НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

**Лабораторна робота № 15**

**Інтерполяція даних. Інтерполяційний поліном Лагранжа. Рівномірне (чебишовське) наближення функцій**

**З курсу Обчислювальна математика**

Вариант № 21

Виконтов студент

групи ДП-82

Мнацаканов Антон

Факультет Електроніки

Викладач: Домбругов М.Р.

Київ-2020

**Мета роботи**: застосування алгоритмів інтерполяції для побудови поліноміального наближення функції.

**Що зробити**: побудувати поліноміальне наближення до функції f (x) за допомогою інтерполяційного полінома з вузлами, що розташовані на кривій f (x). Дослідити величину дефекту наближення в залежності від числа вузлів. Додатково – порівняти випадок рівновіддалених вузлів та вузлів з абсцисами Чебишова.

Хід роботи

Основний на С:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

double a=-1.0;

double b=2.0;

double n=5;

double func(double x){

return x\*x\*10.0\*exp(-x)-x\*3.0;

}

double lagrange\_func(int n, double\*\* nodes, double x){

double res = 0;

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

double koef = 1;

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

if(i!=j){

koef \*= (x-nodes[i][0])/(nodes[j][0]-nodes[i][0]);

}

}

res += koef\*nodes[j][1];

}

return res;

}

double\*\* calc\_lagrange\_nodes(int nn){

double\*\* nodes = malloc(nn\*sizeof(double\*));

for(int i=0;i<nn; ++i){

nodes[i] = malloc(2\*sizeof(double));

nodes[i][0] = a + ((b-a)/(nn-1))\*i;

nodes[i][1] = func(nodes[i][0]);

}

return nodes;

}

double\*\* calc\_chebishev\_nodes(int nn){

double\*\* nodes = malloc(nn\*sizeof(double\*));

for(int i=0;i<nn; ++i){

nodes[i] = malloc(2\*sizeof(double));

nodes[i][0] = (a+b)/2.0+((b-a)/2.0)\*cos(((nn-(i+1)+0.5)/nn)\*M\_PI);

nodes[i][1] = func(nodes[i][0]);

}

return nodes;

}

int main(){

double\*\* lagrange\_nodes3 = calc\_lagrange\_nodes(3);

double\*\* lagrange\_nodes4 = calc\_lagrange\_nodes(4);

double\*\* lagrange\_nodes10 = calc\_lagrange\_nodes(10);

double\*\* lagrange\_nodes16 = calc\_lagrange\_nodes(16);

double\*\* chebishev\_nodes3 = calc\_chebishev\_nodes(3);

double\*\* chebishev\_nodes4 = calc\_chebishev\_nodes(4);

double\*\* chebishev\_nodes10 = calc\_chebishev\_nodes(10);

double\*\* chebishev\_nodes16 = calc\_chebishev\_nodes(16);

double d\_l3 = 0;

double d\_l4 = 0;

double d\_l10 = 0;

double d\_l16 = 0;

double d\_c3 = 0;

double d\_c4 = 0;

double d\_c10 = 0;

double d\_c16 = 0;

FILE\* out = fopen("data.dat", "w");

for(double s=a; s<=b; s+=0.01){

double f = func(s);

double l3 = lagrange\_func(3, lagrange\_nodes3, s);

double l4 = lagrange\_func(4, lagrange\_nodes4, s);

double l10 = lagrange\_func(10, lagrange\_nodes10, s);

double l16 = lagrange\_func(16, lagrange\_nodes16, s);

double c3 = lagrange\_func(3, chebishev\_nodes3, s);

double c4 = lagrange\_func(4, chebishev\_nodes4, s);

double c10 = lagrange\_func(10, chebishev\_nodes10, s);

double c16 = lagrange\_func(16, chebishev\_nodes16, s);

if(fabs(f - l3) > d\_l3){

d\_l3 = fabs(f - l3);

}

if(fabs(f - l4) > d\_l4){

d\_l4 = fabs(f - l4);

}

if(fabs(f - l10) > d\_l10){

d\_l10 = fabs(f - l10);

}

if(fabs(f - l16) > d\_l16){

d\_l16 = fabs(f - l16);

}

if(fabs(f - c3) > d\_c3){

d\_c3 = fabs(f - c3);

}

if(fabs(f - c4) > d\_c4){

d\_c4 = fabs(f - c4);

}

if(fabs(f - c10) > d\_c10){

d\_c10 = fabs(f - c10);

}

if(fabs(f - c16) > d\_c16){

d\_c16 = fabs(f - c16);

}

fprintf(out, "%e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e\n", s, f, l3, l4, l10, l16, f-l3, f-l4, f-l10, f-l16, c3, c4, c10, c16, f-c3, f-c4, f-c10, f-c16);

}

fclose(out);

printf("n\t\tlagrange\t\tchebishev\n");

printf("3\t\t%e\t\t%e\n", d\_l3, d\_c3);

printf("4\t\t%e\t\t%e\n", d\_l4, d\_c4);

printf("10\t\t%e\t\t%e\n", d\_l10, d\_c10);

