
Lab report 3 (Variant 2)

Student :

Herashchenko Serhii

Teacher:

Zhuravlova Zinaida Yuriyivna

06.11. 2024 p.

Task 1

1.1 Condition

1. Побудувати графіки функцій однієї змінної на вказаних інтервалах. Вивести графіки двома способами:

- в одне вікно на одні осі
- в одне вікно на окремі осі (використовуючи розбиття вікна на дві частини)

Дати заголовки, розмістити підписи до осей, легенд, використовувати різні кольори, стилі ліній і типи маркерів, нанести сітки.

$$f(x) = \sin x^2; g(x) = \cos x^2; x \in [-\pi; \pi]$$

$$u(x) = \frac{x}{20}; v(x) = e^x; x \in [-2; 2]$$

1.2 Solution code

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

```
x1 = np.linspace(-np.pi, np.pi, 100)
x2 = np.linspace(-2, 2, 100)
```

```
f = np.sin(x1 ** 2)
g = np.cos(x1 ** 2)
u = x2 / 20
v = np.exp(x2)
```

```
plt.figure(figsize=(10, 5))
plt.plot(x1, f, label=r'$f(x) = \sin(x^2)$', color='b', linestyle='-', marker='o',
markersize=4)
plt.plot(x1, g, label=r'$g(x) = \cos(x^2)$', color='r', linestyle='--', marker='x',
markersize=4)
plt.title('$f(x)$ and $g(x)$')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.grid(True)
```

```
plt.figure(figsize=(10, 5))
plt.plot(x2, u, label=r'$u(x) = \frac{x}{20}$', color='g', linestyle='-.', marker='s',
markersize=4)
plt.plot(x2, v, label=r'$v(x) = e^x$', color='m', linestyle=':', marker='d', markersize=4)
plt.title('$u(x)$ and $v(x)$')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.grid(True)
```

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 5))
```

```
ax1.plot(x1, f, label=r'$f(x) = \sin(x^2)$', color='b', linestyle='-', marker='o',
markersize=4)
ax1.plot(x1, g, label=r'$g(x) = \cos(x^2)$', color='r', linestyle='--', marker='x',
markersize=2)
ax1.set_title('$f(x)$ and $g(x)$')
ax1.set_xlabel('x')
ax1.set_ylabel('y')
ax1.legend()
ax1.grid(True)
```

```
ax2.plot(x2, u, label=r'$u(x) = \frac{x}{20}$', color='g', linestyle='-.', marker='s',
markersize=4)
ax2.plot(x2, v, label=r'$v(x) = e^x$', color='m', linestyle=':', marker='d', markersize=3)
ax2.set_title('$u(x)$ and $v(x)$')
ax2.set_xlabel('x')
ax2.set_ylabel('y')
ax2.legend()
ax2.grid(True)
```

