Laboratorium 4

Piotr Witek

14 kwietnia 2021

1 Zadania

1.1 Obliczyć $\mathrm{I} = {}_0\int^1 1/(1+\mathrm{x})\mathrm{dx}$ wg wzoru prostokątów, trapezów i wzoru Simpsona (zwykłego i złożonego $\mathrm{n} = 3, 5$). Porównać wyniki i błędy.

Dokładna wartość całki wynosi:

$$\int_0^1 1/(1+x)dx = \ln(x+1)|_0^1 = \ln 2 \approx 0.69314718056$$

Całkowanie metodą prostokatów na zadanym przedziale [a,b] polega na obliczeniu następującej sumy:

$$I_n = \sum_{i=1}^{n} f\left(c_i\right) * \frac{|b-a|}{n}$$

gdzie: c_i to środek każdego z n podprzedziałów przedziału $[a, b]$ Całka jest więc równa sumie pól prostokątów o szerokości każdego podprzedziału i wysokości równej wartości funkcji całkowanej w środku tego przedziału.

Całkowanie metodą trapezów polega na policzeniu poniższej sumy:

$$I_n = \sum_{i=2}^{n} \frac{h}{2}\left(f\left(c_{i-1}\right) + f\left(c_i\right)\right)$$

gdzie: c_i to każdy z n punktów, powstałych poprzez podział przedziału [a,b] na $n-1$ przedziałów. Punkty oddalone są od siebie o h

Całkowanie metoda Simpsona polega na policzeniu poniższej sumy:

$$I_n = \sum_{i=3}^{n} \frac{h}{3}\left(f\left(c_i\right) + 4f\left(c_{i-1}\right) + f\left(c_{i-2}\right)\right)$$

gdzie: c_i to każdy z n punktów, gdzie $n = 2k+1$, powstałych poprzez podział przedziału [a,b] na $n-1$ równych podprzedziałów.

1. Zwykła kwadratura prostokatów jest tożsama ze złożoną kwadraturą prostokątów dla $n = 1$
2. Zwykla kwadratura trapezów jest tożsama ze złożoną kwadraturą trapezów dla $n = 2$
3. Zwykla kwadratura Simpsona jest tożsama ze złożoną kwadraturą Simpsona dla $n = 3(k = 1)$

Obliczenia powyższych kwadratur wykonałem przy pomocy programu w Pythonie, któego kod znajduje się poniżej.

```
import numpy as np
from math import log

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

def rectangle():
    for steps in [1,3,5]:
        length = 1
        xs = np.linspace(0, length, num=steps+1, endpoint=False)
        Int = sum(map(lambda x: f(x)*length/steps, xs[1::]))

        print("n =", steps, Int, "bład bezwględny:", Int - log(2))

def trapez():
    for steps in [2,3,5]:
        length = 1
        h = length/(steps-1)
        xs = np.linspace(0, length, num=steps, endpoint=True)
        Int = sum(map(lambda x: (f(x)+f(x-h))*h/2, xs[1::]))

        print("n =", steps, Int, "bład bezwględny:", Int - log(2))

def simpson():
    for steps in [3,5]:
        length = 1
        h = length/(steps-1)
        xs = np.linspace(0, 0+length, num=(steps//2)+1, endpoint=True)
        Int = sum(map(lambda x: (f(x)+4*f(x-h)+f(x-2*h))*(h/3), xs[1::]))

        print("n =", steps, Int, "bład bezwględny:", Int - log(2))
```

Otrzymano wyniki (odpowiednio dla metod prostokatów, trapezów, Simpsona):

```
n = 1 0.6666666666666666 bład bezwględny: -0.026480513893278657
n = 3 0.6793650793650794 bład bezwględny: -0.013782101194865892
n = 5 0.6838528138528138 bład bezwględny: -0.009294366707131463
---------
n = 2 0.75 bład bezwględny: 0.056852819440054714
n = 3 0.7083333333333333 bład bezwględny: 0.015186152773387973
n = 5 0.6970238095238095 bład bezwględny: 0.0038766289638642037
---------
n = 3 0.6944444444444443 bład bezwględny: 0.0012972638844990225
n = 5 0.6932539682539682 bład bezwględny: 0.0001067876940229473
```

1.2 Obliczyć całkę $\mathrm{I} = {}_{-1}\int^{1} 1/\left(1+\mathrm{x}^2\right)$ dx korzystając z wielomianów ortogonalnych (np. Legendre'a) dla $\mathrm{n} = 8$.

