НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
з лабораторної роботи № 2
із дисципліни «Методи обчислень-1»
на тему
Інтерполяція сплайнами

Варіант 11

Виконав:

Студент групи КМ-93
Костенко Олександр

Керівник:

викладач Любашенко Н. Д.

Київ-2021

Зміст

Завдання №1

Умова:

За даною таблицею значень функції побудувати кубічний сплайн.

Сплайн повинен задовольняти граничним умовам другого типу. Обчислити наближене значення функції та її похідні другого порядку в заданій точці.

Опис методу:

На кожному з відрізків $[x_i, x_{i+1}]$ ($i=\overline{1,N}$) будуємо поліном третього порядку:

$$S_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3$$

Таким чином, для визначення сплайну на всьому відрізку $[x_0, x_N]$ потрібно визначити 4N невідомих $a_i,\ b_i, c_i, d_i$ ($i=\overline{1,N}$). Для цього потрібно скласти 4N незалежних рівнянь.

Перші 2N рівняння отримуємо згідно умови проходження сплайна через вузли інтерполювання:

$$a_i = y_{i-1}\ (i = \overline{1,N})$$

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

Оскільки:

$$S'(x) = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2$$

$$S''(x) = 2c_i + 6d_i(x - x_{i-1}),$$

наступні 2N-2 рівняння відповідають умовам неперервності першої та другої похідної від функції у внутрішніх вузлах інтерполювання:

$$b_i + c_i h_i + 3d_i {h_i}^2 = b_{i+1}\ (i = \overline{1,N-1})$$

$$2c_i + 6d_i h_i = 2c_{i+1}\ (i = \overline{1,N-1})$$

Таким чином, для визначення невідомих $a_i,\ b_i, c_i, d_i$ ($i=\overline{1,N}$) отримано 4N-2 рівняння, які після елементарних перетворень набувають наступного вигляду:

$$\begin{cases} a_i = y_{i-1}\ (i = \overline{1,N}) \\ b_i + c_i h_i + d_i {h_i}^2 = \dfrac{y_i - y_{i-1}}{h_i}\ (i = \overline{1,N}) \\ b_i + h_i(c_i + c_{i+1}) = b_{i+1}\ (i = \overline{1,N-1}) \\ d_i = \dfrac{c_{i+1} - c_i}{3h_i}\ (i = \overline{1,N-1}) \end{cases}$$

Два відсутніх рівняння задаються і вигляді обмежень похідних сплайна на кінцях відрізка інтерполювання $[x_0, x_N]$.

Для граничних умов першого типу:

$$S'(x_0) = f'(x_0)$$

$$S'(x_N) = f'(x_N)$$

В наведених позначеннях ці умови записуються наступним чином:

$$\begin{cases} b_1 = f'(x_0) \\ b_N + 2c_N h_N + 3d_N {h_N}^2 = f'(x_N) \end{cases}$$

Для граничних умов другого типу:

$$S''(x_0) = f''(x_0)$$

$$S''(x_N) = f''(x_N)$$

В наведених позначеннях ці умови записуються наступним чином:

$$\begin{cases} 2c_1 = f''(x_0) \\ 2c_N + 6d_N h_N \quad = f''(x_N) \end{cases}$$

Розв'язок наведених СЛАР може бути отриманий методом прогонки, оскільки матриця системи при невідомих є трьохдіагональною.

Розв'язання:

Маємо таблицю значень:

i	0	1	2	3	4
x	-8.1	-0.9	4	-4.8	6.5
y	3.325	0.895	2.011	1.108	2.698

Згідно варіанту:

$$S''(x_0) = f''(x_0) = 11$$

$$S''(x_N) = f''(x_N) = -11$$

Оскільки кількість вузлів більша 4N, то скористаємося першим методом.

На кожному з відрізків $[x_i, x_{i+1}]$ (i=$\overline{1, N}$) будуємо поліном третього порядку:

$$S_i(t) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3$$

Відсортуємо значення в заданій таблиці:

i	0	1	2	3	4
x	-8.1	-4.8	-0.9	4	6.5
y	3.325	1.108	0.895	2.011	2.698

Згідно умов інтерполяції кількість вузлів дорівнює 5, тоді N=5,

i	1	2	3	4

h	3,3	3,9	4,9	2,5

Якщо задані умови другого типу, виконуючи перетворення, приводимо систему для визначення невідомих коефіцієнтів, із урахуванням граничних умов другого типу до вигляду:

$$\begin{cases} \mathrm{c}_1 = 0.5f''(x_0) \\ h_{i-1}c_{i-1} + 2c_i(h_i + h_{i-1}) + h_i c_{i+1} = 3\left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}}\right), (i = \overline{2, N-1}) \\ h_{N-1}c_{N-1} + 2c_N(h_N + h_{N-1}) = -f''(x_N)\frac{h_N}{2} + 3\left(\frac{y_N - y_{N-1}}{h_N} - \frac{y_{N-1} - y_{N-2}}{h_{N-1}}\right) \end{cases}$$

