

**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 $a = x_0, x_1, \dots, x_N = b$.
2. Рассчитать значения функции $f(x)$ в узлах интерполяции: $y_0 = f(x_0), y_1 = f(x_1), \dots, y_N = f(x_N)$.
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_j .
4. Построить равномерное разбиение отрезка $[a, b]$ из задания на $M = 3N$ частей точками $a = \bar{x}_0, \bar{x}_1, \dots, \bar{x}_M = b$.
5. Посчитать значения исходной функции $f(x)$ из задания и построенного в п.3 полинома Лагранжа $L_N(x)$ в точках $\bar{x}_0, \bar{x}_1, \dots, \bar{x}_M$, полученных в п.4, также посчитать погрешность интерполяции в точке \bar{x}_j .
6. Подобрать такое значение N , при котором максимальная погрешность меньше 0,01

Код:

```
#include <iostream>
#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
равных 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
прибавляем шаг
        s[i] = s[i-1] + step;
    }
    return s; //возвращаем массив из узлов интерполяции
}

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];
    }
}
```

```

        return f; //возвращаем массив из функций
    }

double lagrangePolinomial(const double *x, const double *y, int n, double
xx){//функция для подсчета полинома в точке
    double l = 0.0, q;
    for(int i = 0; i <= n; i++){//проходим по циклу и считаем полином в каждой
точке
        q = 1.0;
        for(int j = 0; j <= n; j++){//проходим по циклу и если j не равно i
считаем q для в каждой точке
            if(j != i)
                q = q * (xx - x[j])/(x[i] - x[j]);
            }
        l = l + y[i] * q;
    }
    return l;//возвращаем полином в точке
}

double* lagrangePolinomialAll(double* x, double* y, int n){//функция для
подсчета полинома в каждой точке
    double* lag = new double [n]; //создаем массив для записи полинома в каждой
точке
    for(int i = 0; i <= n; i++){//проходим по циклу и в каждой точке считаем
значения полинома и записываем в массив
        lag[i] = lagrangePolinomial(x, y, n, x[i]);
    }
    return lag;//возвращаем массив
}

int main() {
    int n = 10, m = 3*n;
    double a = 0, b= 2;
    double *x = split(a, b, n);
    double *y = function(x, n);
    double *l = lagrangePolinomialAll(x, y, n);
    double *x1 = split(a, b, m);
    double *y1 = function(x1, m);
    double *l1 = lagrangePolinomialAll(x1, y1, m);
    print(l, n, x, y);
    cout << endl;
    print(l1, n, x1, y1);
    return 0;
}

```

Работа программы:

```
Консоль отладки Microsoft Visual Studio

0          0          0          0
0.2        0.2472    0.2472    0
0.4        0.2325    0.2325    0
0.6        0.1746    0.1746    0
0.8        0.09443   0.09443   0
1          0          0          0
1.2        -0.1046   -0.1046   0
1.4        -0.2168   -0.2168   0
1.6        -0.3351   -0.3351   0
1.8        -0.4584   -0.4584   0
2          -0.5858   -0.5858   0

0          0          0          0
0.06667    0.1915    0.1915    0
0.1333     0.2318    0.2318    0
0.2        0.2472    0.2472    0
0.2667     0.2497    0.2497    0
0.3333     0.244     0.244     0
0.4        0.2325    0.2325    0
0.4667     0.2165    0.2165    0
0.5333     0.197     0.197     0
0.6        0.1746    0.1746    0
0.6667     0.1498    0.1498    0

C:\Users\magom\source\repos\computationalmethods1\x64\Debug\computationalmethods1.exe (процесс 12676) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно: _
```

При $N=10$ погрешность интерполяции равна 0, таким образом подходит условию пункта №6.

Вывод:

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