



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

«Построение и программная реализация алгоритма
сплайн-интерполяции табличных функций»

Студент Маслова Марина Дмитриевна
фамилия, имя, отчество

Группа ИУ7-43Б

Оценка (баллы) _____

Преподаватель Градов Владимир Михайлович
фамилия, имя, отчество

Оглавление

Исходные данные	3
Описание алгоритма	3
Код программы	4
Результат работы	10
Контрольные вопросы	11

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

Исходные данные

1. Таблица функции, заданная с помощью формулы $y = x^2$ в диапазоне $[0.. 10]$ с шагом 1.
2. Значение аргумента x в первом интервале и в середине таблицы.

Описание алгоритма

Участок между каждой парой соседних точек интерполируется кубическим полиномом вида:

$$\psi(x) = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3$$

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

Коэффициенты определяются следующим образом:

1. Определяются коэффициенты c_i методом прогонки:

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

где ξ_{i+1} , η_{i+1} — прогоночные коэффициенты, определяющиеся по формулам:

$$\xi_{i+1} = - \frac{h_i}{h_{i-1} \xi_i + 2(h_{i-1} + h_i)},$$

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

где

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

При этом начальные и конечные условия определяются из равенства нулю производных в крайних точках заданного промежутка:

$$c_1 = 0, c_{N+1} = 0, \xi_2 = 0, \eta_2 = 0$$

2. По известным коэффициентам c_i остальные коэффициенты определяются следующим образом:

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

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

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

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

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

Код программы

Код программы представлен на листингах 1-2.

Листинг 1. spline.py

```
"""
    Модуль сплайн-интерполяции табличных функций
"""
def find_h_coefs(func):
    """
        Поиск коэффициентов h
    """
    h_coefs = [func[i][0] - func[i - 1][0] for i in range(1, len(func))]
    return h_coefs

def find_f_coefs(func, h):
    """
        Поиск коэффициентов f
    """
    f_coefs = [0.]
    for i in range(2, len(func)):
```

```

        dy1 = func[i][1] - func[i - 1][1]
        dy2 = func[i - 1][1] - func[i - 2][1]
        f_coefs.append(3 * (dy1 / h[i - 1] - dy2 / h[i - 2]))

    return f_coefs

def find_ksi_coefs(h):
    """
        Поиск коэффициентов  $\xi$ 
    """
    ksi_coefs = [0., 0.]

    for i in range(2, len(h) + 1):
        cur_ksi = (- h[i - 1] / (h[i - 2] * ksi_coefs[i - 1]
                                + 2 * (h[i - 2] + h[i - 1])))
        ksi_coefs.append(cur_ksi)

    return ksi_coefs

def find_eta_coefs(ksi, h, f):
    """
        Поиск коэффициентов  $\eta$ 
    """
    eta_coefs = [0., 0.]

    for i in range(2, len(h) + 1):
        cur_eta = ((f[i - 1] - h[i - 2] * eta_coefs[i - 1]) /
                    (h[i - 2] * ksi[i - 1] + 2 * (h[i - 2] + h[i - 1])))
        eta_coefs.append(cur_eta)

    return eta_coefs

def find_c_coefs(ksi, eta):
    """
        Подсчет коэффициентов  $c$ 
    """
    num = len(eta)
    c_coefs = [0., 0.]

    for i in range(num - 1, 1, -1):
        c_coefs.insert(1, ksi[i] * c_coefs[1] + eta[i])

    return c_coefs[:-1]

def get_c_coefs(func, h_coefs):
    """
        Поиск коэффициентов при второй степени сплайна
    """
    ksi_coefs = find_ksi_coefs(h_coefs)
    f_coefs = find_f_coefs(func, h_coefs)
    eta_coefs = find_eta_coefs(ksi_coefs, h_coefs, f_coefs)
    c_coefs = find_c_coefs(ksi_coefs, eta_coefs)
    return c_coefs

def get_b_coefs(func, h, c):

```

```

"""
    Подсчет коэффициентов b
"""
b_coefs = []
N = len(func) - 1

for i in range(1, N):
    cur_b = ((func[i][1] - func[i - 1][1]) / h[i - 1]
              - (h[i - 1] * (c[i] + 2 * c[i - 1])) / 3)

    b_coefs.append(cur_b)

b_coefs.append((func[N][1] - func[N - 1][1]) / h[N - 1]
                - h[N - 1] * 2 * c[N - 1] / 3)

return b_coefs

def get_d_coefs(h, c):
    """
        Подсчет коэффициентов b
    """
    d_coefs = []
    N = len(h)

    for i in range(1, N):
        cur_d = (c[i] - c[i - 1]) / (3 * h[i - 1])
        d_coefs.append(cur_d)

    d_coefs.append(- c[N - 1] / (3 * h[N - 1]))

    return d_coefs

def get_x_position(func, x):
    """
        Поиск положения элемента в таблице
    """
    pos = 0

    while pos < len(func) and x > func[pos][0]:
        pos += 1

    pos -= 1

    if pos > len(func) - 2:
        pos = len(func) - 2

    if pos < 0:
        pos = 0

    return pos

def count_spline(func, x):
    """
        Подсчет значения функции сплайн-интерполяцией
    """
    position = get_x_position(func, x)

