Лабораторна робота №11

*Числове інтегрування*

Виконала Гальчинська Софія, студентка ФІТ 2-8. Варіант 4.

[*https://github.com/GalchynskaSofiia/Numerical\_Methods\_Of\_Programming*](https://github.com/GalchynskaSofiia/Numerical_Methods_Of_Programming)

**Завдання:** з точністю до 0.0001 обчислитизначенняінтегралуметодом прямокутників за умови *п*=10 (3 методи)

from scipy import integrate

import numpy as np

from math import sqrt

eps = 0.001

a = 0.2

b = 1.2

n = 10

def f1(x):

return 1/sqrt(x\*\*2 + 1)

def left\_rec(f1,a,b,n):

h=(b-a)/n

sum=0

for i in range(0,n):

sum+=f1(a+i\*h)

return sum\*h

v,err = integrate.quad(f1,a,b)

print("left rectangle:",round (left\_rec(f1,a,b,n), 5))

def right\_rec(f1,a,b,n):

h=(b-a)/n

sum=0

for i in range(1,n+1):

sum+=f1(a+i\*h)

return sum\*h

print("right rectangle:",round (right\_rec(f1,a,b,n), 5))

def aver\_rec(f1,a,b,n):

h=0.1

sum=0

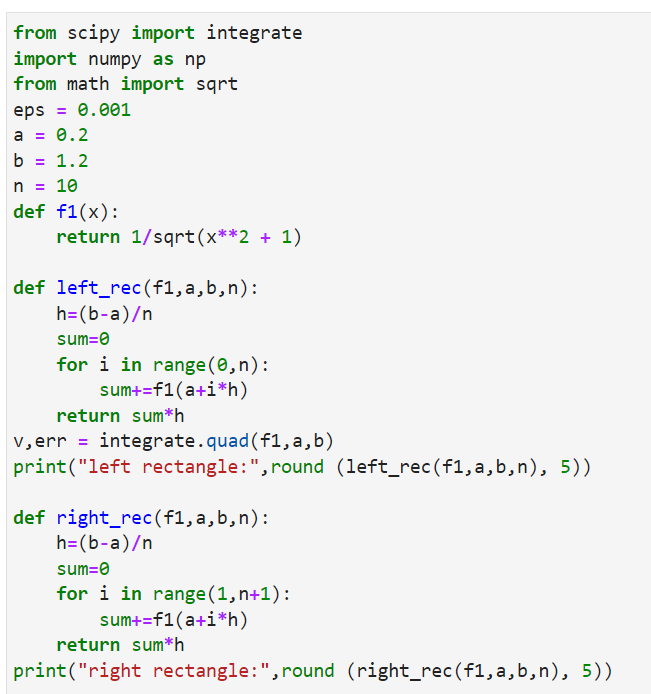
for i in range(0,n):

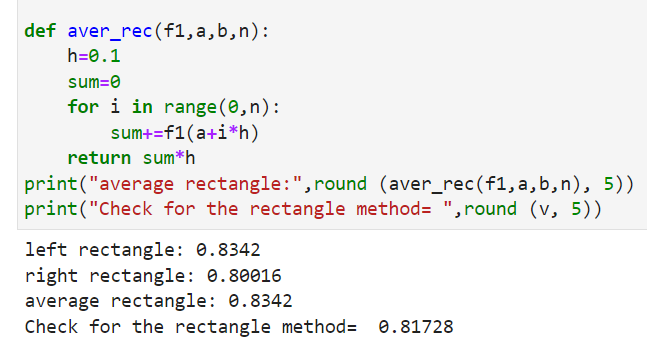
sum+=f1(a+i\*h)

return sum\*h

print("average rectangle:",round (aver\_rec(f1,a,b,n), 5))

print("Check for the rectangle method= ",round (v, 5))





**Завдання:** з точністю до 0.0001 обчислитизначенняінтегралуметодом Сімпсона за умови *п*=8

from scipy import integrate

import numpy as np

def f(x):

return np.cos(x)/(x+1)

a = 0.6

b = 1.4

n = 8

def simpson\_rule(f, a, b, n):

h = (b - a) / n

integr = f(a) + f(b)

for i in range(1,n):

k = a + i\*h

if i%2 == 0:

integr += 2 \* f(k)

else:

integr += 4 \* f(k)

integr \*= h/3

return integr

integral1 = simpson\_rule(f, a, b, n)

n \*= 2

integral2 = simpson\_rule(f, a, b, n)

while abs(integral2 - integral1) / 15 > 0.001:

