

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи

з дисципліни «АЛГОРИТМИ І СИСТЕМИ КОМП'ЮТЕРНОЇ
МАТЕМАТИКИ 1.МАТЕМАТИЧНІ АЛГОРИТМИ»

на тему

“Розв’язання звичайних диференціальних рівнянь”

Виконав:

студент групи КМ-03

Шаповалов Г. Г.

Перевірила:

Асистент кафедри ПМА

Ковальчук-Химюк Л. О.

Зміст

Вступ.....	3
Основна частина	4
Варіант 13	4
Теоретичні відомості.....	4
Алгоритми розв’язання.....	5
Висновки	6
Відповіді на контрольні питання	7
Перелік посилань.....	8
Додаток А – Скріншоти роботи програми.....	9
Додаток В – Код програми	10

Вступ

Метою даної роботи є вивчення правил використання програмних засобів чисельного розв'язання звичайних диференціальних рівнянь, практичне розв'язання рівнянь на ЕОМ з використанням СКМ та порівняльний аналіз методів інтегрування ОДУ.

Основна частина

Варіант 13

Диференційне рівняння	Початкова умова	Проміжок інтегрування	Крок h
$y'' + y = 1 + e^x$	$y(0) = 2.5$ $y'(0) = 1.5$	[0; 1]	0.1

Метод розв'язання – метод Рунге-Кутта 4-го порядку [4]

Теоретичні відомості

Метод Рунге-Кутта [4] використовується для ітеративного обчислення значень рішення диференціального рівняння у вигляді $y=f(x, y)$, використовуючи початкові умови (x_0, y_0) . Метод Рунге-Кутта 4-го порядку забезпечує точність 4-го порядку. При його застосуванні виникає питання вибору конкретних формул і кроку мережі.

Якщо права частина диференціального рівняння - це функція, яка є неперервною і обмеженою разом із своїми четвертими похідними, то метод Рунге-Кутта 4-го порядку дає добрі результати. Тут важливо вибрати відповідні формули для кожного конкретного випадку і визначити оптимальний крок мережі. Зменшення кроку призводить до швидшого зростання точності.

У випадку, коли права частина не має обмежених четвертих похідних, максимальний порядок точності методу Рунге-Кутта не досягається. Тут рекомендовано використовувати обчислювальні схеми меншого порядку точності, який відповідає порядку похідних. Крок мережі обирається настільки малим, щоб забезпечити необхідну точність розрахунків. Для визначення точності часто використовується апостеріорна оцінка похибки, яка базується на використанні двох мереж і правила Рунге. Формула методу Рунге-Кутта 4-го порядку:

$$\begin{cases} y_{k+1} = y_k + (k_1 + 2k_2 + 2k_3 + k_4)/6 \\ k_1 = hf(x_k, y_k) \\ k_2 = hf\left(x_k + \frac{h}{2}, y_k + k_1/2\right) \\ k_3 = hf\left(x_k + \frac{h}{2}, y_k + \frac{k_2}{2}\right) \\ k_4 = hf(x_k + h, y_k + k_3) \end{cases}$$

Алгоритми розв'язання

Імпортуємо бібліотеки Oct2py [1], pandas [2], matplotlib [3]

Створюємо власну функцію, яка приймає дані і рахує за формулою Рунге-Кутта [4] 4-го порядку. У цю функцію передаємо дані індивідуального варіанту, а саме, початкові умови, проміжок інтегрування та крок. Оскільки задане диференційне рівняння є рівнянням другого порядку, то треба ввести заміну

$$\begin{cases} w = y' \\ w' = y'' \end{cases}$$

Отже рівняння з заміною виглядає наступним чином:

$$w' + w = 1 + e^x$$

У кінці виводимо 2 графіки створений бібліотекою matplotlib:

1. Розв'язки мовою Python
2. Розв'язки мовою Octave

Висновки

Рівняння $y'' + y = 1 + e^x$ після введення заміни

$$\begin{cases} w = y' \\ w' = y'' \end{cases}$$

Виглядає наступним чином $w' + w = 1 + e^x$

Заміна треба для можливості застосування методу Рунге-Кутта

У результаті розв'язання рівняння було знайдено такі результати:

Розв'язок Python:

	x	y'(x)	y(x)
0	0.0	1.500000	2.500000
1	0.1	1.307742	2.640256
2	0.2	1.131871	2.762095
3	0.3	0.973606	2.867218
4	0.4	0.834002	2.957439
5	0.5	0.713961	3.034670
6	0.6	0.614258	3.100909
7	0.7	0.535556	3.158222
8	0.8	0.478431	3.208740
9	0.9	0.443390	3.254645
10	1.0	0.430892	3.298169

