

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Звіт
з лабораторної роботи №8
з дисципліни «Чисельні методи програмування»

Виконав:
Студент групи ФІТ 2-16
Пархоменко Іван Дмитрович

Київ 2024

i	0	1	2	3	4
x	0,9	1,2	1,6	2,1	2,8
y	3,47	4,53	2,86	1,64	2,25

```

import numpy as np
import matplotlib.pyplot as plt

def cubic_spline_coefficients(x, y):
    n = len(x) - 1
    h = np.diff(x)
    a = y
    b = np.zeros(n)
    d = np.zeros(n)
    c = np.zeros(n)

    alpha = np.zeros(n)
    for i in range(1, n):
        alpha[i] = (3 / h[i]) * (a[i+1] - a[i]) - (3 / h[i-1]) * (a[i] - a[i-1])

    l = np.ones(n)
    mu = np.zeros(n)
    z = np.zeros(n)

    for i in range(1, n):
        l[i] = 2 * (x[i+1] - x[i-1]) - h[i-1] * mu[i-1]
        mu[i] = h[i] / l[i]
        z[i] = (alpha[i] - h[i-1] * z[i-1]) / l[i]

    c[n-1] = (alpha[n-1] - h[n-2] * z[n-2]) / (2 * (h[n-2] + mu[n-1]))
    b[n-1] = (a[n] - a[n-1]) / h[n-1] - h[n-1] * (2 * c[n-1] + c[n-2]) / 3
    d[n-1] = (c[n-1] - c[n-2]) / (3 * h[n-1])

    for j in range(n - 2, -1, -1):
        c[j] = z[j] - mu[j] * c[j+1]
        b[j] = (a[j+1] - a[j]) / h[j] - h[j] * (c[j+1] + 2 * c[j]) / 3
        d[j] = (c[j+1] - c[j]) / (3 * h[j])

    return a, b, c, d

def cubic_spline_equations(x, y, a, b, c, d):
    equations = []
    n = len(x) - 1
    for i in range(n):
        equation = f"Відпізок {i+1}: "
        equation += f"S_{i}(x) = {a[i]} + {b[i]:.4f}(x - {x[i]}) + {c[i]:.4f}(x - {x[i]})^2 + {d[i]:.4f}(x - {x[i]})^3, x належить [{x[i]}, {x[i+1]}]"
        equations.append(equation)
    return equations

x = np.array([0.9, 1.2, 1.6, 2.1, 2.8])
y = np.array([3.47, 4.53, 2.86, 1.64, 2.25])

