



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 № 4**  
**по курсу «Методы оптимизации»**  
**«Метод градиентного спуска»**

Студент группы ИУ9-81Б Окутин Д.А.

Преподаватель Посевин Д. П.

*Mosква 2025*

# 1 Задание

1. Реализовать метод градиентного спуска для поиска минимума функции Розенброка.

# 2 Реализация

Исходный код программы представлен в листинге 1.

Листинг 1: code

```
1
2 using Plots
3 using LinearAlgebra
4
5 plotly()
6
7
8 function approximate_gradient(f, x, h=1e-6)
9     n = length(x)
10    grad = zeros(n)
11
12    for i in 1:n
13        x_plus_h = copy(x)
14        x_plus_h[i] += h
15
16        x_minus_h = copy(x)
17        x_minus_h[i] -= h
18
19        grad[i] = (f(x_plus_h) - f(x_minus_h)) / (2*h)
20    end
21
22    return grad
23 end
24
25 function gradient_descent(f, x0, max_iterations=100000, tol=1e-6)
26     x = copy(x0)
27     xs = [copy(x0)]
28
29     learning_rate = 0.01
30
31     for i in 1:max_iterations
32         grad = approximate_gradient(f, x)
33
34         if norm(grad) < tol
```

```

35         break
36     end
37
38     x_new = x - learning_rate * grad
39
40     if f(x_new) < f(x)
41         x = x_new
42         learning_rate *= 1.05
43     else
44         learning_rate *= 0.5
45         x_new = x - learning_rate * grad
46         x = x_new
47     end
48
49     push!(xs, copy(x))
50
51     if i > 1 && norm(xs[end] - xs[end-1]) < tol
52         break
53     end
54 end
55
56 return x, xs
57 end
58
59 x = -2.5:0.1:2.5
60 y = -2.5:0.1:2.5
61
62 plt5 = surface(x, y, (x, y) -> f([x, y]), color=:thermal, alpha=0.5,
63 legend=true)
64
65 Z = map((a,b) -> f([a,b]), x_coords_grad, y_coords_grad)
66
67 plot!(plt5,
68     x_coords_grad,
69     y_coords_grad,
70     Z,
71     linecolor=:blue,
72     label="Connecting Lines"
73 )
74
75 scatter3d!(plt5, [x0[1]], [x0[2]], [f(x0)], color=:green, label="Start")
76 scatter3d!(plt5, [r[1]], [r[2]], [f(r)], color=:blue, label="End")
77
78 xlabel!(plt5, "x")
79 ylabel!(plt5, "y")
80 zlabel!(plt5, "f(x,y)")

```

```

80
81 plt1 = contour(x, y, (x, y) -> f([x, y]), 
82     color=:thermal,
83     levels=20,
84     xlabel=" x  ",
85     ylabel=" x  ",
86     title="          (2D) " ,
87     legend=:topleft ,
88     size=(700, 500)
89 )
90
91 scatter!(plt1 ,
92     [x0[1], r[1]],
93     [x0[2], r[2]],
94     color=[:green :blue],
95     markersize=8,
96     label=["          " "          "])
97 )
98
99
100 plt1 = plot!(x_coords_grad, y_coords_grad, label="
101             ", line=:blue, color=:blue, marker=:circle, markersize
102             =4)
103 display(plt5)
104 display(plt1)

```

### 3 Результаты

Результаты запуска представлены на рисунках 1.

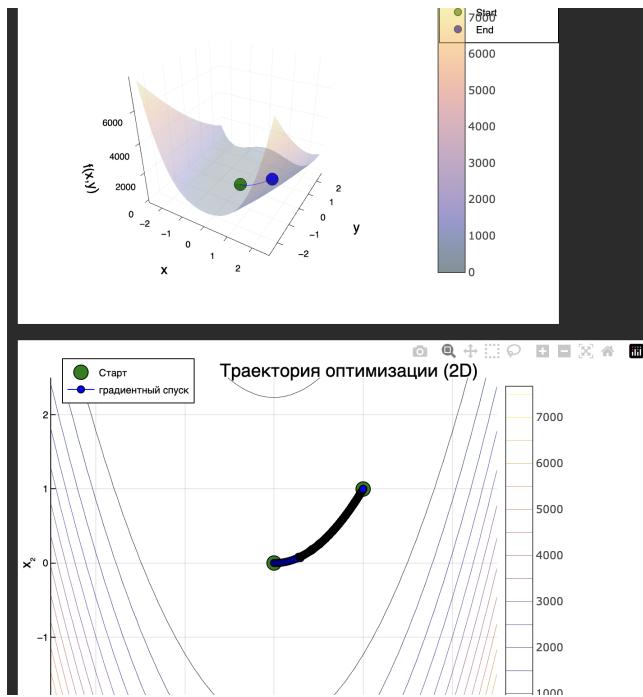


Рис. 1 — Метод радиентного спуска (Розенброк)

## **4 Выводы**

В ходе выполнения лабораторной работы был реализован метод градиентного спуска (метод оптимизации первого порядка) для поиска минимума функции. Он хорошо показал себя на овражной функции Розенброка.