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-01
Романецький М. С.

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 $\lambda \ge 0$:

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

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:

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

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

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

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,

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

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

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

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

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).

Відповідно варіанту №16 було обрано наступну задачу варіаційного числення:

16. $$\int_0^{1/2} x^{-1}(1+(x')^2)^{\frac{1}{2}}\,dt \to extr\;,\; x(0)=1\,,\;\; x(1/2)=\sqrt{5/2}.$$

Основна частина

Задача варіаційного числення

Зафіксуємо наступні параметри:

```
N = 10              #  Число точок апроксимації
A = 0               #  Нижня межа інтегрування
B = 1/2             #  Верхня межа інтегрування
YA = 1              #  x(0) = 1
YB = np.sqrt(5/2)   #  x(1/2) = sqrt(5/2)
FUNC = "x**(-1)*(1+(x')**2)**(1/2)"
```

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

Чисельний (наближений) розв'язок будемо знаходити функцією scipy.optimize.minimize, яка використовує градієнтний метод оптимізації BFGS за замовчанням.

Аналітичний (точний) розв'язок було знайдено вручну розв'язавши відповідну задачу варіаційного числення. Розв'язок наведено в розділі Додаток А.

Аналітичний розв'язок: $x(t) = \pm\sqrt{\frac{65}{16} - (t - \frac{7}{4})^2}$

Тому на графіку зобразимо 2 розв'язки (додатній та від'ємний):

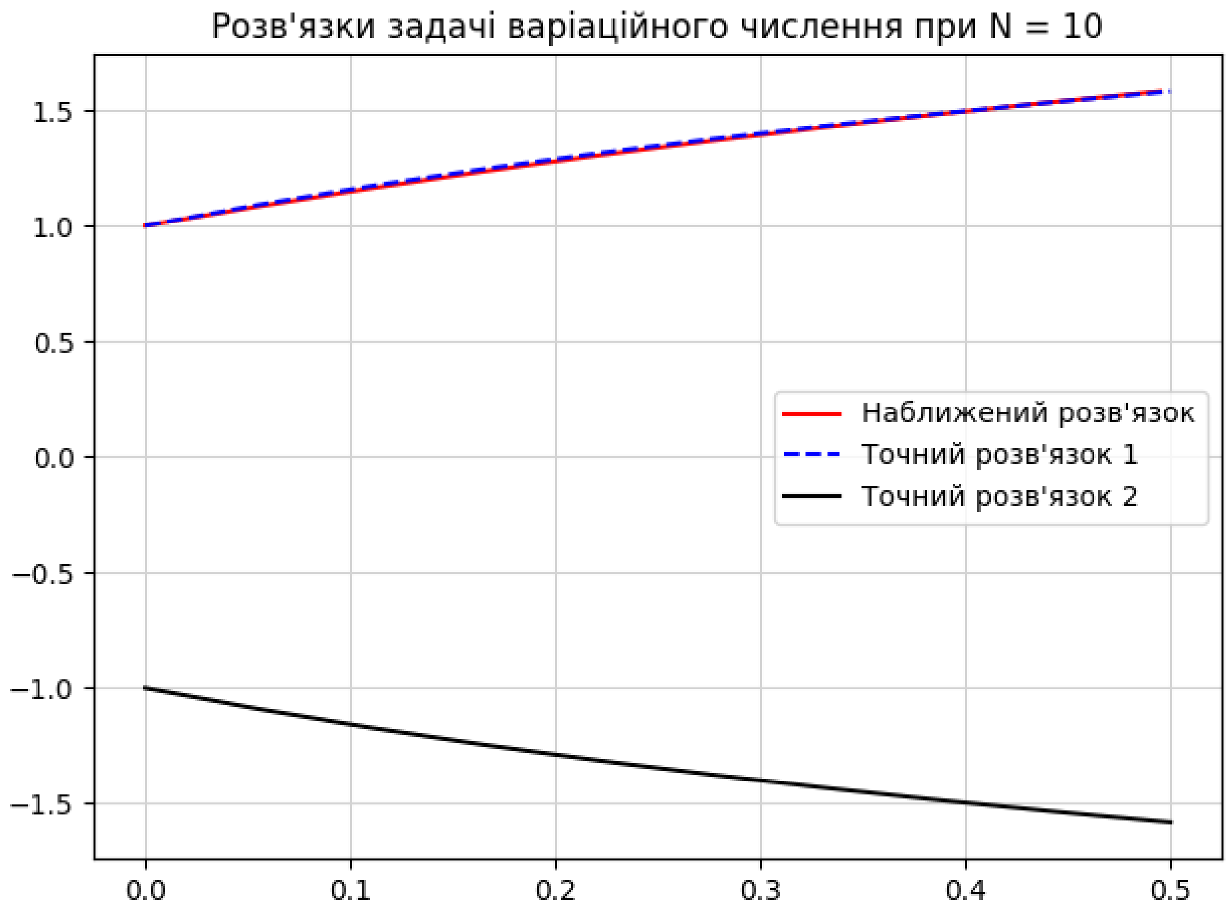


Рис. 1 – Розв'язки задачі варіаційного числення при N = 10

Як бачимо, наближений розв'язок співпадає з додатнім точним розв'язком. Отже, від'ємний далі буде ігноровано.

```
Задача варіаційного числення для N = 10

Значення мінімізованої функції: 0.068

Похибка (Mean Squared Error) = 7.394769006296111e-05

Витрачено часу: 0.97 секунд
```

Спробуємо збільшити кількість точок апроксимації та проаналізувати результати.