integral1 = integral2

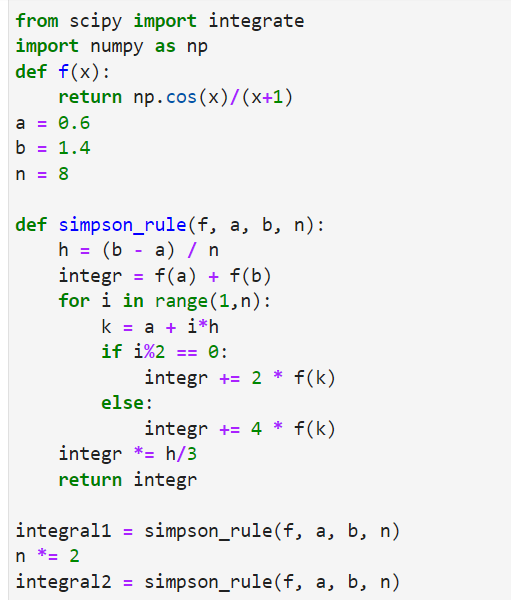
n \*= 2

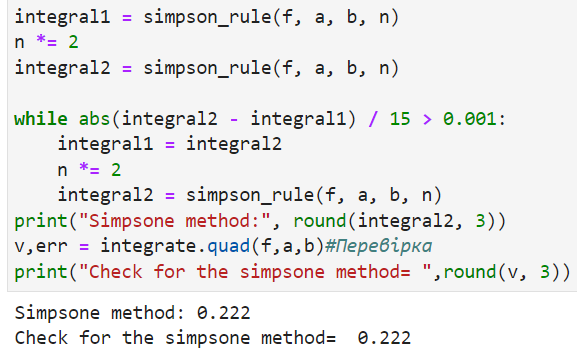
integral2 = simpson\_rule(f, a, b, n)

print("Simpsone method:", round(integral2, 3))

v,err = integrate.quad(f,a,b)#Перевірка

print("Check for the simpsone method= ",round(v, 3))



  
**Завдання:** з точністю до 0.0001 обчислитизначенняінтегралуметодом трапецій за умови *п*=20

from scipy import integrate

import numpy as np

from math import sqrt

def f(x):

return 1/sqrt(1.5 \* x\*\*2 + 0.7)

a = 1.4

b = 2.6

n = 20

def trapezoidal\_rule(f, a, b, n):

h = (b - a) / n

x = a

sum = 0

for i in range(1, n):

x += h

sum += 2 \* f(x)

sum += f(b)

integral = h / 2 \* sum

return integral

integral1 = trapezoidal\_rule(f, a, b, n)

n \*= 2

integral2 = trapezoidal\_rule(f, a, b, n)

while abs(integral2 - integral1) / 3 > 0.001:

integral1 = integral2

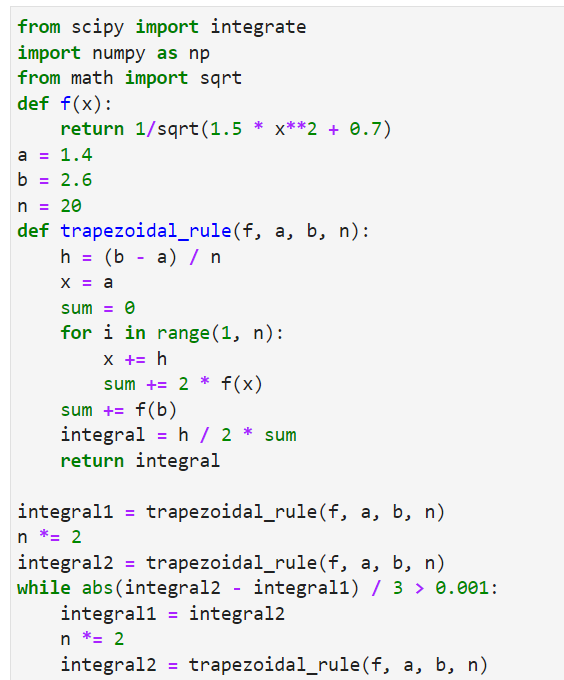
n \*= 2

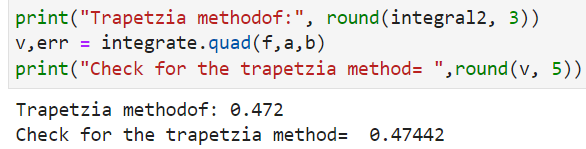
integral2 = trapezoidal\_rule(f, a, b, n)

print("Trapetzia methodof:", round(integral2, 3))

v,err = integrate.quad(f,a,b)

print("Check for the trapetzia method= ",round(v, 5))





**Висновки:** Отже, під час виконання практичної роботи детальніше ознайомились з різними методами обчислення визначених інтегралів. Було реалізовано код для обчислення інтегралу методом лівих, правих та середніх прямокутників. Потім проведено перевірку. Інші два інтеграли були обчислення іншими методами, зокрема метод Сімпсона та методом трапецій. Для них також були проведені відповідні перевірки. Результати перевірок співпадають з нашими обрахунками.