Підставляючи, отримуємо:

$$\begin{cases} c_0 & & & & = 5.5 \\ 3.3c_0 & +14.4c_1 & +3.9c_2 & & = 1.852 \\ & 3.9c_1 & +17.6c_2 & +4.9c_3 & = 0.847 \\ & & 4.9c_2 & +14.8c_3 & = 13.235 \end{cases}$$

Оскільки $|a_{ii}|$ >суми інших коефіцієнтів відповідного рядка, то виконується умова діагонального домінування, тому можемо скористатися методом перегонки:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3.3 & 14.4 & 3.9 & 0 \\ 0 & 3.9 & 17.6 & 4.9 \\ 0 & 0 & 4.9 & 14.8 \end{bmatrix} = \begin{bmatrix} -b_1 & c_1 & 0 & 0 \\ a_2 & -b_2 & c_2 & 0 \\ 0 & a_3 & -b_3 & c_3 \\ 0 & 0 & a_4 & -b_4 \end{bmatrix}$$

$$d = \begin{bmatrix} 5.5 \\ 1.852 \\ 0.847 \\ 13.235 \end{bmatrix}$$

$$\begin{array}{lll} a_1 = 0 & b_1 = -1 & c_1 = 0 \\ a_2 = 3.3 & b_2 = -14.4 & c_2 = 3.9 \\ a_3 = 3.9 & b_3 = -17.6 & c_3 = 4.9 \\ a_4 = 4.9 & b_4 = -14.8 & c_4 = 0 \end{array}$$

Прямий хід:

$$P_0 = \frac{c_1}{b_1} = 0$$

$$P_1 = \frac{c_2}{b_2 - a_2 P_0} = -0.271$$

$$P_2 = \frac{c_3}{b_3 - a_3 P_1} = -0.296$$

$$P_3 = \frac{c_4}{b_4 - a_4 P_2} = 0$$

$$Q_0 = \frac{-d_1}{b_1} = 5.5$$

$$Q_1 = \frac{-d_2 + a_2 Q_0}{b_2 - a_2 P_1} = -1.132$$

$$Q_2 = \frac{-d_3 + a_3 Q_1}{b_3 - a_3 P_2} = 0.318$$

$$Q_3 = \frac{-d_4 + a_4 Q_2}{b_4 - a_4 P_3} = 0.875$$

Зворотній хід:

$$c_3 = Q_3 = 0.875$$

$$c_2 = P_2 c_3 + Q_2 = 0.059$$

$$c_1 = P_1 c_2 + Q_1 = -1.148$$

$$c = P_0 c_1 + Q_0 = 5.5$$

$$c = \begin{bmatrix} 5.5 \\ -1.148 \\ 0.059 \\ 5.5 \end{bmatrix}$$

Після визначення усіх коефіцієнтів, знаходимо інші невідомі:

$$\begin{cases} d_N = \dfrac{f''(x_N) - 2c_N}{6h_N}, \qquad d_i = \dfrac{c_{i+1} - c_i}{3h_i}, (i = \overline{1,N}) \\ b_N = \dfrac{y_N - y_{N-1}}{h_N} - \dfrac{h_N}{6} f''(x_N) - \dfrac{2}{3} c_N h_N \\ b_i = b_{i+1} - h_i(c_i + c_{i+1}), (i = \overline{N-1,1}) \\ a_i = y_{i-1}\ (i = \overline{1,N}) \end{cases}$$

Отримуємо:

