

5. Лабораторна робота. Застосування бібліотеки Matplotlib до візуалізації функцій

Мета: засвоїти можливості роботи бібліотеки Matplotlib до візуалізації функцій у різних системах координат та візуалізації для аналізу даних.

Теоретичні відомості та методичні рекомендації

Бібліотека Matplotlib – це одна з найпопулярніших бібліотек для візуалізації даних у мові Python. Matplotlib – це бібліотека для візуалізації даних двовимірною та тривимірною графікою. Matplotlib є гнучким, легко конфігурованим пакетом, який разом з NumPy, SciPy і IPython надає можливості, подібні до MatLab.

Основні можливості Matplotlib:

- побудова лінійних графіків, гістограм, діаграм розсіювання, кругових діаграм тощо;
- гнучке налаштування кольорів, підписів, осей та стилів;
- можливість збереження графіків у різних форматах (png, pdf, svg);
- підтримка роботи з іншими бібліотеками (NumPy, Pandas, Seaborn).

Основні компоненти (див. рис. 5.1):

- pyplot — основний модуль, що містить функції для побудови графіків;
- Figure — основний об'єкт для побудови графіків;
- Axes — область побудови графіка всередині Figure;
- plot — безпосередньо сам графік (лінії, точки, гістограми тощо).

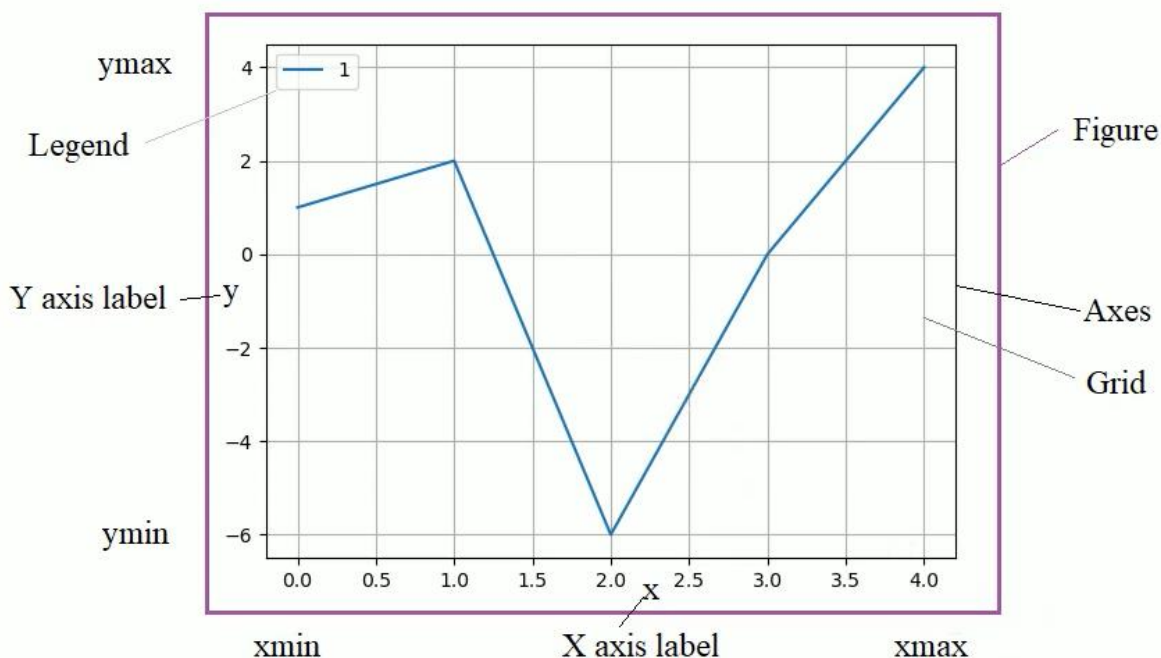


Рисунок 5.1 – Базові компоненти бібліотека Matplotlib

У процесі роботи бібліотека Matplotlib виводить графічну інформацію на екран, використовуючи пакет Tkinter для роботи з користувацьким інтерфейсом. Пакет Tkinter поставляється з самою мовою Python. Виконаємо

метод `get_backend()`, який покаже, що використовується Tkinter у якості backend за замовченням:

```
import matplotlib as plt
```

```
print(plt.get_backend())
```

у консолі з'явиться **TkAgg**. Якщо за якимось причинами TkInter не встановлено, то будуть проблеми при відображенні графіків. При бажанні можна переключитися на інший backend, наприклад, на **Qt5Agg**. Для цього треба виконати інструкцію `plt.use('Name_backend')`.

Бібліотека Matplotlib підтримує широкий спектр типів графіків для візуалізації даних. Ось основні з них:

1. Лінійний графік (Line plot).

Використовується для відображення змін величин у часі або інших послідовностях (див. рис. 5.2).