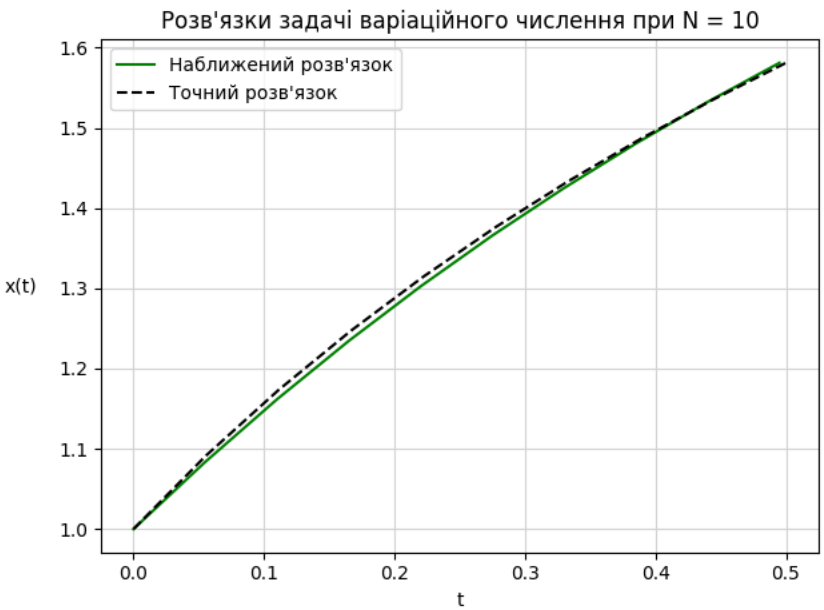


Рис. 2 – Розв'язки задачі варіаційного числення при N = 10

```
Задача варіаційного числення для
N = 10

Значення мінімізованої функції:
0.068

Похибка (Mean Squared Error) =
7.394769006296111e-05

Витрачено часу: 0.98 секунд
```

