Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет електроніки

Кафедра мікроелектроніки

Лабораторна робота №15

Варіант №3

Виконав: студент групи ДП-71

Грабар Олександр

Перевірив : доц. Домбругов М.Р.

**Інтерполяція даних.**

**Інтерполяційний поліном Лагранжа.**

**Рівномірне (Чебишевське) наближення функцій**

**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. Розглянути

випадок вузлів з рівновіддаленими абсцисами та

абсцисами Чебишева.

*Код програми:*

#include <iostream>

#include <math.h>

#include <iomanip>

using namespace std;

double f(double x) {

return (cosh(x)-2\*x\*x+3);

}

int main()

{

setlocale (LC\_ALL, "Russian");

float a=-0.2, b=0.2;

const int n=10;

double D, Dmax, Dmaxcheb;

double x\_[n+1], y\_[n+1], xCheb[n+1], yCheb[n+1];

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

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

x\_[i]=a+(b-a)\*i/n;

y\_[i]=f(x\_[i]);

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

}

D=0;

Dmax=0;

cout<<endl<<"x"<<setw(15)<<"L(x)"<<setw(28)<<"f(x)-L(x)"<<endl;

double L, ch, zn;

for (float x=-0.2; x<=0.2; x+=0.04){

L=0;

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

ch=zn=1;

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

if(i!=j){

ch \*=x-x\_[j];

zn \*=x\_[i]-x\_[j];

}

}

L +=y\_[i] \* ch/zn;

D=f(x)-L;

}

cout<<"x="<<setw(2)<<x<<setw(10)<<"L="<<setw(10)<<L<<setw(10)<<"D="<<D<<endl;

if(fabs(D)>Dmax){

Dmax=fabs(D);

}

}

cout<<"\nmaxD="<<Dmax<<endl;

cout<<"\nЧебишевське"<<endl;

cout<<"x[i]"<<setw(29)<<"y[i]"<<endl;

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

xCheb[i]=(a+b)/2 + ((b-a)/2)\*cos((i+0.5)\*3.1415/(n+1));

yCheb[i]=f(xCheb[i]);

cout<<"X(cheb)="<<xCheb[i]<<setw(20)<<"Y(cheb)="<<yCheb[i]<<endl;

}

cout<<endl<<"x"<<setw(15)<<"L(x)"<<setw(28)<<"f(x)-L(x)"<<endl;

D=0;

Dmaxcheb=0;

for(float x=-0.2;x<=0.2;x+=0.04){

L=0;

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

ch=zn=1;

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

if(i!=j){

ch \*=x-xCheb[j];

zn \*=xCheb[i]-xCheb[j];

}}

L +=yCheb[i]\*ch/zn;

D=f(x)-L;

}

if(fabs(D)>Dmaxcheb){

Dmaxcheb=fabs(D);

}

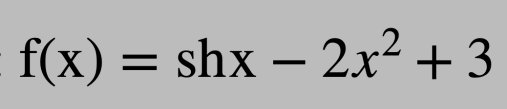
cout<<"x="<<setw(2)<<x<<setw(10)<<"L="<<setw(10)<<L<<setw(10)<<"D="<<D<<endl;

}

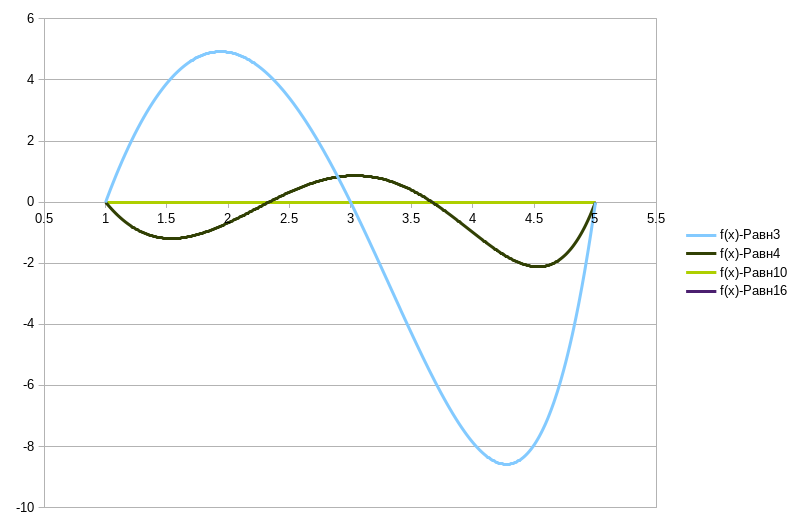
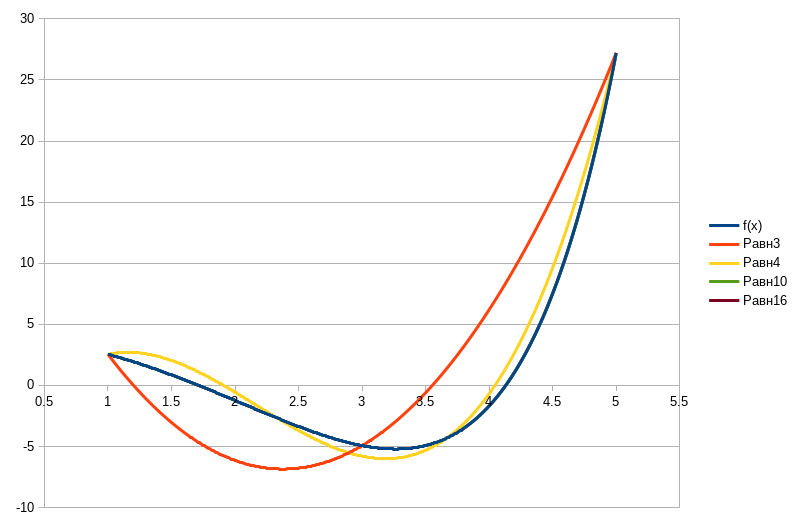
cout<<"\nmaxD(cheb)= "<<Dmaxcheb<<endl; return 0;}