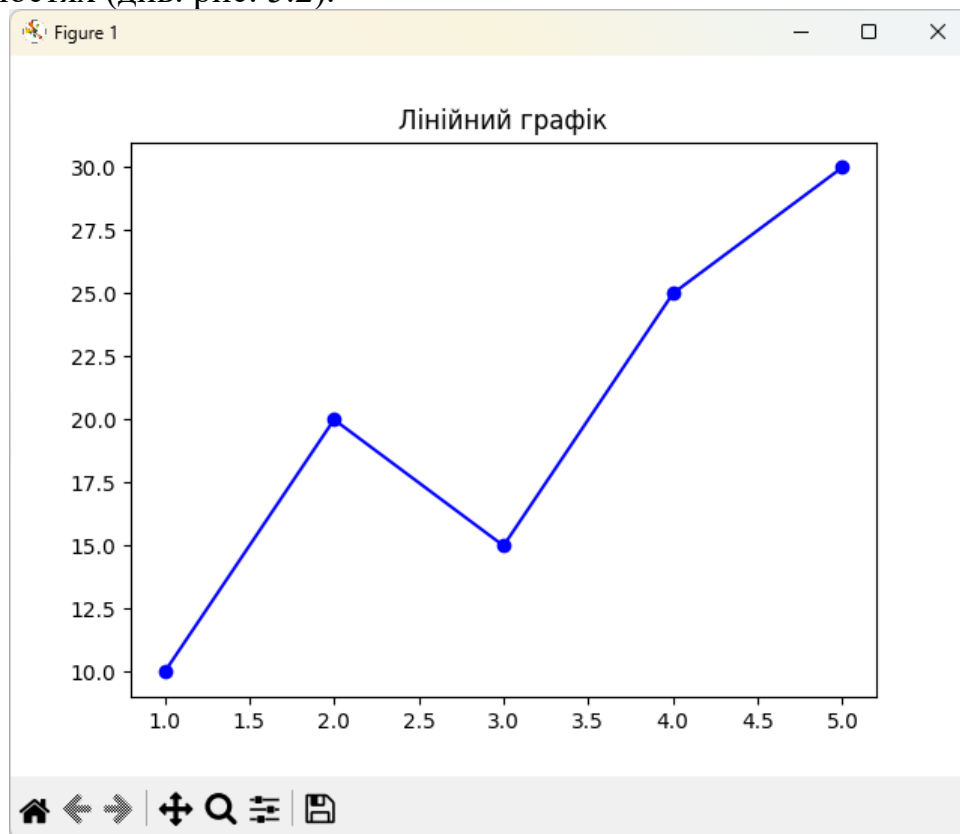


Рисунок 5.2 – Лінійний графік у Matplotlib

```
import matplotlib.pyplot as plt
```

```
x = [1, 2, 3, 4, 5]  
y = [10, 20, 15, 25, 30]
```

```
plt.plot(x, y, marker='o', linestyle='-', color='b')  
plt.title("Лінійний графік")  
plt.show()
```

2. Точковий графік (Scatter plot).

Корисний для аналізу взаємозв'язку між двома змінними (див. рис. 5.3).

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

x = np.random.rand(50)
y = np.random.rand(50)

plt.scatter(x, y, color='r')
plt.title("Точковий графік")
plt.show()
```

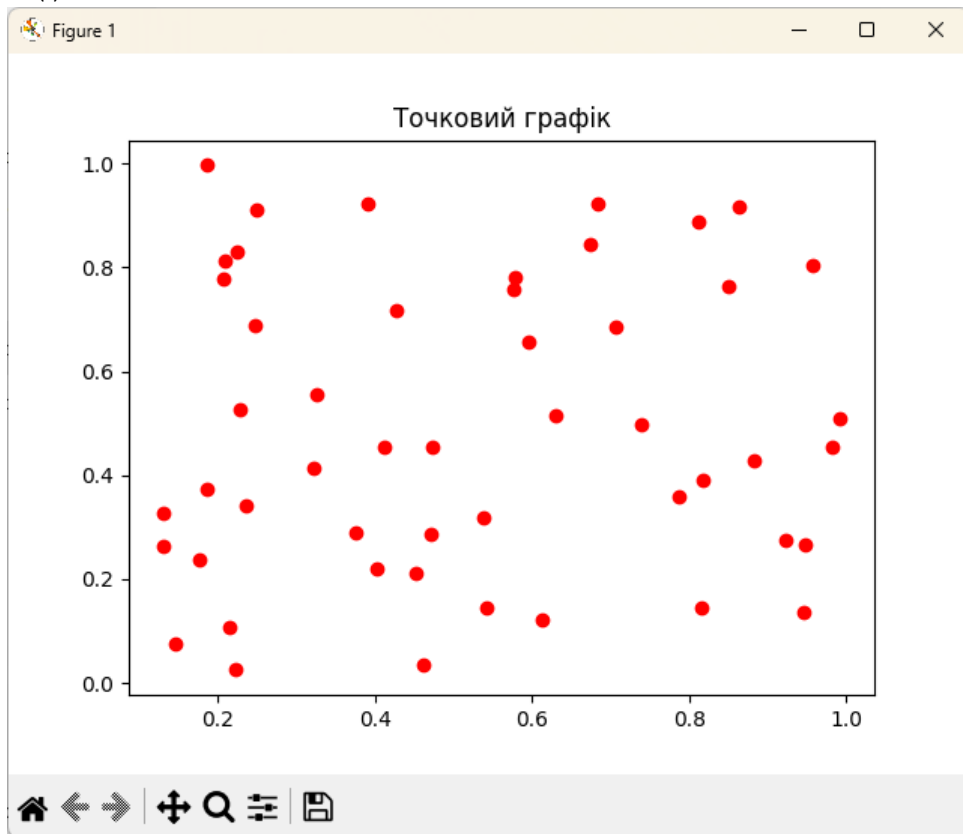


Рисунок 5.3 – Точковий графік у Matplotlib

3. Гістограма (Histogram).

Використовується для аналізу розподілу даних (див. рис. 5.4).

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

data = np.random.randn(1000)

plt.hist(data, bins=30, color='g', alpha=0.7)
plt.title("Гістограма")
plt.show()
```