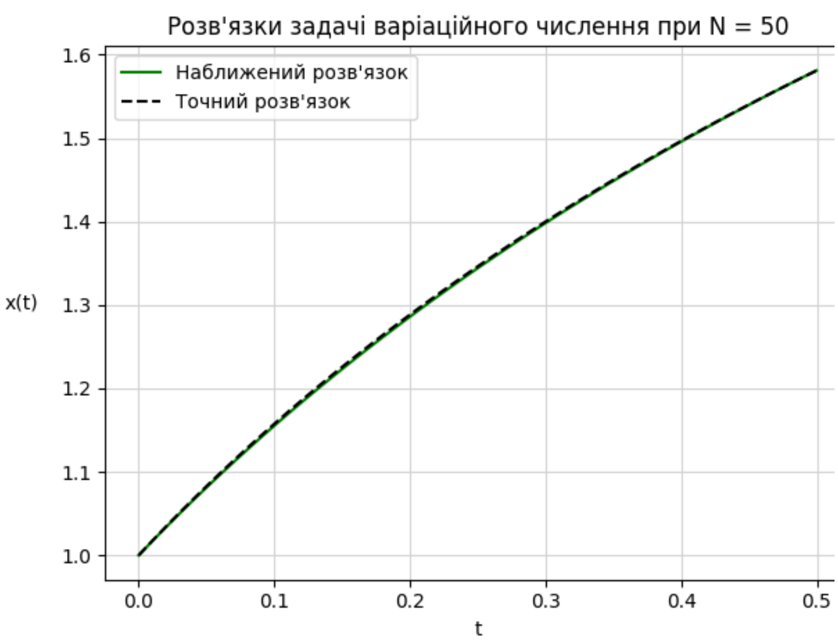


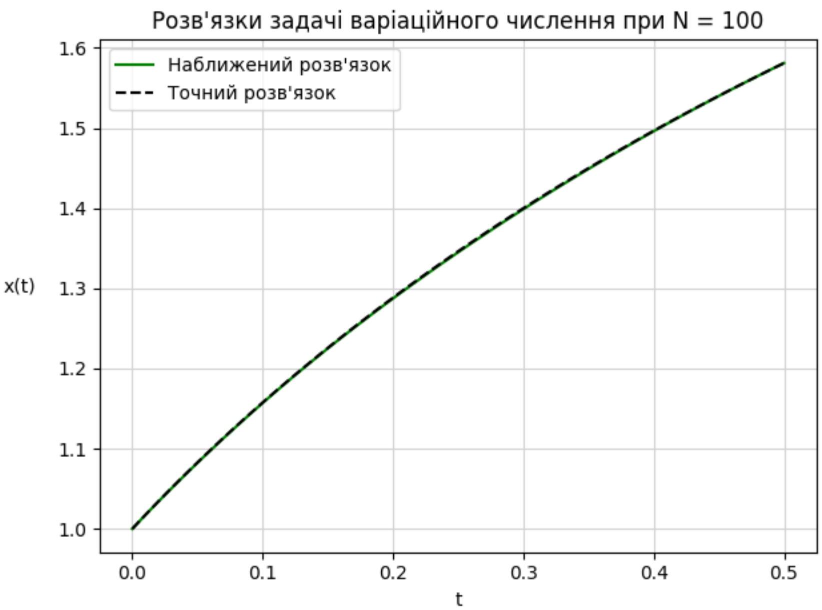
Рис. 3 – Розв'язки задачі варіаційного числення при N = 50

```
Задача варіаційного числення для
N = 50

Значення мінімізованої функції:
0.012

Похибка (Mean Squared Error) =
3.1762700913567677e-06

Витрачено часу: 36.07 секунд
```




```
Задача варіаційного числення для
N = 100

Значення мінімізованої функції:
0.006

Похибка (Mean Squared Error) =
6.203098378462466e-07

Витрачено часу: 235.68 секунд
```

Рис. 4 – Розв'язки задачі варіаційного числення при N = 100

Отже, попереднім висновком є те, що підвищення значення N (кількість точок апроксимації) сприяло покращенню точності. Тому зараз аналітичний та наближений розв'язки візуально не відрізняються. Дослідимо детальніше:

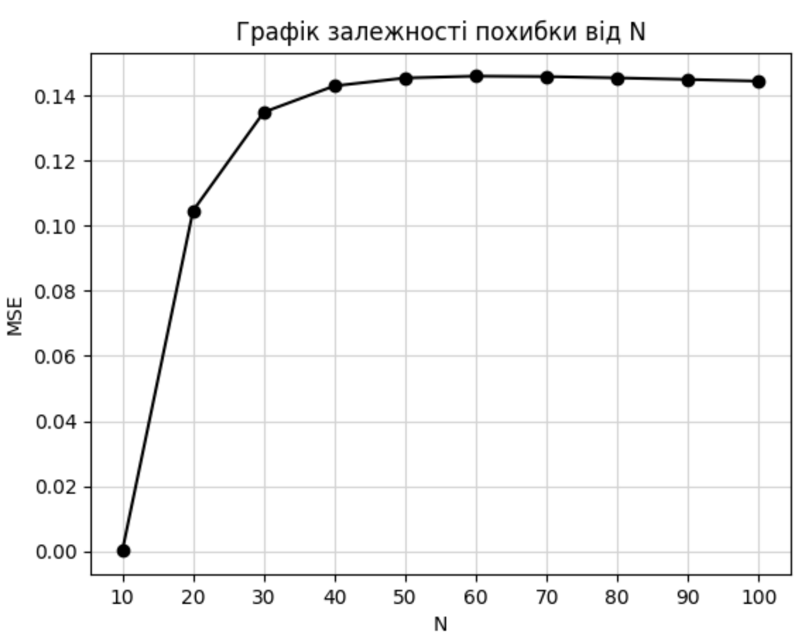


Рис. 5 – Графік залежності похибки від N

Як бачимо, після N=60 збільшення параметру не давало особливого результат, навпаки, похибка ставала трохи більшою.

Нехай параметр регуляризації буде рівним 0.1:

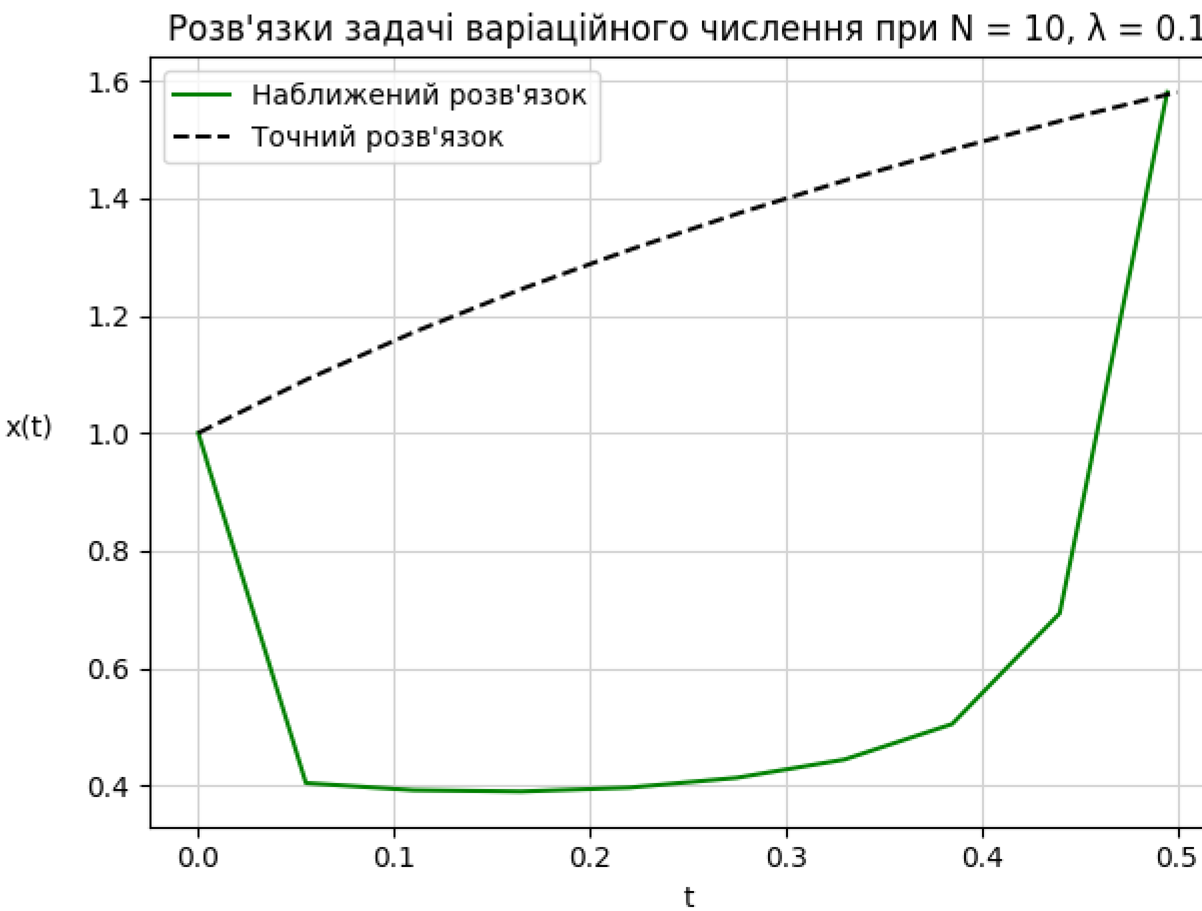


Рис. 6 – Розв'язки задачі варіаційного числення при N = 10 та lmd = 0.1

```
Задача варіаційного числення для N = 10 та lmd = 0.1

Значення мінімізованої функції: 0.608

Похибка (Mean Squared Error) = 0.6242351704091991

Витрачено часу: 1.04 секунд
```

Як бачимо, наближений розв’язок став значно гіршим за такого параметру регуляризації. Дослідимо що буде, якщо збільшити параметр до 0.5.

