Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы численного анализа

**ОТЧЁТ**

к лабораторной работе

на тему

Интерполяционные многочлены

Выполнил: студент группы 253502

Шишко Виктор Викторович

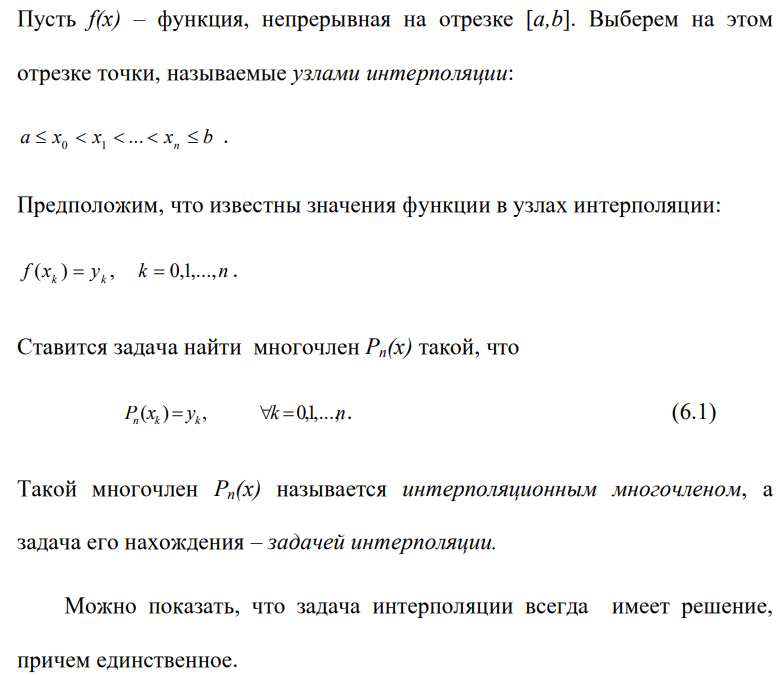
Проверил: Анисимов Владимир Яковлевич

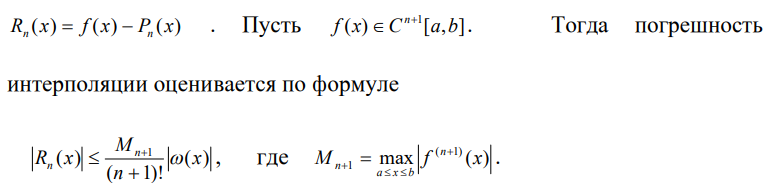
Минск 2023

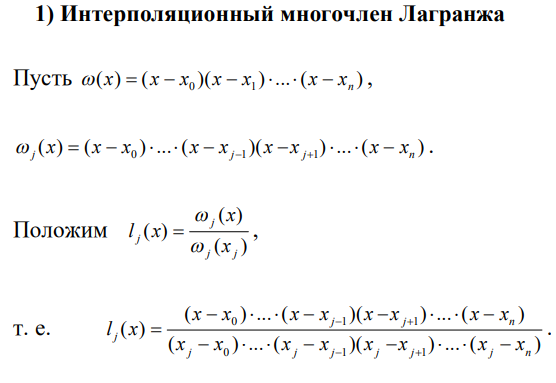
**Цель работы**

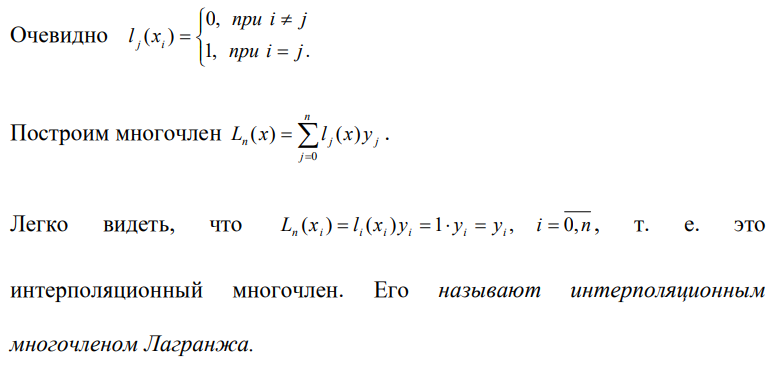
1. Изучить интерполяцию функций с помощью интерполяционных многочленов Лагранжа и Ньютона
2. Реализовать интерполяцию функций методами Лагранжа и Ньютона
3. Реализовать тестовые примеры

