НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
із лабораторної роботи №3
з дисципліни «Теорія керування»
на тему
«Числове розв’язання задач варіаційного числення»

Виконав:

студент групи КМ-03
Шаповалов Г. Г.

Перевірили:

Професор ПМА ФПМ
Норкін В.І.
Асистент ПМА ФПМ
Жук І.С.

Київ — 2024

Зміст

Теоретичні відомості

Розглянемо задачу варіаційного числення, тобто задачу нескінченновимірної оптимізації у просторі неперервно диференційованих функцій з фіксованими кінцями:

$$J(x(\cdot)) = \int_{\alpha}^{\beta} I(t, x(t), \dot{x}(t))dt \to \min_{x(\cdot):x(\alpha)=a, x(\beta)=b}.$$

Апроксимуємо цю задачу скінчено-вимірною задачею оптимізації наступним чином.

На інтервалі інтегрування $[\alpha, \beta]$ визначимо точки $t_i = \alpha + i(\beta - \alpha)/N$, $i = 0,...,N+1$, де $(N+1)$ - число точок, $N \sim 10$. Довжина інтервалу між сусідньома точками дорівнює $\Delta t = (\beta - \alpha)/N$.

Введемо $N-1$ змінну $x_i = x(t_i)$, $i = 1,...,N-1$, позначимо $x_0 = a$, $x_N = b$.

Апроксимуємо інтеграл в задачі варіаційного числення за формулою прямокутників:

$$J(x(\cdot)) \approx \sum_{i=0}^{N-1} I\left(t_i, x(t_i), \dot{x}(t_i)\right)\Delta t.$$

Похідну $\dot{x}(t_i)$ апроксимуємо скінченою різницею:

$$\dot{x}(t_i) \approx (x(t_{i+1}) - x(t_i))/\Delta t = (x_{i+1} - x_i)/\Delta t.$$

Друга похідна апроксимується за формулою:

$$\ddot{x}(t_i) = \left(\left(x(t_{i+1}) - x(t_i)\right)/\Delta t - \left(x(t_i) - x(t_{i-1})\right)/\Delta t\right)/\Delta t =$$
$$= \left(x(t_{i+1}) - 2x(t_i) + x(t_i)\right)/\left(\Delta t\right)^2 = \left(x_{i+1} - 2x_i + x_i\right)/\left(\Delta t\right)^2.$$

Тепер замість вихідної нескінченновимірної оптимізаційної задачі маємо оптимізаційну задачу у $(N-1)$-вимірному просторі:

$$J_N(x_1,...,x_{N-1}) = \sum_{i=0}^{N-1} I\left(t_i, x_i, (x_{i+1} - x_i)/\Delta t\right) \cdot \Delta t \to \min_{x_1,...,x_{N-1}}.$$

Також можна розглянути так звану регулярізовану задачу з додатковим квадратичним членом в цільової функції з параметром регулярізації $\lambda \geq 0$:

$$J_N^\lambda(x_1,...,x_{N-1}) = \sum_{i=0}^{N-1} I\left(t_i, x_i, (x_{i+1}-x_i)/\Delta t\right)\Delta t + \lambda \sum_{i=0}^{N-1}\left((x_{i+1}-x_i)/\Delta t\right)^2 \Delta t \to \min_{x_1,...,x_{N-1}}$$

Якщо вихідна задача містить декілька функцій $x_1(t), x_2(t),...$, то для кожної із них вводяться свої змінні, $x_{1i} = x_1(t_i)$, $x_{2i} = x_2(t_i)$. Таким чином, оптимізаційна задачу буде містити $2N$ змінних, $x_{11}, x_{12},..., x_{1N}, x_{21}, x_{22},..., x_{2N}$.