$$a = \begin{bmatrix} 3.325 \\ 1.108 \\ 0.895 \\ 2.011 \end{bmatrix}$$

$$b = \begin{bmatrix} -11.29 \\ 3.072 \\ -1.175 \\ 3.4 \end{bmatrix}$$

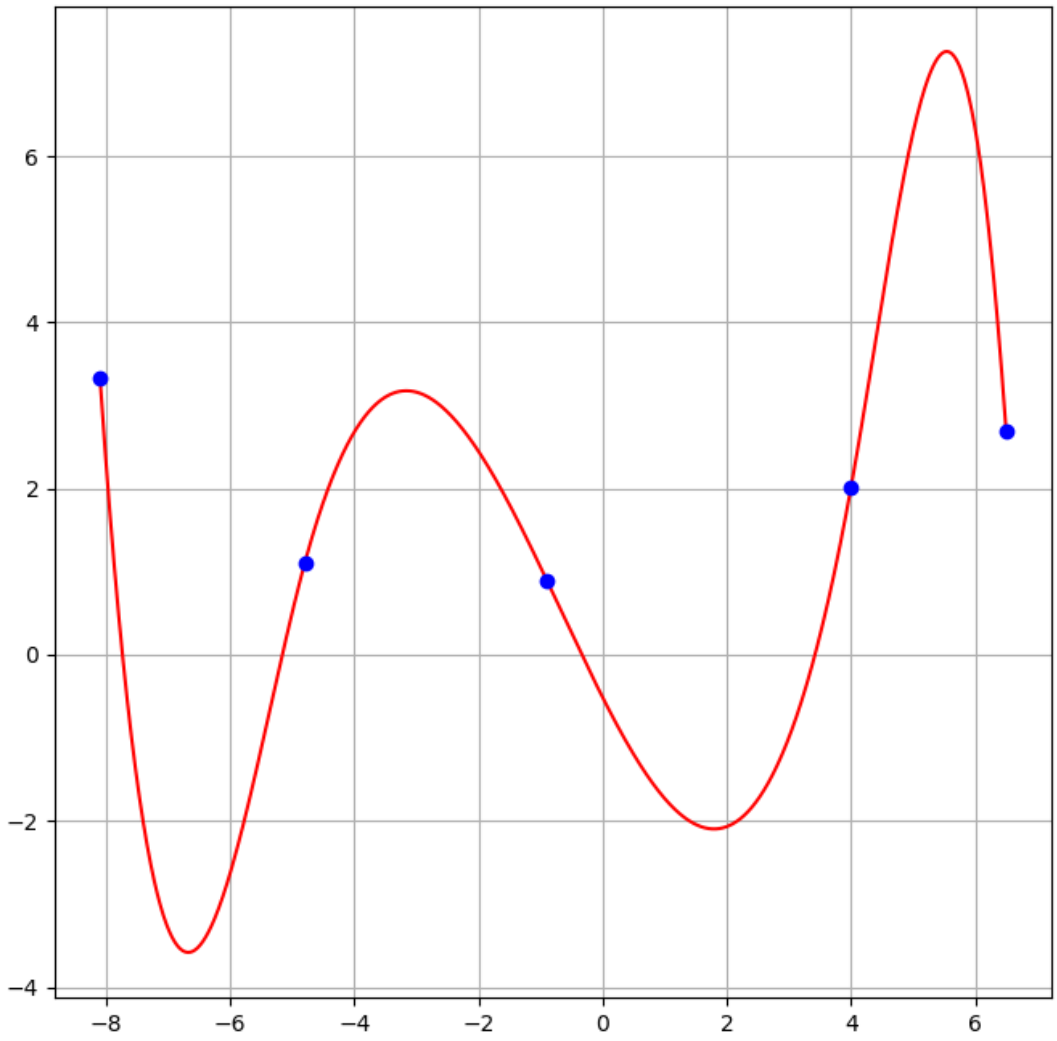
$$c = \begin{bmatrix} 5.5 \\ -1.148 \\ 0.059 \\ 5.5 \end{bmatrix}$$

$$d = \begin{bmatrix} -0.671 \\ 0.103 \\ 0.055 \\ -0.85 \end{bmatrix}$$

Тоді, підставляючи обчислені коефіцієнти у формулу саплайна, знаходимо рівняння сплана:

$$S(x) = \begin{cases} 3.325 - 11.29(x + 8.1) + 5.5(x + 8.1)^2 - 0.671(x + 8.1)^3, & x \in (-8.1, -4.8) \\ 1.108 + 3.072(x + 4.8) - 1.148(x + 4.8)^2 + 0.103(x + 4.8)^3, & x \in (-4.8, -0.9) \\ 0.895 - 1.175(x + 0.9) + 0.059(x + 0.9)^2 + 0.055(x + 0.9)^3, & x \in (-0.9, 4) \\ 2.011 + 3.4(x - 4) + 5.5(x - 4)^2 - 0.85(x - 4)^3, & x \in (4, 6.5) \end{cases}$$

Графік сплайну:



Маємо значення $\bar{x}$ для обчислення функції:

$$\bar{x} = -6.6$$

Отримане значення попадає в проміжок: $x \in (-8.1, -4.8)$,

Тоді значення функції в цій точці дорівнює

$$f(\bar{x}) = 3.325 - 11.29(\bar{x} + 8.1) + 5.5(\bar{x} + 8.1)^2 - 0.671(\bar{x} + 8.1)^3 = -4.873$$

Друга похідна в цій точці дорівнює

$$f''(\bar{x}) = S''(\bar{x}) = 2 * 5.5 - 6 * 0.671(\bar{x} + 8.1) = 4.772$$

Завдання №2

Умова:

Розробити програмне забезпечення задачі побудови кубічного сплайну із довільною кількістю вузлів інтерполяції та можливістю застосування на кінцях відрізка інтерполювання граничних умов першого та/або другого типу.

