МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Лабораторная работа №3

по математическому программированию

“Методы оптимизации первого и второго порядков”

Студент Станиславчук С. М.

Группа АС-21-1

Руководитель Качановский Ю. П.

Липецк 2023 г.

Содержание

2. Задание кафедры, соответствующее варианту, номер варианта

3. Уравнение функции, подлежащей минимизации

4. Аналитическое выражение градиента

5. График поверхности минимизируемой функции

6. Графики линий уровня функции с указанными на них траекториями минимизации всеми методами первого и второго порядка.

7. Таблица результатов минимизации функции вышеперечисленными методами.

8. Выводы о траекториях минимизации полученными различными методами

2. Задание.

1) Написать условие задачи и аналитическое выражение для градиента.

2) Используя программу optimization решить задачу методом Коши,

используя различные методы нахождения шага: метод квадратичной

интерполяции, метод кубической интерполяции и метод первого

приемлемого значения.

3) Используя программу optimization, решить задачу методами Флетчера-Ривса и Полака-Рибьера, DFP и BFGS, Ньютона-Рафсона и Марквардта.

4) Представить результаты решения задачи различными методами в

таблице.

5) Сделать выводы о влиянии способа отыскания шага на ход решения

задачи.

6) Сравнить результаты, полученные методами первого порядка,

методами переменной метрики и методами второго порядка

**1) Написать условие задачи и аналитическое выражение для градиента**

Номер задачи: 4

Номер варианта данной задачи: 2

**Задача №4**

Эллиптический параболоид и плоскость пересекаются в точке (, ). Определить будет ли данная точка точкой минимума этого параболоида:

найти min

Значения коэффициентов:

; q = 0

Функция с подставленными значениями коэффициентов имеет вид:

Аналитическое выражение для градиента функции найдем, определив значения производных функции по каждой из её переменных

df/dx\_1 = x\_1^2 – 8\*x\_1 + 16 = 2\*x\_1 - 8

df/dx\_2 = 2\*x\_2

▼f = [df/dx\_1, df/dx\_2]

▼f = [2\*x\_1 - 8, 2\*x\_2] - это и есть аналитическое выражение для градиента функции.

**2. Используя программу optimization, решить задачу методом Коши**

Установим начальную точку (1, 1).

Решение задачи методом Коши, используя метод квадратичной интерполяции:

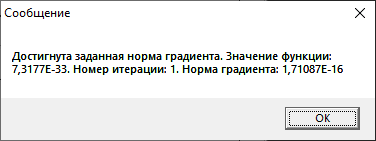


Рисунок 1 – сообщение о результате решения задачи методом Коши, используя метод квадратичной интерполяции.

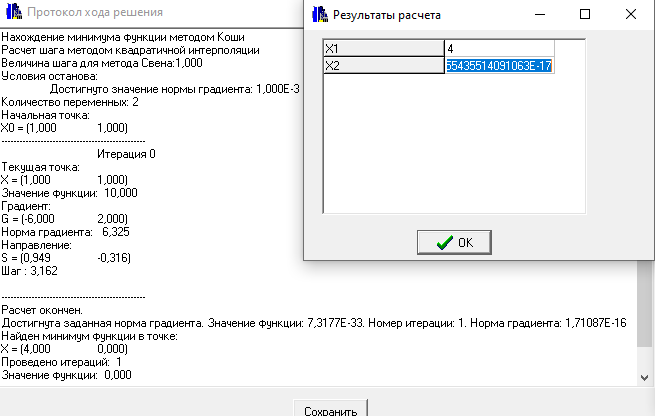


Рисунок 2 – результаты расчета точек минимума функции методом Коши, используя метод квадратичной интерполяции.

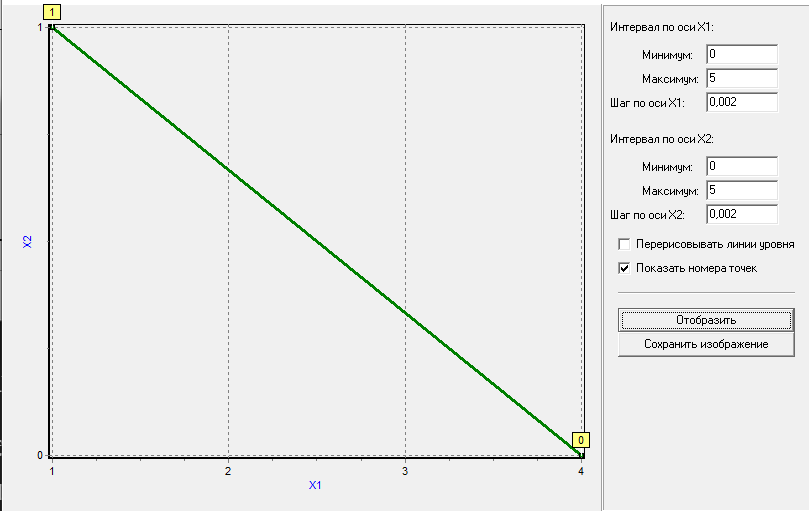


Рисунок 3 – траектория оптимизации методом Коши, используя метод квадратичной интерполяции.

Решение задачи методом Коши, используя метод кубической интерполяции:

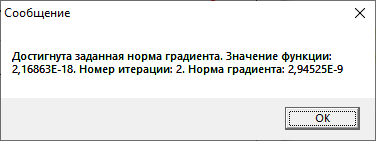


Рисунок 4 – сообщение о результате решения задачи методом Коши, используя метод кубической интерполяции.

