|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
| МИНОБРНАУКИ РОССИИ | | | |
| 1. Федеральное государственное бюджетное образовательное учреждение 2. высшего профессионального образования 3. **"Московский технологический университет"** 4. МИРЭА | | | |
| Институт Кибернетики | | |  |
| Кафедра программного обеспечения систем радиоэлектронной аппаратуры при АО «Концерн радиостроения «ВЕГА» | | |  |

|  |  |
| --- | --- |
| **Отчет по лабораторной работе №4** | |
| **по дисциплине** | |
| **«** Численные методы **»** | |
| **Вариант 7** | |
| Студент 3-го курса  группы КМБО-2-16 | Миронов Д.А. |
| Преподаватель | Даева С.Г. |
| Рецензент |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| Работа представлена к защите | «\_\_»\_\_\_\_\_\_\_2018 г. |  |
|  |  |  |
| «Допущен к защите» | «\_\_»\_\_\_\_\_\_\_2018 г. |  |

Москва 2019

Содержание

Задание 3

Теоретическая часть 4

Практическая часть 7

Приложения 9

# Задание

**Задание 1**

Используя первую или вторую формулы Ньютона, вычислить значения

функции, заданной таблично, для заданных значений аргумента.

|  |  |
| --- | --- |
| X | Y |
| 0.101 | 1.26183 |
| 0.106 | 1.27644 |
| 0.111 | 1.29122 |
| 0.116 | 1.30617 |
| 0.121 | 1.3213 |
| 0.126 | 1.3366 |
| 0.131 | 1.35207 |
| 0.136 | 1.36773 |
| 0.141 | 1.38357 |
| 0.146 | 1.39959 |
| 0.151 | 1.41579 |
| 0.156 | 1.42683 |
| 0.161 | 1.43356 |

х=0.1026+0.002\*n n=1,2,....30

**Задание 2**

Используя интерполяционные формулы Гаусса, Стирлинга и Бесселя, вы-

числить приближенные значения функции Y=F(X) при заданных значениях аргумента, если

исходные значения функции Y=F(X) представлены таблицей.:

|  |  |
| --- | --- |
| X | Y |
| 1.5 | 15.132 |
| 1.55 | 17.422 |
| 1.6 | 20.393 |
| 1.65 | 23.994 |
| 1.7 | 28.16 |
| 1.75 | 32.812 |
| 1.8 | 37.857 |
| 1.85 | 43.189 |
| 1.9 | 48.689 |
| 1.95 | 54.225 |
| 2 | 59.158 |
| 2.05 | 64.817 |
| 2.1 | 69.55 |
| 2.15 | 74.782 |
| 2.2 | 79.548 |

Значения аргумента для интерполяционных формул:

Гаусса: х=1.60+0.006 \* N

Стирлинга: x=1.725+0.002 \* N

Бесселя: x=1.83+0.003 \* N

**Задание 3**

Вычислить значения функции при заданных значениях аргумента, используя

интерполяционную формулу Ньютона для неравноотстоящих узлов и многочлен Лагранжа.

|  |  |
| --- | --- |
| X | Y |
| 0.593 | 0.53205 |
| 0.598 | 0.535625 |
| 0.605 | 0.540598 |
| 0.613 | 0.546235 |
| 0.619 | 0.550431 |
| 0.627 | 0.555983 |
| 0.632 | 0.559428 |
| 0.64 | 0.568738 |
| 0.65 | 0.575298 |

Значения аргумента для интерполяционных формул:

Ньютона с неравноотстоящими узлами: 0.608 + 0.025 \* (от 3 до 2 N)

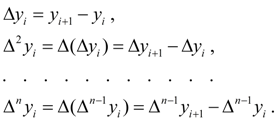
Многочлена Лагранжа: 0.63 – 0.025 \* (от N до 3 N)

## Теоретическая часть

Первая интерполяционная формула Ньютона:

Пусть в равноотстоящих точках http://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn001.png, где h – шаг интерполяции, заданы значения http://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn002.png для функцииhttp://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn003.png. Требуется подобрать полином http://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn004.pngстепени не выше n , удовлетворяющий условиям

http://www.simumath.net/library/materials/Interpol_introduction/images/Eqn011.png.  
Введем конечные разности для последовательности значений http://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn005.png :

