L10Z10Maurycy

December 16, 2020

[1]:
```
import requests
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
def mystery_f(x):
    URL = 'http://roxy.pythonanywhere.com/f3?x={}'.format(x)
    r = requests.get(url = URL)
    num = r.text
    return float(num)
```

[3]:
```
def Netwon_Cotes(f, a=0, b=1):
    N1 = (b-a)/2 * (f(a) + f(b))
    N2 = (b-a)/6 * (f(a) + 4*f((a+b/2)) +f(b))

    return N1, N2
```

[4]:
```
def trapezoids(n, f=mystery_f, a=0, b=1):
    h = (b-a)/n
    s = np.linspace(a,b,n)

    return np.sum(np.array([h*f(x) for x in s])) - 0.5*h*(f(0) + f(1))
```

[5]:
```
def get_results(f, a=0, b=1):
    T = np.array([trapezoids(x) for x in 2**np.arange(0,6)])
    S = 1/3 * np.array([4*T[i] - T[i-1] for i in range(1,len(T))])

    return S, T
```

[6]:
```
S, T = get_results(mystery_f)
N1, N2 = Netwon_Cotes(mystery_f)
```

Przybliżenia wartości $\int_0^1 f(x)dx$ dla poszczególnych metod i n

[7]:
```
print('n:               ',2**np.arange(1,6))
print('Metoda Simpsona: ',S)
print('Metoda trapezów: ',T[1:])
```

```
n:                  [ 2  4  8 16 32]
Metoda Simpsona:  [1.38337223 1.04538701 1.77869929 1.70566395 1.74400614]
Metoda trapezów:  [0.85835278 0.99862845 1.58368158 1.67516835 1.7267967 ]
```

Wyniki dla kwadratur Newtona-Cotesa:

[8]:
```python
print('n = 1: ',N1,'\nn = 2: ',N2)
```

```
n = 1:  1.7167055642043803
n = 2:  1.9908180376296116
```

0.0.1 Błąd

Z wykładu wiemy, że błąd w metodzie Simpsona jest postaci: Przy całkowaniu funkcji $f(x) \in C^{(4)}$ na przedziale $[a,\, b]$ błąd metody wynosi:

$$R = -\tfrac{(b-a)h^4}{180} f^{IV}(x), \quad x \in [a,\, b]$$

zatem w naszym przypadku:

$$R = -\frac{1}{180 \cdot n^4} f^{IV}(x) \leq -\frac{1}{180 \cdot n^4} \cdot 1,61 \cdot 10^5$$

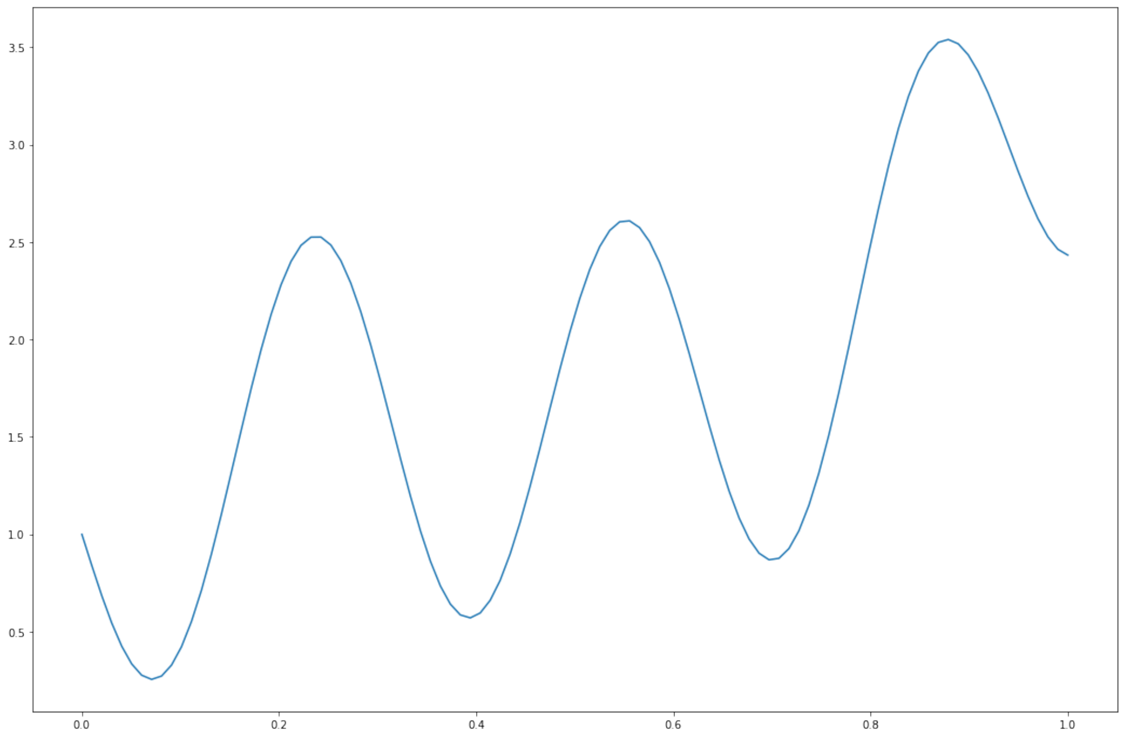
Błędy metody Simpsona dla $n \in [2, 4, 8, 16, 32]$

[9]:
```python
print('n:     ',2**np.arange(1,6))
print('Błędy: ',- 1 / (180 * (2**np.arange(1,6))**4) * 1.61 * 1e5)
```

```
n:      [ 2  4  8 16 32]
Błędy:  [-5.59027778e+01 -3.49392361e+00 -2.18370226e-01 -1.36481391e-02
 -8.53008694e-04]
```

[11]:
```python
x = np.linspace(0,1,100)
y = np.array([mystery_f(xs) for xs in x])
```

[12]:
```python
plt.figure(figsize=(18,12))
plt.plot(x,y)
plt.show()
```


3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.2
0.4
0.6
0.8
1.0