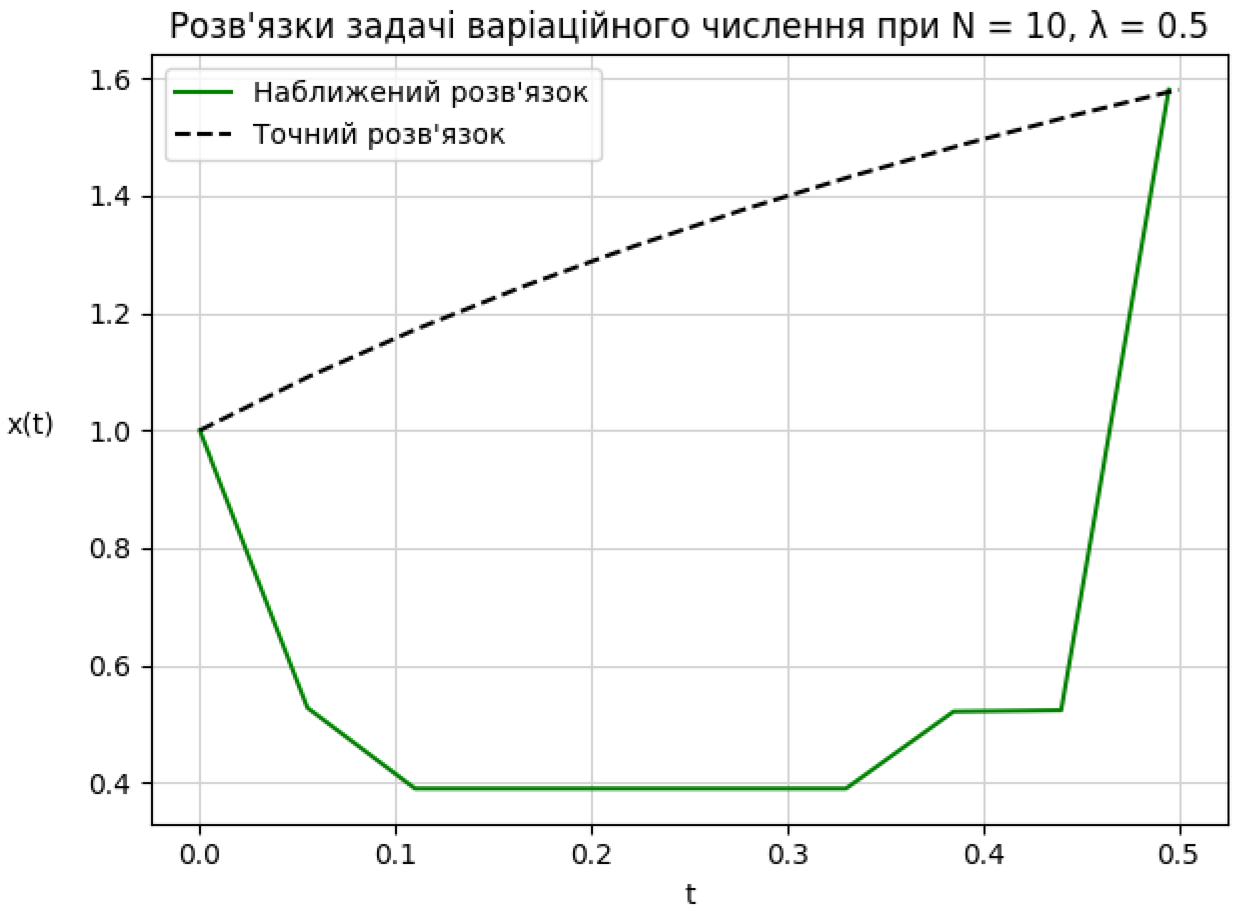


Рис. 7 – Розв'язки задачі варіаційного числення при N = 10 та lmd = 0.5

```
Задача варіаційного числення для N = 10 та lmd = 0.5

Значення мінімізованої функції: 1.587

Похибка (Mean Squared Error) = 0.6539098009117215

Витрачено часу: 1.68 секунд
```

Отже, збільшення параметру лямбда навпаки погіршило результати. Скоріше за все це може бути пов’язано тим що лямбда – це штраф до функції втрат, який допомагає зменшити складність моделі і зробити її більш узагальненою. Спробуємо зменшити параметр лямбда.

