# Численное интегрирование

Методическое пособие

# Содержание

# 1 Введение

## 1.1 Постановка задачи

Численное интегрирование (числовая квадратура) — совокупность численных методов для вычисления значения определённого интеграла:

$$I = \int_a^b f(x)\,dx$$

когда аналитическое вычисление невозможно или затруднительно.

## 1.2 Общая формула квадратур

Большинство методов численного интегрирования можно представить в виде:

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{n} w_i f(x_i)$$

где:

- $x_i$ — узлы интегрирования ($a \leq x_0 < x_1 < \ldots < x_n \leq b$)
- $w_i$ — весовые коэффициенты
- $n$ — количество узлов

## 1.3 Алгебраическая степень точности

Алгебраической степенью точности квадратурной формулы называется наибольшее натуральное число $m$ такое, что формула точна для всех многочленов степени не выше $m$.

# 2 Простейшие квадратурные формулы

## 2.1 Формулы прямоугольников

### 2.1.1 Левые прямоугольники

$$\int_a^b f(x)\,dx \approx h \sum_{i=0}^{n-1} f(x_i), \quad h = \frac{b-a}{n}, \quad x_i = a + ih$$

### 2.1.2 Правые прямоугольники

$$\int_a^b f(x)\,dx \approx h \sum_{i=1}^{n} f(x_i)$$

### 2.1.3 Средние прямоугольники

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

### 2.2 Формула трапеций

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

### 2.3 Формула Симпсона

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\left[f(a) + 4\sum_{i=1,3,5,\ldots}^{n-1} f(x_i) + 2\sum_{i=2,4,6,\ldots}^{n-2} f(x_i) + f(b)\right]$$

**Условие:** $n$ должно быть чётным.

## 3 Вывод квадратурных формул

### 3.1 Интерполяционный подход

Квадратурные формулы можно получать, заменяя функцию $f(x)$ интерполяционным многочленом и интегрируя его точно.

#### 3.1.1 Вывод формулы трапеций

Заменим функцию $f(x)$ линейным многочленом Лагранжа на отрезке $[a, b]$:

$$P_1(x) = f(a)\frac{x-b}{a-b} + f(b)\frac{x-a}{b-a}$$

Проинтегрируем многочлен:

$$\int_a^b P_1(x)\,dx = \frac{b-a}{2}[f(a) + f(b)]$$

#### 3.1.2 Вывод формулы Симпсона

Для трёх равноотстоящих узлов $x_0 = a$, $x_1 = \frac{a+b}{2}$, $x_2 = b$ строим интерполяционный многочлен Лагранжа степени 2:

$$P_2(x) = \sum_{i=0}^{2} f(x_i)L_i(x)$$

Интегрирование даёт:

$$\int_a^b P_2(x)\,dx = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

## 4 Погрешности методов

### 4.1 Общий вид остаточного члена

Для квадратурной формулы с алгебраической степенью точности $m$ остаточный член можно представить в виде:

$$R[f] = Kf^{(m+1)}(\xi), \quad \xi \in [a, b]$$

## 4.2 Погрешности конкретных методов

### 4.2.1 Метод прямоугольников

- Левые/правые прямоугольники: $R = \frac{(b-a)^2}{2n} f'(\xi)$
- Средние прямоугольники: $R = \frac{(b-a)^3}{24n^2} f''(\xi)$

### 4.2.2 Метод трапеций

$$R = -\frac{(b-a)^3}{12n^2} f''(\xi), \quad \xi \in [a, b]$$

### 4.2.3 Метод Симпсона

$$R = -\frac{(b-a)^5}{2880n^4} f^{(4)}(\xi), \quad \xi \in [a, b]$$

# 5 Алгоритмы численного интегрирования

# 6 Метод Рунге оценки погрешности

## 6.1 Основная идея

Оценка погрешности вычисляется путем сравнения результатов с разным шагом интегрирования.

## 6.2 Формула Рунге

Если квадратурная формула имеет порядок точности $p$, то:

$$I \approx I_h + \frac{I_h - I_{2h}}{2^p - 1}$$

где $I_h$ — значение интеграла с шагом $h$, $I_{2h}$ — с шагом $2h$.

## 6.3 Алгоритм оценки погрешности

# 7 Адаптивные методы

## 7.1 Идея адаптивного интегрирования

Адаптивные методы автоматически регулируют шаг интегрирования в зависимости от локальных свойств функции: уменьшают шаг на участках с большей изменчивостью и увеличивают на гладких участках.

## 7.2 Алгоритм адаптивного интегрирования

# 8 Квадратурные формулы Гаусса

## 8.1 Общая теория

Квадратурные формулы Гаусса достигают максимально возможной алгебраической степени точности $2n - 1$ для $n$ узлов.