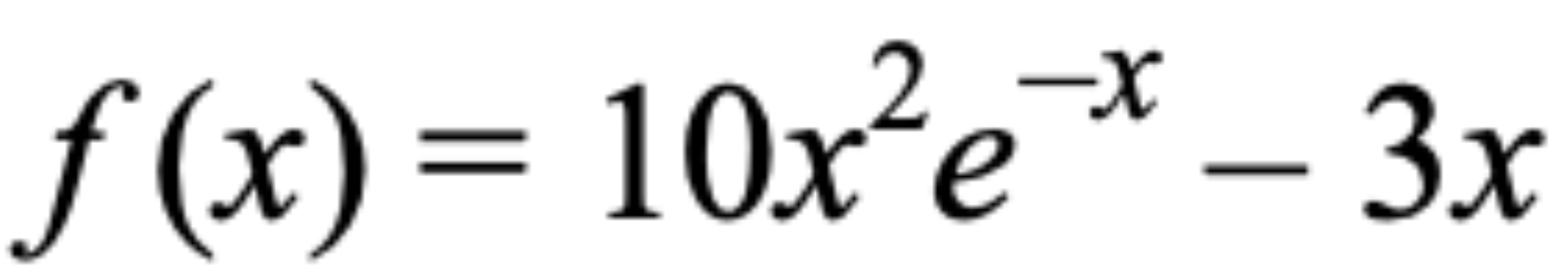
printf("16\t\t%e\t\t%e\n", d\_l16, d\_c16);

return 0;