```
plt.tight_layout()
plt.show()
```

1.3 Output

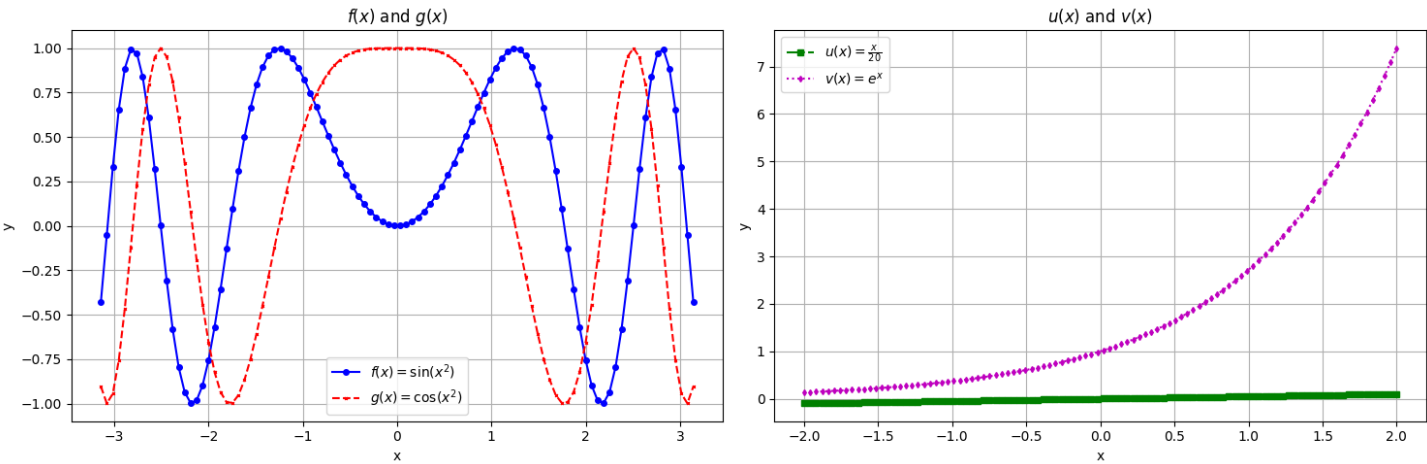


Figure 1

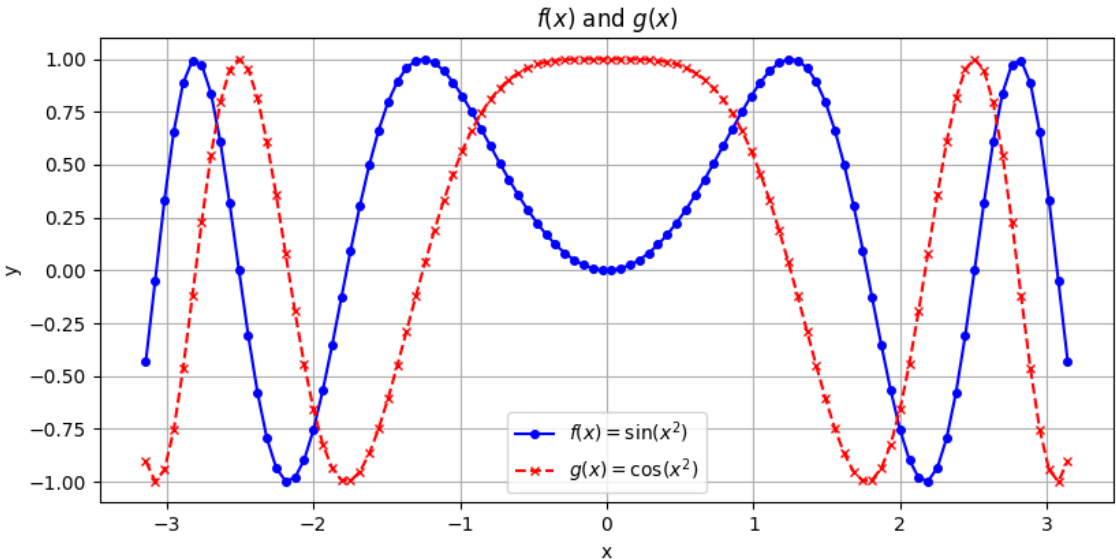


Figure 2

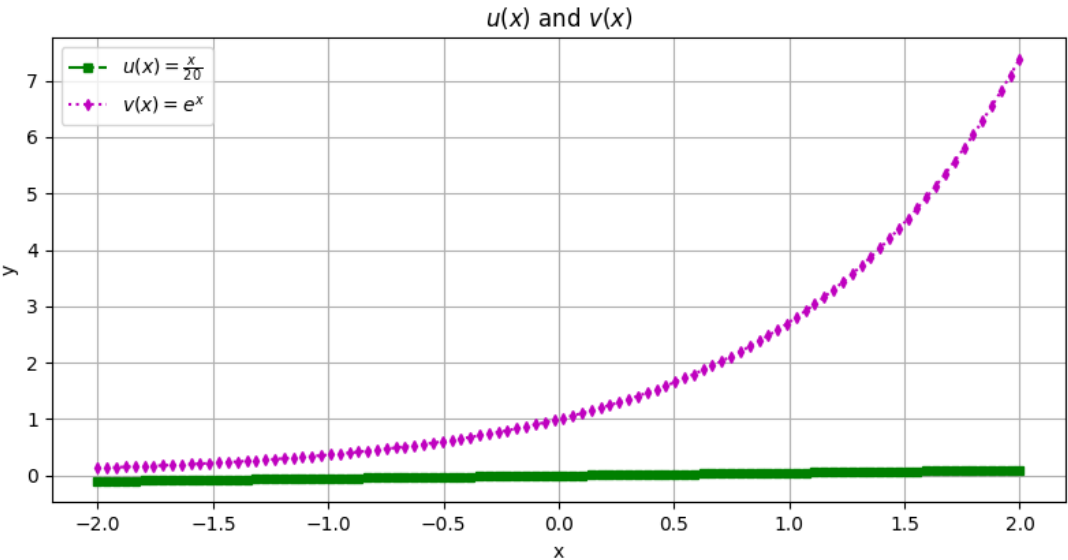


Figure 2

Task 2

2.1 Condition

Побудувати графік функції в полярній системі координат. Колір: зелений, тип маркера: зірка, товщина ліній: 1.

$$\rho = 2\varphi, \varphi \in [0; 8\pi]$$

