ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

Приближенное вычисление интегралов
(Вариант 8)

Выполнил студент 3 курса ПМиИ
Ковшов Максим

Цель занятия: изучение различных методов вычисления определенных интегралов, практическое интегрирование функций на ЭВМ.

Задания к работе: 1. Вычислить приближенно с заданной точностью интеграл $I = \int_a^b f(x)dx$ по формулам прямоугольников, трапеций и Симпсона. Величину шага определить с помощью двойного пересчета.

2. Определить относительную погрешность вычислений каждого метода по формуле: $\delta = \left| \frac{I - I_h}{I} \right| \cdot 100\ \%$, где I – точное значение интеграла; I_h – приближенное.

3. Составить таблицу в которой указать значение интеграла, полученное с заданной точностью, величину последнего шага интегрирования, количество точек разбиения, относительную погрешность метода.

Метод прямоугольников

Левых:

$$I = h\sum_{i=0}^{n-1} f(x_i), \quad \text{где } h = \frac{b-a}{n}$$

Правых

$$I = h\sum_{i=1}^{n} f(x_i), \quad \text{где } h = \frac{b-a}{n}$$

Погрешность абсолютная

$$\Delta = max\left|\frac{f'(x)}{2}\right|(b-a)h$$

Средних:

$$I = h\sum_{i=1}^{n} f\left(x_{i-1} + \frac{h}{2}\right), \quad \text{где } h = \frac{b-a}{n}$$

Оценка погрешности

$$\Delta = max\left|\frac{f''(x)}{24}\right|(b-a)h^2$$

Метод трапеций

$$I = h(\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(x_i)), \quad \text{где } h = \frac{b-a}{n}$$

$$\Delta = max|\frac{f''(x)}{12}|(b-a)h^2$$

Метод Симпсона

$$I = \frac{h}{3}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_{2i}) + 4\sum_{i=1}^{n} f(x_{2i-1}) + f(x_n)\right]$$

$$\Delta = \frac{b-a}{2880}h^4 max\left|f^{(4)}(x)\right|$$

Результаты:

№	Подынтегральная функция $f(x)$	Заданная точность	Интервал $[a, b]$	Первообразная функции $F(x)$
8	$\operatorname{arctg} x$	10^{-5}	$[0; 1]$	$x \cdot \operatorname{arctg} x - \frac{\ln(1+x^2)}{2}$

Метод решения	Значение интеграла	Величина последнего шага	Количество точек разбиения	Относительная погрешность
1. Метод левых прямоугольников	0.438819	0.000015258789062	65536	0.001%
2. Метод правых прямоугольников	0.438831	0.000015258789062	65536	0.001%
3. Метод средних прямоугольников	0.438826	0.007812500000000	128	0.0003%
4. Метод трапеций	0.438831	0.000007629394531	131072	0.001%
5. Метод Симпсона	0.438831	0.000015258789062	65536	0.002%

Вывод: Метод средних прямоугольников оказался самым быстрым и точным - 128 итераций и относительная погрешность 0.0003%.

Все исходные тексты программ в Приложении

```
Метод левых прямоугольников:
I: 0.438818580995323
h: 0.000015258789062
n: 65536
Error 0.001365493757665

Метод правых прямоугольников:
I: 0.438830565220228
h: 0.000015258789062
n: 65536
Error 0.001365489336604

Метод средних прямоугольников:
I: 0.438825844694549
h: 0.007812500000000
n: 128
Error 0.000289768885192

Метод трапеций:
I: 0.438830565227506
h: 0.000007629394531
n: 131072
Error 0.001365490995229

Метод Симсона:
I: 0.438832562600745
h: 0.000015258789062
n: 65536
Error 0.001820655395956
```

