****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**Raport**

**pentru lucrarea de laborator Nr.4**

***la cursul de “Metode Numerice”***

Elaborat : Calancea Catalin  
Grupa: MI-222

Verificat:

Conferentiar

Buzurniuc Stefan

**Chișinău – 20****23**

**CUPRINS**

[INTRODUCERE 3](#_Toc151369371)

[SARCINA 4](#_Toc151369372)

[COD 4](#_Toc151369373)

[CONCLUZIA 11](#_Toc151369374)

# INTRODUCERE

# SARCINA

1. Sa se calculeze integrala definita

# 

# COD

#include <stdio.h>

#include <math.h>

// Definirea funcției integrată

double f(double x) {

return sin(x) / (x \* x);

}

// Formula trapezelor

double trapezeRule(double a, double b, int n) {

double h = (b - a) / n;

double sum = 0.5 \* (f(a) + f(b));

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

sum += f(a + i \* h);

}

return h \* sum;

}

// Formula Simpson

double simpsonRule(double a, double b, int n) {

double h = (b - a) / n;

double sum1 = 0, sum2 = 0;

for (int i = 1; i < n; i += 2) {

sum1 += f(a + i \* h);

}

for (int i = 2; i < n - 1; i += 2) {

sum2 += f(a + i \* h);

}

return (h / 3) \* (f(a) + 4 \* sum1 + 2 \* sum2 + f(b));

}

// Regula lui Runge pentru formula trapezelor

int rungeTrapezeRule(double a, double b, double e) {

int n = 2;

double integral\_prev, integral\_curr;

do {

integral\_prev = trapezeRule(a, b, n);

n \*= 2;

integral\_curr = trapezeRule(a, b, n);

} while (fabs(integral\_curr - integral\_prev) >= e);

return n;

}

// Regula lui Runge pentru formula Simpson

int rungeSimpsonRule(double a, double b, double epsilon) {

int m = 2;

double integral\_prev, integral\_curr;

do {

integral\_prev = simpsonRule(a, b, m);

m \*= 2;

integral\_curr = simpsonRule(a, b, m);

} while (fabs(integral\_curr - integral\_prev) >= epsilon);

return m;

}

int main() {

double a = 0.5;

double b = 1.5;

// Calculul integralei folosind formulele trapezelor și Simpson

double trapezeResult = trapezeRule(a, b, 1000);

double simpsonResult = simpsonRule(a, b, 1000);

// Aplicarea regulei lui Runge pentru trapeze cu o eroare mai mică decât 10

int nTrapeze = rungeTrapezeRule(a, b, 10);

// Aplicarea regulei lui Runge pentru Simpson cu o eroare mai mică decât 10^3

int nSimpson = rungeSimpsonRule(a, b, 103);

// Afișarea rezultatelor

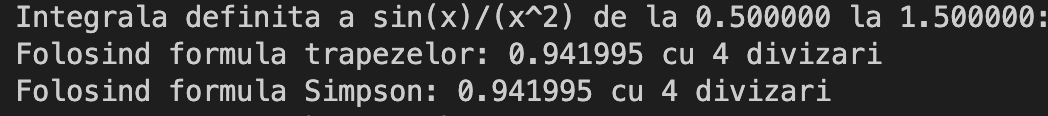
printf("Integrala definita a sin(x)/(x^2) de la %f la %f:\n", a, b);

printf("Folosind formula trapezelor: %lf cu %d divizari\n", trapezeResult, nTrapeze);

printf("Folosind formula Simpson: %lf cu %d divizari\n", simpsonResult, nSimpson)

return 0;

}



# CONCLUZIA

Prin intermediul acestei lucrări, am investigat și aplicat metoda interpolării funcțiilor, cu accent pe polinoamele de interpolare ale lui Lagrange. Construirea polinomului L n ( x ) L n ​ (x) ne-a permis să aproximăm funcția în punctul x = ξ x=ξ, având la dispoziție valorile cunoscute în n n puncte distincte. De asemenea, am extins analiza prin construirea polinomului L m ( x ) L m ​ (x) pentru m = n − 2 m=n−2, observând cum variază aproximarea funcției în același punct x = ξ x=ξ. Interpolarea funcțiilor rămâne un instrument puternic în analiza numerică, oferind soluții utile în aproximarea și predicția valorilor în contexte în care avem informații limitate. Această lucrare a adus în prim-plan abordarea și rezolvarea practică a problemei interpolării funcțiilor, evidențiind importanța și versatilitatea metodei lui Lagrange în acest context.