

**Міністерство освіти і науки України**  
**Національний університет «Запорізька Політехніка»**

Кафедра програмних засобів

**ЗВІТ**

з самостійної роботи №1

з дисципліни «Методи Оптимізації та Дослідження Операцій» на тему:  
«Огляд розділу `scipy.integrate` під назвою "Інтегрування функцій за  
фіксованими вибірками"»

**Виконав**

Студент групи КНТ-122

О. А. Онищенко

**Прийняли**

Викладач

Л. Ю. Дейнега

2024

## Зміст звіту

Огляд розділу <code>scipy.integrate</code> під назвою "Інтегрування функцій за фіксованими вибірками" .....	3
Мета роботи .....	3
Постановка задачі.....	3
<code>scipy.integrate.trapezoid</code> .....	3
Опис .....	4
Параметри.....	4
Повернення.....	4
Приклади .....	5
<code>scipy.integrate.cumulative_trapezoid</code> .....	6
Опис .....	7
Параметри.....	7
Повернення.....	7
Приклади .....	7
<code>scipy.integrate.simpson</code> .....	8
Опис .....	8
Параметри.....	9
Повернення.....	9
Приклади .....	9
<code>scipy.integrate.cumulative_simpson</code> .....	10
Опис .....	10
Параметри.....	10
Повернення.....	11
Приклади .....	11
<code>scipy.integrate.romb</code> .....	13
Опис .....	13
Параметри.....	13
Повернення.....	14
Приклади .....	14
Висновки .....	15

# **ОГЛЯД РОЗДІЛУ SCIPY.INTEGRATE ПІД НАЗВОЮ "ІНТЕГРУВАННЯ ФУНКЦІЙ ЗА ФІКСОВАНИМИ ВИБІРКАМИ"**

## **Мета роботи**

Метою роботи є детальний огляд розділу документації до бібліотеки scipy підрозділу integrate під назвою "Інтегрування функцій за фіксованими вибірками".

## **Постановка задачі**

Постановка задачі на поточну самостійну роботу з університетської дисципліни під назвою Методи Оптимізації та Дослідження Операцій є наступні підпункти:

- розглянути кожний із запропонованих методів з розділу технічної документації до бібліотеки scipy розділу integrate підрозділу "Інтегрування функцій за фіксованими вибірками";

- для кожного методу визначити наступні пункти:
  - загальна інформацію про метод - опис,
  - параметри методу - аргументи, що він приймає,
  - повертаємі значення методу - значення що повертаються (return values)

## **scipy.integrate.trapezoid**

```
scipy.integrate.trapezoid(y, x=None, dx=1.0, axis=-1)
```

## Опис

Проводить інтегрування вздовж заданої осі, використовуючи правило складеної трапеції.

Якщо задано  $x$ , то інтегрування відбувається послідовно вздовж його елементів - вони не сортуються.

Інтегруємо  $y(x)$  вздовж кожного 1d відрізка на заданій осі, обчислюємо  $\int y(x)dx$ . Коли  $x$  задано, інтегрування відбувається вздовж параметричної кривої, обчислюючи  $\int_t y(t)dt = \int_t y(t) \frac{dx}{dt} |_{x=x(t)} dt$ .

## Параметри

Назва	Тип	Опис
$x$	array_like	Вхідний масив для інтегрування.
$x$	array_like, опціональний	Точки вибірки, що відповідають значенням $y$ . Якщо $x$ дорівнює None, точки вибірки вважаються рівномірно розташованими на відстані $dx$ одна від одної. За замовчуванням приймається значення None.
$dx$	scalar, опціональний	Відстань між точками вибірки, коли $x$ дорівнює None. Значення за замовчуванням дорівнює 1.0.
axis	int, опціональний	Вісь, вздовж якої інтегрувати.