ПРИЛОЖЕНИЕ

```cpp
#define ACCURACY pow(10, -5)
#define Exact_Solution (primitive(1) - primitive(0))

double function(double x)
{
    return atan(x);
}

double primitive(double x)
{
    return x * atan(x) - (log(1 + x * x) / 2);
}

double relativeError(double answer)
{
    return abs(100 * (Exact_Solution - answer) / Exact_Solution);
}

double leftRectMethod(double a, double b)
{
    int n = 2;
    double answer = 0;
    while (true)
    {
        double h = (b - a) / n;
        double x = a;
        double prev = answer;
        answer = 0;
        for (int i = 0; i < n; i++)
        {
            answer += function(x) * h;
            x += h;
        }
        double calc_error = abs(answer - prev);
        if (calc_error < ACCURACY)
        {
            std::cout << "\nМетод левых прямоугольников:\nI: " <<
                std::fixed << std::setprecision(15) << answer
                << "\nh: " << h << "\nn: " << n <<
                "\nError " << relativeError(answer) << std::endl;
            return answer;
        }
        n *= 2;
    }
}
```

```cpp
double rightRectMethod(double a, double b)
{
    int n = 2;
    double answer = 0;
    while (true)
    {
        double h = (b - a) / n;
        double x = a + h;
        double prev = answer;
        answer = 0;
        for (int i = 0; i < n; i++)
        {
            answer += function(x) * h;
            x += h;
        }
        double calc_error = abs(answer - prev);
        if (calc_error < ACCURACY)
        {
            std::cout << "\nМетод правых прямоугольников:\nI: " <<
                std::fixed << std::setprecision(15) << answer
                << "\nh: "  << h << "\nn: " << n <<
                "\nError " << relativeError(answer) << std::endl;
            return answer;
        }
        n *= 2;
    }
}

double midRectMethod(double a, double b)
{
    int n = 2;
    double answer = 0;
    while (true)
    {
        double h = (b - a) / n;
        double x = a + h / 2;
        double prev = answer;
        answer = 0;
        for (int i = 0; i < n; i++)
        {
            answer += function(x) * h;
            x += h;
        }
        double calc_error = abs(answer - prev);
        if (calc_error < ACCURACY)
        {
            std::cout << "\nМетод средних прямоугольников:\nI: " <<
```

```
                std::fixed << std::setprecision(15) << answer
                << "\nh: " << h << "\nn: " << n <<
                "\nError " << relativeError(answer) << std::endl;
            return answer;
        }
        n *= 2;
    }
}

double trapezoidalMethod(double a, double b)
{
    int n = 2;
    double answer = 0;
    while (true)
    {
        double h = (b - a) / n;
        double x = a;
        double prev = answer;
        answer = (function(a)+function(b)) / 2;
        for (int i = 0; i < n; i++)
        {
            x += h;
            answer += function(x);
        }
        answer *= h;
        double calc_error = abs(answer - prev);
        if (calc_error < ACCURACY)
        {
            std::cout << "\nМетод трапеций:\nI: " <<
                std::fixed << std::setprecision(15) << answer
                << "\nh: " << h << "\nn: " << n <<
                "\nError " << relativeError(answer) << std::endl;
            return answer;
        }
        n *= 2;
    }
}
```

```
double simpsonMethod(double a, double b)
{
    int n = 2;
    double answer = 0;
    while (true)
    {
        double h = (b - a) / n;
        double x = a;
        double prev = answer;
        answer = function(a) + function(b);
        for (int i = 1; i <= n; i++)
        {
            x += h;
            if (i % 2 ==0)
                answer += 2 * function(x);
            else
                answer += 4 * function(x);
        }
        answer *= h/3;
        double calc_error = abs(answer - prev);
        if (calc_error < ACCURACY)
        {
            std::cout << "\nМетод Симсона:\nI: " <<
                std::fixed << std::setprecision(15) << answer
                << "\nh: " << h << "\nn: " << n <<
                "\nError " << relativeError(answer) << std::endl;
            return answer;
        }
        n *= 2;
    }
}

int main()
{
    setlocale(LC_ALL, "Russian");
    double a = 0, b = 1;
    leftRectMethod(a, b);
    rightRectMethod(a, b);
    midRectMethod(a, b);
    trapezoidalMethod(a, b);
    simpsonMethod(a, b);
}
```