**Краткие теоретические сведения**

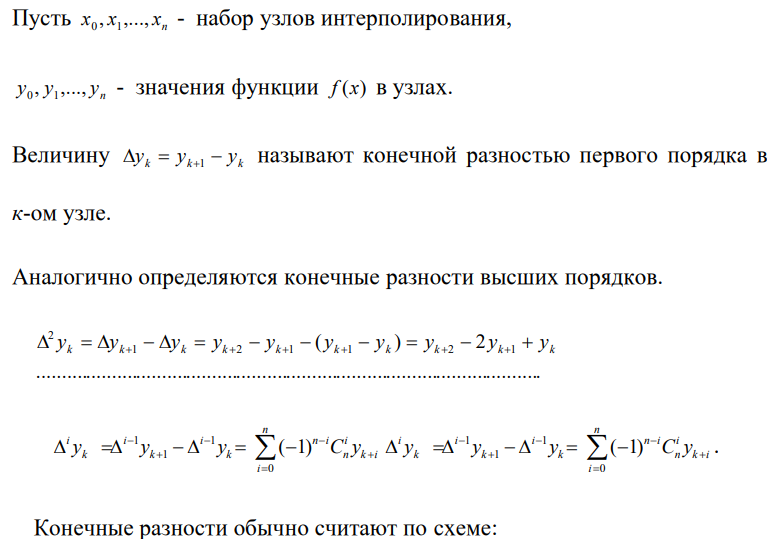
****

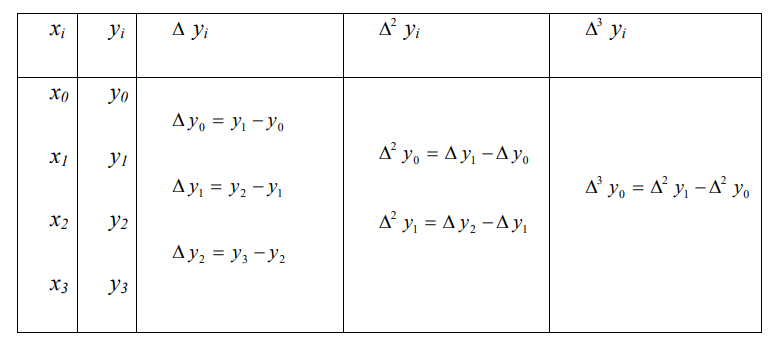
****

****

****

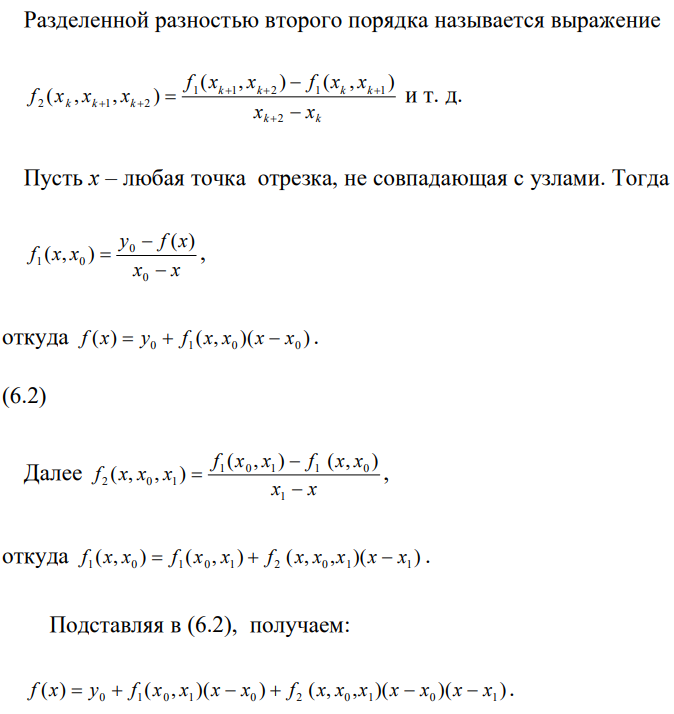
****

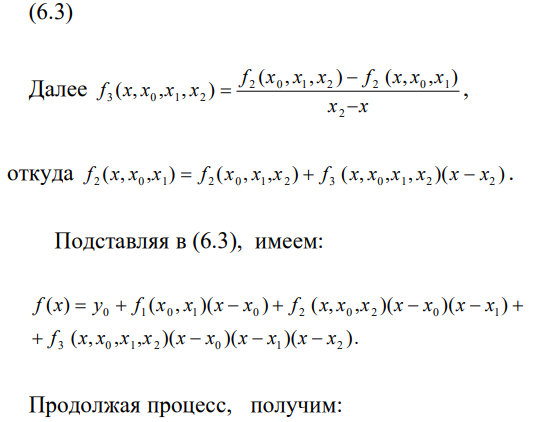
****

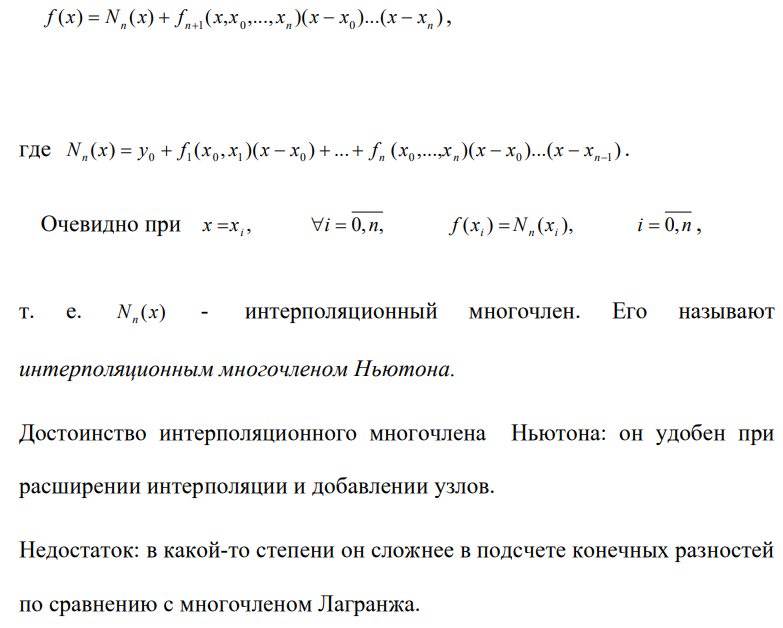
****

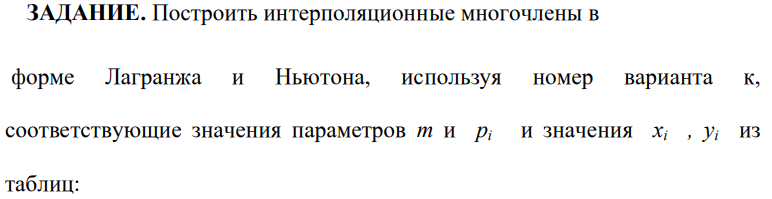
