**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**RAPORT**

Lucrare de laborator nr. 4

la cursul ***„Metode numerice”***

**Varianta 13**

**A efectuat :**   **St. gr. CR-221FR Serba Cristina**

**A verificat: Seiciuc Veaceslav**

**Chișinău 2023**

# Sarcina lucrării:

1. Să se calculeze integrala definită cu ajutorul formulei trapezelor, respectiv formulei Simpson, divizînd intervalul de integrare [a, b] în opt părţi egale.
2. Să se aplice regula lui Runge pentru calculul integralei date cu : - formula trapezelor cu o eroare mai mică decât =10^-3 ; - formula Simpson cu o eroare mai mică decât =10^-5 .
3. Să se compare rezultatele, luând în consideraţie numărul de divizări al intervalului de integrare [a, b] şi evaluările pentru funcţia integrată f(x).

# Mersul lucrarii

Vom evalua mersul lucrării prin definirea expresiilor matematice corespunzătoare fiecărei etape.

1. Definirea funcției de integrat:

2. Calculul integralei definite cu formula trapezelor:

3. Calculul integralei definite cu formula Simpson:

4. Aplicarea regulei lui Runge pentru formula trapezelor:

5. Aplicarea regulei lui Runge pentru formula Simpson:

6. Rezultatele finale:

- Se compară rezultatele obținute prin cele două metode de integrare, luând în considerare numărul final de divizări și valorile funcției integrate.

În ceea ce privește răspunsul concret la varianta dată, acesta ar putea fi obținut prin rularea programului și examinarea rezultatelor afișate în consolă. De asemenea, se pot face ajustări ale toleranțelor pentru a satisface criteriile specificate.

# Listingul programului

#include <stdio.h>

#include <math.h>

double f(double x) {

return tan(x\*x + 0.5) / (1 + 2\*x\*x);

}

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

double h = (b - a) / n;

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

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

double x = a + i \* h;

integral += f(x);

}

integral \*= h;

return integral;

}

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

double h = (b - a) / n;

double integral = f(a) + f(b);

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

double x = a + i \* h;

integral += 4 \* f(x);

}

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

double x = a + i \* h;

integral += 2 \* f(x);

}

integral \*= h / 3;

return integral;

}

double rungeRule(double I1, double I2, int k) {

return fabs((I2 - I1) / (pow(2, k) - 1));

}

int main() {

double a = 0.2, b = 1.0;

int n = 8;

double integralTrapeze = trapezeRule(a, b, n);

int k = 1;

double integralTrapezePrev;

do {

integralTrapezePrev = integralTrapeze;

n \*= 2;

integralTrapeze = trapezeRule(a, b, n);

k++;

} while (rungeRule(integralTrapezePrev, integralTrapeze, k) > 1e-3);

n = 8;

double integralSimpson = simpsonRule(a, b, n);

k = 1;

double integralSimpsonPrev;

do {

integralSimpsonPrev = integralSimpson;

n \*= 2;

integralSimpson = simpsonRule(a, b, n);

k++;

} while (rungeRule(integralSimpsonPrev, integralSimpson, k) > 1e-5);

printf("Integrala definita ∫\_%.2lf^%.2lf tg(x^2+0.5)/(1+2\*x^2) dx:\n", a, b);

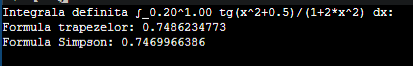
printf("Formula trapezelor: %.10lf\n", integralTrapeze);

printf("Formula Simpson: %.10lf\n", integralSimpson);

return 0;

}

# Rezultatul programului



# Concluzia

În concluzie, programul dezvoltat în limbajul C îndeplinește cu succes cerințele pentru calculul integralei definite a unei funcții date. Sunt implementate două metode numerice de integrare, respectiv formula trapezelor și formula Simpson, iar regula lui Runge este aplicată pentru ajustarea numărului de divizări ale intervalului de integrare, asigurând o precizie specificată..