**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3**

**ПРИБЛИЖЕННОЕ ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ**

**(Вариант 10)**

*Выполнил студент 3 курса МОиАИС*

*Соколов Арсений*

**Задание:**

1. Вычислить приближенно с заданной точностью интеграл по формулам прямоугольников (левых, правых, центральных), трапеций и Симпсона. Величину шага определить с помощью двойного пересчета

2. Определить относительную погрешность вычислений каждого метода по формуле:

, где *I* – точное значение интеграла; – 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:

Правых:

Центральных:

Оценка погрешности:

## Метод трапеций

Оценка погрешности:

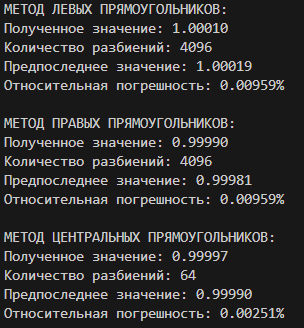
***Метод Симпсона***

Оценка погрешности:

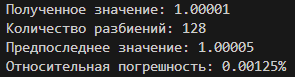
**Дано:**

|  |  |  |  |
| --- | --- | --- | --- |
| Подынтегральная функция *f*(*x*) | Заданная точность | Интервал | Первообразная  функции *F*(*x*) |
|  | 10-4 |  |  |

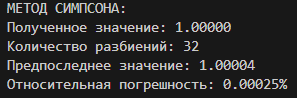
***Методы прямоугольников***

******

***Метод трапеций***

******

***Метод Симпсона***

******

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Метод прямоугольников** | | | **Метод трапеций** | **Метод Симпсона** |
| **Левых** | **Правых** | **Центральных** |
| **Полученное значение** | 1.00010 | 0.99990 | 0.99997 | 1.00001 | 1.00000 |
| **Количество разбиений** | 4096 | 4096 | 64 | 128 | 32 |
| **Предпоследнее значение** | 1.00019 | 0.99981 | 0.99990 | 1.00005 | 1.00004 |
| **Относительная погрешность (%)** | 0.00959 | 0.00959 | 0.00251 | 0.00125 | 0.00025 |

**Вывод:** метод Симпсона оказался самым эффективным. Данный метод требует наименьшее количество разбиений и дает наименьшую погрешность из предложенных методов.

**ПРИЛОЖЕНИЕ 1**

***Программа вычисления приближенного значения интеграла по формулам прямоугольников (левых, правых, центральных)***

#include <iostream>

#include <iomanip>

#include <cmath>

// Функция для интегрирования

double f(double x) {

    return 1 / (1 + std::sin(x));

}

// Метод левых прямоугольников

void leftRectangle(double a, double b, double epsilon) {

    int n = 1;

    double prevSum = 1;

    double currentSum = 0;

    while (std::abs(currentSum - prevSum) > epsilon){

        prevSum = currentSum;

        currentSum = 0;

        n \*= 2;

        double h = (b - a) / n;

        for (int i = 0; i < n; ++i) {

            currentSum += f(a + i \* h);

        }

        currentSum \*= h;

    }

    std::cout << "Полученное значение: " << currentSum << std::endl;

    std::cout << "Количество разбиений: " << n << std::endl;

     std::cout << "Предпоследнее значение: " << prevSum << std::endl;

    std::cout << "Относительная погрешность: " << std::abs(1 - currentSum) \* 100 << "%" << std::endl << std::endl;

}

// Метод правых прямоугольников

void rightRectangle(double a, double b, double epsilon) {

    int n = 1;

    double prevSum = 1;

    double currentSum = 0;

    while (std::abs(currentSum - prevSum) > epsilon){

        prevSum = currentSum;

        currentSum = 0;

        n \*= 2;

        double h = (b - a) / n;

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

            currentSum += f(a + i \* h);

        }

        currentSum \*= h;

    }

    std::cout << "Полученное значение: " << currentSum << std::endl;

    std::cout << "Количество разбиений: " << n << std::endl;

     std::cout << "Предпоследнее значение: " << prevSum << std::endl;

    std::cout << "Относительная погрешность: " << std::abs(1 - currentSum) \* 100 << "%" << std::endl << std::endl;

}

// Метод центральных прямоугольников

void centerRectangle(double a, double b, double epsilon) {

    int n = 1;

    double prevSum = 1;

    double currentSum = 0;

    while (std::abs(currentSum - prevSum) > epsilon){

        prevSum = currentSum;

        currentSum = 0;

        n \*= 2;

        double h = (b - a) / n;

        for (int i = 0; i < n; ++i) {

            currentSum += f(a + (i + 0.5) \* h);