****

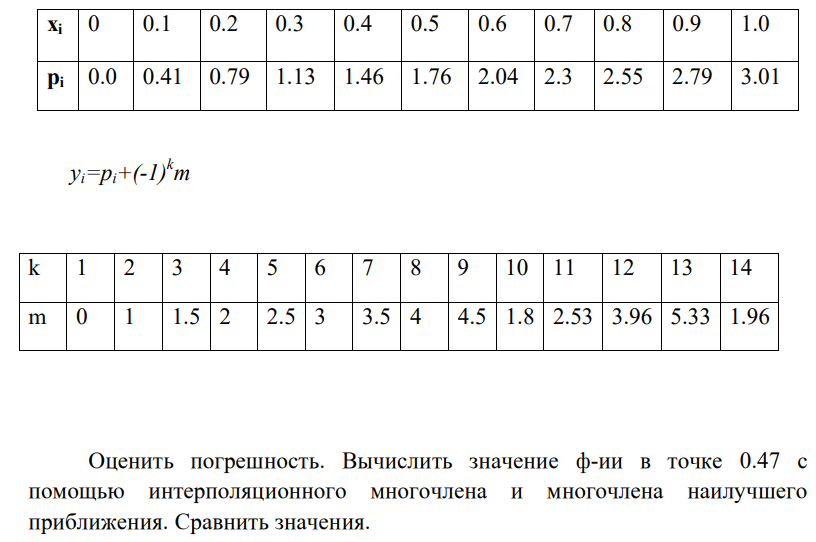
****

****

****

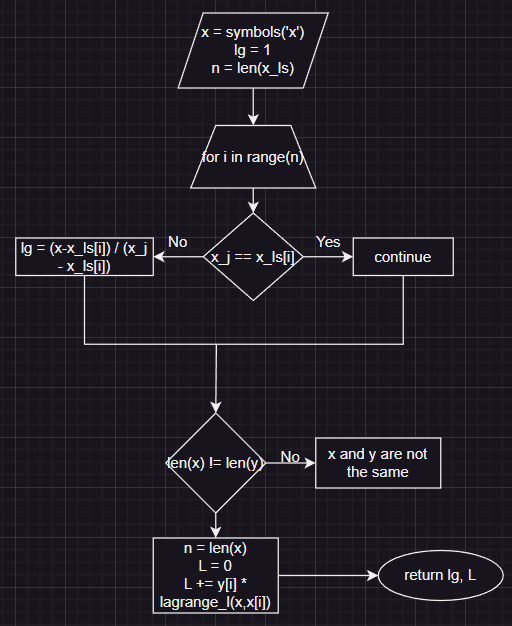
****

****

****

**Программная реализация**

1. Lagrange



def lagrange\_l(x\_ls: list, x\_j):

    x = symbols('x')

    lg = 1

    n = len(x\_ls)

    for i in range(n):

        if x\_j == x\_ls[i]:

            continue

        lg \*= (x - x\_ls[i]) / (x\_j - x\_ls[i])

    return lg

def lagrange\_poly(x, y):

    if len(x) != len(y):

        raise ValueError("x and y are not same length")

    n = len(x)

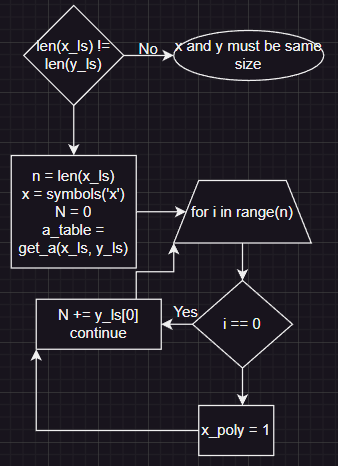
    L = 0

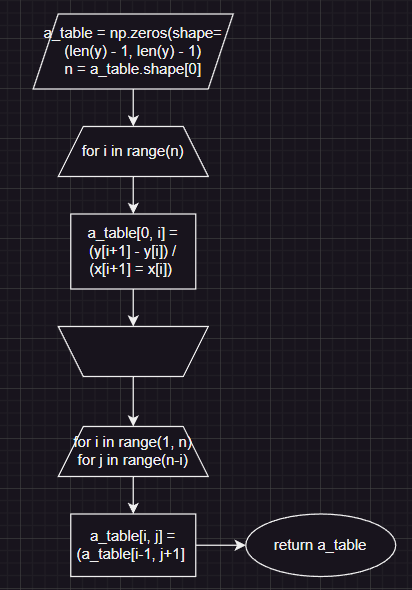
    for i in range(n):