## Повернення

Назва	Тип	Опис
trapezoid	float або ndarray	Визначений інтеграл від $y = n$ -вимірного масиву, апроксимованого вздовж однієї осі за правилом трапеції. Якщо $y$ - одновимірний масив, то результатом є число з плаваючою

		крапкою. Якщо $n$ більше за 1, то результатом є $n-1$ -вимірний масив.
--	--	--

## Приклади

Імпортуємо необхідні бібліотеки для всіх розглянутих прикладів:

```
import numpy as np
from scipy import integrate
from rich.console import Console
from rich.traceback import install
install()
console = Console()
```

Використаємо правило трапеції на рівномірно розташованих точках:

```
console.print(integrate.trapezoid([1, 2, 3]))
4.0
```

Відстань між точками вибірки можна вибрати за допомогою аргументів  $x$  або  $dx$ :

```
console.print(integrate.trapezoid([1, 2, 3], x=[4, 6, 8]))
8.0
console.print(integrate.trapezoid([1, 2, 3], dx=2))
8.0
```

Використаємо спадний  $x$ , що відповідає інтегруванню у зворотному напрямку:

```
console.print(integrate.trapezoid([1, 2, 3], x=[8, 6, 4]))
-8.0
```

У ширшому розумінні  $x$  використовується для інтегрування вздовж параметричної кривої. Ми можемо наближено обчислити інтеграл  $\int_0^1 x^2 = \frac{1}{3}$  за допомогою:

```
x = np.linspace(0, 1, num=50)
y = x**2
console.print(integrate.trapezoid(y, x))
0.33340274885464394
```

Або наближено обчислити площу кола, враховуючи, що ми повторюємо зразок, який замикає криву:

```
theta = np.linspace(0, 2 * np.pi, num=1000, endpoint=True)
console.print(integrate.trapezoid(np.cos(theta), x=np.sin(theta)))
3.141571941375841
```

*trapezoid* можна застосувати вздовж заданої осі, аби виконати декілька обчислень за один виклик:

```
a = np.arange(6).reshape(2, 3)
console.print(a)
console.print(integrate.trapezoid(a, axis=0))
console.print(integrate.trapezoid(a, axis=1))
[[0 1 2]
 [3 4 5]]
[1.5 2.5 3.5]
[2. 8.]
```

### **scipy.integrate.cumulative\_trapezoid**

```
scipy.integrate.cumulative_trapezoid(y, x=None, dx=1.0, axis=-1,
initial=None)
```

## Опис

Кумулятивно інтегрує  $y(x)$  за правилом складеної трапеції.

## Параметри

Назва	Тип	Опис
y	array_like	Значення для інтегрування.
x	array_like, опціональний	Координата, вздовж якої будемо інтегрувати. Якщо значення None (за замовчуванням), використовуємо відстань dx між послідовними елементами в y.
dx	float, опціональний	Відстань між елементами y. Використовується тільки якщо x дорівнює None.
axis	int, опціональний	Визначає вісь для накопичення (кумуляції). За замовчуванням -1 (остання вісь).
initial	scalar, опціональний	Якщо задано, додамо його на початку результату, що повертається. Єдиними допустимими значеннями є 0 або None. За замовчуванням - None, що означає, що res має на один елемент менше, ніж y вздовж осі інтегрування.

