Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота №3

Чисельні методи

Інтерполяція функцій. Сплайн інтерполяція

Варіант 17

Виконала
студентка групи ІТ-91:

Луцай К. А.

Перевірила:

ас. Тимофєєва Ю. С.

Київ 2021

Мета: навчитися інтерполяції функцій за допомогою мови програмування Python, використати метод кубічної сплайн-інтерполяції.

Теоретичні відомості:

Ідея сплайн-інтерполяції полягає в побудові поліномів між парами сусідніх вузлів інтерполяції, причому для кожної пари вузлів будується свій поліном. Найпоширеніший у практиці є кубічний сплайн, для побудови якого необхідно побудувати n многочленів третьої степені:

$$S(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, x_{i-1} \leq x \leq x_i, i = 1,2,\dots,n \quad (1)$$

$$a_i = y_{i-1}, i = \overline{1,n} \quad (2)$$

$$a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i, i = \overline{1,n} \quad (3)$$

де $h_i = x_i - x_{i-1}$

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, i = \overline{1,n-1}, d_n = -\frac{c_n}{3h_n}$$

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i), i = \overline{1,n-1}, b_n = \frac{y_n - y_{n-1}}{h_n} - \frac{2h_n c_n}{3}$$

$$c_1 = 0, c_n = 0, h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3\left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}}\right), i = \overline{2,n-1}$$

Розв'язок задачі в аналітичній формі:

X = -0.5

x_i	-2	-1	0	1	2
y_i	-1.8647	-0.63212	1	3.7183	9.3891

-1 < -0.5 < 0

i = 1, 2;

a2 = -0,63212

b2 = 1,36576

c2 = 0,39954

d2 = -0,13318

y(-0.5) = -0,63212 + 1,36576*(0.5 + 1) + 0,39954*(1.5)^2 - 0,13318*(1.5)^3

y(-0.5) = 1.2660025

Лістинги програм розв'язування задачі:

Методи для обрахування коефіцієнтів та самої інтерполяції в заданій точці наведено далі. Відбувається генерація масиві коефіцієнтів, на основі яких потім вираховується наближене значення у точці X. Основна функція приймає масиви координат та обрану точку, повертає значення у цій точці.

```
def curv(x, y):
    n = len(x) - 1
    c = np.zeros((n), dtype=float)
    d = np.ones((n+1), dtype=float)
    e = np.zeros((n), dtype=float)
    k = np.zeros((n+1), dtype=float)

    c[0:n-1] = x[0:n-1] - x[1:n]
    d[1:n] = 2 * (x[0:n-1] - x[2:n+1])
    e[1:n] = x[1:n] - x[2:n+1]

    k[1:n] = 6 * (y[0:n-1] - y[1:n]) / (x[0:n-1] - x[1:n]) \
        -6 * (y[1:n] - y[2:n+1]) / (x[1:n] - x[2:n+1])

    n = len(d)
    for i in range(1, n):
        lam = c[i-1]/d[i-1]
        d[i] = d[i] - lam * e[i-1]
        c[i-1] = lam

    for i in range(1, n):
        k[i] = k[i] - c[i-1] * k[i-1]
    k[n-1] = k[n-1] / d[n-1]
    for j in range(n-2, -1, -1):
        k[j] = (k[j] - e[j] * k[j+1]) / d[j]

    return k

def spline(x, y, X):
    def seg(x, X):
        left = 0
        right = len(x) - 1
        while True:
            if (right - left) <= 1:
                return left
            i = int((left + right) / 2)
            if X < x[i]:
                right = i
            else:
                left = i

    k = curv(X17, Y17)
    i = seg(x, X)
    h = x[i] - x[i+1]
```

```
    Y = ((X - x[i+1])**3/h - (X - x[i+1])*h)*k[i]/6 \
        - ((X - x[i])**3/h - (X - x[i])*h)*k[i+1]/6 \
            + (y[i] * (X - x[i+1]) - y[i+1] * (X - x[i]))/h
    return Y
```

Задані за умовою величини, та потрібні модулі:

```
import matplotlib.pyplot as plt
import numpy as np

X17 = np.array([-2, -1, 0, 1, 2], dtype=float)
Y17 = np.array([-1.8647, -0.63212, 1, 3.7183, 9.3891], dtype=float)

xT = -0.5
```

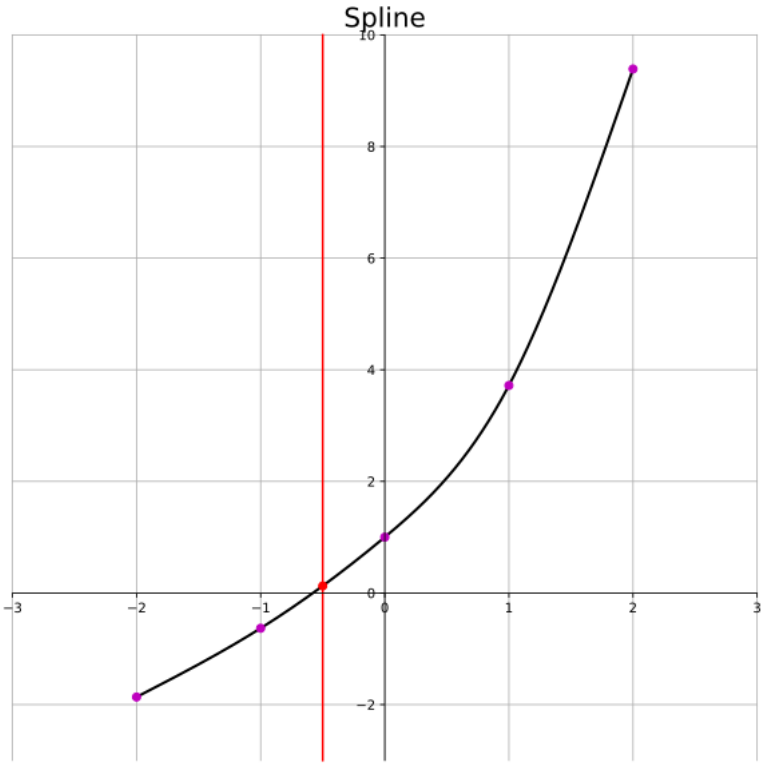
Вивід результатів у текстовому та графічному вигляді:

```
for i in range(len(X17)):
    print(str(X17[i]).rjust(4), str(Y17[i]).rjust(8))
print(xT, spline(X17, Y17, xT))

x = np.linspace(-2, 2, 1000)
y = [spline(X17, Y17, i) for i in x]
fig = plt.figure(figsize=(10, 10))
ax = plt.subplot(ylim=(-3, 10), xlim=(-3, 3))
plt.axvline(xT, color="red")
ax.plot(x, y, color="black", lw=2)
ax.plot(X17, Y17, "mo")
ax.plot(xT, spline(X17, Y17, xT), "ro")
ax.grid(True)
ax.spines["left"].set_position("zero")
ax.spines["right"].set_color("none")
ax.spines["bottom"].set_position("zero")
ax.spines["top"].set_color("none")
plt.title("Spline", fontsize=20)
plt.show()
```

Результати виконання програм:

```
-2.0  -1.8647
-1.0 -0.63212
 0.0      1.0
 1.0   3.7183
 2.0   9.3891
-0.5 0.12654089285714276
```



Висновки: було написано програму для інтерполяції функцій на мові Python: було розроблено метод кубічної сплайн-інтерполяції.