**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 БЕЗУСЛОВНОГО И УСЛОВНОГО ЭКСТРЕМУМА ФУНКЦИИ**

Студент: Яровикова А. С.

Группа: ИУ9-81Б

Преподаватель: Посевин Д. П.

Москва, 2024

СОДЕРЖАНИЕ

[ЦЕЛЬ И ПОСТАНОВКА ЗАДАЧИ 3](#_Toc158799284)

[ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ 4](#_Toc158799285)

[РЕЗУЛЬТАТЫ 9](#_Toc158799286)

[ВЫВОДЫ 11](#_Toc158799287)

# **ЦЕЛЬ И ПОСТАНОВКА ЗАДАЧИ**

**Цель работы:**

Реализовать поиск безусловного и условного экстремума функции.

**Постановка задачи:**

1. Построить график целевой функции.
2. Построить график множества условия связи.
3. Построить график пересечения целевой функции и множества условия связи.
4. Найти безусловный экстремум целевой функции, при этом первую производную целевой функции и вторую производную рассчитываем численно так, как если бы мы не знали аналитический вид целевой функции и не было бы возможности получить аналитические значения первой и второй производной.
5. Найти условный экстремум, используя полином Лагранжа и анализируя значение второго дифференциала, при этом систему линейных уравнений для поиска стационарной точки полинома Лагранжа можно получить аналитически, полученную СЛАУ решаем методом Гаусса, значения вторых производных от полинома Лагранжа в стационарной точке для вычисления второго дифференциала рассчитываем численно. Анализируем знак второго дифференциала по всем направлениям , делаем вывод о типе экстремальной точки согласно теоремы 4.7 лекций.
6. Определить тип экстремума (минимум или максимум) с помощью значения определителя, состоящего из соответствующих значений первых производных функции связи и вторых производных полинома Лагранжа.

Вариант 4. Целевая функция множество условия связи

# **ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ**

График целевой функции представлен на рисунке 1.

1. Отрисовка графика целевой функции.

**using** Plots

xs = range(-**10**, stop=**10**, length=**100**)

ys = range(-**10**, stop=**10**, length=**100**)

f(x,y) = **2**\*x^**2** + **10**\*x\*y + **3**\*y^**2**

zs = [f(xi, yi) **for** xi **in** xs, yi **in** ys]

surface(xs, ys, zs, xlabel="x", ylabel="y", zlabel="f(x, y)", color=:blues, alpha=**1**, label="f(x, y)", zlims = (-**40**,**40**))

1. Отрисовка графика множества условия связи

**using** Plots

xs = range(-**10**, stop=**10**, length=**100**)

ys = range(-**10**, stop=**10**, length=**100**)

g(x,y) = **9**x + **13**y - **31**

gzs = [g(xi, yi) **for** xi **in** xs, yi **in** ys]

trace2 = surface(xs, ys, gzs, xlabel="x", ylabel="y", zlabel="f(x, y)", color=:blues, alpha=**1**, label="f(x, y)", zlims = (-**40**,**40**))

trace2

1. Построения графика пересечения целевой функции и множества условия связи

**using** Plots

# Определение функций

f(x, y) = **2**\*x^**2** + **10**\*x\*y + **3**\*y^**2**

g(x, y) = **9**x + **13**y - **31**

# Создание сетки точек

x = range(-**40**, stop = **40**, length = **100**)

y = range(-**40**, stop = **40**, length = **100**)

z = [f(i, j) - g(i, j) **for** i **in** x, j **in** y]

zs = [f(i, j) **for** i **in** x, j **in** y]

gzs = [g(i, j) **for** i **in** x, j **in** y]

# Нахождение точек пересечения

intersection\_points = []

**for** i **in** **1**:length(x)

**for** j **in** **1**:length(y)

**if** abs(z[i, j]) < **1e-8**

push!(intersection\_points, (x[i], y[j], g(x[i],y[i])))

