Университет ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Вычислительная математика»

**Отчет**

По лабораторной работе №3

Вариант 6

Выполнил:

*Манжиков Н.C*

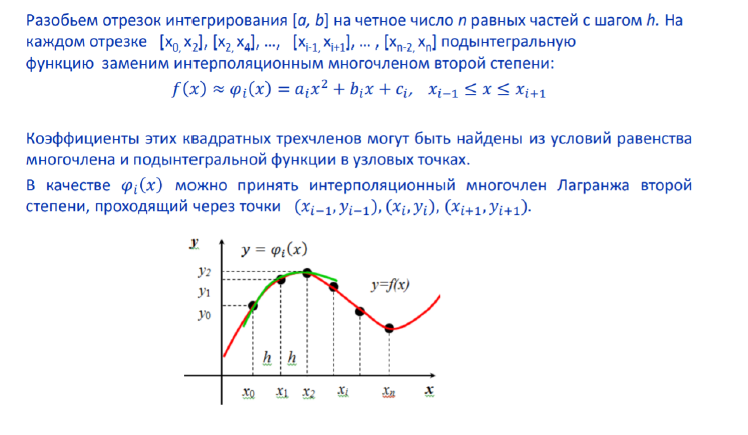
Преподаватель: Рыбаков Степан Дмитриевич

Санкт-Петербург, 2023 г.

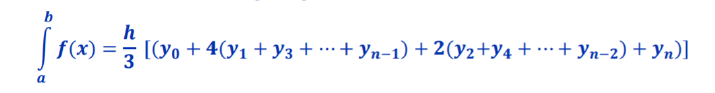
Цель работы

Решение интегралов методом Симпсона, методом трапеций и методом прямоугольников.

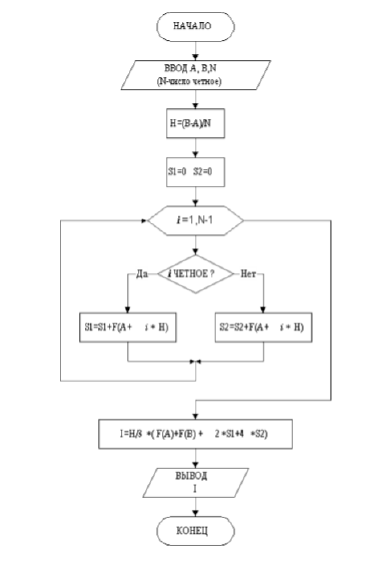
Метод Симпсона



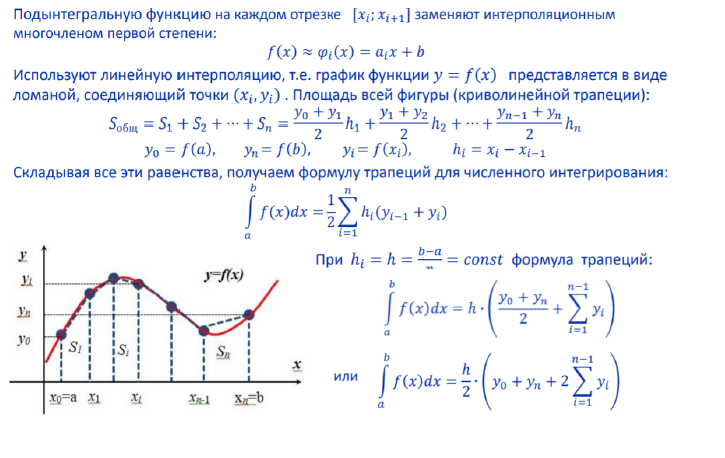
Рабочая формула:

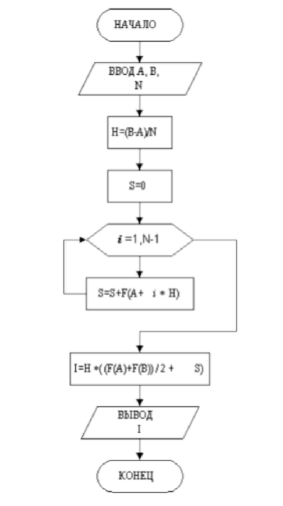


Блок схема метода хорд:

