**MINISTERUL EDUCAŢIEI, CULTURII ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**UNIVERSITATEA TEHNICĂ A MOLDOVEI**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică**

**RAPORT**

**la lucrarea de laborator nr.3**

**Tema: „Interpolarea funcțiilor cu ajutorul polinomului Langrange”**

**Disciplina: „Metode Numerice”**

**A elaborat *st. gr. SI-211, Vozian Vladimir***

**A verificat *lect.univ V.Struna***

**Chișinău 2022**

**Scopul lucrării**

Pentru funcţia f:[a, b]→R se cunosc valorile y0, y1, y2,…,yn în nodurile distincte x0, x1, x2,…, xn, adică yi=f(xi), i=0,1,2,…,n.

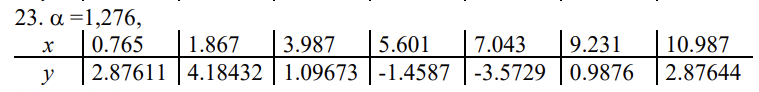
1) Să se construiască polinomul de interpolare Lagrange Ln(x) ce aproximează funcţia dată.

2) Să se calculeze valoarea funcţiei f(x) într-un punct x=α utilizând polinomul de interpolare Lagrange Ln(x).

3) Să se aproximeze valoarea funcţiei f(x) pentru x=α cu eroarea ε= 10-4 (sau cu cea mai bună exactitate posibilă), calculînd polinomul de interpolare Lagrange Lm(x), unde m<n.

4) Să se compare şi să se explice rezultatele obţinute în 2) şi 3).

***Varianta 23***



**Codul**

import sympy as sym

import matplotlib.pyplot as plt

import csv

eps = 10 \*\* -4

def export(rows):

    with open(f"var\_{var}.csv", 'w') as f:

        csv.writer(f).writerows(rows)

def print\_points(x,y):

    for x,y in zip(x,y):

        print(f'({x}, {y})')

def print\_function(f):

    print("Polinomul lagrange:")

    print(f"L{sym.degree(f)} = {str(f).replace('\*\*','^')}")

def lagrange(x, y):

    v = sym.Symbol('x')

    lagrange = 0

    for i in range(len(x)):

        Ln = y[i]

        for j in range(len(x)):

            if j == i:

                continue

            Ln \*= (v - x[j])/(x[i] - x[j])

        lagrange += Ln

    lagrange = sym.simplify(lagrange).evalf()

    return lagrange

def L\_n(x0, x1, y0, y1):

    return (y0 \* x1 - y1 \* x0)/(x1-x0)

def aitken(a, x, y):

    headers = ["Xi", "Yi", "Xi - a"]

    tabel = []

    for i in range(len(x)):

        line = [x[i], y[i], round(x[i] -a,5)]

        title = "L\_"

        for j in range(i,0,-1):

            if i == 0:

                continue

            title += f'i-{j}\_'

            line.append(0)

        if i != 0:

            headers.append(title+'i')

        tabel.append(line)

    val = 0

    error = 0

    for i in range(len(tabel)):

        for j in range(1, len(tabel[i])):

            try:

                x0, x1, y0, y1 = tabel[i-j+2][2], tabel[i][2], tabel[i-1][j-2], tabel[i][j-2]

            except:

                pass

            if j > 3:

                y0 = tabel[i-1][j-1]

                y1 = tabel[i][j-1]

            if tabel[i][j] == 0:

                val = L\_n(x0, x1, y0, y1)

                tabel[i][j] = round(val, 4)

            error = abs(tabel[i][j] - tabel[i-1][j-1])

            if error < eps:

                break

    print('\n'.join(['|'.join(['{:10}'.format(item) for item in row]).replace("\_",",") +"|" for row in [headers]+tabel]))

    export([headers]+tabel)

    print("error:", error)

    print(f'f({a})≈L({a})= {val}')

    return val

var = 23

a = 1.276

x = [0.765, 1.867, 3.987, 5.601, 7.043, 9.231, 10.987]

y = [2.87611, 4.18432, 1.09673, -1.4587, -3.5729, 0.9876, 2.87644]

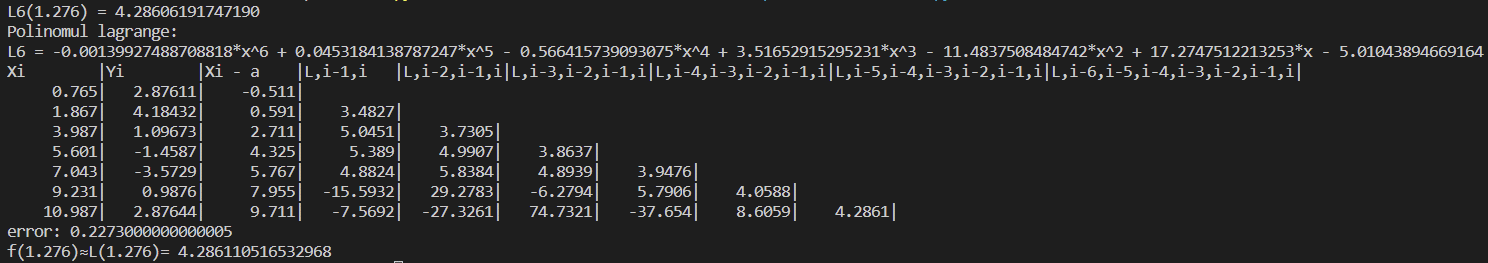
f = lagrange(x, y)

print(f"L{sym.degree(f)}({a}) = {f.subs('x', a)}")

print\_function(f)

aitken(a, x, y)

**Rezultatele**



**Concluzie :**

În urma efectuării acestei lucrări de laborator am realizat interpolarea funcției cu ajutorul polinomului Lagrange. Am efectuat funcții diferite cât pentru calculul însăși a polinomului, cât și aceea pentru efectuarea calculului funcției în punctul necesar cu exactitatea ε folosind schema Aitken. Ca rezultat am obținut varoarea de aproximativ 4.286 .