**Звіт до лабораторної роботи 2**

Базовий імпорт та визначення номеру варіанту:

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
n = 5

Побудувати таблицю даних, що п'ять серій спостережень умовної залежності концентрації певного реагенту від часу в хімічній реакції. Перший рядок таблиці повинен містити імена стовпчиків – серія, час, концентрація. Перший стовпчик має містити номери серій. Другий стовпчик має містити п'ять однакових послідовностей моментів спостережень, що відповідають різним серіям спостережень. Ці дані треба сформувати як послідовності від 0 до 12 + n довжиною по n + 13 елементів. Третій стовпчик призначений для запису результатів спостережень, що відповідають вказаним у попередніх стовпчиках серії та моменту часу. Їх треба розрахувати як квадратні корені з суми елементів перших двох стовпчиків відповідного рядка. Для цієї таблиці побудувати:

- графік, що зображує залежності концентрації від часу для кожної серії;

- графік, що зображує залежність середньої концентрації від часу.

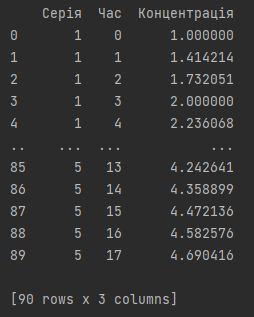
Позначити на графіках назви координатних осей.

Код виконання:

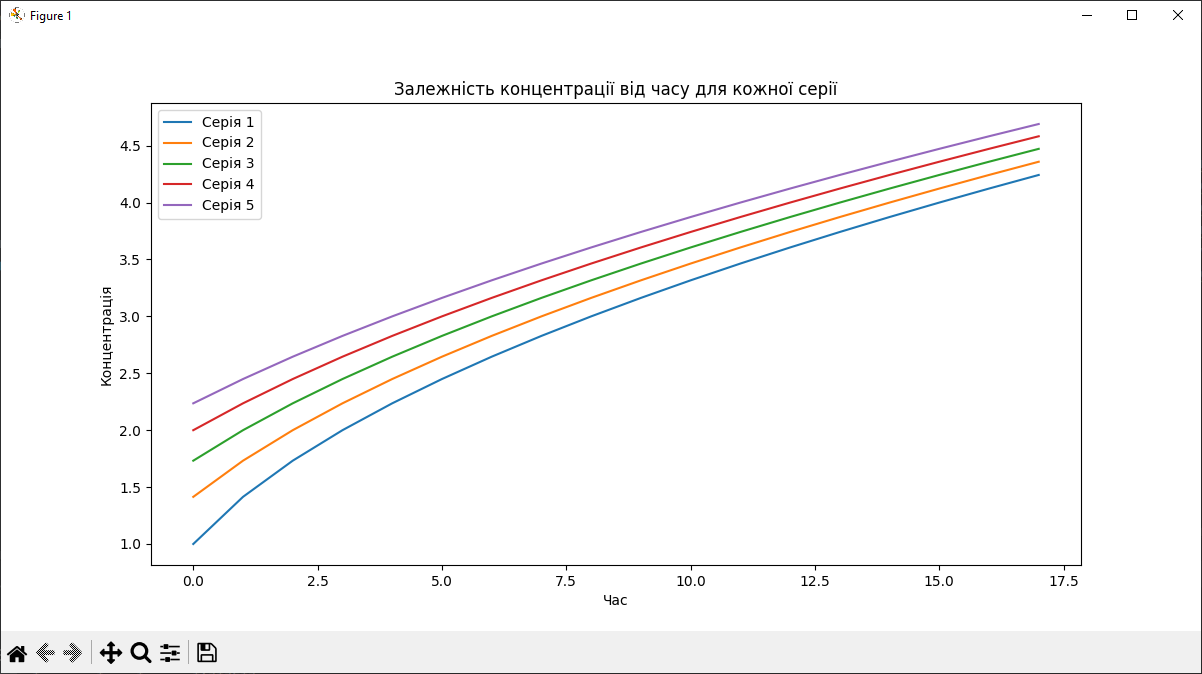
series\_numbers = np.arange(1, 6)  
  
# Створення таблиці даних  
data = {'Серія': np.repeat(series\_numbers, n + 13),  
 'Час': np.tile(np.arange(0, 13 + n), len(series\_numbers)),  
 'Концентрація': np.nan}  
  
df = pd.DataFrame(data)  
  
