

## УДК 004.4

*Зимич А. П., здобувач 2 курсу спеціальності 122 Комп'ютерні науки,  
Фриз І. В., канд. фіз.-мат. наук, старший викладач кафедри інформаційних  
технологій*

### ОГЛЯД МАТЕМАТИЧНИХ БІБЛІОТЕК PYTHON

*Донецький національний університет імені Василя Стуса, м. Вінниця*

**Вступ.** У світі наук про дані та їх дослідження мова програмування Python займає одну із центральних позицій, надаючи різноманітні математичні бібліотеки для обробки, аналізу та візуалізації даних. Ці інструменти стали невід'ємною частиною роботи дослідників, аналітиків та розробників, надаючи їм зручні засоби для вирішення практичних завдань.

**Опис бібліотек.** Бібліотека NumPy (скорочено від Numerical Python) створена для ефективної роботи з багатовимірними масивами числових даних завдяки зусиллям Тревіса Оліфанта. Вона стала основою для наукових обчислень та аналізу даних у мові програмування Python. Особливостями NumPy [1, 2] є:

- NumPy надає ефективні структури даних для зберігання  $n$ -вимірних масивів даних та здійснення операцій над числовими даними;
- бібліотека містить великий набір математичних функцій, починаючи від простих операцій (наприклад, синус, косинус) і закінчуючи складними алгоритмами лінійної алгебри;
- може інтегруватися з кодом на низькорівневих мовах, як-от C та Fortran, що сприяє оптимізації продуктивності;
- завдяки оптимізованим операціям та підтримці багатопотоковості NumPy забезпечує високу продуктивність під час обробки великих обсягів даних;
- NumPy є проектом з відкритим кодом, що дає змогу користувачам адаптувати та розширювати його під свої потреби.

Бібліотека SciPy є відкритою бібліотекою для наукових обчислень у Python. Розроблялася Тревісом Оліфантом для розширення функціональності NumPy. SciPy надає високорівневі алгоритми для оптимізації, інтеграції, обробки сигналів, статистики та багатьох інших галузей, ставши ключовим компонентом екосистеми Python для наукових та інженерних розрахунків. Відмінності з NumPy та її особливості [3]:

- SciPy надає широкий набір функцій для чисельного інтегрування, розв'язування диференціальних рівнянь, оптимізації та інших математичних і наукових завдань;
- SciPy надає інструментарій алгоритмічної реалізації методів оптимізації, зокрема методи мінімізації функцій та глобальної оптимізації, що робить її

корисною для вирішення задач машинного навчання та інших оптимізаційних задач;

- бібліотека забезпечує засоби інтеграції з кодом, написаним низькорівневими мовами, що розширює її можливості та покращує продуктивність;

- бібліотека надає широкий спектр інструментарію для обробки та аналізу інформації на основі статистичних методів, зокрема тести для перевірки гіпотез, дисперсійний аналіз тощо;

- SciPy взаємодіє з NumPy, Matplotlib та іншими бібліотеками, створюючи сукупну екосистему для наукових обчислень та аналізу даних у Python;

- як і NumPy, SciPy є проєктом з відкритим кодом, що сприяє активній участі спільноти та її постійному розвитку.

SciPy розширює можливості NumPy, а деякі часті дії в ній реалізовані як окремі функції, тому бібліотека спрощує роботу зі складними завданнями з використанням просунутої математики. Для низки простіших завдань вона надмірна, буде достатньо NumPy чи інших бібліотек.

Matplotlib написана і підтримувалася в основному Джоном Хантером та призначена для створення візуалізацій на мові програмування Python. Згенеровані у різних форматах зображення можуть бути використані в інтерактивних графіках, наукових публікаціях, графічному інтерфейсі користувача, вебзастосунках тощо [4]. Основними можливостями Matplotlib є робота з різними видами графічного подання даних:

- графіки;
- діаграми розсіювання;
- стовпчасті діаграми та гістограми;
- кругові діаграми;
- діаграми стовбур-листя;
- контурні графіки;
- поля градієнтів;
- спектральні діаграми.

**Приклади використання бібліотек Python.** Далі розглянемо прості приклади застосування описаних бібліотек до розв'язування певних задач [5].

1. *Реалізація методу найменших квадратів для лінійних функцій.* Код, наведений у Лістингу 1, створює набір випадкових даних  $(x, y)$ , де  $y$  залежить від  $x$  лінійно з додаванням випадкового шуму. Потім він використовує метод найменших квадратів (least-squares) для знаходження коефіцієнтів лінійної регресії ( $\alpha$ ), які найкраще відповідають цим даним. Нарешті код будує графік вихідних даних та лінії регресії (рис. 1).

```
import numpy as np
import matplotlib.pyplot as plt
```

```

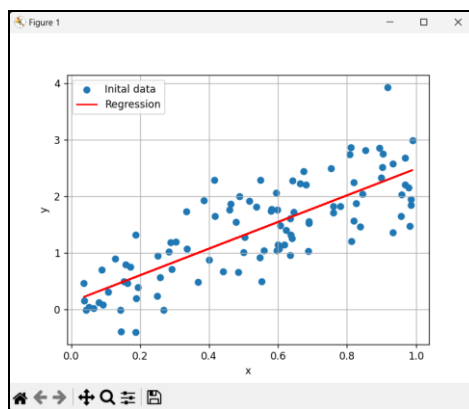
NUM_POINTS = 100
x = np.random.rand(NUM_POINTS)
y = 2.5 * x + 0.5 * np.random.randn(NUM_POINTS)
A = np.vstack([x, np.ones(len(x))]).T
alpha = np.linalg.lstsq(A, y, rcond=None)[0]

print("(alpha):", *alpha)

plt.scatter(x, y, label='Initial data')
plt.plot(x, alpha[0] * x + alpha[1], 'r', label='Regression')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.grid(True)
plt.show()