2.2 Solution code

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

phi = np.linspace(0, 8 * np.pi, 100)
rho = 2 * phi

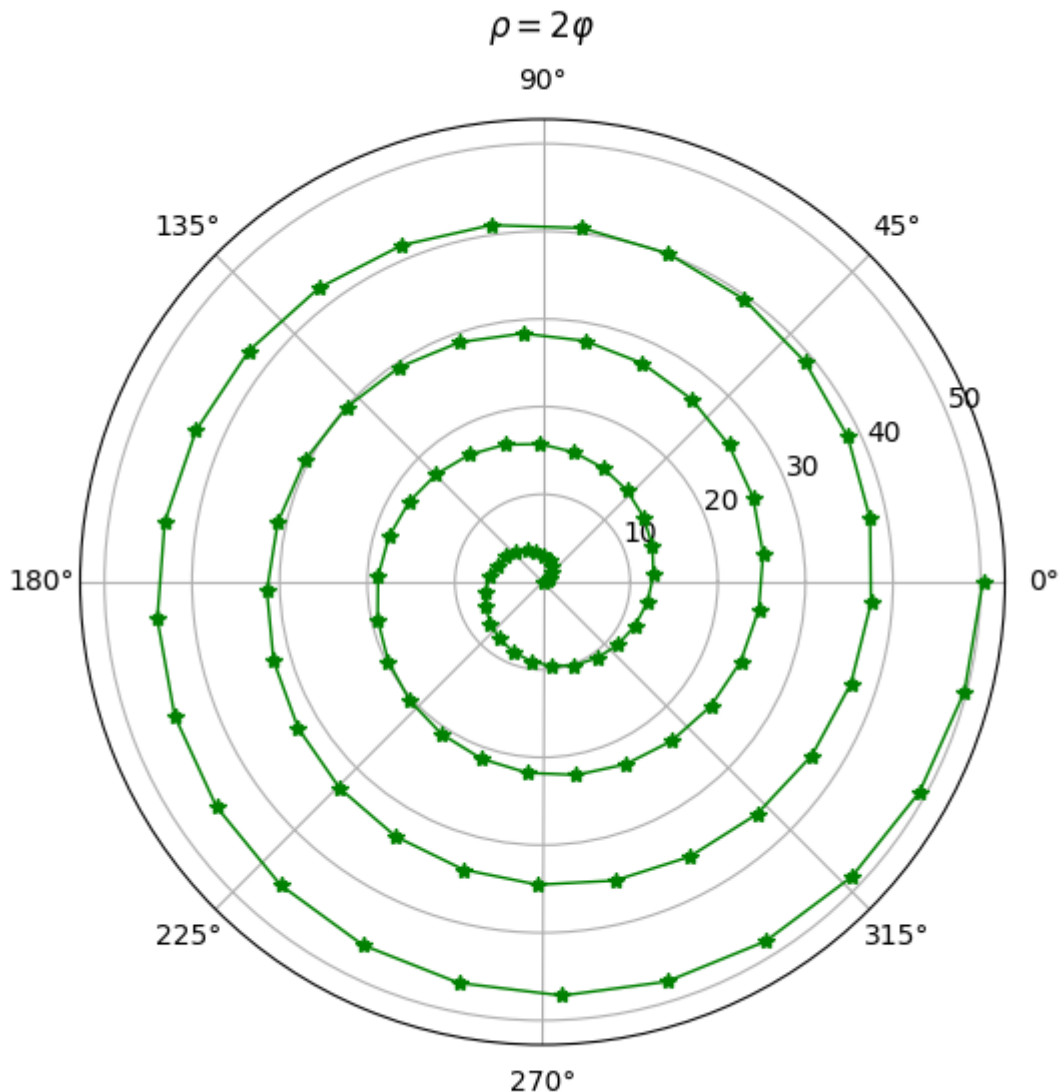
plt.figure(figsize=(6, 6))
ax = plt.subplot(111, projection='polar')

ax.plot(phi, rho, color='green', marker='*', linewidth=1)

ax.set_title(r'$\rho = 2\varphi$', va='bottom')

plt.show()
```