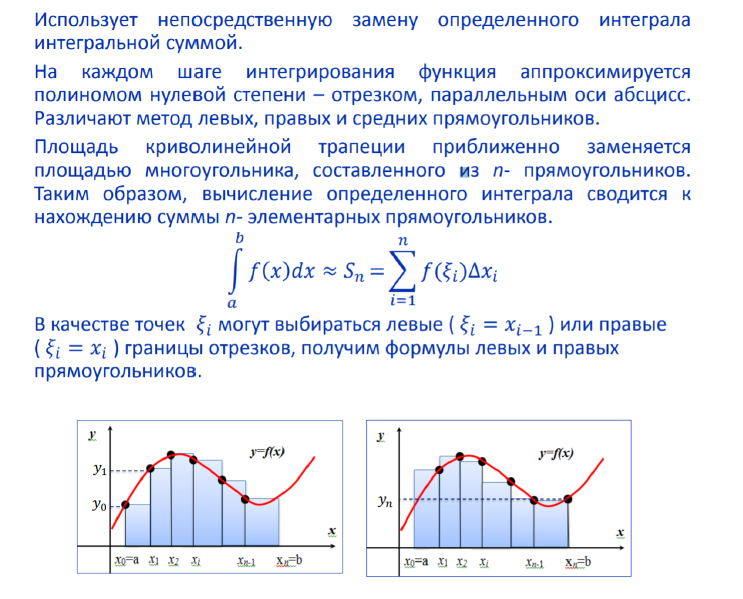
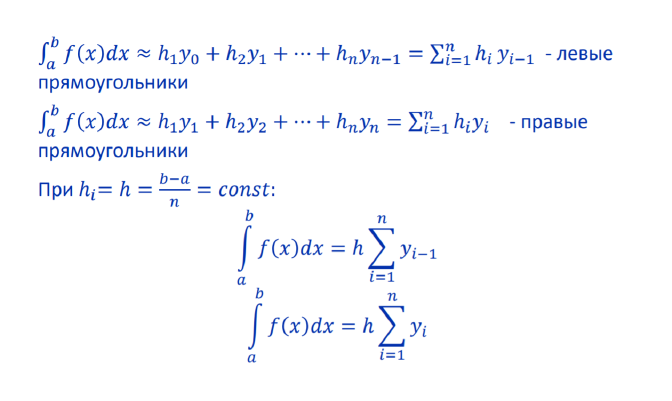