Ці задачі можна розв'язувати методами безградієнтного типу (по координатний спуск, Нелдера-Міда, випадкових напрямків, скінчено-різницевий). Альтернативно, ці задачі можна також розв'язувати методами градієнтного типу. Градієнт

$$\nabla J_N(x_1,...,x_{N-1}) = \left(\frac{\partial J_N}{\partial x_1},...,\frac{\partial J_N}{\partial x_{N-1}}\right)$$

функції $J_N(x_1,...,x_{N-1})$ наближено обчислюється наступним чином:

$$\frac{\partial J_N}{\partial x_i} = \left.\frac{\partial J}{\partial x}\right|_{t=t_{i-1}}\left(\frac{1}{\Delta t}\right) + \left.\frac{\partial J}{\partial x}\right|_{t=t_i} + \left.\frac{\partial J}{\partial x}\right|_{t=t_i}\left(\frac{-1}{\Delta t}\right) \approx$$

$$\approx \left(\frac{\partial J}{\partial x}\left(t_{i-1}, x_{i-1}, (x_i - x_{i-1})/\Delta t\right)\right) + \left(\frac{\partial J}{\partial x}\left(t_i, x_i, (x_{i+1}-x_i)/\Delta t\right)\right)\Delta t -$$

$$-\left(\frac{\partial J}{\partial x}\left(t_i, x_i, (x_{i+1}-x_i)/\Delta t\right)\right), \qquad i = 1,..., N-1.$$

Тут спочатку для даної підінтегральної функції $I(t, x(t), \dot{x}(t))$ аналітично обчислюються формули для часткових похідних $\frac{\partial J}{\partial x}$ та $\frac{\partial J}{\partial \dot{x}}$ як функцій змінних $(t, x(t), \dot{x}(t))$, а потім замість них підставляються трійки аргументів $\left(t_{i-1}, x_{i-1}, (x_i - x_{i-1})/\Delta t\right)$ та $\left(t_i, x_i, (x_{i+1}-x_i)/\Delta t\right)$.

Аналогічно,

$$\nabla J_N^\lambda(x_1,...,x_{N-1}) = \left(\frac{\partial J_N^\lambda}{\partial x_1},...,\frac{\partial J_N^\lambda}{\partial x_{N-1}}\right),$$

$$\frac{\partial J_N^\lambda}{\partial x_i} = \left.\frac{\partial J}{\partial x}\right|_{t=t_{i-1}}\left(\frac{1}{\Delta t}\right) + \left.\frac{\partial J}{\partial x}\right|_{t=t_i} + \left.\frac{\partial J}{\partial x}\right|_{t=t_i}\left(\frac{-1}{\Delta t}\right) + \frac{2\lambda}{(\Delta t)}\left(x_i - x_{i-1}\right) - \frac{2\lambda}{(\Delta t)}\left(x_{i+1} - x_i\right) \approx$$

$$\approx \left(\frac{\partial J}{\partial x}\left(t_{i-1}, x_{i-1}, (x_i - x_{i-1})/\Delta t\right)\right) + \left(\frac{\partial J}{\partial x}\left(t_i, x_i, (x_{i+1}-x_i)/\Delta t\right)\right)\Delta t -$$

$$-\left(\frac{\partial J}{\partial x}\left(t_i, x_i, (x_{i+1}-x_i)/\Delta t\right)\right) + \frac{2\lambda}{(\Delta t)}\left(x_i - x_{i-1}\right) - \frac{2\lambda}{(\Delta t)}\left(x_{i+1} - x_i\right).$$

Порядок виконання роботи

1) Обрати приклад задачі варіаційного числення зі списку та повідомить викладача.
2) Обрати число точок апроксимації N.
3) Запрограмувати функцію $J_N(x_1,...,x_{N-1})$.
4) Обрати метод безградієнтний (або градієнтний) метод мінімізації $J_N(x_1,...,x_{N-1})$.
5) Застосувати метод до мінімізації $J_N(x_1,...,x_{N-1})$ та побудувати графік залежності отриманого значення мінімізованої функції від числа ітерацій, обрати критерій зупинки методу.
6) Для отриманого розв'язку $(x_1^*,...,x_{N-1}^*)$ (мінімуму) побудувати кусково-лінійну функцію по точкам $\left((\alpha,a),(t_1,x_1^*),...,(t_{N-1},x_{N-1}^*),(\beta,b)\right)$ та зобразити її графічно.
7) Графічно порівняти отриманий наближений розв'язок з точним розв'язком, отриманим аналітично.
8) Дослідити точність апроксимації від числа точок апроксимації.
9) Підготувати письмовий звіт про роботу (та надіслати викладачу).

