

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-01

Іваник Ю. П.

Перевірила:

Асистент кафедри ПМА

Ковальчук-Химюк Л. О.

Зміст

Вступ.....	3
Основна частина	4
Варіант 7	4
Вимоги до ПЗ	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 7

Диференційне рівняння	Початкова умова	Проміжок інтегрування	Крок h
$y'' + 2y' + 2y = xe^x$	$y(0) = 0$ $y'(0) = 0$	[0; 1.5]	0.1

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

Вимоги до ПЗ

1. Реалізувати перевірки на некоректний ввід (порожнє введення, символи замість числа, тощо).
2. У програмі повинно бути передбачено можливість гнучкого налагодження розмірності розв'язуваної задачі.
3. Має зберігатись логічно правильне розв'язання (нижня межа інтегрування не має перевищувати верхню, тощо)

Теоретичні відомості

Суть методу Рунге-Кутта в покроковому обчисленні значень рішення $y=y(x)$ диференціального рівняння виду $y = f(x, y)$ з початковою умовою (x_0, y_0) . Метод Рунге-Кутта 4-го порядку є методом 4-го порядку точності. При використанні формул методу Рунге-Кутта виникають питання: якими з формул доцільно користуватись у кожному конкретному випадку, як вибирати крок сітки? Якщо права частина диференційного рівняння - функція, неперервна й обмежена разом із своїми четвертими похідними, тоді добрі результати дає метод Рунге-Кутта четвертого порядку завдяки швидкому зростанню точності зі зменшенням кроку сітки. Якщо права частина не має обмежених четвертих похідних, тоді максимального порядку точності такої схеми не можна досягнути. В цьому випадку доцільно використовувати обчислювальні схеми методу меншого порядку точності, який відповідає порядку похідних. Крок сітки вибирають настільки малим, щоб забезпечити необхідну точність розрахунків. Основним практичним способом одержання заданої точності є апостеріорна оцінка похибки. Для її відшукування розрахунок проводять на

двох сітках з кроками та застосовують правило 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

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

Цей алгоритм розв'язання реалізований на мові Python та на мові Octave.

У кінці ми виводимо 2 графіки:

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

Висновки

У результаті розв'язку диференційного рівняння досягнуто таких результатів:

Розв'язок Python:

x	f(x)	f'(x)
0.0	0.000000	0.000000
0.1	0.005005	0.000166
0.2	0.020072	0.001336
0.3	0.045378	0.004521
0.4	0.081234	0.010762
0.5	0.128111	0.021134
0.6	0.186656	0.036771
0.7	0.257711	0.058881
0.8	0.342338	0.088764
0.9	0.441834	0.127842
1.0	0.557765	0.177676
1.1	0.691985	0.240002
1.2	0.846671	0.316752
1.3	1.024353	0.410099
1.4	1.227954	0.522483
1.5	1.460833	0.656661

Розв'язок Octave:

x	f(x)	f'(x)
0.0	-0.004083	-0.000287
0.1	0.001749	-0.000487
0.2	0.017554	0.000394
0.3	0.043508	0.003361
0.4	0.079929	0.009444
0.5	0.127292	0.019711
0.6	0.186250	0.035287
0.7	0.257651	0.057374
0.8	0.342561	0.087266
0.9	0.442286	0.126377
1.0	0.558396	0.176267
1.1	0.692751	0.238662
1.2	0.847533	0.315494
1.3	1.025279	0.408930
1.4	1.228915	0.521409
1.5	1.461807	0.655684

Як бачимо, результати є ідентичними, щоб переконатись у цьому не занурюючись у порівняння чисел можна подивитись на графіки розв'язків у Додатку А.