        L += y[i] \* lagrange\_l(x, x[i])

    return L

1. Newton





def get\_a(x: list, y: list):

    a\_table = np.zeros(shape=(len(y) - 1, len(y) - 1))

    n = a\_table.shape[0]

    for i in range(n):

        a\_table[0, i] = (y[i + 1] - y[i]) / (x[i+1] - x[i])

    for i in range(1, n):

        for j in range(n - i):

            a\_table[i, j] = (a\_table[i - 1, j + 1] - a\_table[i - 1, j]) / (x[i+1+j] - x[j])

    return a\_table

def newton\_poly(x\_ls: list, y\_ls: list):

    if len(x\_ls) != len(y\_ls):

        raise TypeError("x and y must be same size")

    n = len(x\_ls)

    x = symbols('x')

    N = 0

    a\_table = get\_a(x\_ls, y\_ls)

    for i in range(n):

        if i == 0:

            N += y\_ls[0]

            continue

        x\_poly = 1

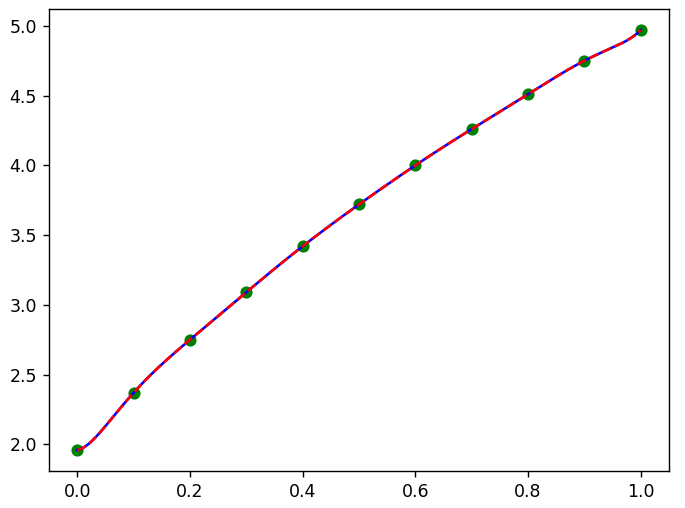
        for j in range(i):

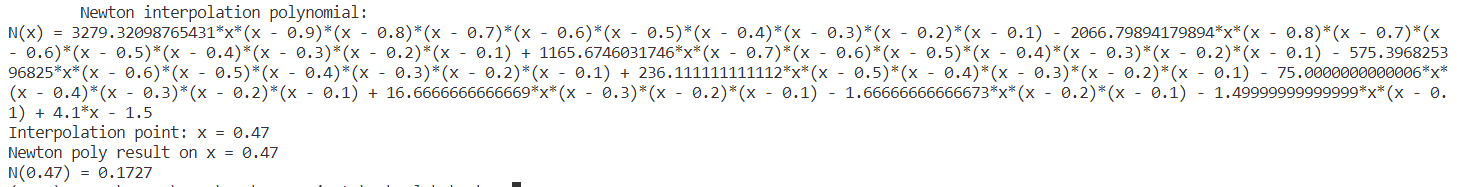
            x\_poly \*= (x-x\_ls[j])