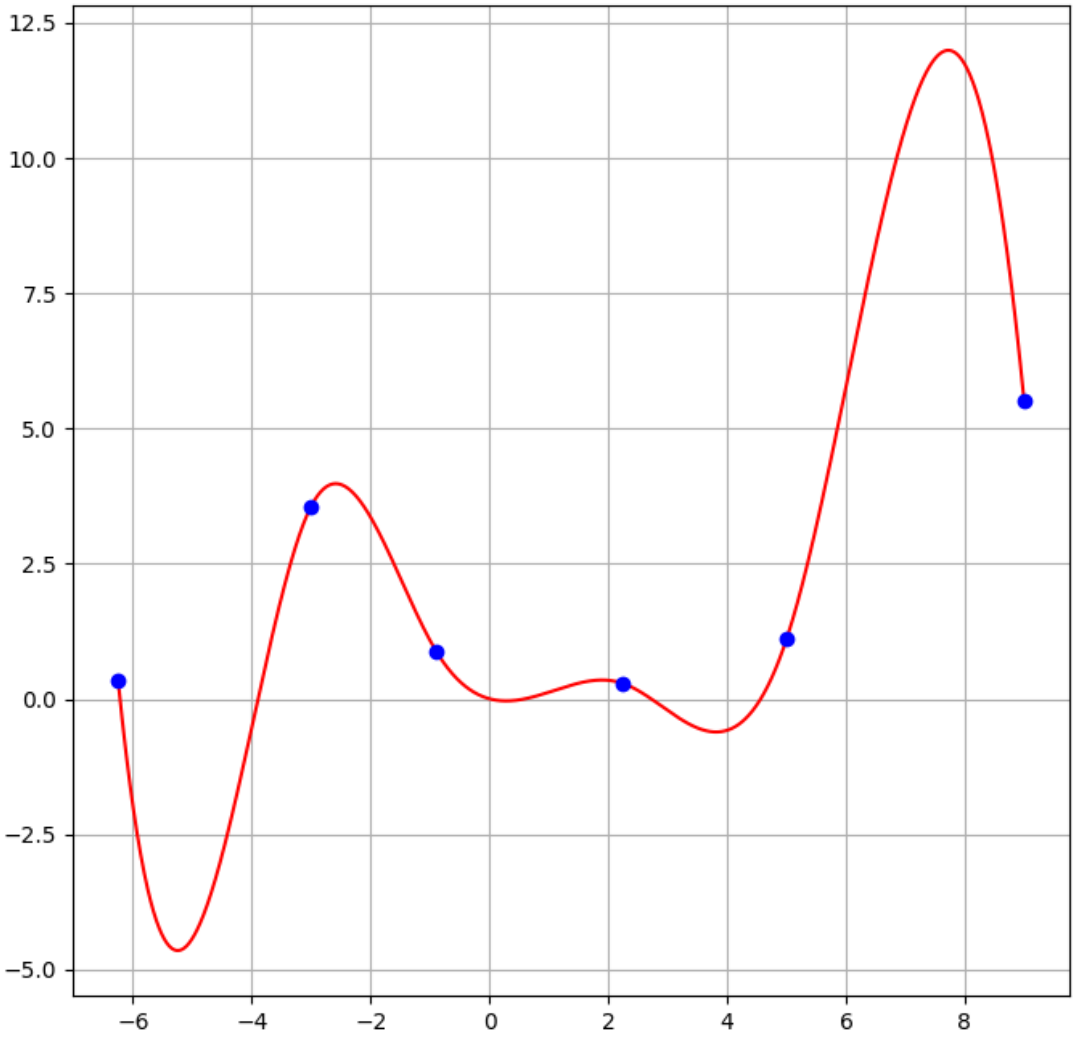
Розв'язання:

У програмі було використано такі бібліотеки: matplotlib, numpy.

Код програми наведено в **Додатку 1**.