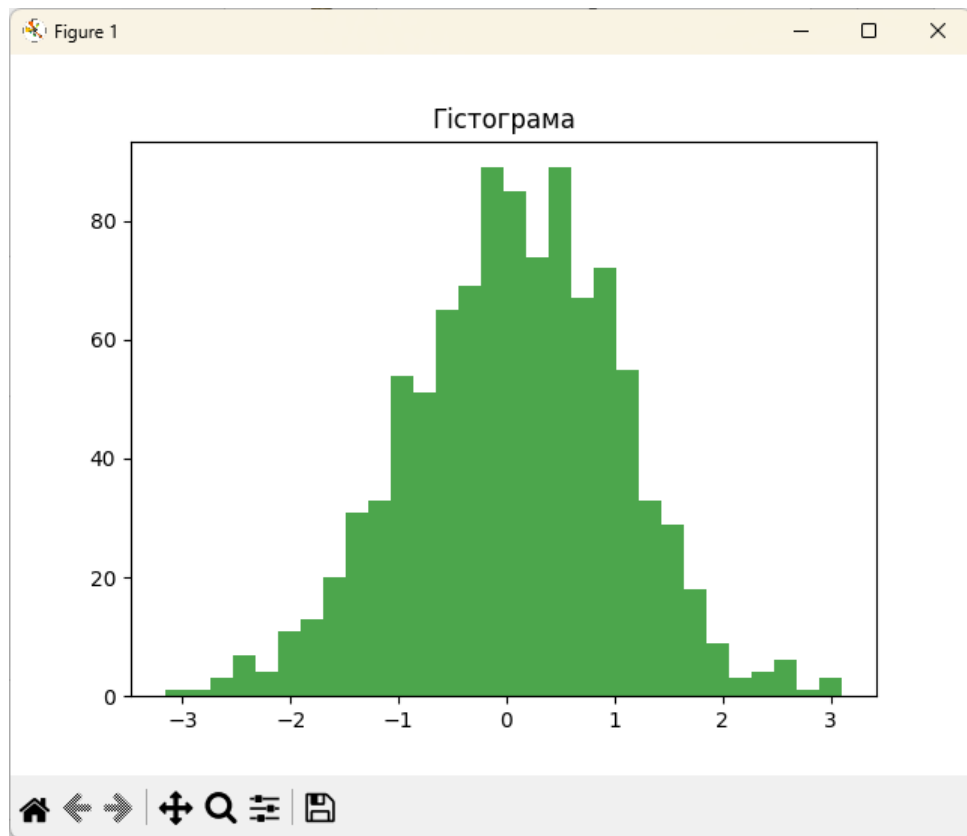


Рисунок 5.4 – Гістограма у Matplotlib

4. Стовпчаста діаграма (Bar chart).

Підходить для порівняння категорій (див. рис. 5.5).