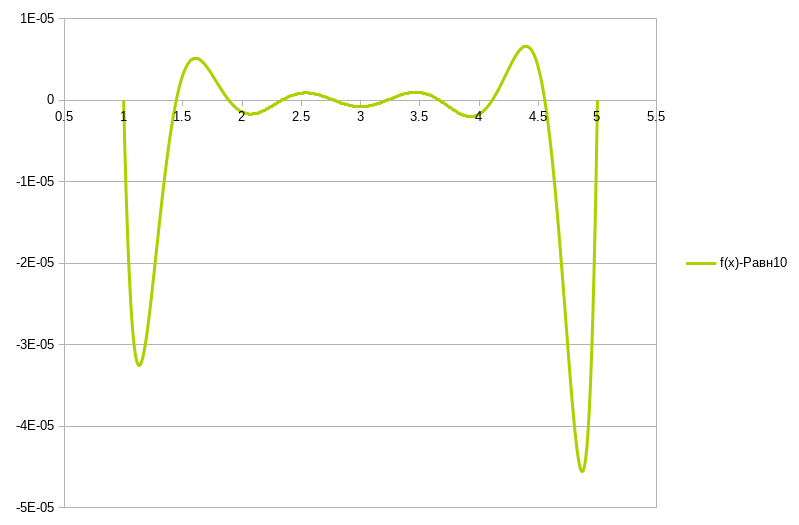
**Результати виконання програми**

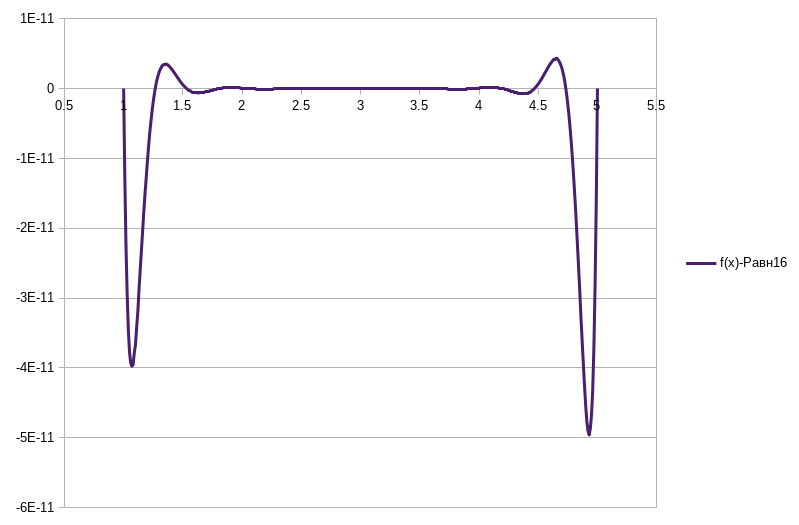
головна функція

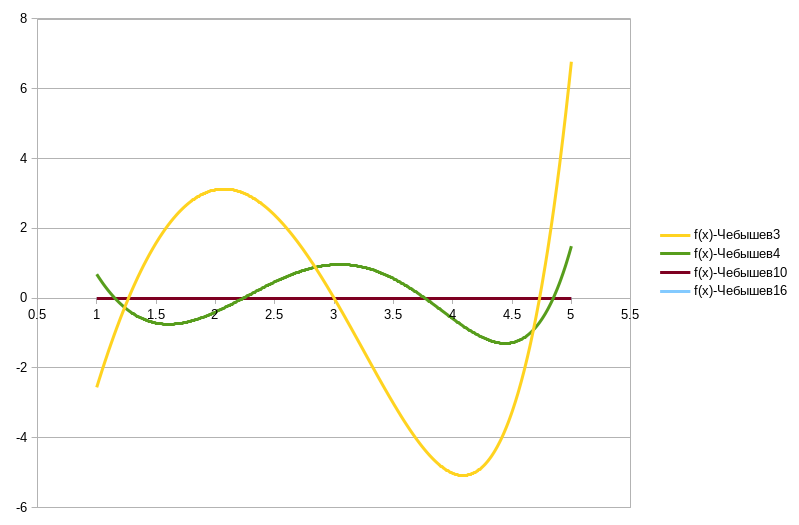


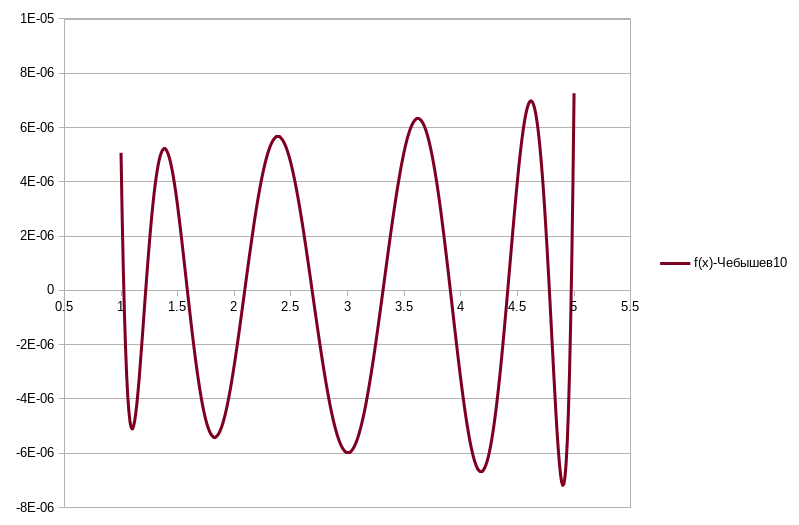
**Для рівновіддалених точок**

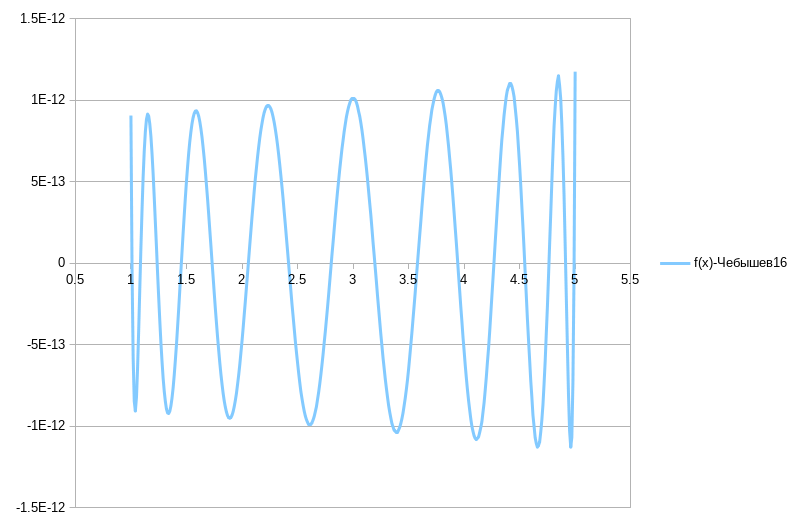
****

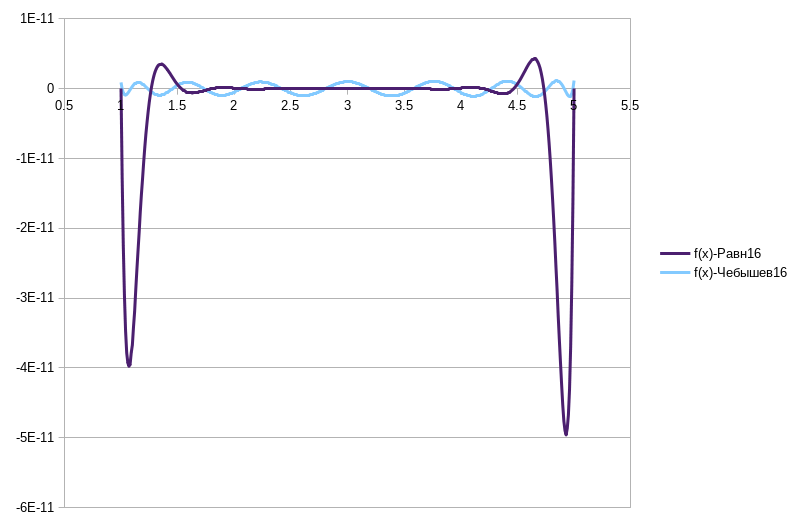
****

**Для Чебишева**

****

****

***порівняння***

****

|  |  |  |
| --- | --- | --- |
| N | Equidistant | Cherished |
| 3 | 8.584 | 6.773 |
| 4 | 2.117 | 1.489 |
| 10 | 4.560e-5 | 7.263e-6 |
| 16 | 4.959e-11 | 1.175e-12 |

**Висновок:** Я застосував алгоритм інтерполяції для побудови поліноміального наближення функції за допомогою інтерполяційного полінома з вузлами, що розташовані на кривій f (x). Дослідив величину дефекту наближення в залежності від числа вузлів. Розглянув випадок вузлів з рівновіддаленими абсцисами та абсцисами Чебишева. В залежності від кількості вузлів похибка змінюється, чим менше вузлів використовується тим більша похибка.