Результат роботи програми:

```
Ввведіть кількість елементів: 6
Введіть 0-ий x: 9
Введіть 0-ий y: 5.512
Введіть 1-ий x: -3
Введіть 1-ий y: 3.564
Введіть 2-ий x: 2.23
Введіть 2-ий y: 0.298
Введіть 3-ий x: -6.25
Введіть 3-ий y: 0.359
Введіть 4-ий x: -0.9
Введіть 4-ий y: 0.895
Введіть 5-ий x: 5
Введіть 5-ий y: 1.105
[-6.25, -3.0, -0.9, 2.23, 5.0, 9.0]
[0.359, 3.564, 0.895, 0.298, 1.105, 5.512]
Оберіть тип умов (введіть 1, якщо бажаєте обрати перший тип, або 2, якщщо другий):1
f=[[2, 1, 0, 0, 0, 11.064142011834319], [3.25, 10.7, 2.1, 0, 0, -6.7713186813186805], [0, 2.1, 10.46, 3.13
, 0, 3.240652670013692], [0, 0, 3.13, 11.8, 2.77, 1.4462116930600568], [0, 0, 0, 1.8466666666666667, 7.693
333333333333, 13.901750000000002]]
Коефіцієнти:
a= [0.359, 3.564, 0.895, 0.298, 1.105]
b= [-10.999999999999996, 2.1351311978747622, -1.8230524002389006, -0.3570022823634304, 3.3268748530517103]
c= [7.022563181719008, -2.9809843516036976, 1.09613501916862, -0.6277484000390767, 1.9576679435102464]
d= [-1.0260048752125852, 0.6471618048844947, -0.18358715859506888, 0.31112110030677775, -0.628487289193293
6]
Рівняння сплана:
S0=0.359+-10.999999999999996(x--6.25)+7.022563181719008(x--6.25)^2+-1.0260048752125852(x--6.25)^3, для x∈[
-6.25, -3.0]
S1=3.564+2.1351311978747622(x--3.0)+-2.9809843516036976(x--3.0)^2+0.6471618048844947(x--3.0)^3, для x∈[-3.
0, -0.9]
S2=0.895+-1.8230524002389006(x--0.9)+1.09613501916862(x--0.9)^2+-0.18358715859506888(x--0.9)^3, для x∈[-0.
9, 2.23]
S3=0.298+-0.3570022823634304(x-2.23)+-0.6277484000390767(x-2.23)^2+0.31112110030677775(x-2.23)^3, для x∈[2
.23, 5.0]
S4=1.105+3.3268748530517103(x-5.0)+1.9576679435102464(x-5.0)^2+-0.6284872891932936(x-5.0)^3, для x∈[5.0, 9
.0]
Введіть задане значення x: 0.3
Функція в точці 0.3 дорівнює -0.03146706273614658
Друга похідна функції в точці 0.3 дорівнює 0.8704424964527444
Бажаєте побудувати сплайн з іншими граничними умоваим? (Y/n): Y
f=[[1, 0, 0, 0, 0, 5.5], [3.25, 10.7, 2.1, 0, 0, -6.7713186813186805], [0, 2.1, 10.46, 3.13, 0, 3.24065267
0013692], [0, 0, 3.13, 11.8, 2.77, 1.4462116930600568], [0, 0, 0, 2.77, 13.54, 24.7]]
Коефіцієнти:
a= [0.359, 3.564, 0.895, 0.298, 1.105]
b= [-8.314291369836596, 1.4432870579181238, -1.723085309151509, -0.4930898425742414, 3.2448482616451635]
c= [5.5, -2.497668176075471, 0.9898718108042174, -0.5969020131756974, 1.9463381518830636]
d= [-0.8202736590846638, 0.5535777756951886, -0.16898549776143926, 0.30604574790117456, -0.620528179323588
6]
Рівняння сплана:
S0=0.359+-8.314291369836596(x--6.25)+5.5(x--6.25)^2+-0.8202736590846638(x--6.25)^3, для x∈[-6.25, -3.0]
S1=3.564+1.4432870579181238(x--3.0)+-2.497668176075471(x--3.0)^2+0.5535777756951886(x--3.0)^3, для x∈[-3.0
, -0.9]
S2=0.895+-1.723085309151509(x--0.9)+0.9898718108042174(x--0.9)^2+-0.16898549776143926(x--0.9)^3, для x∈[-0
.9, 2.23]
S3=0.298+-0.4930898425742414(x-2.23)+-0.5969020131756974(x-2.23)^2+0.30604574790117456(x-2.23)^3, для x∈[2
.23, 5.0]
S4=1.105+3.2448482616451635(x-5.0)+1.9463381518830636(x-5.0)^2+-0.6205281793235886(x-5.0)^3, для x∈[5.0, 9
.0]
Введіть задане значення x: 0.3
Функція в точці 0.3 дорівнює -0.03929390355550477
Друга похідна функції в точці 0.3 дорівнює 0.763048037726072
Бажаєте побудувати сплайн з іншими граничними умоваим? (Y/n): n
```


