Integração Numérica : Regra dos Trapézios

Prof. Doherty Andrade

www.metodosnumericos.com.br

1 Introdução

A regra dos trapézios é o método mais simples para determinar uma aproximação da integralde uma função $f(x)$ sobre um intervalo $[a,b]$:

$$\int_a^b f(x)dx.$$

A Regra dos Trapézios aproxima a integral pela soma:

$$\int_a^b f(x)dx \approx \frac{h}{2}\sum_{k=1}^{N}(f(x_k)+f(x_{k-1}))$$

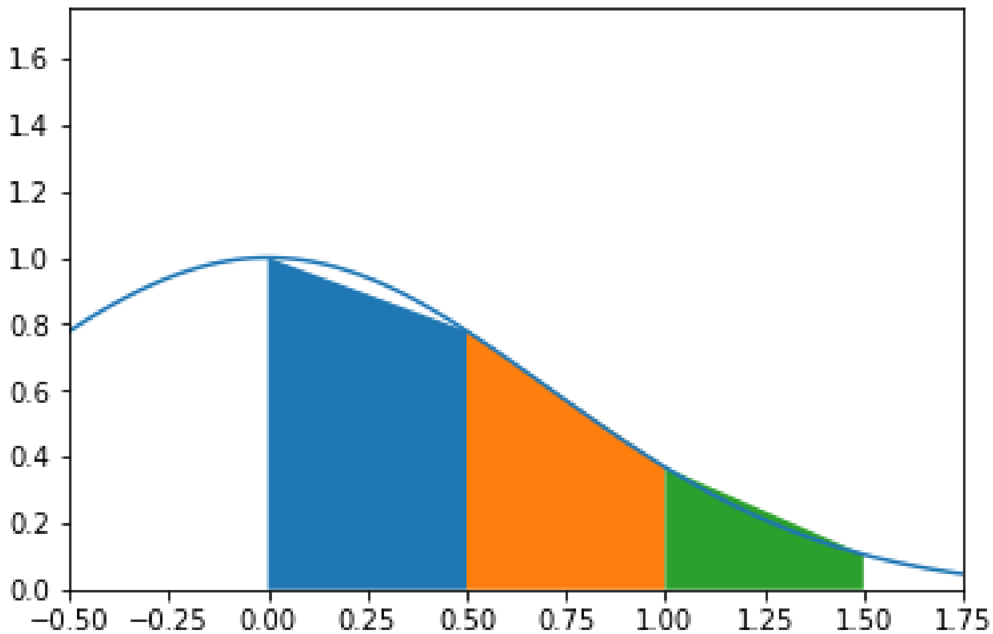
onde $x_k = a + kh$ e $h = \frac{b-a}{N}$.

Geometricamente, se $f(x) \geq 0, \forall x \in [a,b]$, a integral $\int_a^b f(x)dx$ é igual a área da região compreendida entre as retas $x = a$, $x = b$, o eixo OX e abaixo do gráfico de f. O método dos trapézios divide o intervalo $[a,b]$ em N subintervalos $[x_{i-1}, x_i]$ de comprimento $h = \frac{b-a}{N}$, e calcula área do trapézio com bases $f(x_{i-1})$ e $f(x_i)$ e altura $x_i - x_{i-1} = h$. A soma das áreas dos trapézios aproxima a área abaixo do gráfico de f.

Ilustramos a seguir um trapézio.

```
In [14]: import numpy as np
         import matplotlib.pyplot as plt
         %matplotlib inline

         "Ilustrando a regra dos trapézios"
         f = lambda x: np.exp(-x**2)
         x = np.linspace(-0.5,1.75,100)
         y = f(x)
         plt.plot(x,y)

         x0 = 0; x1 = 0.5;
         y0 = f(x0); y1 = f(x1);

         x2 = 0.5; x3 = 1.0;
         y2 = f(x2); y3 = f(x3);
```

```
x4 = 1.0; x5 = 1.5;
y4 = f(x4); y5 = f(x5);

plt.fill_between([x0,x1],[y0,y1])
plt.fill_between([x2,x3],[y2,y3])
plt.fill_between([x4,x5],[y4,y5])

plt.xlim([-0.5,1.75]); plt.ylim([0,1.75]);
plt.show()
```



2 Procedimento em Python

No procedimento abaixo a sintaxe é trapezios(f,a,b,N), onde f é a função a ser integrada, a e b são os extremos do intervalo e N é o número de subintervalos em que dividimos $[a, b]$.

```
In [34]: def trapezios(f,a,b,N=50):
             '''Aproxima a integral de  f(x) no intervalo  [a,b] pela Regra dos Trapézios.

             A Regra dos Trapézios aproxima a integral de  \int_a^b f(x) dx pela soma:
             (h/2) \sum_{k=1}^N (f(x_k) + f(x_{k-1}))
             onde x_k = a + k*h e h = (b - a)/N.

             Parâmetros
             ----------
             f : função
```

```
        Função de uma variável vetorizada
    a , b : números
        Intervalo de integração [a,b]
    N : inteiro
        Número de subintervalos de [a,b]

    Retorno
    -------
        Aproximação da integral de f(x) no intervalo [a,b]
        usando a Regra dos Trapézios com N subintervalos de mesmo tamanho h.

    Exemplo
    --------
    >>> trapezios(np.sin,a=0,b=np.pi/2,N=1000)
    0.99999979438323316
    '''
    x = np.linspace(a,b,N+1)
    y = f(x)
    y_right = y[1:] # extremos a direita
    y_left = y[:-1] # extremos a esquerda
    h = (b - a)/N
    T = (h/2) * np.sum(y_right + y_left)
    print('A aproximação da integral é:')
    return T
```

3 Exemplos

3.1 Exemplo 1 : calcular $\int_0^{\frac{\pi}{2}} \sin(x)dx$.

```
In [35]: trapezios(np.sin,0,np.pi/2,1000)

A aproximação da integral é:


Out[35]: 0.9999997943832332
```

3.2 Exemplo 2: calcular $\int_0^1 \exp(-x^2)dx$.

```
In [36]: g = lambda x: np.exp(-x**2)

In [31]: trapezios(g,0,1,1000)

A aproximação da integral é:


Out[31]: 0.7468240714991848

In [ ]:
```