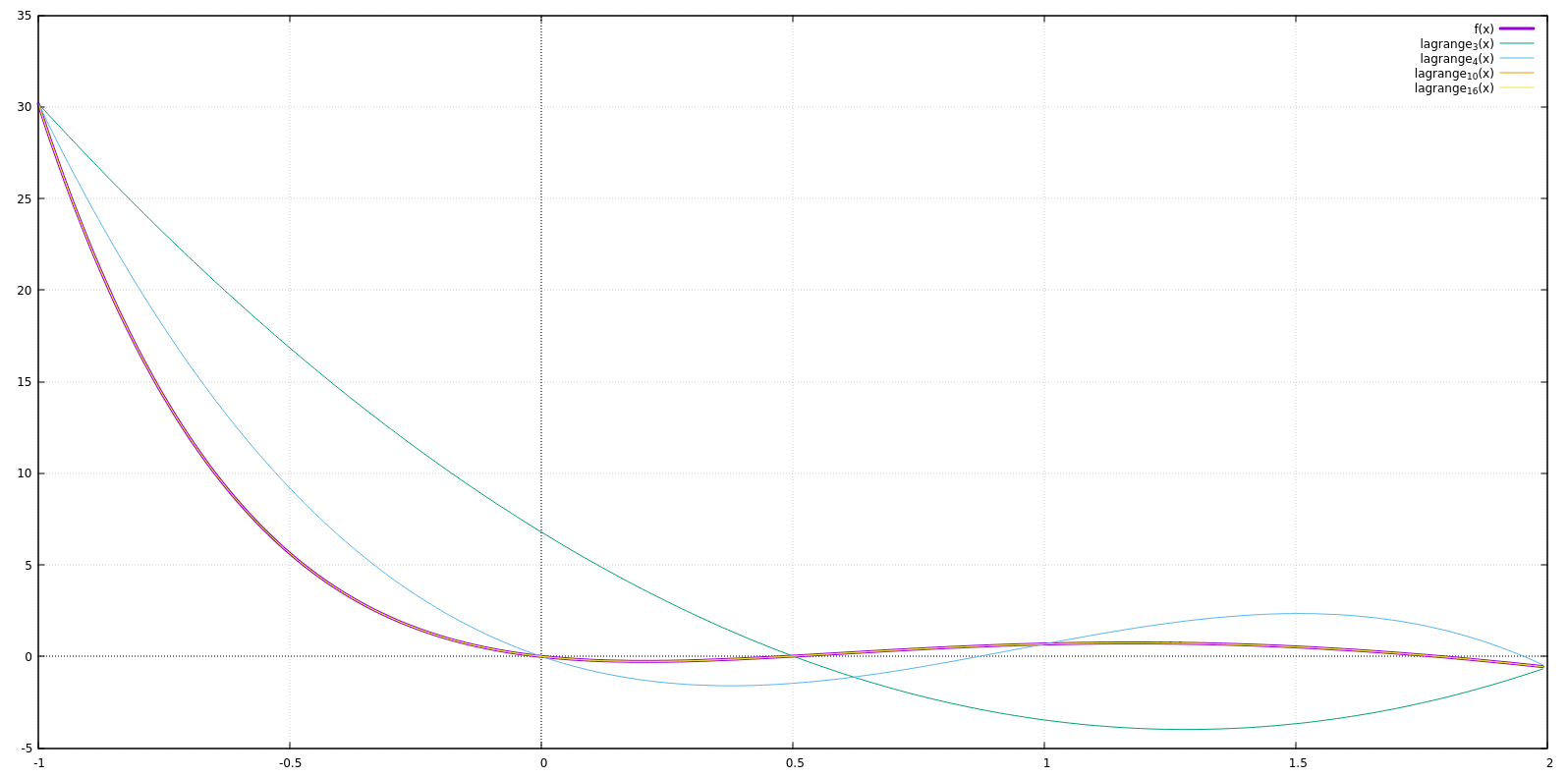
}

Результати

