**Ministerul Educației, Culturii și Cercetării**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Raport**

Lucrarea de laborator nr.3

Disciplina: Metode și modele de calcul

Tema: Interpolarea funcțiilor cu ajutorul polinomului langrange

**Efectuat**: st.gr.TI-207 Bunescu Gabriel.

**Verificat**: conf. univ. dr. Dohotaru Leonid

Chișinău 2021

**Scopul lucrării:**

Pentru funcţia f :[a,b] R se cunosc valorile yi =f(xi), i=0,1,…,n în punctele distincte a=x0, x1,… ,xn=b.

1. Sa se construiasca polinomul de interpolare Lagrange Ln(x) ce aproximeaza functia data.
2. Sa se calculeze valoarea functiei f(x) intr-un punct x=ξ utilizind polinomul de interpolare Lagrange Ln(x).
3. Sa se aproximeze valoarea functiei f(x) pentru x=ξ cu eroarea ε=10-4 (sau cu cea mai buna exactitatea posibila), calculind polinomul de interpolare Lagrange Lm(x) pentu m<n.
4. Sa se compare si sa se explice rezultatele date.

**Notiuni teoretice:**

Fie funcţia y=f(x) dată sub forma unei tabele de valori :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | x0 | x1 | … | xn |
|  |  |  |  |  |
| y | y0 | y1 | … | yn |
|  |  |  |  |  |

unde yi=f(xi) i=0,1,2,…n

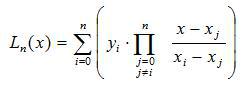
Există numeroase procedee de interpolare pentru găsirea unor valori intermediare ale lui f(x) pentru x≠xi, i=0,1,..n. De foarte multe ori pentru aproximarea funcţiilor prin interpolare se utilizează polinoamele algebrice :

Pn(x)=anxn+an-1xn-1+…+a1x+a0

Aceasta se datorează că funcţia f(x) poate fi aproximată foarte bine cu ajutorul curbelor a căror reprezentare analitică sunt polinoame( teorema Weierstrass ). Pe de altă parte, valoarea polinomului se calculează uşor (cu ajutorul schemei lui Horner). Nu apar dificultăţi şi la integrarea sau derivarea polinoamelor. Pentru ca un polinom Pn(x) de grad.≤n să interpoleze funcţia dată, trebuie ca valorile sale în nodurile x0,x1,…xn să coincidă cu valorile funcţiei, adică :

Pn(x)=yi, i=0,1,…,n.

Se demonstreză că condiţiile de interpolare determină un polinom unic, care se poate exprima sub forma:



Polinomul Ln(x) se numeşte **polinomul de interpolare Lagarange.**

Varianta:5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | 1.415 | 1.418 | 1.420 | 1.424 | 1.430 | 1.434 | 1.437 |
| y | 3.76831 | 2.17946 | 1.37197 | 0.43672 | -0.9763 | -1.6798 | -2.3712 |

**Codul programului**

#include <iostream>

#include <math.h>

#include <stdio.h>

using namespace std;

void Lagrange(float x[10], float y[10], int n, float x1);

float X[15];

float Y[15];

float function(float x) {

return ((x+2)/(x\*x+1))-x+2;

}

int main() {

int z,num;

float point;

cout << "Indicati numarul de noduri de interpolare: ";

cin >> num;

cout << "1.Functia este cunoscuta " << endl;

cout << "2.Functia nu este cunoscuta ";

cout << endl << ">>> ";

cin >> z;

if (z == 1) {

cout << "Introduceti elementele tabloului absciselor nodurilor" <<

endl;

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

cout << "x" << i << " = "; cin >> X[i];

function(X[i]);

Y[i] = function(X[i]);

}

cout << "Elementele tabloului valorilor functiei in aceste puncte" <<

endl;

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

cout << "y" << i << " = " << function(X[i]) << endl;

}

}

if (z == 2) {

cout << "Introduceti elementele tabloului absciselor nodurilor" <<

endl;

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

cout << "x" << i << " = "; cin >> X[i];

}

cout << "Elementele tabloului valorilor functiei in aceste puncte" <<

endl;

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

cout << "y" << i << " = "; cin >> Y[i];

}

}

cout << "Introduceti punctul in care doriti sa se efectueze interpolarea: ";

cin >> point;

float eps = 0.0001;

cin.get();

cout << " X ";

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

cout << " " << X[i] << " ";

cout << endl;

cout << " Y ";

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

cout << " " << Y[i] << " ";

cout << endl;

Lagrange(X, Y, num - 1, point);

cin.get();

}

void Lagrange(float x[10], float y[10], int n, float x1) { float A[10], C[10], B[10], r, y1;

int i, j, k;

C[0] = 1;

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

C[i + 1] = C[i];

for (j = i; j >= 1; j--)

C[j] = C[j - 1] - C[j] \* x[i];

C[0] = -C[0] \* x[i];

}

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

A[i] = 0; B[n] = C[n + 1];

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

r = 1; for (j = 0; j <= n; j++)

if (i != j) r = r\*(x[i] - x[j]);

for (k = n - 1; k >= 0; k--)

B[k] = C[k + 1] + x[i] \* B[k + 1];

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

A[k] = A[k] + y[i] \* B[k] / r;

