|  |  |  |  |
| --- | --- | --- | --- |
| Толкованов  Николай  Романович | Лаб. Раб. №4 | Выполнение: |  |
| Численное интегрирование | Зачет: |  |

**Задача:** вычислить интеграл методами прямоугольников, трапеции, и симпсона.

**Теория:**

Численное интегрирование функции целесообразно использовать в тех случаях, когда:   
1) первообразная F(x) не может быть найдена с помощью элементарных функций;

2) F(x) является слишком сложной;

3) подынтегральная функция f(x) задана таблично или неявно. Будем рассматривать формулы приближенного вычисления интегралов

Метод Прямоугольников:

1. Считаем высоту
2. Считаем площадь прямоугольника
3. Добавляем к сумме

Метод Трапеции:

1. Считаем высоту
2. Считаем сумму
3. Считаем площадь трапеции
4. Добавляем к сумме площадь

Метод Симпсона:

1. Вычисляем высоту
2. Вычисляем сумму
3. Считаем следующую точку и добавляем к сумме
4. Умножаем сумму на высоту, деленную на 3

**Таблица:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **n** | **Левый**  **прямоугольник** | **Правый**  **прямоугольник** | **Трапеция** | **Симпсон** | **tсек** |
| 10 | 0.847212 | 0.752091 | 0.799652 | 0.804057 | 0.002 |
| 100 | 0.804445 | 0.803494 | 0.803969 | 0.80397 | 0.003 |
| 1000 | 0.804445 | 0.803494 | 0.803969 | 0.80397 | 0.004 |

**Код программы:**

*// f(x) = cos(x^2 \* cos(x)); a = 5; b = 7; n = 4, 5, 7*

#include <iostream>

#include <cmath>

#include <ctime>

using namespace std;

double myFunction(double x)

{

    return cos(pow(x, 2) \* cos(x));

}

double leftRect(double a, double b, int n) {

    double h = (b - a) / n;

    double sum = 0.0;

    for (int i = 0; i <= n - 1; i++) {

        sum += h \* myFunction(a + i \* h);

    }

    return sum;

}

double rightRect(double a, double b, int n) {

    double h = (b - a) / n;

    double sum = 0.0;

    for (int i = 1; i <= n; i++) {

        sum += h \* myFunction(a + i \* h);

    }

    return sum;

}

double trap(double a, double b, int n) {

    double h = (b - a) / n;

    double sum = myFunction(a) + myFunction(b);

    for (int i = 1; i <= n - 1; i++) {

        sum += 2 \* myFunction(a + i \* h);

    }

    sum \*= h / 2;

    return sum;

}

double simpson(double a, double b, int n) {

    double h = (b - a) / n;

    double sum = myFunction(a) + myFunction(b);

    int k;

    for (int i = 1; i <= n - 1; i++) {

        k = 2 + 2 \* ( i % 2);

        sum += k \* myFunction(a + i \* h);

    }

    sum \*= h / 3;

    return sum;

}

int main()

{

    int a = 1, b = 2;

    int n = 10;

    int start\_time, end\_time;

    start\_time = clock();

    cout << "leftRect: " << leftRect(a, b, n) << endl;

    cout << "rightRect: " << rightRect(a, b, n) << endl;

    cout << "trapeeze: " << trap(a, b, n) << endl;

    cout << "simpson: " << simpson(a, b, n) << endl;

    end\_time = clock() - start\_time;

    cout << "time: " << (float)end\_time / CLOCKS\_PER\_SEC << endl;

}

**Заключение:** мы вычислили интеграл методами прямоугольников, трапеции, и симпсона.