Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности электронно–вычислительных систем (КИБЭВС)

ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

Отчет по лабораторной работе №4

по дисциплине «Численные методы»

Выполнил

Студент гр. 728–2

\_\_\_\_\_\_\_ Д. Р. Геворгян

\_\_\_.12.2019

Принял

Доцент кафедры КИБЭВС

\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ С. С. Катаева  
 оценка подпись \_\_\_.12.2019

**1 Введение**

Необходимо для интеграла рассчитать значение определённого интеграла следующими методами: методом левых прямоугольников, методом правых прямоугольников, методом трапеций, методом Симпсона, методом «трёх восьмых». За числа разбиений были взяты следующие значения: 12, 48, 144, 600.

**2 Ход работы**

**2.1 Метод прямоугольников**

Интервал интегрирования [a,b] с помощью точек , разбили на n элементарных отрезков. Если , а участки разбиения равны, то формула примет вид формулы 2.1 для метода левых прямоугольников и вид формулы 2.2 для метода правых прямоугольников.

|  |  |
| --- | --- |
|  | (2.1) |
|  | (2.2) |

**2.2 Метод трапеций**

Геометрический смысл определённого интеграла – площадь криволинейной трапеции. В этом методе площадь под кривой заменяется площадью элементарной трапеции. Будем интегрировать подынтегральную функцию полиномом Лагранжа первого порядка. Конечная формула примет вид:

|  |  |
| --- | --- |
|  | (2.3) |

**2.3 Метод Симпсона**

Если *f(x)* интерполировать полиномом Лагранжа второй степени (на сдвоенном отрезке [𝑥0, 𝑥2]) тогда, проведя аналогичные выкладки получим:

|  |  |
| --- | --- |
|  | (2.4) |

При использовании метода Симпсона [a,b] делят на чётное число частей, длиной , при этом n – чётное.

**2.4 Метод «трёх восьмых»**

Проинтерполируем подынтегральную функцию *f(x)* полиномом Лагранжа 3 степени, для чего отрезок [a,b] разделим на n частей, длина каждой части , где n – число, кратное трём. Получаем форму метода Ньютона (трёх восьмых):

|  |  |
| --- | --- |
|  | (2.5) |

**2.5 Итоги нахождения значений интеграла**

Результаты нахождения значения интеграла, полученные разными методами при разных разбиениях приведены в таблице 2.1.

Таблица 2.1 – Результаты нахождения интеграла разными способами

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | n=12 | n=48 | n=144 | n=600 |
| Метод левых прямоугольников | 0.690342 | 0.648066 | 0.638349 | 0.634625 |
| Метод правых прямоугольников | 0.57233 | 0.618563 | 0.628514 | 0.632265 |
| Метод трапеций | 0.631336 | 0.633315 | 0.633432 | 0.633445 |
| Метод Симпсона | 0.633468 | 0.633446 | 0.633446 | 0.633446 |
| Метод трёх восьмых | 0.633495 | 0.633446 | 0.633446 | 0.633446 |

Из таблицы видно, что наиболее точные способы – метод Симпсона и метод трёх восьмых, так как уже на малом числе разбиений они дают близкие к точному результату значения, в то время как метод прямоугольников даёт достаточно неточное значение, а метод трапеций выдаёт ожидаемы результат только при большом количестве разбиений.

Примеры работы программы представлены на рисунках 2.1–2.4.

Код программы представлен в Приложении А.

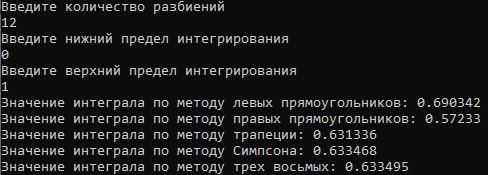


Рисунок 2.1 – Пример работы программы при n=12

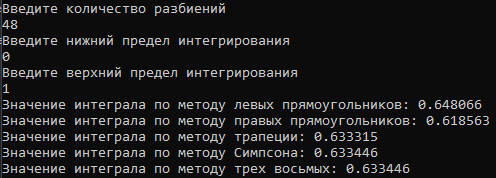


Рисунок 2.2 – Пример работы программы при n=48

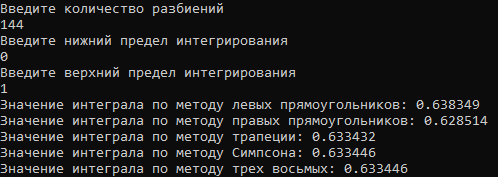


Рисунок 2.3 – Пример работы программы при n=144

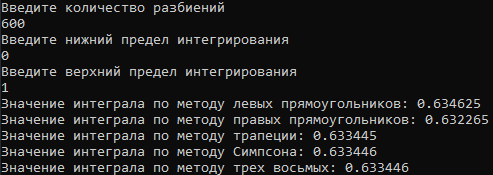


Рисунок 2.4 – Пример работы программы при n=600

**3 Заключение**

В процессе выполнения лабораторной работы была разработана программа, рассчитывающая значение определённого интеграла следующими методами: методом левых прямоугольников, методом правых прямоугольников, методом трапеций, методом Симпсона, методом «трёх восьмых».

Отчет был написан согласно ОС ТУСУР.

**Приложение А**

**(обязательное)**

Код программы:

#include <iostream>

using namespace std;

double function(double x)

{

return cos(x + pow(x, 3));

}

int main()

{

setlocale(LC\_ALL, "Russian");

cout << "Введите количество разбиений" << endl;

double n;

cin >> n;

cout <<"Введите нижний предел интегрирования" <<endl;

double a;

cin >> a;

cout<< "Введите верхний предел интегрирования" << endl;

double b;

cin >> b;

double h = (b - a) / n;

double Sumlp = 0;

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

{

Sumlp += function(double(i) \* h + a);

}

Sumlp \*= h;

cout << "Значение интеграла по методу левых прямоугольников: " << Sumlp<<endl;

double Sumpp = 0;

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

{

Sumpp += function(double(i) \* h + a);

}

Sumpp \*= h;

cout << "Значение интеграла по методу правых прямоугольников: " << Sumpp<<endl;

double Sumtr = 0;

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

{

Sumtr += 2\*function(double(i) \* h + a);

}

Sumtr += function(a)+function(b);

Sumtr \*= h/2;

cout << "Значение интеграла по методу трапеции: " << Sumtr<<endl;

double SumSimon = 0;

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

{

if (i%2==0)

SumSimon += 2 \* function(double(i) \* h + a);

else

SumSimon += 4 \* function(double(i) \* h + a);

}

SumSimon += function(a) + function(b);

SumSimon \*= h/3;

cout << "Значение интеграла по методу Симпсона: " << SumSimon<<endl;

double Sum38 = 0;

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

{

if (i % 3 == 0)

Sum38 += 2 \* function(double(i) \* h + a);

else

Sum38 += 3 \* function(double(i) \* h + a);

}

Sum38 += function(a) + function(b);

Sum38 \*= h\*3/8;

cout << "Значение интеграла по методу трех восьмых: " << Sum38 << endl;

system("pause");

return 0;

}