Пальонка Анастасія,2-8,Варіант 20

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 |
| X | 0,3 | 0,5 | 0,8 | 1,2 | 1,8 |
| y | 1,19 | 2,65 | 1,83 | 3,84 | 2,86 |

import numpy as np

from scipy.interpolate import CubicSpline

import matplotlib.pyplot as plt

# Задані точки

x = np.array([0.3, 0.5, 0.8, 1.2, 1.8])

y = np.array([1.19, 2.65, 1.83, 3.84, 2.86])

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]

# Виправлення обчислення коефіцієнтів для останнього відрізка

#c[n-1] = 0

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])

# Виведення аналітичного вигляду кубічного сплайна для кожного відрізка

for i in range(n):

print(f"Відрізок {i+1}:")

print(f"S\_{i}(x) = {a[i]} + {b[i].round(4)}(x - {x[i]}) + {c[i].round(4)}(x - {x[i]})^2 + {d[i].round(4)}(x - {x[i]})^3, x

належить [{x[i]}, {x[i+1]}]")

# Побудова кубічного сплайна

cs = CubicSpline(x, y)

# Генерація нових точок для гладкого графіку сплайна

x\_new = np.linspace(np.min(x), np.max(x), 10)

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()