}

i = n - 1;

cout << "\n 1. Polinomul Lagrange\n\n Ln(x) = " << A[n] << "\*X^" << n;

for (k = n - 1; k>0; k--) {

printf(" %+1.4f\*X^%d", A[k], i);

if (k == 1) cout << ""; i--;

}

printf(" %+1.4f", A[0]);

y1 = A[n];

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

y1 = x1\*y1 + A[n - i];

cout << "\n\n\n 2. Valoarea functiei";

cout << endl<< endl << " f(" << x1 << ") = " << y1;

}

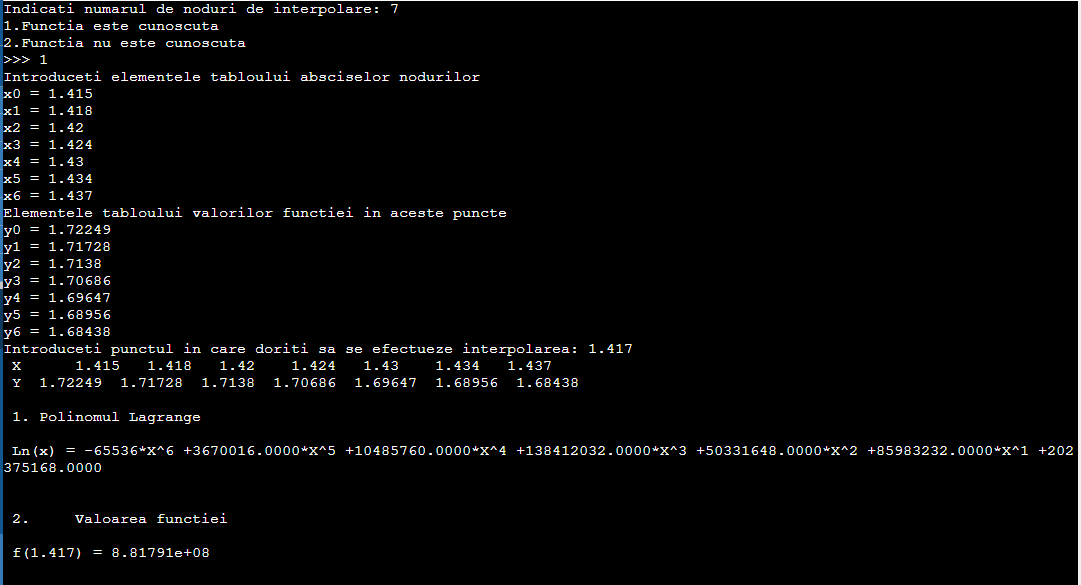


Fig.1.1. Functia este cunoscută

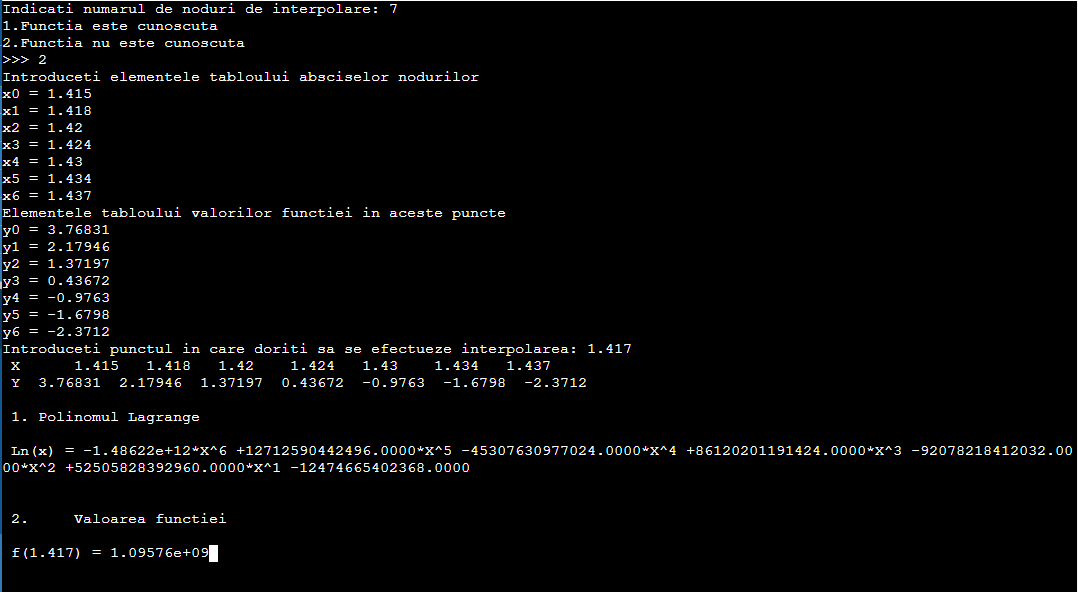


Fig.1.2. Functia este ne cunoscută

**Concluzie:**

In cadrul acestei lucrari s-a elaborat programul ce construieste polinomul de interpolare Lagrange Ln (x), calculeaza valorea functiei f(x) intr-un punct dat utilizind polinomul de interpolare Lagrange. De asemenea la afisarea raspunsului a fost efectuata şi afisarea insasi a polinomului Lagrange. Pentru ca programul sa fie mai functional, s-a realizat introducerea manuala a datelor, astfel acest program poate fi utilizat pentru rezolvarea oricarei probleme propuse, insa cu o restrictie, aceasta poate contine maximum 15 noduri de interpolari.