Розв'язок Octave:

	x	y'(x)	y(x)
0	0.0	1.500000	2.500000
1	0.1	1.307742	2.640256
2	0.2	1.131871	2.762095
3	0.3	0.973606	2.867218
4	0.4	0.834002	2.957439
5	0.5	0.713961	3.034670
6	0.6	0.614258	3.100909
7	0.7	0.535556	3.158222
8	0.8	0.478431	3.208740
9	0.9	0.443390	3.254645
10	1.0	0.430893	3.298169

Отримано однакові результати для розв'язку диференціального рівняння за допомогою Python та Octave. Значення функції та її похідної на кожному кроці наведені вище. Схожість результатів обчислень означає правильність вибору методу для даного диференціального рівняння.

Відповіді на контрольні питання

- 1. Що значить вирішити диференціальні рівняння чисельним методом?**
Вирішення диференціальних рівнянь чисельним методом означає знаходження наближеного чисельного розв'язку диференціального рівняння за допомогою обчислювальних методів. Це необхідно, коли аналітичне розв'язання диференціального рівняння неможливе або складне, і ми використовуємо чисельні методи для наближеного знаходження функції, яка задовольняє рівняння.
- 2. Охарактеризуйте метод Рунге-Кутта четвертого порядку –** Метод Рунге-Кутта четвертого порядку - це чисельний метод для розв'язання звичайних диференціальних рівнянь. Він є одним з найпоширеніших методів і має четвертий порядок точності. Метод використовує ітерації для оновлення значень функції на кожному кроці, використовуючи середні значення на кількох під кроках для покращення точності результатів.
- 3. Дайте порівняльну характеристику методів Рунге-Кутта і методів прогнозу-корекції –** Методи Рунге-Кутта та методи прогнозу-корекції - це дві різні класи чисельних методів для розв'язання диференціальних рівнянь. Методи Рунге-Кутта використовують ітерації та середні значення для оновлення функцій на кожному кроці, тоді як методи прогнозу-корекції використовують два окремі кроки: прогноз та корекцію. Обидва підходи мають свої переваги та недоліки, і вибір залежить від конкретного завдання та вимог до точності.
- 4. Опишіть процедуру автоматичного обрання кроку в методі Рунге-Кутта –** Процедура автоматичного обрання кроку в методі Рунге-Кутта називається "кроком адаптації". Вона полягає в тому, щоб динамічно змінювати розмір кроку в залежності від поточних умов розв'язку. На кожному кроці обчислюється деякий показник точності, і наступний крок вибирається так, щоб досягнути заданого рівня точності. Це дозволяє ефективно розв'язувати різні диференціальні рівняння з різними характеристиками розв'язків.
- 5. Як оцінюють точність отриманого розв'язку в методах прогнозу і корекції?** Оцінка точності отриманого розв'язку в методах прогнозу і корекції відбувається шляхом порівняння результату прогнозу (наближеного розв'язку) з результатом корекції (кінцевим розв'язком). Точність оцінюється за допомогою різниці між цими двома значеннями, і зазвичай використовується показник помилки, такий як середньоквадратична помилка або абсолютна помилка. Мета полягає в тому, щоб зробити прогнози та корекції настільки точними, наскільки це можливо, і зменшити вплив наближень на кінцевий результат.

Перелік посилань

1. Oct2py - <https://pypi.org/project/oct2py/>
2. Pandas - <https://pandas.pydata.org>
3. Matplotlib - <https://matplotlib.org>
4. Метод Рунге-Кутта - <https://matica.org.ua/metodichki-i-knigi-po-matematike/vychislitelnaia-matematika/7-4-metod-runge-kutta-4-poriadka>