Для цієї лабораторної роботи відповідно варіанту №24, було вибрано наступну задачу:

24. $\int\limits_0^1 (x_1^2 + x_2^2 + 2x_1'x_2')dt \to extr$, $x_1(0) = x_2(0) = 0$, $x_1(1) = x_2(1) = sh(1)$.

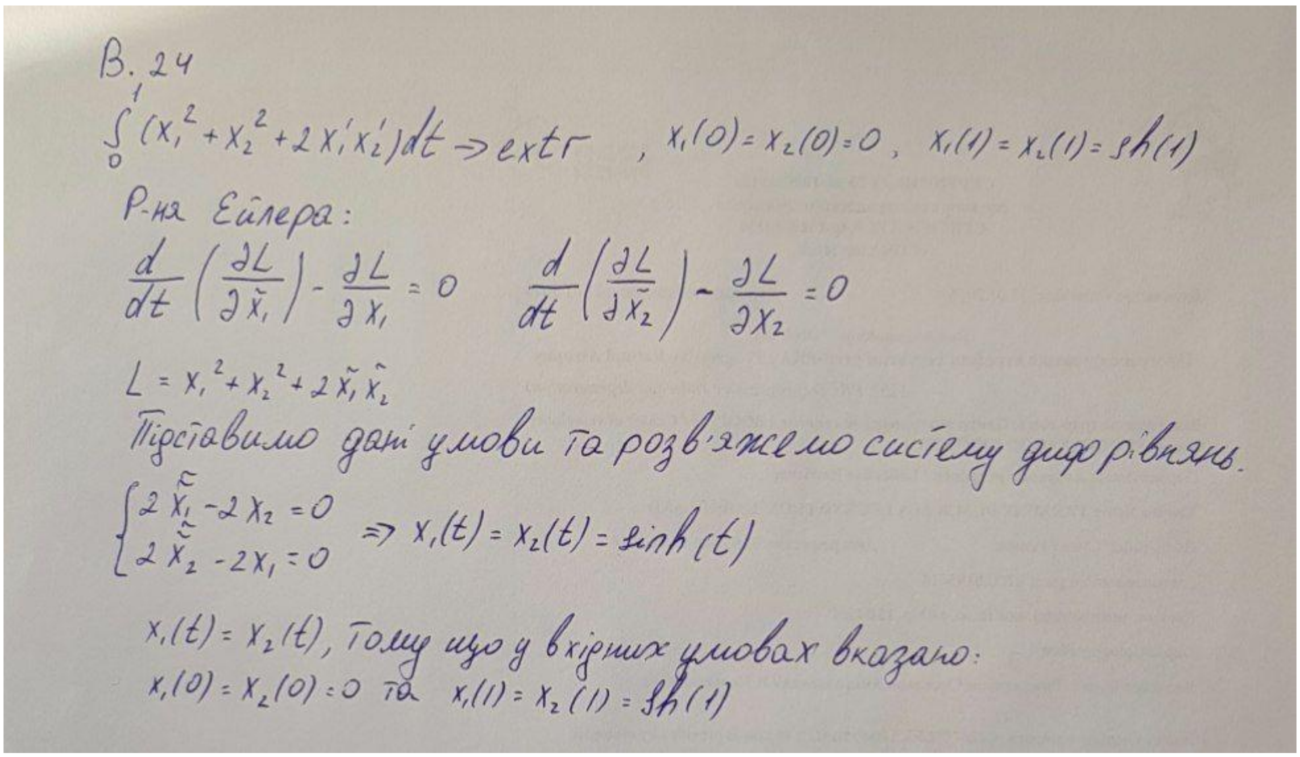
Основна частина

Зафіксуємо точність виводу результатів до 3х чисел після коми.

Запрограмуємо функцію $J_N(x_1,...,x_{N-1})$.