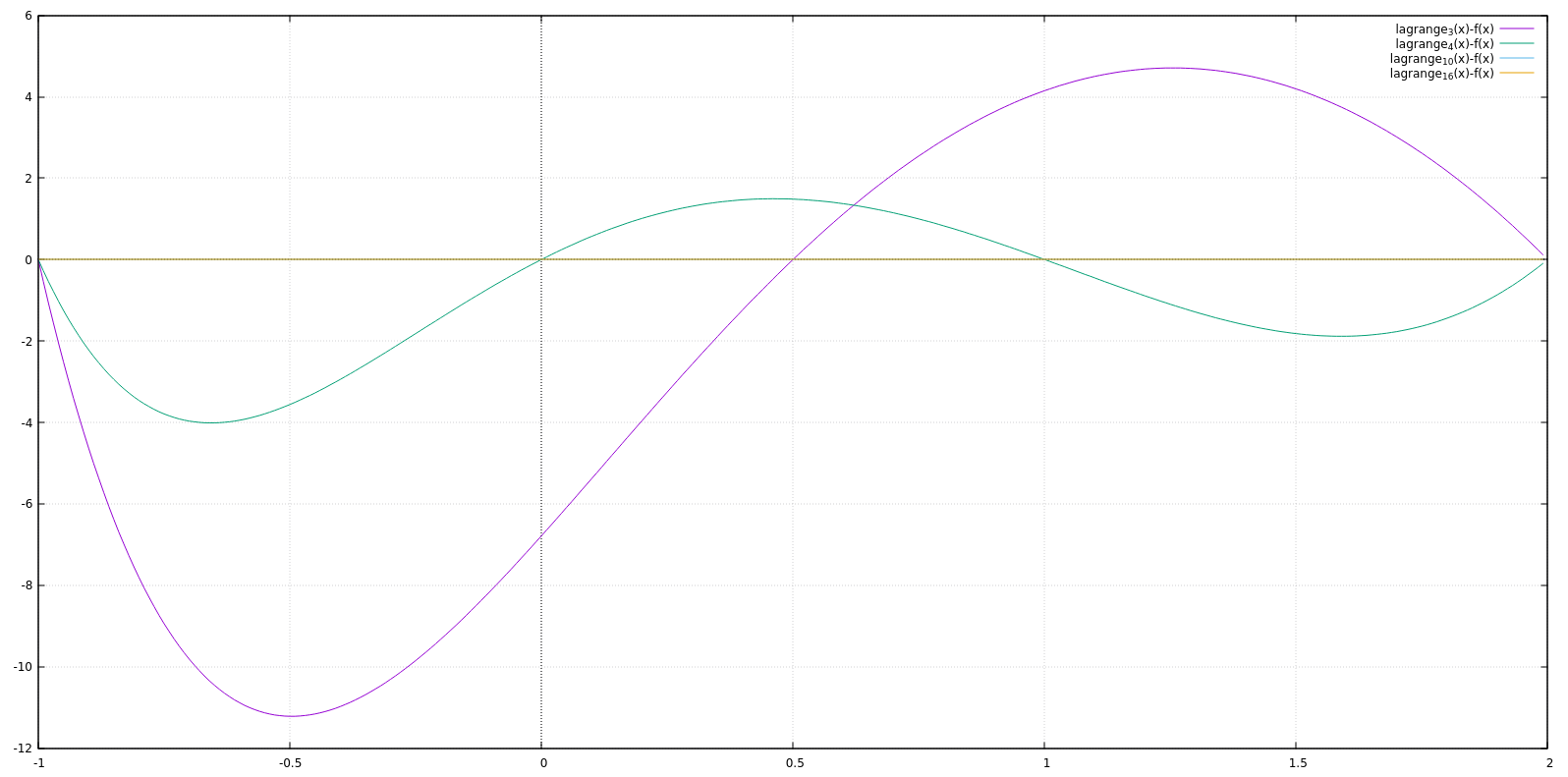
Це моя головна функція