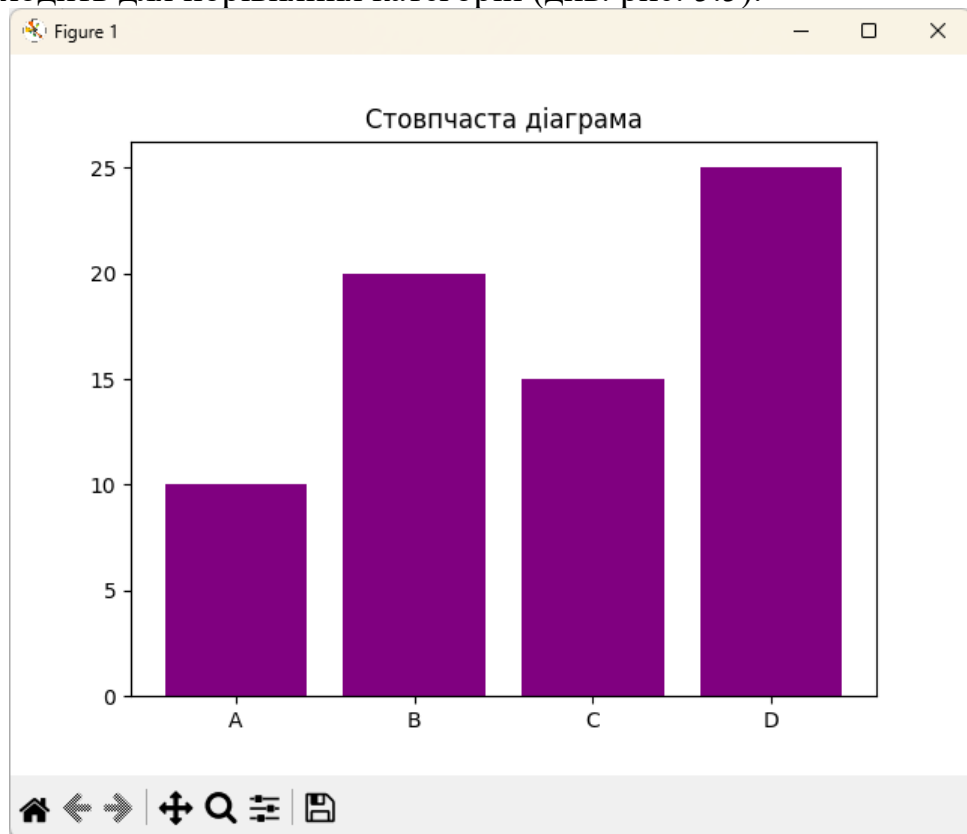


Рисунок 5.5 – Діаграма у Matplotlib

```
import matplotlib.pyplot as plt

labels = ['A', 'B', 'C', 'D']
values = [10, 20, 15, 25]

plt.bar(labels, values, color='purple')
plt.title("Стовпчаста діаграма")
plt.show()
```

5. Кругова діаграма (Pie chart).

Корисна для представлення часток у загальній сумі (див. рис. 5.6).

```
import matplotlib.pyplot as plt

sizes = [30, 20, 25, 25]
labels = ['A', 'B', 'C', 'D']

plt.pie(sizes, labels=labels, autopct='%1.1f%%',
        colors=['blue', 'red', 'green', 'yellow'])
plt.title("Кругова діаграма")
plt.show()
```

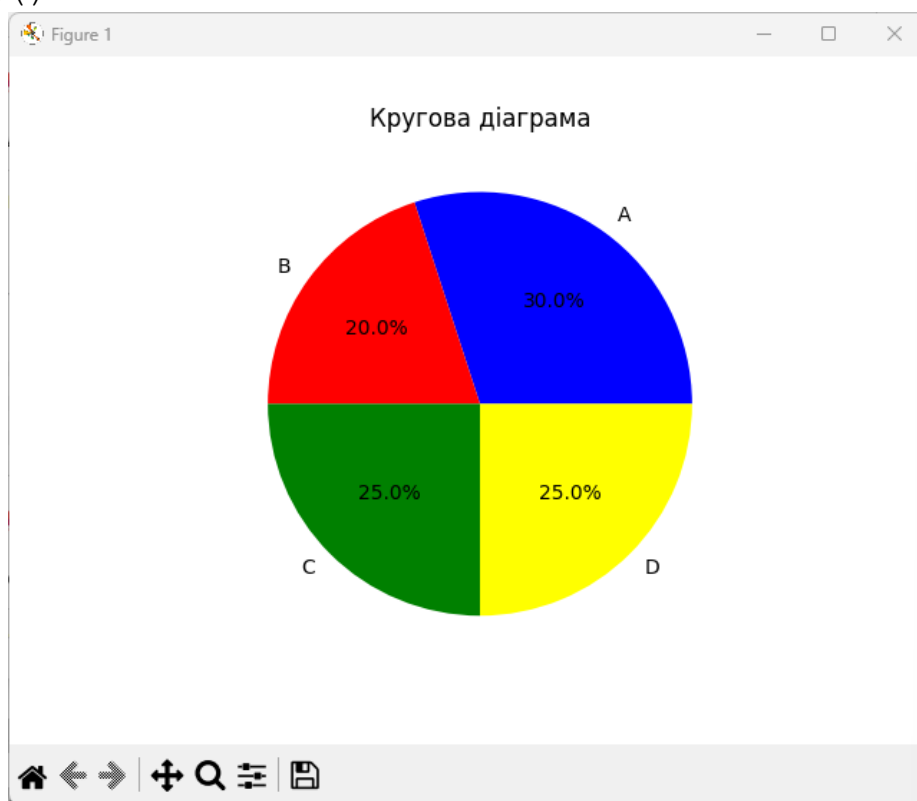


Рисунок 5.6 – Кругова діаграма у Matplotlib

6. Графік із заповненою площею (Area plot).

Використовується для аналізу накопичених даних. Наприклад, $y = \sin x$, $x \in [0; 2\pi]$ (див. рис. 5.7).