Метод трапеций



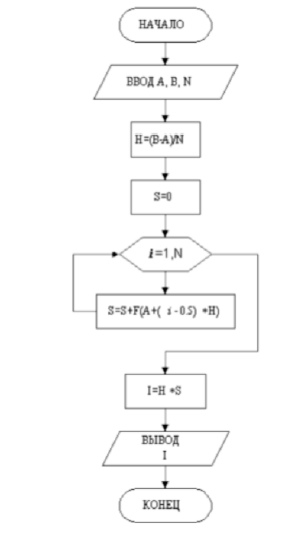
Блок схема метода секущих

Метод прямоугольников

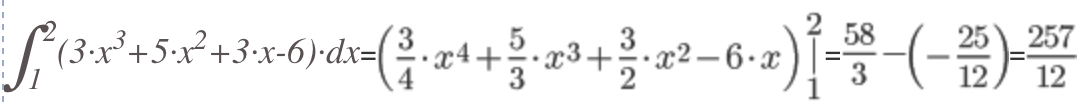


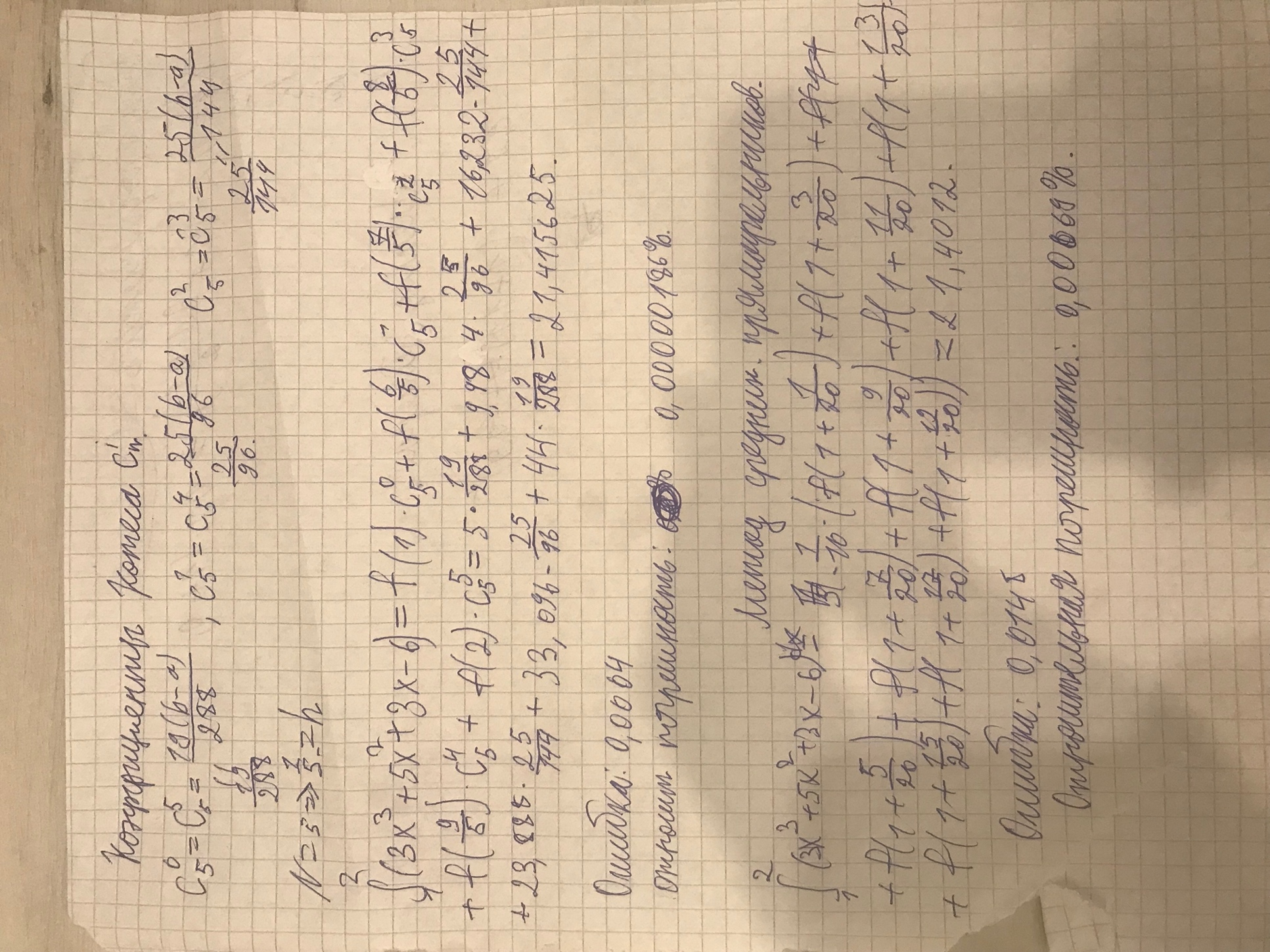
Метод средних прямоугольников

Блок схема алгоритма



**Вычислительная реализация**





Метод Трапеций

|  |  |  |
| --- | --- | --- |
| i | xi | yi |
| 0 | 1 | 5 |
| 1 | 1.1 | 7.343 |
| 2 | 1.2 | 9.984 |
| 3 | 1.3 | 12.941 |
| 4 | 1.4 | 16.232 |
| 5 | 1.5 | 19.875 |
| 6 | 1.6 | 23.888 |
| 7 | 1.7 | 28.289 |
| 8 | 1.8 | 33.096 |
| 9 | 1.9 | 38.327 |
| 10 | 2 | 44 |