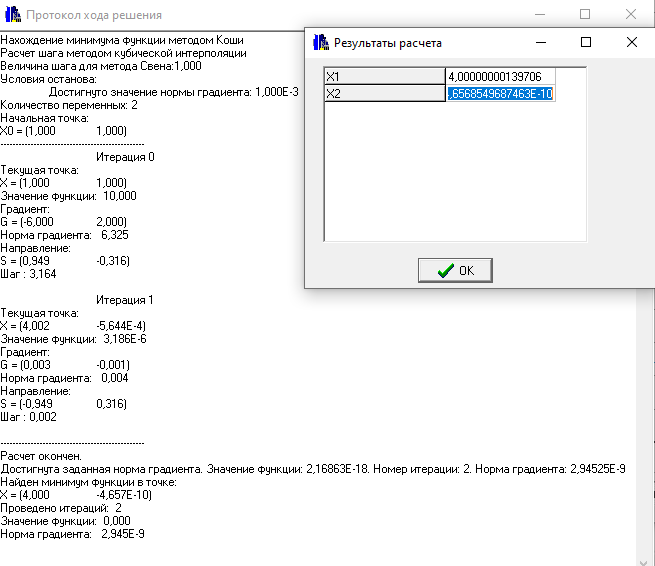


Рисунок 5 – результаты расчета точек минимума функции методом Коши, используя метод кубической интерполяции.

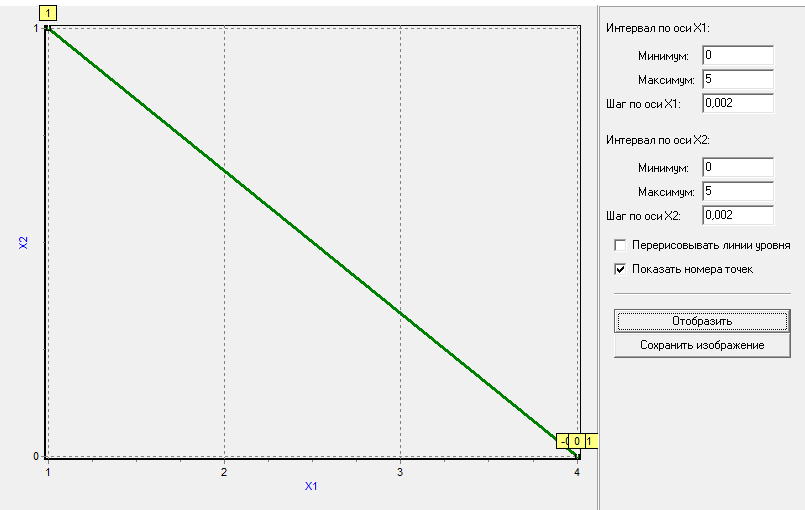


Рисунок 6 – траектория оптимизации методом Коши, используя метод кубической интерполяции.

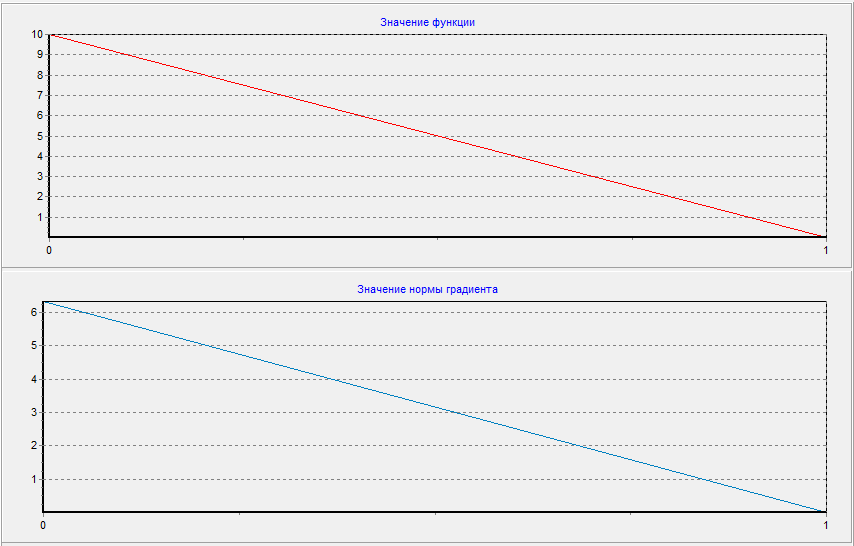


Рисунок 7 – графики значений функции и нормы градиента, полученных методом Коши, используя метод кубической интерполяции

Решение задачи методом Коши, используя метод первого

приемлемого значения:

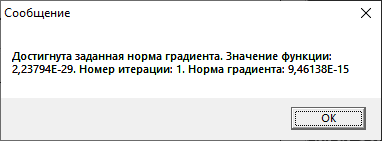


Рисунок 8 - результаты расчета точек минимума функции методом Коши, используя метод первого приемлемого значения

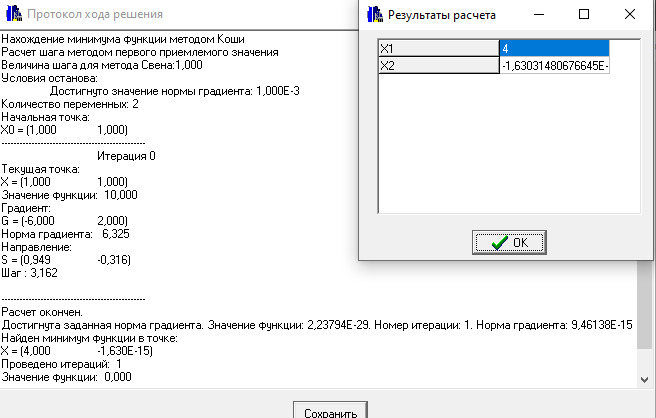


Рисунок 9 – результаты расчета точек минимума функции методом Коши, используя метод первого приемлемого значения