12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−6
−4
−2
0
2
4
6
8

Додаток 1
Код програми:

```
import numpy as np
from matplotlib import pyplot as plt
import math
import collections

#Сортування
def sort_xy(x,y):
    dictionary={}
    n=len(x)
    for key in x:
        for value in y:
            dictionary[key]=value
            y.remove(value)
            break
    x.clear()
    y.clear()
    odictionary=collections.OrderedDict(sorted(dictionary.item
s()))
    for k,v in odictionary.items():
        x.append(k)
        y.append(v)
    res=[]
    res.append(x)
    res.append(y)
    return res


#Вибір типу умов
def type(an):
    func=[]
    if an==1:
        func.append(-11)
        func.append(-11)
    else:
        func.append(11)
        func.append(-11)
    return func


#СЛАР для першого типу при N<4
def slar_1_3(h,y,func,n):
```

```
    f=[]
    for i in range(n):
        f.append([])
    for i in range(n):
        for j in range(n+1):
            f[i].append(0)
    f[0][0]=1
    f[0][n]=func[0]
    for i in range(n-2):
        f[i+1][i]=1/h[i]
        f[i+1][i+1]=2*(1/h[i]+1/h[i+1])
        f[i+1][i+2]=1/h[i+1]
        f[i+1][i+3]=3*((y[i+2]-
y[i+1])/math.pow(h[i+1],2)+(y[i+1]-y[i])/math.pow(h[i],2))
    f[-1][-2]=1
    f[-1][-1]=func[1]
    return f

#СЛАР для другого типу при N<4
def slar_2_3(h,y,t,n):
    f=[]
    for i in range(n):
        f.append([])
    for i in range(n):
        for j in range(n+1):
            f[i].append(0)
    f[0][0]=-2*h[0]
    f[0][1]=-h[0]
    f[0][n]=type(t)[0]-(2/(math.pow(h[0],2)*3*(y[1]-y[0])))
    for i in range(n-2):
        f[i+1][i]=1/h[i]
        f[i+1][i+1]=2*(1/h[i]+1/h[i+1])
        f[i+1][i+2]=1/h[i+1]
        f[i+1][-1]=3*((y[i+2]-
y[i+1])/math.pow(h[i+1],2)+(y[i+1]-y[i])/math.pow(h[i],2))
    f[-1][-3]=h[-2]
    f[-1][-2]=2*h[-2]
    f[-1][-1]=type(t)[1]-(2/((math.pow(h[-2],2))*3*(y[-1]-y[-
2])))
    return f


#СЛАР для 1 типу умов
def slar_1(n,func, h, y):
    f=[]
    for i in range(n):
```

```
        f.append([])
    for i in range(n):
        for j in range (n+1):
            f[i].append(0)
    f[0][0]=2
    f[0][1]=1
    f[0][-1]=3*((y[1]-y[0])/math.pow(h[0],2))-3*func[0]/h[0]
    for i in range(1, n-1):
        f[i][i-1]=h[i-1]
        f[i][i]=2*(h[i]+h[i-1])
        f[i][i+1]=h[i]
        f[i][-1]=3*((y[i+1]-y[i])/h[i]-(y[i]-y[i-1])/h[i-1])
    f[-1][-3]=2*h[-2]/3
    f[-1][-2]=(h[-1]+4*h[-2]/3)
    f[-1][-1]=-func[1]+3*(y[-1]-y[-2])/h[-1]-2*(y[-2]-y[-
3])/h[n-1]
    print("f="+str(f))
    return f

#СЛАР для другого типу умов
def slar_2(n, func, h, y):
    f=[]
    for i in range(n):
        f.append([])
    for i in range(n):
        for j in range (n+1):
            f[i].append(0)
    f[0][0]=1
    f[0][-1]=0.5*func[0]
    for i in range(1, n-1):
        f[i][i-1]=h[i-1]
        f[i][i]=2*(h[i]+h[i-1])
        f[i][i+1]=h[i]
        f[i][-1]=3*((y[i+1]-y[i])/h[i]-(y[i]-y[i-1])/h[i-1])
    f[-1][-3]=h[-2]
    f[-1][-2]=2*(h[-1]+h[-2])
    f[-1][-1]=-func[1]*h[-1]/2+3*((y[-1]-y[-2])/h[-1]-(y[-2]-
y[-3])/h[n-1])
    print("f="+str(f))
    return f


#метод перегонки
def dist(f,n):
    #обчислюємо коефіцієнти
    a=[]
```

```
    b=[]
    c=[]
    d=[]

    a.append(0)
    for i in range(1, n):
        a.append(f[i][i-1])