**end**

**end**

**end**

# Отрисовка графиков и точек пересечения

surface(x, y, gzs, xlabel = "x", ylabel = "y", zlabel = "z", zlims = (-**40**,**40**), title = "Графики функций")

surface!(x, y, zs, xlabel = "x", ylabel = "y", zlabel = "z", zlims = (-**40**,**40**), title = "Графики функций")

scatter3d!([point[**1**] **for** point **in** intersection\_points], [point[**2**] **for** point **in** intersection\_points], [point[**3**] **for** \_ **in** intersection\_points], color = :red, marker = :circle, label = "Точки пересечения")

# Отрисовка линии из точек пересечения

phi(x) = (**31** - **9**\*x)/**13**

**function intersection\_line**()

intersection\_points = []

**for** x **in** -**10**:**0.001**:**10**

**for** y **in** -**10**:**0.001**:**10**

**if** phi(x) == y

push!(intersection\_points, [x, y, f(x, y)])

**end**

**end**

**end**

**return** intersection\_points

**end**

intersection\_points = intersection\_line()

x = [point[**1**] **for** point **in** intersection\_points]

y = [point[**2**] **for** point **in** intersection\_points]

z = [point[**3**] **for** point **in** intersection\_points]

scatter3d(x,y,z, color = :red)

1. Поиск безусловного экстремума

Pkg.add("SymPy")

**using** SymPy

# Объявление переменных

x, y = symbols("x y")

# Определение функции

h = **2**x^**2** + **10**x\*y + **3**y^**2**

# Вычисление частных производных

df\_dx = diff(h, x)

df\_dy = diff(h, y)

df\_dxdx = diff(diff(h, x), x)

df\_dydy = diff(diff(h, y), y)

df\_dxdy = diff(diff(h, x), y)

# Вывод результатов

println("Первая производная по x: ", df\_dx)

println("Первая производная по y: ", df\_dy)

println("Вторая производная по x: ", df\_dxdx)

println("Вторая производная по y: ", df\_dydy)

println("Смешанная производная по x по y: ", df\_dxdy)

После численного рассчета производных целевой функции, решаем систему , находим стационарные точки.

Далее проверяем стационарную точку , является ли она экстремальной.

# Подстановка значений стационарной точки x=0 y=0 (ее нашли аналитически)

extr\_points = []

push!(extr\_points, [**0**, **0**])

df\_dxdx\_point = df\_dxdx.subs([(x, **0**), (y, **0**)])

df\_dxdy\_point = df\_dxdy.subs([(x, **0**), (y, **0**)])

df\_dydy\_point = df\_dydy.subs([(x, **0**), (y, **0**)])

hessian\_point = [df\_dxdx\_point df\_dxdy\_point;

df\_dxdy\_point df\_dydy\_point]

Получаем матрицу Гессе : , используем критерий Сильвестра:

**using** LinearAlgebra

**function sylvester\_criterion**(matrix)

n = size(matrix, **1**)

**for** i **in** **1**:n

submatrix = matrix[**1**:i, **1**:i]

**if** det(submatrix) <= **0**

**return** false

**end**

**end**

**return** true

**end**

**if** sylvester\_criterion(hessian\_point)

println("Матрица положительно определенная")

**else**

println("матрица не является положительно определенной")

**end**

# главныe миноры матрицы Гессе

D1 = df\_dxdx

D2 = det(hessian\_point)

#Критерий Сильвестра

**if** D1 > **0** && D2 > **0**

println("Точка является точкой локального минимума.")

**elseif** D1 < **0** || D2 < **0**

println("Точка не является точкой локального экстремума. Седловая точка")

**end**

Получаем, что матрица не является ни положительно определенной, ни отрицательно определенной, а следовательно стационарная точка не является экстремумом. Это седловая точка.