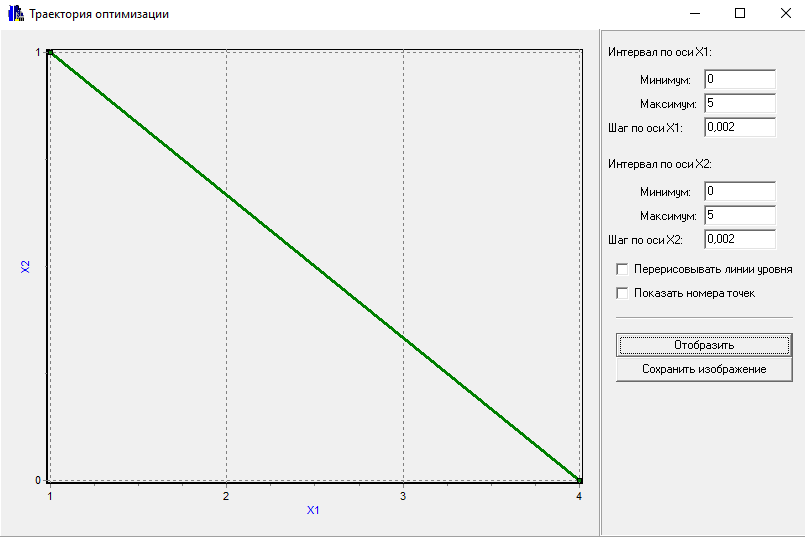


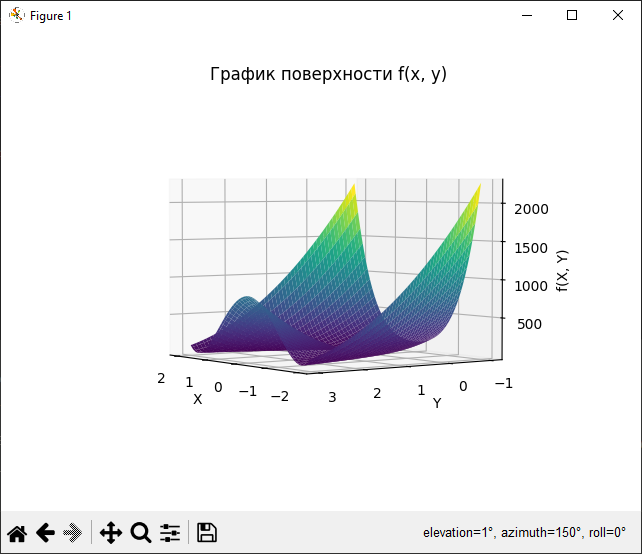
Рисунок 10 – траектория оптимизации методом Коши, используя метод первого приемлемого значения

3) Используя программу optimization, решить задачу

4. Уравнение функции, подлежащей минимизации:

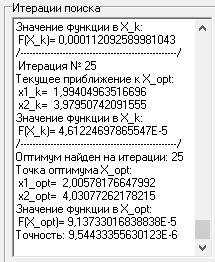
5. График поверхности функции f(x,y):

Для построения графика поверхности функции использовалась библиотека Python Matplotlib в сочетании с библиотекой NumPy для создания сетки точек (x, y) и вычисления значений функции на этой сетке.

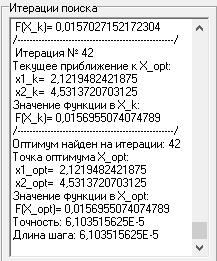


С помощью программы MoDS найдем минимум функции следующими методами:

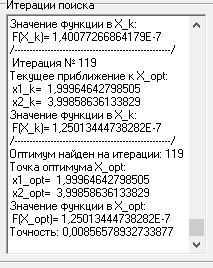
1. Метод Нелдера-Мида:



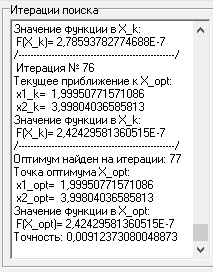
2. Метод Хука-Дживса:



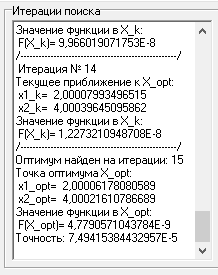
3. Метод покоординатного спуска:



4. Метод Пауэла:



5. Метод Розенброка:



6. Для отображения графиков линий уровня функции с указанными на них траекториями минимизации, напишем Python-программу, которая считает значения x1\_k, x2\_k, x1\_opt, x2\_opt из сгенерировавшихся в программе MoDS .txt файлов.

import re

def extract\_values\_from\_file(file\_path):

with open(file\_path, 'r') as file:

content = file.read()

x1\_k = re.findall(r'x1\_k=\s\*([\d.-]+)', content)

x2\_k = re.findall(r'x2\_k=\s\*([\d.-]+)', content)

x1\_opt = re.search(r'x1\_opt=\s\*([\d.-]+)', content).group(1)

x2\_opt = re.search(r'x2\_opt=\s\*([\d.-]+)', content).group(1)

return x1\_k, x2\_k, x1\_opt, x2\_opt

file\_paths = ["Результат (Нелдер-Мид).txt", "Результат (Пауэл).txt", "Результат (Покоординатный спуск).txt",

"Результат (Розенброк).txt", "Результат (Хук-Дживс).txt"]

methods\_name = ["Нелдер-Мид", "Пауэл", "Покоординатный спуск", "Розенброк", "Хук-Дживс"]

current\_method\_index = 4

method\_name = methods\_name[current\_method\_index]

file\_path = file\_paths[current\_method\_index]

x1\_k, x2\_k, x1\_opt, x2\_opt = extract\_values\_from\_file(file\_path)

x1\_k = [float(x.replace(',', '.')) for x in x1\_k]

x2\_k = [float(x.replace(',', '.')) for x in x2\_k]

x1\_opt = float(x1\_opt.replace(',', '.'))