```

```

a, b, c, d = cubic_spline_coefficients(x, y)

equations = cubic_spline_equations(x, y, a, b, c, d)
for equation in equations:
    print(equation)

Відрізок 1:  $S_0(x) = 3.47 + 5.3517(x - 0.9) + 0.0000(x - 0.9)^2 + -20.2043(x - 0.9)^3$ ,  $x$  належить  $[0.9, 1.2]$ 
Відрізок 2:  $S_1(x) = 4.53 + -0.1034(x - 1.2) + -18.1839(x - 1.2)^2 + 20.0125(x - 1.2)^3$ ,  $x$  належить  $[1.2, 1.6]$ 
Відрізок 3:  $S_2(x) = 2.86 + -5.0446(x - 1.6) + 5.8311(x - 1.6)^2 + -1.2439(x - 1.6)^3$ ,  $x$  належить  $[1.6, 2.1]$ 
Відрізок 4:  $S_3(x) = 1.64 + -0.9790(x - 2.1) + 3.9653(x - 2.1)^2 + 1.8882(x - 2.1)^3$ ,  $x$  належить  $[2.1, 2.8]$ 

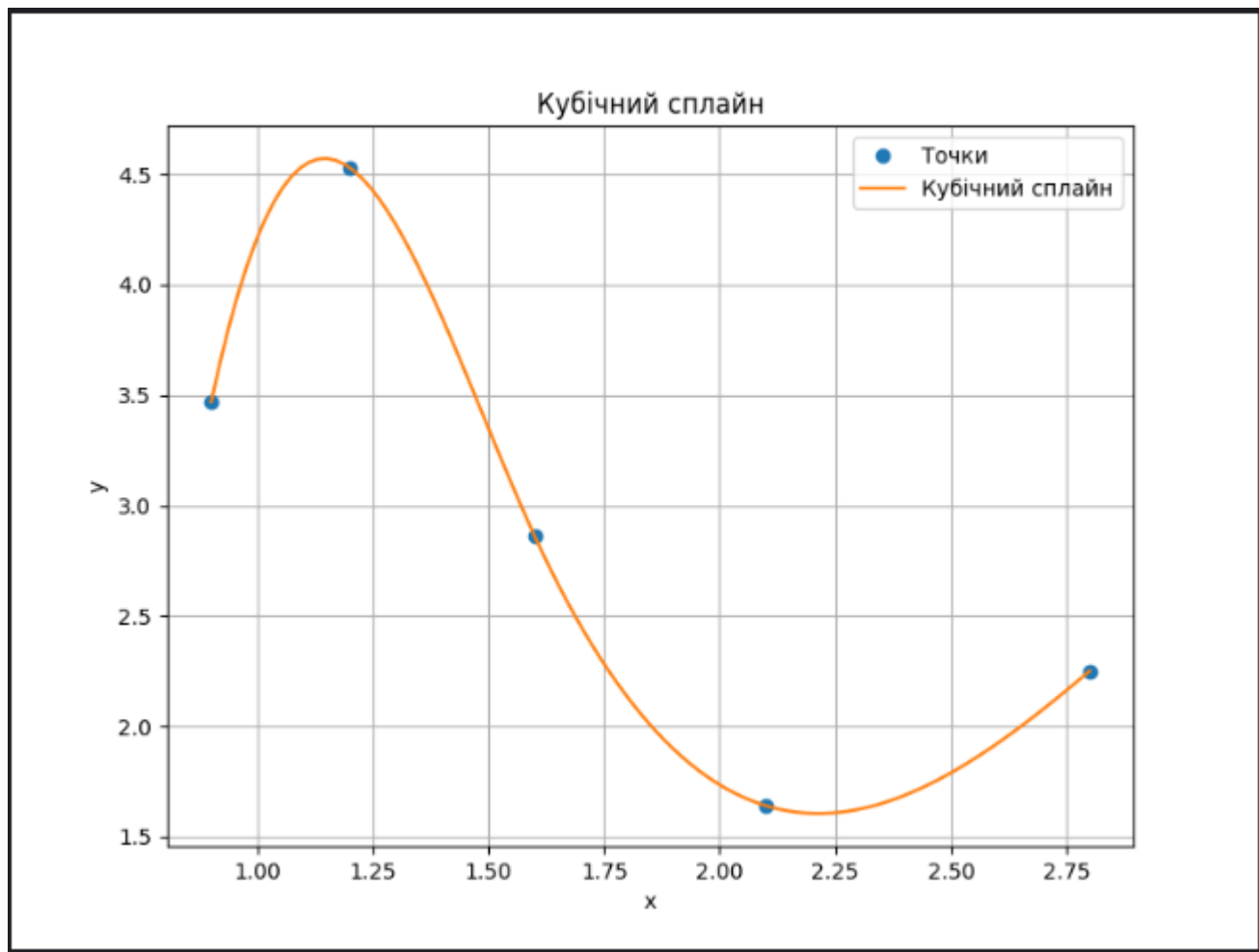
# Побудова кубічного сплайна з використанням CubicSpline
cs = CubicSpline(x, y)

# Генерація нових точок для гладкого графіку сплайна
x_new = np.linspace(np.min(x), np.max(x), 100)
y_new = cs(x_new)

# Роздрукуйте значення сплайна
print("Значення сплайна:")
for i in range(len(x_new)):
    print(f"x = {x_new[i]:.2f}, y = {y_new[i]:.3f}")

# Побудова графіку
plt.figure(figsize=(8, 6))
plt.plot(x, y, 'o', label='Точки')
plt.plot(x_new, y_new, label='Кубічний сплайн')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Кубічний сплайн')
plt.legend()
plt.grid(True)
plt.show()

```



<https://github.com/Bloorel/Numerical-Methods>