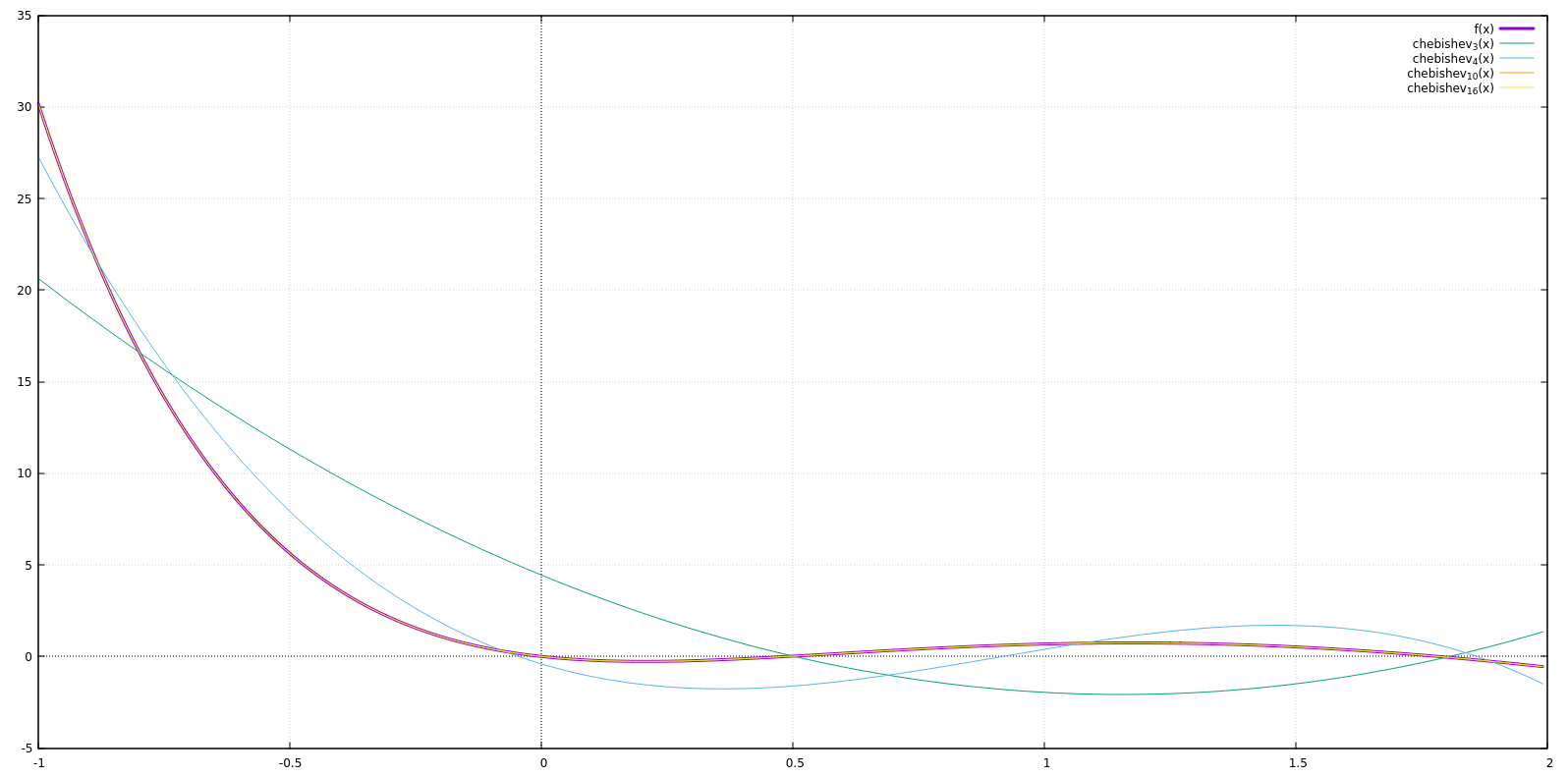
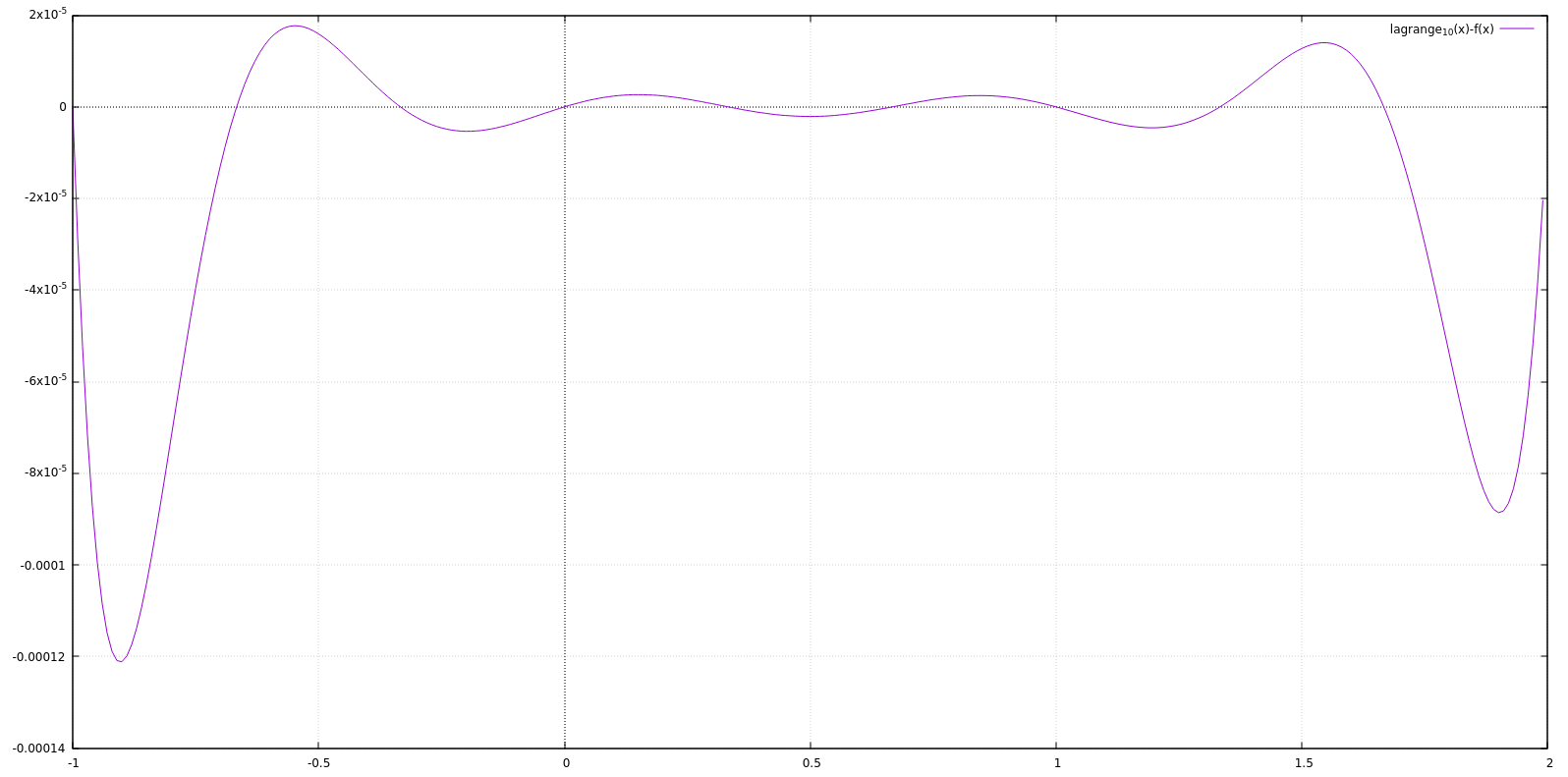