x2\_opt = float(x2\_opt.replace(',', '.'))

print("x1\_k:", x1\_k)

print("x2\_k:", x2\_k)

print("x1\_opt:", x1\_opt)

print("x2\_opt:", x2\_opt)

import matplotlib.pyplot as plt

import numpy as np

# Целевая функция

def target\_function(x1, x2):

return 90 \* (x2 - x1 \*\* 2) \*\* 2 + (2 - x1) \*\* 2

# Сетка точек для построения линий уровня

x1 = np.linspace(-3, 3, 400)

x2 = np.linspace(-3, 5, 400)

X1, X2 = np.meshgrid(x1, x2)

Z = target\_function(X1, X2)

plt.contour(X1, X2, Z, levels=np.logspace(-1, 3, 10), alpha = 0.5)

plt.plot(x1\_k, x2\_k, marker='o', label=method\_name, color='red')

plt.scatter(x1\_opt, x2\_opt, color='red', marker='x', s=100, label=f'Оптимум {method\_name} ({x1\_opt} : {x2\_opt})')

plt.xlabel('x1')

plt.ylabel('x2')

plt.title('Линии уровня и траектория минимизации')

plt.legend()

# Путь к папке, в которой сохранятся изображения

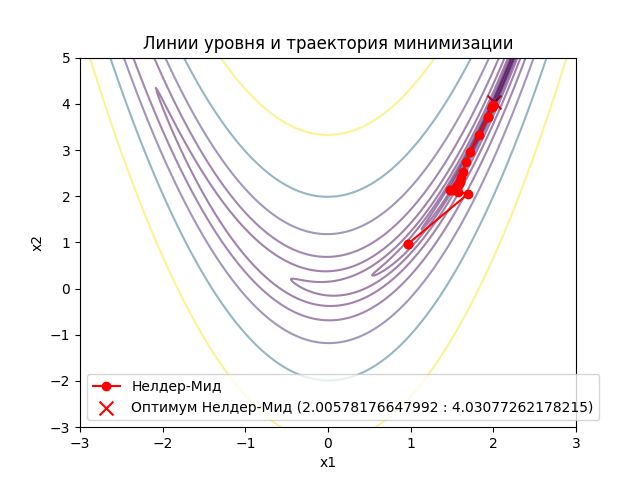
save\_folder = "D:/Программер/3 курс 1/Математическое программирование/лр 2/Результаты/png/automated"

file\_name = f"{method\_name}.png"

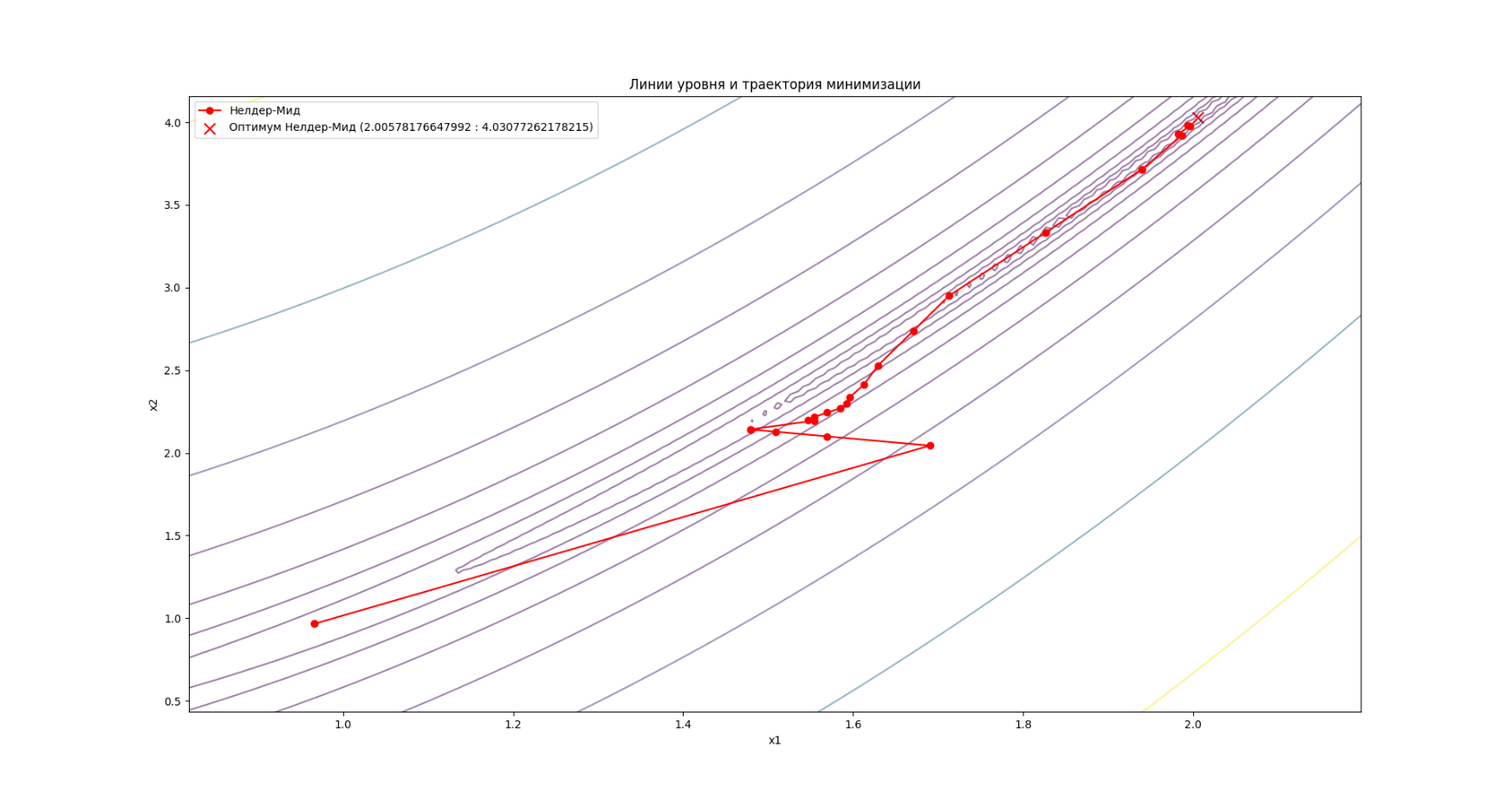
plt.savefig(f"{save\_folder}/{file\_name}")

plt.show()