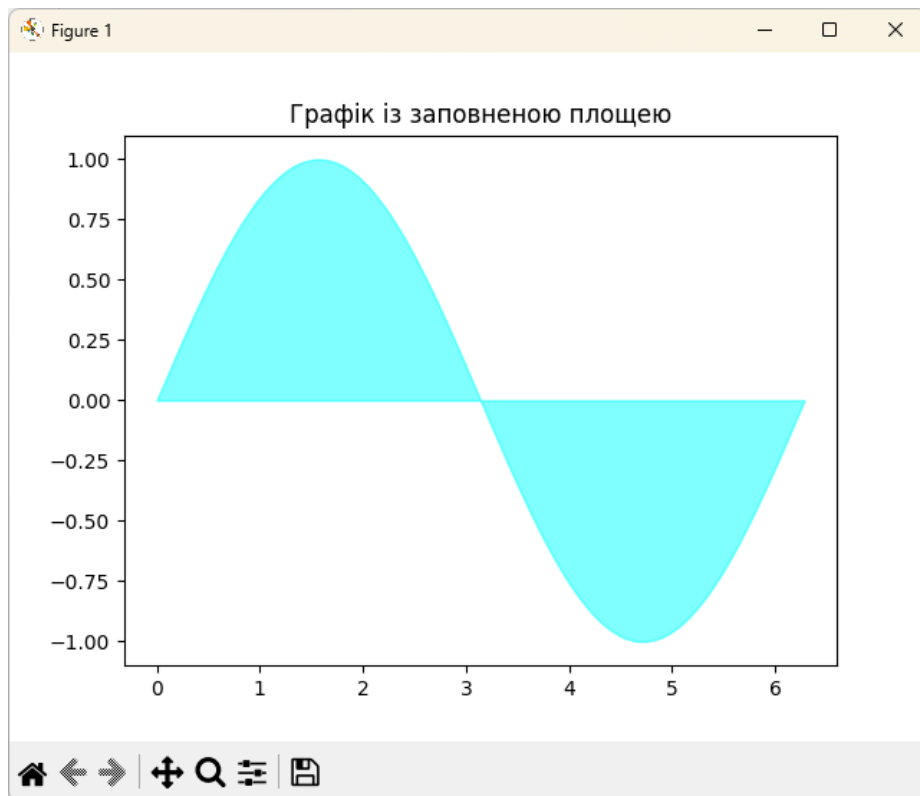


Рисунок 5.7 – Графік із заповненою площею у Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0, 2*np.pi, 0.001)
y = np.sin(x)

plt.fill_between(x, y, color='cyan', alpha=0.5)
plt.title("Графік із заповненою площею")
plt.show()
```

7. Боксплот (Box plot).

Допомагає виявити розкид даних та аномальні значення (див. рис. 5.8).

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

data = [np.random.randn(100) for _ in range(4)]

plt.boxplot(data, labels=['A', 'B', 'C', 'D'])
plt.title("Боксплот")
plt.show()
```

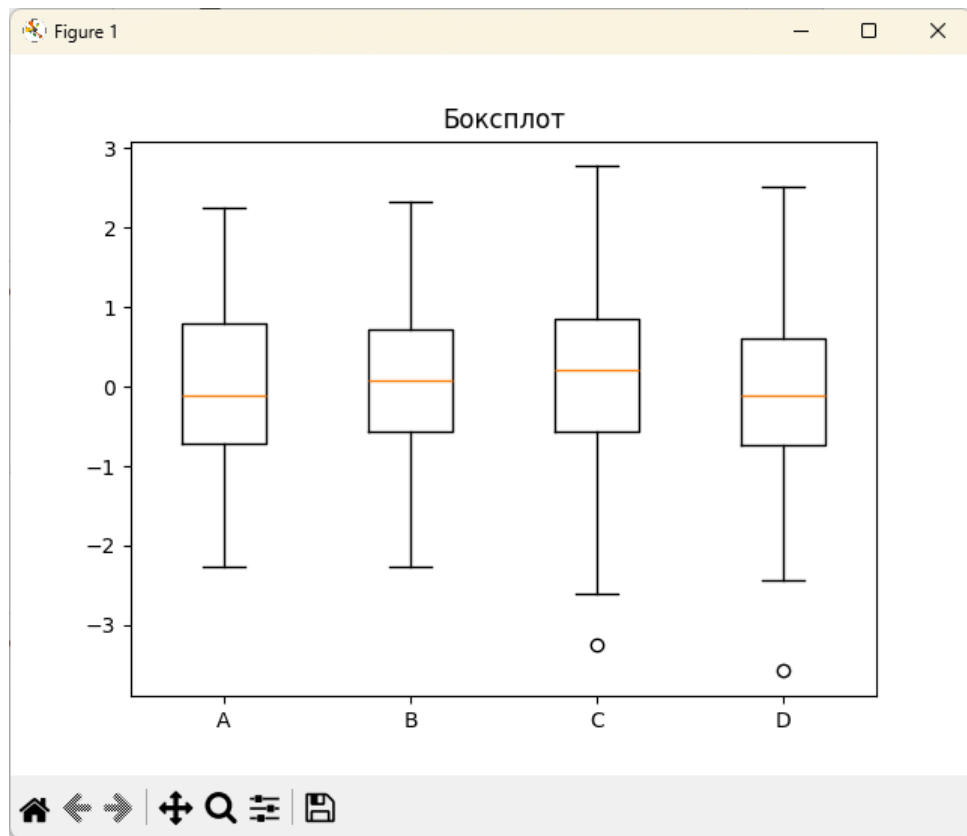


Рисунок 5.8 – Боксплот у Matplotlib

Matplotlib також підтримує 3D-графіки.

8. 3D-лінійний графік (див. рис. 5.9).