          (2)

Условия эквивалентны равенствам:

http://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn007.png

при http://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn008.png

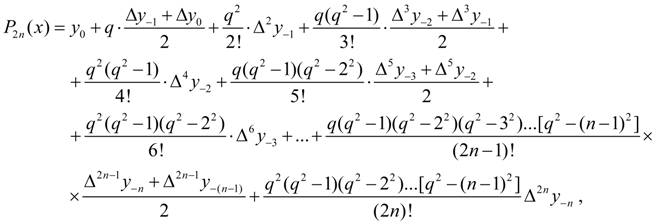
Окончательно получим первую интерполяционную формулу Ньютона:

http://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn009.png        (3)

где  http://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn010.png  – число шагов интерполяции от начальной точки http://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn011.png до точки х .  
Формулу (3) целесообразно использовать для интерполяции функции http://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn012.png в окрестности начальной точки http://www.simumath.net/library/materials/Interpol_1formula_Newton/images/Eqn013.png, где  q по абсолютной величине мало.

Интерполяционная формула Гаусса:

Интерполяционная формула Стирлинга:



Интерполяционная формула Бесселя:

где, как и раньше, http://www.simumath.net/library/materials/Interpol_formula_Stirling/images/Eqn002.png.

Интерполяционная формула Ньютона для неравноотстоящих значений:

Где

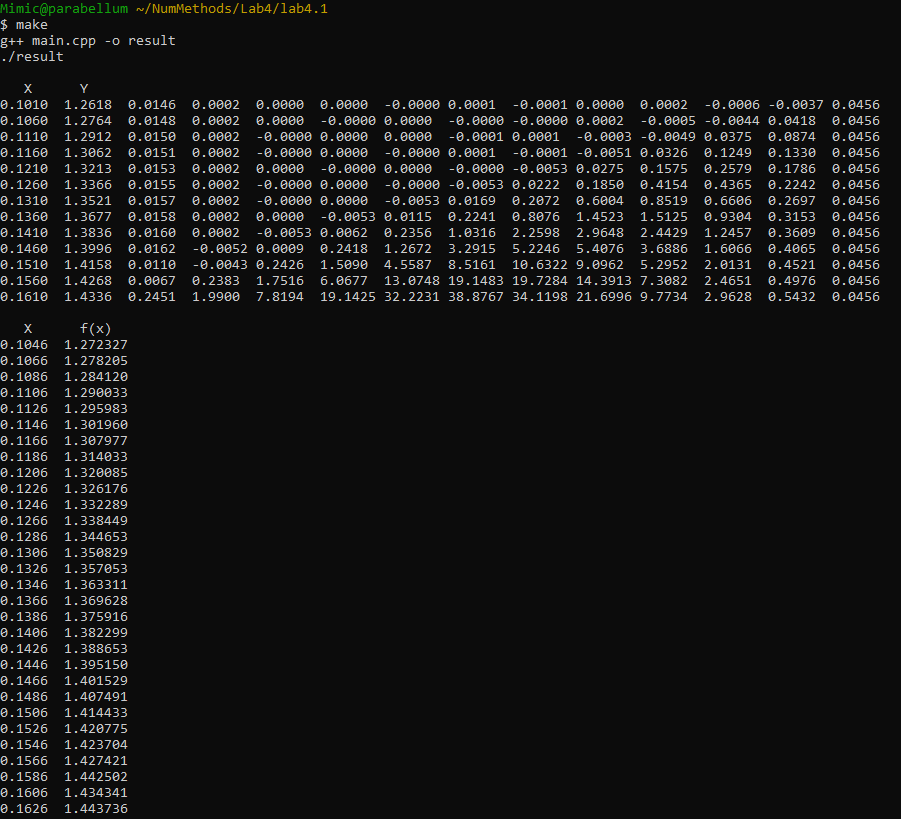
Интерполяционный многочлен Лагранжа:

**Практическая часть**

Для решения первого задания была разработана программа lab4.1

Программа составляет таблицу разностей, которую использует для вычислений. Далее, в цикле выводятся значения функции для каждого из аргументов соответственно.

