**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 многомерной интерполяции табличных функций.

**Студент** Алахов А.Г.

**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. Задаётся с помощью формулы y = x^2 в диапазоне [0..10] с шагом 1.
2. Значение аргумента x в первом интервале, например, при х=0.5 и в середине таблицы, например, при x= 5.5. Сравнить с точным значением.

# Код программы

Код программы представлен на листингах 1-2.

Листинг 1. functions.py

**from** math **import** fabs

**def** copy\_table**(**table**):**

new **=** **[]**

**for** i **in** table**:**

new**.**append**(**i**[:])**

**return** new

**def** neuton**(**table**,** x**,** n**):**

n **+=** 1

func\_table **=** copy\_table**(**table**)**

i **=** 0

**while** func\_table**[**i**][**0**]** **<** x **and** i **<** **len(**func\_table**):**

i **+=** 1

left\_part **=** right\_part **=** n **//** 2

**if** n **%** 2**:**

**if** **(**x **-** func\_table**[**i **-** 1**][**0**])** **-** **(**func\_table**[**i**][**0**]** **-** x**)** **<=** 0.000001**:**

left\_part **+=** 1

**else:**

right\_part **+=** 1

**if** i **+** right\_part **>** **len(**func\_table**):**

left\_part **+=** i **+** right\_part **-** **len(**func\_table**)**

start **=** **max(**i **-** left\_part**,** 0**)**

**for** j **in** **range(**1**,** n**):**

**for** i **in** **range(**start **+** n **-** 1**,** start **+** j **-** 1**,** **-**1**):**

func\_table**[**i**][**1**]** **=** **((**func\_table**[**i**][**1**]** **-** func\_table**[**i **-** 1**][**1**])** **/**

**(**func\_table**[**i**][**0**]** **-** func\_table**[**i **-** j**][**0**]))**

result **=** 0

**for** i **in** **range(**start**,** start **+** n**):**

mult **=** func\_table**[**i**][**1**]**

**for** j **in** **range(**start**,** i**):**

mult **\*=** **(**x **-** func\_table**[**j**][**0**])**

result **+=** mult

**return** result

**def** spline**(**table**,** x**):**

N **=** **len(**table**)** **-** 1

h **=** **[**0**,** table**[**1**][**0**]** **-** table**[**0**][**0**]]**

kci **=** **[**0**,** 0**,** 0**]**

etta **=** **[**0**,** 0**,** 0**]**

**for** i **in** **range(**2**,** N **+** 1**):**

h**.**append**(**table**[**i**][**0**]** **-** table**[**i **-** 1**][**0**])**

f **=** 3 **\*** **((**table**[**i**][**1**]** **-** table**[**i**-**1**][**1**])** **/** h**[**i**]** **-**

**(**table**[**i **-** 1**][**1**]** **-** table**[**i**-**2**][**1**])** **/** h**[**i**-**1**])**

kci**.**append**(-**h**[**i**]** **/** **(**h**[**i**-**1**]** **\*** kci**[**i**]** **+** 2**\*(**h**[**i**-**1**]** **+** h**[**i**])))**

etta**.**append**((**f **-** h**[**i**-**1**]** **\*** etta**[**i**])** **/**

**(**h**[**i**-**1**]** **\*** kci**[**i**]** **+** 2**\*(**h**[**i**-**1**]** **+** h**[**i**])))**

c **=** **[**0**,** 0**]**

**for** i **in** **range(**N **-** 1**,** 0**,** **-**1**):**

c**.**insert**(**0**,** kci**[**i**+**1**]** **\*** c**[**0**]** **+** etta**[**i**+**1**])**

c**.**insert**(**0**,** 0**)**

a **=** **[**0**]**

b **=** **[**0**]**

d **=** **[**0**]**

**for** i **in** **range(**1**,** N **+** 1**):**

a**.**append**(**table**[**i**-**1**][**1**])**

b**.**append**((**table**[**i**][**1**]** **-** table**[**i **-** 1**][**1**])** **/** h**[**i**]** **-** h**[**i**]** **\*** **(**c**[**i**+**1**]** **+** 2**\***c**[**i**])/**3**)**

d**.**append**((**c**[**i**+**1**]** **-** c**[**i**])** **/** **(**3**\***h**[**i**]))**

**for** i **in** **range(**N**):**

**if** table**[**i**][**0**]** **<=** x **and** table**[**i**+**1**][**0**]** **>** x**:**

pos **=** i **+** 1

**break**

**return** **(**a**[**pos**]** **+** b**[**pos**]** **\*** **(**x **-** table**[**pos**-**1**][**0**])** **+** c**[**pos**]** **\***

**(**x **-** table**[**pos**-**1**][**0**])\*\***2 **+** d**[**pos**]** **\*** **(**x **-** table**[**pos**-**1**][**0**])\*\***3**)**

Листинг 2. main.py

**from** functions **import** **\***

**def** main**():**

func\_table **=** **[]**

**for** i **in** **range(**11**):**

func\_table**.**append**([**i**,** i**\*\***2**])**

x **=** 0.5

**print(**'Заданная таблица:\n\

| X |Y(x)|'**)**

**for** i **in** func\_table**:**

**print(**'|{:3d}|{:4d}|'**.format(**i**[**0**],** i**[**1**]))**

**print(**'\nЗаданный аргумент: X ='**,** x**)**

**print(**'Результат интерполяции кубическим сплайном: {:.6f}'**.format(**spline**(**func\_table**,** x**)))**

**print(**'Точное значение: {:.2f}'**.format(**x**\*\***2**))**

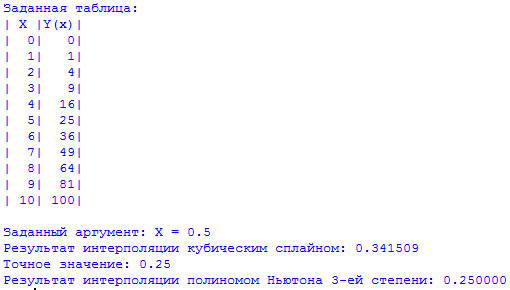
**print(**'Результат интерполяции полиномом Ньютона 3-ей степени: {:.6f}'**.format(**neuton**(**func\_table**,** x**,** 3**)))**

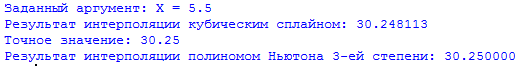
**if** \_\_name\_\_ **==** "\_\_main\_\_"**:**

main**()**

# Результаты работы

1. Значения y(x) при заданных x, сравнение результатов интерполяции кубическим сплайном и полиномом Ньютона 3-ей степени.





# Вопросы при защите лабораторной работы

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

с = 0, т.к. дано всего 2 точки

a = x0

b = (y1 – y0) / (x1 – x0)

d = 0

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

1-2) Условия совпадения значений 1-го многочлена и интерполируемой функции в 0 и 1 точках

3-4) Условия совпадения значений 2-го многочлена и интерполируемой функции в 1 и 2 точках

5-6) Равенство в точке 1 первой и второй производных, вычисляемых по коэффициентам на соседних участках

7-8) Условие равенства второй производной нулю в точках 0 и 2

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

Из условия: c1 = ξ \* c2 + η, получаем: ξ = 1, η = 0

4. Написать формулу для определения последнего коэффициента сплайна СN , 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.

Из условия: CN-1 = ξN \* CN + ηN, получаем: CN = (p – k \* ηN) / (k \* ξ + m)