Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică ăi Microelectronică

Departamentul Informatică Sofware și Automate

RAPORT

despre lucrarea de laborator nr. 3

la disciplina Metode și modele de calcul

Tema: Aproximarea funcțiilor

A efectuat: st. gr. TI-173 Heghea Nicolae

A verificat: conf. univ Tutunaru Eleonora

Cuprins

[1. Sarcina lucrării 3](#_Toc531829376)

[2. Noțiuni generale metoda Lagrange 3](#_Toc531829377)

[2.1 Schema bloc 4](#_Toc531829378)

[2.2 Codul Sursă 6](#_Toc531829379)

[2.3 Rezultate 7](#_Toc531829380)

[3. Concluzia 8](#_Toc531829381)

# Sarcina lucrării

Utilizând polinomul de interpolare, metoda Lagrange, pe noduri, , și vectorii , se introduc de la tastieră sau din fișier.

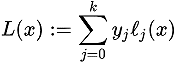
Să se calculeze valoare polinomului într-un punct dat, introdus de la tastieră. Afisarea polinomului și valoarea polinomului în punct.

# Noțiuni generale metoda Lagrange

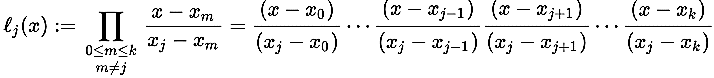
Fiind date un set de puncte :



atunci forma polinomului de interpolare Lagrange este :



unde :



Exemplu :

|  |  |  |  |
| --- | --- | --- | --- |
| x | 1 | 2 | 3 |
| y | 1 | 4 | 9 |

Polinomul de interpolare va fi :

|  |  |
| --- | --- |
|  |  |

## Schema bloc

Nu

Da

coef = CalcCoefLagrange(peNoduri);

valYinPuncte = CalcValoareaInPuncte();

Pseudocod funcției :

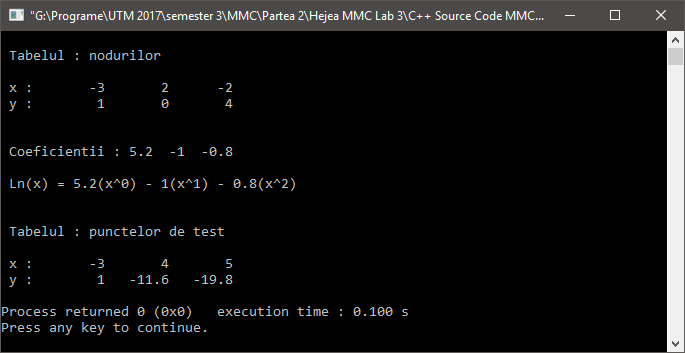
|  |
| --- |
| **penrtu** **:** // *ca i*  nou vectror [n];  ; // ca punct de start, cind nu sunt puncte.  ;  **pentru** **din** **:** // *ca j*  **if** **atunci** terci la umatorul;  impartitor \*= curpoint.x - point.x;  **pentru** **din** **:**  ;  ;  ;  **sfisit**;  **sfisit**; // *ca j*    **pentru** **din** **:**  ;  **sfisit**;    **return** finalCoef;  **sfisit**; |

## Codul Sursă

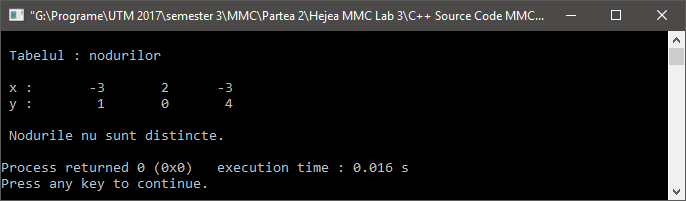
|  |
| --- |
| int main() {  citire();  if(verifica()) {  coefs = \*lagrange\_coeffs();  solvPoints();  cout << "Tabelul : nodurilor";  showTable(table);  showLN();  cout << "\n\nTabelul : punctelor de test";  showTable(points);  } else {  cout << "\n Nodurile nu sunt distincte.\n";  }  return 0;  } |
| vector<double>\* CalcCoefLagrange() {  auto len = table.size();  auto finalCoef = new vector<double> (len, 0);    for (auto curpoint : table) {  vector<double> tmpcoefs (len, 0);  tmpcoefs[0] = curpoint.y;  double impartitor = 1;    for(auto point : table) {  if (curpoint.x == point.x) continue;  impartitor \*= curpoint.x - point.x;  double precedent = 0;    for (auto resptr = tmpcoefs.begin(); resptr < tmpcoefs.end(); resptr++) {  double newres = (\*resptr) \* (-point.x) + precedent;  precedent = \*resptr;  \*resptr = newres;  }    }  transform(finalCoef->begin(), finalCoef->end(),  tmpcoefs.begin(),  finalCoef->begin(),  [=] (double oldcoef, double addcoef) {  return oldcoef + addcoef/impartitor;  }  );    }  return finalCoef;  } |

## Rezultate

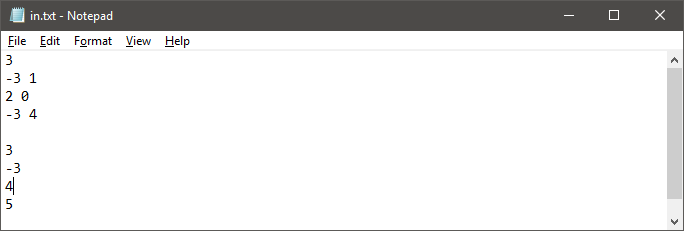
Când are soluții :



Citeva erori care pot să apară :



Fișierul de intrare :



# Concluzia

Forma Lagrange a polinomului de interpolare prezintă caracterul liniar al interpolării polinomiale și unicitatea polinomului de interpolare.

Dar, așa cum se poate observa din construcție, de fiecare dată când un nod se modifică, toate polinoamele bazate pe Lagrange trebuie să fie recalculate. O formă mai bună a polinomului de interpolare pentru scopuri practice (sau computaționale) este forma barycentrică a interpolației Lagrange sau polinomul de interpolare Newton.

Lagrange și altă interpolare la punctele egal distanțate, ca în exemplul de mai sus, dau un polinom oscilând deasupra și sub funcția adevărată. Acest comportament tinde să crească odată cu numărul de puncte.

Polinoamele de bază Lagrange pot fi utilizate în integrarea numerică pentru a deriva formulele Newton-Cotes.