

# Integração Numérica: Regra de Simpson

Prof. Doherty Andrade

www.metodosnumericos.com.br

## 1 Regra de Simpson

A Regra de Simpson calcula uma aproximação para  $\int_a^b f(x)dx$  pela fórmula

$$\int_a^b f(x)dx \approx \frac{h}{3} \left[ f(x_0) + 4 \sum_{i=1}^{\frac{n}{2}} f(x_{2i-1}) + 2 \sum_{i=1}^{\frac{n-2}{2}} f(x_{2i}) + f(x_n) \right].$$

Aqui  $h = \frac{b-a}{N}$  e  $N$  é o número de subintervalos em que  $[a, b]$  foi dividido. O método de Simpson consiste em interpolar em cada três pontos  $(x_{i-2}, y_{i-2}), (x_{i-1}, y_{i-1}), (x_i, y_i), i = 2, \dots, N$  uma parábola, então  $N$  tem que ser par. Note que iniciamos a contagem a partir de zero.

## 2 Procedimento em Python

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: def simpson(f, a, b, N=50):
'''Determina uma aproximação para a integral de f(x)
de a até b pela Regra de Simpson.
Parâmetros
-----
f : função
    função de uma variável vetorizada
a , b : números
    Intervalo de integração [a,b]
N : (par) inteiro par
    Número de subintervalos de [a,b]

Retorna
```

-----  
*Aproximação da integral de  $f(x)$  de  $a$  até  $b$  usando  
a Regra de Simpson com  $N$  (par) subintervalos de mesmo comprimento.*

*Exemplo:*

```
-----  
simpson(lambda x : 3*x**2,0,1,10)  
'''  
if N % 2 == 1:  
    raise ValueError("N deve ser um inteiro par.")  
h = (b-a)/N  
x = np.linspace(a,b,N+1)  
y = f(x)  
S = (h/3) * np.sum(y[0:-1:2] + 4*y[1::2] + y[2::2])  
print('A aproximação da integral por Simpson é:')  
return S
```

### 3 Exemplo: calcular

$$\int_0^1 \exp(-x^2) dx$$

```
In [3]: simpson(lambda x: np.exp(-x**2),0,1,1000)
```

A aproximação da integral por Simpson é:

```
Out[3]: 0.7468241328124351
```

### 4 Exemplo: calcular

$$\int_0^1 3x^2 dx.$$

```
In [4]: simpson(lambda x : 3*x**2,0,1,10)
```

A aproximação da integral por Simpson é:

```
Out[4]: 1.0
```

```
In [ ]:
```