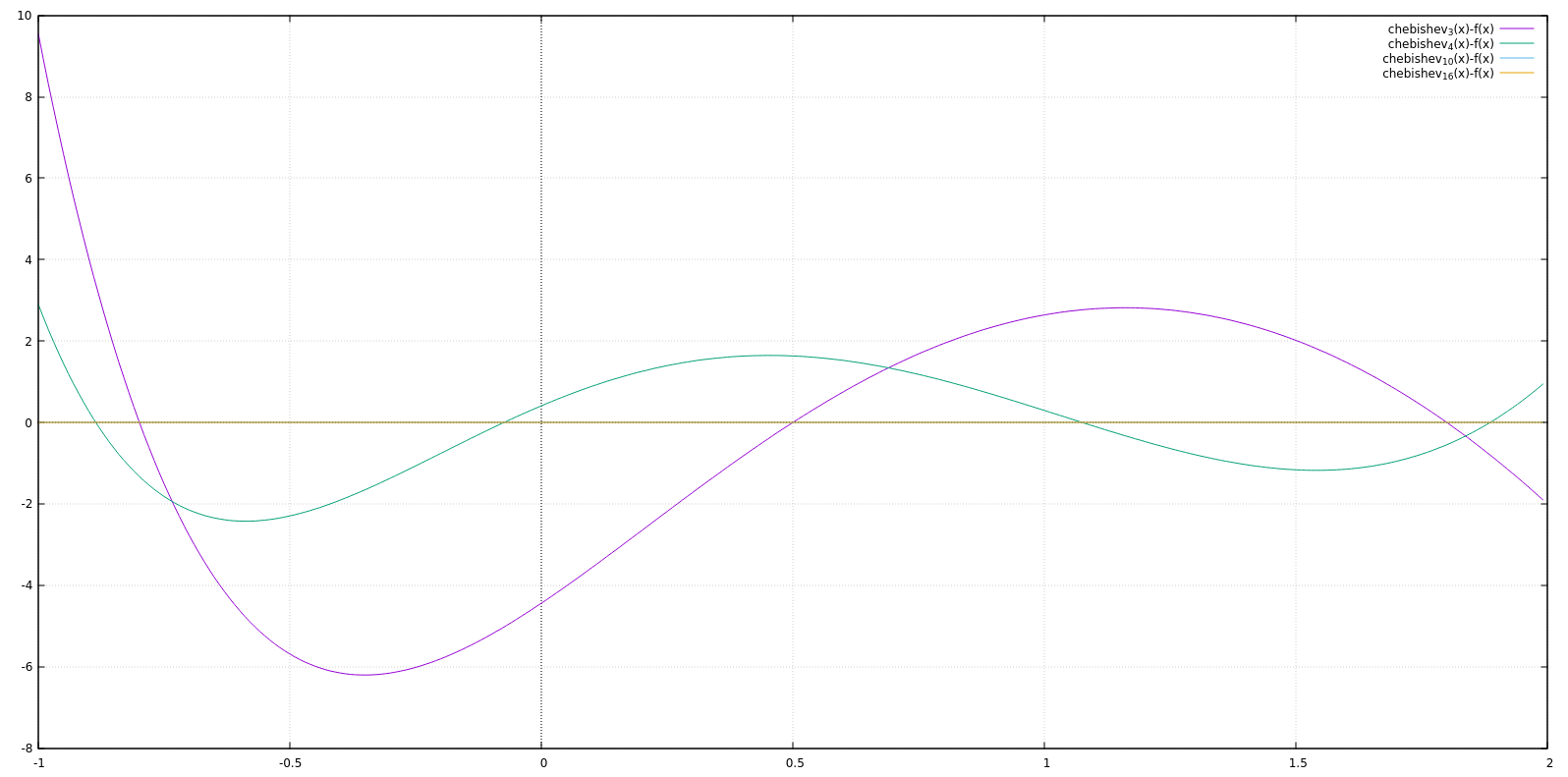
Графіки всіх функцій



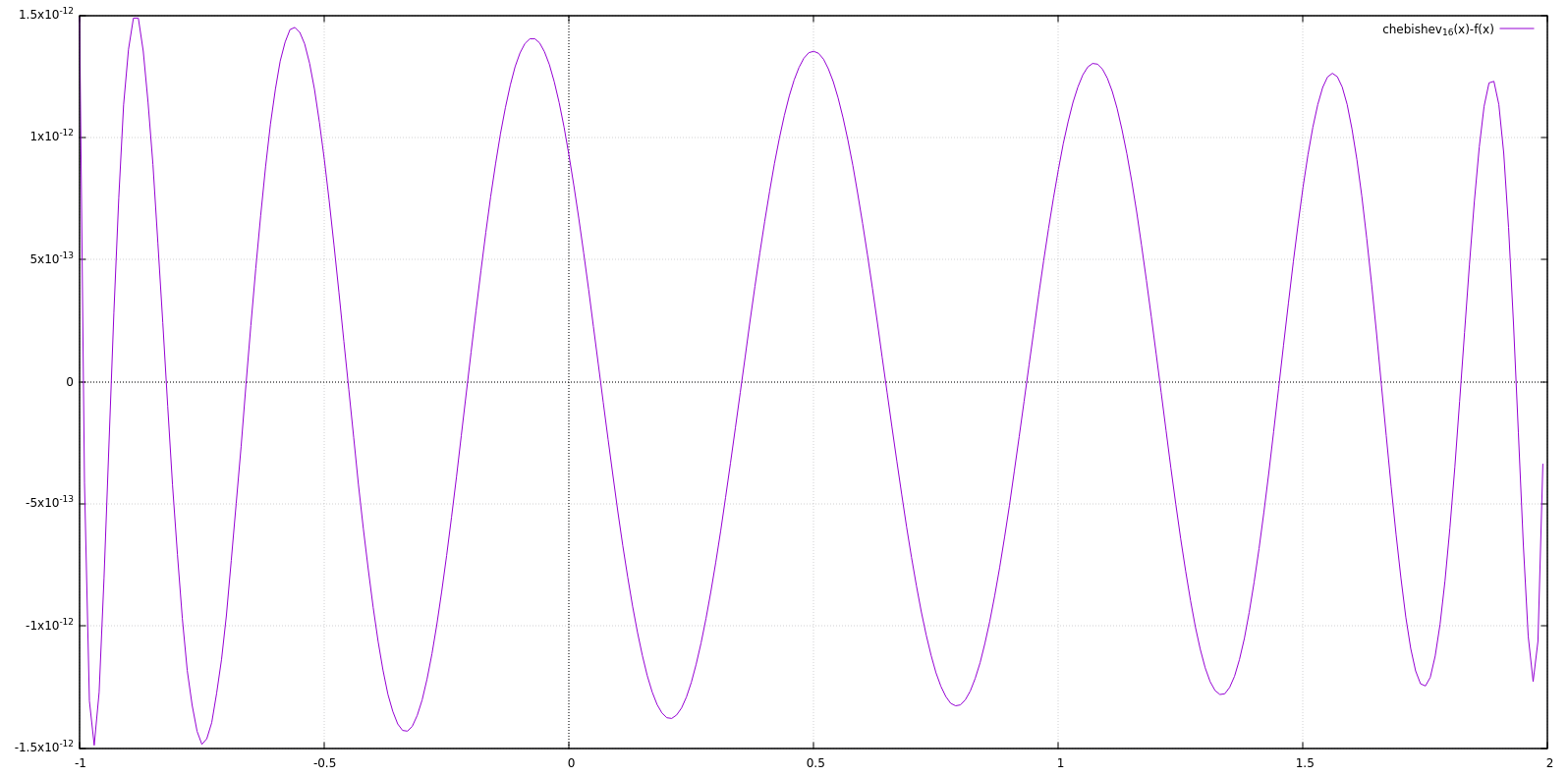
