



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 № 3**  
**по курсу «Методы оптимизации»**  
**«Поиск минимума функции методом перебора и дихотомии»**

Студент группы ИУ9-82Б Гимазов А. Р.

Преподаватель Посевин Д. П.

*Mosква 2022*

# 1 Цель

Определить интервал, на котором функция является унимодальной, алгоритм определения унимодальности должен принимать на вход левую и правую точку отрезка и возвращать `false` — если функция на этом отрезке не унимодальная, в противном случае `true`.

Реализовать поиск минимума унимодальной функции на полученном интервале методом прямого перебора и дихотомии с заданной точностью по вариантам. Результат должен быть представлен на графике, точки минимизирующей последовательности должны быть выделены красным цветом, интервалы деления синим.

Точность вычисления точки минимума должна варьироваться.

# 2 Персональный вариант

$$f(x) = x^4 + 8x^3 - 6x^2 - 72x$$

### 3 Практическая реализация

Код представлен в Листинге 1.

```
using PyPlot

n = 100

function f(x)
    return x^4 + 8 * x^3 - 6 * x^2 - 72 * x
end

function makePlot(a, b)
    x = range(a; stop = b, length = n)
    y = [f(x[i]) for i=1:n]
    plot(x, y)
end

function checkUniModal(a, b)
    x = range(a; stop = b, length = n)
    y = [f(x[i]) for i=1:n]
    flag = false
    for i=2:n
        if y[i-1] < y[i]
            flag = true
        else
            if flag
                return false
            end
        end
    end
    return true
end

function findMinIter(a, b)
    x = range(a; stop = b, length = n)
    y = [f(x[i]) for i=1:n]
    minX = x[1]
    minY = y[1]
    for i=2:n
        if y[i] < minY
            minX = x[i]
            minY = y[i]
        end
    end
    return minX, minY
end
```

```

function findMinDi(a, b, eps)
    delta = eps / 2
    xs = []
    ys = []
    append!(xs, a)
    append!(ys, f(a))
    append!(xs, b)
    append!(ys, f(b))
    while abs(b-a) > eps
        x1 = (a + b - delta) / 2
        x2 = (a + b + delta) / 2
        fx1 = f(x1)
        fx2 = f(x2)
        if fx1 > fx2
            a = x1
            append!(xs, a)
            append!(ys, f(a))
        else
            b = x2
            append!(xs, b)
            append!(ys, f(b))
        end
    end
    scatter(xs, ys, color="red")
    x = (a + b) / 2
    y = f(x)
    return x, y
end

println(checkUniModal(-8, -4))
println(checkUniModal(-8, 4))

a = -8
b = -4
eps = 0.000000000001

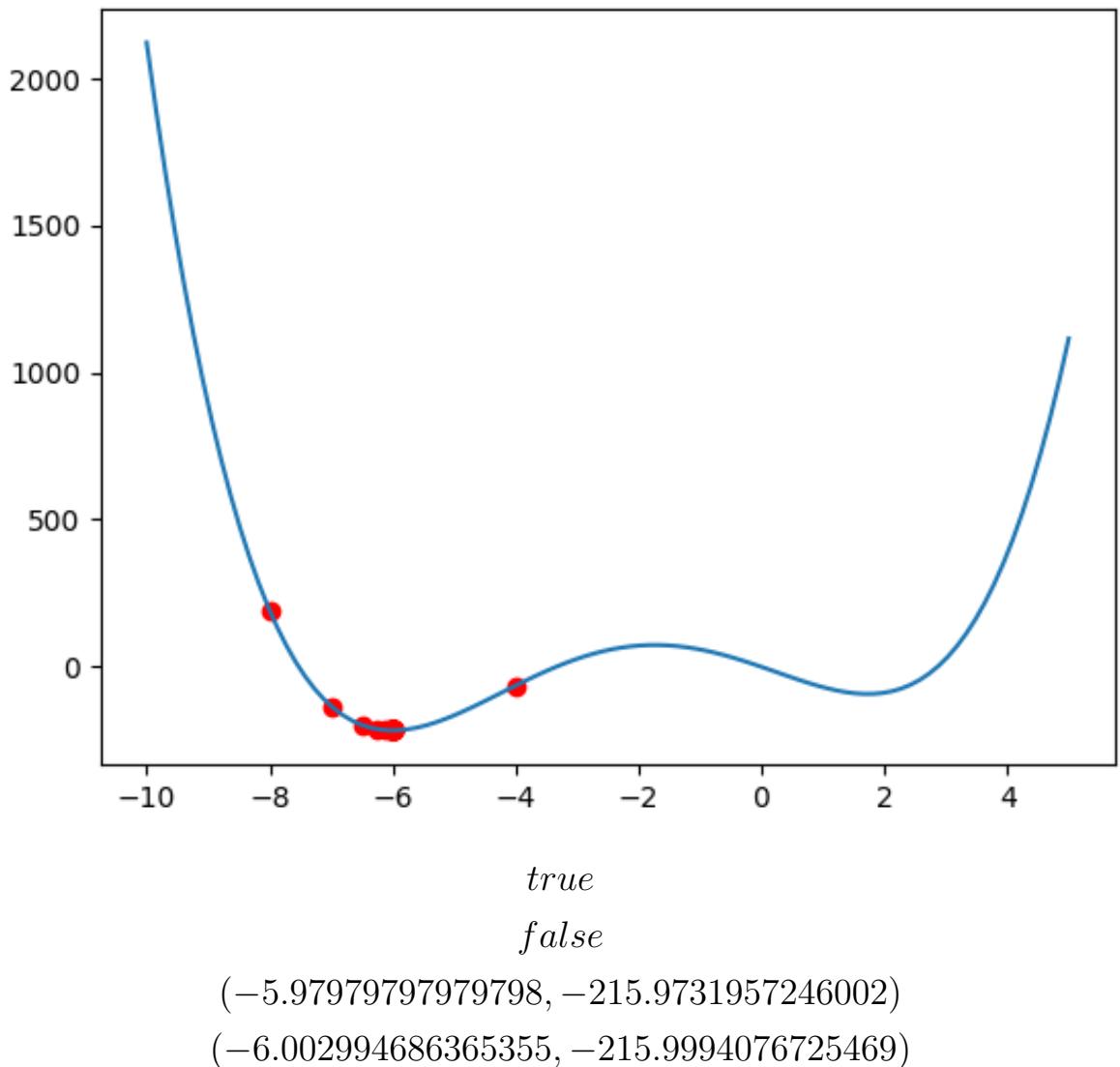
makePlot(a, b)

println(findMinIter(a, b))
println(findMinDi(a, b, eps))

```

## 4 Результаты

В результате работы программы получился следующий график:



## 5 Выводы

В данной лабораторной работе был реализован поиск минимума унимодальной функции на полученном интервале методом прямого перебора и дихотомии с заданной точностью по вариантам.