**Национальный исследовательский технологический университет «МИСиС»**

**Институт Информационных технологий и компьютерных наук (ИТКН)**

**Курс «Методы оптимизации»**

Лабораторная работа № 5

по теме

«Численные методы многомерной условной минимизации»

Вариант №23

Выполнил:

Студент группы БИВТ-20-1

Смирнов А.А.

Проверил:

к.ф.-м.н., доцент, Лычев А.В.

Москва, 2023

Цель: приобретение практических навыков для решения задач условной минимизации

# Ход работы:

## Задача

Вариант задания – №23.

Рассматриваемая функция – .

Начальные условия:

Требуется найти безусловный минимум функции многих переменной y = f(, …, ), то есть найти такую точку

## Графическое представление

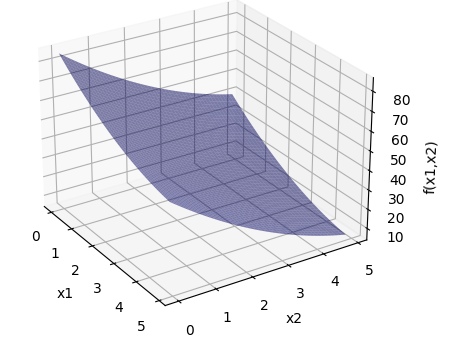


Рисунок 1 – График функции.

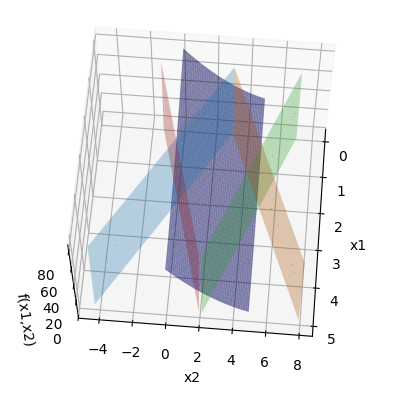


Рисунок 2а – График функции с областью ограничения.

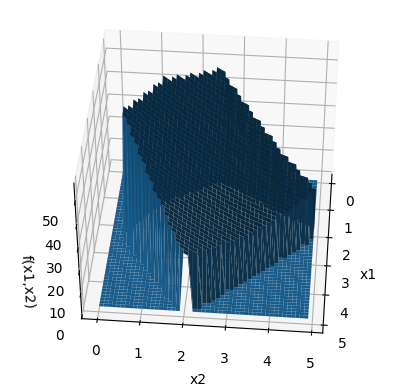


Рисунок 2б – График функции с областью ограничения.

## Листинг программы

Методы для вычисления функции и производных:

def f(x1, x2):

return (x1-6)\*\*2 +(x2-7)\*\*2

def f\_x1(x1, x2):

return 2\*x1 - 12

def f\_x2(x1, x2):

return 2\*x2 – 14

Методы для вычисления барьерных функций:

def g1(x1, x2):

return -3\*x1 - 2\*x2 + 6

def g2(x1, x2):

return -x1 + x2 - 3

def g3(x1, x2):

return x1 + x2 - 7

def g4(x1, x2):

return 2\*x1 - 3\*x2 – 4

Методы для проверки выхода за пределы области ограничений:

def g1\_bool(x1, x2):

return -3\*x1 - 2\*x2 + 6 <= 0

def g2\_bool(x1, x2):

return -x1 + x2 - 3 <= 0

def g3\_bool(x1, x2):

return x1 + x2 - 7 <= 0

def g4\_bool(x1, x2):

return 2\*x1 - 3\*x2 - 4 <= 0

def barier(x1, x2):

return (g1\_bool(x1, x2) and g2\_bool(x1, x2) and g3\_bool(x1, x2) and g4\_bool(x1, x2))

Методы для вычисления вспомогательных функций:

def F(x1, x2, r):

return f(x1,x2) + P(x1, x2, r)

def F\_x1(x1, x2, r):

return f\_x1(x1, x2) + P\_x1(x1, x2, r)

def F\_x2(x1, x2, r):

return f\_x2(x1, x2) + P\_x2(x1, x2, r)