```

```

h_coefs = find_h_coefs(func)
a_coefs = [point[1] for point in func[:-1]]
c_coefs = get_c_coefs(func, h_coefs)
b_coefs = get_b_coefs(func, h_coefs, c_coefs)
d_coefs = get_d_coefs(h_coefs, c_coefs)

dif = x - func[position][0]
answer = a_coefs[position]
answer += b_coefs[position] * dif
answer += c_coefs[position] * dif ** 2
answer += d_coefs[position] * dif ** 3

return answer

```

Листинг 2. main.py

```

"""
Модуль для запуска программы
ЛАБОРАТОРНАЯ РАБОТА №3
ПОСТРОЕНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА
СПЛАЙН-ИНТЕРПОЛЯЦИИ ТАБЛИЧНЫХ ФУНКЦИЙ
"""

import argparse

import table_func
import newton
import spline

def create_args():
    """
    Добавление аргументов командной строки
    """
    parser = argparse.ArgumentParser()
    parser.add_argument('file_name', nargs='?', default='data/data.csv')
    args = parser.parse_args()

    return args

if __name__ == "__main__":
    ARGS = create_args()

    try:
        func_table = table_func.read_table(ARGS.file_name)

        func_table.sort(key=lambda table: table[0])
        table_func.print_table(func_table)

        x = float(input("\nВведите значения аргумента для интерполяции:
    "))

    except FileNotFoundError:
        print("\nТакого файла не существует!")

    except ValueError:
        print("\nНечисловые данные недопустимы!")
        print("\nПроверьте содержимое файла или введенный аргумент!")

    except EOFError:

```

```

        print("\nПустой файл!")

except TypeError:
    print("\nВ файле должно быть три столбца с данными!")

else:
    print("Точное значение:")
    print("{:.6f}".format(x * x))

    print("Результат сплайн-интерполяции:")
    print("{:.6f}".format(spline.count_spline(func_table, x)))

    print("Результат интерполяции полиномом Ньютона 3-ей степени:")
    print("{:.6f}".format(newton.find_newton_y(func_table, x, 3)))

```

Листинг 3. table_func.py

```

"""
Модуль ввода-вывода табличных функций
"""

def read_table(file_name):
    """
    Чтение табличной функции из файла
    """
    func_table = []

    with open(file_name, "r") as file:
        for i, rec in enumerate(file):
            func_table.append(list(map(float, rec.split())))

            if len(func_table[i]) != 2:
                raise TypeError

    if not func_table:
        raise EOFError

    return func_table

def print_table(table):
    """
    Вывод табличной функции
    """
    if table:
        print("Загруженная таблица:")
        print("      x      y")
    else:
        print("Пустой файл!")

    for rec in table:
        print("      {:.2f}   {:.9.2f}".format(rec[0], rec[1]))

```

Листинг 4. newton.py

```

"""
Модуль интерполяции полиномом Ньютона
"""
def find_x_position(table, arg):
    """
    Поиск положения заданного аргумента в таблице
    """

```



```

"""
prev_arg_index = 0
num_table_args = len(table)

while (prev_arg_index < num_table_args and
       arg > table[prev_arg_index][0]):
    prev_arg_index += 1

return prev_arg_index - 1

def create_calc_table(table, arg_position, coef_num):
    """
        Выбор значений для подсчета коэффициентов полинома
    """
    res_table = []

    begin = arg_position - coef_num // 2 + 1
    begin = begin if begin >= 0 else 0
    begin = (begin if begin + coef_num < len(table)
            else len(table) - coef_num)

    for i in range(begin, begin + coef_num):
        res_table.append([x for x in table[i]])

    return res_table

def calc_divided_difference(y0, y1, x0, x1):
    """
        Подсчет разделенной разности
    """
    return (y0 - y1) / (x0 - x1)

def calc_coef(calc_table, first_col):
    """
        Подсчет коэффициентов полинома
        с помощью разделенных разностей
    """
    for y in range(first_col, len(calc_table)):
        for i in range(0, len(calc_table) - y):
            calc_table[i].append(calc_divided_difference(
                calc_table[i][y],
                calc_table[i + 1][y],
                calc_table[i][0], calc_table[i + y][0]))

def calc_func(calc_table, arg):
    """
        Подсчет значения функции с помощью таблицы разделенных разностей
    """
    result = 0
    mul = 1

    for i in range(1, len(calc_table[0])):
        result += calc_table[0][i] * mul
        mul *= arg - calc_table[i - 1][0]

    return result

