1. Метод левых прямоугольников

fn main() {

    // Функция для интегрирования: cosh(1.3\*x) \* ln(1.4 + x) / (x^2 + 1.5)

    fn f(x: f64) -> f64 {

        x.cosh() \* (1.4 + x).ln() / (x.powi(2) + 1.5)

    }

    // Производная функции 3\*sin(x)^2 \* e^(0.2\*x) - ln(tan(0.05\*x + 1))

    fn f\_prime(x: f64) -> f64 {

        // Производная первого слагаемого: 3\*sin(x)^2\*e^(0.2\*x)

        let d\_term1 = 6.0 \* x.sin() \* x.cos() \* (0.2 \* x).exp()

                    + 0.6 \* x.sin().powi(2) \* (0.2 \* x).exp();

        // Производная второго слагаемого: -ln(tan(0.05\*x + 1))

        let inner = 0.05 \* x + 1.0;

        let d\_term2 = 0.05 / (inner.sin() \* inner.cos());

        d\_term1 - d\_term2

    }

    let a = 0.0;

    let b = 10.0;

    let n = 10000;

    let h = (b - a) / n as f64;

    // Вычисление интеграла методом левых прямоугольников

    let mut integral = 0.0;

    for i in 0..n {

        let x = a + i as f64 \* h;

        integral += f(x) \* h;

    }

    println!("Вычисленный интеграл методом левых прямоугольников: {:.6}", integral);

    // Вычисление производных в точках и поиск максимума

    let mut max\_derivative = f64::NEG\_INFINITY;

    for i in 0..=n {

        let x = a + i as f64 \* h;

        let derivative = f\_prime(x);

        if derivative > max\_derivative {

            max\_derivative = derivative;

        }

    }

    // Оценка погрешности

    let error = 0.5 \* max\_derivative \* (b - a) \* h;

    println!("Максимальная производная: {:.6}", max\_derivative);

    println!("Погрешность: {:.6}", error);

    println!("Шаг: {:.6}", h);

}

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Контент, сгенерированный ИИ, может содержать ошибки.