МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по практической работе №7

по дисциплине «Вычислительная математика»

Тема: Составные формулы прямоугольников, трапеций, Симпсона

Студентка гр. 8382	____________________	Наконечная А.Ю.
Преподаватель	____________________	Щеголева Н.Л.

Санкт-Петербург
2019

Цель работы.

Изучение и сравнение различных методов численного интегрирования на примере составных формул прямоугольников, трапеций, Симпсона.

Задание.

Используя квадратурные формулы прямоугольников, трапеций и Симпсона, вычислить значения заданного интеграла и, применив правило Рунге, найти наименьшее значение n (наибольшее значение шага h), при котором каждая из указанных формул дает приближенное значение интеграла с погрешностью ε, не превышающей заданную.

Вариант №11: $\int_1^2 \exp\left(-x-\frac{1}{x}\right)dx$ (1)

Основные теоретические положения.

Пусть требуется найти определенный интеграл

$$I=\int_a^b f(x)dx \quad , \tag{2}$$

где функция *f(x)* непрерывна на отрезке *[a, b]*.

Для приближенного вычисления интегралов чаще всего подынтегральную функцию заменяют «близкой» ей вспомогательной функцией, интеграл от которой вычисляется аналитически. За приближенное значение интеграла принимают значение интеграла от вспомогательной функции.

Заменим функцию на отрезке *[a, b]* её значением в середине отрезка. Искомый интеграл, равный площади криволинейной фигуры, заменяется на площадь прямоугольника. Из геометрических соображений нетрудно записать формулу прямоугольников:

$$\int_a^b f(x)dx \approx f\left(\frac{a+b}{2}\right)(b-a) \tag{3}$$

Приблизив *f(x)* линейной функцией и вычислив площадь соответствующей трапеции, получим формулу трапеций:

$$\int_a^b f(x)dx \approx \frac{1}{2}(f(a)+f(b))(b-a) \qquad (4)$$

Если же приблизить подынтегральную функцию параболой, проходящей через точки *(a, f(a)),* $(\frac{a+b}{2}, f(\frac{a+b}{2}))$ *, (b, f(b)),* то получим формулу Симпсона (парабол):

$$\int_a^b f(x)dx \approx \frac{1}{6}(f(a)+4f(\frac{a+b}{2})+f(b))(b-a) \qquad (5)$$

Все три формулы хорошо иллюстрируются геометрически и представлены на *рис. 1*.

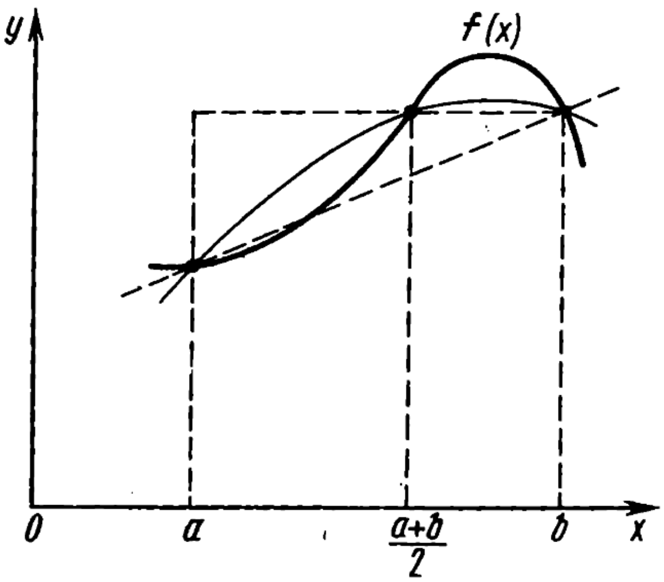

рис. 1 — Геометрическое представление формулы прямоугольников, трапеций и парабол

Для повышения точности интегрирования применяют составные формулы. Для этого разбивают отрезок *[a, b]* на чётное *n = 2m* число отрезков длины $h = \frac{b-a}{n}$ на каждом из отрезков длины *2h* применяют соответствующую формулу. Таким образом получают составные формулы прямоугольников, трапеций и Симпсона.

На сетке $x_i = a + ih$, $y = f(x_i)$, *i = 0, 1, 2, …, 2m*, составные формулы имеют следующий вид:

Формула прямоугольников:

$$\int_a^b f(x)dx = h\sum_{i=0}^{n-1}\left(f\left(x_i + \frac{h}{2}\right)\right) + R_1 \qquad (6)$$

Формула трапеций:

$$\int_a^b f(x)dx = \frac{h}{2}\sum_{i=0}^{n-1}\left(f(x_i) + f(x_{i+1})\right) + R_2 \qquad (7)$$

Формула парабол:

$$\int_a^b f(x)dx = \frac{h}{3}\sum_{i=0}^{m-1}\left(f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})\right) + R_3 \ , \qquad (8)$$

где $R_{1,} R_{2,} R_3$ - остаточные члены, $n \to \infty$ приближенные значения интегралов для всех трех формул (в предположении отсутствия погрешностей округления) стремятся к точному значению интеграла.