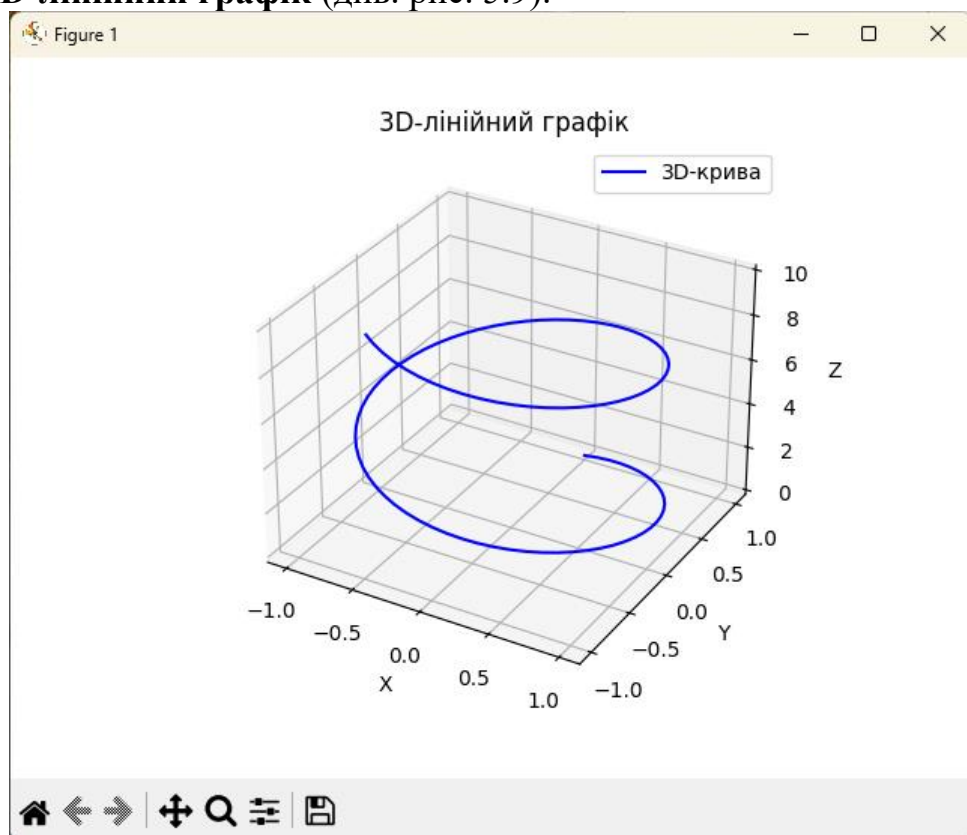


Рисунок 5.9 – 3D-лінійний графік у Matplotlib

```

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

# Створюємо 3D-фігуру
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Дані
t = np.linspace(0, 10, 100)
x = np.sin(t)
y = np.cos(t)
z = t

# Побудова 3D-лінії
ax.plot(x, y, z, color='b', label='3D-крива')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('3D-лінійний графік')
plt.legend()
plt.show()

```

9. 3D-точковий графік (Scatter plot) (див. рис. 5.10).

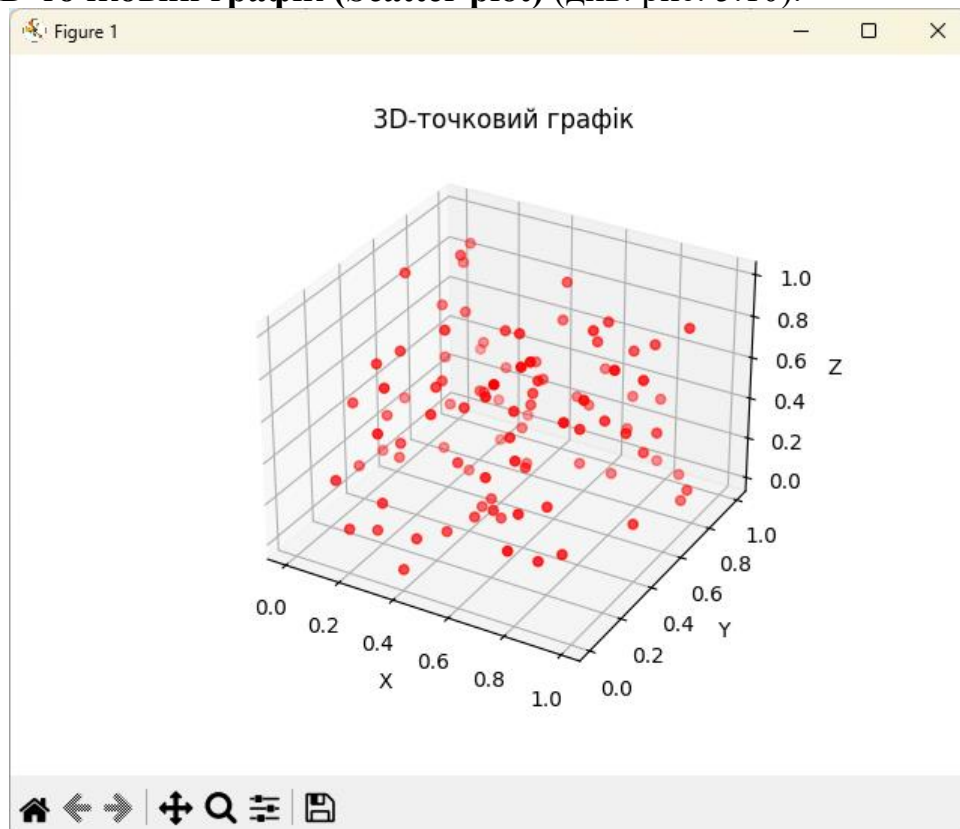


Рисунок 5.10 – 3D-точковий графік у Matplotlib

```

import matplotlib.pyplot as plt
import numpy as np

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

# Генерація випадкових точок
x = np.random.rand(100)
y = np.random.rand(100)
z = np.random.rand(100)

# Побудова точкового графіка
ax.scatter(x, y, z, c='r', marker='o')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('3D-точковий графік')

plt.show()

```

10. Поверхні (Surface plot).