2.3 Output



Task 3

3.1 Condition

Побудувати графік кусочно-заданої функції, відобразити гілки різними кольорами і стилями ліній.

$$f(x) = \begin{cases} \sqrt{x}, & 0 \leq x \leq 1 \\ 1, & 1 < x \leq 3 \\ (x - 4)^2, & 3 < x \leq 5 \end{cases}$$

3.2 Solution code

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

x1 = np.linspace(0, 1, 100)
y1 = np.sqrt(x1)

x2 = np.linspace(1.01, 3, 100)
y2 = np.ones_like(x2)

x3 = np.linspace(3.01, 5, 100)
y3 = (x3 - 4) ** 2

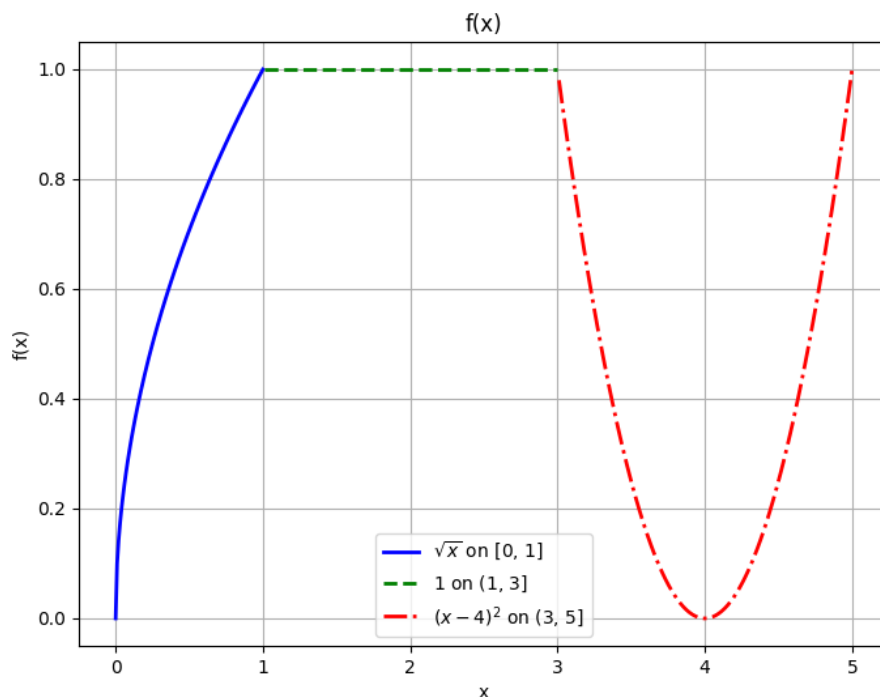
plt.figure(figsize=(8, 6))

plt.plot(x1, y1, label=r'$\sqrt{x}$ on [0, 1]', color='b', linestyle='-', linewidth=2)
plt.plot(x2, y2, label=r'$1$ on (1, 3]', color='g', linestyle='--', linewidth=2)
plt.plot(x3, y3, label=r'$(x - 4)^2$ on (3, 5]', color='r', linestyle='-.', linewidth=2)

plt.title('f(x)')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.legend()
plt.grid(True)