=

Ошибка:0.032

Относительная погрешность: 0.01491%

**Формула Симпсона**:

|  |  |  |
| --- | --- | --- |
| i | xi | yi |
| 0 | 1 | 5 |
| 1 | 1.1 | 7.343 |
| 2 | 1.2 | 9.984 |
| 3 | 1.3 | 12.941 |
| 4 | 1.4 | 16.232 |
| 5 | 1.5 | 19.875 |
| 6 | 1.6 | 23.888 |
| 7 | 1.7 | 28.289 |
| 8 | 1.8 | 33.096 |
| 9 | 1.9 | 38.327 |
| 10 | 2 | 44 |

=

Ошибка:0.001

Относительная погрешность:0,00004669%

Пример работы программы

Введите левую границу интегрирования a: 1

Введите правую границу интегрирования b: 2

Введите количество разбиений n: 4

Точное значение интеграла: 21.416000

Метод прямоугольников (левые):

Значение интеграла: 19.027344

Число разбиений: 8

Относительная погрешность: 2.292969

Метод прямоугольников (правые):

Значение интеграла: 23.902344

Число разбиений: 8

Относительная погрешность: 2.582031

Метод прямоугольников (средние):

Значение интеграла: 21.392578

Число разбиений: 8

Относительная погрешность: 0.072266

Метод трапеций:

Значение интеграла: 21.464844

Число разбиений: 8

Относительная погрешность: 0.048177

Метод Симпсона:

Значение интеграла: 21.416667

Число разбиений: 8

Относительная погрешность: 0.000000

Код программы:

#include <stdio.h>  
#include <math.h>  
  
// Функция, для которой нужно вычислить интеграл  
double myFunction(double x) {  
 return (3 \* x \* x \* x) + (5 \* x \* x) + (3 \* x) - 6;  
}  
  
// Метод трапеций  
double trapezoidalMethod(double a, double b, int n) {  
 double h = (b - a) / n;  
 double integralSum = (myFunction(a) + myFunction(b)) / 2;  
 for (int i = 1; i < n; i++) {  
 double x = a + i \* h;  
 integralSum += myFunction(x);  
 }  
 double integralValue = h \* integralSum;  
 return integralValue;  
}  
// Оценка погрешности с использованием правила Рунге  
double rungeRule(double result1, double result2, double p) {  
 return fabs(result1 - result2) / (pow(2, p) - 1);  
}  
  
// Метод Симпсона  
double simpsonMethod(double a, double b, int n) {  
 double h = (b - a) / n;  
 double integralSum = myFunction(a) + myFunction(b);  
 for (int i = 1; i < n; i++) {  
 double x = a + i \* h;  
 if (i % 2 == 0) {  
 integralSum += 2 \* myFunction(x);  
 } else {  
 integralSum += 4 \* myFunction(x);  
 }  
 }  
 double integralValue = h \* integralSum / 3;  
 return integralValue;  
}  
// Метод прямоугольников (правые)  
double rightRectangleMethod(double a, double b, int n) {  
 double h = (b - a) / n; // Шаг разбиения  
 double sum = 0.0;  
  
 for (int i = 1; i <= n; i++) {  
 double x = a + i \* h;  
 double fx = myFunction(x); // Значение функции в точке x  
 sum += fx;  
 }  
  
