Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМУПРАВЛЕНИЯ 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

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

Точность: ε = 10^4.

Вариант задания:

2) $f(x) = x_1^3 + x_2^3 - 15x_1x_2$

$\bar{x}(0;0)$; $\bar{x}^0 = (5{,}23; 4{,}41)$

Вид исходной функции представлен на рисунке 1.1.

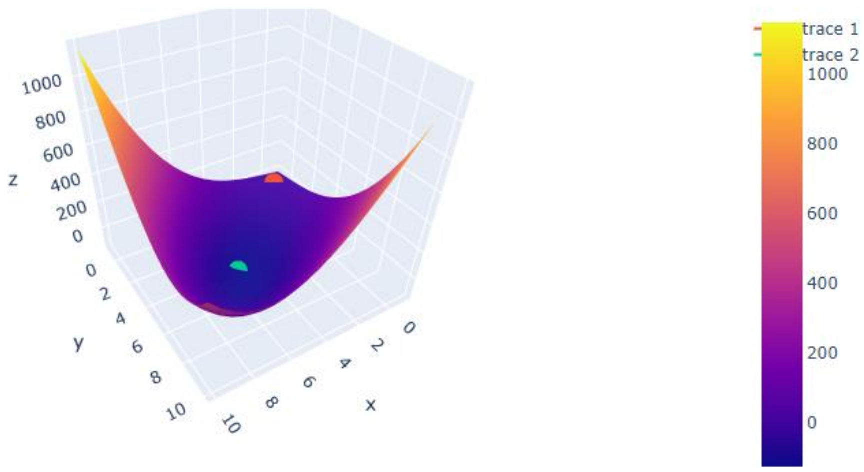

Рисунок 1.1 – Изображение исходной функции

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.

```
f(x1,x2)=x1**3 - 15*x1*x2 + x2**3
x=[0, 0]
x0=[5.23, 4.41]
[4.86365868 4.87298872]
[5.0075631  4.98901653]
[4.9988435  4.99886603]
[5.00003847 4.99999248]
[5.00000939 5.0000109 ]
[5.0000014  4.99999832]
[4.99999997 4.99999984]
Метод Коши: x*=[4.99999997 4.99999984], k=6, f(x*)=-124.99999999999969
x0=[6, 7]
[5.9097627  5.28549121]
[5.1429406  5.43098949]
[5.20656778 5.08980731]
[5.05596679 5.10130678]
[5.04996596 5.0220124 ]
[5.01369441 5.02471558]
[5.01223469 5.00540587]
[5.00335327 5.00606505]
[5.00300662 5.00133484]
[5.00082551 5.0014917 ]
[5.0007395  5.00032757]
[5.00020285 5.00036688]
[5.00018191 5.00008051]
[5.00004988 5.00009024]
[5.00004475 5.0000198 ]
[5.00001227 5.0000222 ]
[5.00001101 5.00000495]
[5.000003   5.00000547]
[5.00000271 5.0000011 ]
[5.00000073 5.00000133]
Метод Коши: x*=[5.00000073 5.00000133], k=19, f(x*)=-124.99999999997998
```

Рисунок 2.1 – Результат применения метода Коши

2.1 Метод Коши

Алгоритм:

1. Выбрать начальную точку x^0

2. На k -ой итерации, где $d_k=-\nabla f(x^k)$, найти такое λ_k, что

$$f(x^k + \lambda_k d_k) = \min_{\lambda \geq 0} f(x^k + \lambda d_k)$$

Положить $x^{k+1} = x^k + \lambda_k d_k$.

3. 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()
```