plt.show()
```

3.3 Output



Task 4

4.1 Condition

Побудувати графік параметрично заданої функції.

$$x(t) = 2\sin t - \frac{2}{3}\sin(2t); \quad y(t) = 2\cos t - \frac{2}{3}\cos(2t); \quad t \in [0, 2\pi]$$

4.2 Solution code

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

t = np.linspace(0, 2 * np.pi, 1000)

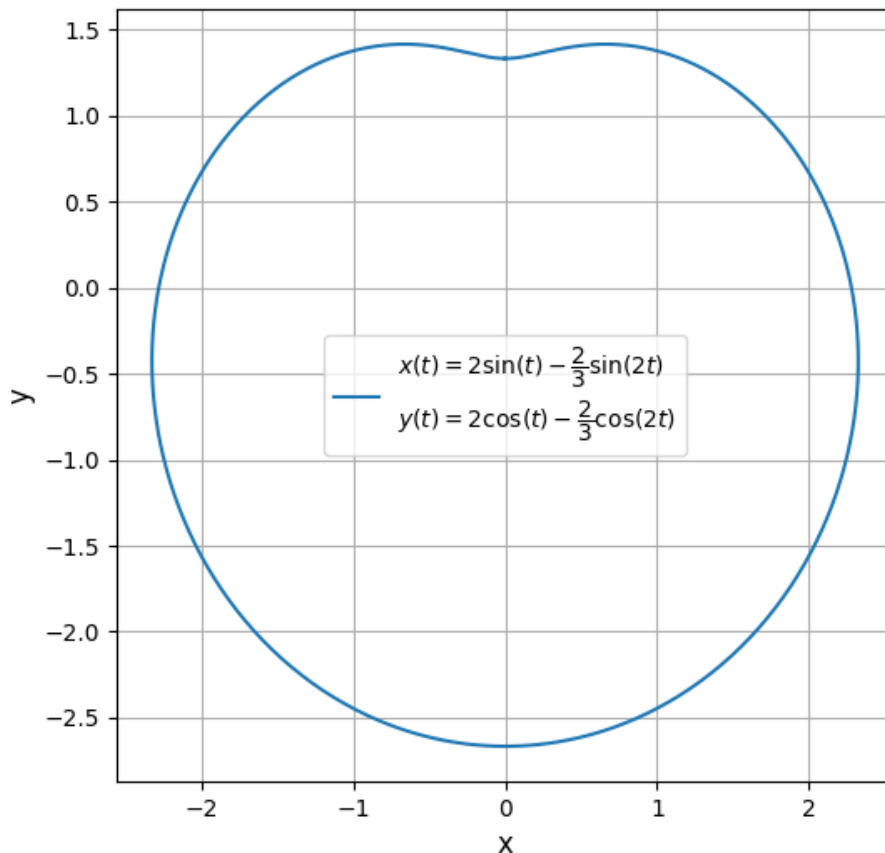
x = 2 * np.sin(t) - (2 / 3) * np.sin(2 * t)
y = 2 * np.cos(t) - (2 / 3) * np.cos(2 * t)

plt.figure(figsize=(6, 6))
plt.plot(x, y, label=r'$x(t)=2\sin(t)-\frac{2}{3}\sin(2t)$' + '\n' + r'$y(t)=2\cos(t)-\frac{2}{3}\cos(2t)$')

plt.xlabel('x', fontsize=12)
plt.ylabel('y', fontsize=12)
plt.grid(True)
plt.legend(loc='best')

plt.show()
```

4.3 Output



Task 5

5.1 Condition

Намалювати 2 прямі, що перетинаються. Обчислити координати точки їх перетинання та зобразити цю точку на рисунку. Підписати на графіці рівняння прямих і координати точки перетину.

$$y_1 = 5x + 3; y_2 = 3x - 7$$

5.2 Solution code

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

A = np.array([[5, 1], [-3, 1]])
b = np.array([3, -7])

solution = np.linalg.solve(A, b)
x_intersection, y_intersection = solution

x = np.linspace(-10, 10, 400)

def y1(x):
    return 5 * x + 3

def y2(x):
    return 3 * x - 7

plt.figure(figsize=(8, 6))

