



Министерство науки и высшего образования 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»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 3

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

Студент: Артемьев И.О.

Группа: ИУ7-43Б

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) Значение аргумента x
- 2) Таблица функции с количеством узлов N , которая задается с помощью формулы $y = x^3$ в диапазоне $[0..10]$ с шагом 1

Выходные данные

- 1) Значение $y(x)$
- 2) Сравнить результат интерполяции кубического сплайна с полиномом Ньютона 3-ей степени

Анализ алгоритма

Для кубического сплайна значение y вычисляется следующим образом

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

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

Прежде необходимо найти все коэффициенты:

- 1) Коэффициент a :

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

- 2) Коэффициент c (вычисляется раньше, поскольку участвует в вычислении коэффициентов b и d):

С помощью метода Гаусса получаем, что

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

где ξ_{i+1}, η_{i+1} - некоторые, не известные пока прогоночные коэффициенты;

Затем сами прогоночные коэффициенты должны вычисляться следующим образом

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

Используя дополнительное обозначение:

$$f_i = 3\left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}}\right).$$

3) Коэффициент b:

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

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

4) Коэффициент d:

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

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

Код программы

```
#####

main.py

#####

from process_funcs import get_val_pol # С прошлых лабораторных
from myio import read_data
from spline import Spline

def main():
    table = read_data("test.txt")
    if table is None:
        exit(0)

    newton_table = []

    for i in range(len(table[0])):
        newton_table.append([table[0][i], table[1][i]])

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

    print("Spline: ", "{:.5f}".format(Spline(table).get_spline(x)))
    print("The Newton polynomial of the third degree: ", get_val_pol(newton_table, 3, x))

if __name__ == "__main__":
    main()
```

```
#####

spline.py

#####

class Spline:
    def __init__(self, table):
        self.table = table
        self.x_arr = table[0]
        self.y_arr = table[1]
        self.size = len(self.y_arr)
```

```

def __get_a_coeff(self):
    return self.y_arr[:-1]

def __get_b_coeff(self, c_coeff):
    b_coeff = []

    for i in range(1, self.size - 1):
        h = self.x_arr[i] - self.x_arr[i - 1]

        b_cur = (self.y_arr[i] - self.y_arr[i - 1]) / h - (h * (c_coeff[i] + 2 * c_coeff[i - 1])) / 3

        b_coeff.append(b_cur)

    h = self.x_arr[self.size - 1] - self.x_arr[self.size - 2]
    b_coeff.append((self.y_arr[self.size - 1] - self.y_arr[self.size - 2]) / h - (h * 2 * c_coeff[i]) / 3)

    return b_coeff

def __get_c_coeff(self):
    c_coeff = [0] * (self.size - 1)

    ksi_arr = [0, 0]
    teta_arr = [0, 0]

    for i in range(2, self.size):
        h1 = self.x_arr[i] - self.x_arr[i - 1]
        h2 = self.x_arr[i - 1] - self.x_arr[i - 2]

        fi = 3 * ((self.y_arr[i] - self.y_arr[i - 1]) / h1 - (self.y_arr[i - 1] - self.y_arr[i - 2]) / h2)

        ksi_cur = - h1 / (h2 * ksi_arr[i - 1] + 2 * (h2 + h1))
        teta_cur = (fi - h1 * teta_arr[i - 1]) / (h1 * ksi_arr[i - 1] + 2 * (h2 + h1))

        ksi_arr.append(ksi_cur)
        teta_arr.append(teta_cur)

    c_coeff[self.size - 2] = teta_arr[len(teta_arr) - 1]

    for i in range(self.size - 2, 0, -1):
        c_coeff[i - 1] = ksi_arr[i] * c_coeff[i] + teta_arr[i]