# Розрахунок концентрації за заданою умовною формулою  
df['Концентрація'] = np.sqrt(df['Серія'] + df['Час'])  
  
# Виведення таблиці  
print(df)  
  
# Побудова графіків  
plt.figure(figsize=(12, 6))  
  
# Графік для кожної серії  
for series in series\_numbers:  
 subset = df[df['Серія'] == series]  
 plt.plot(subset['Час'], subset['Концентрація'], label=f'Серія {series}')  
  
plt.title('Залежність концентрації від часу для кожної серії')  
plt.xlabel('Час')  
plt.ylabel('Концентрація')  
plt.legend()  
plt.show()  
  
# Графік середньої концентрації  
average\_concentration = df.groupby('Час')['Концентрація'].mean().reset\_index()  
  
plt.figure(figsize=(12, 6))  
plt.plot(average\_concentration['Час'], average\_concentration['Концентрація'], color='red', marker='o')  
plt.title('Середня концентрація від часу')  
plt.xlabel('Час')  
plt.ylabel('Середня концентрація')  
plt.show()

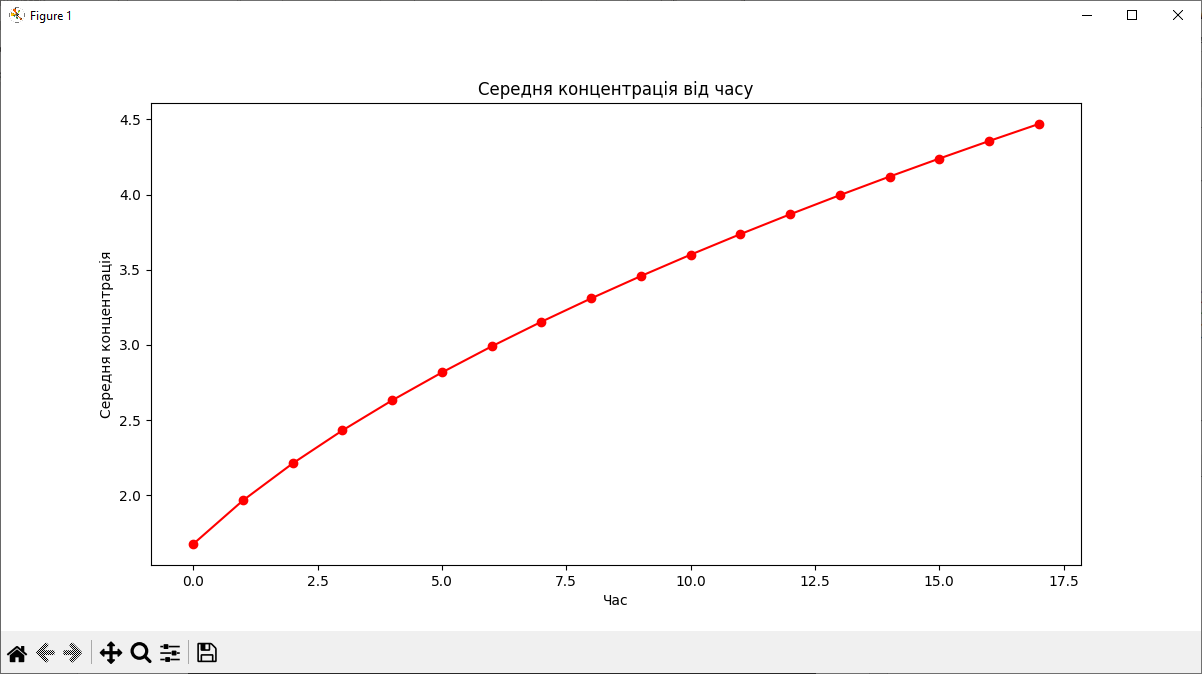
Результат:

Таблиця даних:



Графіки:





Для отриманої при виконанні попереднього завдання залежності середньої концентрації від часу побудувати в одному вікні основні типи графіків (крім графіка типу "n"). При цьому використати кольори з номерами від n до n + 4. Крім того:

- для першого графіка задати діапазон значень за віссю абсцис від 5 до n + 10, а за віссю ординат – від 2 до 2n;

- для другого графіка не вказувати підписи осей;

- для третього графіка не вказувати підписи осей та не будувати координатні осі;

- четвертий графік побудувати, використовуючи логарифмічну вісь абсцис;