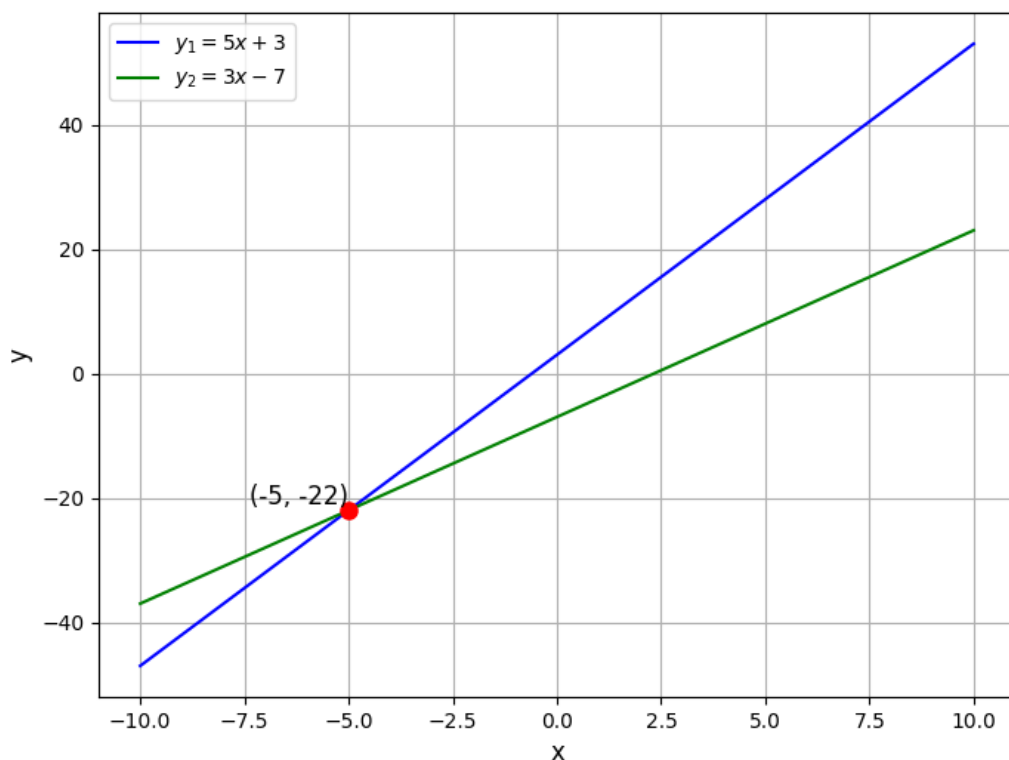
plt.plot(x, y1(x), label=r'$y_1 = 5x + 3$', color='b')
plt.plot(x, y2(x), label=r'$y_2 = 3x - 7$', color='g')

plt.plot(x_intersection, y_intersection, 'ro', markersize=8)
plt.text(x_intersection, y_intersection,
         f'({int(x_intersection)}, {int(y_intersection)})', fontsize=12, ha='right',
         va='bottom')

plt.xlabel('x', fontsize=12)
plt.ylabel('y', fontsize=12)
plt.grid(True)
plt.legend(loc='best')

plt.show()
```

5.3 Output



Task 6

6.1 Condition

Побудувати графік неявної функції.

$$(x^2 + y^2 - 3)\sqrt{x^2 + y^2} + \sin\left(8\sqrt{x^2 + y^2}\right)\cos\left(6\arctg \frac{y}{|x|}\right) = \frac{3}{4}\left(\sin\left(5\arctg \frac{y}{|x|}\right) - 1\right); x, y \in [-2; 2]$$

6.2 Solution code

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

x = np.linspace(-2, 2, 400)
y = np.linspace(-2, 2, 400)
X, Y = np.meshgrid(x, y)

R = np.sqrt(X**2 + Y**2)
Theta = np.arctan2(Y, np.abs(X))

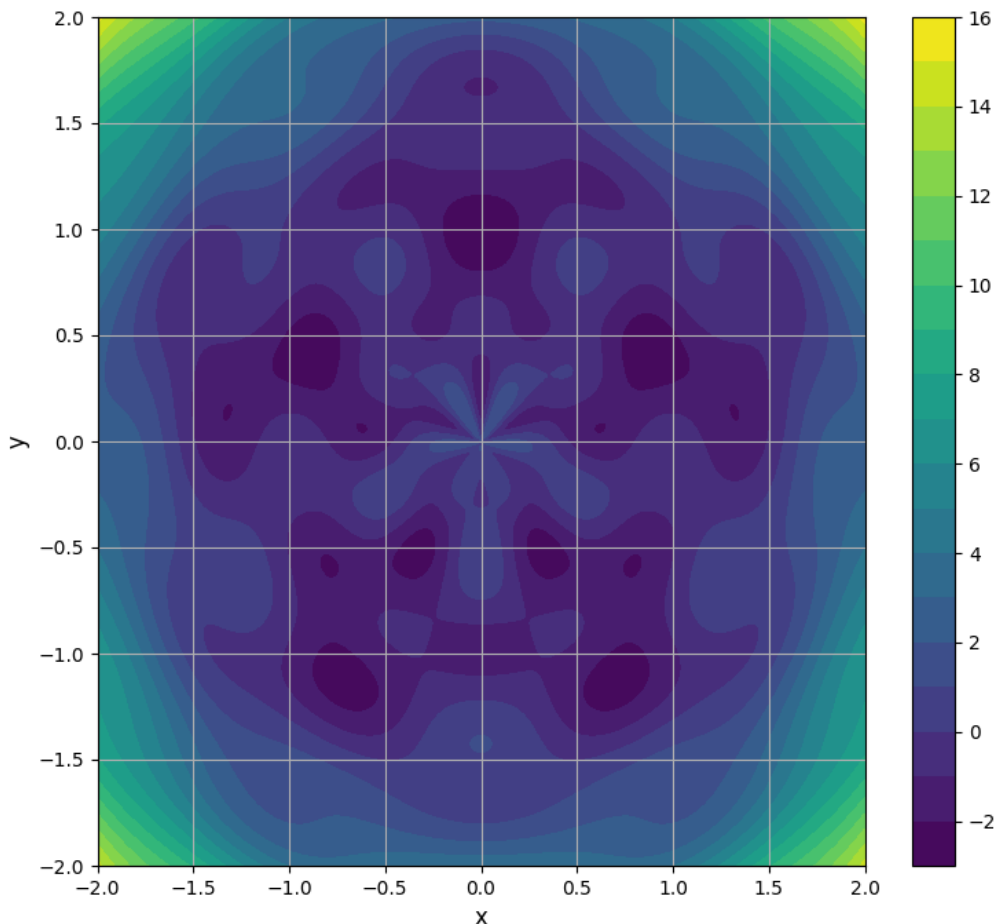
left = (X**2 + Y**2 - 3) * R + np.sin(8 * R) * np.cos(6 * Theta)
right = (3 / 4) * (np.sin(5 * Theta) - 1)

Z = left - right

plt.figure(figsize=(9, 8))
plt.contourf(X, Y, Z, levels=20, cmap='viridis')
plt.colorbar()
plt.xlabel('x', fontsize=12)
plt.ylabel('y', fontsize=12)
plt.grid(True)

plt.show()
```

