Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ОТЧЕТ**

о выполнении лабораторной работы

по дисциплине Алгоритмы вычислительной математики

Выполнил: ст. гр. 29/1

Абдюков З.М.

Проверил: Ассистент каф. ВТ

Шиян В.И.

Тема:Интерполяционный полином Ньютона.

**Задание**

Построить для заданной функции интерполяционный полином Ньютона.

**Краткие пояснения**

Интерполяция – способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений. Нахождение приближенной функции называется интерполяцией, а точки  
 – узлами интерполяции.

Если узлы интерполяции равноотстоящие по величине, так что  
, где – шаг интерполяции, т.е. =, то интерполяционный многочлен можно записать в форме, предложенной Ньютоном. Интерполяционные полиномы Ньютона удобно использовать, если точка интерполирования находится в начале таблицы – первая интерполяционная формула Ньютона или конце таблицы – вторая формула.

Первая интерполяционная формула Ньютона.

Интерполирующий полином ищется в виде:

Построение многочлена сводится к определению коэффициентов . При записи коэффициентов пользуются конечными разностями. Конечные разности первого порядка запишутся в виде:

∆∆,

где – значения функции при соответствующих значениях . Конечные разности второго порядка:

∆2∆2,

Конечные разности высших порядков найдутся аналогично:

∆k,

Коэффициенты находятся из условия . Находим , полагая ,

,

Далее подставляя значения , получим:

# 

Общая формула для нахождения всех коэффициентов имеет вид

,

где

В результате (5) примет вид

+…

+. (6)

Данный многочлен называют первым полиномом Ньютона.

Вторая интерполяционная формула Ньютона.

Для нахождения значений функции в конце интервала интерполирования интерполяционный полином запишется в виде

Коэффициенты находятся из условия .

Подставляя в (7) найдем

.

Для :

Для :

.

Формула для нахождения всех коэффициентов запишется как:

Подставив выражения для определения коэффициентов в формулу (7), получим вторую интерполяционную формулу Ньютона:

+…

+. (8)

В своем коде я использую переменную q как:  
 => . Почему так, , следовательно

, где

аналогично для . Получим формулу Ньютону следующую:

На рисунках иллюстрация работы программы для функций и при для синуса и для косинуса. Программа сначала выводит значение точек, затем значений функции в этих точках, после выводится таблица вычисленных значений Т.к. полином представлен в виде (6) и (8), то можно задать вручную для построенного полинома либо присвоить автоматически приближенное значений , после чего вычисляется значения первой и второй формулы интерполирующего полинома Ньютона в заданной точке и выводятся на экран.

**Листинг программы:**

#include <iostream>

#include <iomanip>

#include <vector>

#include <fstream>

#include <cmath>

#define PI 3.14159265358979323846

using namespace std;

void read\_dataFile(vector <double>& , vector<double>& );

void Newton(vector <double> , vector <double>);

void Input\_Decidee();

double function(double);

void printVector(vector<vector<double>>, int );

int main()

{

setlocale(LC\_ALL, "russian");

cout << " \t\t\tИнтерполяционный полином Ньютона\n";

for (int i = 0; i < 80; i++)

cout << "-";

vector <double>x, y;

cout << "\n 1. Считать данные с файла.\n"

" 2. Ввести данные в ручную.\n";

int chose = 0;

cin >> chose;

switch (chose)

{

case 1:

cout << " В файле должно быть следующие расположение: X Y[X]\n";

read\_dataFile(x, y);

Newton(x, y);

break;

case 2:

Input\_Decidee();

break;

default:

break;

}

}

void read\_dataFile(vector <double>& x, vector<double>& y)

{

ifstream file("data.txt");

if (!file.is\_open())

{

cout << "Файла не существует.";

return;

}

double temp = 0;

while (!file.eof())

{

file >> temp;

x.push\_back(temp);

file >> temp;

y.push\_back(temp);

}

file.close();

}

void Newton(vector <double> x, vector <double>y)

