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
$x\_{0}, x\_{1}, x\_{2}, …, x\_{n}$ – узлами интерполяции.

Если узлы интерполяции равноотстоящие по величине, так что
$x\_{i+1}–x=h=const$, где $h$ – шаг интерполяции, т.е. $x\_{i}$=$x\_{0}+nh$, 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:

$$P\_{n}\left(x\right)=a\_{0}+a\_{1}\left(x-x\_{0}\right)+…+a\_{n}\left(x-x\_{0}\right)..\left(x-x\_{n-1}\right) (5)$$

Построение многочлена сводится к определению коэффициентов $a\_{i}$. При записи коэффициентов пользуются конечными разностями. Конечные разности первого порядка запишутся в виде:

∆$y\_{0}=y\_{1}-y\_{0}… $∆$y\_{n-1}=y\_{n}-y\_{n-1}$,

где $y\_{i}$ – значения функции при соответствующих значениях $x\_{i}$ . Конечные разности второго порядка:

∆2$y\_{0}=∆y\_{1}-∆y\_{0}… $∆2$y\_{n-2}=∆y\_{n-1}-∆y\_{n-2}$,

Конечные разности высших порядков найдутся аналогично:

∆k$y\_{0}=∆^{k-1}y\_{1}-∆^{k-1}y\_{0}… ∆^{k}y\_{n-2}=∆^{k-1}y\_{n-1}-∆^{k-1}y\_{n-2}$,

Коэффициенты $a\_{0},a\_{1}…a\_{n}$ находятся из условия $P\_{n}\left(x\_{i}\right)=y\_{i}$. Находим $a\_{0}$, полагая $x=x\_{0}$,

$a\_{0}=P\left(x\_{0}\right)=y\_{0}$,

Далее подставляя значения $x=x\_{1}$, получим:

#

Общая формула для нахождения всех коэффициентов имеет вид

$a\_{i}=\frac{∆^{i}y\_{0}}{i!h^{i}}$,

где $i=1…n.$

В результате (5) примет вид

$P\_{n}\left(x\right)=y\_{0}+\frac{∆y\_{0}}{1!h}\left(x-x\_{0}\right)+\frac{∆^{2}y\_{0}}{2!h^{2}}\left(x-x\_{0}\right)\left(x-x\_{1}\right)$+…

+$\frac{∆^{n}y\_{0}}{n!h^{n}}\left(x-x\_{0}\right)…\left(x-x\_{n-1}\right)$. (6)

Данный многочлен называют первым полиномом Ньютона.

Вторая интерполяционная формула Ньютона.

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

$P\_{n}\left(x\right)=a\_{0}+a\_{1}\left(x-т\right)+…+a\_{n}\left(x-x\_{т}\right)..\left(x-x\_{1}\right) $ $(7)$

Коэффициенты $a\_{0},a\_{1}…a\_{n}$ находятся из условия $P\_{n}\left(x\_{i}\right)=y\_{i}$.

Подставляя в (7) $x = x\_{n},$ найдем

$P\_{n}\left(x\_{n}\right)=y\_{n}=a\_{0}$.

Для $x=x\_{n-1}$:

$$P\_{n}\left(x\_{n-1}\right)=y\_{n-1}=y\_{n}+a\_{1}\left(x\_{n-1}-x\_{n}\right), a\_{1}=\frac{y\_{n}-y\_{n-1}}{h}=\frac{∆y\_{n-1}}{h}.$$

Для $x=x\_{n-2}$:

$$P\_{n}\left(2\right)=y\_{n-2}=y\_{n}+\frac{∆y\_{n-1}}{h}\left(x\_{n-2}-x\_{n}\right)+a\_{2}\left(x\_{n-2}-x\_{n}\right)\left(x\_{n-2}-x\_{n-1}\right)=y\_{n}+\frac{∆y\_{n-1}}{h}\left(-2h\right)+a\_{2}\*2h^{2}=y\_{n}-2 ∆y\_{n-1}+a\_{2}2h^{2};$$

$a\_{2}=\frac{∆^{2}y\_{n-2}}{2!h^{2}}$.

Формула для нахождения всех коэффициентов запишется как:

$$a\_{i}=\frac{∆^{i}y\_{n-i}}{i!h^{i}}$$

Подставив выражения для определения коэффициентов $a\_{i} $в формулу (7), получим вторую интерполяционную формулу Ньютона:

$P\_{n}\left(x\right)=y\_{n}+\frac{∆y\_{n-1}}{1!h}\left(x-x\_{n}\right)+\frac{∆^{2}y\_{n-2}}{2!h^{2}}\left(x-x\_{n}\right)\left(x-x\_{n-1}\right)$+…

+$\frac{∆^{n}y\_{0}}{n!h^{n}}\left(x-x\_{n}\right)…\left(x-x\_{1}\right)$. (8)

В своем коде я использую переменную q как:
$q=\frac{x-x\_{n}}{h}$ => $x-x\_{n}=qh=>x-x\_{n-1}=\left(q+1\right)\*h$. Почему так, $x\_{i}=x\_{0}+ih$, следовательно

$x-\left(x\_{n}-h\right)=x-x\_{n}+h=qh+h=\left(q+1\right)h$ , где $q=x-x\_{n}$

аналогично для $i $. Получим формулу Ньютону следующую:

$$P\_{n}\left(x\right)=y\_{n}=\frac{q ∆y\_{n-1}}{1!}+\frac{q\left(q+1\right)∆^{2}y\_{n-2}}{2!}+…+\frac{q\left(q+1\right)..\left(q+n-1\right)∆^{n}y\_{0}}{n!}.$$

На рисунках иллюстрация работы программы для функций $sin(x)$ и $cos(x) $при $x = 20$ для синуса и$ x = 23$ для косинуса. Программа сначала выводит значение точек, затем значений функции в этих точках, после выводится таблица вычисленных значений $∆^{i}y .$Т.к. полином представлен в виде (6) и (8), то $x$ можно задать вручную для построенного полинома либо присвоить автоматически приближенное значений $\frac{π}{9}$ , 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 $\frac{pi}{9}$



Рисунок 2 – Сравнение



Рисунок 3 – Результаты функции cos(x) в точке $\frac{pi}{9}$



Рисунок 4 – Сравнение



Рисунок 5 – Результаты функции cos(x)$\sqrt{x}$ в точке $\frac{pi}{9}$



Рисунок 6 – Результат

**Вывод**

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