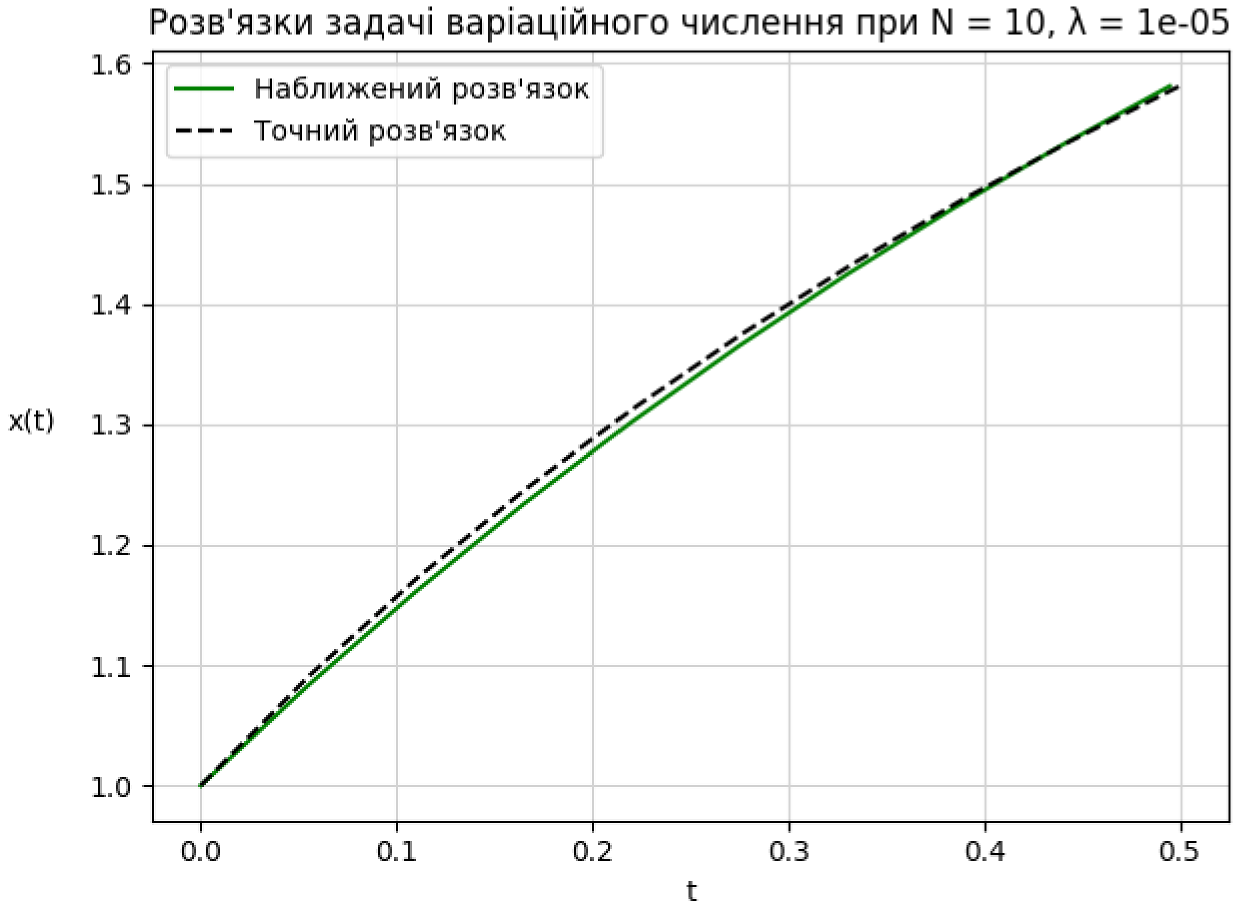


Рис. 7 – Розв'язки задачі варіаційного числення при N = 10 та lmd = 1e-05

```
Задача варіаційного числення для N = 10 та lmd = 1e-05

Значення мінімізованої функції: 0.069

Похибка (Mean Squared Error) = 8.106211872171318e-05

Витрачено часу: 1.47 секунд
```

Можна зробити висновок, що параметр лямбда дає можливість зробити модель більш узагальненою, але якщо треба мінімізувати похибку і наблизити наближений розв’язок до точного, то варто зменшити параметр лямбда.

Висновок

В ході виконання цієї лабораторної роботи було досліджено задачу варіаційного числення, а також аналітичні та наближені методи її вирішення.

Збільшення параметру N покращує точність обчислень, проте в ході досліджень було з’ясовано, що на певному етапі точність почала падати.

Було розв’язано регуляризаційну задачу і в ході досліджень було з’ясовано, що параметр лямбда погано впливав на розв’язки задачі та лише погіршував точність розв’язків базової задачі варіаційного числення. Чим менше буде штраф, тим точніше буде розв’язок.

Додаток А – Аналітичний розв’язок варіаційної задачі

$$\int_0^{1/2} x^{-1}(1+(x')^2)^{1/2}\,dt \to extr \qquad x(0)=1$$
$$x(1/2)=\sqrt{5/2}$$

$$L = x^{-1}(1+(x')^2)^{1/2} \qquad L'_x = -\frac{\sqrt{1+(x')^2}}{x^2}$$

$$L'_{x'} = \frac{\cancel{2}x'}{\cancel{2}x\sqrt{1+(x')^2}} = \frac{x'}{x}(1+(x')^2)^{-\frac{1}{2}}$$

$$\frac{d}{dt}L'_x = x'\left(-\frac{x'}{x^2}(1+(x')^2)^{-\frac{1}{2}}\right) + \left(\frac{(1+(x')^2)^{-\frac{1}{2}}}{x} - \frac{x'}{x}\frac{1}{\cancel{2}}\cdot\right.$$

$$\left.\cdot(1+(x')^2)^{-\frac{3}{2}}\cdot\cancel{2}x'\right)x'' = -\frac{(x')^2}{x^2}(1+(x')^2)^{-\frac{1}{2}} +$$

$$+\frac{(1+(x')^2)^{-\frac{1}{2}}}{x}x''\left(1-(x')^2(1+(x')^2)^{-1}\right) =$$

$$= \frac{(1+(x')^2)^{-\frac{1}{2}}}{x}\left(-\frac{(x')^2}{x} + x'' - \frac{x''\cdot(x')^2}{1+(x')^2}\right) =$$

$$= \frac{(1+(x')^2)^{-\frac{1}{2}}}{x}\cdot\left(-\frac{(x')^2}{x} + \frac{x''+x''(x')^2-x''(x')^2}{1+(x')^2}\right) =$$

$$= x''\frac{(1+(x')^2)^{-\frac{3}{2}}}{x} - (x')^2\cdot\frac{(1+(x')^2)^{-\frac{1}{2}}}{x^2} = 0 \;\Big/\; x^2(1+(x')^2)^{\frac{3}{2}}$$

$$\left[\begin{array}{l} x=0 \\ 1+(x')^2=0 \\ -x''\cdot x + (x')^2(1+(x')^2) - (1+(x')^2)^2 = 0 \end{array}\right.$$

