Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра теоретичних основ радіотехніки

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №2**

з дисципліни: «Інформатика 1»

|  |  |
| --- | --- |
|  | Виконав: Філобоченко Ліза  Група: РЕ-11  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.

**Задана функція:** інтеграл від А до В **pow(x,3)/(pow(x,8)+1).**

**Ключові моменти:  
1.** Задаємо функцію

void Dump(double S, double Poxb, int n)

{

//S сума аля ответ интеграла

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

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

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

}

**2.** Росписуємо 3 функції double для трьох різних варіантів вирішення інтегралу.

**3.** В int main () задаємо змінні

int varik, n, i, j=1;

double h, d, x, S=0.0, SS=0.0, A, B;

**4.** Використовуємо безкінечний цикл while(1) в якому розташовано

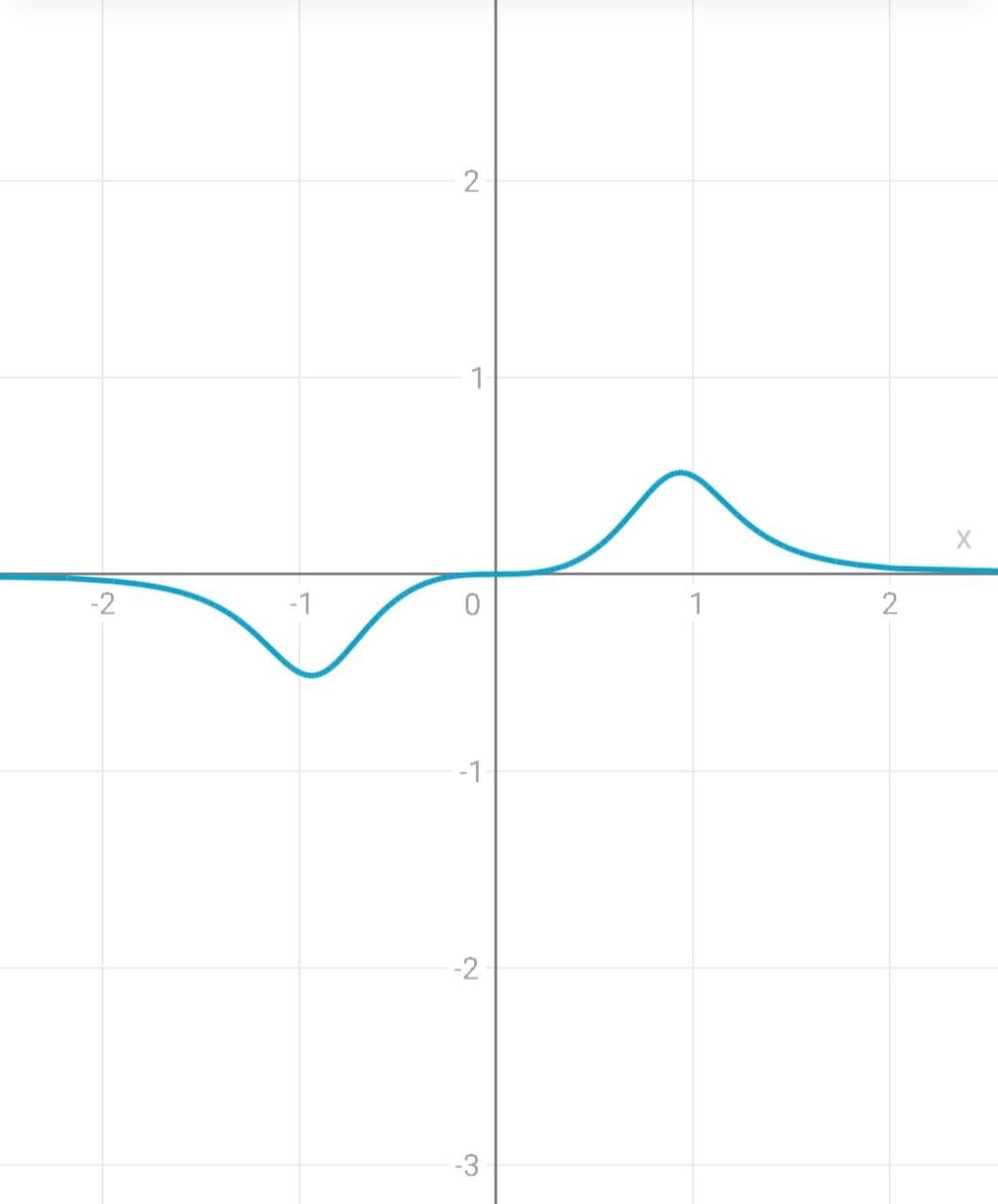
printf("Choise one method: \n");

printf("1 - Method of rectangles\n2 - Method of trapezoidal\n3 - Simpson method\n4 - Usual method\n5 - Demo version(y = 2x)\n");

scanf("%d",&varik);

Та switch("%d",varik)

**5.** Графік моєї функції **pow(x,3)/(pow(x,8)+1).**



**Код:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

void Dump(double S, double Poxb, int n) //вывод информации на консоль после просьбы