6.3 Output



Task 7

7.1 Condition

Візуалізувати функцію двох змінних на прямокутній області визначення 2 способами:

- каркасною поверхнею (wire-frame)
- поверхнею с лініями рівня (surface)

Розташувати графіки в окремих графічних вікнах. Представити вид поверхні з декількох точок огляду з відповідною кількістю осей.

$$z(x, y) = \sin^2 x \ln y, x \in [0; 2\pi], y \in [1; 10]$$

7.2 Solution code

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

x = np.linspace(0, 2 * np.pi, 100)
y = np.linspace(1, 10, 100)
X, Y = np.meshgrid(x, y)

Z = np.sin(X) ** 2 * np.log(Y)

fig1 = plt.figure()
ax1 = fig1.add_subplot(111, projection='3d')
ax1.plot_wireframe(X, Y, Z, color='b', linewidth=0.5)
ax1.set_title('Wireframe')
ax1.set_xlabel('x')
ax1.set_ylabel('y')
ax1.set_zlabel('z')

fig2 = plt.figure()
ax2 = fig2.add_subplot(111, projection='3d')
surf = ax2.plot_surface(X, Y, Z, cmap='viridis')
ax2.contour(X, Y, Z, zdir='z', offset=Z.min(), cmap='viridis')
fig2.colorbar(surf, shrink=0.5, aspect=10)
ax2.set_title('Surface plot')
ax2.set_xlabel('x')
ax2.set_ylabel('y')
ax2.set_zlabel('z')

fig3 = plt.figure(figsize=(12, 6))

