Факультет программной инженерии и компьютерной техники

Вычислительная математика

Лабораторная работа №4 – Приближение функций

Метод Эйлера

Преподаватель: Перл Ольга Вячеславовна

Выполнили: Кульбако Артемий Юрьевич P3212

Санкт-Петербург
2020

Описание метода

Метод Эйлера – одношаговый метод 1-го порядка, предназначенный для нахождения решения ОДУ по формуле:

$$y_{i+1} = y_i + hf(x_i, y_i)$$

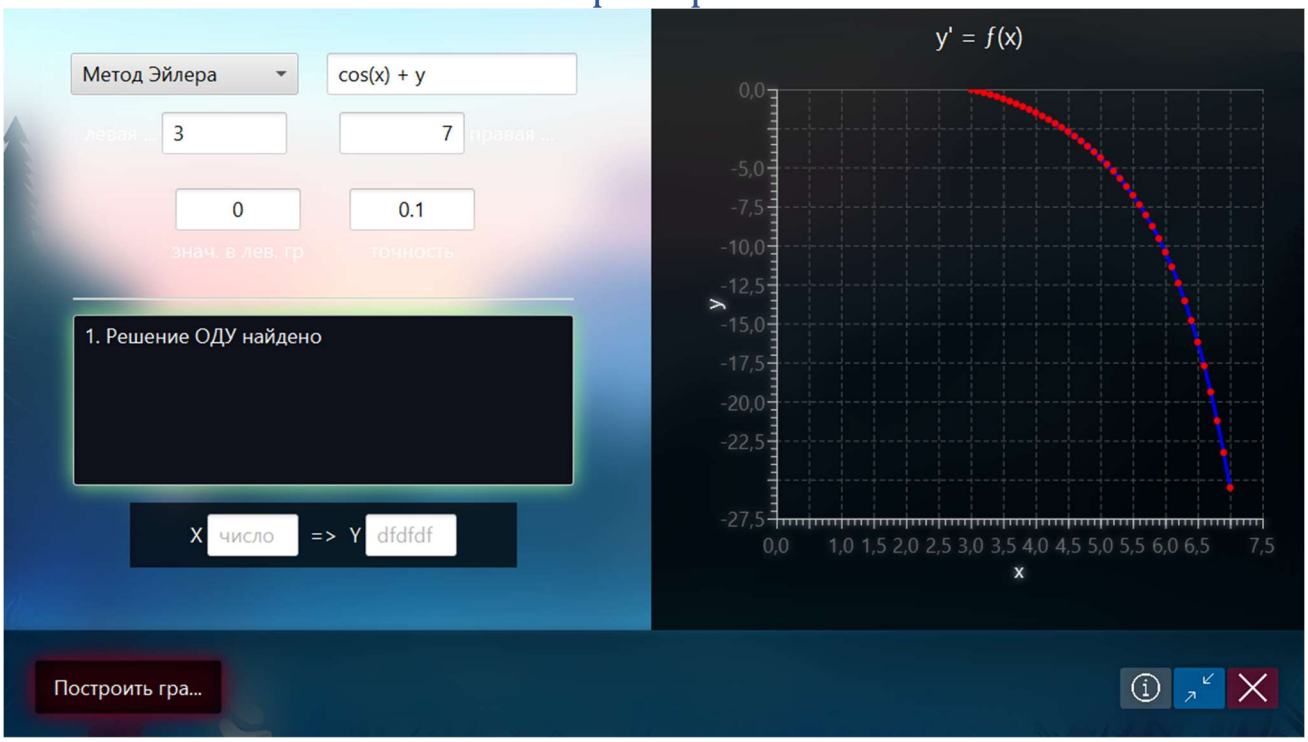
Где y_{i+1} - следующее значение сеточной функции. Находя значения y_{i+1} на интервале $[x_0..x_n]$ с шагом h мы получаем узлы, по которым можно найти необходимую функцию интерполирующим методом.

