**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 scipy.integrate під назвою "Інтегрування функцій за фіксованими вибірками" 3](#_Toc167463780)

[Мета роботи 3](#_Toc167463781)

[Постановка задачі 3](#_Toc167463782)

[scipy.integrate.trapezoid 3](#_Toc167463783)

[Опис 4](#_Toc167463784)

[Параметри 4](#_Toc167463785)

[Повернення 4](#_Toc167463786)

[Приклади 5](#_Toc167463787)

[scipy.integrate.cumulative\_trapezoid 6](#_Toc167463788)

[Опис 7](#_Toc167463789)

[Параметри 7](#_Toc167463790)

[Повернення 7](#_Toc167463791)

[Приклади 7](#_Toc167463792)

[scipy.integrate.simpson 8](#_Toc167463793)

[Опис 8](#_Toc167463794)

[Параметри 9](#_Toc167463795)

[Повернення 9](#_Toc167463796)

[Приклади 9](#_Toc167463797)

[scipy.integrate.cumulative\_simpson 10](#_Toc167463798)

[Опис 10](#_Toc167463799)

[Параметри 10](#_Toc167463800)

[Повернення 11](#_Toc167463801)

[Приклади 11](#_Toc167463802)

[scipy.integrate.romb 13](#_Toc167463803)

[Опис 13](#_Toc167463804)

[Параметри 13](#_Toc167463805)

[Повернення 14](#_Toc167463806)

[Приклади 14](#_Toc167463807)

[Висновки 15](#_Toc167463808)

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 , то інтегрування відбувається послідовно вздовж його елементів - вони не сортуються.

Інтегруємо y(x) вздовж кожного 1d відрізка на заданій осі, обчислюємо . Коли x задано, інтегрування відбувається вздовж параметричної кривої, обчислюючи .

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]))

console.print(integrate.trapezoid([1, 2, 3], dx=2))

8.0

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 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 у(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

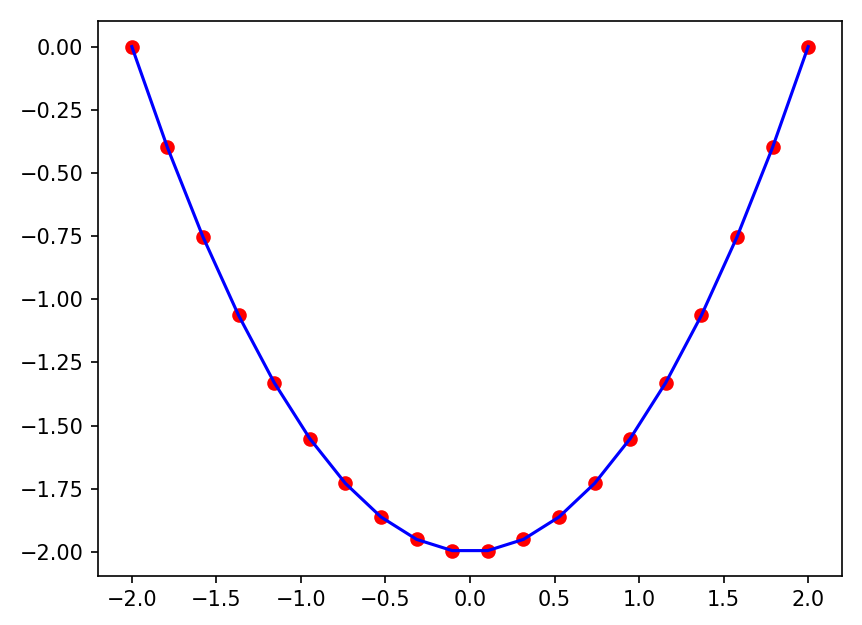
x = np.linspace(-2, 2, num=20)

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** |
| y | array\_like | Масив для інтегрування. |
| x | array\_like, опціональний | Якщо задано, то то є точки, в яких відбувається вибірка y. |
| dx | float, опціональний | Відстань між точками інтегрування вздовж осі x. Використовується лише тоді, коли x дорівнює None. За замовчуванням 1. |
| axis | int, опціональний | Вісь, вздовж якої інтегрувати. За замовчуванням - остання вісь. |

Повернення

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип** | **Опис** |
| res | float | 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 y 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 y. |
| dx | scalar або array\_like, опціональний | Відстань між елементами y. Використовується тільки якщо x дорівнює None. Може бути або float, або масивом тієї ж форми, що і y, 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 y вздовж осі інтегрування. Може бути або float, або масивом тієї ж форми, що і y, але довжиною в одиницю вздовж осі. |

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 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

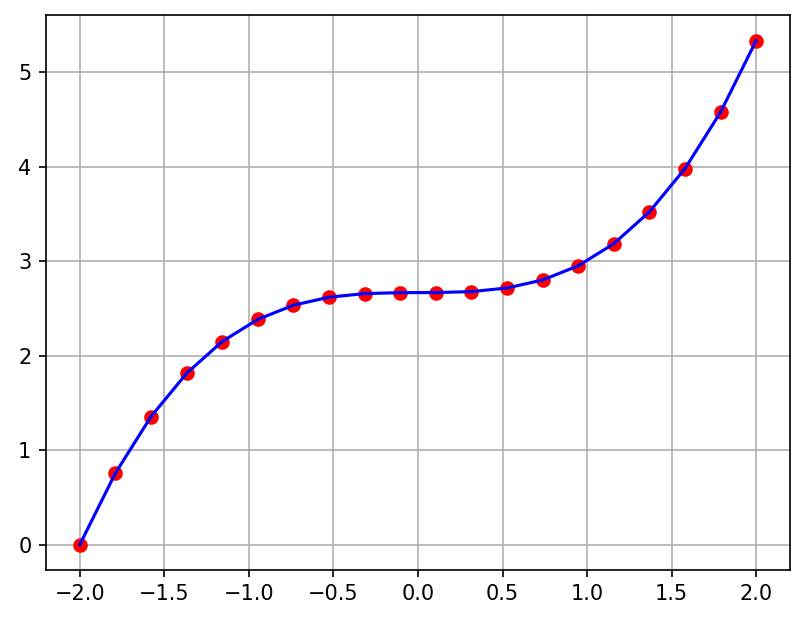
y\_int = integrate.cumulative\_simpson(y, x=x, initial=0)

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 y 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](https://www.bible.com/uk/bible/compare/JHN.3.16))