    for i in range(n):
        b.append(-f[i][i])
        d.append(f[i][-1])
    for i in range(n-1):
        c.append(f[i][i+1])
    c.append(0)

    #Обчислюємо перегоночні поефіцієнти
    p=[]
    q=[]

    p.append(c[0]/b[0])
    q.append(-d[0]/b[0])
    for i in range(1,n):
        p.append(c[i]/(b[i]-a[i]*p[i-1]))
        q.append((a[i]*q[i-1]-d[i])/(b[i]-a[i]*p[i-1]))

    m=[]
    for i in range(n):
        m.append(0)
    m[-1]=q[-1]
    for i in range(n-2,-1,-1):
        m[i]=p[i]*m[i+1]+q[i]
    return m

#обчислення значення функціїї та її похідної у заданій точці
def der(coef,x):
    xav=float(input("Введіть задане значення x: "))
    a=coef[0]
    b=coef[1]
    c=coef[2]
    d=coef[3]
    for i in range(len(x)-1):
        if xav<x[i+1] and xav>x[i]:
            iav=i
    fx=a[iav]+b[iav]*(xav-x[iav])+c[iav]*math.pow((xav-
x[iav]),2)+d[iav]*math.pow((xav-x[iav]),3)
    fdx=2*c[iav]+6*d[iav]*(xav-x[iav])
```

```
    print("Функція в точці "+str(xav)+" дорівнює "+str(fx))
    print("Друга похідна функції в точці "+str(xav)+" дорівнює
"+str(fdx))

#обчислення значення функціїї та її похідної у заданій точці
при N<4
def der_3(x,s):
    xav=float(input("Введіть задане значення x: "))
    for i in range(len(x)-1):
        if xav<x[i+1] and xav>x[i]:
            iav=i
    fx=math.pow(xav,3)*s[iav][0]+math.pow(xav,2)*s[iav][1]+xav
*s[iav][2]+s[iav][3]
    fdx=6*s[iav][0]*xav+2*s[iav][1]
    print("Функція в точці "+str(xav)+" дорівнює "+str(fx))
    print("Друга похідна функції в точці "+str(xav)+" дорівнює
"+str(fdx))

#Обчислення коефіцієнтів для умов другого типу
def coefficient_2(c,n, func,y,h):

    a=[]
    b=[]
    d=[]
    for i in range(n):
        a.append(y[i])

    for i in range(n):
        b.append(0)
    b[-1]=((y[-1]-y[-2])/h[-1])-(func[1]*h[-1]/6)-(c[-1]*h[-
1]*2/3)
    for i in range(n-2,-1,-1):
        b[i]=b[i+1]-(h[i]*(c[i]+c[i+1]))

    for i in range(n-1):
        d.append((c[i+1]-c[i])/(3*h[i]))
    d.append((func[1]-2*c[-1])/(6*h[-1]))
    coef=[]
    coef.append(a)
    coef.append(b)
    coef.append(c)
    coef.append(d)
    print("Коефіцієнти:")
    print("a= "+str(a))
    print("b= "+str(b))
```

```
    print("c= "+str(c))
    print("d= "+str(d))
    return coef

#Обчислення коефіцієнтів для умов першого типу
def coefficient_1(c,n,func, y,h):
    a=[]
    b=[]
    d=[]
    for i in range(n):
        a.append(y[i])
    for i in range(n-1):
        b.append((y[i+1]-y[i])/h[i]-h[i]*(c[i+1]+2*c[i])/3)
    b.append((y[-2]-y[-3])/h[-2]+h[-2]*(2*c[-1]+c[-2])/3)
    for i in range(n-1):
        d.append((c[i+1]-c[i])/(3*h[i]))
    d.append(((y[-1]-y[-2])/h[-1]-(y[-2]-y[-3])/h[-2]-h[-
2]*(c[-2]+2*c[-1])/3-c[-1]*h[-1])/math.pow(h[-1],2))
    coef=[]
    coef.append(a)
    coef.append(b)
    coef.append(c)
    coef.append(d)
    print("Коефіцієнти:")
    print("a= "+str(a))
    print("b= "+str(b))
    print("c= "+str(c))
    print("d= "+str(d))
    return coef

#Обчислення сплайну
def spline(n,x,coef):
    a=coef[0]
    b=coef[1]
    c=coef[2]
    d=coef[3]
    print("Рівняння сплана:")
    for i in range(n):
        print("S"+str(i)+"="+str(a[i])+"+"+str(b[i])+"(x-
"+str(x[i])+")+"+str(c[i])+"(x-
"+str(x[i])+")^2+"+str(d[i])+"(x-"+str(x[i])+")^3, для
