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 $f(x)$ требуется найти значение её определённого интеграла на отрезке $[a,\ b]$ с точностью ε.

2. Методы интегрирования

A. метод средних прямоугольников

Формула интегрирования:

$$\int_a^b f(x) \approx h \sum_{i=0}^{n-1} f(x_i + h/2),$$

Погрешность:

$$\frac{M_2(b-a)^3}{24n^2}$$

B. метод трапеций

Формула интегрирования:

$$\int_a^b f(x)\,dx = h\left(\frac{f_0 + f_n}{2} + \sum_{i=1}^{n-1} f_i\right) + E_n(f),$$

Погрешность:

$$\frac{f''(\xi)}{12}(b-a)h^2.$$

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

Формула интегрирования:

$$\int_a^b f(x) \approx h \sum_{i=0}^{n-1} f(x_i + h/2),$$

Погрешность:

$$r_n = |I - I_n| \leq \frac{|I_n - I_{2n}|}{2^m - 1}$$

Условие остановки вычислений:

$$r_n \leq \varepsilon$$

D. метод Гаусса

Формула интегрирования:

$$\int_a^b \rho(x) f(x) dx \approx \frac{b-a}{2} \sum_{i=1}^{n} A_i f\left(\frac{b-a}{2} x_i + \frac{a+b}{2}\right),$$

$$\sum_{i=1}^{n} A_i x_i^k = \int_a^b \rho(x) x^k dx, \qquad k = \overline{1, 2n-1}$$

3. Исходные данные

$$I = \int_0^{\pi} e^x \, sin(x) \, dx$$

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

```
Method: Rectangles
n: 32
h: 0.09817477042468103
I: 12.0703463163896
In: 12.080035649927249
R: 0.009689333537648892
EPS: 0.001

Method: Rectangles
n: 319
h: 0.009848252832569885
I: 12.0703463163896
In: 12.070443872484413
R: 0.00009755609481310046
EPS: 0.00001

Method: Rectangles
n: 10074
h: 0.0003118515637869585
I: 12.0703463163896
In: 12.070346414211159
R: 9.782155885318389e-8
EPS: 1e-8

Method: Trapezoids
n: 46
h: 0.06829549246934333
I: 12.0703463163896
In: 12.060964541243775
R: 0.009381775145824278
EPS: 0.001

Method: Trapezoids
n: 451
h: 0.0069658373693786985
I: 12.0703463163896
In: 12.07024870186591
R: 0.0000976145236890602
EPS: 0.00001

Method: Trapezoids
n: 14246
h: 0.00022052454398355982
I: 12.0703463163896
In: 12.07034621855734
R: 9.783225962678443e-8
EPS: 1e-8

Method: Runge
n: 64
h: 0.04908738521234052
I: 12.0703463163896
In: 12.070952219286934
R: 0.0006059028973339764
EPS: 0.001

Method: Runge
n: 512
h: 0.006135923151542565
I: 12.0703463163896
In: 12.070355783949887
R: 0.000009467560287745869
EPS: 0.00001

Method: Runge
n: 16384
h: 0.00019174759848570515
I: 12.0703463163896
In: 12.070346325635304
R: 9.245704646332342e-9
EPS: 1e-8

Method: Gauss
n: 3
h: 1.0471975511965976
I: 12.0703463163896
In: 12.061676002292167
R: 0.008670314097432907
EPS: Not set
```

6. Вывод

Число узлов разбиения отрезка, необходимое для достижения заданной точности (10^{-5}) 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.