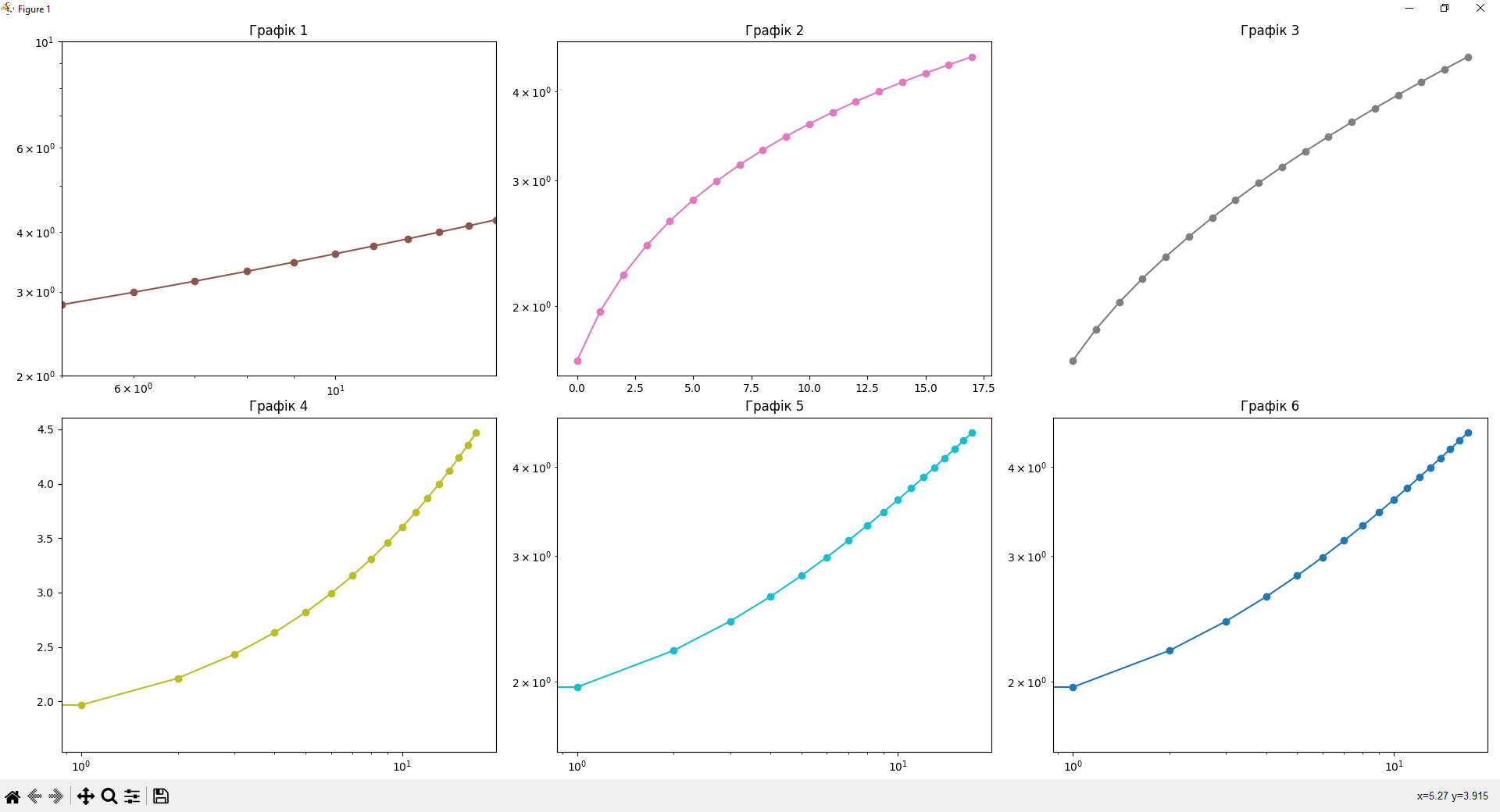
- п’ятий графік побудувати, використовуючи логарифмічну вісь ординат;

- шостий графік побудувати у логарифмічній системі координат.