{

//S сума аля ответ интеграла

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

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

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

}

double Funk(double x) // Функция расчета интеграла ,возвращает результат интеграла

{

/\* Вариант 25

интеграл от 0 до 1 pow(x,3)/(pow(x,8)+1)

\*/

return(pow(x,3)/(pow(x,8)+1));

}

double Rectangles(int n,double A, double B) //метод интеграла с помощью прямоугольника

{

double h, S, x;

h=(B-A)/n;

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

{

S=S+Funk(x);

}

S=S\*h;

return S;

}

double Trapezia(int n, double A, double B) // метод трапеции

{

double h, S=0, x;

h=(B-A)/n;

S=Funk(A)+Funk(B);

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

{

S=S+Funk(x);

}

S=S\*h;

return S;

}

double Sims(int n, double A, double B) // способ симсона

{

// S для непарных SS для парных

double h, S=0, SS=0;

int i=0;

h=(B-A)/n;

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

{

if (i%2 !=0)

{

S=S+Funk(A+h\*i);

}

if (i%2 ==0)

{

SS=SS+Funk(A+h\*i);

}

}

S=((B-A)/(3\*n))\*(Funk(A)+4\*S+2\*SS+Funk(B));

return S;

}

/\* Вариант 25

интеграл от 0 до 1 (pow(x,3)/(pow(x,8)+1))

\*/

int main ()

{

int varik, n, i, j=1;

double h, d, x, S=0.0, SS=0.0, A, B;

while(1) //бессконечный цыкл когда в скобках 1, если там 0 то цыкл не повторяется вообще

{

printf("Choise one method: \n");

printf("1 - Method of rectangles\n2 - Method of trapezoidal\n3 - Simpson method\n4 - Usual method\n5 - Demo version(y = 2x)\n");

scanf("%d",&varik);

switch("%d",varik)

{

case 1: // метод прямоугольника

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

scanf("%d", &n);

printf("Write limit: \n");

scanf("%lf",&A);

printf("upper border: \n");

scanf("%lf",&B);

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

{

if(j==1)

{

S=Rectangles(n,A,B);

n=n+2;

}

SS=Rectangles(n,A,B);

if(fabs(S-SS)>0.00001 && fabs(S-SS)<0.001) //позволяет ли на ошибка fabs=модуль

{

SS=fabs(S-SS);

Dump(S, SS, n);

break;

}

else

{

j=0; //если цыкл фигня то он начинается с начала

}

}

break;

case 2: // метод трапеции

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

scanf("%d", &n);

printf("Write limit: \n");

scanf("%lf",&A);

printf("upper border: \n");

scanf("%lf",&B);

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

{

if(j==1)

{

S=Trapezia(n,A,B);

n=n+2;

}

SS=Trapezia(n,A,B);

if(fabs(S-SS)>0.00001 && fabs(S-SS)<0.001)

{

SS=fabs(S-SS);

Dump(S, SS, n);

break;

}

else

{

j=0;

}

}

break;

case 3: //метод симсона

do

{

printf("This method must have an even number of intervals \n");

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

scanf("%d",&n);

}

while(n%2!=0);

printf("Write limit: \n");

scanf("%lf",&A);

printf("upper border: \n");

scanf("%lf",&B);

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

{

if(j==1)

{

S = Sims(n,A,B);

n=n+2;

}

SS = Sims(n,A,B);

if(fabs(S-SS)>0.00001 && fabs(S-SS)<0.001)

{

SS = fabs(S-SS);

Dump(S,SS,n);

break;

}

else

{

j=0;

}

}

break;

case 4: //обычный метод ,функцию мы перетворили

printf("Write limit: \n");

scanf("%lf",&A);

printf("upper border: \n");

scanf("%lf",&B);

S=(atan(pow(A,4))/4);

SS=(atan(pow(B,4))/4);

S=SS-S; //формула рассчёта

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

break;

case 5: //Demo version(y = 2x)

printf("My integral is (0->1)y=2x \n");

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

scanf("%d",&n);

printf("Write limit: \n");

scanf("%lf",&A);

printf("upper border: \n");

scanf("%lf",&B);

printf("Answer =1 if limit is 0->1\n"); // сейчас буду сопоставлять ответы калькулятора с ответами по трем методам

// 1 met

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

h=(B-A)/n;

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

{

S=S+(2\*x);

}

S=S\*h;

printf("Method of rectangles: %lf \n",S);

// 2 met

h=(B-A)/n;

S=(2\*A)+(2\*B);

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

{

S=S+(2\*x);

}

S=S\*h;

printf("Method of trapezoidal: %lf \n",S);

// 3 met

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

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

{

if(i%2 !=0)

{

S=S+(2\*(A+h\*i));

}

if(i%2 ==0)

{

SS=SS+((A+h\*i)\*2);

}

}

S=((B-A)/(3\*n))\*((2\*A)+4\*S+2\*SS+(2\*B));

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

break;

default:

printf("No of variants \n");// выбрал то что не подходит

break;

}

}

return 0;

}