Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

Лабораторная работа №3

Тема: Построение и программная реализация алгоритма сплайн-интерполяции табличных функций.

Студент Сучкова Т.М.
Группа ИУ7-42Б
Оценка (баллы) ______________
Преподаватель Градов В.М.

Москва
2021 г.

Цель работы. Получение навыков владения методами интерполяции таблично заданных функций с помощью кубических сплайнов.

Исходные данные.

1. Таблица функции с количеством узлов N. Задать с помощью формулы $y = x^2$ в диапазоне [0..10] с шагом 1.

2. Значение аргумента x в первом интервале, например, при x=0.5 и в середине таблицы, например, при x= 5.5. Сравнить с точным значением.

Алгоритм
Интерполяция сплайнами

На участке между каждой парой соседних точек имеет кубический полином вида:

$$\psi(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, \qquad (1.10)$$

$$x_{i-1} \le x \le x_i, 0 \le i \le N.$$

В узлах значения многочлена и интерполируемой функции совпадают:

$$\psi(x_{i-1}) = y_{i-1} = a_i, \qquad (1.11)$$

$$\psi(x_i) = y_i = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3, \qquad (1.12)$$

$$h_i = x_i - x_{i-1}, 1 \le i \le N.$$

Число таких уравнений меньше числа неизвестных в два раза. Недостающие уравнения получают, приравнивая во внутренних узлах первые и вторые производные, вычисляемые по коэффициентам на соседних участках:

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

$$\psi''(x) = 2c_i + 6d_i(x - x_{i-1}), \quad x_{i-1} \le x \le x_i,$$

$$c_{i+1} = c_i + 3d_i h_i, \quad 1 \le i \le N - 1. \qquad (1.14)$$

Недостающие условия можно получить, полагая, например, что вторая производная равна нулю на концах участка интерполирования:

$$\psi''(x_0) = 0, c_1 = 0, \qquad (1.15)$$

$$\psi''(x_N) = 0, c_N + 3d_N h_N = 0, \qquad (1.16)$$

Уравнения (1.11)-(1.16) позволяют определить все 4N неизвестных коэффициентов: a_i ,b_i ,c_i ,d_i (1<=i<=N).

Для упрощения решения системы приведем ее к специальному виду.
Из (1.11) находятся сразу все коэффициенты a_i. Из (1.14) и (1.16) следует

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, \quad 1 \le i \le N-1. \tag{1.17}$$

$$d_N = -\frac{c_N}{3h_N} \tag{1.18}$$

Исключим из (1.12) d_i с помощью (1.17), получим:

$$b_i = \frac{y_i - y_{i-1}}{h_i} - h_i\frac{c_{i+1} - 2c_i}{3}, \quad 1 \le i \le N-1. \tag{1.19}$$

Из (1.12) и (1.18):

$$b_N = \frac{y_N - y_{N-1}}{h_N} - h_N\frac{2c_N}{3} \tag{1.20}$$

Для определения коэффициентов c_i получим следующую систему:

$$c_1 = 0,$$

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3\left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}}\right), 2 \le i \le N-1 \tag{1.21}$$

$$c_{N+1} = 0.$$

После нахождения коэффициентов c_i остальные коэффициенты определяют по следующим формулам:

$$a_i = y_{i-1}, \ 1 \le i \le N,$$

$$b_i = \frac{y_i - y_{i-1}}{h_i} - h_i\frac{c_{i+1} - 2c_i}{3}, \quad 1 \le i \le N-1,$$

$$b_N = \frac{y_N - y_{N-1}}{h_N} - h_N\frac{2c_N}{3},$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, \quad 1 \le i \le N-1,$$

$$d_N = -\frac{c_N}{3h_N}.$$

Применительно к (1.21) можно записать:

$$c_i = \xi_{i+1}c_{i+1} + \eta_{i+1}, \tag{1.22}$$

где ξ_{i+1}, η_{i+1} - некоторые, не известные пока прогоночные коэффициенты; $c_{i-1} = \xi_i c_i + \eta_i$.

Подставляя последнее выражение в (1.21) и преобразуя, получим

$$c_i = -\frac{h_i}{h_{i-1}\xi_i + 2(h_{i-1} + h_i)} c_{i+1} + \frac{f_i - h_{i-1}\eta_i}{h_{i-1}\xi_i + 2(h_{i-1} + h_i)}. \qquad (1.23)$$

Сравнивая (1.22) и (1.23), получим

$$\xi_{i+1} = -\frac{h_i}{h_{i-1}\xi_i + 2(h_{i-1} + h_i)}, \eta_{i+1} = \frac{f_i - h_{i-1}\eta_i}{h_{i-1}\xi_i + 2(h_{i-1} + h_i)}. \qquad (1.24)$$

В этих формулах введено обозначение $f_i = 3(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}})$.