## Повернення

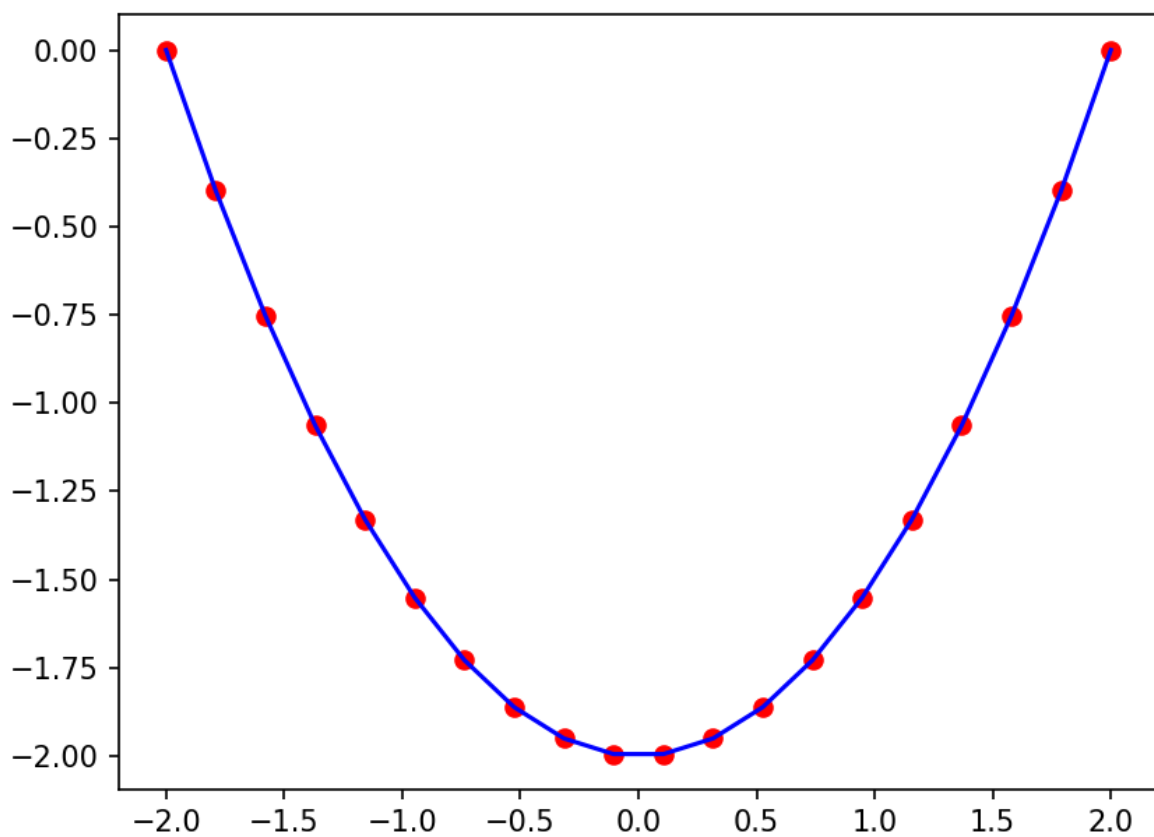
Назва	Тип	Опис
res	ndarray	Результат кумулятивного інтегрування y вздовж осі. Якщо initial дорівнює None, форма така, що вісь інтегрування має на одне значення менше, ніж y. Якщо initial задано, то форма дорівнює формі y.

## Приклади

Імпортуємо необхідні бібліотеки для всіх розглянутих прикладів:

```
from scipy import integrate
import numpy as np
import matplotlib.pyplot as plt
```

```
x = np.linspace(-2, 2, num=20)
y = x
y_int = integrate.cumulative_trapezoid(y, x, initial=0)
plt.plot(x, y_int, "ro", x, y[0] + 0.5 * x**2, "b-")
plt.show()
```



**scipy.integrate.simpson**

```
scipy.integrate.simpson(y, *, x=None, dx=1.0, axis=-1)
```

**Опис**



Інтегруємо  $y(x)$ , використовуючи вибірки вздовж заданої осі та комбіноване правило Сімпсона. Якщо  $x$  дорівнює `None`, то вважаємо, що інтервал між відліками дорівнює  $dx$ .

Якщо є парна кількість вибірок,  $N$ , то є непарна кількість інтервалів  $(N-1)$ , але правило Сімпсона вимагає парної кількості інтервалів. Параметр `'even'` контролює, як це вирішується.

## Параметри

Назва	Тип	Опис
<code>y</code>	<code>array_like</code>	Масив для інтегрування.
<code>x</code>	<code>array_like</code> , опціональний	Якщо задано, то то є точки, в яких відбувається вибірка <code>y</code> .
<code>dx</code>	<code>float</code> , опціональний	Відстань між точками інтегрування вздовж осі <code>x</code> . Використовується лише тоді, коли <code>x</code> дорівнює <code>None</code> . За замовчуванням 1.
<code>axis</code>	<code>int</code> , опціональний	Вісь, вздовж якої інтегрувати. За замовчуванням - остання вісь.

## Повернення

Назва	Тип	Опис
<code>res</code>	<code>float</code>	Обчислений інтеграл за складеним правилом Сімпсона.

## Приклади

Імпортуємо необхідні бібліотеки для всіх розглянутих прикладів:

```
from rich.console import Console
from rich.traceback import install
install()
```

```

console = Console()
from scipy import integrate
import numpy as np

x = np.arange(0, 10)
y = np.arange(0, 10)
console.print(integrate.simpson(y, x=x))
40.5

y = np.power(x, 3)
console.print(integrate.simpson(y, x=x))
console.print(integrate.quad(lambda x: x**3, 0, 9)[0])
1640.5
1640.25

```

## scipy.integrate.cumulative\_simpson

```

scipy.integrate.cumulative_simpson(y, *, x=None, dx=1.0, axis=-1,
initial=None)

```

## Опис

Кумулятивно інтегруємо  $y(x)$  за допомогою складеного правила  $1/3$  Сімпсона. Інтеграл вибірок у кожній точці обчислюється, припускаючи квадратичну залежність між кожною точкою та двома сусідніми точками.