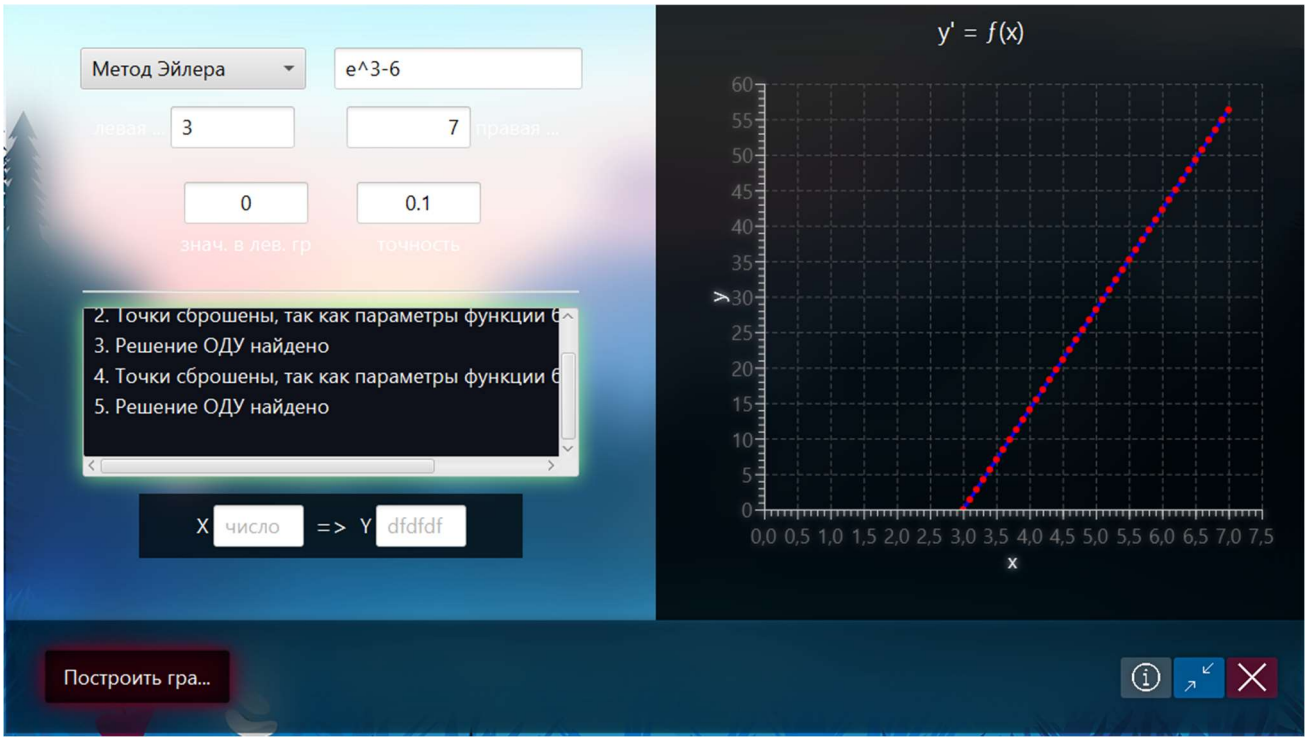
Имеет 1-ый порядок точности.

Вывод

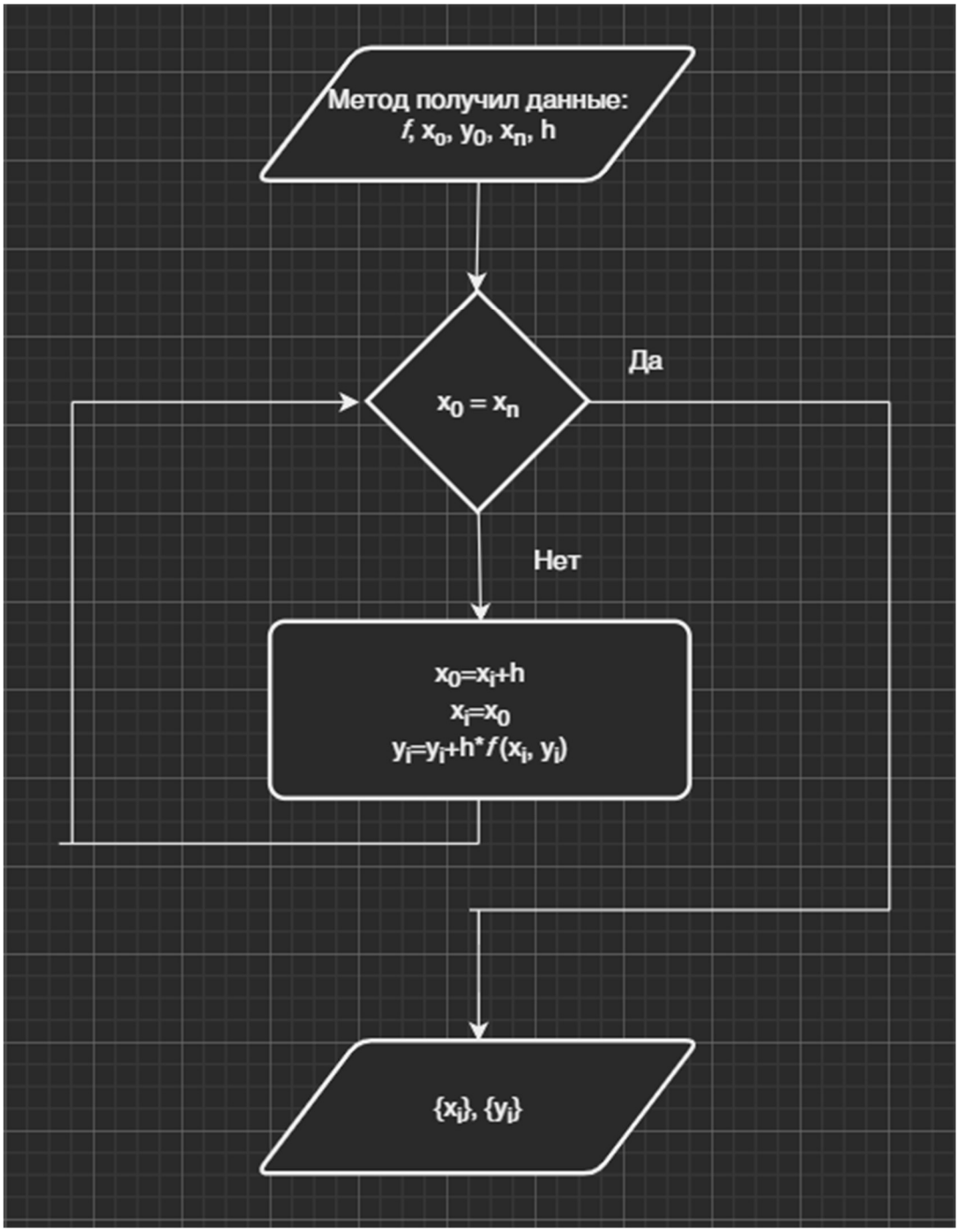
Метод	+	-
Эйлера	Крайне прост в реализации	Увеличение числа узлов ведёт к накоплению погрешности результата
Усовершенствованный Эйлера	На порядок точнее метода Эйлера	Увеличение числа узлов ведёт к накоплению погрешности результата Чуть сложнее чем метод Эйлера в реализации
Рунге-Кутта 4-го порядка	На 2 порядка точнее усов. метода Эйлера; Позволяет проводить вычисления с большим шагом	Необходимо провести значительно больше вычислений чем у 2-х предыдущих; Зависимость от метода Эйлера
Адамса	Экономичнее м. Рунге-Кутты при том же порядке точности	Невозможно изменить шаг в процессе счёта; Зависимость от любого другого метода одношагового метода
Милна	4-ый порядок точности	Зависимость от любого другого метода одношагового метода