$$-x''\cdot x + (x')^2 + \cancel{(x')^4} - 1 - 2(x')^2 - \cancel{(x')^4} = 0$$

$$-x''\cdot x - (x')^2 - 1 = 0$$

$$x''\cdot x + (x')^2 + 1 = 0$$

$$x' = w(x)$$

$$w' = x''$$

$$w'\cdot x + w^2 + 1 = 0$$

$$x' = \pm\sqrt{\frac{\tilde{\tilde{C}}}{x^2} - 1}$$

$$\frac{dx}{dt} = \pm\sqrt{\frac{\tilde{\tilde{C}} - x^2}{x^2}}$$

$$\pm\sqrt{\frac{x^2}{\tilde{\tilde{C}} - x^2}}\,dx = dt$$

$$\frac{\pm x}{\sqrt{\tilde{\tilde{C}} - x^2}}\,dx = dt$$

$$\pm\int \frac{x\,dx}{\sqrt{\tilde{\tilde{C}} - x^2}} = \int dt$$

$$-\int \frac{-2x\,dx}{2\sqrt{\tilde{\tilde{C}} - x^2}} = -\int \frac{d(\tilde{\tilde{C}} - x^2)}{2\sqrt{\tilde{\tilde{C}} - x^2}} =$$

$$= -\sqrt{\tilde{\tilde{C}} - x^2} = t + C_1$$

$$x(0) = 1, \quad x(1/2) = \sqrt{5/2}$$

$$\tilde{\tilde{C}} - x^2 = (t + C_1)^2 \Rightarrow$$

$$\Rightarrow x = \pm\sqrt{\tilde{\tilde{C}} - (t + C_1)^2}$$

$$\tilde{\tilde{C}} - 1 = C_1^2 \qquad \tilde{\tilde{C}} = C_1 + 1$$

$$\tilde{\tilde{C}} - \frac{5}{2} = \left(\frac{1}{2} + C_1\right)^2 \Rightarrow$$

$$\Rightarrow C_1^2 - \frac{3}{2} = \frac{1}{4} + C_1^2 + C_1$$

$$C_1 = -\frac{3}{2} - \frac{1}{4} = -\frac{7}{4}$$

$$\tilde{\tilde{C}} = \frac{49}{16} + 1 = \frac{65}{16}$$

$$x(t) = \pm\sqrt{\frac{65}{16} - \left(t - \frac{7}{4}\right)^2}$$

$$x'(t) = -\frac{4t - 7}{2\sqrt{-4t^2 + 14t + 4}}$$

Додаток Б – Код програм

Romanetskiy_KM-01_Lab3:

```python
import time

import numpy as np
import scipy.optimize
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error


PRECISION = 3
np.set_printoptions(precision=PRECISION)


def x(t):
    return -1j*t


def j(args):
    global x_list
    x_list = []
    _b = B + step
    dt = (_b-A)/N

    func_str = ""
    for i in range(1, N + 2):
        x_list.append(A + (i - 1) * dt)
        func_str += "(" + FUNC.replace("x'", f"((y{i + 1}-y{i})/{dt})")\
                                .replace("x", f"(x{i})")\
                                .replace("t", f"{dt}") + f")*{dt}"
        if i != N+1:
            func_str += "+"
    func_str = func_str.replace("y", "x")
    func_str = f"(1/{N})*(" + func_str + ")"

    f_str = func_str

    f_str = f_str.replace(f'x{N}', str(YB))
    for i in range(len(args), 1, -1):
        f_str = f_str.replace(f"x{i}", str(args[i - 1]))
    f_str = f_str.replace('x1', str(YA))
    return eval(f_str)
```

```python
def variant_16(n):
    print(f"Задача варіаційного числення для N = {n}")
    x0 = np.array([1 for _ in range(n + 2)])
    res = scipy.optimize.minimize(j, x0)
    print(f"Значення мінімізованої функції: {round(res.fun, PRECISION)}")

    result = res.x[1:-2]
    result[-1] = YB
    y_list = np.concatenate([[YA], result])

    t_plot = np.linspace(A, B, n)
    analytic = [np.sqrt(65/16 - (t-7/4)**2) for t in t_plot]

    print(f"Похибка (Mean Squared Error) = {mean_squared_error(analytic, y_list)}")

    plt.plot(x_list[:-1], y_list, label="Наближений розв'язок", color='green')
    plt.plot(t_plot, analytic, label="Точний розв'язок", color='black',
linestyle='dashed')

    plt.title(f"Розв'язки задачі варіаційного числення при N = {n}")
    plt.legend()
    ax = plt.gca()
    ax.set_axisbelow(True)
    plt.grid(c='lightgrey')
    plt.xlabel('t')
    plt.ylabel('x(t)', rotation=0, labelpad=20)
    plt.tight_layout()
    print(f"Витрачено часу: {time.time() - start_time:.2f} секунд\n")
    plt.show()


def calculate_solution_without_plot(n):
    x0 = np.array([1 for _ in range(n + 2)])
    res = scipy.optimize.minimize(j, x0)

    result = res.x[1:-2]
    result[-1] = YB
    y_list = np.concatenate([[YA], result])

    t_plot = np.linspace(A, B, n)
    analytic = [np.sqrt(65/16 - (t-7/4)**2) for t in t_plot]

    error = mean_squared_error(analytic, y_list)

    return error
```

```python
def calculate_errors_and_plot(N_values):
    errors = []

    for N in N_values:
        error = calculate_solution_without_plot(N)
        errors.append(error)

    plt.figure()
    plt.plot(N_values, errors, marker='o', color='black')
    plt.title('Графік залежності похибки від N')
    plt.xlabel('N')
    plt.ylabel('MSE')
    plt.xticks(N_values)
    plt.grid(color='lightgrey')
    plt.show()


if __name__ == "__main__":
    start_time = time.time()
    N = 10              #  Число точок апроксимації
    A = 0               #  Нижня межа інтегрування
    B = 1/2             #  Верхня межа інтегрування
    YA = 1              #  x(0) = 1
    YB = np.sqrt(5/2)   #  x(1/2) = sqrt(5/2)
    FUNC = "x**(-1)*(1+(x')**2)**(1/2)"
    step = (B-A)/N

    print('\n\nЛабораторна робота №3\nРоманецький Микита\nВаріант 16\n')
    # Графіки розв'язків задачі варіаційного числення
    for N in [10, 50, 100]:
        step = (B-A)/N
        variant_16(N)

    #  Графік залежності похибки від N
    calculate_errors_and_plot(range(10, 110, 10))
```