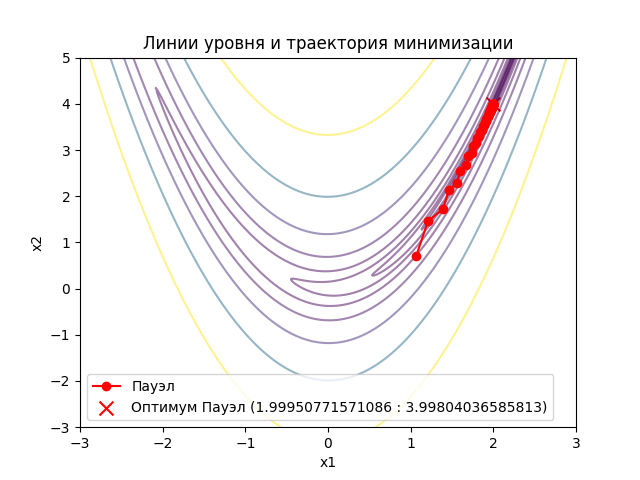
Результаты:



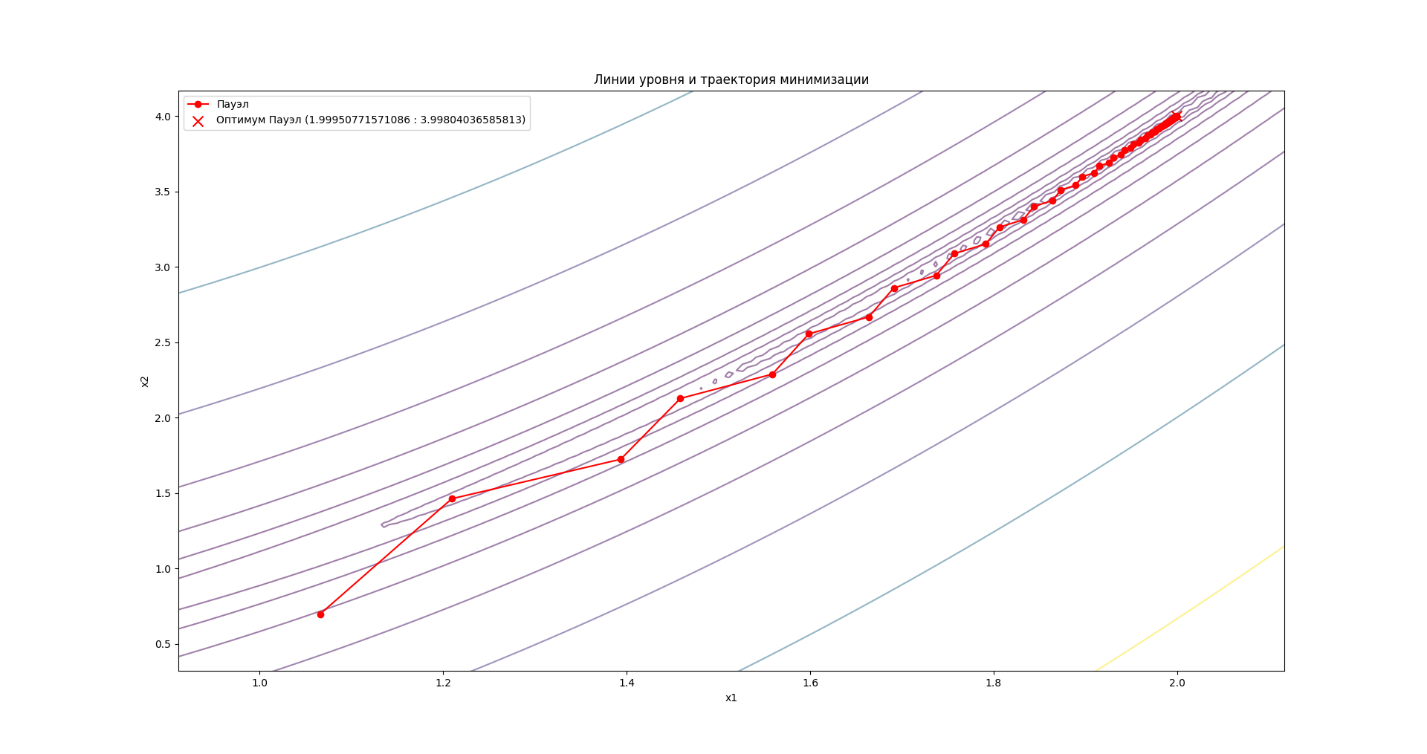
Линии уровня и траектория минимизации методом **Нелдера-Мида**



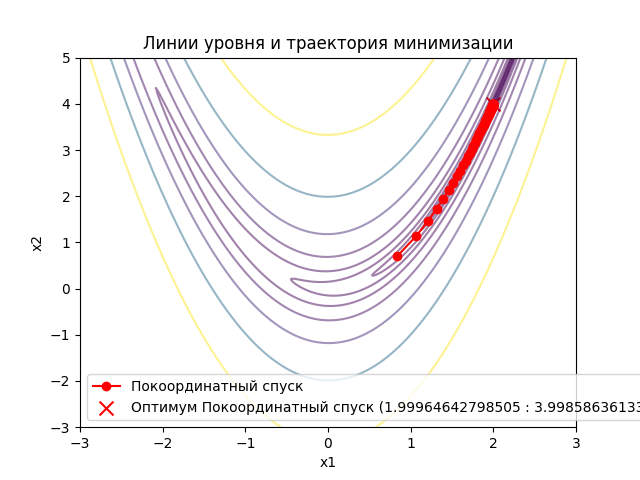
Линии уровня и траектория минимизации методом **Нелдера-Мида** (приближено)