Додаток А – Скріншоти роботи програми

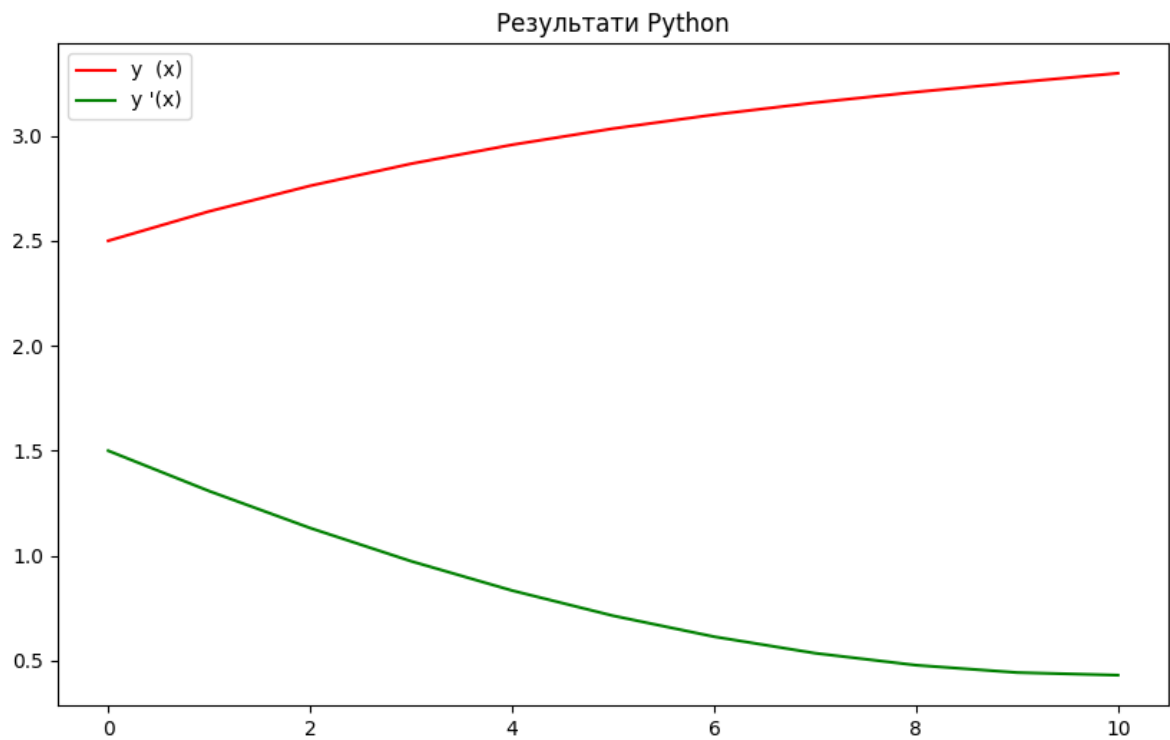


Рис. 1 – Графік результатів на Python

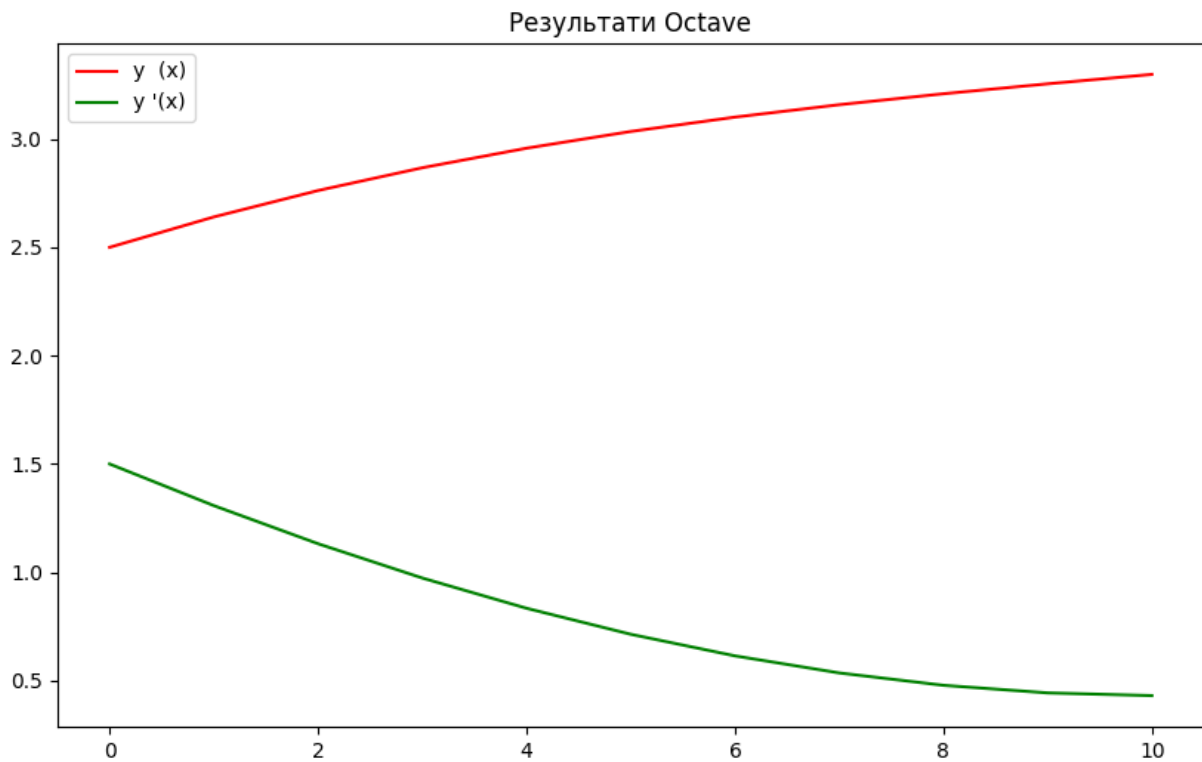


Рис. 2 – Графік результатів на Octave

Додаток В – Код програми

Вміст файлу main.py :

```
from oct2py import Oct2Py
import pandas as pd
import matplotlib.pyplot as plt

def equation_func(x, y, z):
    return 1 + 2.71828 ** x - y - z

def runge_kutta(x0, y0, z0, h, x_target, equation_func):
    x = x0
    y = y0
    z = z0

    results = [('x', 'y\'(x)', 'y(x)')]

    while x <= x_target:
        results.append((x, z, y))

        k1y = h * z
        k1z = h * equation_func(x, y, z)

        k2y = h * (z + 0.5 * k1z)
        k2z = h * equation_func(x + 0.5 * h, y + 0.5 * k1y, z + 0.5 * k1z)

        k3y = h * (z + 0.5 * k2z)
        k3z = h * equation_func(x + 0.5 * h, y + 0.5 * k2y, z + 0.5 * k2z)

        k4y = h * (z + k3z)
        k4z = h * equation_func(x + h, y + k3y, z + k3z)

        y = y + (1/6) * (k1y + 2 * k2y + 2 * k3y + k4y)
        z = z + (1/6) * (k1z + 2 * k2z + 2 * k3z + k4z)
        x = x + h

    return results

x0 = 0
y0 = 2.5
z0 = 1.5
h = 0.1
x_target = 1.0

results = runge_kutta(x0, y0, z0, h, x_target, equation_func)

# Створюємо датафрейм з результатами
```

```

df_python = pd.DataFrame(results[1:], columns=results[0])

print('\n\nШаповалов Г. Г. Варіант 13 Лаб 3\n')
print('\nРозв\'язок Python:\n')
print(df_python)

# Запускаємо Octave
ос = Oct2Py()

# Визначаємо функцію рівняння
ос.eval("""
function result = equation_func(x, y, z)
    result = 1 + exp(x) - y - z;
end
""")

# Визначаємо функцію для методу Рунге-Кутта
ос.eval("""