Regular.py:

```python
import time

import numpy as np
import scipy.optimize
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
```

```
PRECISION = 3
np.set_printoptions(precision=PRECISION)


def x(t):
    return -1j*t


def j(args):
    global x_list
    x_list = []
    _b = B + step
    dt = (_b-A)/N

    func_str = ""
    for i in range(1, N + 2):
        x_list.append(A + (i - 1) * dt)
        func_str += "(" + FUNC.replace("x'", f"((y{i + 1}-y{i})/{dt})")\
                              .replace("x", f"(x{i})")\
                              .replace("t", f"{dt}") + f")*{dt}"
        if i != N+1:
            func_str += "+"
    func_str = func_str.replace("y", "x")
    func_str = f"(1/{N})*(" + func_str + ")"

    # Add regularization term
    reg_term = LAMBDA * np.sum(np.square(args))
    func_str += f" + {reg_term}"

    f_str = func_str

    f_str = f_str.replace(f'x{N}', str(YB))
    for i in range(len(args), 1, -1):
        f_str = f_str.replace(f"x{i}", str(args[i - 1]))
    f_str = f_str.replace('x1', str(YA))
    return eval(f_str)


def variant_16(n):
    print(f"Задача варіаційного числення для N = {n} та lmd = {LAMBDA}")
    x0 = np.array([1 for _ in range(n + 2)])
    res = scipy.optimize.minimize(j, x0)
    print(f"Значення мінімізованої функції: {round(res.fun, PRECISION)}")

    result = res.x[1:-2]
    result[-1] = YB
    y_list = np.concatenate([[YA], result])
```

```python
    t_plot = np.linspace(A, B, n)
    analytic = [np.sqrt(65/16 - (t-7/4)**2) for t in t_plot]

    print(f"Похибка (Mean Squared Error) = {mean_squared_error(analytic, y_list)}")

    plt.plot(x_list[:-1], y_list, label="Наближений розв'язок", color='green')
    plt.plot(t_plot, analytic, label="Точний розв'язок", color='black',
linestyle='dashed')

    plt.title(f"Розв'язки задачі варіаційного числення при N = {n}, λ = {LAMBDA}")
    plt.legend()
    ax = plt.gca()
    ax.set_axisbelow(True)
    plt.grid(c='lightgrey')
    plt.xlabel('t')
    plt.ylabel('x(t)', rotation=0, labelpad=20)
    plt.tight_layout()
    print(f"Витрачено часу: {time.time() - start_time:.2f} секунд\n")
    plt.show()


if __name__ == "__main__":
    start_time = time.time()
    N = 10              #  Число точок апроксимації
    A = 0               #  Нижня межа інтегрування
    B = 1/2             #  Верхня межа інтегрування
    YA = 1              #  x(0) = 1
    YB = np.sqrt(5/2)   #  x(1/2) = sqrt(5/2)
    LAMBDA = 10**(-5)   #  Параметр регуляризації
    FUNC = "x**(-1)*(1+(x')**2)**(1/2)"
    step = (B-A)/N

    print('\n\nЛабораторна робота №3\nРоманецький Микита\nВаріант 16\n')
    step = (B-A)/N
    variant_16(N)
```