Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 2

по курсу «Методы оптимизации»

«Сравнение одномерных методов оптимизации»

Студент группы ИУ9-82Б Виленский С. Д.

Преподаватель Посевин Д. П.

Москва 2024

1 Задание

Проанализировать функцию на унимодальность и выпуклость. Сравнить по скорости сходимости такие методы оптимизации, как метод сходящихся отрезков, метод деления золотым сечением и метод деления числами последовательности Фибоначчи.

2 Результаты

Исходный код программы представлен в листингах 1– 3.

Листинг 1 — Нахождение минимумов функции

```
function derivativeAtPoint(f, x)
    delta = 1e-3
    return (f(x + delta) - f(x)) / delta
end

function secondDerivativeAtPoint(f, x)
    delta = 1e-3
    return (derivativeAtPoint(f, x + delta) - derivativeAtPoint(f, x)) / delta
end

function checkUnimodal(f, a, b, step, eps)
    extremum = nothing

    for x in a:step:b
        if abs(derivativeAtPoint(f, x)) < eps
            if !isnothing(extremum)
                return nothing
            end
            extremum = x
        end

        if secondDerivativeAtPoint(f, x) <= 0
            return nothing
        end
    end

    return extremum
end

function findExtrBySegments(f, a, b, step, eps)
    iters = 0

    while abs(a - b) > eps
        x1 = a + (b - a) / 3
        x2 = a + (b - a) * 2 / 3

        if f(x1) > f(x2)
```

Листинг 2 — Нахождение минимумов функции

```
            a = x1
        else
            b = x2
        end

        iters += 1
    end

    return (a + b) / 2, iters
end

function findExtrByGoldRatio(f, a, b, step, eps)
    iters = 0

    goldRatio = (5^.5 - 1) / 2
    x1 = a + (1 - goldRatio) * (b - a)
    x2 = a + goldRatio * (b - a)
    x = (a + b) / 2

    while abs(a - b) > eps
        if f(x1) > f(x2)
            x = x2
            a = x1
            x1 = x2
            x2 = a + b - x2
        else
            x = x1
            b = x2
            x2 = x1
            x1 = a + b - x1
        end

        iters += 1
    end

    return x, iters
end

function findExtrByFibbonachi(f, a, b, step, eps)
    iters = 0

    fib1, fib2, fib3 = 0, 1, 1
    for i in 1:16
        fib1 = fib2
        fib2 = fib3
        fib3 = fib1 + fib2
    end
    x1 = a + (fib1 / fib3) * (b - a)
    x2 = a + b - x1
    x = (a + b) / 2

    while abs(a - b) > eps
        if f(x1) > f(x2)
            x = x2
            a = x1
```

Листинг 3 — Нахождение минимумов функции

```
            x1 = x2
            x2 = a + b - x2
        else
            x = x1
            b = x2
            x2 = x1
            x1 = a + b - x1
        end

        iters += 1
    end

    return x, iters
end

f = x -> ((x - 9.876)^2 + 12.345)

checkUnimodal(f, -100, 100, 2e-3, 1e-3)
println(findExtrBySegments(f, -10, 10, 2e-3, 1e-3))
println(findExtrByGoldRatio(f, -10, 10, 2e-3, 1e-3))
println(findExtrByFibbonachi(f, -10, 10, 2e-3, 1e-3))
```

Результат запуска представлен в листинге 4.

Листинг 4 — Нахождение минимумов функции

```
9.876

(9.876023150046766, 25)
(9.875775199394639, 21)
(9.87616099071189, 18)
```

3 Выводы

Исходя из результатов исследования можно сделать вывод о том, что метод Фибоначи является в данном случае наиболее подходящим, судя по скорости сходимости алгоритма, в то время как метод золотого сечения показал себя лучше метода схлодящимися отрезками.