Результат работы программы:

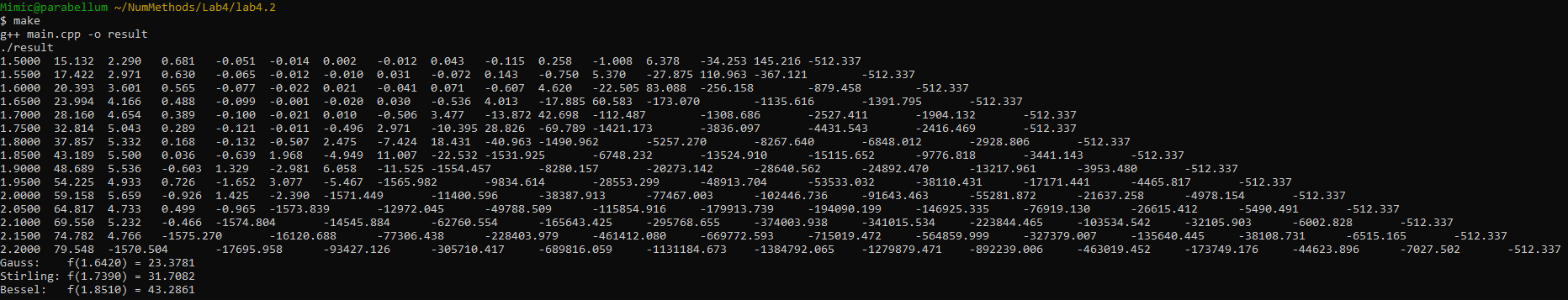


Для решения второго задания была разработана программа lab4.2

В программе реализованы интерполяционные формулы Гаусса, Стирлинга и Бесселя. Вычисления производятся, основываясь на таблице разностей, которая была построена методом из предыдущего задания, но с новыми значениями.

Результат работы программы:

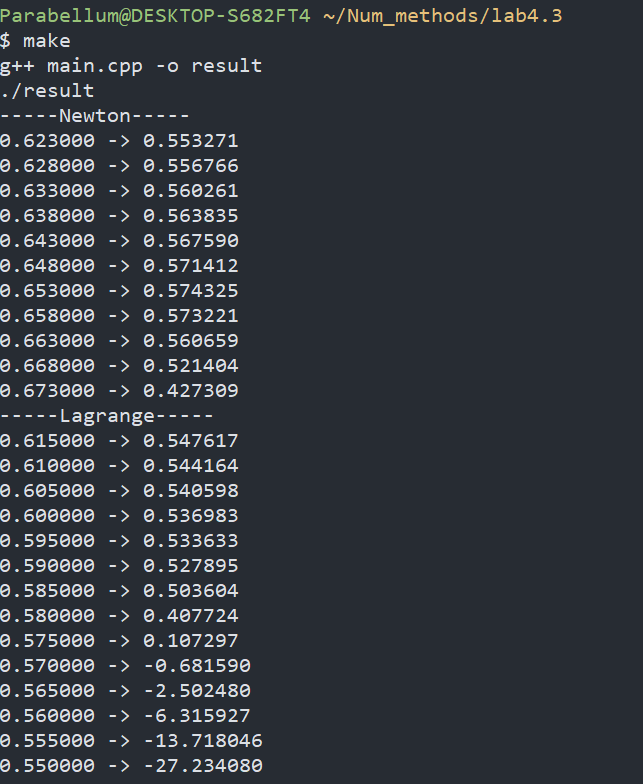




Для решения третьего задания была разработана программа lab4.3

В программе были реализованы вычисления с помощью интерполяционной формулы Ньютона для неравноотстоящих узлов и многочлена Лагранжа

Результат работы программы:



**Выводы**

Из реализации первого номера был сделан вывод о падении точности, при использовании первой интерполяционной формулы Ньютона ближе к концу таблицы.

При написании второго номера были получены сведения о различном поведении формул Бесселя, Гаусса и Стирлинга в разных местах таблицы разностей. На самых первых и самых последних значениях в таблицы функции дают неопределенный или неправдивый результат.

Из результатов третьего номера был сделан вывод о том, что результаты функции, при вычислении ее вне заданного интервала имеют чрезмерно большую погрешность, которая возрастает при отдалении от концов интервала.