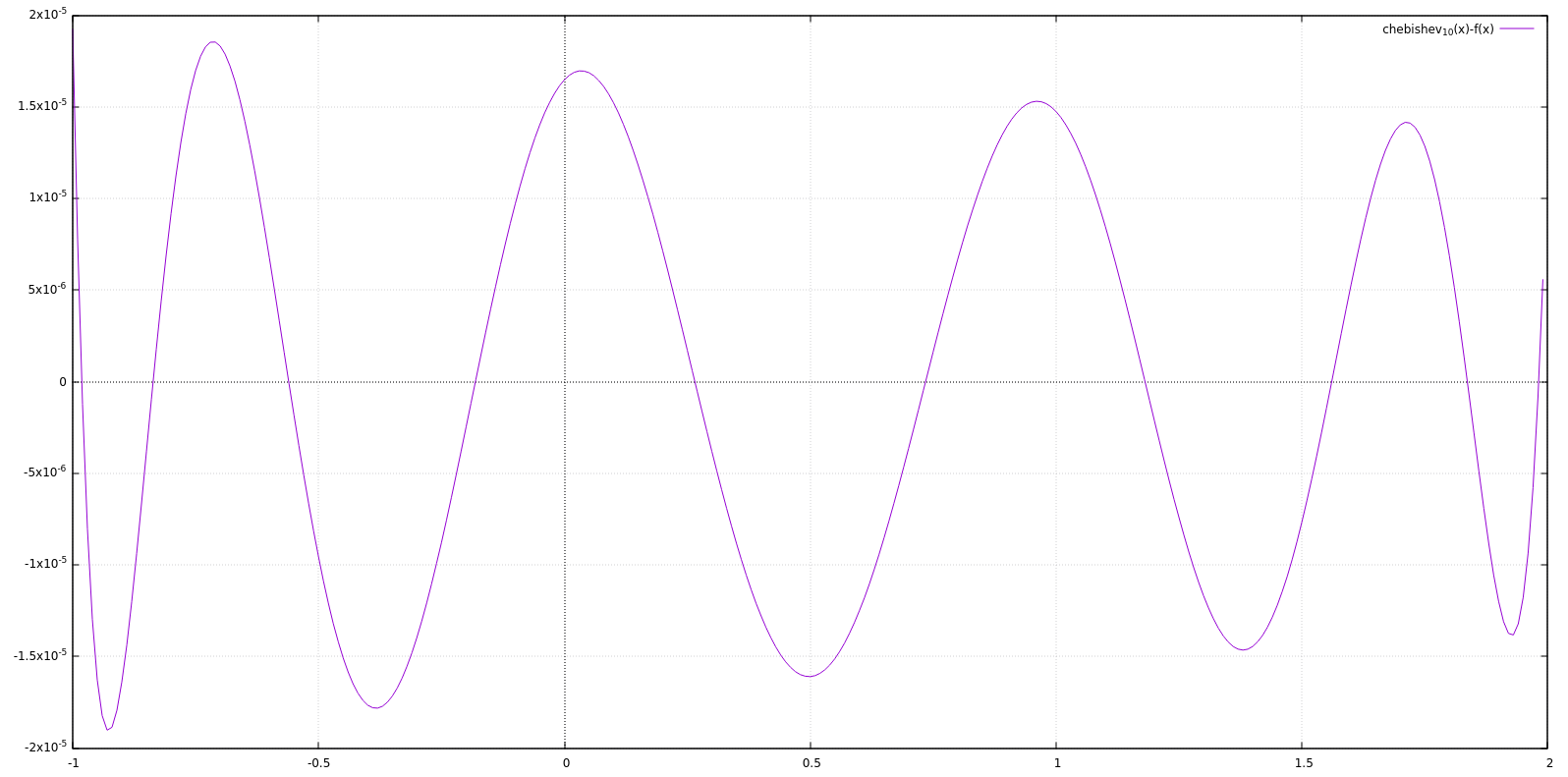
Різниця між «предсказанной функцией и оригинальной» для 3,4,10 та 16 точок