```

```
def find_newton_y(table, arg, power):
    """
        Поиск значения отсортированной по аргументу табличной
        функции с помощью интерполяции полиномом Ньютона
    """
    arg_position = find_x_position(table, arg)
    calculaton_table = create_calc_table(table, arg_position, power + 1)
    calc_coef(calculaton_table, 1)
    result = calc_func(calculaton_table, arg)

    return result
```

Результат работы

Результаты интерполяции кубическим сплайном, полиномом Ньютона
3 степени и точное значение функции:

x	Точное значение	Сплайн	Полином Ньютона
0.5	0.25	0.341506	0.25
4.0	16.00	16.00	16.00
5.5	30.25	30.250345	30.25
7.5	56.25	56.256561	56.25
9.5	90.25	90.341506	90.25

Контрольные вопросы

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

Пусть даны две точки (x_0, y_0) и (x_1, y_1) .

В интервале аргумента $[x_0, x_1]$ значение коэффициента $c = c_1 = 0$.

Тогда значения остальных коэффициентов:

$$a = a_1 = y_0$$

$$b = b_1 = \frac{y_1 - y_0}{x_1 - x_0} - (x_1 - x_0) \frac{2c_1}{3} = \frac{y_1 - y_0}{x_1 - x_0} - 0 = \frac{y_1 - y_0}{x_1 - x_0},$$

$$d = d_1 = -\frac{c_1}{3(x_1 - x_0)} = 0.$$

Таким образом, полином будет иметь вид:

$$\psi(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x - x_0)$$

То есть сплайн будет представлять прямую, проходящую через две заданные точки.

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

На трех точках образуются два интервала аргумента, на каждой строится сплайн, задающийся полиномом с четырьмя коэффициентами. Общее количество коэффициентов, которые необходимо определить, равно 8, то есть необходимо 8 условий.

Первые четыре определяются по совпадению значений полинома сплайна со значениями функции в узлах, а именно:

$$\psi(x_0) = y_0, \psi(x_1) = y_1, \psi(x_2) = y_2.$$

Откуда получаем:

- по $\psi(x_{i-1}) = y_{i-1} = a_i$, имеем (условия 1-2):

$$a_1 = y_0,$$

$$a_2 = y_1;$$

- по $\psi(x_i) = y_i = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3$, имеем (условия 3-4):

$$a_1 + b_1(x_1 - x_0) + c_1(x_1 - x_0)^2 + d_1(x_1 - x_0)^3 = y_1$$

$$a_2 + b_2(x_2 - x_1) + c_2(x_2 - x_1)^2 + d_2(x_2 - x_1)^3 = y_2$$

Последние четыре получаем по совпадению первых и вторых производных во внутренних узлах, а также предположению, что на концах участка интерполирования вторая производная равна нулю:

- по $\psi_1'(x_1) = \psi_2'(x_1)$, имеем (условие 5):

$$b_1 + 2c_1(x_1 - x_0) + 3d_1(x_1 - x_0)^2 = b_2,$$

- по $\psi_1''(x_1) = \psi_2''(x_1)$, имеем (условие 6):

$$c_1 + 3d_1(x_1 - x_0) = c_2,$$

- по $\psi_1''(x_0) = 0$ и $\psi_2''(x_2) = 0$, имеем (условия 7-8):

$$c_1 + 3d_1(x_0 - x_0) = 0 \Rightarrow c_1 = 0$$

$$c_2 + 3d_2(x_2 - x_1) = 0 \Rightarrow d_2 = -\frac{c_2}{3(x_2 - x_1)}$$

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

Коэффициенты c_i определяются через прогоночные коэффициенты по формуле:

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

В случае если $c_1 = c_2$, формула имеет вид:

$$c_1 = 1 \cdot c_2 + 0.$$

Таким образом:

$$\xi_2 = 1, \eta_2 = 0$$

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

Коэффициенты c_i определяются через прогоночные коэффициенты по формуле:

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

В случае $i = N$:

$$c_{N-1} = \xi_N c_N + \eta_N.$$

Откуда:

$$c_N = \frac{c_{N-1} - \eta_N}{\xi_N}.$$

Из граничного условия получаем:

$$c_{N-1} = \frac{p - mC_N}{k}.$$

Откуда:

$$c_N = \frac{\frac{p - mC_N}{k} - \eta_N}{\xi_N}$$

$$c_N \xi_N = \frac{p - mC_N}{k} - \eta_N$$

$$c_N \xi_N k = p - mC_N - \eta_N k$$

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

Таким образом:

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