**Приложения**

*Код программы «lab4.1»*

#include <iostream>

#include <cmath>

#include <stdlib.h>

#include <iomanip>

#include <stdio.h>

#include <string.h>

using namespace std;

double A[13][13];

double X[13] = {0.101, 0.106, 0.111, 0.116, 0.121, 0.126, 0.131, 0.136, 0.141, 0.146, 0.151, 0.156, 0.161};

double Y[13] = {1.26183, 1.27644, 1.29122, 1.30617, 1.32130, 1.33660, 1.35207, 1.36773, 1.38357, 1.39959, 1.41579, 1.42683, 1.43356};

bool IT (double x, double y) // interval test

{

//if( (x < X[0]) || (x > X[13]) )

//return false;

int nearest\_idx = 0;

double mod\_x = fabs(x - X[0]);

for(int i = 1; i < 13; i++)

{

if(x < X[i])

{

if((y < Y[i])&&(y>Y[i-1]))

return true;

else

return false;

break;

}

}

}

void init\_A()

{

for(int i = 0; i < 13; i++)

{

memset(A[i], 0, sizeof(double)\*13);

}

memcpy(A[0], Y, sizeof(double)\*13);

for(int i = 1; i < 13; i++)

{

for(int j = 0; j < 13-i; j++)

{

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

}

}

for(int i = 1; i < 13; i++)

{

A[12][i] = A[12][0];

}

for(int i = 1; i < 13; i++)

{

for(int j = 1+i; j < 13; j++)

{

A[12-i][j]=A[12-i][j-1]+A[13-i][j-1];

}

}

}

void show\_A()

{

printf("\n X\t Y\n");

for(int i = 0; i < 13; i++)

{

printf("%.4f\t", X[i]); //Эта строка выведет значения Х в таблице перед дельта У

for(int j = 0; j < 13; j++)

{

printf("%.4f\t", A[j][i]);

}

printf("\n");

}

}

double calculate\_for(double x)

{

int nearest\_idx = 0;

double mod\_x = fabs(x - X[0]);

for(int i = 0; i < 13; i++)

{

if(fabs(x - X[i]) < mod\_x)

{

mod\_x = fabs(x - X[i]);

nearest\_idx = i;

}

}

double h = X[1] - X[0];

double q = (x - X[nearest\_idx])/h;

printf("%.4f\t", x);

double result = Y[nearest\_idx];

double buf;

for(int i = 1; i < 13; i++)

{

buf = q;

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

{

buf \*= (q-j);

}

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

{

buf /= i;

}

buf\*=A[i][nearest\_idx];

result += buf;

}

printf("%.6f %s", result, (IT(x,result))?"":"");

printf("\n");

}

int main()

{

init\_A();

show\_A();

printf("\n X\t f(x)\n");

for(int i = 1; i < 31; i++)

{

calculate\_for(0.1026+0.002\*i);

}

return 0;

}

*Код программы «lab4.2»*

#include <iostream>

#include <cmath>

#include <stdlib.h>

#include <iomanip>

#include <stdio.h>

#include <string.h>

using namespace std;

int N = 7; // Вариант

double A[15][15];

double X[15] = {1.50, 1.55, 1.60, 1.65, 1.70, 1.75, 1.80, 1.85, 1.90, 1.95, 2.00, 2.05, 2.10, 2.15, 2.20};

double Y[15] = {15.132, 17.422, 20.393, 23.994, 28.160, 32.814, 37.857, 43.189, 48.689, 54.225, 59.158, 64.817, 69.550, 74.782, 79.548};

unsigned factorial (unsigned n){

if (1==n || 0==n)

return 1;

else

return n\*factorial(n-1);

}

void init\_A()

{

for(int i = 0; i < 15; i++)

{

memset(A[i], 0, sizeof(double)\*15);

}

memcpy(A[0], Y, sizeof(double)\*15);

for(int i = 1; i < 15; i++)

{

for(int j = 0; j < 15-i; j++)

{

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

}

}

for(int i = 1; i < 15; i++)

{

A[14][i] = A[14][0];

}

for(int i = 1; i < 15; i++)

{

for(int j = 1+i; j < 15; j++)

