**Міністерство освіти і науки України**

**Національний університет «Запорізька політехніка»**

Кафедра програмних засобів

**ЗВІТ**

Дисципліна «Розробка прикладних програм»

Робота №2

Тема «Використання бібліотек Python для високопродуктивних наукових обчислень»

**Виконав варіант 19**

Студент КНТ-122 Онищенко О. А.

**Прийняли**

Викладач Дейнега Л. Ю.

2024

[Мета роботи 3](#_Toc180506223)

[Індивідуальне завдання 3](#_Toc180506224)

[Тексти файлів 3](#_Toc180506225)

[run.py 3](#_Toc180506226)

[Результати виконання 4](#_Toc180506227)

[Контрольні питання 5](#_Toc180506228)

[Засоби бібліотеки numpy 5](#_Toc180506229)

[Засоби бібліотеки scipy 6](#_Toc180506230)

[Призначення бібліотеки matplotlib 6](#_Toc180506231)

[Типи графіків через matplotlib 7](#_Toc180506232)

[Налаштування графіку matplotlib 12](#_Toc180506233)

[Побудова кількох графіків у одному вікні matplotlib 15](#_Toc180506234)

Мета роботи

Ознайомитися з основними можливостями бібліотек numpu та scipy і навчитися використовувати їх на практиці для виконання високопродуктивних наукових обчислень.

Індивідуальне завдання

1. Обчислити визначений інтеграл

2. Побудувати графік фукнції, знайти її мінімум методом Нелдера-Міда з точністю 10

Тексти файлів

run.py

import numpy as np

import scipy as sp

from matplotlib import pyplot as plt

def one():

'''

Обчислити визначений інтеграл

'''

PI=np.pi

bounds={

'lower':PI/4,

'upper':PI/7

}

target\_function=lambda x:(np.cos(2\*x)+np.sin(x)\*\*2)/np.sin(3\*x)

solution=sp.integrate.quad(target\_function,bounds['lower'],bounds['upper'])

print(solution)

def two():

'''

Побудувати графік функції

'''

target\_function=lambda x,y:x\*\*2+y\*\*2-4\*x-y-x\*y

b=np.arange(-3,3,0.1)

d=np.arange(-3,3,0.1)

B,D=np.meshgrid(b,d)

F=target\_function(B,D)

fig=plt.figure()

ax=fig.add\_subplot(111,projection='3d')

ax.plot\_surface(B,D,F)

plt.show()

'''

Знайти її мінімум методом Нелдера-Міда: точність 10

'''

def f(params):

x,y=params

return x\*\*2+y\*\*2-4\*x-y-x\*y

min\_value=sp.optimize.minimize(f,[3,3],method='Nelder-Mead')

print([float(f'{val:.10f}') for val in min\_value.x.tolist()])

one()

two()

Результати виконання

Вивід у консоль:

(-0.24129889066690466, 2.6789558423490586e-15)

[2.999986397, 2.0000283497]

Зображення, що містить схема, текст, знімок екрана, ряд

Автоматично згенерований опис

Рисунок 1.1 – Графік функції

Контрольні питання

Слава ІСУСУ ХРИСТУ

Засоби бібліотеки numpy

Бібліотека numpy дає засоби роботи з масивами: тип даних ndarray. Аби використати бібліотеку потрібно встановити її: у терміналі

pip install numpy

Далі імпортувати її: у .py файлі

import numpy as np

Аби використати масив можна передати методу array список:

a=np.array([3,7,12,40])

print(a)

> [ 3 7 12 40]

Аби взяти елемент масиву типу ndarray можна використати індекси:

print(a[-1])

> 40

Масиви типу ndarray також можна змінювати як і списки:

a[-1]=12

print(a)

> [ 3 7 12 12]

[Джерело](https://numpy.org/devdocs/user/absolute_beginners.html#array-fundamentals)

Засоби бібліотеки scipy

Спеціальні функції (scipy.special)

Інтеграція (scipy.integrate)

Оптимізація (scipy.optimize)

Інтерполяція (scipy.interpolate)

Перетворення Фур'є (scipy.fft)

Обробка сигналів (scipy.signal)

Лінійна алгебра (scipy.linalg)

Розріджені масиви (scipy.sparse)

Розріджені задачі на власні значення з ARPACK

Процедури стиснення розріджених графів (scipy.sparse.csgraph)

Просторові структури даних та алгоритми (scipy.spatial)

Статистика (scipy.stats)

Багатовимірна обробка зображень (scipy.ndimage)

Файловий ввід/вивід (scipy.io)

[Джерело](https://docs.scipy.org/doc/scipy/tutorial/index.html#:~:text=organized%20by%20subpackages.-,User%20guide,-Special%20functions%20()