```

```

    return c_coeff

def __get_d_coeff(self, c_coeff):
    d_coeff = []

    for i in range(1, len(self.x_arr) - 1):
        h = self.x_arr[i] - self.x_arr[i - 1]

        d_cur = (c_coeff[i] - c_coeff[i - 1]) / (3 * h)

        d_coeff.append(d_cur)

    h = self.x_arr[self.size - 1] - self.x_arr[self.size - 2]
    d_coeff.append((- c_coeff[i]) / (3 * h))

    return d_coeff

def get_spline_coeffs(self):
    a_coeff = self.__get_a_coeff()
    c_coeff = self.__get_c_coeff()
    b_coeff = self.__get_b_coeff(c_coeff)
    d_coeff = self.__get_d_coeff(c_coeff)

    return a_coeff, b_coeff, c_coeff, d_coeff

def get_spline(self, x):
    coeffs = self.get_spline_coeffs()

    pos = 1
    while (pos < len(self.x_arr) and self.x_arr[pos] < x):
        pos += 1
    pos -= 1

    h = x - self.x_arr[pos]

    result = 0
    for i in range(len(coeffs)):
        result += coeffs[i][pos] * pow(h, i)

```

```
    return result

#####

myio.py

#####

def read_data(filename):

    try:

        file = open(filename, "r")

    except:

        print("Invalid filename")

        return

    table = []
    x_arr = []
    y_arr = []

    for line in file:

        try:

            arr = [float(numb) for numb in line.split()]

            x_arr.append(arr[0])
            y_arr.append(arr[1])

        except:

            print("Invalid line data")

            file.close()

            return

    table.append(x_arr)
    table.append(y_arr)

    file.close()

    return table
```

Пример работы программы



```
ilyaartemev@Air-Ilya: ~/Desktop/iu7-ca/lab_03/src
> python3 main.py
Input x: 0.5
Spline: 0.34151
The Newton polynomial of the third degree: 0.25
> python3 main.py
Input x: 5.5
Spline: 30.25035
The Newton polynomial of the third degree: 30.25
>
```

Результаты

x	y	Кубический сплайн	Полином Ньютона 3-ей степени
0.5	0.25	0.34151	0.25
5.5	30.25	30.25035	30.25

Вопросы при защите лабораторной работы

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

Пусть заданы две точки: (x_1, y_1) , (x_2, y_2)

Тогда для кубического сплайна, который имеет вид:

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

Коэффициент $A = y_1$

Поскольку $C(N) = 0$ и сплайн строится всего по двум точкам, получаем, что $C(N) = C$, значит $C = 0$

Также через коэффициент C находим:

коэффициент $D = -C(N) / 3h(N) = 0$

$$\text{коэффициент } B = \frac{y_N - y_{N-1}}{h_N} - h_N \frac{2c_N}{3} = (y_2 - y_1) / (x_2 - x_1)$$

Ответ: $A = y_1$, $B = (y_2 - y_1) / (x_2 - x_1)$, $C = 0$, $D = 0$

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

Пусть дано три точки : (x_1, y_1) , (x_2, y_2) , (x_3, y_3)

Будет 8 коэффициентов, которые будут находиться следующим образом:

1) Через значения узлов: $w_1(x_1) = y_1$, $w_2(x_2) = y_2$, $w_1(x_2) = y_2$, $w_2(x_3) = y_3$, то есть:

- $a_1 = y_1$
- $a_2 = y_2$
- $a_1 + b_1(x_2 - x_1) + c_1(x_2 - x_1)^2 + d_1(x_2 - x_1)^3 = y_2$
- $a_2 + b_2(x_3 - x_2) + c_2(x_3 - x_2)^2 + d_2(x_3 - x_2)^3 = y_3$

2) Через производные:

- $w_1'(x_2) = w_2'(x_2) \Rightarrow b_1 + 2c_1(x_2 - x_1) + 3d_1(x_2 - x_1)^2 = b_2$
- $w_1''(x_2) = w_2''(x_2) \Rightarrow c_1 + 3d_1(x_2 - x_1) = c_2$

- $w_1''(x_1) = 0$
- $w_2''(x_3) = 0$

3. Определить начальные значения прогоночных коэффициентов, если принять, что для коэффициентов сплайна справедливо $C_1=C_2$.

$$C_{i-1} = \xi_i C_i + \eta_i$$

Тогда, если $C_1 = C_2$, то: $\xi_2 = 1, \eta_2 = 0$

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.

$$C(N-1) = e(N) * C(N) + n(N)$$

Тогда, если $k * C(N-1) + m * C(N) = p$, имеем:

$$C(N) = (p - k * n(N)) / (k * e(N) + m)$$