БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Методы интегрирования

Выполнил: Аквух Джеймс

2 курс 3 группа

Преподаватель: Будник А.М.

Минск 2016

**1. Постановка задачи**

Для заданной функции требуется найти значение её определённого интеграла на отрезке с точностью .

**2. Методы интегрирования**

1. метод средних прямоугольников

Формула интегрирования:



Погрешность:



1. метод трапеций

Формула интегрирования:



Погрешность:



1. метод средних прямоугольников с оценкой погрешности методом Рунге-Кутты

Формула интегрирования:



Погрешность:



Условие остановки вычислений:



1. метод Гаусса

Формула интегрирования:





**3. Исходные данные**

**4. Листинг**

const {sum} = require('mathjs');

const {exp, sin, sqrt, ceil, pow, abs, PI} = Math

const {log} = console

const a = 0, b = PI

const I = 12.0703463163896

const f = x => exp(x) \* sin(x)

const fd2max = PI / 4

const print = (name, n, h, I, In, R, EPS) => {

log(`

Method: ${name}

n: ${n}

h: ${h}

I: ${I}

In: ${In}

R: ${R}

EPS: ${EPS || 'Not set'}`)

}

const rectangles = (f, a, b, n) => {

const h = (b - a) / n

const nodes = [...Array(n).keys()].map(i => a + h \* i + h / 2)

return h \* sum(nodes.map(f))

}

const rectanglesTest = EPS => {

const n = ceil(sqrt(fd2max \* pow(b - a, 3) / 24 / EPS))

const In = rectangles(f, 0, PI, n)

print('Rectangles', n, (b - a) / n, I, In, abs(I - In), EPS)

}

const trapezoids = (f, a, b, n) => {

const h = (b - a) / n

const nodes = [...Array(n - 1).keys()].map(i => a + h \* (i + 1))

return h \* ((f(a) + f(b)) / 2 + sum(nodes.map(f)))

}

const trapezoidsTest = EPS => {

const n = ceil(sqrt(fd2max \* pow(b - a, 3) / 12 / EPS))

const In = trapezoids(f, 0, PI, n)

print('Trapezoids', n, (b - a) / n, I, In, abs(I - In), EPS)

}

const runge = (I, m, n, EPS) => {

while (abs(I(n) - I(2 \* n)) / (pow(2, m) - 1) > EPS)

return runge(I, m, n \* 2, EPS)

return {n, In: I(2 \* n)}

}

const rungeTest = EPS => {

const S = n => rectangles(f, a, b, n)

const {n, In} = runge(S, 1, 1, EPS)

print('Runge', n, (b - a) / n, I, In, abs(I - In), EPS)

}

const gauss = (f, a, b) => {

const x = [-sqrt(0.6), 0, sqrt(0.6)]

const A = [5 / 9, 8 / 9, 5 / 9]

return (b - a) / 2 \* sum([0, 1, 2].map(i => A[i] \* f((b - a) / 2 \* x[i] + (a + b) / 2)))

}

const gaussTest = () => {

const n = 3

const In = gauss(f, 0, PI)

print('Gauss', n, (b - a) / n, I, In, abs(I - In))

}

[rectanglesTest, trapezoidsTest, rungeTest].map(f => {

f(pow(10, -3))

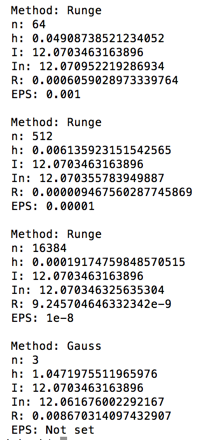
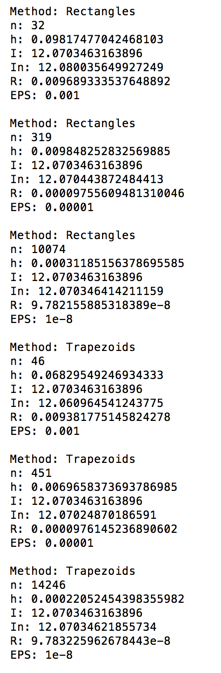
f(pow(10, -5))

f(pow(10, -8))

})

gaussTest()

**5. Результаты**



**6. Вывод**

Число узлов разбиения отрезка, необходимое для достижения заданной точности () имеет одинаковый порядок для метода трапеций и метода средних прямоугольников. Это объясняется тем, что оба метода обладают одинаковым порядком алгебраической точности (=1). Следует отметить, что для определения числа узлов разбиения отрезка интегрирования в методах Ньютона-Котеса требуется знать производные интегрируемой функции определенного порядка (зависит от метода). Чтобы этого избежать используется оценка погрешности вычисления с помощью правила Рунге-Кутта. Число узлов разбиения, определенное по правилу Рунге-Кутта равно ближайшей степени двойки к теоретическому числу узлов разбиения. Это подтверждает справедливость использованного правила для определения погрешности.

Тем не менее, рассмотренные выше методы не являются оптимальными с точки зрения степени алгебраической точности при заданном числе узлов разбиения. Оптимальной формулой интегрирования для фиксированного числа узлов разбиения является формула Гаусса. Из результатов видно, что погрешность метода Гаусса на 3 узлах меньше, чем, к примеру, погрешность метода средних прямоугольников на 32 узлах. Однако для использования метода Гаусса необходимо знать аналитическое выражение интегрируемой функции, или по крайней мере ее точечные значения с большой точностью, что ограничивает ее практическое применение.