НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

про виконання лабораторної роботи № 3

по дисципліні «Алгоритми і системи комп’ютерної математики - 1 »

на тему

«Розв’язання звичайних диференціальних рівнянь»

|  |  |
| --- | --- |
| Виконала: | Керівник: |
| студентка групи КМ-81 | Викладач кафедри ПМА |
| Верзун П.В. | Ліскін В.О. |
|  |  |
|  |  |
|  |  |

Київ — 2021

ЗМІСТ

ВСТУП 3

1. ПОСТАНОВКА ЗАДАЧІ 4
2. БЛОК СХЕМА АЛГОРИТМУ 5
3. ОПИС МЕТОДІВ 6
4. ОПИС ПРОГРАМ 8
   1. Розв’язання звичайного диференційного рівняння за допомогою мови програмування Python 8
   2. Розв’язання звичайного диференційного рівняння за допомогою СКМ Octave 10

4 ВИСНОВОК 12

ДОДАТОК А (код програм) 13

ВСТУП

Метою виконання даної лабораторної роботи є програмна реалізація чисельних методів розв’язання звичайних диференційних рівнянь. А саме за допомогою програмних засобів мови Python та за допомогою системи Octave було реалізовано методи Рунге-Кутта.

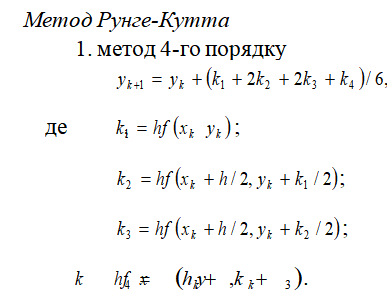
1 ПОСТАНОВКА ЗАДАЧІ

1. Скласти блок-схему алгоритму розв’язання диференціального рівняння.
2. Скласти програму розв’язання диференціального рівняння, одержати розв'язок за допомогою розробленої програми і за допомогою підпрограми СКМ.
3. Результати обчислень оформити у вигляді графіків.
4. Визначити близькість отриманого заданим методом розв'язку до точного значення. Як значення використовувати розв'язок, отриманий за допомогою СКМ.

Роботу було виконано відповідно до номеру в списку групи:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Диференційне  рівняння | | | | Початкова  умова | Проміжок  інтегрування | Крок |
|  |  |  |  |  | [0;2] | 0,2 |

