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

ОТЧЕТ

по практической работе №7
по дисциплине «Вычислительная математика»
Тема: Численное интегрирование

Студент гр. 7383	____________________	Кирсанов А.Я.
Преподаватель	____________________	Сучков А.И.

Санкт-Петербург
2018

Цель работы.

В работе требуется, 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.

Основные теоретические положения.

Повышения точности численного интегрирования добиваются путем применения составных формул. Для этого при нахождении определенного интеграла отрезок $[a,b]$ разбивают на четное $n = 2m$ число отрезков длины $h = (b-a)/n$ и на каждом из отрезков длины $2h$ применяют соответствующую формулу. Таким способом 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:

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

формула трапеций:

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

формула Симпсона:

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

где R_1, R_2, R_3 – остаточные члены. При $n \to \infty$ приближенные значения интегралов для всех трех формул (в предположении отсутствия погрешностей округления) стремятся к точному значению интеграла.

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

$$I_{h/2} + \left| I_{h/2} - I_h \right| / 3; \tag{1}$$

для формулы трапеций:

$$I_{h/2} - \left| I_{h/2} - I_h \right| / 3; \tag{2}$$

для формулы Симпсона:

$$I_{h/2} - \left| I_{h/2} - I_h \right| / 15. \tag{3}$$

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

Постановка задачи.

Порядок выполнения:

1) Составить программы-функции для вычисления интегралов по формулам прямоугольников, трапеций и Симпсона.

2) Составить программу-функцию для вычисления подынтегральной функции.

3) Составить головную программу, содержащую оценку по Рунге погрешности каждой из перечисленных выше 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.

4) Провести вычисления по программе, добиваясь, чтобы результат удовлетворял требуемой точности.

Выполнение работы.

В задании, согласно варианту, был вычислен интеграл

$$\int_0^{\pi} x^2 e^{-x^2} dx. \qquad (4)$$

В головной программе составлены функции вычисления подынтегральной функции, функции вычисления интегралов по формулам прямоугольников, трапеций и Симпсона (1), (2) и (3) соответственно, и их оценки по Рунге. Исходный код программы представлен в приложении А.

В табл. 1, 2 и 3 представлены результаты тестирования программы расчета интеграла по формулам (1), (2), (3) соответственно. ε меняется от 0.1 до 0.0001. Точное значение интеграла – 0.443028.

Таблица 1 – Результаты вычислений интеграла (4) по формуле (2)

Значение ε	Значение интеграла (4)	Число отрезков n	Требуемая точность достигнута
0.1	0.480962	2	Да
0. 01	0.443056	4	Да
0.001	0.443056	4	Да
0.0001	0.443056	4	Да

Таблица 2 – Результаты вычислений интеграла (4) по формуле (2)

Значение ε	Значение интеграла (4)	Число отрезков n	Требуемая точность достигнута
0.1	0.404967	2	Да
0. 01	0.442965	4	Да
0.001	0.442965	4	Да
0.0001	0.442965	4	Да

Таблица 3 – Результаты вычислений интеграла (4) по формуле (3)

Значение ε	Значение интеграла (4)	Число отрезков n	Требуемая точность достигнута
0.1	0.409673	2	Да
0. 01	0.44059	4	Да
0.001	0.443113	8	Да
0.0001	0.443113	8	Да

Выводы.

В работе изучено численное интегрирование, реализованное с помощью формул прямоугольников, трапеций и Симпсона. Вычисленное по ним значение интеграла (4) равно истинному с заданной погрешностью, причем число отрезков n тем больше, чем точнее требуется вычислить интеграл. Преимуществом таких вычислений является возможность считать интегралы, которые нельзя представить в виде элементарных функций. Таким интегралом является интеграл (4).

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <iostream>
#include <cmath>
using namespace std;
double rectangle(double a, double b, int n);
double trapezium(double a, double b, int n);
double Simpson(double a, double b, int n);
double F(double x);

int main()
{
    double exact = 0.443028;
    double a = 0;
    double b = 3.1415926535;
    double eps, check, rect, tr, sm;
    int n = 1;
    cout << "Enter Eps" << endl;
    cin >> eps;
    do{
        check = abs(rectangle(a,b,2*n)-rectangle(a,b,n))/3;
        rect = rectangle(a,b,2*n) + check;
        n *= 2;
    }while(abs(rect - exact) > eps);
    cout <<"rectangle: " << rect << " n: " << n/2 << endl;
    n = 1;

    do{
        check = abs(trapezium(a,b,2*n)-trapezium(a,b,n))/3;
        tr = trapezium(a,b,2*n) - check;
        n *= 2;
    }while (abs(tr - exact) > eps);
    cout <<"trapezium: " << tr << " n: " << n/2 << endl;
    n = 1;

    do{
        check = abs(Simpson(a,b,2*n)-Simpson(a,b,n))/15;
        sm = Simpson(a,b,2*n) - check;
        n *= 2;
        cout << sm - exact << " " << n << endl;
    }while (abs(sm - exact) > eps);
    cout <<"Simpson: " << sm << " n: " << n/2 << endl;
    return 0;
```

```
}

double F(double x){
    return pow(x,2)*exp(-pow(x,2));
}

double rectangle(double a, double b, int n){
    double rect = 0;
    double h = (b-a)/n;
    for(int i = 0; i < n; i++){
        rect += F(a+i*h+h/2);
    }
    rect *= h;
    return rect;
}

double trapezium(double a, double b, int n){
    double tr = 0;
    double h = (b-a)/n;
    for(int i = 0; i < n; i++){
        tr += F(a+i*h) + F(a+(i+1)*h);
    }
    tr *= h/2;
    return tr;
}

double Simpson(double a, double b, int n){
    double sm = 0;
    double h = (b-a)/n;
    for(int i = 0; i < n; i++){
        sm += F(a+2*i*h) + 4*F(a+(2*i+1)*h) + F(a+(2*i+2)*h);
    }
    sm *= h/3;
    return sm;
}
```