f,a,b):
  N_s =  np.logspace(2,20, num = 10, base =2, dtype=int)

  list_simpson=[]

  for i in range(10):
    soma_imp = 0
    soma_par = 0
    h=((b-a)/N_s[i])

    for k in range(1,(N_s[i]), 2):
      soma_imp += f(a + (k* h))

    for j in range(2,(N_s[i]), 2):
      soma_par += f(a + (j * h))

    integral_simpson = (h/3) * (f(a) + f(b) + (4*soma_imp) + (2*soma_par))

    errosimpson=(abs(2.498091544796508851659-integral_simpson))
    list_simpson.append(errosimpson)
  return list_simpson

erro_sp = simpson(f,-3,3)


lista_h=[]
```

```
for k in range(10):
  h = (b-a)/N_grafico[k]
  lista_h.append(h)

#erro_simpson_log = np.log10(erro_sp)
#h_log = np.log10(lista_h)
```

3 Resultados

Após a implementação do código em python, foram gerados gráficos com os resultados das funções dos erros obtidos.

Pela regra do trapézio o gráfico obtido foi:

```
plt.plot(dx_log, erro_trapezio_log,  'k', label= 'Integral Método do Trapézio', color = 'pir
plt.xlabel('log()')
plt.ylabel('log(x)')
plt.legend()
plt.show()
```

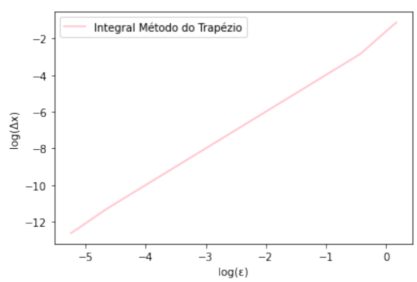

Figure 1: Gráfico Erro Método Trapézio

Seguindo a regra de Simpson o gráfico obtido foi:

```
plt.plot(lista_h, erro_sp,  'k', label= 'Integral Método de Simpson', color = 'purple')
plt.xscale('log')
plt.yscale('log')
plt.xlabel('log()')
plt.ylabel('log(x)')
plt.legend()
plt.show()
```

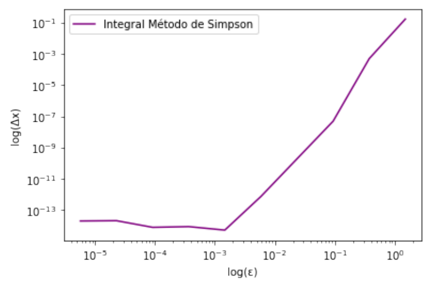

Figure 2: Gráfico Erro Método Simpson