Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

**ОПТИМИЗАЦИЯ ФУНКЦИЙ ДВУХ ПЕРЕМЕННЫХ**

**Отчёт по лабораторной работе №4**

**По дисциплине**

**«Методы оптимизации»**

Студент гр. 430-2:

А.А. Лузинсан

« » 2022 г.

Проверил:

к.т.н, доцент каф. АСУ   
(должность уч.степень, уч.звание)

А.А. Шелестов

« » 2022 г.

Томск 2022

Оглавление

[ВВЕДЕНИЕ 3](#_Toc121085000)

[1 ТЕОРИЯ 4](#_Toc121085001)

[1.1 Метод Коши 4](#_Toc121085002)

[2 ОПИСАНИЕ АЛГОРИТА 5](#_Toc121085003)

[2.1 Метод Коши 5](#_Toc121085004)

[ЗАКЛЮЧЕНИЕ 6](#_Toc121085005)

[ЛИСТИНГ ПРОГРАММЫ 7](#_Toc121085006)

# **ВВЕДЕНИЕ**

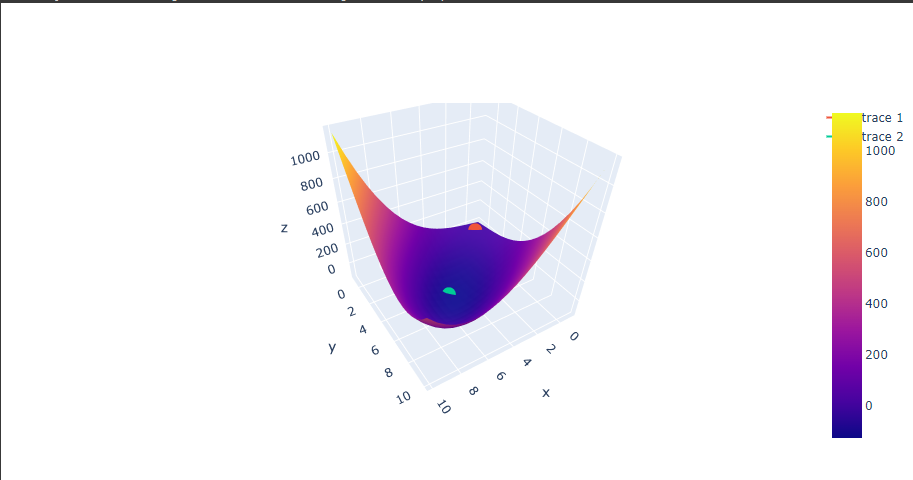
Задание: найти минимум функции двух переменных, используя метод Коши.

Точность: ε = .

Вариант задания:

2)

Вид исходной функции представлен на рисунке 1.1.

Рисунок 1.1 – Изображение исходной функции

# **1** **ТЕОРИЯ**

# **1.1 Метод Коши**

Суть метода: движение в направлении антиградиента путем осуществления одномерной минимизации. Подробнее о методе описано в алгоритме ниже.

# **2** **ОПИСАНИЕ АЛГОРИТА**

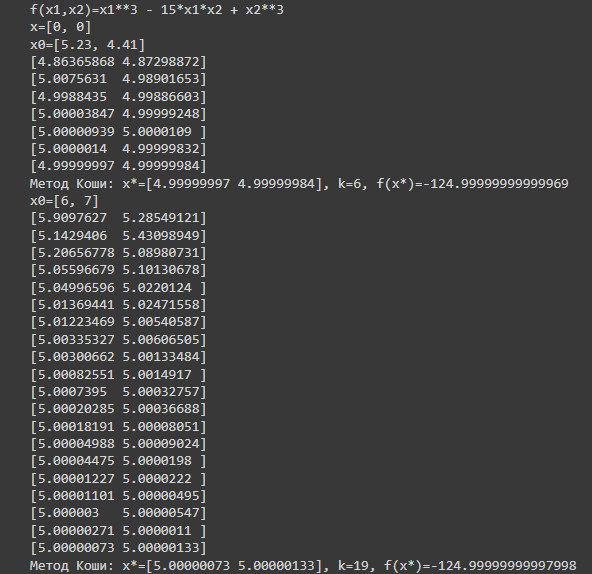
В результате применения градиентного метода над заданной функцией были получены результаты, представленные на рисунке 2.1.

Рисунок 2.1 – Результат применения метода Коши

## 2.1 Метод Коши

Алгоритм:

1. Выбрать начальную точку x0
2. На k -ой итерации, где dk=-∇f(xk), найти такое λk, что

Положить

1. Проверка критерия останова: да: окончание поиска = конец. нет: k = k +1, => переходим на пункт 2.

# ЗАКЛЮЧЕНИЕ

Я изучила методы нахождения минимумов функций двух переменных и нашла минимум функции двух переменных заданного варианта, используя градиентный метод: метод Коши.

# ЛИСТИНГ ПРОГРАММЫ

import plotly.express as px

import plotly.graph\_objects as go

import sympy

import numpy as np

def Cauchy(f, x, eps, df):

k = 0

startla = 0.02

la = startla

while True:

d = np.array([dif(\*x) for dif in df])

temp = x - la \* d

while True:

oldla = la

old = temp

la \*= 1.5

temp = x - la \* d

if f(\*temp) > f(\*old):

a, b = oldla, la

while abs(b - a) > eps \* 2:

y = (a + b - eps) / 2

z = (a + b + eps) / 2

if f(\*(x - y \* d)) <= f(\*(x - z \* d)):

b = z

else:

a = y

temp = x - ((a + b) / 2) \* d

break

print(temp)

if np.sqrt((d \*\* 2).sum()) < eps and np.sqrt(((f(\*temp) - f(\*x)) \*\* 2).sum()):

x = temp

break

x = temp

la = startla

k += 1

return x, k, f(\*x)

def lab4():

x1 = sympy.Symbol('x1')

x2 = sympy.Symbol('x2')

f = x1 \*\* 3 + x2 \*\* 3 - 15 \* x1 \* x2

print('f(x1,x2)={}'.format(f))

\_x = [0, 0]

\_x0 = [5.23, 4.41]

print('x={}\nx0={}'.format(\_x, \_x0))

eps = 1e-4

foo = sympy.lambdify([x1, x2], f, 'numpy')

print("Метод Коши: x\*={}, k={}, f(x\*)={}".format(

\*Cauchy(foo, np.array(\_x0,dtype=np.float64), eps, [sympy.lambdify([x1, x2], f.diff(arg), 'numpy') for arg in [x1, x2]])))

\_x0 = [6,7]

print('x0={}'.format(\_x0))

print("Метод Коши: x\*={}, k={}, f(x\*)={}".format(

\*Cauchy(foo, np.array(\_x0,dtype=np.float64), eps, [sympy.lambdify([x1, x2], f.diff(arg), 'numpy') for arg in [x1, x2]])))

x = np.linspace(-1, 10, 55)

y = np.linspace(-1, 10, 55)

X, Y = np.meshgrid(x, y)

Z = foo(X, Y)

fig = go.Figure(data=[go.Surface(x=X, y=Y, z=Z)])

fig.add\_trace(go.Scatter3d(x=[\_x[0]], y=[\_x[1]], z=[foo(\*\_x)]))

fig.add\_trace(go.Scatter3d(x=[\_x0[0]], y=[\_x0[1]], z=[foo(\*\_x0)]))

fig.show()