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

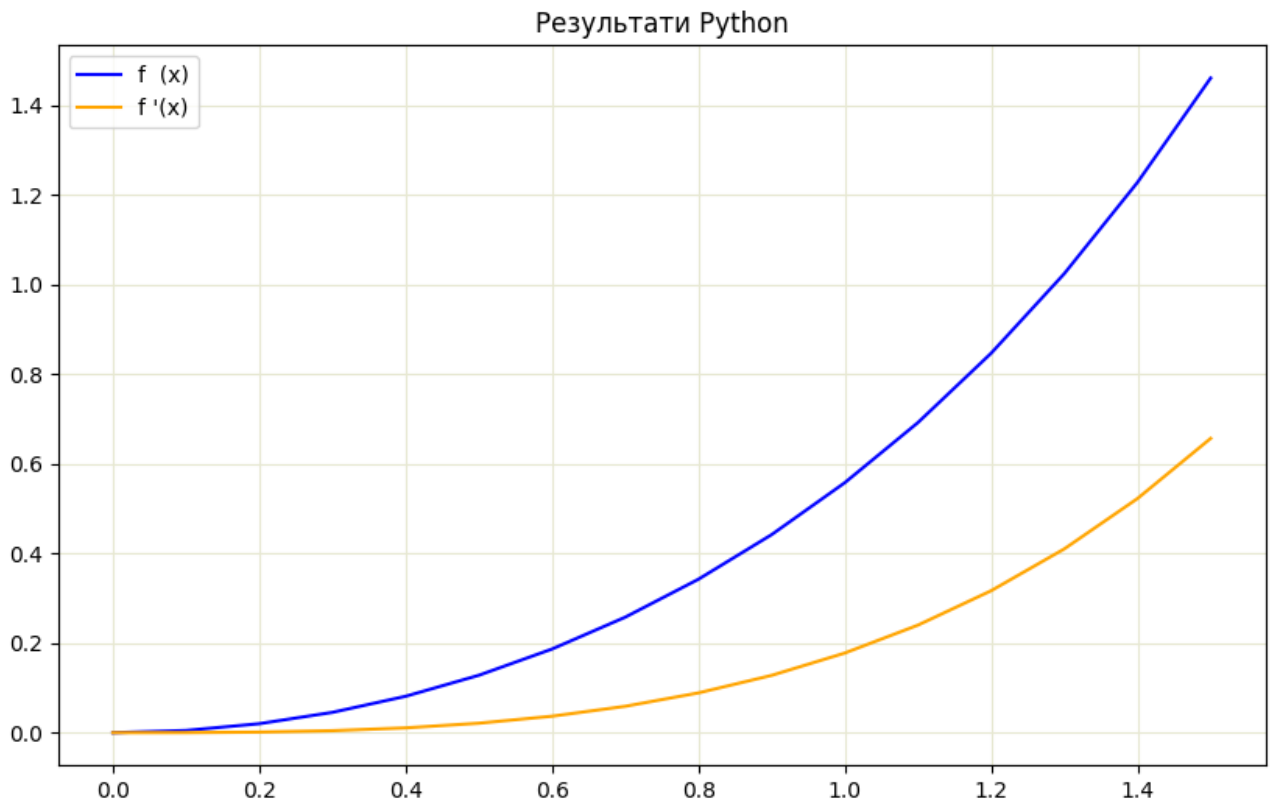


Рис. 1 – Графічне представлення результатів отриманих мовою Python

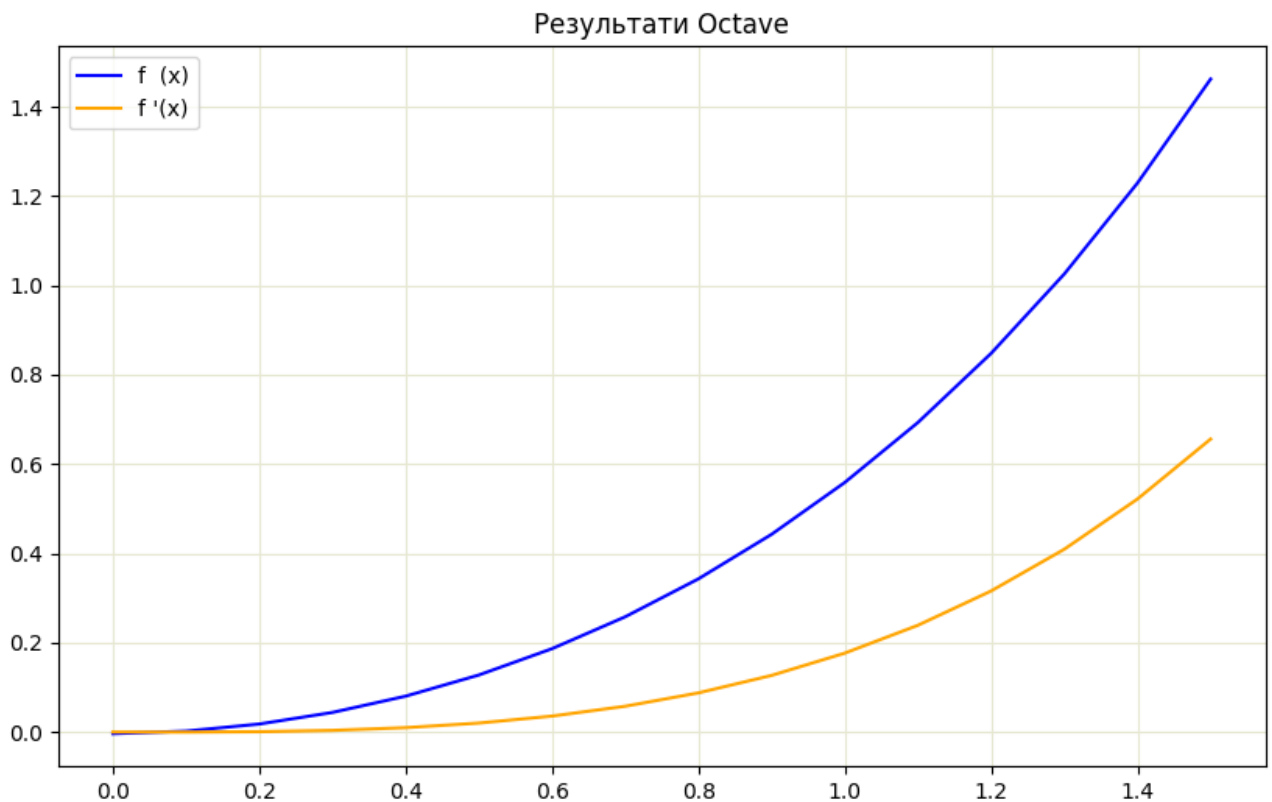


Рис. 2 – Графічне представлення результатів отриманих мовою Octave

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 x * (2.71828 ** x) - 2 * z - 2 * y

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

    results = [('x', 'f(x)', 'f\'(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 = 0
z0 = 0
h = 0.1
x_target = 1.5

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

# Створюємо датафрейм з результатами
```

```

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

print('\n\nРозв\'язок Python:\n')
print(df_python)

# Запускаємо Octave
ос = Oct2Py()

# Визначаємо функцію рівняння
ос.eval("""
function result = equation_func(x, y, z)
    result = x * exp(x) - 2 * z - 2 * y;
end
""")

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

    results = zeros(1, 3);

    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 = -0.1;
y0 = 0;
z0 = 0;
h = 0.1;

```

```

x_target = 1.5;

# Викликаємо функцію методу Рунге-Кутти
results_octave = os.runge_kutta_octave(x0, y0, z0, h, x_target+h)

# Створюємо датафрейм з результатами
df_octave = pd.DataFrame(results_octave[2:, :], columns=["x", "f(x)",
"f\'(x)"]).set_index('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['f(x)'], label='f (x)', color='blue')
plt.plot(df_python.index, df_python['f\'(x)'], label='f \'(x)', color='orange')
plt.title('Результати Python')
plt.legend()
plt.grid(c='#E7E8D2')
plt.show()

# Створюємо графік для результатів Octave
plt.figure(figsize=(10, 6))
plt.plot(df_octave.index, df_octave['f(x)'], label='f (x)', color='blue')
plt.plot(df_octave.index, df_octave['f\'(x)'], label='f \'(x)', color='orange')
plt.title('Результати Octave')
plt.legend()
plt.grid(c='#E7E8D2')
plt.show()

```