1. Поиск условного экстремума

Для поиска условного экстремума, используем полином Лагранжа:

Далее систему линейных уравнений для поиска стационарной точки полинома Лагранжа получаем аналитически:

С помощью метода Гаусса решаем систему и находим решение:

**function gauss**(A, b)

n = length(b)

**for** k **in** **1**:n-**1**

**for** i **in** k+**1**:n

**for** j **in** k+**1**:n

A[i,j] -= (A[i,k] / A[k,k]) \* A[k,j]

**end**

b[i] -= (A[i,k] / A[k,k]) \* b[k]

**end**

**end**

#

x = similar(b, size(b))

x[n] = b[n] / A[n,n]

**for** i **in** n-**1**:-**1**:**1**

x[i] = b[i]

**for** j **in** i+**1**:n

x[i] -= A[i,j] \* x[j]

**end**

x[i] /= A[i,i]

**end**

**return** x

**end**

L(x,y,lam) = **2**\*x^**2** + **10**\*x\*y + **3**\*y^**2** + lam\*(**9**\*x+**13**\*x-**31**)

A = [**4.0** **10.0** **9.0**;**10.0** **6.0** **13.0**;**9.0** **13.0** **0.0**]

b = [**0.0** **0.0** **31.0**]

res = gauss(A, b)

res

Значения вторых производных от полинома Лагранжа в стационарной точке для вычисления второго дифференциала рассчитываем численно:

**using** SymPy

x, y = symbols("x y")

ph = **9**\*x + **13**\*y - **31**

# Вычисление частных производных

dphi\_dx = diff(ph, x)

dphi\_dy = diff(ph, y)

**function dL\_dx**( x, y,l, i)

**return** (L(x + i, y, l) - L(x - i, y, l)) / (**2**i)

**end**

**function dL\_dy**(x, y,l, i)

**return** (L(x, y + i,l) - L(x, y - i, l)) / (**2**i)

**end**

**function d2L\_dx2**(x, y,l, i)

**return** (L(x + i, y,l) - **2**\*L(x, y,l) + L(x - i, y,l)) / (i^**2**)

**end**

**function d2L\_dxdy**(x, y, l, i)

**return** (L(x + i, y+i,l) - L(x+i, y-i,l) - L(x - i, y+i,l) + L(x-i, y-i, l)) / (**4**\*i^**2**)

**end**

**function d2L\_dy2**(x, y, l,i)

**return** (L(x, y + i,l) - **2**\*L(x, y,l) + L(x, y - i,l)) / (i^**2**)

**end**

phi\_dx = dphi\_dx.subs([(x, res[**1**]), (y, res[**2**])])

phi\_dy = dphi\_dy.subs([(x, res[**1**]), (y, res[**2**])])

Lxx = d2L\_dx2(res[**1**], res[**2**], res[**3**], **0.001**)

Lxy = d2L\_dxdy(res[**1**], res[**2**], res[**3**], **0.001**)

Lyy = d2L\_dy2(res[**1**], res[**2**], res[**3**], **0.001**)

1. Определение типа экстремума с помощью значения определителя из соответствующих значений первых производных функции связи и вторых производных полинома Лагранжа:

x\_st = res[**1**]

y\_st = res[**2**]

d = -**1**((-**1**)\*phi\_dx\*(phi\_dx\*Lyy - Lxy\*phi\_dy) + phi\_dy\*(phi\_dx\*Lxy - Lxx\*phi\_dy))

**if**(d < **0**)

print("Условный максимум ($x\_st $y\_st)")

**elseif**(d > **0**)

print("условный минимум ($x\_st $y\_st)")

**else**

print("Не является экстримумом")

**end**

В итоге точка является условным максимумом.