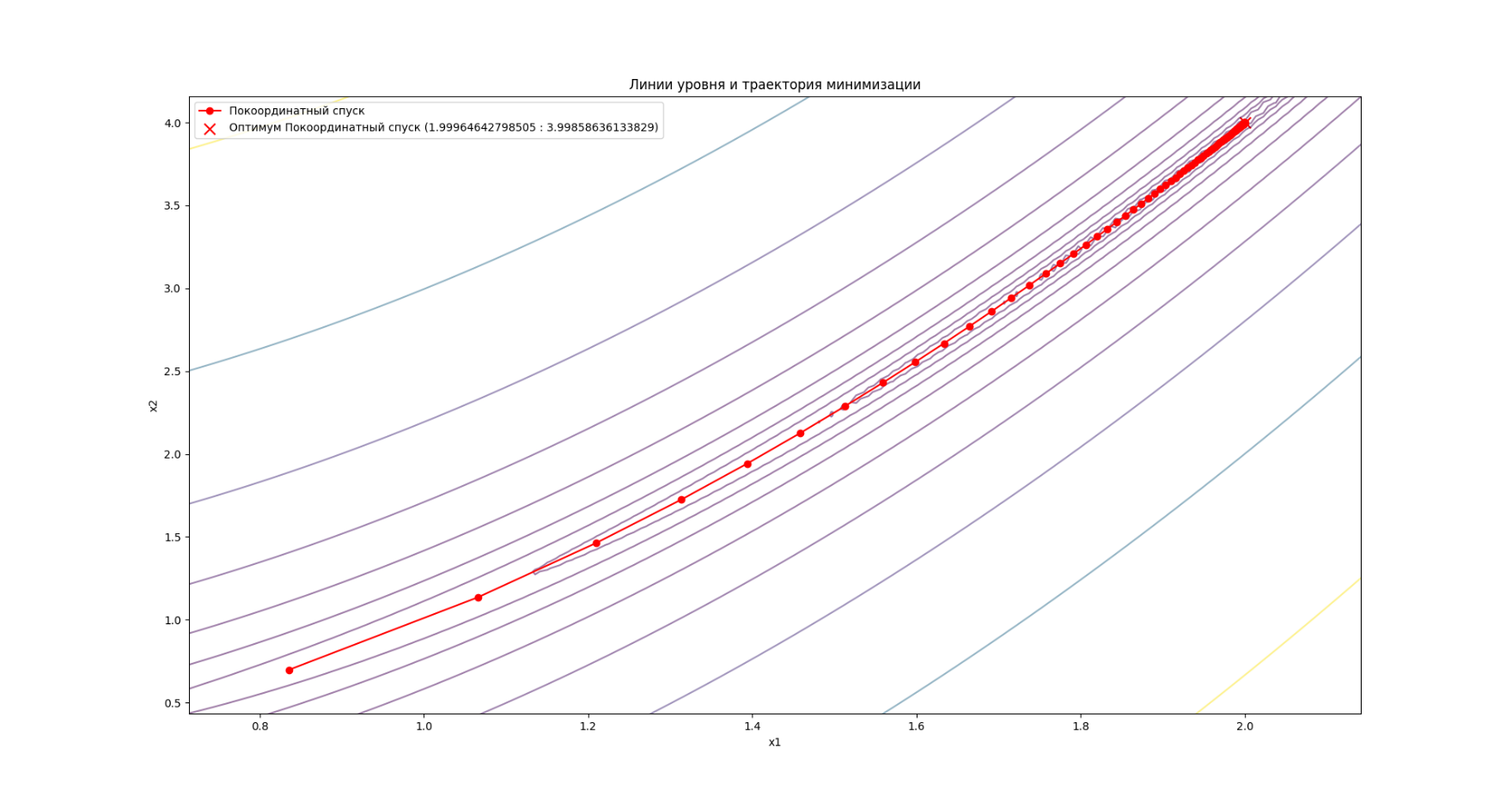
Линии уровня и траектория минимизации методом **Пауэла**



Линии уровня и траектория минимизации методом **Пауэла** (приближено)