Для знаходження чисельного розв'язку будемо використовувати функцію Scipy.optimize.minimize. Ця функція мінімізує використовуючи метод Broyden–Fletcher–Goldfarb–Shanno – градієнтний метод.

Знайдемо аналітичний розв'язок даної задачі:



В. 24

$\int_0^1 (x_1^2 + x_2^2 + 2x_1'x_2')dt \to extr$, $x_1(0) = x_2(0) = 0$, $x_1(1) = x_2(1) = sh(1)$

Р-ня Ейлера:

$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_1}\right) - \frac{\partial L}{\partial x_1} = 0$ $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_2}\right) - \frac{\partial L}{\partial x_2} = 0$

$L = x_1^2 + x_2^2 + 2\dot{x}_1\dot{x}_2$

Підставимо дані умови та розв'яжемо систему диф рівнянь.

$\begin{cases} 2\ddot{x}_1 - 2x_2 = 0 \\ 2\ddot{x}_2 - 2x_1 = 0 \end{cases} \Rightarrow x_1(t) = x_2(t) = \sinh(t)$

$x_1(t) = x_2(t)$, тому що у вхідних умовах вказано:

$x_1(0) = x_2(0) = 0$ та $x_1(1) = x_2(1) = sh(1)$

Рис. 1 – Аналітичний розв'язок для даної задачі

Давайте порівняємо отримані результати за допомогою графіків.:

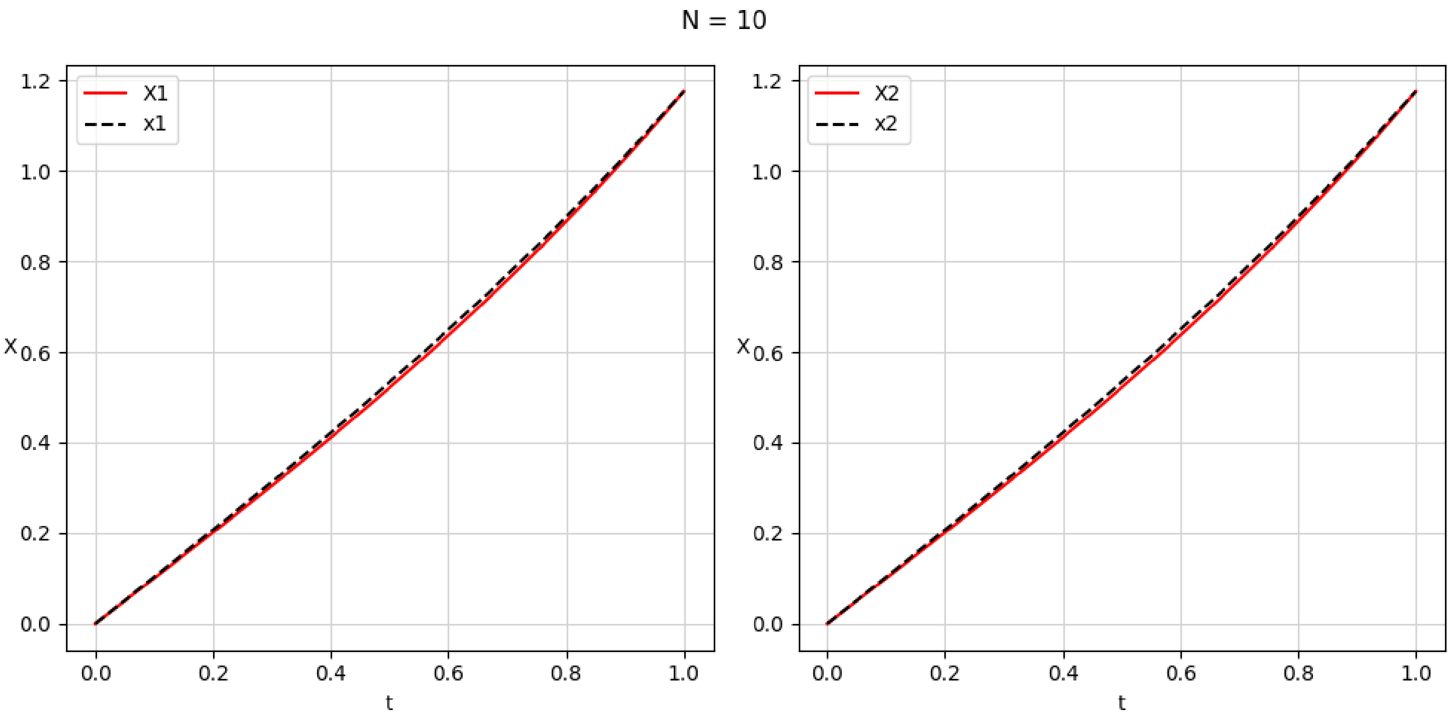


Рис. 2 – Порівняння розв’язків при N=10

Оскільки, функції $x_1(t)$ та $x_2(t)$ співпадають, то і графіки в них будуть однакові. Але через те що змінні різні, вони будуть представлені на окремих графіках.

N = 10, Значення функції в мінімумі = 3.724

Похибка X1 = 0.012

Похибка X2 = 0.012

При такому значення N похибка мала, але дослідимо, що буде, якщо збільшити параметр N.