Для практической оценки погрешности квадратурной формулы можно использовать правило Рунге. Для этого проводят вычисления на сетках с шагом *h*, и *h/2* получают приближённые значения интеграла I_h и $I_{h/2}$ за окончательные значения интеграла принимают величины:

Для формулы прямоугольников:

$$I_{h/2} + \frac{I_{h/2} - I_h}{3} \qquad (9)$$

Для формулы трапеций:

$$I_{h/2} + \frac{I_{h/2} - I_h}{3} \qquad (10)$$

Для формулы парабол:

$$I_{h/2} + \frac{I_{h/2} - I_h}{15} \qquad (11)$$

За погрешность приближенного значения интеграла для формул прямоугольников и трапеций тогда принимают величину $|I_{h/2} - I_h|/3$, а для формулы Симпсона $|I_{h/2} - I_h|/15$.

Выполнение работы.

1. Составим программы-функции для вычисления интегралов по формулам прямоугольников (RECT), трапеций (TRAP) и Симпсона (SIMPS), а также подпрограмму-функцию F для вычисления подынтегральной функции.

```
double F(double x) {
    return exp(-x - (1 / x));
}

double RECT(int n, int a, int b) {
    double x, h, sum = 0;
    h = (double)(b - a) / n;

    for (int i = 0; i <= n - 1; ++i) {
        x = a + i * h;
        sum += F(x + (h / 2));
    }

    return (h * sum);
}

double TRAP(int n, int a, int b) {
    double x, x1, h, sum = 0;
    h = (double)(b - a) / n;

    for (int i = 0; i<n; ++i) {
        x = a + i * h;
        x1 = a + (i + 1) * h;
        sum += (F(x) + F(x1));
    }

    return (h / 2 * sum);
}

double SIMPS(int n, int a, int b) {
    int m;
    double x, x1, x2, h, sum = 0;
    m = n >> 1;
    h = (double)(b - a) / n;

  for (int i = 0; i<m; ++i) {
        x = a + (2 * i) * h;
        x1 = a + (2 * i + 1) * h;
        x2 = a + (2 * i + 2) * h;
        sum += (F(x) + 4 * F(x1) + F(x2));
    }

    return (h / 3 * sum);
```

```
}
```

2. Составим головную программу, содержащую оценку по Рунге погрешности каждой из перечисленных ранее квадратурных формул, удваивающих n до тех пор, пока погрешность не станет меньше ε, и осуществляющих печать результатов значения интеграла и значения n для каждой формулы.

```
int main() {
    int nr = 2, nt = 2, ns = 2;
    int a = 1, b = 2;
    double eps, Ir, It, Is, eps1, eps2, eps3;

    cout << ">> Input eps" << endl;
    cin >> eps;
    cout << endl;
    cout << ">> eps is: " << eps << endl;

    eps1 = fabs((RECT(nr << 1, a, b) - RECT(nr, a, b)) / 3);
    while(eps1 >= eps) {
        nr <<= 1;
        eps1 = fabs((RECT(nr << 1, a, b) - RECT(nr, a, b)) / 3);
    }
    Ir = RECT(nr << 1, a, b) + eps1;

    eps2 = fabs((TRAP(nt << 1, a, b) - TRAP(nt, a, b)) / 3);
    while(eps2 >= eps) {
        nt <<= 1;
        eps2 = fabs((TRAP(nt << 1, a, b) - TRAP(nt, a, b)) / 3);
    }
    It = TRAP(nt << 1, a, b) + eps2;

    eps3 = fabs((SIMPS(ns << 1, a, b) - SIMPS(ns, a, b)) / 15);
    while(eps3 >= eps) {
        ns <<= 1;
        eps3 = fabs((SIMPS(ns << 1, a, b) - SIMPS(ns, a, b)) / 
15);
    }
    Is = SIMPS(ns << 1, a, b) + eps3;

    cout << endl;
    cout << ">> Result calculated using the rectangle formula is: 
" << Ir << "         n is:" << nr << endl;
    cout << ">> Result calculated using the trapezoidal formula 
is: " << It << "       n is:" << nt << endl;
    cout << ">> Result calculated using the Simpson formula is: " 
<< Is << "           n is:" << ns << endl;

    return 0;
}
```

3. Проведём вычисления по программе, добиваясь, чтобы результат удовлетворял требуемой точности.

табл.1 — Вычислительный эксперимент

Eps	Формула прямоугольников	Формула трапеций	Формула парабол
0.1	I = 0.11315; n = 2	I = 0.112834; n = 2	I = 0.112849; n = 2
0.0001	I = 0.11293; n = 4	I = 0.11285; n = 4	I = 0.112849; n = 2
0.00001	I = 0.112871; n = 8	I = 0.112851; n = 16	I = 0.112851; n = 4
0.000001	I = 0.112852; n = 32	I = 0.112851; n = 62	I = 0.112851; n = 8
0.0000001	I = 0.112851; n = 128	I = 0.112851; n = 128	I = 0.112851; n = 8

Выводы.

Проведя вычисления по программе и проанализировав результаты, отображённые в *табл. 1*, можно сделать вывод, что число итераций, необходимых для вычисления интеграла по составным формулам прямоугольника, трапеции и Симпсона, увеличиваются с увеличением задаваемой точности результата.

Меньше всего итераций для получения необходимой точности результата требуется при вычислении интеграла по формуле Симпсона.