**Algorithm 1** Общий алгоритм численного интегрирования

```
procedure NumericalIntegration(f, a, b, n, method)
    h ← (b − a)/n
    sum ← 0
    if method = "left rectangle then
        for i = 0 to n − 1 do
            x ← a + i · h
            sum ← sum + f(x)
        end for
        return h · sum
    else if method = "mid rectangle then
        for i = 0 to n − 1 do
            x ← a + (i + 0.5) · h
            sum ← sum + f(x)
        end for
        return h · sum
    else if method = "trapezoidal then
        sum ← (f(a) + f(b))/2
        for i = 1 to n − 1 do
            x ← a + i · h
            sum ← sum + f(x)
        end for
        return h · sum
    else if method = "simpson then
        if n%2 ≠ 0 then                                    ▷ Проверка четности n
            n ← n + 1
            h ← (b − a)/n
        end if
        sum ← f(a) + f(b)
        for i = 1 to n − 1 do
            x ← a + i · h
            if i%2 = 0 then
                sum ← sum + 2 · f(x)
            else
                sum ← sum + 4 · f(x)
            end if
        end for
        return h · sum/3
    end if
end procedure
```

**Algorithm 2** Оценка погрешности по Рунге

```
procedure RungeErrorEstimate(f, a, b, n, p)
    I_h ← NumericalIntegration(f, a, b, n, method)
    I_2h ← NumericalIntegration(f, a, b, n/2, method)
    error ← |I_h − I_2h|/(2^p − 1)
    return error
end procedure
```

**Algorithm 3** Адаптивное интегрирование методом трапеций

```
procedure AdaptiveIntegration(f, a, b, ϵ)
    RecursiveIntegration(a, b, f(a), f(b), ϵ, 0)
end procedure
procedure RecursiveIntegration(a, b, f_a, f_b, ϵ, depth)
    if depth > max_depth then
        return (f_a + f_b) · (b − a)/2        ▷ Защита от бесконечной рекурсии
    end if
    mid ← (a + b)/2
    f_m ← f(mid)
    whole ← (f_a + f_b) · (b − a)/2
    left ← (f_a + f_m) · (mid − a)/2
    right ← (f_m + f_b) · (b − mid)/2
    total ← left + right
    if |total − whole| ≤ ϵ · (b − a)/(b_0 − a_0) then
        return total
    else
        left_result ← RecursiveIntegration(a, mid, f_a, f_m, ϵ, depth + 1)
        right_result ← RecursiveIntegration(mid, b, f_m, f_b, ϵ, depth + 1)
        return left_result + right_result
    end if
end procedure
```

### 8.2 Узлы и веса

Узлы квадратур Гаусса являются корнями ортогональных многочленов (Лежандра, Чебышева и др.), а веса вычисляются из условий точности для многочленов высокой степени.

### 8.3 Формула Гаусса-Лежандра

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

где $x_i$ — корни многочлена Лежандра степени $n$, $w_i$ — соответствующие веса.

## 9 Метод Монте-Карло

### 9.1 Основная идея

Интеграл оценивается с использованием случайных чисел:

$$\int_{a}^{b} f(x)\,dx \approx (b-a) \cdot \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

где $x_i$ — случайные числа с равномерным распределением на $[a, b]$.

### 9.2 Погрешность метода

Погрешность метода Монте-Карло убывает как $O(1/\sqrt{N})$, что делает его эффективным для многомерных интегралов.

# 10 Сравнительный анализ методов

| Метод | Порядок | Сложность | Условия | Применение |
|---|---|---|---|---|
| Прямоугольники | 1-2 | $O(n)$ | Гладкость $f(x)$ | Быстрые оценки |
| Трапеции | 2 | $O(n)$ | Гладкость $f(x)$ | Универсальный |
| Симпсон | 4 | $O(n)$ | Чётное $n$, $f \in C^4$ | Высокая точность |
| Гаусс | 2n-1 | $O(n)$ | Вычисление узлов | Макс. точность |
| Монте-Карло | 0.5 | $O(N)$ | Случайность | Многомерные |

Таблица 1: Сравнительная характеристика методов интегрирования

# 11 Практические рекомендации

## 11.1 Выбор метода

- Для гладких функций — метод Симпсона или Гаусса
- Для функций с разрывами — адаптивные методы
- Для многомерных интегралов — метод Монте-Карло
- Для быстрых оценок — метод трапеций

## 11.2 Выбор шага интегрирования

- Начать с грубого шага и последовательно уменьшать
- Использовать оценку Рунге для контроля точности
- Для адаптивных методов задать относительную погрешность

## 11.3 Оценка точности

- Всегда оценивать погрешность вычислений
- Использовать метод Рунге для апостериорной оценки
- Сравнивать результаты разных методов

# 12 Заключение

Численное интегрирование является фундаментальным инструментом вычислительной математики. Выбор оптимального метода зависит от свойств интегрируемой функции, требуемой точности и доступных вычислительных ресурсов. Понимание теоретических основ и алгоритмических особенностей методов позволяет эффективно применять их для решения практических задач.