Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Факультет Программной инженерии и компьютерной техники

Кафедра информатики и прикладной математики

**Дисциплина «Вычислительная математика»**

**Лабораторная работа №2**

**Метод прямоугольников**

Выполнил:

Ореховский Антон Михайлович

Группа Р3217

Преподаватель:

Калёнова Ольга Вячеславовна

Санкт-Петербург, 2017 г.

**Описание метода**

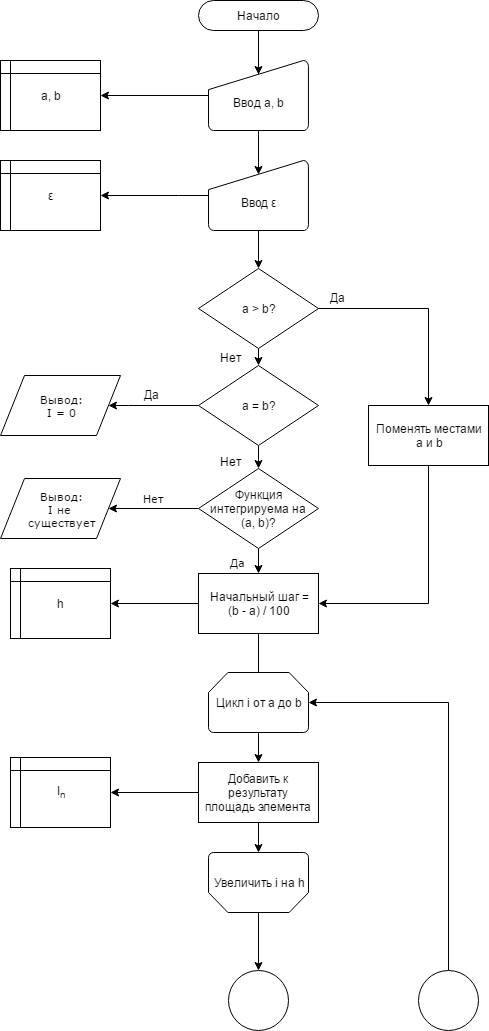
Метод прямоугольников — метод численного интегрирования функции одной переменной, заключающийся в замене подынтегральной функции на константу на каждом элементарном отрезке. Если рассмотреть график подынтегральной функции, то метод будет заключаться в приближённом вычислении площади под графиком суммированием площадей конечного числа прямоугольников, ширина которых будет определяться расстоянием между соответствующими соседними узлами интегрирования, а высота - значением подынтегральной функции в этих узлах.

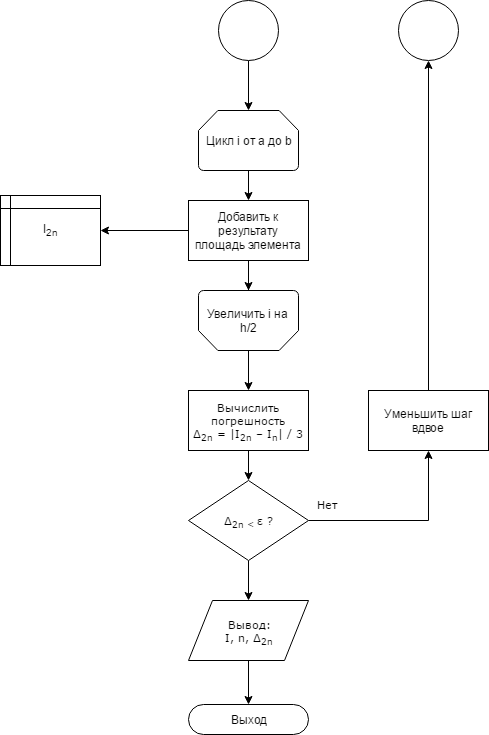
1. **Формула левых прямоугольников**:
2. **Формула правых прямоугольников**:
3. **Формула средних прямоугольников**:

После вычисления интеграла проводится повторное вычисление с удвоенным количеством разбиений и по правилу Рунге вычисляется текущая погрешность:

∆2n = θ |I2n – In|, где θ = 1/3.

Затем шаг уменьшается вдвое. Алгоритм повторяется до тех пор, пока погрешность не окажется меньше заданной пользователем точности.

****



**Листинг программы**

private Tuple Calculate(double a, double b, double precision)

{

if (a == b) {

Tuple result;

result.integral = 0;

result.steps = 0;

result.error = 0;

return result;

}

if (double.IsInfinity(f(a)) || double.IsNaN(f(a)))

a += double.Epsilon;

if (double.IsInfinity(f(b)) || double.IsNaN(f(b)))

b -= double.Epsilon;

double integral\_n, error = precision;

double step = (b - a) / STEPS\_AMOUNT;

do

{

integral\_n = SolveIntegral(a, b, step);

double integral\_2n = SolveIntegral(a, b, step / 2);

error = THETA \* Math.Abs(integral\_2n - integral\_n);

if (error >= precision) step /= 2;

} while (error >= precision);

Tuple tuple;

tuple.integral = integral\_n;

tuple.steps = (int)((b- a) / step);

tuple.error = error;

return tuple;

}

private double SolveIntegral(double leftLimit, double rightLimit, double step)

{

double x, result = 0;

for (x = leftLimit; x <= rightLimit; x += step)

result += Formula(x, x + step);

return result;

}

private double Formula(double a, double b)

{

switch (Modifications.SelectedIndex)

{

case 0:

return f(a) \* (b - a);

case 1:

return f(b) \* (b - a);

case 2:

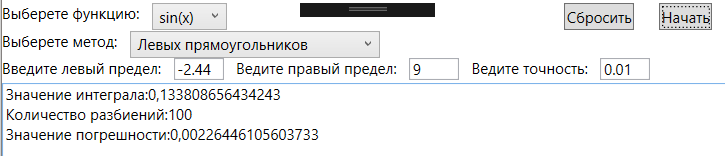
return f((a + b) / 2) \* (b - a);

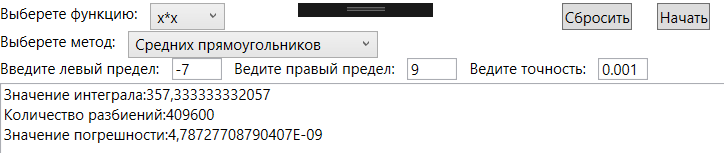
}

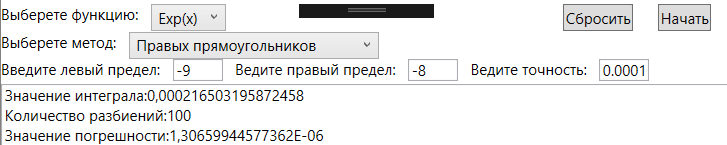
return Double.NaN;

}

**Тестовые данные**







**Вывод**

Метод прямоугольника менее точен, чем метод Симпсона. Порядок точности для левых и правых прямоугольников равен 0, для средних прямоугольников и трапеций – 1, в то время как метод Симпсона имеет порядок точности 3.

Это означает, что методы прямоугольников и трапеций дают низкую точность и большое количество разбиений при интегрировании полиномов степени 2 и более.