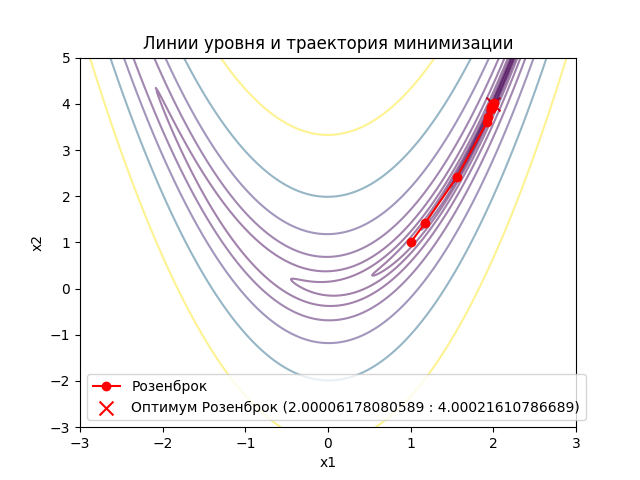


Линии уровня и траектория минимизации методом **покоординатного спуска**

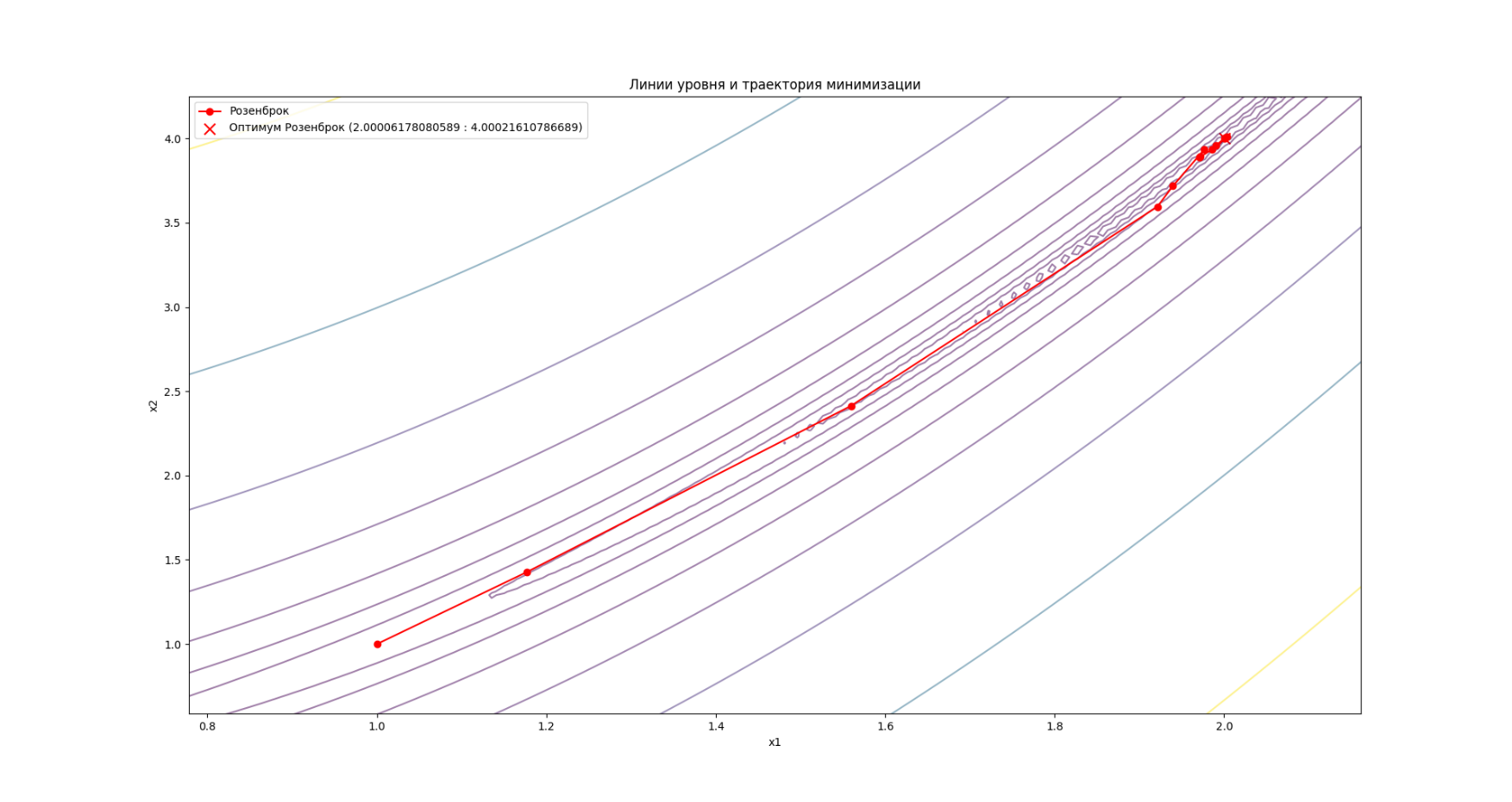


Линии уровня и траектория минимизации методом **покоординатного спуска**

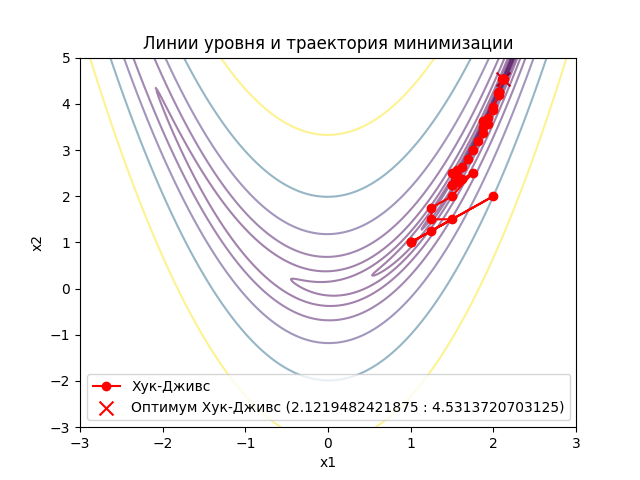
(приближено)



