Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра теоретичних основ радіотехніки

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №2**

з дисципліни: «Інформатика 1»

|  |  |
| --- | --- |
|  | Виконав : Пінчук Олександр Олександрович  Група: РЕ-12  Викладачі: доцент Катін П.Ю.  Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Підпис: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2021

**Мета роботи**: вивчити методи чисельного інтегрування; скласти програму обчислення визначеного інтегралу чисельними методами; дослідити залежність точності розрахунку інтегралу від кількості проміжків розбиття інтервалу інтегрування.

**Варіант**:14

**Обрана функція**: 

**Ключові моменти**:

1. Досліджуємо методи чисельного обчислення інтегралу.
2. Створюємо алгоритм розрахунку.
3. Для кожного методу робимо текстовий інтерфейс (опис роботи).
4. Розрахунки:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод | n=10 | n=100 | n=1000 | N |
| Прямокутники | 5.118382 | 5.118382 | 5.101155 | 148, 148,1006 |
| Трапеції | 5.122420 | 5.122420 | 5.091015 | 148, 148, 1002 |
| Сімсон | 5.145602 | 5.145602 | 5.092830 | 122, 122, 1002 |

Код:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

void info(double I,double Mistake,int n )//Вивід інфомації на консоль, Mistake - похибка

{

printf("I = %lf\n",I);

printf("Mistake= %lf\n",Mistake);

printf("Amount of steps = %d\n",n);

printf("its right\n\n");

}

double Calc(double x)//Функція розрахунку інтеграла

{

/\* Var -25 межі (0;3)

fun= sqrt(2\*pow(x,3))+sqrt(x)

\*/

return (sqrt(2\*pow(x,3))+sqrt(x));

}

double Rectangles(int n,double A, double B)//Функція розрахунку інтеграла за допомогою метода прямокутників

{

double h,I,x;

h=(B-A)/n;

for(x = A; x <= B ;x = x + h)

{

I=I+Calc(x);

}

I=I\*h;

return I;

}

double Trapezium (int n,double A, double B)//Функції розраховування інтегралу за допомогою метода трапецій

{

double h,I=0,x;//S=0 щоб рахувало з початку

h=(B-A)/n;

I=Calc(A)+Calc(B);

for(x = A; x <= B ;x = x + h)

{

I=I+Calc(x);

}

I=I\*h;

return I;

}

double Simpson(int n,double A, double B)//Функции розрахунку інтегралу за допомогою метода Сімсона

{

double h,I=0,x,II=0;//II - парний інтеграл

int i=0;

h=((B-A)/n);

for(i=0;i<=n;i++)

{

if(i%2 != 0)

{

I=I+Calc(A+h\*i);

}

if(i%2 == 0)

{

II=II+Calc(A+h\*i);

}

}

I=((B-A)/(3\*n))\*(Calc(A)+4\*I+2\*II+Calc(B));

return I;

}

int main()

{

/\*

14 Variant

Granitsy integrala ot 0 do 3

F(x)= (sqrt(2\*pow(x,3))+sqrt(x))

\*/

int var,n,i,j=1;

double h,d,x,I=0.0,II=0.0,A,B;

while(1)//нескінченний цикл

{

printf("Choose your method: \n");

printf("1 - Rectangles\n2 - Trapezium\n3 - Simpson\n4 - Usual\n5 - Own version(y = x^3)\n");

scanf("%d",&var);

switch("%d",var)

{

case 1://Rectangles

printf("Amount of steps:\n");

scanf("%d",&n);

printf("lower limit: \n");

scanf("%lf",&A);

printf("upper border: \n");

scanf("%lf",&B);

for (j;j<=2;j++)

{

if(j==1)

{

I=Rectangles(n,A,B);

n=n+2;

}

II=Rectangles(n,A,B);

if(fabs(I-II)>0/00001 && fabs(I-II)<0.001)//перевірка на допустимість похибки

{

II=fabs(I-II);

info(I,II,n);

break;

}

else

{

j=0;//у випадку невиконання умови починає цикл за початку

}

}

break;

case 2://Trapezium

printf("Amount of steps: \n");

scanf("%d",&n);

printf("lower limit: \n");

scanf("%lf",&A);

printf("upper border: \n");

scanf("%lf",&B);

for(j; j<=2;j++)

{

if(j==1)

{

I = Trapezium(n,A,B);

n=n+2;

}

II = Trapezium(n,A,B);

if(fabs(I-II)>0.00001 && fabs(I-II)<0.001)

{

II = fabs(I-II);

info(I,II,n);

break;

}

else

{

j=0;

}

}

break;

case 3://Simpson

do

{

printf("Number of intervals must be even \n");

printf("Amount of steps: \n");

scanf("%d",&n);

}

while(n%2!=0);

printf("lower limit: \n");

scanf("%lf",&A);

printf("upper border: \n");

scanf("%lf",&B);

for(j; j<=2;j++)

{

if(j==1)

{

I = Simpson(n,A,B);

n=n+2;

}

II = Simpson(n,A,B);

if(fabs(I-II)>0.00001 && fabs(I-II)<0.001)

{

II = fabs(I-II);

info(I,II,n);

break;

}

else

{

j=0;

}

}

break;

case 4://Usual

printf("lower limit: \n");

scanf("%lf",&A);

printf("upper border: \n");

scanf("%lf",&B);

I=((pow(A,2)\*pow(A,3))/6)+((2\*sqrt(A))/3);

II=((pow(B,2)\*pow(B,3))/6)+((2\*sqrt(B))/3);

I=II-I;//обрахунок інтеграла звичаним способом

printf("I : %lf\n\n",I);

break;

case 5://Demo

//y = x^3

printf("lower limit: \n");

scanf("%lf",&A);

printf("upper border: \n");

scanf("%lf",&B);

printf("My integral (0->4) y = x^3 \n");

printf("Count of steps: \n");

scanf("%d",&n);

printf("Calculator answer : 64\n");//порівнення відповіді калькулятора та інших трьох методів

//1

h = ((B-A)/n);

h=(B-A)/n;

for(x = A; x <= B ;x = x + h)

{

I=I+pow(x,3);

}

I=I\*h;

printf("Rectangles : %lf\n",I);

//2

h=(B-A)/n;

I=pow(A,3)+pow(B,3);

for(x = A; x <= B ;x = x + h)

{

I=I+pow(x,3);

}

I=I\*h;

printf("Trapezium : %lf\n",I);

//3

h=((B-A)/n);

for(i = 0; i <= n;i++)

{

if(i%2 != 0)

{

I=I+pow(A+h\*i,3);

}

if(i%2 == 0)

{

II=II+pow(A+h\*i,3);

}

}

I=((B-A)/(3\*n))\*(pow(A,2)+4\*I+2\*II+pow(B,2));

printf("Simpson : %lf\n\n",I);

break;

default:

printf("No one variant\n");//якщо користувач не вказав жодного з варіантів

break;

}

}

return 0;

}