Використовуючи наступний метод



2 БЛОК СХЕМА АЛГОРИТМУ

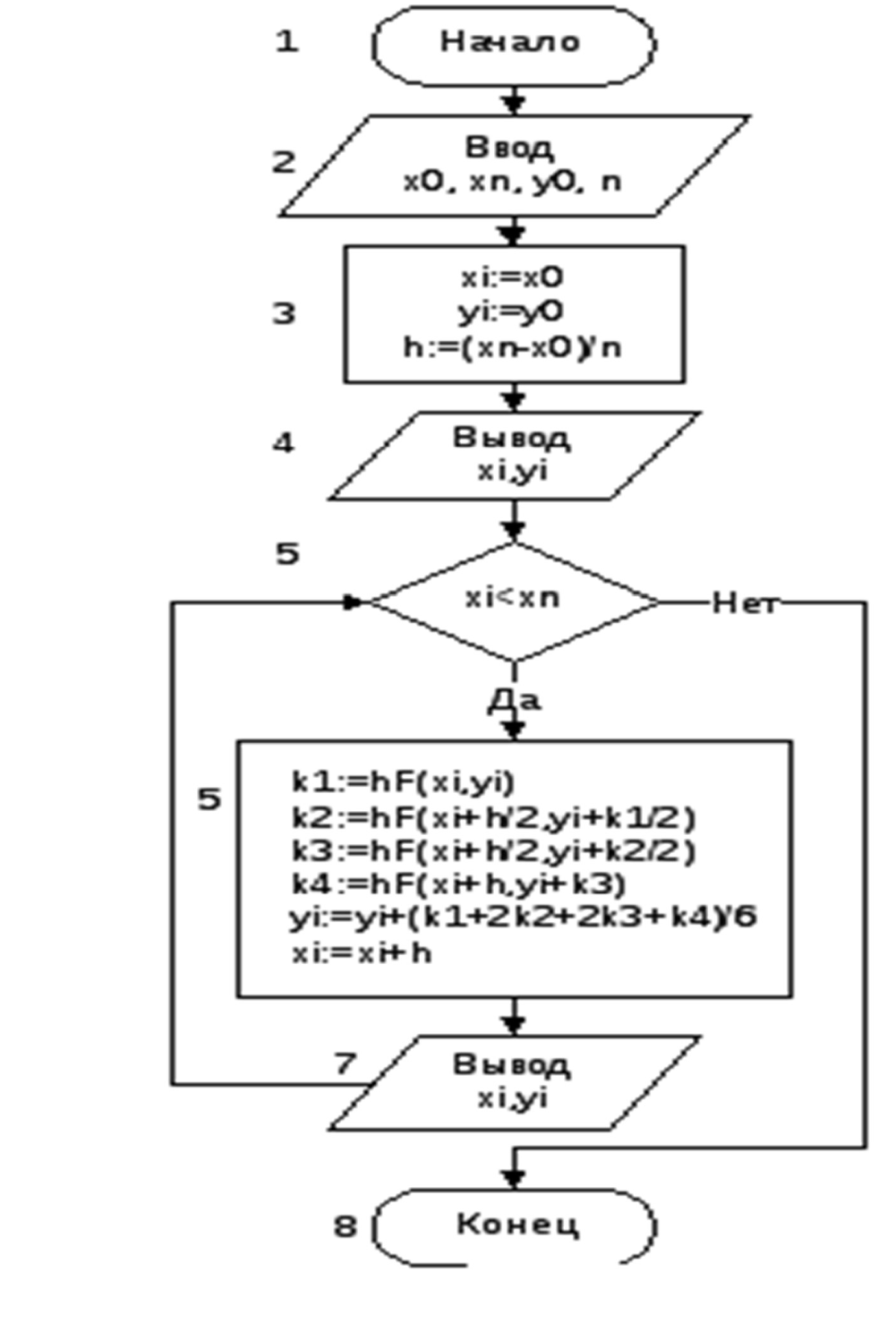


Рис 1. – Блок схема алгоритму Рунге-Кутта

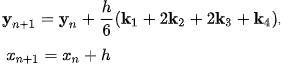
1. ОПИС МЕТОДІВ

Методи Рунге — Кутти — важлива група чисельних методів розв’язування (систем) звичайних диференціальних рівнянь. Названі на честь німецьких математиків Карла Рунге і Мартіна Кутти, які відкрили ці методи.

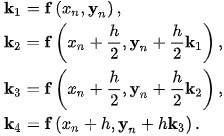
Метод Рунге—Кутти 4-го порядку настільки широко розповсюджений, що його часто називають просто методом Рунге — Кутти або RK4. Розглянемо задачу Коші для системи диференціальних рівнянь довільного порядку, що записується у векторній формі як



Тоді значення невідомої функції в точці 𝑥 ��� обчислюється відносно значення в попередній точці 𝑥� за формулою:



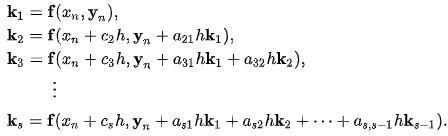
де h — крок інтегрування, а коефіцієнти 𝑘 � розраховуються таким чином:

Це метод 4-го порядку, тобто похибка на кожному кроці становить𝑂(ℎ�), а сумарна похибка на кінцевому інтервалі інтегрування є величиною 𝑂(ℎ�).