        }

        currentSum \*= h;

    }

    std::cout << "Полученное значение: " << currentSum << std::endl;

    std::cout << "Количество разбиений: " << n << std::endl;

     std::cout << "Предпоследнее значение: " << prevSum << std::endl;

    std::cout << "Относительная погрешность: " << std::abs(1 - currentSum) \* 100 << "%" << std::endl << std::endl;

}

int main() {

    double a = 0; // Начало интервала

    double b = M\_PI / 2; // Конец интервала

    double epsilon = 0.0001; // Заданная точность

    std::cout << std::fixed << std::setprecision(5) << std::showpoint;

    std::cout << "МЕТОД ЛЕВЫХ ПРЯМОУГОЛЬНИКОВ: " << std::endl ;

    leftRectangle(a, b, epsilon);

    std::cout << "МЕТОД ПРАВЫХ ПРЯМОУГОЛЬНИКОВ: " << std::endl;

    rightRectangle(a, b, epsilon);

    std::cout << "МЕТОД ЦЕНТРАЛЬНЫХ ПРЯМОУГОЛЬНИКОВ: " << std::endl;

    centerRectangle(a, b, epsilon);

    return 0;

}

**ПРИЛОЖЕНИЕ 2**

***Программа вычисления приближенного значения интеграла методом трапеций***

#include <iostream>

#include <iomanip>

#include <cmath>

// Функция для интегрирования

double f(double x) {

    return 1 / (1 + std::sin(x));

}

// Метод трапеций

void trapezoidal(double a, double b, double epsilon) {

    int n = 1;

    double prevSum = 1;

    double currentSum = 0;

    while (std::abs(currentSum - prevSum) > epsilon){

        prevSum = currentSum;

        currentSum = 0;

        n \*= 2;

        double h = (b - a) / n;

        for (int i = 1; i < n; ++i) {

            currentSum += f(a + i \* h);

        }

        currentSum += (f(a) + f(b)) / 2;

        currentSum \*= h;

    }

    std::cout << "Полученное значение: " << currentSum << std::endl;

    std::cout << "Количество разбиений: " << n << std::endl;

     std::cout << "Предпоследнее значение: " << prevSum << std::endl;

    std::cout << "Относительная погрешность: " << std::abs(1 - currentSum) \* 100 << "%" << std::endl << std::endl;

}

int main() {

    double a = 0; // Начало интервала

    double b = M\_PI / 2; // Конец интервала

    double epsilon = 0.0001; // Заданная точность

    std::cout << std::fixed << std::setprecision(5) << std::showpoint;

    std::cout << "МЕТОД ТРАПЕЦИЙ: " << std::endl ;

    trapezoidal(a, b, epsilon);

    return 0;

}

**ПРИЛОЖЕНИЕ 3**

***Программа вычисления приближенного значения интеграла методом Симпсона***

#include <iostream>

#include <iomanip>

#include <cmath>

// Функция для интегрирования

double f(double x) {

    return 1 / (1 + std::sin(x));

}

// Метод Симпсона

void simpson(double a, double b, double epsilon) {

    int n = 2; // Начальное количество интервалов должно быть четным

    double prevSum = 1;

    double currentSum = 0;

    while (std::abs(currentSum - prevSum) > epsilon){

        prevSum = currentSum;

        currentSum = 0;

        double h = (b - a) / n;

        currentSum = f(a) + f(b);

        for (int i = 1; i < n; i += 2) {

            currentSum += 4 \* f(a + i \* h);

        }

        for (int i = 2; i < n; i += 2) {

            currentSum += 2 \* f(a + i \* h);

        }

        currentSum \*= h / 3;

        n \*= 2;

    }

    std::cout << "Полученное значение: " << currentSum << std::endl;

    std::cout << "Количество разбиений: " << n << std::endl;

     std::cout << "Предпоследнее значение: " << prevSum << std::endl;

    std::cout << "Относительная погрешность: " << std::abs(1 - currentSum) \* 100 << "%" << std::endl << std::endl;

}

int main() {

    double a = 0; // Начало интервала

    double b = M\_PI / 2; // Конец интервала

    double epsilon = 0.0001; // Заданная точность

    std::cout << std::fixed << std::setprecision(5) << std::showpoint;

    std::cout << "МЕТОД СИМПСОНА: " << std::endl ;

    simpson(a, b, epsilon);

    return 0;

}