Из условия $c_1 = 0$, следует $\xi_2 = 0, \eta_2 = 0$.

По формулам (1.24) при известных ξ_2, η_2, равных нулю, вычисляют прогоночные коэффициенты ξ_{i+1}, η_{i+1} (2<=i<=N)(прямой ход). Затем по формулам (1.22) при условии $c_{N+1} = 0$ определяют все c_i (обратный ход).

Код программы

main.py:

```
from math import ceil

def dif_matrix_create(table, n):
    for i in range(n):
        tmp = []

        for j in range(n - i):
            tmp.append((table[i + 1][j] - table[i + 1][j + 1]) / (table[0][j] - table[0][i + j + 1]))
        table.append(tmp)

    return table

def choose_points(table, n, x):
    tab_len = len(table[0])
    i_near = min(range(tab_len), key = lambda i: abs(table[0][i] - x))
    n_required = ceil(n / 2)

    if (i_near + n_required + 1 > tab_len):
        i_end = tab_len
        i_start = tab_len - n
    elif (i_near < n_required):
        i_start = 0
        i_end = n
    else:
```

```
        i_start = i_near - n_required + 1
        i_end = i_start + n

    return [table[0][i_start:i_end], table[1][i_start:i_end]]

def newton_interpolation(table, n, x):
    table = choose_points(table, n + 1, x)

    dif_matrix = dif_matrix_create(table, n)
    tmp = 1
    res = 0

    for i in range(n + 1):
        res += tmp * dif_matrix[i + 1][0]
        tmp *= (x - dif_matrix[0][i])

    return res

def f(x):
    return x * x

def xy_table_create(x_beg, step, n):
    x_arr = [(x_beg + i * step) for i in range(n)]
    y_arr = [f(elem) for elem in x_arr]

    return x_arr, y_arr

def xy_table_print(x_arr, y_arr):
    l = len(x_arr)

    for i in range(l):
        print("%.6f %.6f" % (x_arr[i], y_arr[i]))

    print()

def spline_interpolation(x_arr, y_arr, x_value):
    n = len(x_arr)
    i_near = min(range(n), key = lambda i: abs(x_arr[i] - x_value))

    h = [0 if not i else x_arr[i] - x_arr[i - 1] for i in range(n)] # шаг

    a_arr = [0 if i < 2 else h[i-1] for i in range(n)]
    b_arr = [0 if i < 2 else -2 * (h[i - 1] + h[i]) for i in range(n)]
    d_arr = [0 if i < 2 else h[i] for i in range(n)]
    f_arr = [0 if i < 2 else -3 * ((y_arr[i] - y_arr[i - 1]) / h[i] - (y_arr[i - 1] - y_arr[i - 2]) / h[i - 1]) for i
in range(n)]

    # прямой ход
    ksi = [0 for i in range(n + 1)]
    eta = [0 for i in range(n + 1)]
    for i in range(2, n):
        ksi[i + 1] = d_arr[i] / (b_arr[i] - a_arr[i] * ksi[i])
```

```
        eta[i + 1] = (a_arr[i] * eta[i] + f_arr[i]) / (b_arr[i] - a_arr[i] * ksi[i])

    # обратный ход
    c = [0 for i in range(n + 1)]
    for i in range(n - 2, -1, -1):
        c[i] = ksi[i + 1] * c[i + 1] + eta[i + 1]


    a = [0 if i < 1 else y_arr[i-1] for i in range(n)]
    b = [0 if i < 1 else (y_arr[i] - y_arr[i - 1]) / h[i] - h[i] / 3 * (c[i + 1] + 2 * c[i]) for i in range(n)]
    d = [0 if i < 1 else (c[i + 1] - c[i]) / (3 * h[i]) for i in range(n)]

    res = a[i_near] + b[i_near] * (x_value - x_arr[i_near - 1]) + c[i_near] * ((x_value - x_arr[i_near - 1]) ** 2) + d[i_near] * ((x_value - x_arr[i_near - 1]) ** 3)

    return res

x_beg = float(input("Input beginning value of x: "))
x_step = float(input("Input step for x value: "))
n = int(input("Input points amount: "))

x_tab, y_tab = xy_table_create(x_beg, x_step, n)
print("\nCreated table:")
xy_table_print(x_tab, y_tab)

x = float(input("Input x: "))

res_spline = spline_interpolation(x_tab, y_tab, x)
res_newton = newton_interpolation([x_tab, y_tab], 3, x)
print("\nSpline interpolation: ", res_spline)
print("Newton interpolation: ", res_newton)
print("\nf(x)              : ", f(x))
print("Spline error      : ", abs(f(x) - res_spline))
print("Newton error      : ", abs(f(x) - res_newton), "\n")
```

