МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Физико-технический факультет

Кафедра Вычислительной техники и электроники (ВТиЭ)

Выполнил студент 585 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.М. Губченко

Проверил: к.ф.-м.н,, доцент каф. ВТиЭ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.И. Иордан

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2019

1. **Интерполяция с равноотстоящими узлами**
   1. **Теория**

Если отрезок [a, b] большой и требуется высокая точность аппроксимации функции, то одним интерполяционным многочленом приемлемой степени даже с оптимальным распределением узлов обычно не удаётся обеспечить заданную точность интерполяции на отрезке. В таком случае часто пользуются таблицей значений функции в узлах, расположенных с постоянным шагом, число которых может быть достаточно большим. Основная формула для подсчёта интерполяционного многочлена:

где,

* 1. **Описание алгоритма**

1. Ввод a,b –границы отрезка

n – число отрезков

xg – тестовая точка

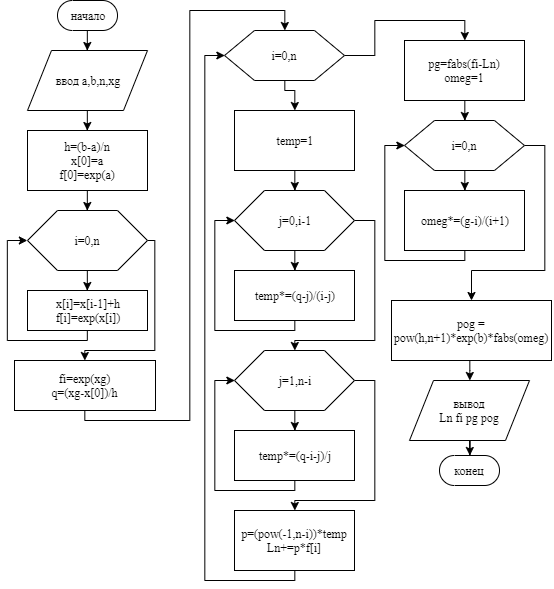
1. h=(b-a)/h
2. x[0] = a, f[0]=
3. i=1
4. x[i]=x[i-1]+h
5. f[i]=
6. i=i+1
7. если i<=n переход к пункту 5
8. fi =
9. q=(xg-x[0])/h – функция интерполяции
10. i=0
11. temp=1
12. j=0
13. temp = temp\*(q-j)/(i-j)
14. j=j+1
15. если j<i переход к пункту 14
16. j=1
17. temp=temp\*(q-i-j)/j
18. j=j+1
19. если j<=(n-i) переход к пункту 18
20. p =
21. Ln=Ln+p\*f[i]
22. i=i+1
23. если i<=n переход к пункту 12
24. pg =|fi-Ln|
25. omeg = 1
26. i=0
27. omeg = omeg \* (q-i) / (i+1)
28. i=i+1
29. если i<=n переход к пункту 28
30. pog = \*|
31. Вывод Ln - Интерполяционный многочлен Лагранжа

fi - функция f(i)

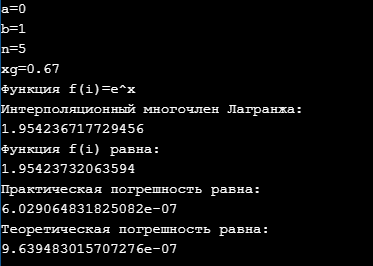
pg - практическая погрешность

pog - теоретическая погрешность

* 1. **Блок-схема**

****

* 1. **Пример выполнения**



* 1. **Список использованной литературы**

В.И. Ракитин, В.Е. Первушин – Практическое руководство по методам вычислений с приложением программ для персональных компьютеров

Е.А. Волков – Численные методы

Н.Н. - Численные методы

**Приложение**

**Код программы**

#include <stdio.h>

#include <iostream>

#include <conio.h>

#include <math.h>

#include <iomanip>

using namespace std;

double fun(double \*x, double \*f, int n)

{

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

{

f[i]=exp(x[i]);

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

int n, i, j;

double a, b;

//ввод отрезка

cout << "a=";

cin >> a;

cout << "b=";

cin >> b;

//ввод n

cout << "n=";

cin >> n;

double \*x = new double[n+1];

double \*f = new double[n+1];

//шаг

double h;

h = (b-a)/n;

x[0] = a;

for(i=1; i <= n; i++)

{

x[i] = x[i-1] + h;

}

fun(x, f, n); //составление функции

double xg;//тестовая точка

cout << "xg=";

cin >> xg;

double Ln=0;//ЛАГРАНЖ

double pg; //погрешность

double q; //ку

double p; //ПЭ

double fi; //значение функции в тестовой точке

double temp; //temp

fi=exp(xg);

cout<<"Функция f(i)=e^x";

//начинаем считать многочлен Лагранджа

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

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

{

temp = 1;

for(j=0; j<i; j++)

{

temp\*=(q-j)/(i-j);

}

for(j=1; j<=(n-i); j++)

{

temp\*=(q-i-j)/j;

}

p =pow(-1,n-i) \* temp;

Ln+=p\*f[i];

}

pg=fabs(fi-Ln); //Практическая погрешность

double omeg = 1;

double pog;

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

{

omeg\*=(q-i)/(i+1);

}

pog =pow(h,n+1) \* exp(b) \*fabs(omeg);

cout<<"**\n**Интерполяционный многочлен Лагранжа:**\n**" << setprecision(16) <<Ln;

cout<<"**\n**Функция f(i) равна:**\n**" << setprecision(16) <<fi;

cout<<"**\n**Практическая погрешность равна:**\n**" <<pg;

cout<<"**\n**Теоретическая погрешность равна:**\n**" <<pog;

delete [] x;

delete [] f;

return 0;

}