Призначення бібліотеки matplotlib

Надання інструментів графічного подання даних для легшого аналізу та розуміння

[Джерело](https://www.geeksforgeeks.org/python-introduction-matplotlib/#:~:text=Matplotlib%E2%80%99s%20primary%20purpose%20is%20to%20provide%20users%20with%20the%20tools%20and%20functionality%20to%20represent%20data%20graphically%2C%20making%20it%20easier%20to%20analyze%20and%20understand)

Типи графіків через matplotlib

1. [Звичайний графік](https://www.w3schools.com/python/img_matplotlib_plotting1.png)

Зображення, що містить ряд, Графік, схема, схил

Автоматично згенерований опис

Рисунок 2.1 – Вигляд звичайного графіку

1. [Підграфік](https://www.w3schools.com/python/img_matplotlib_subplots1.png)

Зображення, що містить ряд, схема, Графік, Паралель

Автоматично згенерований опис

Рисунок 2.2 – Вигляд підграфіку

1. [Стовпчаста діаграма](https://www.w3schools.com/python/img_matplotlib_bars1.png)

Зображення, що містить Прямокутник, схема, знімок екрана, текст

Автоматично згенерований опис

Рисунок 2.3 – Вигляд стовпчастої діаграми

1. [Горизонтальна стовпчаста діагарма](https://www.w3schools.com/python/img_matplotlib_bars2.png)

Зображення, що містить знімок екрана, Прямокутник, ряд, текст

Автоматично згенерований опис

Рисунок 2.4 – Вигляд горизонтальної стовпчастої діаграми

1. [Гістограма](https://www.w3schools.com/python/img_matplotlib_histogram1.png)

Зображення, що містить схема, Графік, піксель, дизайн

Автоматично згенерований опис

Рисунок 2.5 – Вигляд гістограми

1. [Кругова діаграма](https://www.w3schools.com/python/img_matplotlib_pie_explode.png)

Зображення, що містить схема, Графіка, знімок екрана, коло

Автоматично згенерований опис

Рисунок 2.6 – Вигляд кругової діаграми

Налаштування графіку matplotlib

Метод matplotlib.pyplot.plot приймає параметри значень x та y.

Якщо значення x не зазначено, будуть використані стандартні: діапазон з кроком 1.

Третім аргументом можна зазначити marker: змінює маркер кожної точки графіку:

plt.plot(ypoints, marker = 'o')

Зображення, що містить ряд, схема, Графік, схил

Автоматично згенерований опис

Рисунок 2.7 – Вигляд маркеру o

[Різні типи маркерів](https://www.w3schools.com/python/matplotlib_markers.asp#:~:text=any%20of%20these-,markers,-%3A).

Інший аргумент – fmt: змінює вигляд графіку у форматі маркер/лінія/колір:

plt.plot(ypoints, 'o:r')

Зображення, що містить ряд, схема, Графік

Автоматично згенерований опис

Рисунок 2.8 – Вигляд налаштованого графіку типу o:r

[Різні типи ліній](https://www.w3schools.com/python/matplotlib_markers.asp#:~:text=of%20the%20following%3A-,Line%20Reference,-Line%20Syntax).

[Різні значення кольорів](https://www.w3schools.com/python/matplotlib_markers.asp#:~:text=of%20the%20following%3A-,Color%20Reference,-Color%20Syntax).

Ще один аргумент – ms: змінює розмір маркера:

plt.plot(ypoints, marker = 'o', ms = 20)

Зображення, що містить ряд, схема, знімок екрана, Графік

Автоматично згенерований опис

Рисунок 2.9 – Вигляд графіку з розміром маркера 20

[Джерело](https://www.w3schools.com/python/matplotlib_markers.asp), [інше](https://www.w3schools.com/python/matplotlib_plotting.asp)

Побудова кількох графіків у одному вікні matplotlib

Для побудови підграфіку (або відображення кількох графіків у одному вікні) можна використати метод matplotlib.pyplot.subplot:

import matplotlib.pyplot as plt

import numpy as np

x = np.array([0, 1, 2, 3])

y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)

plt.plot(x,y)

x = np.array([0, 1, 2, 3])

y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)

plt.plot(x,y)

plt.show()

Зображення, що містить ряд, схема, Графік, Паралель

Автоматично згенерований опис

Рисунок 3.1 – Вигляд двох графіків у одному вікні

Метод subplot приймає три параметри:

1. Номер рядку на якому розташувати графік
2. Номер стовпця на якму розташувати графік
3. Номер графіку

plt.subplot(1, 2, 1) # графікбуде на 1 ряду у 2 стовпці і це 1-ий графік

[Джерело](https://www.w3schools.com/python/matplotlib_subplot.asp)