Прямая разностная формула первого порядка

use std::f64::consts;

fn f(x: f64) -> f64 {

    3.0 \* x.sin().powi(2) \* (0.2 \* x).exp() - (0.05 \* x + 1.0).tan().ln()

}

fn main() {

    let h = 2.57017626932225e-9;

    let x = 6.0;

    let ans = (f(x + h) - f(x)) / h;

    let pr\_ans = -5.38291344652241;

    println!("Прямая разностная формула первого порядка: {}", ans);

    println!("Принятое за достоверное значение: {}", pr\_ans);

    println!("Абсолютная погрешность: {}", (ans - pr\_ans).abs());

    println!("Относительная погрешность: {}%", (ans - pr\_ans).abs() / pr\_ans.abs() \* 100.0);

}

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Формула второго порядка аппроксимации

fn f(x: f64) -> f64 {

    3.0 \* x.sin().powi(2) \* (0.2 \* x).exp() - (0.05 \* x + 1.0).tan().ln()

}

fn main() {

    let h = 1.21979098107317e-6;

    let x = 6.0;

    let ans = (f(6. + h) - f(6. - h))/(2.\*h);

    let pr\_ans = -5.38291344652241;

    println!("Прямая разностная формула первого порядка: {}", ans);

    println!("Принятое за достоверное значение: {}", pr\_ans);

    println!("Абсолютная погрешность: {}", (ans - pr\_ans).abs());

    println!("Относительная погрешность: {}%", (ans - pr\_ans).abs() / pr\_ans.abs() \* 100.0);

}

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Формула четвёртого порядка аппроксимации

fn f(x: f64) -> f64 {

    3.0 \* x.sin().powi(2) \* (0.2 \* x).exp() - (0.05 \* x + 1.0).tan().ln()

}

fn main() {

    let h = 0.000283723160413577;

    let x = 6.0;

    let ans = (f(6. - 2.\*h) - 8.\*f(6. - h) + 8.\*f(6. + h) - f(6. + 2.\*h))/(12.\*h);

    let pr\_ans = -5.38291344652241;

    println!("Прямая разностная формула первого порядка: {}", ans);

    println!("Принятое за достоверное значение: {}", pr\_ans);

    println!("Абсолютная погрешность: {}", (ans - pr\_ans).abs());

    println!("Относительная погрешность: {}%", (ans - pr\_ans).abs() / pr\_ans.abs() \* 100.0);

}

Изображение выглядит как текст, снимок экрана, программное обеспечение, Веб-сайт

Контент, сгенерированный ИИ, может содержать ошибки.

Найти точное и приближенное значения второй производной функции индивидуального задания в точке x. Подобрать оптимальный шаг h экспериментально. Определить абсолютную, относительную погрешности.

fn f(x: f64) -> f64 {

    3.0 \* x.sin().powi(2) \* (0.2 \* x).exp() - (0.05 \* x + 1.0).tan().ln()

}

fn main() {

    let h = 1.21979098107317e-6;

    let x = 6.0;

    let ans = (f(6. + h) - 2.\*f(6.) + f(6. - h))/(h\*h);

    let pr\_ans = 14.6712412696791;

    println!("Прямая разностная формула первого порядка: {}", ans);

    println!("Принятое за достоверное значение: {}", pr\_ans);

    println!("Абсолютная погрешность: {}", (ans - pr\_ans).abs());

    println!("Относительная погрешность: {}%", (ans - pr\_ans).abs() / pr\_ans.abs() \* 100.0);

}

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Найти точное и приближенное значения третьей производной функции индивидуального задания в точке x. Подобрать оптимальный шаг h экспериментально. Определить абсолютную, относительную погрешности.

fn f(x: f64) -> f64 {

    3.0 \* x.sin().powi(2) \* (0.2 \* x).exp() - (0.05 \* x + 1.0).tan().ln()

}

fn main() {

    let h = 0.000283723160413577;

    let x = 6.0;

    let ans = (f(6. + (3.\*h/2.)) - 3.\*f(6. + (h/2.)) + 3.\*f(6. - (h/2.)) - f(6. - (3.\*h/2.)))/(h\*h\*h);

    let pr\_ans = 30.8161424293681;

    println!("Прямая разностная формула первого порядка: {}", ans);

    println!("Принятое за достоверное значение: {}", pr\_ans);

    println!("Абсолютная погрешность: {}", (ans - pr\_ans).abs());

    println!("Относительная погрешность: {}%", (ans - pr\_ans).abs() / pr\_ans.abs() \* 100.0);

}

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.