

**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

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

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

Метод Симпсона

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

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

Результаты:

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

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 Симпсона:
I: 0.438832562600745
h: 0.000015258789062
n: 65536
Error 0.001820655395956
```

```

5  #define ACCURACY pow(10, -5)
6  #define Exact_Solution (primitive(1) - primitive(0))
7
8  double function(double x)
9  {
10     return atan(x);
11 }
12
13 double primitive(double x)
14 {
15     return x * atan(x) - (log(1 + x * x) / 2);
16 }
17
18 double relativeError(double answer)
19 {
20     return abs(100 * (Exact_Solution - answer) / Exact_Solution);
21 }
22
23 double leftRectMethod(double a, double b)
24 {
25     int n = 2;
26     double answer = 0;
27     while (true)
28     {
29         double h = (b - a) / n;
30         double x = a;
31         double prev = answer;
32         answer = 0;
33         for (int i = 0; i < n; i++)
34         {
35             answer += function(x) * h;
36             x += h;
37         }
38         double calc_error = abs(answer - prev);
39         if (calc_error < ACCURACY)
40         {
41             std::cout << "\nМетод левых прямоугольников:\nI: " <<
42                 std::fixed << std::setprecision(15) << answer
43                 << "\nh: " << h << "\nn: " << n <<
44                 "\nError " << relativeError(answer) << std::endl;
45             return answer;
46         }
47         n *= 2;
48     }
49 }

```

```

51  double rightRectMethod(double a, double b)
52  {
53      int n = 2;
54      double answer = 0;
55      while (true)
56      {
57          double h = (b - a) / n;
58          double x = a + h;
59          double prev = answer;
60          answer = 0;
61          for (int i = 0; i < n; i++)
62          {
63              answer += function(x) * h;
64              x += h;
65          }
66          double calc_error = abs(answer - prev);
67          if (calc_error < ACCURACY)
68          {
69              std::cout << "\nМетод правых прямоугольников:\nI: " <<
70                  std::fixed << std::setprecision(15) << answer
71                  << "\nh: " << h << "\nn: " << n <<
72                  "\nError " << relativeError(answer) << std::endl;
73              return answer;
74          }
75          n *= 2;
76      }
77  }
78
79  double midRectMethod(double a, double b)
80  {
81      int n = 2;
82      double answer = 0;
83      while (true)
84      {
85          double h = (b - a) / n;
86          double x = a + h / 2;
87          double prev = answer;
88          answer = 0;
89          for (int i = 0; i < n; i++)
90          {
91              answer += function(x) * h;
92              x += h;
93          }
94          double calc_error = abs(answer - prev);
95          if (calc_error < ACCURACY)
96          {
97              std::cout << "\nМетод средних прямоугольников:\nI: " <<

```

```

108         std::fixed << std::setprecision(15) << answer
109         << "\nh: " << h << "\nn: " << n <<
110         "\nError " << relativeError(answer) << std::endl;
111         return answer;
112     }
113     n *= 2;
114 }
115 }
116
117 double trapezoidalMethod(double a, double b)
118 {
119     int n = 2;
120     double answer = 0;
121     while (true)
122     {
123         double h = (b - a) / n;
124         double x = a;
125         double prev = answer;
126         answer = (function(a)+function(b)) / 2;
127         for (int i = 0; i < n; i++)
128         {
129             x += h;
130             answer += function(x);
131         }
132         answer *= h;
133         double calc_error = abs(answer - prev);
134         if (calc_error < ACCURACY)
135         {
136             std::cout << "\nМетод трапеций:\nI: " <<
137                 std::fixed << std::setprecision(15) << answer
138                 << "\nh: " << h << "\nn: " << n <<
139                 "\nError " << relativeError(answer) << std::endl;
140             return answer;
141         }
142         n *= 2;
143     }
144 }

```

```

136  double simpsonMethod(double a, double b)
137  {
138      int n = 2;
139      double answer = 0;
140      while (true)
141      {
142          double h = (b - a) / n;
143          double x = a;
144          double prev = answer;
145          answer = function(a) + function(b);
146          for (int i = 1; i <= n; i++)
147          {
148              x += h;
149              if (i % 2 == 0)
150                  answer += 2 * function(x);
151              else
152                  answer += 4 * function(x);
153          }
154          answer *= h/3;
155          double calc_error = abs(answer - prev);
156          if (calc_error < ACCURACY)
157          {
158              std::cout << "\nМетод Симсона:\nI: " <<
159                  std::fixed << std::setprecision(15) << answer
160                  << "\nh: " << h << "\nn: " << n <<
161                  "\nError " << relativeError(answer) << std::endl;
162              return answer;
163          }
164          n *= 2;
165      }
166  }
167
168  int main()
169  {
170      setlocale(LC_ALL, "Russian");
171      double a = 0, b = 1;
172      leftRectMethod(a, b);
173      rightRectMethod(a, b);
174      midRectMethod(a, b);
175      trapezoidalMethod(a, b);
176      simpsonMethod(a, b);
177  }

```