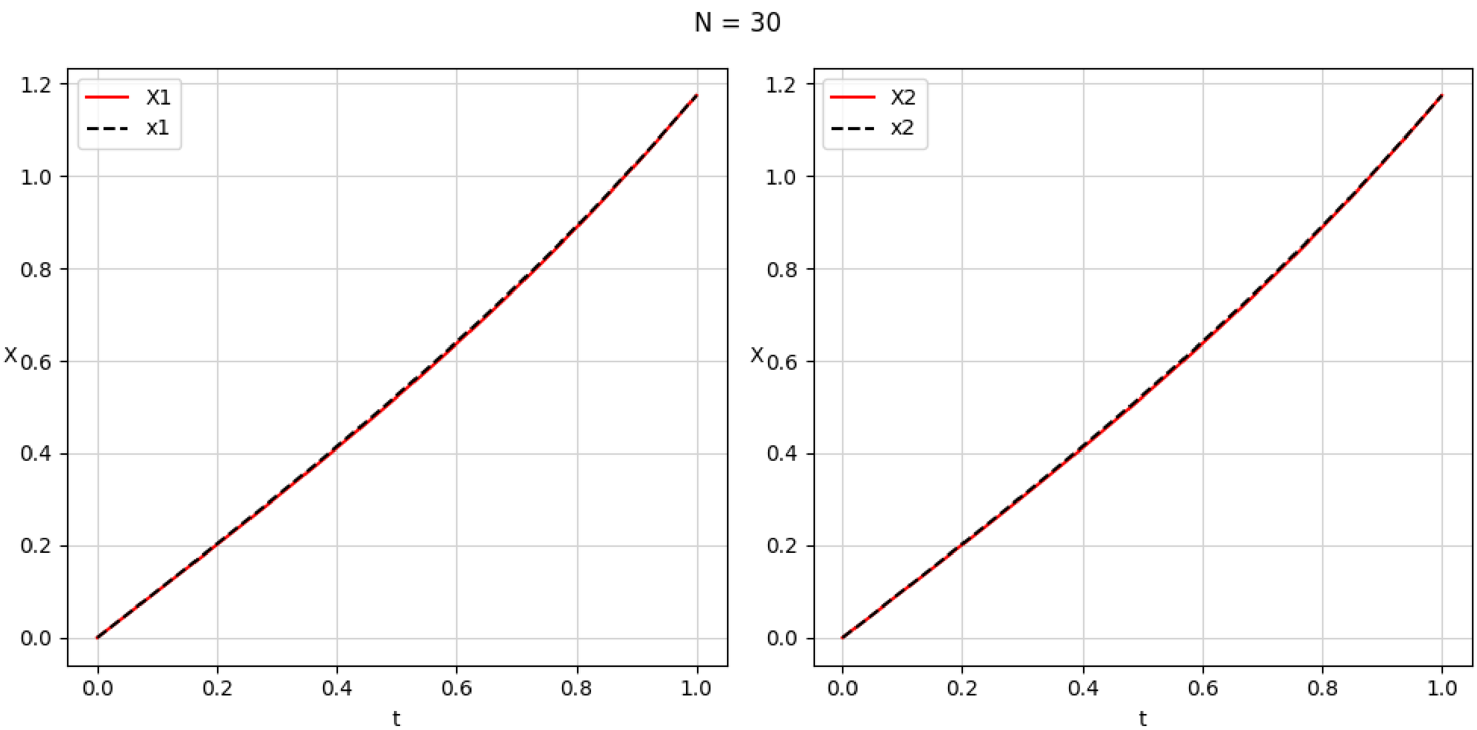


Рис. 3 – Порівняння розв’язків при N=30

N = 30, Значення функції в мінімумі = 3.651

Похибка X1 = 0.004

Похибка X2 = 0.004

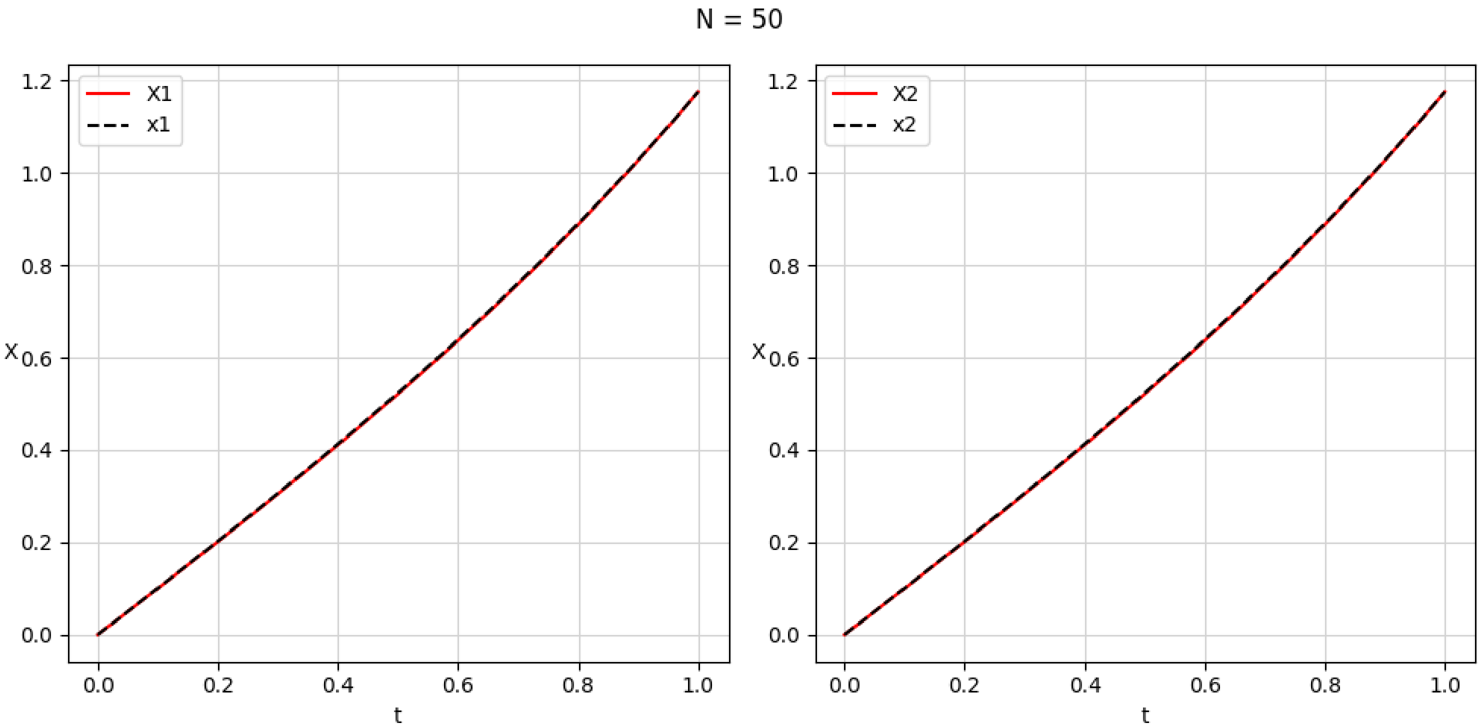


Рис. 4 – Порівняння розв’язків при N=50

N = 50, Значення функції в мінімумі = 3.641

Похибка X1 = 0.002

Похибка X2 = 0.002

Збільшення значення параметру N (кількість точок апроксимації) призвело до покращення точності, тепер аналітичний та чисельний розв'язки мають меншу похибку.