Примеры


Метод Эйлера
cos(x) + y
левая ...
3
7
правая ...
0
0.1
знач. в лев. гр
точность
1. Решение ОДУ найдено
X
число
=> Y
dfdfdf
Построить гра...
y' = f(x)
0,0
-5,0
-7,5
-10,0
-12,5
-15,0
-17,5
-20,0
-22,5
-27,5
y
0,0 1,0 1,5 2,0 2,5 3,0 3,5 4,0 4,5 5,0 5,5 6,0 6,5 7,5
x


Метод Эйлера
e^3-6
левая ...
3
7
правая ...
0
0.1
знач. в лев. гр
точность
2. Точки сброшены, так как параметры функции б
3. Решение ОДУ найдено
4. Точки сброшены, так как параметры функции б
5. Решение ОДУ найдено
X
число
=> Y
dfdfdf
Построить гра...
y' = f(x)
60
55
50
45
40
35
30
25
20
15
10
5
0
y
0,0 0,5 1,0 1,5 2,0 2,5 3,0 3,5 4,0 4,5 5,0 5,5 6,0 6,5 7,0 7,5
x

Блок-схемы


Метод получил данные:
f, x0, y0, xn, h
x0 = xn
Да
Нет
x0=xi+h
xi=x0
yi=yi+h*f(xi, yi)
{xi}, {yi}

```kotlin
package math.differential

import javafx.geometry.Point2D
import org.mariuszgromada.math.mxparser.Expression

/**
 * Содержит методы для решения обыкновенных дифференциальных уравнений.
 * @property EULER решение ОДУ методом Эйлера.
 * @author Артемий Кульбако.
 * @version 1.1
 */
internal enum class OrdinaryDifferentialSolver {

    EULER {
        override fun solve(f: Expression, startPoint: Point2D, h: Double, interval: ClosedFloatingPointRange<Double>) =
            generateSequence(startPoint) {
                Point2D(
                    it.x + h,
                    it.y + h * f.apply {
                        this.setArgumentValue("x", it.x)
                        this.setArgumentValue("y", it.y) }.calculate().let {
                            y -> if (y.isNaN()) throw Exception("Невозможно решить введённое ОДУ") else y }
                ) }.takeWhile { it.x <= interval.endInclusive }.toList()

        override fun toString() = "Метод Эйлера"
    };

    /**
     * Решает обыкновенное дифференциальное уравнение. [f] - функция, для которой нужно вернуть
     * @return точки ОДУ на промежутке [interval] с шагом [h], где [startPoint] - x0 и y0.
     */
    abstract fun solve(f: Expression, startPoint: Point2D, h: Double, interval: ClosedFloatingPointRange<Double>): List<Point2D>
}
```