МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Лабораторная работа № 3

Выполнила: Яроцкас Анастасия, 2 курс, 4 группа Преподаватель: Полещук Максим Александрович

Минск

2018

1. Постановка задачи

   Для функции $f(x)$, взятой в соответствии с вариантом задания (равным номеру в списке академической группы) из лабораторной работы 1, вычислить определённый интеграл $\int_a^b p(x)f(x)dx$ по составным формулам средних прямоугольников и Симпсона с точностью ε, равной $\frac{1}{2}10^{-3}$ и $\frac{1}{2}10^{-5}$. Величину шага определить исходя из априорной оценки погрешности численного интегрирования. Уточнить значение интеграла по Ричардсону.

2. Входные данные

   $2(\pi + 3x)^{1/2}, [1,10]$

3. Формулы и краткие пояснения к ним.

3.1 Формула средних прямоугольников:

$$\int_a^b f(x)dx = h(f_{1/2} + f_{3/2} + f_{5/2} + \dots + f_{\frac{2n-3}{2}} + f_{\frac{2n-1}{2}}) + r_n, r_n = \frac{(b-a)^3}{24*n^2} * f^{(2)}(x)$$

3.3 Формула Симпсона:

$$\int_a^b f(x)dx = \frac{2}{6}(f_0 + f_n + 4(f_1 + f_3 + \dots + f_{2k-1}) + 2(f_2 + f_4 + \dots + f_{2k})) + r_n, r_n = \frac{(b-a)^5}{180*n^4} * f^{(4)}(x)$$

3.4 Оценка:

Будем использовать апостериорную оценку. Выбираем первоначально фиксированный шаг $h = 0.001$, а затем производим вычисления до тех пор, пока величина $\Theta|I_{\frac{h}{2}} - I_h|$, не станет меньше необходимой точности ε, где $\Theta = \frac{1}{3}$ или $\Theta = \frac{1}{15}$ для формул средних прямоугольников и трапеций или формулы Симпсона соответственно. I_h - значение интеграла, вычисленное по квадратурной формуле с шагом h, а - $I_{\frac{h}{2}}$ значение того же интеграла, вычисленное для шага $\frac{h}{2}$.

3.5 Уточнение по Ричардсону:

Будем использовать формулу:

$$I = \frac{2^k \times I_{\frac{h}{2}} - I_h}{2^k - 1},$$

где k = 2 для формулы средних прямоугольников, и k = 4 для формулы Симпсона.

4. Листинг программ

```
#include <iostream>
#include <math.h>

float A = 1.0;
float B = 10.0;

float func(float x) {
    return float(2 * sqrt(M_PI + 3 * x));
}

float rect(float h) {
    float int_sum = 0;
    float x = A + h/2.0;
    while (x < B) {
        int_sum += func(x);
        x += h;
    }

    return h * int_sum;
}

float simpson(float h) {
    float int_sum1 = 0;
    float int_sum2 = 0;
    float x = A + h;
    while (x < B) {
        int_sum1 += func(x);
        x += 2 * h;
    }

    x = A + 2 * h;
    while (x < B) {
        int_sum2 += func(x);
        x += 2 * h;
    }

    return h / 3 * (func(A) + func(B) + 4 * int_sum1 + 2 * int_sum2);

}

void solveRect(float h, float k, float eps) {
    float sum_h, sum_h_half, rich, e;
    while (true) {
        h /= 2;
        sum_h = rect(h);
        sum_h_half = rect(h / 2);
        rich = (pow(2, k) * sum_h_half - sum_h) / (pow(2, k) - 1);
        e = 1/3 * abs(sum_h - sum_h_half);
        if (e <= eps)
            break;
```

```
    }

    std::cout << "rect: " << eps << " " << h << " " << sum_h << " " << rich << " " << e << std::endl;
}

void solveSimpson(float h, float k, float eps) {
    float sum_h, sum_h_half, rich, e;
    while (true) {
        h /= 2;
        sum_h = simpson(h);
        sum_h_half = simpson(h / 2);
        rich = (pow(2, k) * sum_h_half - sum_h) / (pow(2, k) - 1);
        e = 1/15 * abs(sum_h - sum_h_half);
        if (e <= eps)
            break;
    }

    std::cout << "simpson: " << eps << " " << h << " " << sum_h << " " << rich << " " << e <<
std::endl;
}

int main(int argc, const char * argv[]) {
    solveRect(0.001, 2, 0.0005);
    solveRect(0.001, 2, 0.000005);
    solveSimpson(0.001, 4, 0.0005);
    solveSimpson(0.001, 4, 0.000005);
    return 0;
}
```

5. Выходные данные

```
rect: 0.0005 0.0005 78.0282 78.0681 0
rect: 5e-06 0.0005 78.0282 78.0681 0
simpson: 0.0005 0.0005 78.0313 78.0301 0
simpson: 5e-06 0.0005 78.0313 78.0301 0
```