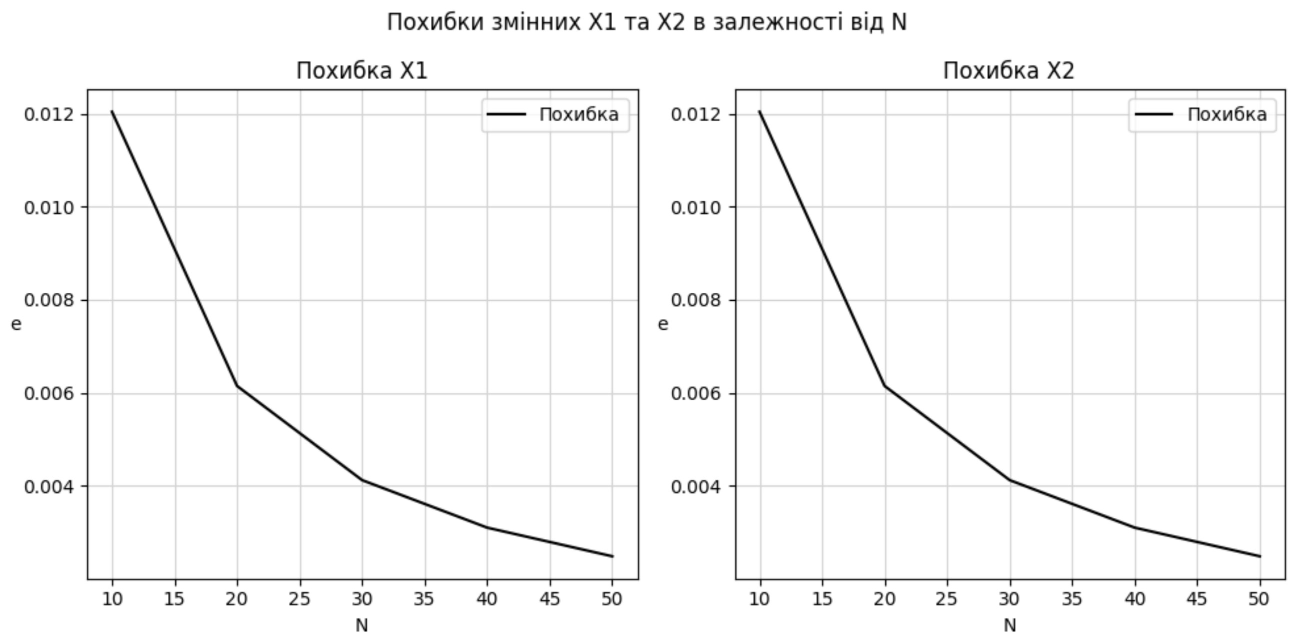


Рис. 5 – Похибки змінних x_1 та x_2 в залежності від N

Отже, як бачимо, чим більше значення N тим менша стає похибка, проте зростає складність обчислення, оскільки N відповідає за кількість точок апроксимації.

Висновок

Під час виконання даної лабораторної роботи було вивчено та запрограмовано задачу варіаційного числення.

А також проаналізовано аналітичні та чисельні методи вирішення подібних задач. Графіки наближеного та теоретичного розв'язків збігаються з дуже малою різницею.

З інкрементом кількості точок апроксимації, точність та складність обчислень збільшується, а значення цільової функції зменшується.

