

ЛАБОРАТОРНА РОБОТА №8

Тема: «Інтерполяція сплайнами»

Трубчанінов Андрій Сергійович

ФІТ 2-8

В-29

Код:

```
[6]: import numpy as np
from scipy.interpolate import CubicSpline
import matplotlib.pyplot as plt

# Задані точки
x = np.array([1, 1.3, 1.7, 2.2, 2.8])
y = np.array([2.95, 3.89, 1.54, 3.38, 2.33])

# Обчислення коефіцієнтів для кубічного сплайну
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])

# 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), 100)
```

```

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), 100)
y_new = cs(x_new)
# 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()

```

```

Відрізок 1:
S_0(x) = 2.95 + 5.8252(x - 1.0) + 0.0(x - 1.0)^2 + -29.9099(x - 1.0)^3, x належить [1.0, 1.3]
Відрізок 2:
S_1(x) = 3.89 + -2.2504(x - 1.3) + -26.9189(x - 1.3)^2 + 44.6438(x - 1.3)^3, x належить [1.3, 1.7]
Відрізок 3:
S_2(x) = 1.54 + -2.3565(x - 1.7) + 26.6537(x - 1.7)^2 + -29.1612(x - 1.7)^3, x належить [1.7, 2.2]
Відрізок 4:
S_3(x) = 3.38 + 5.0852(x - 2.2) + -17.0881(x - 2.2)^2 + -9.4934(x - 2.2)^3, x належить [2.2, 2.8]
Значення сплайна:

```

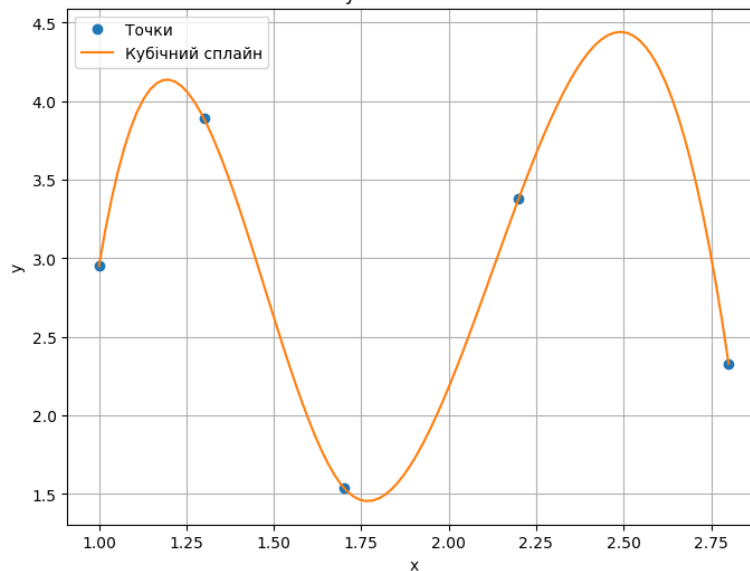
Кубічний сплайн

```

Відрізок 1:
S_0(x) = 2.95 + 5.8252(x - 1.0) + 0.0(x - 1.0)^2 + -29.9099(x - 1.0)^3, x належить [1.0, 1.3]
Відрізок 2:
S_1(x) = 3.89 + -2.2504(x - 1.3) + -26.9189(x - 1.3)^2 + 44.6438(x - 1.3)^3, x належить [1.3, 1.7]
Відрізок 3:
S_2(x) = 1.54 + -2.3565(x - 1.7) + 26.6537(x - 1.7)^2 + -29.1612(x - 1.7)^3, x належить [1.7, 2.2]
Відрізок 4:
S_3(x) = 3.38 + 5.0852(x - 2.2) + -17.0881(x - 2.2)^2 + -9.4934(x - 2.2)^3, x належить [2.2, 2.8]
Значення сплайна:

```

Кубічний сплайн



Код зі скрінів:

import numpy as np

from scipy.interpolate import CubicSpline

import matplotlib.pyplot as plt

```
# Задані точки
```

```
x = np.array([1, 1.3, 1.7, 2.2, 2.8])
```

```
y = np.array([2.95, 3.89, 1.54, 3.38, 2.33])
```

```
# Обчислення коефіцієнтів для кубічного сплайну
```

```
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):

$$c[j] = z[j] - \mu[j] * c[j + 1]$$

$$b[j] = (a[j + 1] - a[j]) / h[j] - h[j] * (c[j + 1] + 2 * c[j]) / 3$$

$$d[j] = (c[j + 1] - c[j]) / (3 * h[j])$$

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), 100)
y_new = cs(x_new)
# 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()
```

GitHub:

<https://github.com/Rey-ui/chiselni-metody>