Ministerul Educaţiei al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare Informatica si Microelectronica

RAPORT

Lucrarea de laborator nr.3
Disciplina: MMC1

A efectuat:
st. gr. CR-191 Frecventa Redusa Balan Ion

A verificat:
lect., sup. Godonoga Anatol

Chisinau 2020

Tema: Interpolarea functiilor cu ajutorul polinomului Lagrange

Scopul lucrarii:

Pentru funcţia $f:[a,b] \longrightarrow R$ se cunosc valorile $y_i = f(x_i)$, $i=0,1,\ldots,n$ în punctele distincte $a=x_0$, $x_1,\ldots,x_n=b$.

1) Sa se construiasca polinomul de interpolare Lagrange $L_n(x)$ ce aproximeaza functia data.
2) Sa se calculeze valoarea functiei $f(x)$ intr-un punct $x=\xi$ utilizind polinomul de interpolare Lagrange $L_n(x)$.
3) Sa se aproximeze valoarea functiei $f(x)$ pentru $x=\xi$ cu eroarea $\varepsilon=10^{-4}$ (sau cu cea mai buna exactitatea posibila), calculind polinomul de interpolare Lagrange $L_m(x)$ 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	x_0	x_1	…	x_n
y	y_0	y_1	…	y_n

unde $y_i=f(x_i)$ $i=0,1,2,\ldots n$

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

$P_n(x)=a_n x^n+a_{n-1}x^{n-1}+\ldots+a_1 x+a_0$

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 $P_n(x)$ de grad.$\leq$n să interpoleze funcţia dată, trebuie ca valorile sale în nodurile $x_0,x_1,\ldots x_n$ să coincidă cu valorile funcţiei, adică :

$P_n(x)=y_i$, $i=0,1,\ldots,n.$

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

$$L_n(x) = \sum_{i=0}^{n} \left(y_i \cdot \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \right)$$

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

Functia propusa spre rezolvare:

f(x) = $\frac{x+2}{x^2} - x + 2$

Tabelul de valori

x	0.150	0.163	0.187	0.262	0.364	0.466	0.469
y	6.616 59	8.394 68	9.278 936	8.193 45	6.234 76	6.251 32	6.314 72

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;
}
```

Rezultatele obtinute

```
options  compilation  execution
Indicati numarul de noduri de interpolare: 7
1.Functia este cunoscuta
2.Functia nu este cunoscuta
>>> 1
Introduceti elementele tabloului absciselor nodurilor
x0 = 0.150
x1 = 0.163
x2 = 0.187
x3 = 0.262
x4 = 0.364
x5 = 0.466
x6 = 0.469
Elementele tabloului valorilor functiei in aceste puncte
y0 = 3.95269
y1 = 3.94402
y2 = 3.92611
y3 = 3.8547
y4 = 3.72342
y5 = 3.56003
y6 = 3.55484
```

```
options  compilation  execution
Indicati numarul de noduri de interpolare: 7
1.Functia este cunoscuta
2.Functia nu este cunoscuta
>>> 2
Introduceti elementele tabloului absciselor nodurilor
x0 = 0.150
x1 = 0.163
x2 = 0.187
x3 = 0.262
x4 = 0.364
x5 = 0.466
x6 = 0.469
Elementele tabloului valorilor functiei in aceste puncte
y0 = 6.61659
y1 = 8.39468
y2 = 9.278936
y3 = 8.19345
y4 = 6.23476
y5 = 6.25132
y6 = 6.31472
Introduceti punctul in care doriti sa se efectueze interpolarea: x1 y1
 X   0.15   0.163   0.187   0.262   0.364   0.466   0.469
 Y  6.61659  8.39468  9.27894  8.19345  6.23476  6.25132  6.31472

 1.  Polinomul Lagrage

 Ln(x) = -653639*X^6 +1271597.0000*X^5 -1003877.7500*X^4 +411441.5000*X^3 -92337.7812*X^2 +10751.0762*X^1 -497.9603


 2.  Valoarea functiei

 f(0) = -497.96
Exit code: 0 (normal program termination)
```

Fig 1. Polinomul Lagrange

Concluzii

In cadrul acestei lucrari s-a elaborat programul ce construieste polinomul de interpolare Lagrange L_n (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.