Додаток А – Код програми

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize


PRECISION = 3
np.set_printoptions(precision=PRECISION)


K = 100
alpha = 0
beta = 1
a1 = 0
b1 = np.sinh(1)
a2 = 0
b2 = np.sinh(1)


def get_f(x, dt, a1, b1, a2, b2):
    x1 = np.concatenate(([a1], x[:len(x) // 2], [b1]))
    x2 = np.concatenate(([a2], x[len(x) // 2:], [b2]))

    f = 0
    for i in range(0, len(x1)-1):
        f += (x1[i]**2 + x2[i]**2 + 2*(x1[i+1]-x1[i])/dt*(x2[i+1]-x2[i])/dt)
    return f * dt


def getX1(t):
    return np.sinh(t)


def getX2(t):
    return np.sinh(t)


def plot_e():
    e_t = [i for i in range(10, 55, 10)]
    e1_list = list()
    e2_list = list()
    for e in e_t:
        dt = (beta - alpha) / e
        t = np.linspace(alpha, beta, e)
        initial_X = np.zeros(2 * e - 4)
        result = minimize(get_f, initial_X, args=(dt, a1, b1, a2, b2))
```

```
        x = result.x
        x1 = np.concatenate(([a1], x[:len(x) // 2], [b1]))
        x2 = np.concatenate(([a2], x[len(x) // 2:], [b2]))
        e1 = max(abs(x1 - [getX1(ti) for ti in t]))
        e2 = max(abs(x2 - [getX2(ti) for ti in t]))
        e1_list.append(e1)
        e2_list.append(e2)

    fig, axs = plt.subplots(1, 2, figsize=(10, 5))
    fig.suptitle('Похибки змінних X1 та X2 в залежності від N')

    axs[0].plot(e_t, e1_list, label='Похибка', color='black')
    axs[0].set_xlabel('N')
    axs[0].set_ylabel('e', rotation=0)
    axs[0].legend()
    axs[0].set_title('Похибка X1')
    axs[0].grid(color='lightgrey')

    axs[1].plot(e_t, e2_list, label='Похибка', color='black')
    axs[1].set_xlabel('N')
    axs[1].set_ylabel('e', rotation=0)
    axs[1].legend()
    axs[1].set_title('Похибка X2')
    axs[1].grid(color='lightgrey')

    plt.tight_layout()
    plt.show()


def var_24(N):
    initial_X = np.zeros(2*N - 4)
    result = minimize(get_f, initial_X, args=(dt, a1, b1, a2, b2))
    x = result.x
    print(f'\n\nN = {N}, Значення функції в мінімумі = {np.round(result.fun,
PRECISION)}')

    T = np.linspace(alpha, beta, K)
    X1 = [getX1(ti) for ti in T]
    X2 = [getX2(ti) for ti in T]

    x1 = np.concatenate(([a1], x[:len(x) // 2], [b1]))
    x2 = np.concatenate(([a2], x[len(x) // 2:], [b2]))

    fig, axs = plt.subplots(1, 2, figsize=(10, 5))
    fig.suptitle(f'N = {N}')

    axs[0].plot(T, X1, label='X1', color='red')
```

```python
    axs[0].plot(t, x1, label='x1', color='black', linestyle='dashed')
    axs[0].set_xlabel('t')
    axs[0].set_ylabel('X', rotation=0)
    axs[0].legend()
    axs[0].grid(color='lightgrey')

    axs[1].plot(T, X2, label='X2', color='red')
    axs[1].plot(t, x2, label='x2', color='black', linestyle='dashed')
    axs[1].set_xlabel('t')
    axs[1].set_ylabel('X', rotation=0)
    axs[1].legend()
    axs[1].grid(color='lightgrey')

    e1 = max(abs(x1 - [getX1(ti) for ti in t]))
    e2 = max(abs(x2 - [getX2(ti) for ti in t]))

    print(f'\nПохибка X1 = {round(e1, PRECISION)}')
    print(f'Похибка X2 = {round(e2, PRECISION)}')

    plt.tight_layout()
    plt.show()


if __name__ == '__main__':
    for N in [10, 30, 50]:
        dt = (beta - alpha) / N
        t = np.linspace(alpha, beta, N)
        var_24(N)
    plot_e()
```