```

*Лістинг 1. Код програми наближення за методом найменших квадратів*



*Рисунок 1. Графічне зображення результату програми*

2. Приклад програми на знаходження екстремумів функції багатьох змінних. Програма з Лістингу 2 оптимізує функцію двох змінних методом Nelder-Mead. Код знаходить мінімум цієї функції, починаючи з початкового припущення [1, 1], а потім виводиться у вигляді тривимірного графіка (рис. 2).

```

import numpy as np
from scipy.optimize import minimize
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

def multivariable_function(x):
    return x[0]**2 + x[1]**2 + 15

initial_guess = [1, 1]

```

```

result = minimize(multivariable_function, initial_guess, method='Nelder-Mead')

minima_location = result.x
minima_value = result.fun

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

x = np.linspace(-2, 2, 100)
y = np.linspace(-2, 2, 100)
X, Y = np.meshgrid(x, y)
Z = multivariable_function([X, Y])

ax.plot_surface(X, Y, Z, alpha=0.5, cmap='viridis', edgecolors='w', linewidth=0.5,
label='Function')
ax.scatter(*minima_location, minima_value, color='red', marker='o', s=100,
label='Minimum')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

ax.legend()

ax.text2D(0, -0.05, f'Minimum Location: ({minima_location[0]:.4f};
{minima_location[1]:.4f})', transform=ax.transAxes)
ax.text2D(0, -0.1, f'Minimum Value: {minima_value:.4f}', transform=ax.transAxes)

plt.show()

```

Лістинг 2. Код програми для знаходження екстремумів функції

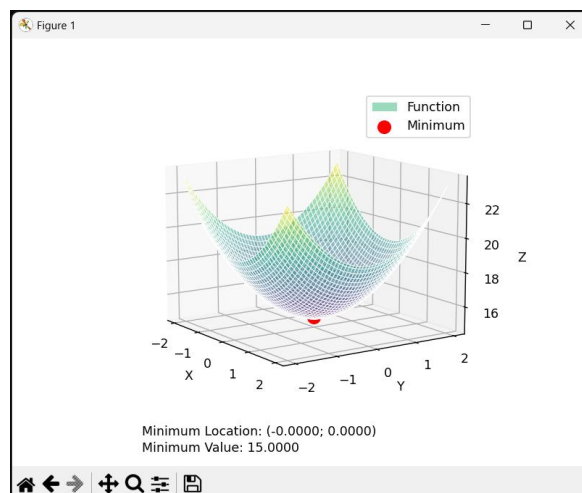


Рисунок 2. Графічне зображення результату програми

**Висновки.** NumPy виступає як основна бібліотека для обробки числових даних, надаючи ефективні структури для зберігання та операції над багатовимірними масивами. SciPy, яка базується на NumPy, доповнює його високорівневими алгоритмами для різних математичних та наукових задач, як-от оптимізація, обробка сигналів, статистика тощо. Matplotlib надає інструменти для візуалізації даних у різних форматах, що робить його потужним інструментом для графічного подання результатів аналізу.

### Список використаних джерел

1. NumPy. *Wikipedia*. URL: <https://en.wikipedia.org/wiki/NumPy> (дата звернення: 26.11.2023).
2. NumPy: the absolute basics for beginners. NumPy v1.26 Manual. URL: [https://numpy.org/doc/stable/user/absolute\\_beginners.html](https://numpy.org/doc/stable/user/absolute_beginners.html) (дата звернення: 26.11.2023).
3. SciPy. *Wikipedia*. URL: <https://en.wikipedia.org/wiki/SciPy> (дата звернення: 26.11.2023).
4. Matplotlib. *Wikipedia*. URL: <https://en.wikipedia.org/wiki/Matplotlib> (дата звернення: 26.11.2023).
5. Kong Q., Siau T., Bayen A. Python Programming And Numerical Methods: A Guide for Engineers and Scientists. URL: <https://pythonnumericalmethods.berkeley.edu/notebooks/Index.html> (дата звернення: 26.11.2023).

### УДК 004.021

*Лик В. В., здобувач 2 курсу спеціальності 122 Комп'ютерні науки,  
Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій*

## СТЕКИ ТА АЛГОРИТМИ З ВИКОРИСТАННЯМ СТЕКІВ

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Стеки та алгоритми з використанням стеків є ключовими компонентами у галузі інформатики та програмування. Стек, який працює за принципом Last In, First Out (LIFO), являє собою важливий інструмент для зберігання та управління даними. З розвитком хмарних технологій та збільшенням обчислювальних завдань ця тематика стає ще більш актуальною.

В обличчі складних завдань у сфері обчислень та зростання обсягів даних розробники стикаються з потребою оптимізації алгоритмів та раціонального використання ресурсів. Стеки відіграють важливу роль у вирішенні цих завдань, сприяючи підвищенню продуктивності та швидкості виконання програм.