РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ им. ПАТРИСА ЛУМУБЫ

Факультет физико-математических и естественных наук

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

дисциплина: Вычислительные методы

Студент:

Мажитов Магомед Асхабович

Группа:

НКНбд-01-21

МОСКВА

2023 г.

Цель:

Написать программу для расчета полинома Лагранжа.

Теоретическая справка:

- 1. Построить равномерное разбиение отрезка [a, b] на N=10 частей точками $a=x_0,\,x_1,\,...,\,x_N=b.$
- 2. Рассчитать значения функции f(x) в узлах интерполяции: $y_0 = f(x_0), y_1 = f(x_1), ..., y_N = f(x_N)$.
- 3. Построить интерполяционный полином Лагранжа $L_N(x) = \sum_{i=0}^N y_i Q_i(x)$, где $Q_i(x) = \prod_{j=0, j \neq i}^N (x-x_j)/(x_i-x_j)$ согласно значениям из п.1, 2, также посчитать погрешность интерполяции в точке x_i .
- 4. Построить равномерное разбиение отрезка [a, b] из задания на M=3N частей точками $a=\overline{x}_0,\overline{x}_1,\ldots,\overline{x}_M=b$.
- 5. Посчитать значения исходной функции f(x) из задания и построенного в п.3 полинома Лагранжа $L_N(x)$ в точках $\overline{x_0}$, $\overline{x_1}$, ..., $\overline{x_M}$, полученных в п.4, также посчитать погрешность интерполяции в точке $\overline{x_i}$.
- 6. Подобрать такое значение N, при котором максимальная погрешность меньше 0.01

Код:

```
#include <iostream>
#include <cmath>
#include <cmath>
#include <iomanip>

using namespace std;

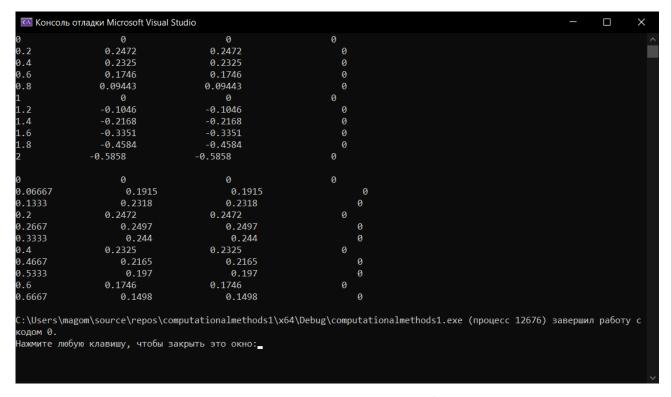
//sqrt(x) - x, a = 0, b = 2.

double* split(double a, double b, int n){//функция для разбиения отрезка на N
paвных отрезков
    double* s = new double [n];//массив для записи точек
    s[0] = a;//начальная точка
    s[n] = b;//конечная точка
    double step = (b - a)/n; //шаг
    for(int i = 1; i < n; i++){ //проходим по циклу и к предыдущей точке
    npuбавляем шаг
        s[i] = s[i-1] + step;
    }
    return s;//возвращаем массив из узлов интерполяций
}

double* function(double* x, int n){//функция для подсчета функции в узлах
интерполяции
    double* f = new double [n]; //массив для записи функций
    for(int i = 0; i <= n; i++){/проходим по циклу и считаем функции в узлах
интерполяции
        f[i] = sqrt(x[i]) - x[i];
}</pre>
```

```
double lagrangePolinomial(const double *x, const double *y, int n, double
xx) {//функция для подсчета полинома в точке
double* lagrangePolinomialAll(double* x, double* y, int n) {//функция для
       lag[i] = lagrangePolinomial(x, y, n, x[i]);
```

Работа программы:



При N=10 погрешность интерполяции ровна 0, таким образом подходит условию пункта N26.

Вывод:

В ходе работы я реализовал на языке С++ полином Лагранжа.