## Параметри

Назва	Тип	Опис
y	array_like	Значення для інтегрування. Вимагає наявності принаймні однієї точки вздовж осі. Якщо надано дві або менше точок вздовж осі, інтегрування Сімпсона неможливе і результат обчислюється за допомогою cumulative_trapezoid.
x	array_like, опціональний	Координата, вздовж якої інтегрувати. Повинна мати ту саму форму, що й y, або бути 1D з тією самою довжиною,

		що й у вздовж осі. x також має бути суворо зростаючою вздовж осі. Якщо x дорівнює None (за замовчуванням), інтегрування виконується з використанням відстані dx між послідовними елементами в у.
dx	scalar або array_like, опціональний	Відстань між елементами у. Використовується тільки якщо x дорівнює None. Може бути або float, або масивом тієї ж форми, що і у, але довжиною в одиницю вздовж осі. За замовчуванням 1.0.
axis	int, опціональний	Визначає вісь, вздовж якої потрібно інтегрувати. За замовчуванням -1 (остання вісь).
initial	scalar або array_like, опціональний	Якщо задано, додаємо це значення на початок результату, що повертається, і додаємо його до решти результату. За замовчуванням - None, що означає, що значення за x[0] не повертається, а res має на один елемент менше, ніж у вздовж осі інтегрування. Може бути або float, або масивом тієї ж форми, що і у, але довжиною в одиницю вздовж осі.

## Повернення

Назва	Тип	Опис
res	ndarray	Результат кумулятивного інтегрування у вздовж осі. Якщо initial дорівнює None, форма така, що вісь інтегрування має на одне значення менше, ніж у. Якщо initial задано, то форма дорівнює формі у.

## Приклади

Імпортуємо необхідні бібліотеки для всіх розглянутих прикладів:

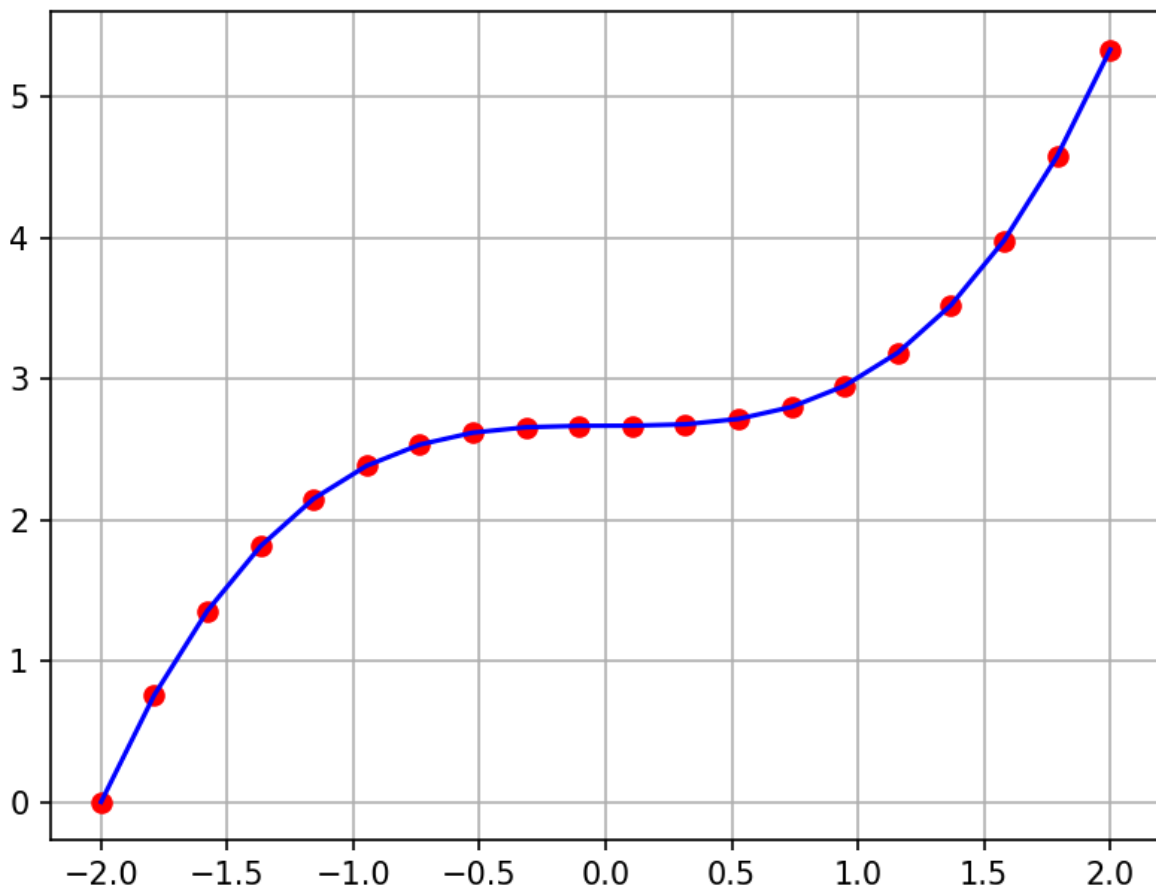
```
from rich.console import Console
from rich.traceback import install
install()
```

```

console = Console()
from scipy import integrate
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-2, 2, num=20)
y = x**2
y_int = integrate.cumulative_simpson(y, x=x, initial=0)
fig, ax = plt.subplots()
ax.plot(x, y_int, 'ro', x, x**3/3 - (x[0])**3/3, 'b-')
ax.grid()
plt.show()

