

ЛАБОРАТОРНА РОБОТА №11

Тема: «Числове інтегрування»

Трубчанінов Андрій Сергійович

ФІТ 2-8

В-29

Код:

```
# Обчислення інтеграла методом прямокутників
from scipy import integrate
import numpy as np

eps = 0.0001
a = 0.32
b = 0.66
n = 10

def f1(x):
    return 1 / np.sqrt(x + 2.5)

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) # Перевірка

# Перевірка точності за правилом Рунге:
if abs(left_rec(f1, a, b, 2 * 10) - left_rec(f1, a, b, 10)) / 3. <= eps:
    print("left rectangle:", round(left_rec(f1, a, b, 10), 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, 10), 5))

def aver_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

print("average rectangle:", round(aver_rec(f1, a, b, 10), 5))

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

```
left rectangle: 0.19727
right rectangle: 0.19615
average rectangle: 0.19727
Check for the rectangle method= 0.19671
```

```

# Обчислення інтеграла методом Сімпсона
from scipy import integrate
import numpy as np

# Задаємо функцію, яку необхідно інтегрувати
def f2(x):
    return np.log10(x**2 + 1) / (x + 1)

# Задаємо межі інтегрування та початкову кількість розбиттів
a = 0.8
b = 1.6
n = 8

# Обчислюємо значення інтегралу методом Сімпсона
def simpson_rule(f2, a, b, n):
    h = (b - a) / n
    integr = f2(a) + f2(b)
    for i in range(1, n):
        k = a + i * h
        if i % 2 == 0:
            integr += 2 * f2(k)
        else:
            integr += 4 * f2(k)
    integr *= h / 3
    return integr

# Обчислюємо значення інтегралу методом Сімпсона з точністю 0.001
integral1 = simpson_rule(f2, a, b, n)
n *= 2
integral2 = simpson_rule(f2, a, b, n)
while abs(integral2 - integral1) / 15 > 0.001:
    integral1 = integral2
    n *= 2
    integral2 = simpson_rule(f2, a, b, n)

# Виводимо результат
print("Simpsons method:", round(integral2, 5))

v, err = integrate.quad(f2, a, b) # Перевірка
print("Check for the Simpsons method= ", round(v, 5))

```

Simpsons method: 0.13808
 Check for the Simpsons method= 0.13808

```

# Обчислення інтеграла методом трапеції
from scipy import integrate
import numpy as np

eps = 0.0001
a = 1.2
b = 2.7
n = 20

def f3(x):
    return 1 / np.sqrt(x**2 + 3.2)

def trapezoidal_rule(f3, a, b, n):
    h = (b - a) / n
    x = a
    sum = 0
    for i in range(1, n):
        x += h
        sum += 2 * f3(x)
    sum += f3(b)
    integral = h / 2 * sum

    return integral

integral1 = trapezoidal_rule(f3, a, b, n)
n *= 2
integral2 = trapezoidal_rule(f3, a, b, n)
while abs(integral2 - integral1) / 3 > 0.001:
    integral1 = integral2
    n *= 2
    integral2 = trapezoidal_rule(f3, a, b, n)

# Виводимо результат
print("Trapezoid method:", round(integral2, 5))

v, err = integrate.quad(f3, a, b) # Перевірка
print("Check for the trapezoid method= ", round(v, 5))

```

Trapezoid method: 0.56916
 Check for the trapezoid method= 0.57134

Код зі скрінів:

Обчислення інтеграла методом прямокутників

from scipy **import** integrate

import numpy **as** np

eps = 0.0001

a = 0.32

b = 0.66

n = 10

def f1(x):

return 1 / np.sqrt(x + 2.5)

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) *# Перевірка*

Перевірка точності за правилом Рунге:

if abs(left_rec(f1, a, b, 2 * 10) - left_rec(f1, a, b, 10)) / 3. <= eps:

 print("left rectangle:", round(left_rec(f1, a, b, 10), 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, 10), 5))
```

```
def aver_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
```

```
print("average rectangle:", round(aver_rec(f1, a, b, 10), 5))
```

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

```
left rectangle: 0.19727
```

```
right rectangle: 0.19615
```

```
average rectangle: 0.19727
```

```
Check for the rectangle method= 0.19671
```

```
# Обчислення інтеграла методом Сімпсона
```

```
from scipy import integrate
```

```
import numpy as np
```

```
# Задаємо функцію, яку необхідно інтегрувати
```

```
def f2(x):
```

```
    return np.log10(x**2 + 1) / (x + 1)
```

```
# Задаємо межі інтегрування та початкову кількість розбиттів
```

```
a = 0.8
```

```
b = 1.6
```

```
n = 8
```

```
# Обчислюємо значення інтегралу методом Сімпсона
```

```
def simpson_rule(f2, a, b, n):
```

```
    h = (b - a) / n
```

```
    integr = f2(a) + f2(b)
```

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

```
        k = a + i * h
```

```
        if i % 2 == 0:
```

```
            integr += 2 * f2(k)
```

```
        else:
```

```
            integr += 4 * f2(k)
```

```
    integr *= h / 3
```

```
    return integr
```

```
# Обчислюємо значення інтегралу методом Сімпсона з точністю 0.001
```

```
integral1 = simpson_rule(f2, a, b, n)
```

```

n *= 2
integral2 = simpson_rule(f2, a, b, n)
while abs(integral2 - integral1) / 15 > 0.001:
    integral1 = integral2
    n *= 2
    integral2 = simpson_rule(f2, a, b, n)

# Виводимо результат
print("Simpsons method:", round(integral2, 5))

v, err = integrate.quad(f2, a, b) # Перевірка
print("Check for the Simpsons method= ", round(v, 5))

# Обчислення інтеграла методом трапецій
from scipy import integrate
import numpy as np

eps = 0.0001
a = 1.2
b = 2.7
n = 20

def f3(x):
    return 1 / np.sqrt(x**2 + 3.2)

def trapezoidal_rule(f3, a, b, n):

```

```
h = (b - a) / n
x = a
sum = 0
for i in range(1, n):
    x += h
    sum += 2 * f3(x)
sum += f3(b)
integral = h / 2 * sum

return integral
```

```
integral1 = trapezoidal_rule(f3, a, b, n)
n *= 2
integral2 = trapezoidal_rule(f3, a, b, n)
while abs(integral2 - integral1) / 3 > 0.001:
    integral1 = integral2
    n *= 2
    integral2 = trapezoidal_rule(f3, a, b, n)
```

```
# Виводимо результат
```

```
print("Trapetzia methodof:", round(integral2, 5))
```

```
v, err = integrate.quad(f3, a, b) # Перевірка
```

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

```
GitHub: https://github.com/Rey-ui/chiselni-metody
```