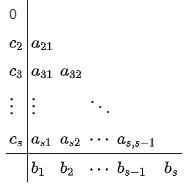
Група прямих методів Рунге — Кутти є узагальненням методу Рунге —

Кутти 4-го порядку. Наближення задається формулами

Де



Конкретний метод визначається числом s і коефіцієнтами 𝑏 �,𝑎 �� і 𝑐�. Ці коефіцієнти часто впорядковують в таблицю

Таблиця і графік отриманого результату:

X Y Z

0.2 1.421 2.221

0.4 1.892 2.492

0.6 2.422 2.822

0.8 3.026 3.226

1.0 3.718 3.718

1.2 4.52 4.32

1.4 5.455 5.055

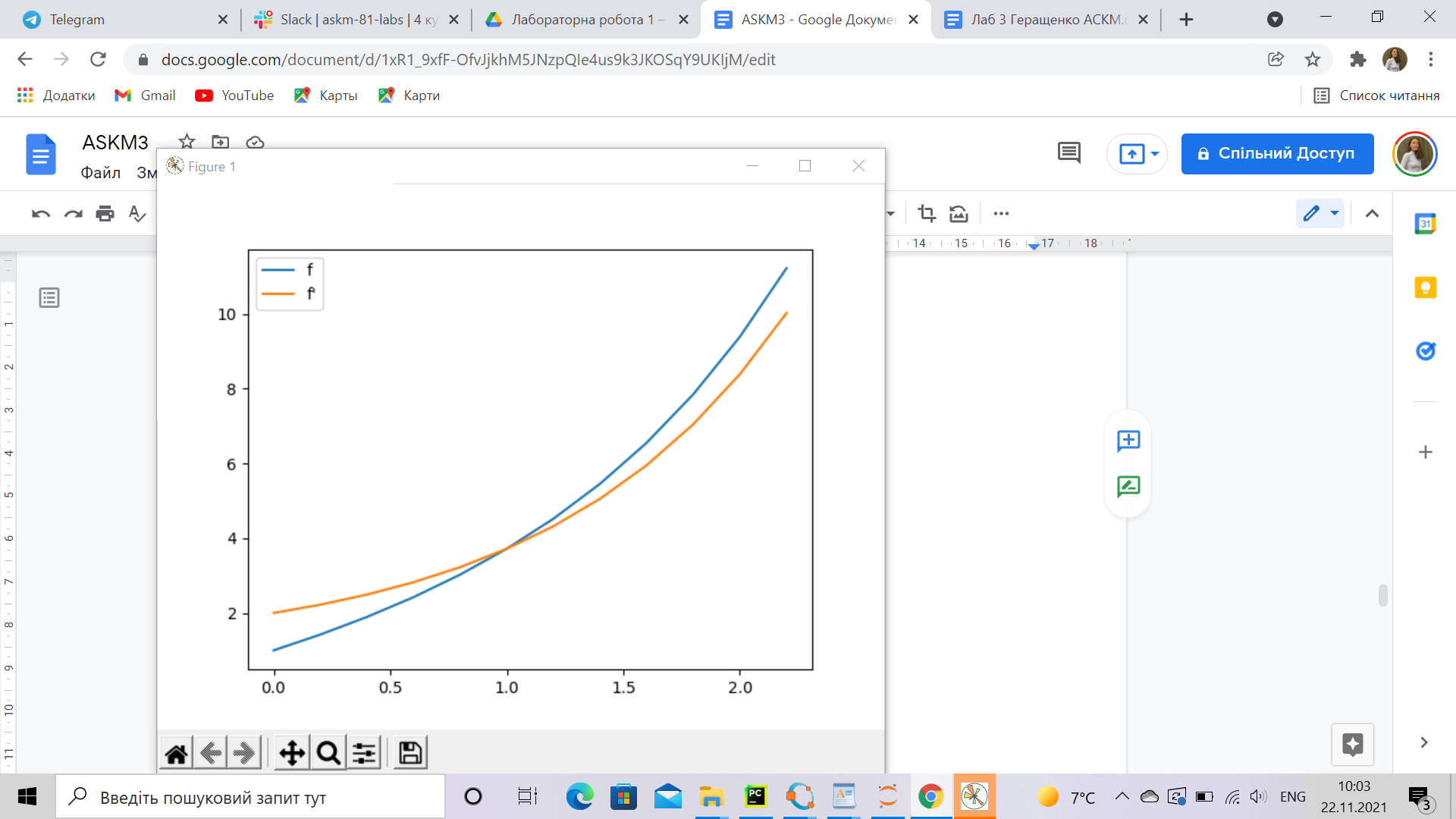
1.6 6.553 5.953

1.8 7.85 7.05

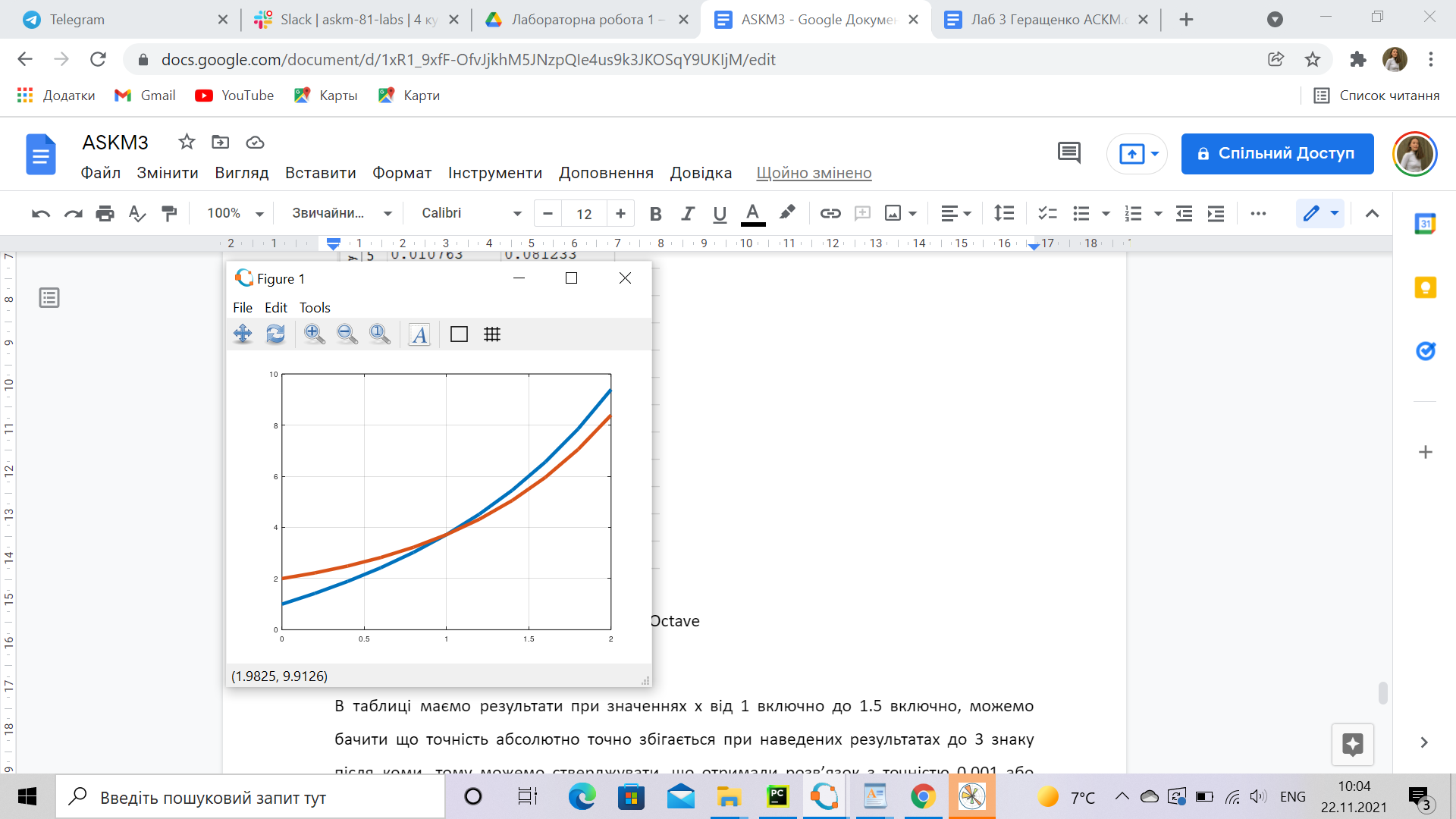
2.0 9.389 8.389

2.2 11.225 10.025

* 1. Розв’язання звичайного диференційного рівняння за допомогою СКМ



Наведемо тепер графік для результату в Octave



Python

import numpy as np

from matplotlib import pyplot as plt

class RungeKutta:

@staticmethod

def f(x, y, z):

return 3 \* z - 2 \* y + 2 \* x - 3

@staticmethod

def g(x, y, z):

return z

@classmethod

def calculate(cls):

x\_0 = 0

y\_0 = 1

z\_0 = 2

h = 0.2

list\_x = [x\_0]

list\_y = [y\_0]

list\_z = [z\_0]

print("\tX\t\tY\t\tZ")

while x\_0 < 2.00:

k1 = h \* cls.f(x\_0, y\_0, z\_0)

q1 = h \* cls.g(x\_0, y\_0, z\_0)

k2 = h \* cls.f(x\_0 + h / 2.0, y\_0 + q1 / 2.0, z\_0 + k1 / 2.0)

q2 = h \* cls.g(x\_0 + h / 2.0, y\_0 + q1 / 2.0, z\_0 + k1 / 2.0)

k3 = h \* cls.f(x\_0 + h / 2.0, y\_0 + q2 / 2.0, z\_0 + k2 / 2.0)

q3 = h \* cls.g(x\_0 + h / 2.0, y\_0 + q2 / 2.0, z\_0 + k2 / 2.0)

k4 = h \* cls.f(x\_0 + h, y\_0 + q3, z\_0 + k3)

q4 = h \* cls.g(x\_0 + h, y\_0 + q3, z\_0 + k3)

z\_1 = z\_0 + (k1 + 2.0 \* k2 + 2.0 \* k3 + k4) / 6.0

y\_1 = y\_0 + (q1 + 2.0 \* q2 + 2.0 \* q3 + q4) / 6.0

print(

"\t"

+ str(np.round(x\_0 + h, 3))

+ "\t\t"

+ str(np.round(y\_1, 3))

+ "\t\t"

+ str(np.round(z\_1, 3))

)

y\_0 = y\_1

z\_0 = z\_1

x\_0 += h

list\_x.append(x\_0)

list\_y.append(y\_0)

list\_z.append(z\_0)

plt.plot(list\_x, list\_y, label="f")

plt.plot(list\_x, list\_z, label="f'")

plt.legend()

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

RungeKutta.calculate()

Octave:

clear; clc;

function Dy = runge(x, y)

Dy = y(:);

Dy(1) = y(2);

Dy(2) = 3\*Dy(1) - 2\*y(1) + 2\*x - 3;

endfunction

h = 0.2;

x\_fin = 2;

y0 = 1;

Dy0 = 2;

[x, y] = ode45('runge', [0:h:x\_fin], [y0 Dy0]);

plot(x, y, 'LineWidth', 2); grid;

legend('y(x)', 'y''(x)', 0);

ВИСНОВОК

В даній лабораторній роботі було розроблено програмне забезпечення для розв’язання звичайних диференціальних рівнянь за допомогою засобів мов Python та середовища Octave. Ми отримали досить непогані результати в плані точності в обох випадках і на Python і на Octave, результати вийшли абсолютно однаковими. Точність досягла 0.001. Графіки розв’язку були наведені раніше.