```



Результат роботи *cumulative\_simpson* подібний до результату ітеративного виклику *simpson* з послідовно вищими верхніми межами інтегрування, але не ідентичний.

```

def cumulative_simpson_reference(y, x):

```

```

    return np.asarray([integrate.simpson(y[:i], x=x[:i]) for i in
range(2, len(y) + 1)])
rng = np.random.default_rng()
x, y = rng.random(size=(2, 10))
x.sort()
res = integrate.cumulative_simpson(y, x=x)
ref = cumulative_simpson_reference(y, x)
equal = np.abs(res - ref) < 1e-15
console.print(equal)
[False True False True False True False True True]

```

Це очікувано: оскільки *cumulative\_simpson* має доступ до більшої кількості інформації, ніж *simpson*, він, як правило, може дати точніші оцінки базового інтеграла на підінтервалах.

## scipy.integrate.romb

```
scipy.integrate.romb(y, dx=1.0, axis=-1, show=False)
```

## Опис

Інтегрування Ромберга за допомогою вибірки з функції.

## Параметри

Назва	Тип	Опис
y	array_like	Вектор $2**k + 1$ рівновіддалених вибірок з функції.
dx	float, опціональний	Інтервал між вибірками. За замовчуванням 1.
axis	int, опціональний	Вісь, вздовж якої інтегрувати. За замовчуванням -1 (остання вісь).
show	bool, опціональний	Якщо у є одновимірним масивом, то якщо цей параметр має значення True, виводимо таблицю, яка показує

		екстраполяцію Річардсона за вибірками. За замовчуванням False.
--	--	--

## Повернення

Назва	Тип	Опис
romb	ndarray	Проінтегрований результат для осі.

## Приклади

Імпортуємо необхідні бібліотеки для всіх розглянутих прикладів:

```
from rich.console import Console
from rich.traceback import install
install()
console = Console()
from scipy import integrate
import numpy as np
```

```
x = np.arange(10, 14.25, 0.25)
y = np.arange(3, 12)
console.print(integrate.romb(y))
56.0
```

```
y = np.sin(np.power(x, 2.5))
console.print(integrate.romb(y))
-0.7425613366722288
```

```
console.print(integrate.romb(y, show=True))
Richardson Extrapolation Table for Romberg Integration
=====
-0.81576
 4.63862  6.45674
-1.10581 -3.02062 -3.65245
-2.57379 -3.06311 -3.06595 -3.05664
-1.34093 -0.92997 -0.78776 -0.75160 -0.74256
=====
-0.7425613366722288
```

## Висновки

Таким чином, ми детально оглянули розділ документації до бібліотеки `scipy` підрозділу `integrate` під назвою "Інтегрування функцій за фіксованими вибірками".

*Бо так полюбив Бог світ, що віддав Сина Свого  
Однородженого (Ісуса), аби кожен, хто вірує в Нього, не  
загинув, але мав життя вічне. ([Йоан 3:16](#))*