**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ СИСТЕМ**

Звіт

з практичної роботи №2

Перевірила: Виконав:

Студент групи 2-8

Філомонова Т.О Вишнівський Владислав

Київ 2023





import numpy as np

def f(x):

return 6 \* x\*\*4 + 4 \* x\*\*3 + x\*\*2 + x - 10

eps = 0.0001

def find\_segments():

search\_range = np.arange(-10, 10, 1)

a = None

previous\_x = None

current\_x = None

segments = []

for x in search\_range:

x = round(x, 4)

current\_x = f(x)

if previous\_x is not None and previous\_x \* current\_x < 0:

segments.append((a, x))

a = x

previous\_x = current\_x

return segments

segments = find\_segments()

for a, b in segments:

print(f'Знайдено сегмент: [{a}, {b}]')

def rec(a, b, eps):

while abs(a - b) > eps:

if f(a) \* f((a + b) / 2) < 0:

b = (a + b) / 2

else:

a = (a + b) / 2

x = (a + b) / 2

print('x =', round(x, 5), '- Метод розділення навпіл')

def hord(a, b, eps):

if f(a) \* f(np.sqrt(a \* b)) > 0:

x0 = a

xi = b

else:

x0 = b

xi = a

xi\_1 = xi - (xi - x0) \* f(xi) / (f(xi) - f(x0))

while abs(xi\_1 - xi) > eps:

xi = xi\_1

xi\_1 = xi - (xi - x0) \* f(xi) / (f(xi) - f(x0))

print('x =', round(xi\_1, 5), '- Метод хорд')

segments = [(-1, 0), (0, 1)] # Визначені відрізки для пошуку коренів

for segment in segments:

print(f'Розв’язок нелінійного рівняння на відрізку {segment}')

rec(\*segment, eps)

hord(\*segment, eps)



<https://github.com/VladVish13/CHM>