Код виконання:

# Побудова графіків для другого завдання  
plt.figure(figsize=(18, 12))  
  
# Графік 1: тип "line", задані діапазони для вісей  
plt.subplot(2, 3, 1)  
plt.loglog(average\_concentration['Час'], average\_concentration['Концентрація'], color='C5', marker='o')  
plt.title('Графік 1')  
plt.xlim(5, n + 10) # діапазон для вісі абсцис  
plt.ylim(2, 2 \* n) # діапазон для вісі ординат  
  
# Графік 2: без підписів осей  
plt.subplot(2, 3, 2)  
plt.semilogy(average\_concentration['Час'], average\_concentration['Концентрація'], color='C6', marker='o')  
plt.title('Графік 2')  
  
# Графік 3: без координатних осей  
plt.subplot(2, 3, 3)  
plt.plot(average\_concentration['Час'], average\_concentration['Концентрація'], color='C7', marker='o')  
plt.title('Графік 3')  
plt.axis('off')  
  
# Графік 4: логарифмічна вісь абсцис  
plt.subplot(2, 3, 4)  
plt.semilogx(average\_concentration['Час'], average\_concentration['Концентрація'], color='C8', marker='o')  
plt.title('Графік 4')  
  
# Графік 5: логарифмічна вісь ординат  
plt.subplot(2, 3, 5)  
plt.loglog(average\_concentration['Час'], average\_concentration['Концентрація'], color='C9', marker='o')  
plt.title('Графік 5')  
  
# Графік 6: логарифмічна система координат  
plt.subplot(2, 3, 6)  
plt.plot(average\_concentration['Час'], average\_concentration['Концентрація'], color='C10', marker='o')  
plt.title('Графік 6')  
plt.xscale('log')  
plt.yscale('log')  
  
plt.tight\_layout()  
plt.show()

Результат:



Побудувати три рівнобічні трапеції, що відповідають таким умовам:

- основи всіх трапецій паралельні осі абсцис;

- довжина більшої (нижньої) основи вдвічі більша за довжину меншої;

- перша трапеція має: довжину більшої основи 3n, лівий кінець більшої основи розташований у точці з координатами (n, n), висота трапеції дорівнює 2n;

- друга трапеція має: довжину більшої основи 2,6n, лівий кінець більшої основи розташований у точці з координатами (1,2n, 1,2n), висота трапеції дорівнює 1,6n;

- третя трапеція має: довжину більшої основи 2n, лівий кінець більшої основи розташований у точці з координатами (1,5n, 1,5n), висота трапеції дорівнює n;

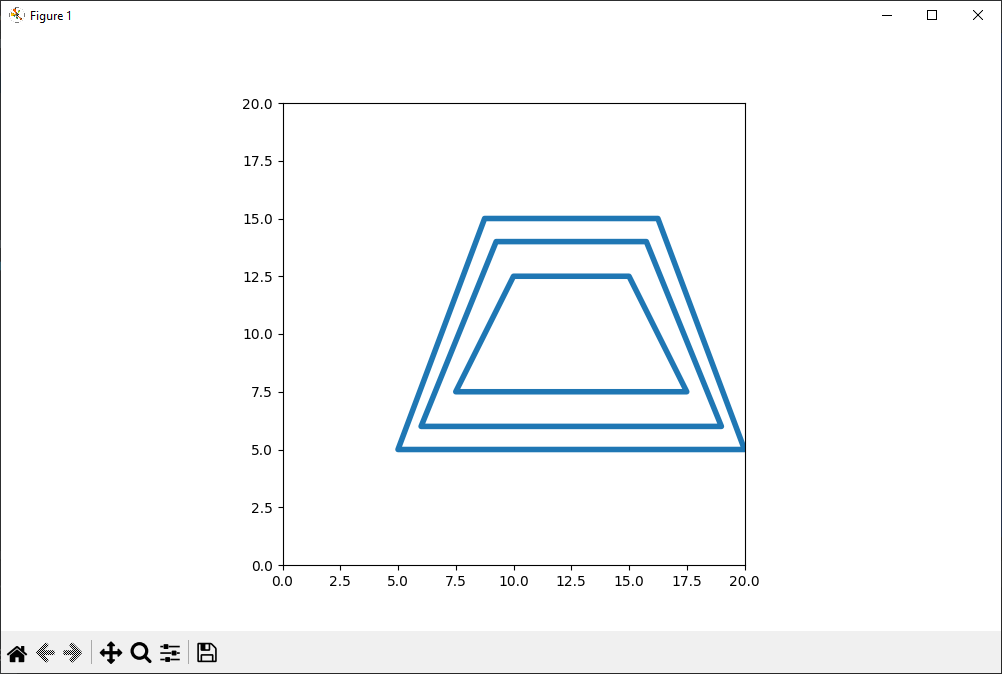
Для побудови використовувати такі властивості ліній:

перша трапеція: тип лінії 1 (для варіантів 1 – 5), 2 (варіанти 6 – 10), 3 (варіанти 11 – 15), 4 (варіанти (16 – 20), 5 (варіанти (21 – 25), 6 (варіанти 26 – 30); колір лінії – 5 + n; товщина лінії – 15 + n/5 (округлити до найближчого цілого); тип з'єднання ліній 0 (для варіантів 6 – 15), 1 (варіанти 16 – 25), 2 (варіанти 26 – 30 та 1 – 5);

код виконання:

import matplotlib.pyplot as plt  
import matplotlib.patches as patches  
  
n = 5 # встановлене значення n  
  
# Побудова графіка  
fig, ax = plt.subplots(figsize=(10, 6))  
  
# Побудова першої трапеції  
trap1 = patches.Polygon([(n, n),(4 \* n, n), (3.25 \* n, 3 \* n), (1.75 \* n, 3 \* n)],  
 closed=True, edgecolor='C10', linewidth=4, linestyle='-', joinstyle='round', facecolor='none')  
ax.add\_patch(trap1)  
  
# Побудова другої трапеції  
trap2 = patches.Polygon([(1.2 \* n, 1.2 \* n), ((1.2+2.6) \* n, 1.2 \* n), ((1.2+2.6\*0.75) \* n, (1.2+1.6) \* n),  
 ((1.2+2.6\*0.25) \* n, (1.2+1.6) \* n)], closed=True, edgecolor='C10', linewidth=4,  
 linestyle='-', joinstyle='round', facecolor='none')  
ax.add\_patch(trap2)  
  
# Побудова третьої трапеції  
trap3 = patches.Polygon([(1.5 \* n, 1.5 \* n), (3.5 \* n, 1.5 \* n), (3 \* n, 2.5 \* n), (2 \* n, 2.5 \* n)],  
 closed=True, edgecolor='C10', linewidth=4, linestyle='-', joinstyle='round', facecolor='none')  
ax.add\_patch(trap3)  
  
# Додаткові налаштування графіка  
ax.set\_xlim(0, 4 \* n)  
ax.set\_ylim(0, 4 \* n)  
ax.set\_aspect('equal', 'box') # Одиничне співвідношення масштабів осей  
  
# Відображення графіка  
plt.show()

Результат виконання:



Створити вектор x, що є арифметичною прогресією, яка містить 100 – n елементів від –(n + 1)/n до (n + 1)/n.

Побудувати в одному полі графіки залежностей f(x) = x, f(x) = x^2, f(x) = x^(0.6n). Для всіх залежностей обрати тип графіка "o", ширину лінії – 3.

При цьому:

- для першої залежності використати тип маркера n (якщо n > 25, взяти тип маркера n – 12), колір n + 10, тип лінії 1 або 2 (для парних та непарних n, відповідно);

- для другої – тип маркера n + 1 (якщо n > 24, взяти тип маркера n – 13), колір – n + 11, тип лінії 3 або 4 (для парних та непарних n, відповідно);

- для третьої – тип маркера n + 2 (якщо n > 23, взяти тип маркера n – 14), колір – n + 12, тип лінії 5 або 6 (для парних та непарних n, відповідно).

На графіку навести легенду, яка має зазначати типи маркерів і ліній, а також кольори для кожної побудованої залежності. Параметри розташування легенди обрати так, щоб вона не накладалася на побудовані залежності та координатні осі.

Код виконання:

import matplotlib.pyplot as plt  
import numpy as np  
  
# Визначення параметрів  
n = 5 # встановлене значення n  
num\_elements = 100 - n  
  
# Створення вектора x арифметичної прогресії  
x = np.linspace(-(n + 1) / n, (n + 1) / n, num\_elements)  
  
# Створення функцій f(x)  
y1 = x  
y2 = x\*\*2  
y3 = x\*\*(0.6 \* n)  
  
# Побудова графіків  
plt.plot(x, y1, marker=5, color=f'C{n+10}', linewidth=3, linestyle='--', label=f'f(x) = x')  
plt.plot(x, y2, marker=6, color=f'C{n+11}', linewidth=3, linestyle='-.', label=f'f(x) = x^2')  
plt.plot(x, y3, marker=7, color=f'C{n+12}', linewidth=3, linestyle='-.', label=f'f(x) = x^(0.6n)')  
  
# Додавання легенди  
plt.legend(loc='upper left', bbox\_to\_anchor=(1, 1))  
  
# Додаткові налаштування графіка  
plt.title('Графіки функцій залежностей від x')  
plt.xlabel('x')  
plt.ylabel('f(x)')  
  
# Показ графіка  
plt.show()

Результат:

