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: Фролов Е.А.

Группа: ИУ7-45Б

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*x 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

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

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

В начале нужно найти все коэффициенты:

1) Коэффициент a:

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

2) Коэффициент c вычисляется до того, как вычисляются b и d, потому что c участвует в процессе вычислении 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;

Сами прогоночные коэффициенты вычисляются 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:

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

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

4) Коэффициент d:

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

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

Код программы:

```
def in_file(file_name):
    try:
        file = open(file_name, "r")
    except:
        print("Ошибка: Нет такого файла")

    table = []
    num_line = 0

    array_x = []
    array_y = []

    for line in file:
        arr = []
        try:
            arr = [float(num) for num in line.split()]
            array_x.append(arr[0])
            array_y.append(arr[1])
        except:
            file.close()
            return []

        num_line += 1

    table.append(array_x)
    table.append(array_y)

    file.close()
    return table
```

```python
def calc_factors_spline(array_x, array_y):
    a_factor = array_y[:-1]

    size = len(array_y)

    c_factor = [0] * (size - 1)

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

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

        f= 3 * ((array_y[i] - array_y[i - 1]) / h1 - (array_y[i - 1] - array_y[i - 2]) / h2)

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

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

    c_factor[size - 2] = teta_arr[len(teta_arr) - 1]

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

    b_factor = []

    for i in range(1, len(array_x) - 1):
        h = array_x[i] - array_x[i - 1]

        b_cur = (array_y[i] - array_y[i - 1]) / h - (h * (c_factor[i] + 2 * c_factor[i - 1])) / 3

        b_factor.append(b_cur)

    h = array_x[size - 1] - array_x[size - 2]
    b_factor.append((array_y[size - 1] - array_y[size - 2]) / h - (h * 2 * c_factor[i]) / 3)

    d_factor = []

    for i in range(1, len(array_x) - 1):
        h = array_x[i] - array_x[i - 1]
        d_cur = (c_factor[i] - c_factor[i - 1]) / (3 * h)
        d_factor.append(d_cur)

    h = array_x[size - 1] - array_x[size - 2]
    d_factor.append((- c_factor[i]) / (3 * h))

    return a_factor, b_factor, c_factor, d_factor
```

```
82  def spline(array_x, array_y, x):
83      factors = calculate_factors_spline(array_x, array_y)
84      side = 1
85
86      while (side < len(array_x) and array_x[side] < x):
87          side += 1
88
89      side -= 1
90
91      h = x - array_x[side]
92      y = 0
93
94      y += factors[0][side]
95      y += factors[1][side] * h
96      y += factors[2][side] * h * h
97      y += factors[3][side] * h * h * h
98
99      return y
100
101 def find_div_difference(x1, y1, x2, y2, proizvod):
102     if (abs(x1 - x2) > 1e-7):
103         return (y1 - y2) / (x1 - x2)
104     else:
105         return proizvod
106
107 def find_index_table(table, x, power):
108     if (power >= len(table)):
109         return -1
110
111     ind = -1
112     flag = 0
113
114     for i in range(len(table)):
115         if (x <= table[i][0]):
116             ind = i - power // 2
117             flag = 1
118             break
119
120     if (ind < 0):
121         ind = 0
122
123     if (flag == 0) or (ind + power + 1 > len(table) - 1):
124         ind = len(table) - power - 1
125
126     return ind
```

```python
def newton_polynom(table, x, power):
    ind = find_index_table(table, x, power)
    if (ind < 0):
        return

    result = table[ind][1]

    for i in range (power):
        factor = 1
        for k in range (i + 1):
            factor *= (x - table[ind + k][0])
            for j in range (power - i):
                table[ind + j][1] = find_div_difference(table[ind + j][0], table[ind + j][1], table[ind + j + i + 1][0], table[ind + j + 1][1], table[ind + j][2])
        result += (factor * table[ind][1])
    return result

if __name__ == "__main__":
    table = read_file("input.txt")
    newton_table = []

    for i in range (len(table[0])):
        newton_table.append([table[0][i], table[1][i]])
    try:
        x = float(input("\nX: "))
        print("\n\nСравнение результатов для x = ", x)
        print("_____________________________________\n")

        print("Сплайном: ", "{:.5f}".format(spline(table[0], table[1], x)))
        print("Полином Ньютона 3ей степени: ", newton_polynom(newton_table, x, 3))
    except:
        print("\n\nОшибка: введено не число")
```

Пример работы программы

```
X: 0.5


Сравнение результатов для x = 0.5

_____________________________________

Сплайном: 0.34151
Полином Ньютона 3ей степени: 0.25
PS C:\Users\gimna\Desktop\BMSTU\ВычАлг\lab_03>  c:; c
STU\ВычАлг\lab_03\main.py'
X: 5.5


Сравнение результатов для x = 5.5

_____________________________________

Сплайном: 30.24811
Полином Ньютона 3ей степени: 30.25
```

Результаты

x	y	Кубический сплайн	Полином Ньютона 3й степени
0.5	0.25	0.34151	0.25
2.5	6.25	6.25660	6.25
5.5	30.25	30.24811	30.25

Вопросы при защите лабораторной работы

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

Пусть заданы две точки: (x1, y1), (x2, y2)

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*** = y1

Так как сплайн строится по двум точкам и ***C***(N) = 0, то ***C***(N) = ***C***, значит ***C*** = 0

Через коэффициент ***C*** находим:

коэффициент ***D*** = - ***C***(N) / 3h(N) = 0

коэффициент ***B*** = (y(N) - y(N-1)) / h(N) - h(N)*(2Cn) / 3 = (y2 - y1) / (x2 - x1)

A = y1,

B = (y2 - y1) / (x2 - x1),

C = 0,

D = 0.

Сплайном будет прямая, проходящая ч-з две точки

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

Пусть дано три точки : (x1, y1), (x2, y2), (x3, y3)

Будет 8 коэффициентов, которые будут находиться следующим образом:

1) Через значения узлов: w1(x1) = y1, w2(x2) = y2, w1(x2) = y2, w2(x3) = y3:
   - a1 = y1
   - a2 = y2
   - a1 + b1*(x2 - x1) + c1*(x2 - x1)^2 + d1*(x2 - x1)^3 = y2
   - a2 + b2*(x3 - x2) + c2*(x3 - x2)^2 + d2*(x3 - x2)^3 = y3

2) Через производные:
   - w1’(x2) = w2’(x2) => b1+ 2*c1*(x2 - x1) + 3*d1*(x2 - x1)^2 = b2
   - w1”(x2) = w2”(x2) => c1 + 3*d1*(x2 - x1) = c2
   - w1”(x1) = 0
   - w2”(x3) = 0

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

$$c_{i-1} = \xi_i c_i + \eta_i$$

Если C1 = C2, то: ξ2 = 1, η2 = 0

4. Написать формулу для определения последнего коэффициента сплайна CN , чтобы можно было выполнить обратный ход метода прогонки, если в качестве граничного условия задано kCN-1+mCN=p, где k,m и p - заданные числа.

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

Если k*C(N-1) + m*C(N) = p, получаем: C(N) = (p - k*n(N)) / (k*e(N) + m)