Адаптированный метод градиентного спуска:

def P(x1, x2, r):

sum = 1/g1(x1, x2) + 1/g2(x1, x2) + 1/g3(x1, x2) + 1/g4(x1, x2)

return -r\*sum

def P\_x1(x1, x2, r):

sum = 3/(g1(x1, x2)\*\*2) + 1/(g2(x1, x2)\*\*2) - 1/(g3(x1, x2)\*\*2) - 1/(g4(x1, x2)\*\*2)

return -r\*sum

def P\_x2(x1, x2, r):

sum = 2/(g1(x1, x2)\*\*2) - 1/(g2(x1, x2)\*\*2) - 1/(g3(x1, x2)\*\*2) + 3/(g4(x1, x2)\*\*2)

return -r\*sum

Методы для вычисления функций штрафа:

def gradient(x1, x2, r):

i = F\_x1(x1, x2, r)

j = F\_x2(x1, x2, r)

return [i, j]

def module\_of\_gradient(grad):

i = 0; j = 1

return sqrt(grad[i]\*\*2 + grad[j]\*\*2)

def method\_of\_gradient\_descent\_with\_a\_constant\_step(x1, x2, e, M, r):

global counter

k = 0

counter += 1

x1\_next = x1

x2\_next = x2

while True:

counter += 2

grad = gradient(x1, x2, r)

module\_grad = module\_of\_gradient(grad)

if ((module\_grad < e) and (k >= M)):

return (x1\_next, x2\_next)

gamma = 0.1

x1\_next = x1 - gamma \* grad[0]

x2\_next = x2 - gamma \* grad[1]

counter += 2

while (F(x1\_next, x2\_next, r) - F(x1, x2, r) >= 0 or not barier(x1\_next, x2\_next)):

gamma /= 4

x1\_next = x1 - gamma \* grad[0]

x2\_next = x2 - gamma \* grad[1]

counter += 1

#print(grad, 'x1 =', x1, 'x2 =', x2, 'x1\_next =', x1\_next, 'x2\_next =', x2\_next, 'gamma =', gamma)

x1\_list.append(x1); x2\_list.append(x2)

if ((sqrt(abs(x1\_next - x1)\*\*2 + abs(x2\_next - x2)\*\*2) <= e)

& (abs(F(x1\_next, x2\_next, r) - F(x1, x2, r)) <= e)):

return (x1\_next, x2\_next)

x1 = x1\_next

x2 = x2\_next

k += 1

Метод барьерных функции:

def barrier\_function\_method(x1, x2, r, C, e, M, k):

min\_x1, min\_x2 = method\_of\_gradient\_descent\_with\_a\_constant\_step(x1, x2, e, M, r)

#print("x1 =", min\_x1, "x2 =", min\_x2)

fine = P(min\_x1, min\_x2, r)

#print("fine =", fine)

if (abs(fine) <= e):

return [(round(min\_x1, round\_num),

round(min\_x2, round\_num),

round(f(min\_x1, min\_x2), round\_num)),

k]

k += 1

r = r/C

return barrier\_function\_method(min\_x1, min\_x2, r, C, e, M, k)

## Результаты вычислений



## Графическое представление траектории

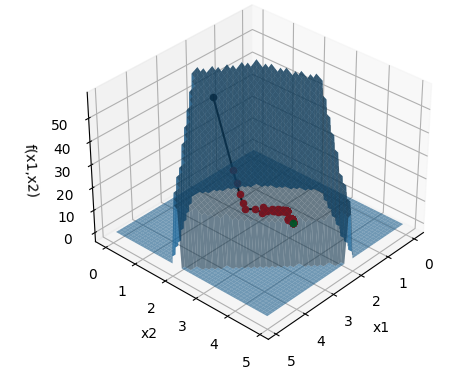


Рисунок 3 – Траектория движения.

## Сравнительная характеристика

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Начальная точка | Погрешность | Число итераций | Оптимальное решение | Оптимальное значение функции |
| (2;2) | 0.001 | 7 | (3.046, 3.954) | (18.005) |
| (2; 2) | 0.0001 | 9 | (3.0063, 3.9937) | (18.0003) |
| (2.5; 1) | 0.001 | 7 | (3.046, 3.954) | (18.005) |
| (2.5; 1) | 0.0001 | 9 | (3.0063, 3.9937) | (18.0003) |

Вывод: в результате выполнение лабораторной работы я приобрел практические навыки для решения задач условной минимизации. Для заданной функции нашел минимум в заданной области методом барьерных функции. Для решение вспомогательной задачи я взял метод градиентного спуска.