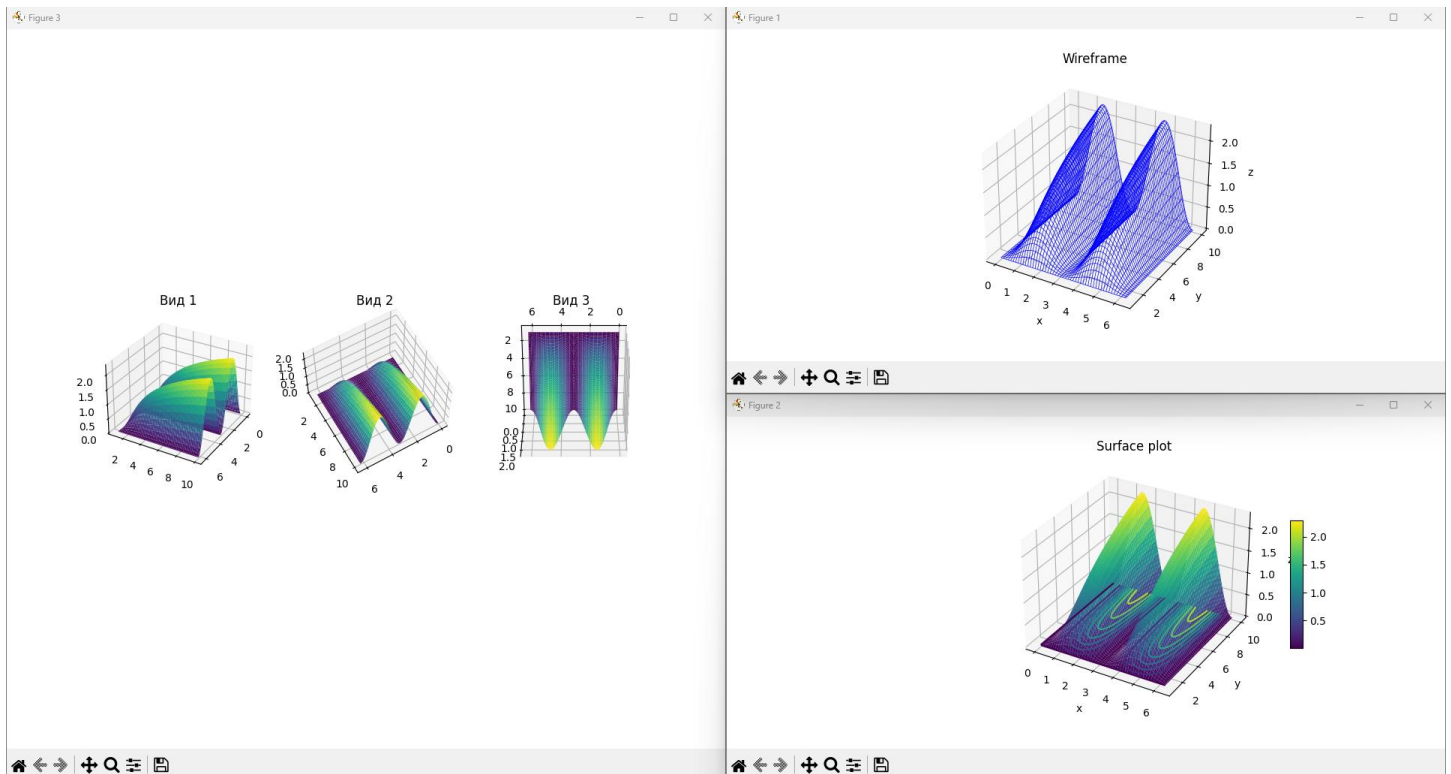
ax3_1 = fig3.add_subplot(131, projection='3d')
ax3_1.plot_surface(X, Y, Z, cmap='viridis')
ax3_1.view_init(30, 30)
ax3_1.set_title('Вид 1')

ax3_2 = fig3.add_subplot(132, projection='3d')
ax3_2.plot_surface(X, Y, Z, cmap='viridis')
ax3_2.view_init(60, 60)
ax3_2.set_title('Вид 2')

ax3_3 = fig3.add_subplot(133, projection='3d')
ax3_3.plot_surface(X, Y, Z, cmap='viridis')
ax3_3.view_init(120, 90)
ax3_3.set_title('Вид 3')

plt.show()
```

7.3 Output



Task 8

8.1 Condition

Намалювати квадрат, що вписаний у коло радіуса r з центром на початку координат. Вирівняти масштаби осей. Отримане зображення зберегти у файл з dpi=400.

8.2 Solution code

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

r = 5

theta = np.linspace(0, 2 * np.pi, 1000)
x_circle = r * np.cos(theta)
y_circle = r * np.sin(theta)

square_x = np.array([-r / np.sqrt(2), r / np.sqrt(2), r / np.sqrt(2), -r / np.sqrt(2), -r /
np.sqrt(2)])
square_y = np.array([-r / np.sqrt(2), -r / np.sqrt(2), r / np.sqrt(2), r / np.sqrt(2), -r /
np.sqrt(2)])

fig, ax = plt.subplots(figsize=(6, 6))

ax.plot(x_circle, y_circle, 'b', label='circle', linewidth=2)
ax.plot(square_x, square_y, 'r-', label='square', linewidth=2)

ax.set_aspect('equal')
ax.grid(True, linestyle='--', alpha=0.7)
ax.legend(loc='best')

ax.set_xlim(-r - 1, r + 1)
ax.set_ylim(-r - 1, r + 1)

plt.savefig('circle_square.png', dpi=400)

plt.show()
```

8.3 Output