{

A[14-i][j]=A[14-i][j-1]+A[15-i][j-1];

}

}

}

void show\_A()

{

for(int i = 0; i < 15; i++)

{

printf("%.4f\t", X[i]); //Эта строка выведет значения Х в таблице перед дельта У

for(int j = 0; j < 15; j++)

{

printf("%.3f\t", A[j][i]);

}

printf("\n");

}

}

double calculate\_gauss(double x)

{

int nearest\_idx = 0;

double mod\_x = fabs(x - X[0]);

for(int i = 0; i < 15; i++)

{

if(fabs(x - X[i]) < mod\_x)

{

mod\_x = fabs(x - X[i]);

nearest\_idx = i;

}

}

double h = fabs(X[1] - X[0]);

double q = (x - X[nearest\_idx])/h;

int n = (nearest\_idx < 15-nearest\_idx)? nearest\_idx : 15-nearest\_idx;

double result = Y[nearest\_idx];

double buf;

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

{

//1

buf = q+i-1;

for(int j = 1; j < 2\*i-1; j++)

{

buf \*= (q+i-1-j);

}

buf/=factorial(2\*i-1);

buf \*= A[i\*2-1][nearest\_idx-(i-1)];

result += buf;

//2

buf = q+i-1;

for(int j = 1; j < 2\*i; j++)

{

buf \*= (q+i-1-j);

}

buf/=factorial(2\*i);

buf \*= A[i\*2][nearest\_idx-i];

result += buf;

}

return result;

}

double calculate\_bessel(double x)

{

int nearest\_idx = 0;

double mod\_x = fabs(x - X[0]);

for(int i = 0; i < 15; i++)

{

if(fabs(x - X[i]) < mod\_x)

{

mod\_x = fabs(x - X[i]);

nearest\_idx = i;

}

}

double h = fabs(X[1] - X[0]);

double q = (x - X[nearest\_idx])/h;

int n = (nearest\_idx < 15-nearest\_idx)? nearest\_idx : 15-nearest\_idx;

double result = (Y[nearest\_idx]+Y[nearest\_idx+1])/2;

double buf;

result += (q-0.5)\*A[1][nearest\_idx];

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

{

//1

buf = q-0.5;

for(int j = 1; j < 2\*i; j++)

{

buf \*= (q+i-1-j);

}

buf/=factorial(2\*i+1);

buf \*= A[i\*2+1][nearest\_idx-i];

result += buf;

//2

buf = q+i-1;

for(int j = 1; j < 2\*i; j++)

{

buf \*= (q+i-1-j);

}

buf/=factorial(2\*i);

buf \*= (A[i\*2][nearest\_idx-i] + A[i\*2][nearest\_idx-i+1])/2;

result += buf;

}

return result;

}

double q\_S\_search (double q, int n\_counter) { /// Поиск коэффициента (q+n-1)\*...\*(q-n) для формулы Стирлинга

double res = q;

for (int i=1; i<n\_counter; i++){

res = res\*(pow(q, 2) - pow(i, 2));

}

return res;

}

double calculate\_stirling (double sought\_x) {

/// Поиск ближайшего x

double res = 0;

int idx\_closest\_x = 0;

for (int j=0; j<15; j++){

if (fabs(X[j]-sought\_x) < fabs(X[idx\_closest\_x]-sought\_x)) {

idx\_closest\_x = j;

}

}

/// Вычисление q (шаг)

double q = 1;

q = (sought\_x-X[idx\_closest\_x])/(X[1]-X[0]);

for (int i = 0, j = 0; i < 2\*idx\_closest\_x; i++){

double coeff;

switch (i){

case 0:

coeff = 1;

break;

case 1:

coeff = q;

break;

case 2:

coeff = pow(q, 2);

default:

coeff = q\_S\_search(q, j)/factorial(i);

break;

}

double y = 0;

if (i%2 == 1){

j++;

y = (A[i][idx\_closest\_x-j] + A[i][idx\_closest\_x-j+1])/2;

} else {

y = A[i][idx\_closest\_x-j];

}

res += coeff\*y;

}

return res;

}

void show\_closest(double x)

{

int nearest\_idx = 0;

double mod\_x = fabs(x - X[0]);

for(int i = 0; i < 15; i++)

