Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ**

**«ОБЧИСЛЕННЯ ВИЗНАЧЕНОГО ІНТЕГРАЛУ»**

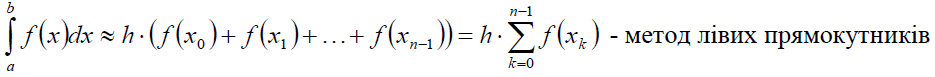
з дисципліни: «Інформатика»

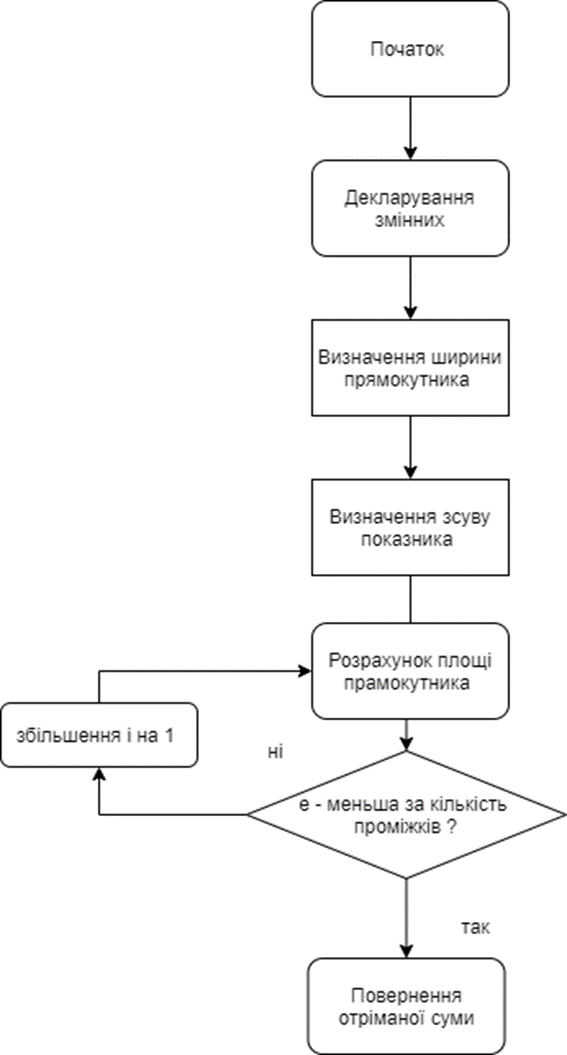
|  |  |
| --- | --- |
|  | Виконав: Черницький Антон  Група: РЕ-11  Викладачі:  Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2021

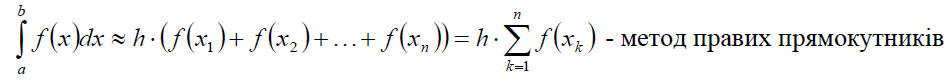
Мета роботи: навчитися використовувати цикли для вирішення інтегралів , виводити значення інтегралу.

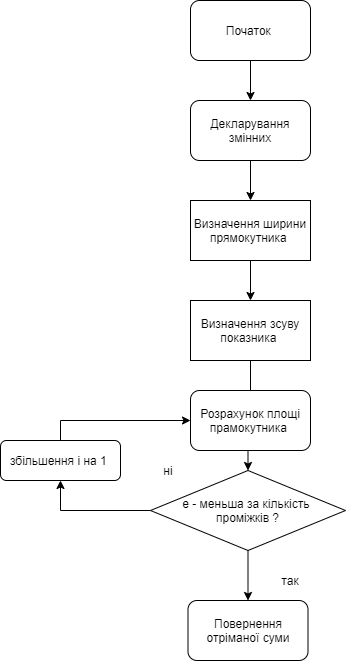
Блок-схеми кожного методу:

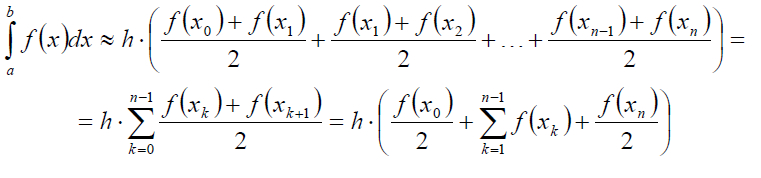
1. Метод лівих прямокутників: 