 double result = h \* sum;  
 return result;  
}  
  
// Метод прямоугольников (средние)  
double midpointRectangleMethod(double a, double b, int n) {  
 double h = (b - a) / n; // Шаг разбиения  
 double sum = 0.0;  
  
 for (int i = 0; i < n; i++) {  
 double x = a + (i + 0.5) \* h;  
 double fx = myFunction(x); // Значение функции в точке x  
 sum += fx;  
 }  
  
 double result = h \* sum;  
 return result;  
}  
  
// Метод прямоугольников (левые прямоугольники)  
double leftRectangleMethod(double a, double b, int n) {  
 double h = (b - a) / n;  
 double integralSum = 0;  
 for (int i = 0; i < n; i++) {  
 double x = a + i \* h;  
 integralSum += myFunction(x);  
 }  
 double integralValue = h \* integralSum;  
 return integralValue;  
}  
  
int main() {  
 double a, b;  
 int n;  
  
 printf("Введите левую границу интегрирования a: ");  
 scanf("%lf", &a);  
  
 printf("Введите правую границу интегрирования b: ");  
 scanf("%lf", &b);  
  
 printf("Введите количество разбиений n: ");  
 scanf("%d", &n);  
  
 // Вычисление интеграла точно  
 double exactIntegral = 21.416;  
 printf("Точное значение интеграла: %lf\n", exactIntegral);  
  
 // Метод прямоугольников (левые)  
 printf("Метод прямоугольников (левые):\n");  
 double result1 = leftRectangleMethod(a, b, n);  
 double result2 = leftRectangleMethod(a, b, 2 \* n);  
 double error1 = rungeRule(result1, result2, 1);  
 printf("Значение интеграла: %lf\n", result2);  
 printf("Число разбиений: %d\n", 2 \* n);  
 printf("Относительная погрешность: %lf\n\n", error1);  
  
 // Метод прямоугольников (правые)  
 printf("Метод прямоугольников (правые):\n");  
 result1 = rightRectangleMethod(a, b, n);  
 result2 = rightRectangleMethod(a, b, 2 \* n);  
 double error2 = rungeRule(result1, result2, 1);  
 printf("Значение интеграла: %lf\n", result2);  
 printf("Число разбиений: %d\n", 2 \* n);  
 printf("Относительная погрешность: %lf\n\n", error2);  
  
 // Метод прямоугольников (средние)  
 printf("Метод прямоугольников (средние):\n");  
 result1 = midpointRectangleMethod(a, b, n);  
 result2 = midpointRectangleMethod(a, b, 2 \* n);  
 double error3 = rungeRule(result1, result2, 1);  
 printf("Значение интеграла: %lf\n", result2);  
 printf("Число разбиений: %d\n", 2 \* n);  
 printf("Относительная погрешность: %lf\n\n", error3);  
  
 // Метод трапеций  
 printf("Метод трапеций:\n");  
 result1 = trapezoidalMethod(a, b, n);  
 result2 = trapezoidalMethod(a, b, 2 \* n);  
 double error4 = rungeRule(result1, result2, 2);  
 printf("Значение интеграла: %lf\n", result2);  
 printf("Число разбиений: %d\n", 2 \* n);  
 printf("Относительная погрешность: %lf\n\n", error4);  
  
 // Метод Симпсона  
 printf("Метод Симпсона:\n");  
 result1 = simpsonMethod(a, b, n);  
 result2 = simpsonMethod(a, b, 2 \* n);  
 double error5 = rungeRule(result1, result2, 4);  
 printf("Значение интеграла: %lf\n", result2);  
 printf("Число разбиений: %d\n", 2 \* n);  
 printf("Относительная погрешность: %lf\n\n", error5);  
  
 return 0;  
}

Вывод

Во время выполнения лабораторной работы я изучил работу метода Симпсона, метода трапеций и метода прямоугольников для нахождения интеграла с заданной точностью. Самым неточным являются методы правых и левых прямоугольников. Метод Симпсона является самым точным. Описание работы каждого метода приведены в отчете.