Наприклад, побудуємо функцію $z = \sin \sqrt{x^2 + y^2}$ у площині XYZ (див. рис. 5.11).

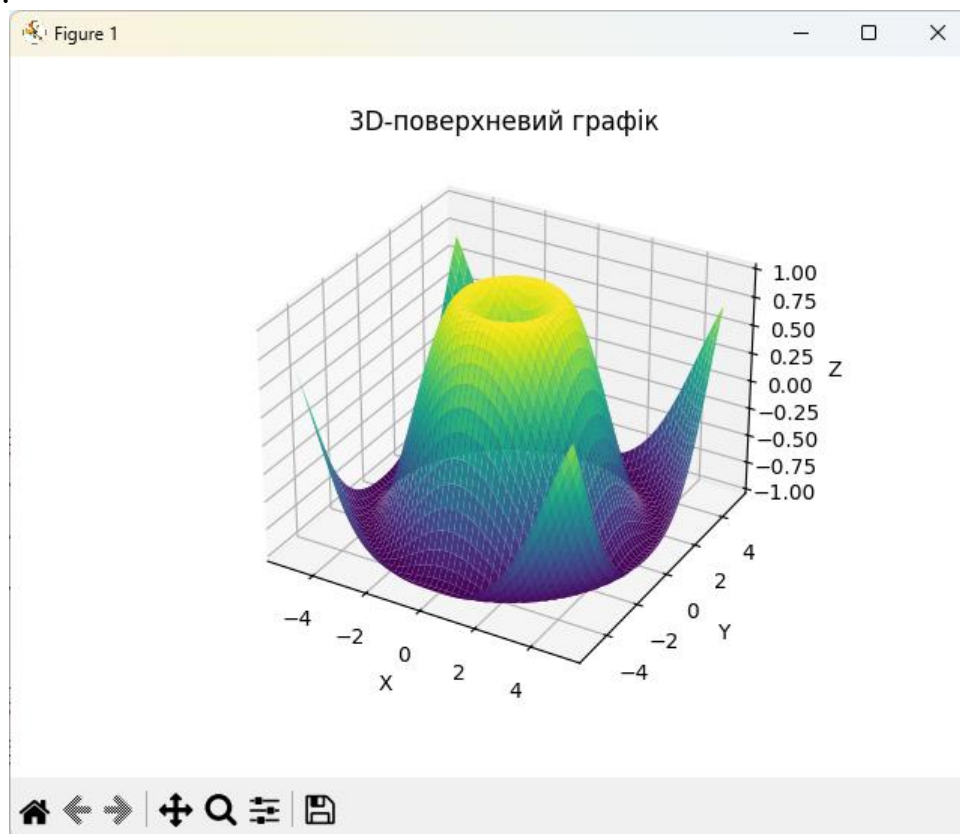


Рисунок 5.11 – Поверхня у Matplotlib

```

import matplotlib.pyplot as plt
import numpy as np

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

# Створення сітки значень
x = np.linspace(-5, 5, 50)
y = np.linspace(-5, 5, 50)
X, Y = np.meshgrid(x, y)
Z = np.sin(np.sqrt(X**2 + Y**2))

# Побудова поверхні
ax.plot_surface(X, Y, Z, cmap='viridis')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('3D-поверхневий графік')
plt.show()

```

У Matplotlib можна відобразити зображення у форматі png, jpg, bmp, tiff тощо за допомогою модуля `imshow()`. Наприклад, виведемо на екран зображення у форматі png, яке лежить у корені проєкту (див. рис. 5.12).

Якщо треба завантажити зображення з URL та відобразити його в Matplotlib, то можна використати бібліотеку `requests` або `PIL`.