function results = runge_kutta_octave(x0, y0, z0, h, x_target)
    x = x0;
    y = y0;
    z = z0;

    results = [];

    while x < x_target
        results = [results; [x, z, y]];

        k1y = h * z;
        k1z = h * equation_func(x, y, z);

        k2y = h * (z + 0.5 * k1z);
        k2z = h * equation_func(x + 0.5 * h, y + 0.5 * k1y, z + 0.5 * k1z);

        k3y = h * (z + 0.5 * k2z);
        k3z = h * equation_func(x + 0.5 * h, y + 0.5 * k2y, z + 0.5 * k2z);

        k4y = h * (z + k3z);
        k4z = h * equation_func(x + h, y + k3y, z + k3z);

        y = y + (1/6) * (k1y + 2 * k2y + 2 * k3y + k4y);
        z = z + (1/6) * (k1z + 2 * k2z + 2 * k3z + k4z);
        x = x + h;
    end
end
""")

# Визначаємо початкові значення
x0 = 0.0
y0 = 2.5
z0 = 1.5

```

```
h = 0.1
x_target = 1.0

# Вивикаємо функцію методу Рунге-Кутта
results_octave = os.runge_kutta_octave(x0, y0, z0, h, x_target)

# Створюємо датафрейм з результатами
df_octave = pd.DataFrame(results_octave[0:, :], columns=["x", "y\'(x)", "y(x)"])

print('\n\nРозв\'язок Octave:\n')
print(df_octave)

# Створюємо графік для результатів Python
plt.figure(figsize=(10, 6))
plt.plot(df_python.index, df_python['y(x)'], label='y (x)', color='red')
plt.plot(df_python.index, df_python['y\'(x)'], label='y \'(x)', color='green')
plt.title('Результати Python')
plt.legend()
plt.show()

# Створюємо графік для результатів Octave
plt.figure(figsize=(10, 6))
plt.plot(df_octave.index, df_octave['y(x)'], label='y (x)', color='red')
plt.plot(df_octave.index, df_octave['y\'(x)'], label='y \'(x)', color='green')
plt.title('Результати Octave')
plt.legend()
plt.show()
```