xє["+str(x[i])+", "+str(x[i+1])+"]")

#обчислення сплайну при N<4
def spline_3(x,y,m,h,n):
    st=[]
```

```
    s=[]
    for i in range(n):
        st.append([])
        s.append([])
    for i in range(n-1):
        st[i].append(2*y[i]-2*y[i+1]-m[i]*h[i]+m[i+1]*h[i])
        st[i].append(-3*y[i]+3*y[i+1]-m[i+1]*h[i])
        st[i].append(m[i]*h[i])
        st[i].append(y[i])

    for i in range(n):
        s[i].append(st[i][0]/math.pow(h[i],3))
        s[i].append(-
3*x[i]/math.pow(h[i],3)+st[i][1]/math.pow(h[i],2))
        s[i].append(3*st[i][0]*math.pow(x[i],2)/math.pow(h[i],
3)-2*x[i]*st[i][1]/math.pow(h[i],2)+st[i][2]/h[i])
        s[i].append(-
math.pow(x[i],3)*st[i][0]/math.pow(h[i],3)+math.pow(x[i],2)*st
[i][1]/math.pow(h[i],2)-x[i]*st[i][2]/h[i]+st[i][3])

    for i in range(n):
        print("S"+str(i)+"=x^3*"+str(s[i][0])+"+x^2*"+str(s[i]
[1])+"+x*"+str(s[i][2])+"+"+str(s[i][3])+", для x є ["+
str(x[i])+", "+ str(x[i+1])+"]")
    return s

#Побудова сплайну
def drawspline(x,y,coef):
    a=coef[0]
    b=coef[1]
    c=coef[2]
    d=coef[3]
    fig, ax=plt.subplots(figsize=(8,8))
    x_plot=[x[0]]
    y_plot=[]
    step=(x[-1]-x[0])/10000
    for i in range(1,10000):
        x_plot.append(x_plot[i-1]+step)
        for k in range(1, len(x)+1):
            omega_list=[]
            if x[k-1]<=x_plot[i-1]<=x[k]:
                omega=x_plot[i-1]-x[k-1]
                y_plot.append(a[k-1]+b[k-1]*omega+c[k-
1]*(omega**2)+d[k-1]*(omega**3))
    ax.grid()
```

```
    y_plot.append(y[-1])
    ax.plot(np.array(x_plot), np.array(y_plot), c="red")
    ax.plot(x,y,"bo")
    plt.show()

#Виконання звдання для першого типу умов
def first(x,n,func,h,y):
    if n>=3:
        f=slar_1(n,func,h,y)
        c=dist(f,n)
        coef=coefficient_1(c,n,func,y,h)
        spline(n,x,coef)
        der(coef,x)
    else:
        f=slar_1_3(n,y,func,n)
        m=dist(f,n)
        s=spline_3(x,y,m,h,n)
        der_3(x,s)
    drawspline(x,y, coef)

#Виконання звдання для другого типу умов
def second(x,n,func,h,y):
    if n>=3:
        f=slar_2(n,func,h,y)
        c=dist(f,n)
        coef=coefficient_2(c,n,func,y,h)
        spline(n,x,coef)
        der(coef,x)
    else:
        f=slar_2_3(n,y,func,n)
        m=dist(f,n)
        s=spline_3(x,y,m,h,n)
        der_3(x,s)
    drawspline(x,y, coef)

def lab2():
    x=[]
    y=[]
    h=[]
    N=int(input("Ввведіть кількість елементів: "))
    for i in range(N):
         x.append(float(input("Введіть "+str(i)+"-ий x: ")))
         y.append(float(input("Введіть "+str(i)+"-ий y: ")))

    x=sort_xy(x,y)[0]
```

```
    y=sort_xy(x,y)[1]
    print(x)
    print(y)
    n=N-1
    for i in range(n):
        h.append(x[i+1]-x[i])
    t=int(input("Оберіть тип умов (введіть 1, якщо бажаєте
обрати перший тип, або 2, якшщо другий):"))

    an="Y"
    while an!="n":
        func=type(t)
        if t==2:
            second(x,n,func,h,y)
        else:
            first(x,n,func,h,y)



        an=input("Бажаєте побудувати сплайн з іншими
граничними умоваим? (Y/n): ")
        if an=="n":
            break
        else:
            if t==2:
                t=1
            else:
                t=2

lab2()
```

Список використаних джерел

1) Методи обчислень, І.А. Костюшко, Н.Д. Любашенко, В.В. Третиник, 2021