{

if(x < X[i])

{

printf("\n%f < %+0.1f > %f\n", Y[i-1], (x-X[i-1])/(X[1]-X[0])\*200-100, Y[i]);

printf(" %f\n\n", (x-X[i-1])/(X[1]-X[0])\*(Y[i]-Y[i-1])+Y[i-1]);

break;

}

}

}

bool IT (double x, double y) // interval test

{

int nearest\_idx = 0;

double mod\_x = fabs(x - X[0]);

for(int i = 0; i < 15; i++)

{

if(x < X[i])

{

if((y < Y[i])&&(y>Y[i-1]))

return true;

else

return false;

break;

}

}

}

int main()

{

init\_A();

show\_A();

double x;

x = 1.618;

//show\_closest(x);

x = 1.6+0.006\*N;

printf("Gauss: f(%.4f) = %.4f %s\n", x, calculate\_gauss(x), (IT(x, calculate\_gauss(x)))?"":"-");

x = 1.725+0.002\*N;

printf("Stirling: f(%.4f) = %.4f %s\n", x, calculate\_stirling(x), (IT(x, calculate\_stirling(x)))?"":"-");

x=1.83+0.003 \* N;

printf("Bessel: f(%.4f) = %.4f %s\n", x, calculate\_bessel(x), (IT(x, calculate\_bessel(x)))?"":"-");

return 0;

}

*Код программы «lab4.3»*

#include <iostream>

#include <cmath>

#include <stdlib.h>

#include <iomanip>

#include <stdio.h>

#include <string.h>

using namespace std;

int N = 7;

double X[9] = {0.593, 0.598, 0.605, 0.613, 0.619, 0.627, 0.632, 0.64, 0.65};

double Y[9] = {0.53205, 0.535625, 0.540598, 0.546235, 0.550431, 0.555983, 0.559428, 0.568738, 0.575298};

//double X[4] = {0, 2.5069, 5.0154, 7.5227};

//double Y[4] = {0.3989423, 0.3988169, 0.3984408, 0.3978138};

double razn (int f, int l)

{

if( l-f == 1 )

{

return (Y[l]-Y[f])/(X[l]-X[f]);

}else

{

return (razn(f+1, l)-razn(f, l-1))/(X[l]-X[0]);

}

}

double newton (double x)

{

double res = Y[0];

double buf;

int num\_of\_parts = 8;

for(int i = 1; i <= num\_of\_parts; i++)

{

buf = razn(0, i);

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

{

buf \*= x-X[j];

}

res += buf;

}

return res;

}

double lagrange (double x)

{

double res = 0;

double buf;

int num\_of\_parts = 9;

for(int i = 0; i < num\_of\_parts; i++)

{

buf = Y[i];

for(int j = 0; j < num\_of\_parts; j++)

{

if(j != i)

{

buf \*= x-X[j];

buf /= X[i]-X[j];

}

}

res += buf;

}

return res;

}

bool IT (double x, double y) // interval test

{

if( (x < X[0]) || (x > X[8]) )

return false;

int nearest\_idx = 0;

double mod\_x = fabs(x - X[0]);

for(int i = 1; i < 9; i++)

{

if(x < X[i])

{

if((y < Y[i])&&(y>Y[i-1]))

return true;

else

return false;

break;

}

}

}

int main()

{

double x;

printf("-------Newton-------\n");

for(int i = 3; i < 2\*N; i++) //Newton

{

x = 0.608 + 0.025 \* i;

printf("f(%.3f) = %f %s\n", x, newton(x), (IT(x, newton(x)))?"+":"" );

}

printf("-------Lagrange-----\n");

for(int i = N; i < 3\*N; i++) //Lagrange

{

x = 0.630 - 0.025 \* i;

printf("f(%.3f) = %f %s\n", x, lagrange(x), (IT(x, lagrange(x)))?"+":"");

}

/\*

for(int i = 0; i < 20; i++)

{

x = 0.6 + 0.0025 \*i;

printf("%f -> %f %s\n", x, newton(x), (IT(x, newton(x)))?"+":"--" );

printf("%f -> %f %s\n", x, lagrange(x), (IT(x, lagrange(x)))?"+":"--");

}

\*/

return 0;

}