Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «Вычислительная и прикладная математика»

**Отчет по лабораторной работе № 4**

Построение интерполяционных многочленов

по дисциплине: «Вычислительная математика»

Выполнил:  
студент группы 143  
Попов К.И.  
Проверил:   
доц. каф. ВПМ  
Крошилина С.В.

# Рязань 2024

# 1 Задание (Вариант 2)

Для функции f(x), заданной своими значениями в точках x0,...,xn, построить интерполяционный многочлен Ньютона с использованием разделенных разностей. Для вариантов с четными номерами использовать формулу интерполирования вперед, для вариантов с нечетными номерами - формулу интерполирования назад. Сравнить графики аппроксимируемой функции f(x) и интерполяционного многочлена внутри и за пределами отрезка интерполирования. Построить график погрешности интерполяции.

f(x) = Sin(π Cos(π x)), xmin=0, xmax=1, n=5.

# 2 Блок-схемы алгоритмов

## 2.1 Вычисление разделенных разностей

Схема вычисления разделенных разностей представлена на рисунке 1.

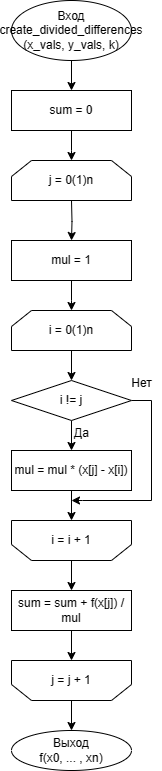


Рисунок – Схема вычисления разделенных разностей

## 2.2 Построение полинома Ньютона

Схема метода дихотомии представлена на рисунке 3.

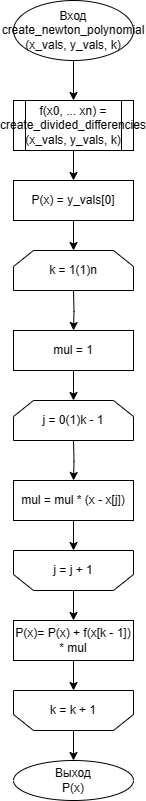


Рисунок – Схема метода построения полинома Ньютона

# 3 Листинг алгоритмов

Реализация алгоритмов приведена на языке Python.

## 3.1 Листинг метода вычисления разделенных разностей

def create\_divided\_differences(self, x\_vals, y\_values, k):

        return sum(y\_values[j] / reduce(lambda acc, i: acc \* (x\_vals[j] - x\_vals[i]) if i != j else acc, range(k + 1), 1) for j in range(k + 1))

## 3.2 Листинг метода построения полинома Ньютона

def create\_newton\_polynomial(self, x\_vals, y\_vals):

        divided\_differences = [self.create\_divided\_differences(x\_vals, y\_vals, i) for i in range(1, len(x\_vals))]

        def newton\_polynomial(x):

            return y\_vals[0] + sum(divided\_differences[k - 1] \* reduce(lambda acc, j: acc \* (x - x\_vals[j]), range(k), 1) for k in range(1, len(y\_vals)))

        return newton\_polynomial

# 4 Пользовательский интерфейс

На рисунке 3 показано главное пользовательское окно взаимодействия с программой, в котором пользователь может задать интервал интерполяции, а также имеет возможность вручную ввести точки внутри заданного интервала.

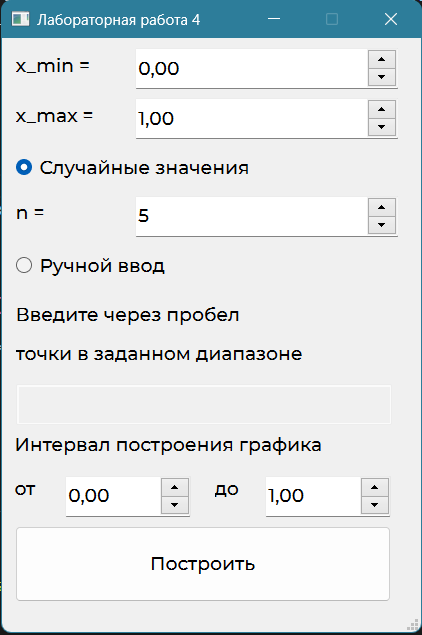


Рисунок – Главное окно

# 5 Результат работы программы

График при заданных на рисунке 3 значения представлен на рисунке 4.

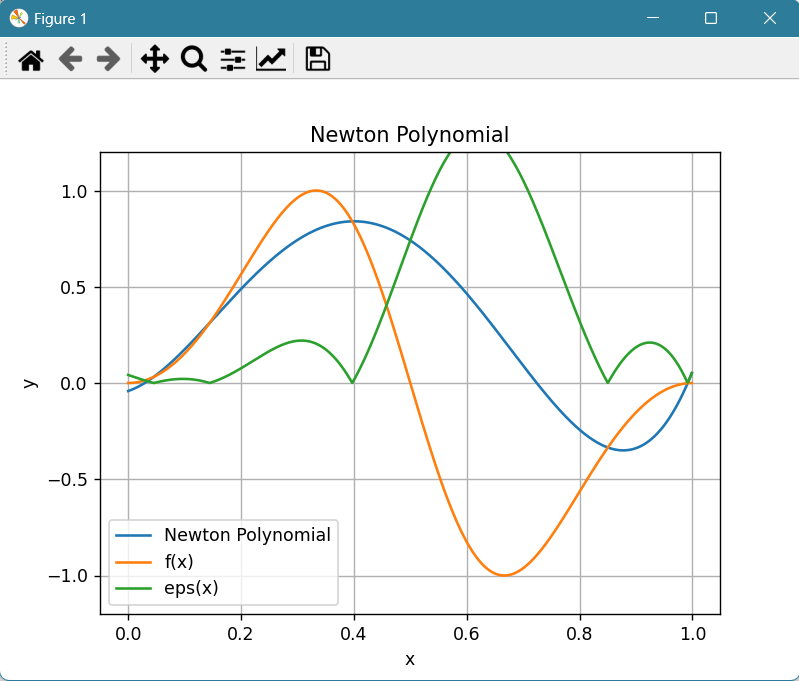


Рисунок – Результат работы программы

Синяя линия – график полученного полинома, красная – исходная функция. Зеленой линией отмечен график разности по модулю значений функции и полинома для заданного икса.