Dokładna wartość naszej całki wynosi:

$$I = \int_{-1}^{1} 1/\left(1+x^2\right) dx = \operatorname{arctg}(x)|_{-1}^{1} = \frac{\pi}{2}$$

Wykorzystuje wielomiany Legendre'a, które zadane są wzorem rekurencyjnym:

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$
$$P_0(x) = 1$$
$$P_1(x) = x$$

Są one ortogonalne na przedziale całkowania $[-1,1]$ z wagą $w(x) = 1$.

Całkę będziemy przybliżać całką wielomianu aproksymującego szukaną funkcję, w naszym przypadku funkcja aproksymująca będzie mieć postać:

$$F(x) = \sum_{i=0}^{2} c_i P_i(x), \quad x \in [-1,1]$$

Wyznaczam współczynniki c_i ze wzoru:

$$c_i = \frac{\int_{-1}^{1} f(x)L_i(t)dx}{\int_{-1}^{1} L_i^2(x)dx}$$

Następnie przy pomocy programu napisanego w języku Python wygenerowałem wielomiany Legendre'a (def genLegendre), wyznaczyłem współczynniki c_i (def approxF). Kod poniżej.

```
import numpy as np
from scipy.integrate import quadrature

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


def genLegendre(n):
    p = []
    p.append(np.poly1d([1]))
    p.append(np.poly1d([1,0]))
    x = np.poly1d([1,0])
    for i in range(1,n):
        p.append((2*i+1)/(i+1)*p[i]*x-(i/(i+1))*p[i-1])
    return p

def approxF(n):
    L = genLegendre(n)
    c = []
    fA = []
    fA.append(np.poly1d([0]))
    for i in range(0,n):
        c.append(quadrature((lambda x: f(x)*L[i](x)), -1,1)[0]/
        quadrature((lambda x: L[i](x)**2), -1, 1)[0])
        fA.append(c[i]*L[i])
    return sum(fA)

def Int(n):
    i = approxF(n).integ()
    return i(1) - i(-1)
```

Dzięki funkcji genLegendre() otrzymuję funkcję aproksymującą, której całka będzie przybliżeniem całki, którą mam obliczyć. Wynik programu:

```
Wartość obliczona:  1.5707963251251216 błąd bezwględny:  -1.6697749849470256e-09
```

2 Zadania domowe

2.1 Obliczyć całkę $_0\int^1 1/\left(1+\mathrm{x}^2\right)$ dx korzystając ze wzoru prostokątów, trapezów i wzoru Simpsona dla $\mathrm{h} = 0.1$

Dokładna wartość całki:

$$\int_0^1 \frac{1}{(1+x^2)} = \tan^{-1}(x) = \frac{\pi}{4} \approx 1.5707963267948966192$$

Analogicznie do zadania 1, po niewielkich modyfikacjach, korzystam z przygotowanego wcześniej programu:

```
import numpy as np
from math import log

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

def rectangle():
    length = 1
    h=0.1
    xs = np.linspace(0, 1, num=10, endpoint=False)
    Int = sum(map(lambda x: f(x)*h, xs[1::]))

    print("wartość: ", Int, "bład bezwględny: ", Int - np.pi/4)

def trapez():

    length = 1
    h = 0.1
    xs = np.linspace(0, 1, num=11, endpoint=True)
    Int = sum(map(lambda x: (f(x)+f(x-h))*h/2, xs[1::]))

    print("wartość: ", Int, "bład bezwględny: ", Int - np.pi/4)

def simpson():
    length = 1
    h = 0.1
    xs = np.linspace(0, 1, num=6, endpoint=True)
    Int = sum(map(lambda x: (f(x)+4*f(x-h)+f(x-2*h))*(h/3), xs[1::]))

    print("wartość: ", Int, "bład bezwględny: ", Int - np.pi/4)
```

Otrzymuję następujące wyniki (odpowiednio dla metod prostokątów, trapezów i Simpsona):

```
wartość:  0.7099814972267896 bład bezwględny:  -0.07541666617065867
---------
wartość:  0.7849814972267897 bład bezwględny:  -0.00041666617065860834
---------
wartość:  0.7853981534848038 bład bezwględny:  -9.912644483023314e-09
```

3 Bibliografia

1. http://home.agh.edu.pl/~funika/mownit/lab4/calkowanie.pdf
2. https://eduinf.waw.pl/inf/alg/004_int/0004.php