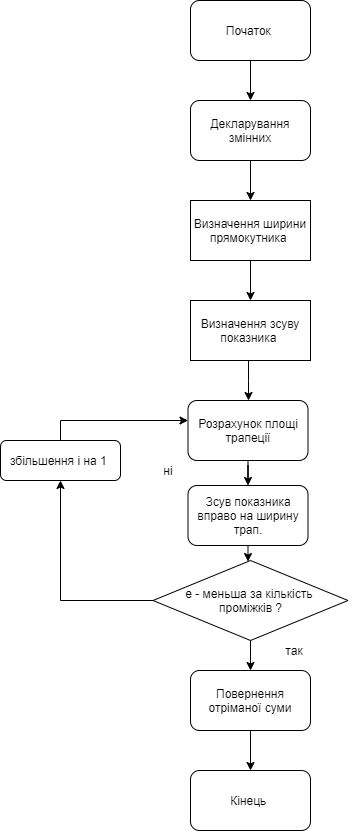


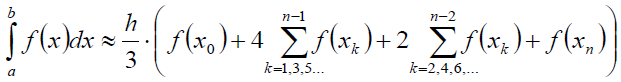
1. Метод правих прямокутників:

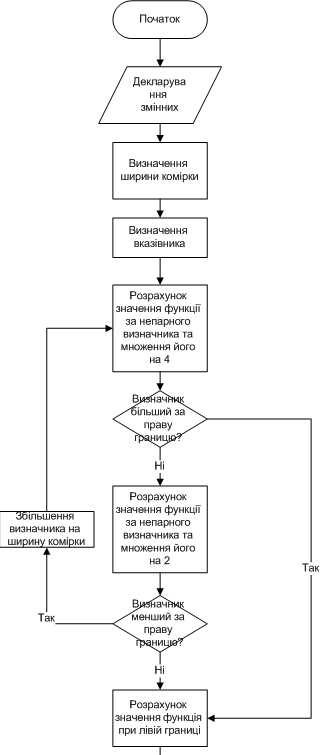


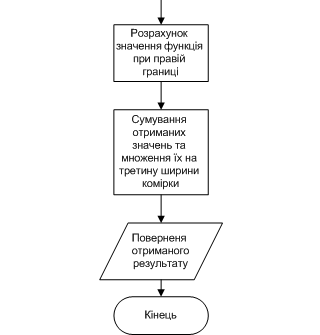


1. Метод трапецій: 



1. Метод Сімпсона (метод парабол): 





Обрана функція:

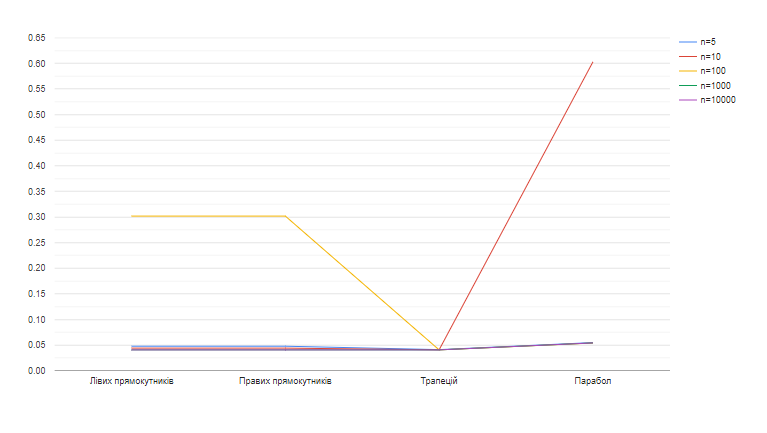
Обрахований аналітично інтеграл в символьному виді: 1/9+ln(134/185)+17/222

Чисельне значення аналітично обрахованого інтеграла: 0,0407415

Таблиця з результатами обрахунків заданого інтеграла всіма методами при 4-5 кількостях проміжків:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Метод | n=5 | n=10 | n=100 | n=1000 | n=10000 |
| Лівих прямокутників | 0.04788996 | 0.04419092 | 0.30187574 | 0.04077471 | 0.04074479 |
| Правих прямокутників | 0.04788996 | 0.04419092 | 0.30187574 | 0.04077471 | 0.04074479 |
| Трапецій | 0.04124336 | 0.04086762 | 0.040742730 | 0.04074148 | 0.04074146 |
| Парабол | 0.05500396 | 0.60272012 | 0.05410208 | 0.054324510 | 0.05432221 |

Графік залежності значення інтеграла від кількості проміжків:



Висновки щодо точності та швидкодії (кількості ітерацій) кожного з методів:

Найближче значення до істинного найшвидше отримало методом трапецій,інші ж наблизились тільки за великої кількості ітерацій