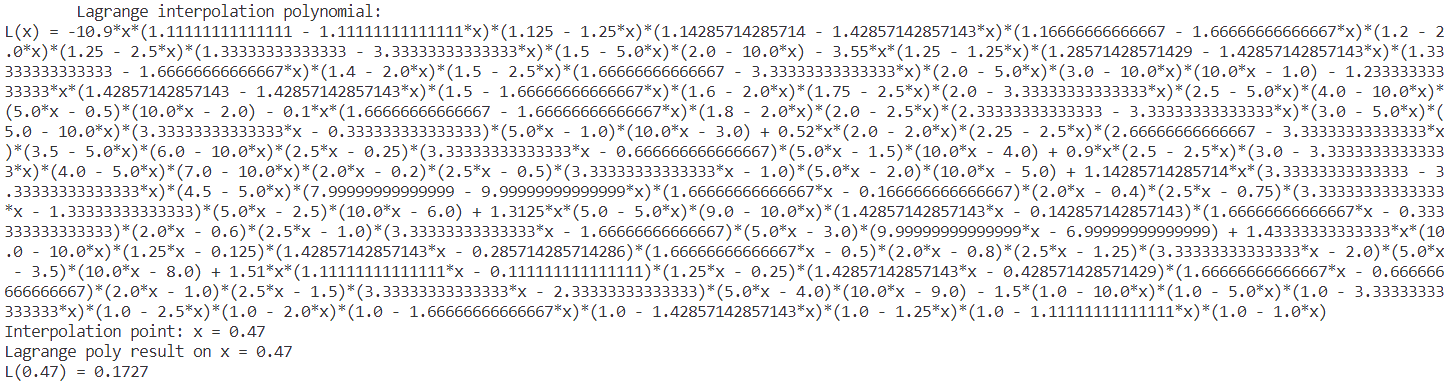
        N += a\_table[i-1, 0] \* x\_poly

    return N

**Результаты работы программной реализации**

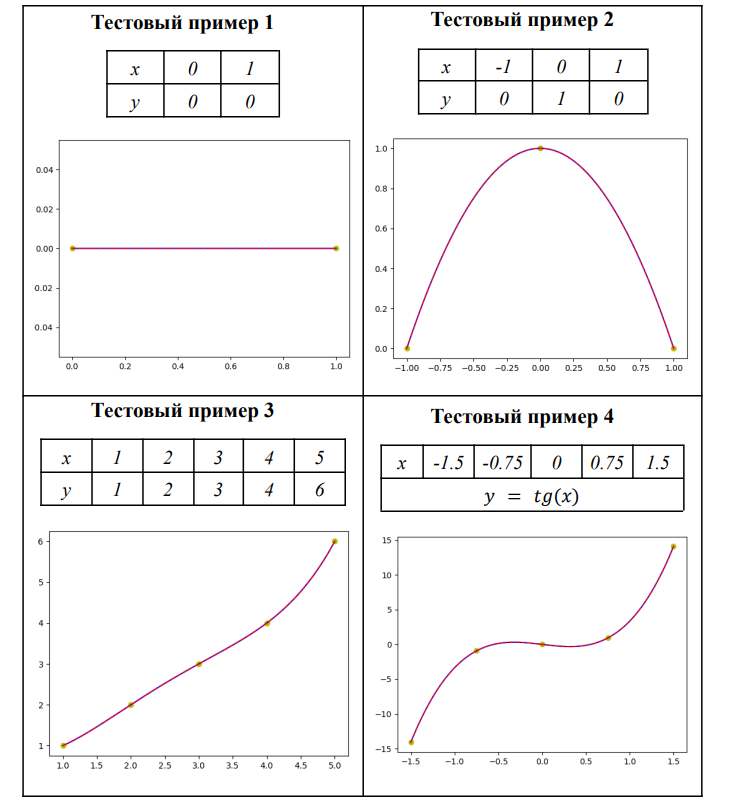






**Реализация тестовых примеров**

1. Построить интерполяционные многочлены в форме Лагранжа и Ньютона.



**Вывод**

Таким образом, в ходе выполнения лабораторной работы была освоена интерполяция функций с помощью интерполяционных многочленов Лагранжа и Ньютона. Составлена компьютерная программа, на тестовых примерах проверена правильность её работы, построены интерполяционные многочлены в форме Лагранжа и Ньютона, построен многочлен наилучшего приближения по методу наименьших квадратов, вычислено значение функции в точке согласно заданному варианту