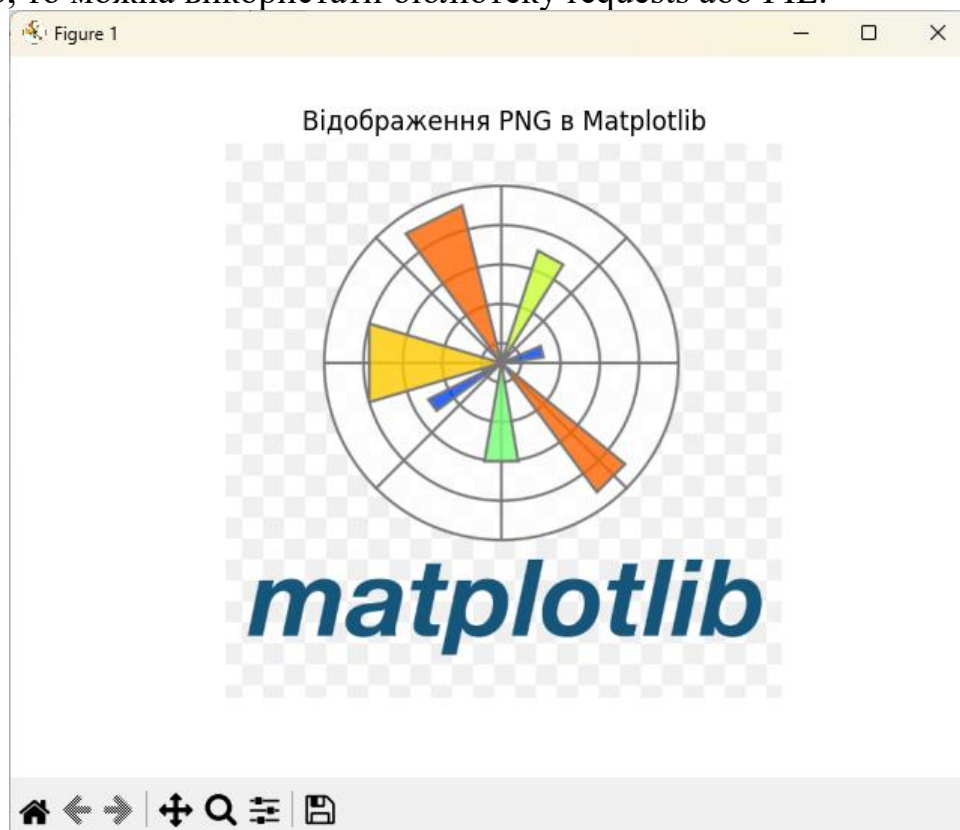


Рисунок 5.12 – Відображення зображення у форматі png у Matplotlib

Завдання до лабораторної роботи

1. Зобразити графіки функцій на одному рисунку різними кольорами, підписати кожен функцію та зберегти рисунок у файл з розширенням .png:

1) $y = \sin(x) + \frac{1}{2}\sin(2x) + \frac{1}{3}\sin(3x), y = \sqrt[7]{x^3 - 3x^2 + 5}.$

2) $y = 8x^2(x^2 - 1)^3, y = \operatorname{tg}\left(9x - \frac{\pi}{2}\right).$

3) $y = \frac{x}{\ln x}, y = x^5 + 6x^4 - \frac{1}{x}.$

4) $y = \frac{x^3 - 2x^2 - x + 2}{x}, y = \operatorname{ctg}\left(3x + \frac{\pi}{2}\right).$

5) $y = (x^2 - 2)e^{-2x}, y = \frac{x^5 - 8x^4 + 3x - 5}{\sqrt{x}}.$

6) $y = \frac{x^3}{x^2 - 1}, y = \sin\left(5x - \frac{\pi}{2}\right).$

7) $y = \sin x - \ln(\sin x), y = e^{2x+3}.$

8) $y = \cos x + \frac{1}{2}\cos(2x) + \frac{1}{3}\cos(3x), y = \sqrt[3]{7x^6 - 5}.$

9) $y = \sqrt[3]{x(7-x)^2} - \frac{x}{2}, y = \cos\left(2x + \frac{\pi}{2}\right).$

10) $y = \left(\frac{x+2}{x-2}\right)^2.$

2. Зобразити графіки функції у полярній системі координат та зберегти рисунок у файл з розширенням .png:

1) $\rho = 5 \sin \varphi.$

2) $\rho = 3 \cos \varphi.$

3) $\rho = 4(1 + \cos \varphi).$

4) $\rho = 5 \sin^2 4\varphi.$

5) $\rho = 2 \sin^2 3\varphi.$

6) $\rho = 3 \sin 4\varphi.$

7) $\rho = 7 \sin 5\varphi.$

8) $\rho = 2(2 + \cos \varphi).$

9) $\rho = 6 \cos 2\varphi.$

10) $\rho = 7 \cos 4\varphi.$

3. Зобразити графік поверхні функції та зберегти рисунок у файл з розширенням .png:

1) $z = 4x^2 + 9y^2.$

2) $z = 6 - (x^2 + y^2).$

3) $z = 9 - x^2.$

4) $z = \frac{y^2}{15}.$

5) $z = 3x^2 - 2y^2.$

6) $z = 2x^2 - 3y^2.$

7) $z = 5x^2 - 4.$

$$8) z = 4x^2 + \frac{y^2}{10} + 1.$$

$$9) z = 9x^2 + \frac{y^2}{9} + 1.$$

$$10) z = 16x^2 - y^2.$$

4. Зобразити кругову діаграму для компанії, в якій працює $(500 \cdot n)$ співробітників, де n – номер варіанта, з яких: 4% говорить португальською мовою, 5% – українською, 8% – арабською, 14% – японською, 16% – іспанською, 22% – французькою, 31% – англійською.

5. Зобразити гістограму зарплати вчителя по місяцям з попередньої лабораторної роботи за останні два роки.

Питання для самоконтролю

1. Яке призначення бібліотеки Matplotlib?
2. Як додати до графіка сітку?
3. Чи можна одночасно виводити декілька графіків на одному зображенні?
4. Як змінюється стиль відображення лінії?
5. Чи можна змінювати маркери у точках лінії?
6. Як зробити заливку області між графіком і прямою $y = 0$?
7. Чи можливе відображення декількох графіків на різних координатних осях у Matplotlib?
8. Чи можна на зображення у форматі png додати графік функції?
9. Що у Matplotlib у якості backend за замовченням використовується? Наведіть приклади інших підтримуваних Matplotlib backend.
10. Як діаграми можна будувати у Matplotlib?
11. Охарактеризуйте задання логарифмічних і полярних координат.
12. Чи можна реалізувати вбудовування двовимірного рисунка у тривимірний?
13. Чи можна створювати тривимірні стовпчикові діаграми?
14. Охарактеризуйте можливості побудови поверхонь у Matplotlib.
15. Чи можна робити візуалізацію двовимірних векторних полів?