Код програми:

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <math.h>

#define eps 0.000000001

unsigned int method\_select( void );

double f( double x );

void result(double a, double b, unsigned int n, double I );

double left\_rectangle( double a, double b, unsigned int n );

double right\_rectangle( double a, double b, unsigned int n );

double trapez( double a, double b, unsigned int n );

double simpson( double a, double b, unsigned int n );

int main()

{

unsigned int methodn; // 1 - left, 2 - right, 3 - trapez, 4 - Simpson

const double a = 3;

const double b = 4.7;

unsigned int n = 10000;

double I;

double I1, I2;

unsigned int N;

methodn = method\_select();

if (methodn == 1 ){

I = left\_rectangle(a, b, n);

result(a, b, n, I);

N = 0;

do{

N = N + 2;

I1 = left\_rectangle(a, b, N);

I2 = left\_rectangle(a, b, N+2);

}while( fabs(I2-I1) > eps );

printf("\n\nN=%u, I1(N)=%.8lf\n", N, I1);

}else if( methodn == 2 ){

I=right\_rectangle(a, b, n);

result(a, b, n, I);

N = 0;

do{

N = N + 2;

I1 = left\_rectangle(a, b, N);

I2 = left\_rectangle(a, b, N+2);

}while( fabs(I2-I1) > eps );

printf("\n\nN=%u, I1(N)=%.8lf\n", N, I1);

}else if( methodn == 3 ){

I=trapez(a, b, n);

result(a, b, n, I);

N= 0;

do{

N = N + 2;

I1 = left\_rectangle(a, b, N);

I2 = left\_rectangle(a, b, N+2);

}while( fabs(I2-I1) > eps );

printf("\n\nN=%u, I1(N)=%.8lf\n", N, I1);

}

else

{

I=simpson(a, b, n);

result(a, b, n, I);

N= 0;

do{

N = N + 2;

I1 = left\_rectangle(a, b, N);

I2 = left\_rectangle(a, b, N+2);

}while( fabs(I2-I1) > eps );

printf("\n\nN=%u, I1(N)=%.8lf\n", N, I1);

}

return 0;

}

unsigned int method\_select(void)

{

unsigned int temp;

printf("\n\n Choose your method:\n1 - Left rectangle \n2 - Right rectangle \n3 - Trapezoid \n4 - Simpson method\n>");

scanf("%u", &temp);

while( temp < 1 || temp > 4 ){

printf("\n!!!!Invalid data. Method: 1, 2, 3 or 4: ");

scanf("%u", &temp);

}

switch(temp){

case 1:

printf("\nYou chose method of LEFT RECTANGLE");

break;

case 2:

printf("\nYou chose method of RIGHT RECTANGLE");

break;

case 3:

printf("\nYou chose method of TRAPEZOID");

break;

case 4:

printf("\nYou chose method of SIMPSON");

break;

}

return temp;

}

double f( double x )

{

double y;

y = 1/(pow(x,3)-3\*x+2);

return y;

}

double left\_rectangle(double a, double b, unsigned int n)

{

double h;

unsigned int k;

double x;

double sum = 0;

h = ( b - a ) / n;

x = a;

for (k = 0; k <= (n-1); k++ ){

sum = sum + f(x);

x = x + h;

}

return sum \* h;

}

double right\_rectangle(double a, double b, unsigned int n)

{

double h;

unsigned int k;

double x;

double sum = 0;

h = ( b - a ) / n;

x = a;

for (k = 1; k <= n; k++ ){

sum = sum + f(x);

x = x + h;

}

return sum \* h;

}

double trapez(double a, double b, unsigned int n)

{

double h;

unsigned int k;

double x;

double sum = 0;

h = ( b - a ) / n;

x = a;

for (k = 0; k <= n-1; k++ ){

sum = sum + (f(x)+f(x + h))/2;

x = x + h;

}

return sum \* h;

}

//-------------------------------------------------------------------

double simpson(double a, double b, unsigned int n)

{

double h;

double x;

double I=0;

h = (b - a) / n;

x = a + h;

while (x < b)

{

I += 4\*f(x);

x += h;

if (x >= b)

{

I += 2\*f(x);

x += h;

}

}

I = (h / 3) \* (I + f(a) + f(b));

return I;

return I \* h;

}

void result(double a, double b, unsigned int n, double I)

{

system("cls");

printf("\n\* Results \*");

printf("a=%.2lf b=%.2lf n=%u I = %.8lf", a, b, n, I);

}