{

vector<vector<double>> dy;

double h=0; //шаг

h = (x[1] - x[0]);

for (unsigned i = 0; i < x.size(); ++i)

cout << "x[" << i << "] = " << x[i] << "\t" << "y[" << x[i] << "] = " << y[i] << endl;

int maxIndex = x.size() - 1;

dy.resize(maxIndex);

for (int i = 0; i < maxIndex; i++)

{

dy[i].resize(maxIndex);

for (int j = 0; j < maxIndex; j++)

dy[i][j] = 0;

}

//Первые конечные разности

for (int i = 0; i < maxIndex; i++)

dy[i][0] = y[i + 1] - y[i];

//Вторые

for (int j = 1; j < maxIndex; j++)

{

for (int i = 0; i < maxIndex - j; i++) // потому что идем лестницей

dy[i][j] = dy[i + 1][j - 1] - dy[i][j - 1];

}

cout << endl;

printVector(dy,maxIndex);

double x0 = PI / 9;

double mult = 1;

double sum = y[0];

double q = (x0 - x[0]) / h;

int factorial = 1;

for (int i = 0; i < maxIndex; i++)

{

mult \*= (q - i);

factorial \*= i + 1;

sum += mult \* dy[0][i] / (factorial);

}

cout << "\n Первая формула Ньютона: " << setprecision(5) << sum << endl;

q = (x0 - x[maxIndex]) / h;

sum = y[maxIndex];

mult = 1;

factorial = 1;

for (int i = 0; i < maxIndex; i++)

{

mult \*= (q + i);

factorial \*= i+1;

sum += mult \* dy[maxIndex - i - 1][i] / (factorial); // по мнимой диагонали

}

cout << "\n Вторая формула Ньютона: " << setprecision(5) << sum << endl;

}

void Input\_Decidee()

{

vector<double> x, y;

cout << " Укажите количество точек: ";

int n; cin >> n;

for (int i = 0; i < n; i++)

{

double temp = 0;

cin >> temp;

x.push\_back(temp);

y.push\_back(function(temp));

}

system("cls");

cout << endl;

Newton(x, y);

}

double function(double x)

{

return sin(x)\* x;

}

void printVector(vector<vector<double>> V,int maxIndex)

{

for (int i = 0; i < maxIndex; i++)

{

for (int j = 0; j < maxIndex; j++)

cout << V[i][j] << setprecision(4) << " ";

cout << endl;

}

}

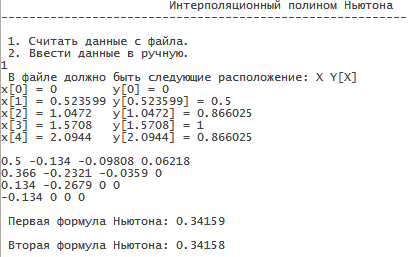


Рисунок 1 – Результаты функции sin(x) в точке



Рисунок 2 – Сравнение

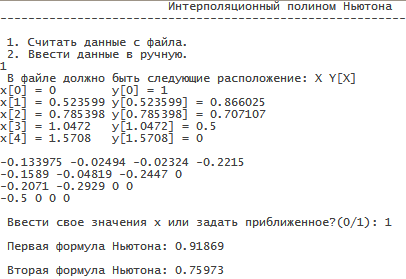


Рисунок 3 – Результаты функции cos(x) в точке

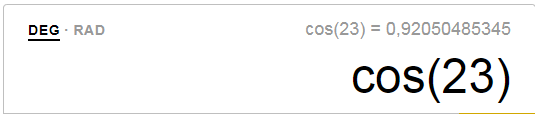


Рисунок 4 – Сравнение

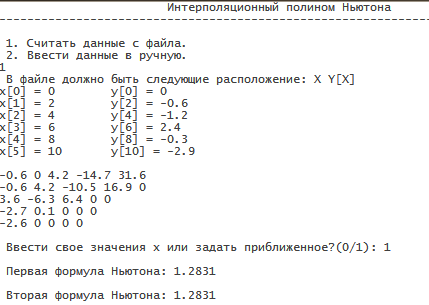


Рисунок 5 – Результаты функции cos(x) в точке



Рисунок 6 – Результат

**Вывод**

В ходе работы изучен: интерполирующий полином Ньютона для заданных функций.