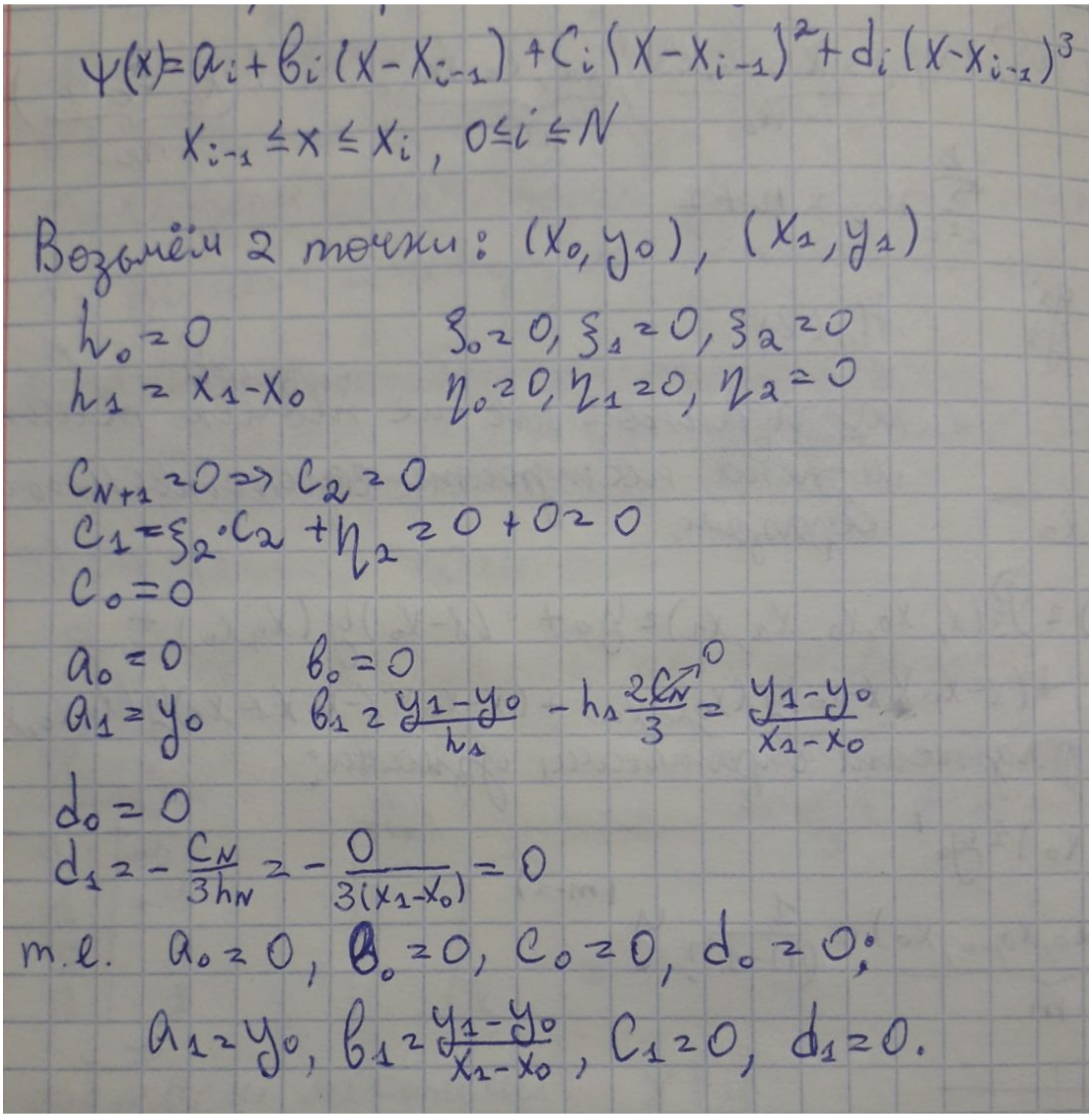
Результаты работы программы.

```
Input beginning value of x: 0
Input step for x value: 1
Input points amount: 11

Created table:
0.000000 0.000000
1.000000 1.000000
2.000000 4.000000
3.000000 9.000000
4.000000 16.000000
5.000000 25.000000
6.000000 36.000000
7.000000 49.000000
8.000000 64.000000
9.000000 81.000000
10.000000 100.000000

Input x: 0.5

Spline interpolation:  0.0
Newton interpolation:  0.25

f(x)                :  0.25
Spline error        :  0.25
Newton error        :  0.0
```

```
Input beginning value of x: 0
Input step for x value: 1
Input points amount: 11

Created table:
0.000000 0.000000
1.000000 1.000000
2.000000 4.000000
3.000000 9.000000
4.000000 16.000000
5.000000 25.000000
6.000000 36.000000
7.000000 49.000000
8.000000 64.000000
9.000000 81.000000
10.000000 100.000000

Input x: 5.5

Spline interpolation:  30.247169811320756
Newton interpolation:  30.25

f(x)                :  30.25
Spline error        :  0.0028301886792441167
Newton error        :  0.0
```

Вопросы при защите лабораторной работы.

1. Получить выражения для коэффициентов кубического сплайна, построенного на двух точках.



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

$$x_{i-1} \le x \le x_i, \quad 0 \le i \le N$$

Возьмём 2 точки: (x_0, y_0), (x_1, y_1)

$h_0 = 0$ $\quad$ $\xi_0 = 0, \xi_1 = 0, \xi_2 = 0$

$h_1 = x_1 - x_0$ $\quad$ $\eta_0 = 0, \eta_1 = 0, \eta_2 = 0$

$c_{N+1} = 0 \Rightarrow c_2 = 0$

$c_1 = \xi_2 \cdot c_2 + \eta_2 = 0 + 0 = 0$

$c_0 = 0$

$a_0 = 0$ $\quad$ $b_0 = 0$

$a_1 = y_0$ $\quad$ $b_1 = \frac{y_1 - y_0}{h_1} - h_1 \frac{2\cancelto{0}{c_N}}{3} = \frac{y_1 - y_0}{x_1 - x_0}$

$d_0 = 0$

$d_1 = -\frac{c_N}{3h_N} = -\frac{0}{3(x_1 - x_0)} = 0$

т.е. $a_0 = 0$, $b_0 = 0$, $c_0 = 0$, $d_0 = 0$;

$a_1 = y_0$, $b_1 = \frac{y_1 - y_0}{x_1 - x_0}$, $c_1 = 0$, $d_1 = 0$.

2. Выписать все условия для определения коэффициентов сплайна, построенного на 3-х точках

Если задано 3 точки, значит, имеется 2 интервала. В каждом интервале по 4 коэффициента. Сплайн должен пройти через 2 точки — 4 условия, производные в средней точке совпадают — 2 условия, производные на концах = 0 — 2 условия.

Итого: 8 условий.

3. Определить начальные значения прогоночных коэффициентов, если принять, что для коэффициентов сплайна справедливо C1=C2.

$$\begin{cases} C_i = \xi_{i+1} C_{i+1} + \eta_{i+1} \\ C_1 = C_2 = 0 \end{cases} \Rightarrow \begin{cases} C_1 = \xi_2 \cdot C_2 + \eta_2 \\ C_1 = C_2 = 0 \end{cases} \Rightarrow$$

$$\Rightarrow C_1(1-\xi_2) = \eta_2 = 0$$

$$\underline{\eta_2 = 0} \Rightarrow 1 - \xi_2 = 0$$

$$\underline{\xi_2 = 1}$$

4. Написать формулу для определения последнего коэффициента сплайна C_N , чтобы можно было выполнить обратный ход метода прогонки, если в качестве граничного условия задано $kC_{N-1}+mC_N=p$, где k, m и p - заданные числа.

4. $kC_{N-1} + mC_N = p \; ; C_N - ?$

$$C_{i-1} = \xi_i C_i + \eta_i \Rightarrow C_{N-1} = \xi_N C_N + \eta_N$$

$$k(\xi_N C_N + \eta_N) + mC_N = p$$

$$C_N(k\xi_N + m) + k\eta_N = p$$

$$C_N(k\xi_N + m) = p - k\eta_N$$

$$\boxed{C_N = \frac{p - k\eta_N}{k\xi_N + m}}$$