Описание метода, расчетные формулы

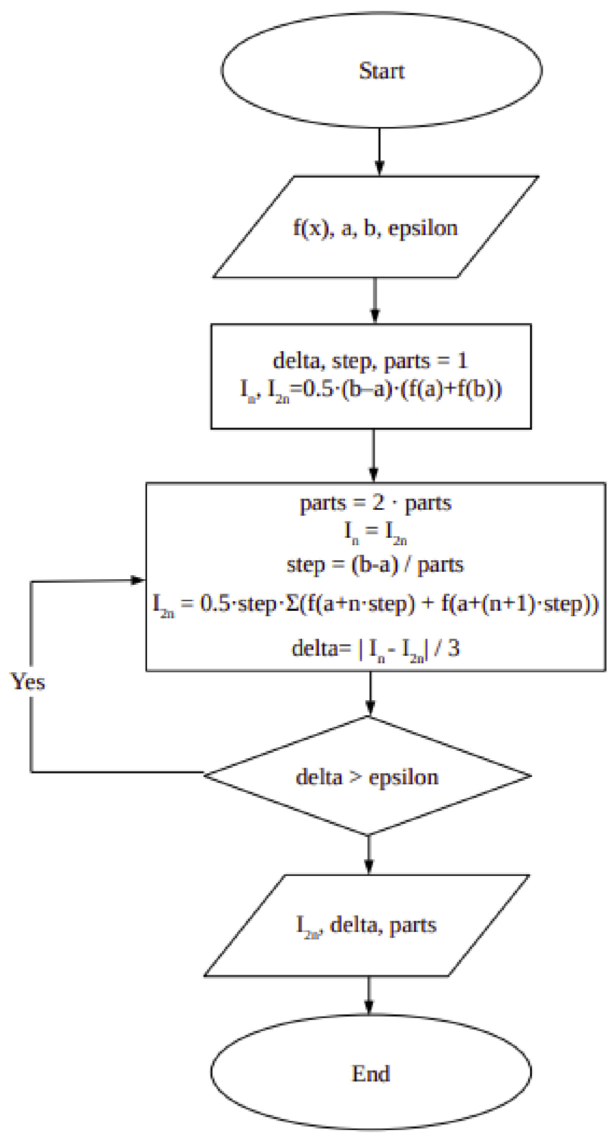
Метод трапеций – модификация метода прямоугольников, дающая более точные результаты. Идея заключается в разбиении площади под графиком подынтегральной функции на равные по ширине трапеции, и суммированиии их площадей.

$$S_{all} = S_1 + S_2 + \dots + S_n = \frac{y_0 + y_1}{2} \cdot h_1 + \frac{y_1 + y_2}{2} \cdot h_2 + \dots + \frac{y_{y-1} + y_n}{2} \cdot h_n =$$
$$= \int\limits_a^b f(x)dx = \frac{1}{2}\sum_{i=1}^{n} h_i(y_{i-1} + yi) = I_n.$$

После вычисления I_n проводится повторное интегрирование для $I_{2n} = \frac{1}{2}\sum\limits_{i=1}^{2n} h_i(y_{i-1} + y_i)$ и вычисляется погрешность по правилу Рунге.

$$\Delta_{2n} = \frac{|I_n - I_{2n}|}{3}$$

Блок-схема численного метода



Листинг реализованного численного метода программы

```java
package ru.ifmo.cmath.algebra;

public class Integrator implements TrapezoidalRule {
    private final Integer limit = (1 << 22);

    public IntegralAnswer integrate(Integral integral, Double accuracy) {
        return this.approximate(integral, accuracy);
    }

    public IntegralAnswer approximate(Integral integral, Double accuracy) {
        Integer parts = 1;
        Double integral1n, integral2n = area(integral, parts);
        Double delta;
        do {
            integral1n = integral2n;
            parts <<= 1;
            integral2n = area(integral, parts);
            delta = Math.abs(integral1n - integral2n) / 3;
        }
        while (delta > accuracy && parts < limit);
        if (parts.equals(limit)) {
            throw new DivergeException("Integral is diverge!");
        }
        return new IntegralAnswer(integral2n, delta, parts);
    }

    private Double area(Integral integral, Integer parts) {
        Double area = 0.0;
        Double step = integral.bounds().difference() / parts;
        for (int i = 1; i <= parts; i++) {
            area += this.trapezoid(
                    integral.function(),
                    integral.bounds().lower() + (i - 1) * step,
                    integral.bounds().lower() + i * step
            );
            if (area.isNaN() || area.isInfinite()) {
                throw new DivergeException("Integral is diverge!");
            }
        }
        return area;
    }
}
```

```java
package ru.ifmo.cmath.algebra;

public interface TrapezoidalRule {
    default Double trapezoid(Function function, Double a, Double b) {
        Double val1 = function.apply(a);
        if (val1.isNaN() || val1.isInfinite() && !a.equals(b)) {
            a += 0.000000001D;
        }
        Double val2 = function.apply(b);
        if (val2.isNaN() || val2.isInfinite() && !a.equals(b)) {
            b -= 0.000000001D;
        }
        return 0.5 * (b - a) * (function.apply(a) + function.apply(b));
    }
}
```

Примеры и результаты работы программы на разных данных

```
integrate$ integrate

f(x)=sin(x)/x
a=-0.24
b=9.9887324
accuracy = 0.00000122

area: 1.898188573454991 70000
error: 0.00000033009136209591
partitions: 2048

integrate$ integrate

f(x)=1/log(x)
a=0.1
b=2.09
accuracy = 0.000123

Integral is diverge!

integrate$ exit
```

Вывод

Все методы, т.е. прямоугольников, трапеций, парабол (Симпсона), построены на основе трех критерий:

(1) разбиении отрезка интегрирования на равные промежутки;

(2) аппроксимации подинтегральной функции на выбранных промежутках многочленами;

(3) нахождении суммарной площади полученных криволинейных трапеций.

Погрешность для каждого метода определяется следующими формулами:

(a) для прямоугольников: $|\Delta| \leq \max_{x\in[a,b]} |f''(x)] \cdot \dfrac{(b-a)^3}{24n^2},\ k=2$

(b) для трапеций: $|\Delta| \leq \max_{x\in[a,b]} |f''(x)| \cdot \dfrac{(b-a)^3}{12n^2},\ k=2$

(c) для парабол: $|\Delta| \leq \max_{x\in[a,b]} |f''(x)| \cdot \dfrac{(b-a)^5}{189n^4},\ k=4$

где k – порядок точности и n – количество разбиений.

Можно увидеть, что метод трапеций менее точный, чем метод парабол при равном количестве разбиений.