# **РЕЗУЛЬТАТЫ**

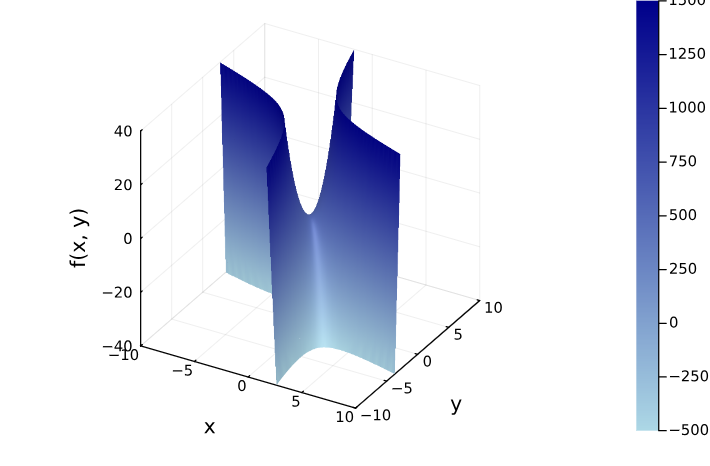


Рисунок 1 - график целевой функции

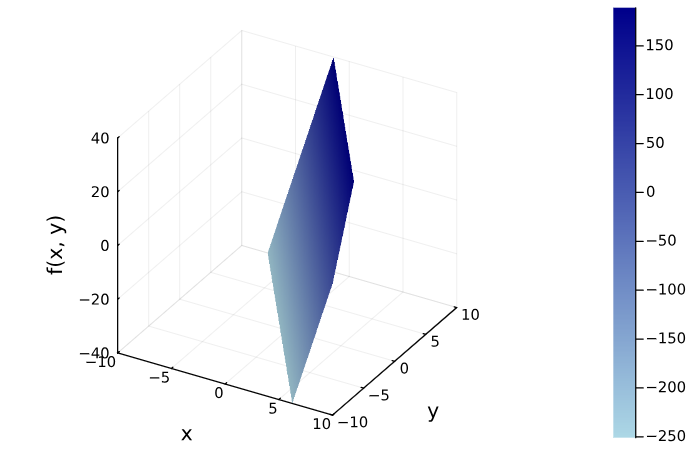


Рисунок 2 – График множества условия связи

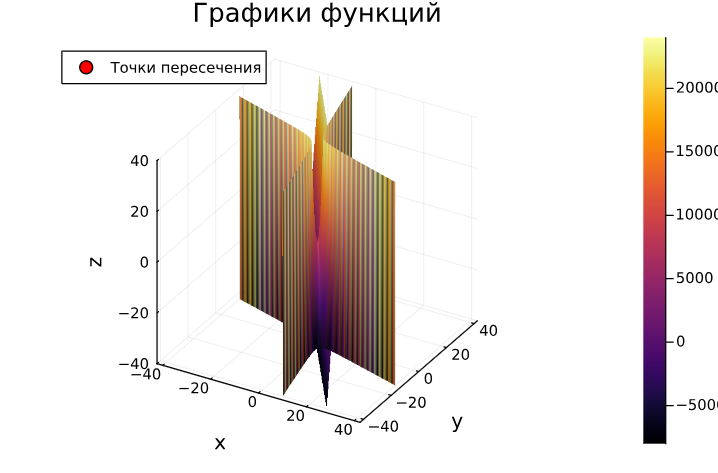


Рисунок 3 – График пересечения целевой функции и множества условия связи

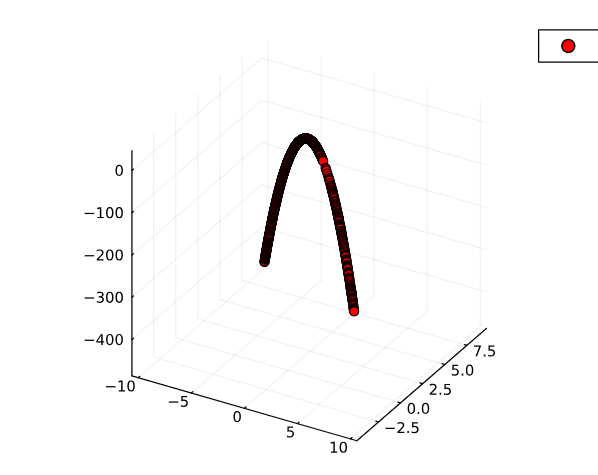


Рисунок 4 – Линия пересечения целевой функции и множества условия связи

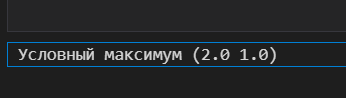


Рисунок 5 - Условный максимум

# **ВЫВОДЫ**

В ходе выполнения лабораторной работы были реализованы методы поиска условного и безусловного экстремума функции.