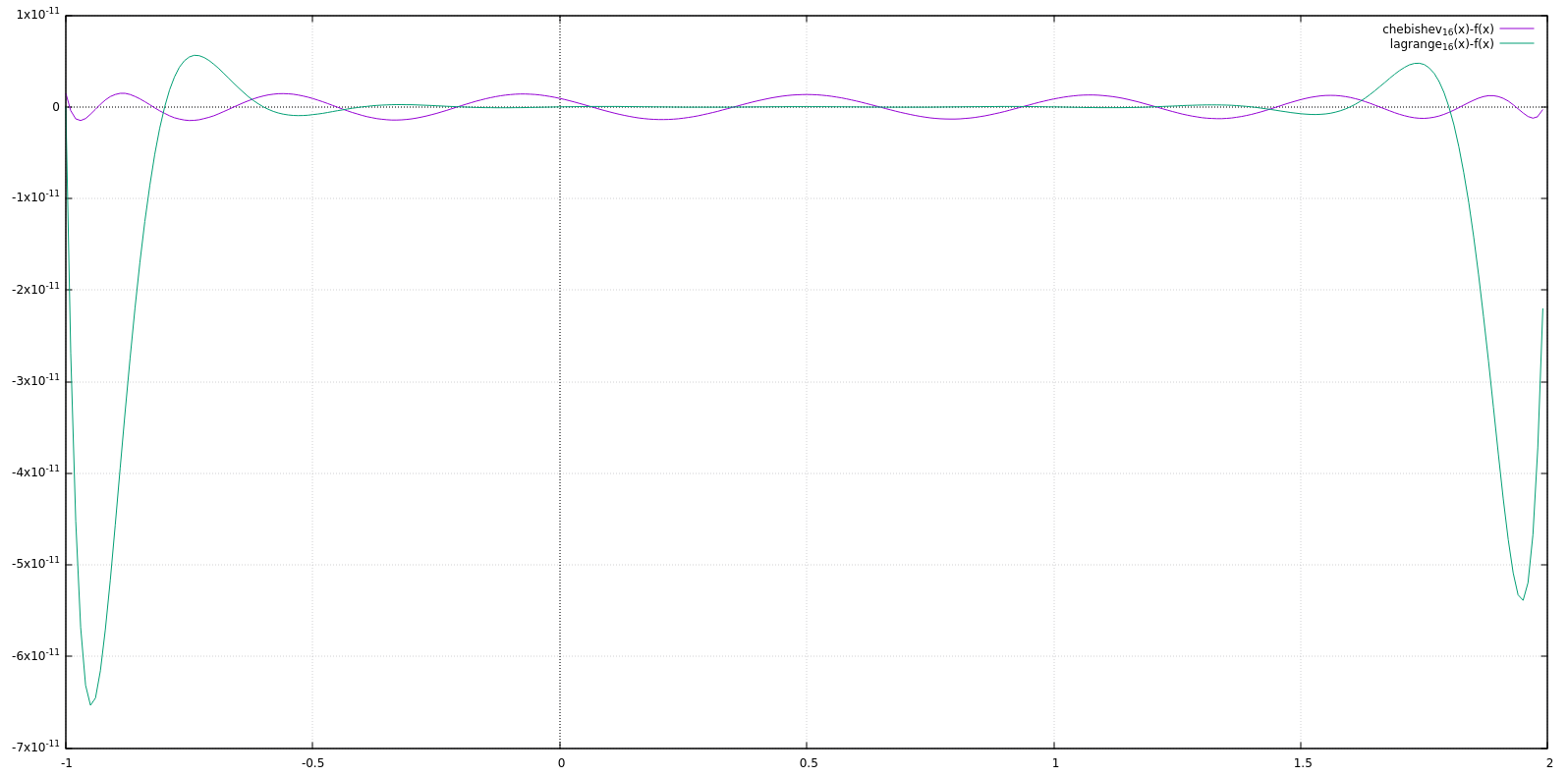
похибка в лагранже

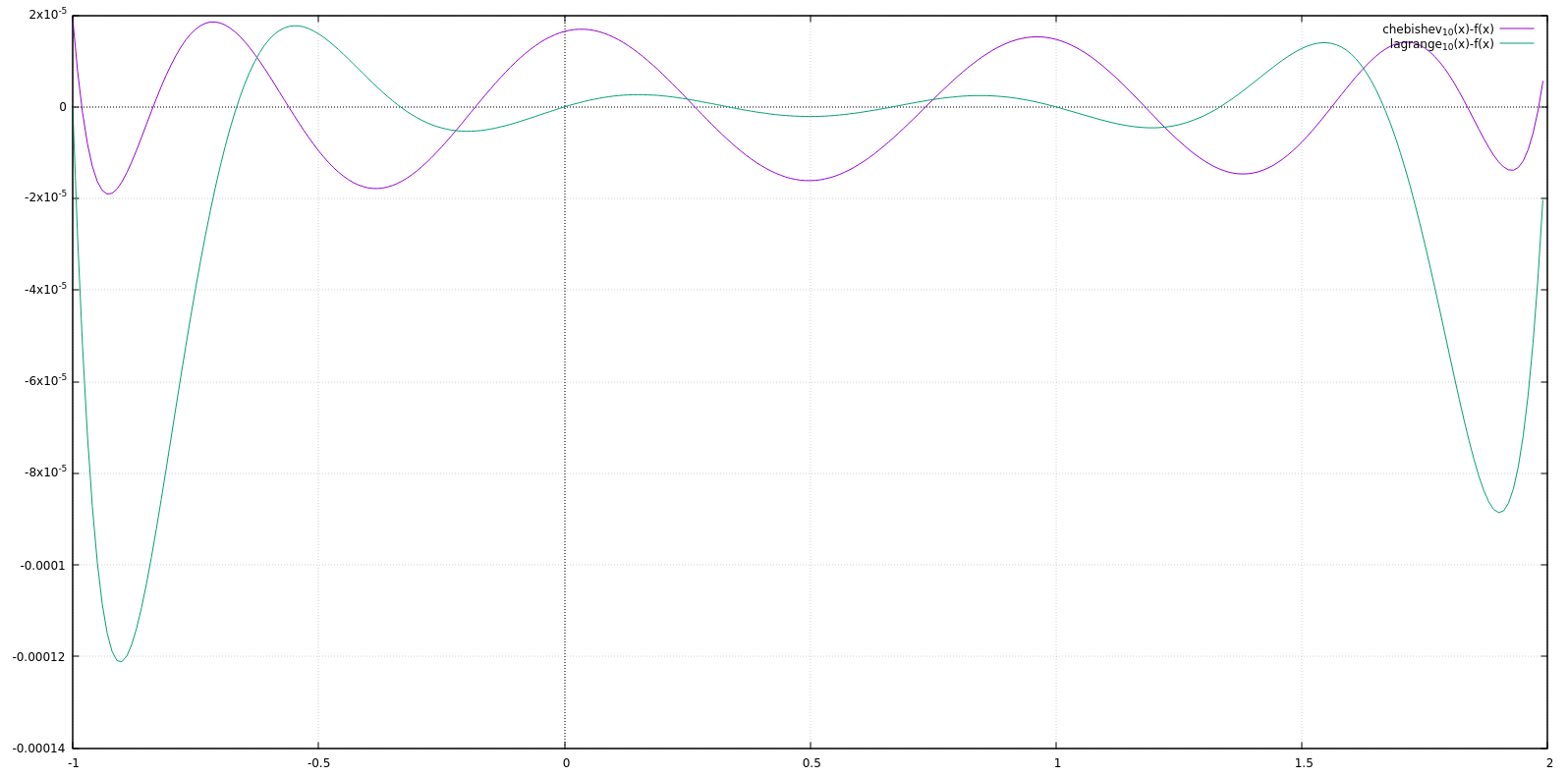


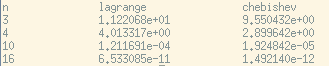
Графіки для Чебишева для 10 та 16 точок



Порівняння Чебышева и Лагранжа для 10 16.







**Висновок:**  написав программу застосування алгоритмів інтерполяції для побудови поліноміального наближення функці, побудував поліноміальне наближення до функції f (x) за допомогою інтерполяційного полінома з вузлами, що розташовані на кривій f (x). Дослідив величину дефекту наближення в залежності від числа вузлів також порівняв випадок рівновіддалених вузлів та вузлів з абсцисами Чебишова.

Проаналізував залежність дефекту наближення D від способу призначення вузлів, отримані результати заніс в таблицю, а всі графіки були побудовані по точкам ( x;y1;y2) за допомогою программи **gnuplot** ( [свободная](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%9F%D0%9E) программа для создания двух- и [трёхмерных](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D1%91%D1%85%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0) [графиков](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D0%BA_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8))