

Университет ИТМО
Факультет программной инженерии и компьютерной техники

Лабораторная работа №6
«Вычислительная математика»

Выполнил:
Студент группы Р32102
Гулямов Т.И.

Преподаватель:
Рыбаков С.Д.

Санкт-Петербург
2023

Цель лабораторной работы

Решить задачу Коши для обыкновенных дифференциальных уравнений численными методами.

Порядок выполнения работы

1. Программная реализация задачи

Листинг программы

```
func euler(_ input: MethodInput) -> [Point] {
    var x = input.interval.from
    var y = input.initialY
    var result = [Point(x: x, y: y)]
    while x <= input.interval.to {
        let nextX = x + input.stepLength
        let nextY = y + input.stepLength * input.equation(x, y)
        result.append(Point(x: nextX, y: nextY))
        x = nextX
        y = nextY
    }
    return result
}
```

```
func modifiedEuler(_ input: MethodInput) -> [Point] {
    var x = input.interval.from
    var y = input.initialY
    var result = [Point(x: x, y: y)]
    while x <= input.interval.to {
        let nextX = x + input.stepLength
        let f0 = input.equation(x, y)
        let f1 = input.equation(nextX, y + input.stepLength * f0)
        let nextY = y + input.stepLength / 2 * (f0 + f1)
        result.append(Point(x: nextX, y: nextY))
        x = nextX
        y = nextY
    }
    return result
}
```

```
func rungeKutta(_ input: MethodInput) -> [Point] {
    var x = input.interval.from
```

```

var y = input.initialY
var result = [Point(x: x, y: y)]
while x <= input.interval.to {
    let nextX = x + input.stepLength
    let k1 = input.stepLength * input.equation(x, y)
    let k2 = input.stepLength * input.equation(
        x + input.stepLength / 2,
        y + k1 / 2)
    let k3 = input.stepLength * input.equation(
        x + input.stepLength / 2,
        y + k2 / 2)
    let k4 = input.stepLength * input.equation(
        x + input.stepLength,
        y + k3)
    let nextY = y + 1 / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
    result.append(Point(x: nextX, y: nextY))
    x = nextX
    y = nextY
}
return result
}

```

```

func milne(_ input: MilneMethodInput) throws -> [Point] {
    let rungeInput = input.with(pointsCount: 4)
    let rungePoints = rungeKutta(rungeInput)
    guard input.interval.to > rungeInput.interval.to else {
        throw MilneMethodError.incorrectPointsCount
    }
    guard var point = rungePoints.last else {
        throw MilneMethodError.incorrectPointsCount
    }
    var points = rungePoints
    var fs = points.map { input.equation($0.x, $0.y) }
    while point.x < input.interval.to {
        let x = point.x + input.stepLength
        let y2 = points[back: 2].y
        let y4 = points[back: 4].y
        let f1 = fs[back: 1]
        let f2 = fs[back: 2]
        let f3 = fs[back: 3]
        var predicted = y4 + 4 / 3 * input.stepLength * (2 * f3 - f2 + 2
* f1)
        while true {

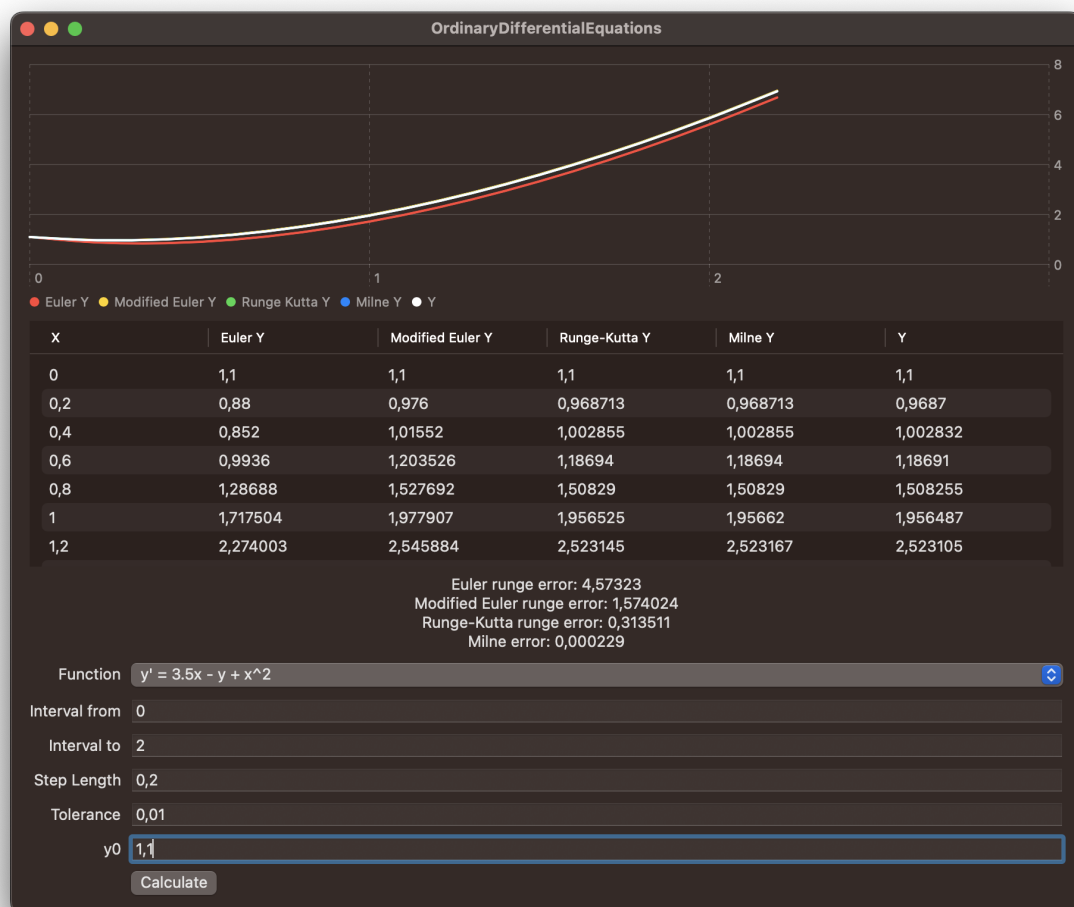
```

```

let f = input.equation(x, predicted)
let corrected = y2 + input.stepLength / 3 * (f2 + 4 * f1 + f)
if abs(predicted - corrected) > input.tolerance {
    predicted = corrected
} else {
    break
}
}
point = Point(x: x, y: predicted)
points.append(point)
fs.append(input.equation(x, predicted))
}
return points
}

```

Результаты выполнения программы



Вывод

Во время выполнения лабораторной работы познакомился с методами решения задачи Коши для обыкновенных дифференциальных уравнений. Научился использовать и реализовывать программно метод Эйлера, метод Рунге-Кутты и метод Милна. Получил ценные знания, которые несомненно пригодятся в будущем.