

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Национальный исследовательский ядерный университет «МИФИ»  
(НИЯУ МИФИ)  
Институт Финансовых Технологий и Экономической Безопасности  
Кафедра Финансового Мониторинга

Лабораторная работа №3:  
По курсу «Численные методы»

Работу выполнил: студент группы С18-712:  
Проверил:

Кольца И. В.  
Саманчук В.Н.

Москва 2020

## Постановка задачи

Интерполировать таблично заданную функцию, используя полином Лагранжа четвертого порядка.

X	1	2	3	4	5	6	7	8	9	10
Y	5	6	8	10	12	13	12	10	8	10
X	11	12	13	14	15	16	17	18	19	20
Y	8	11	7	9	11	10	9	12	11	6

## Методика решения

Для решения поставленной задачи была написана программа на языке Python, в которой реализован полином Лагранжа для интерполяции таблично заданной функции.

## Теоретическая справка

### Интерполяционный многочлен Лагранжа

Пусть функция  $y = f(x)$  задана таблицей. Построим интерполяционный многочлен  $L_n(x)$ , степень которого не больше  $n$  и выполняются условия:  $L_n(x_i) = y_i, i = 0, 1, \dots, n$ . Будем искать  $L_n(x)$  в виде

$$L_n(x) = p_0(x)y_0 + p_1(x)y_1 + \dots + p_n(x)y_n = \sum_{i=0}^n p_i(x)y_i,$$

где  $p_i(x)$  — многочлен степени  $n$ ,  $p_i(x_j) = \delta_{ij}$  т. е.  $p_i(x)$  только в одной точке отличен от 0,  $i \neq j$ .

нуля при  $i = j$ , а в остальных точках он обращается в нуль. Следовательно, все эти точки являются для него корнями:

$$p_i(x) = c(x - x_0)(x - x_1)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_n);$$

при  $x = x_i$ ,

$$p_i(x_i) = c(x_i - x_0)(x_i - x_1)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n);$$

$$c = [(x_i - x_0)(x_i - x_1)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)]^{-1};$$

подставим  $c$  в формулу  $p_i(x)$ , получим:

$$p_i(x) = \frac{(x - x_0)(x - x_1)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_n)}{(x_i - x_0)(x_i - x_1)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)},$$
 отсюда

$$L_n(x) = \sum_{i=0}^n \frac{(x - x_0)(x - x_1)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_n)}{(x_i - x_0)(x_i - x_1)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)} y_i.$$

Это и есть интерполяционный многочлен Лагранжа. По исходной таблице формула позволяет весьма просто составить вид многочлена.

## Решение задачи

```
import numpy as np
import matplotlib.pyplot as plt

def lagrange(x, X, Y):
    L = 0
    for i in range(len(Y)):
        li = 1
        for j in range(len(X)):
            if (j == i):
                continue
            li *= (x - X[j]) / (X[i] - X[j])
        L += Y[i] * li
    return L
```

```

n = 4
X = np.arange(1, 21)
Y = np.array([5, 6, 8, 10, 12, 13, 12, 10, 8, 10, 8, 11, 7, 9, 11, 10, 9, 12, 11, 6])
x = np.linspace(1, 20, 77)

f = np.array([])

N = 4
i = 0

f = np.append(f, lagrange(x[: N], X[: N + 1], Y[: N + 1]))

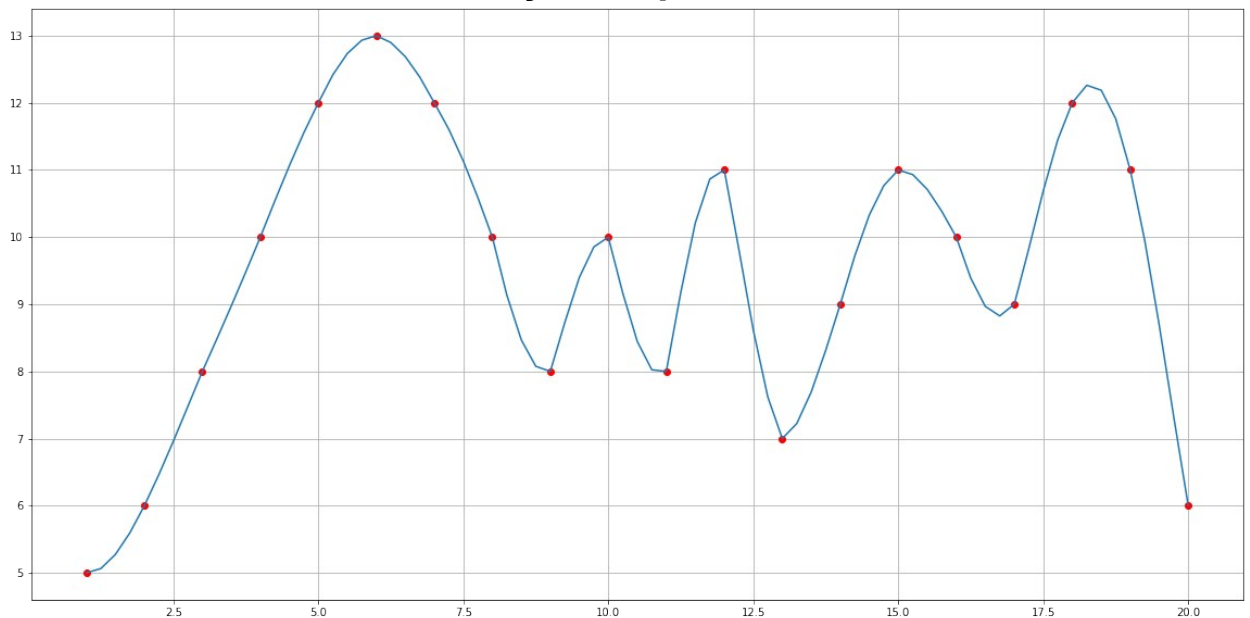
while (i < len(X) - n):
    f = np.append(f, lagrange(x[(i + 1) * N:(i + 2) * N], X[i:i + n + 1], Y[i:i + n + 1]))
    i += 1

f = np.append(f, lagrange(x[(i + 1) * N:], X[len(X) - n - 1:], Y[len(Y) - n - 1:]))

plt.figure(figsize=(20, 10))
plt.plot(X, Y, 'ro', x, f)
plt.grid(True)
plt.show()

```

### Результат работы



### Заключение

В работе требовалось интерполировать таблично заданную функцию с помощью полинома Лагранжа. Для решения задачи была написана программа на языке программирования Python.