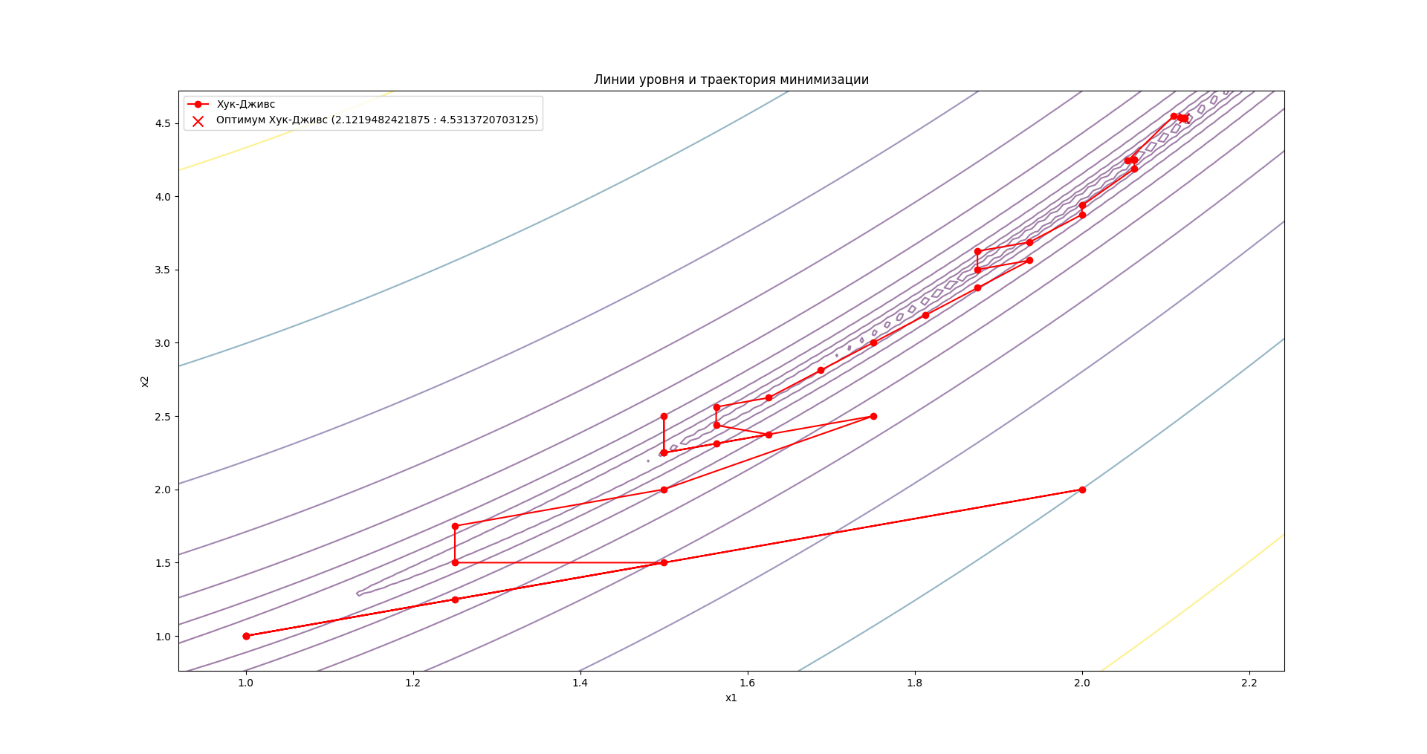
Линии уровня и траектория минимизации методом Розенброка



Линии уровня и траектория минимизации методом Розенброка (приближено)



Линии уровня и траектория минимизации методом Хука-Дживса



Линии уровня и траектория минимизации методом Хука-Дживса (приближено)

7. Таблица результатов для целевой функции: 90(x2-x1^2)^2+(2-x1)^2, точность 0.01

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Метод | x1\_opt | x2\_opt | Число операций | Точность | Длина шага |
| Нелдер-Мид | 2,00578176647992 | 4,03077262178215 | 25 | 9,13733016838838E-5 | 9,54433355630123E-6 |
| Пауэл | 1,99950771571086 | 3,99804036585813 | 77 | 0,00912373080048873 | - |
| Покоординатный спуск | 1,99964642798505 | 3,99858636133829 | 119 | 0,00856578932733877 | - |
| Розенброк | 2,00006178080589 | 4,00021610786689 | 15 | 7,49415384432957E-5 | - |
| Хук-Дживс | 2,121948242187 | 4,5313720703125 | 42 | 6,103515625E-5 | 6,103515625E-5 |

8. Из полученных результатов (на изображениях) можно сделать вывод, что

1) Метод Нелдера-Мида сильно “ускоряется” в заданной начальной точке, так как сначала берутся самые большие значения для шага, а по мере приближения к точке значения уменьшаются. Данная траектория совсем непредсказуема в начале: её вектор направления может изменить свой знак на первых итерациях.

2) Метод Пауэла тоже на старте берет значения поменьше, благодаря чему график траектории минимизации в нашем случае является строго возрастающим, но из-за меньшей длины шага методу понадобилось больше итераций. Зато можно сказать, что данный метод более предсказуемо будет себя вести.

3) Метод покоординатного спуска имеет форму гладкой функции, что делает её более предсказуемой на дальнейших итерациях, но из-за этого и самым медленным из-за его примитивности в расчете шага и вычислении следующей точки

4) Метод Розенброка имеет другую формулу расчета шага, в отличии от других методов, при первой итерации его шаг намного меньше, чем от второй и третьей, в то же время на четвертой итерации шаг становится даже короче чем на первой итерации, что нам говорит о непредсказуемости данного метода, зато скорость его сходимости, благодаря данной особенности, выше чем у любого другого метода

5) Метод Хука-Дживса самый непредсказуемый из всех: его траектория минимизации имеет вид ломаной линии, а также шаг и направление может как увеличиваться, так и уменьшаться, в зависимости от итерации и меры приближения к точке оптимума.

9. Эффективности использования предлагаемых методов  
прямого поиска для данной функции:

Исходя из таблицы результатов минимизации для каждого метода, можно сделать вывод, что:

Метод Нелдера-Мида оказался самым точным с наименьшей точностью (9,13733016838838E-5) среди всех методов. Метод Розенброка также показал хорошую точность (7,49415384432957E-5), но при меньшем числе операций (15), что делает его более эффективным с точки зрения числа операций. Метод покоординатного спуска и метод Пауэла имеют одни из самых неточных и затратных по итерациям результатов. Метод Хука-Дживса оказался “золотой серединой” среди всех методов в: сложности алгоритма, числе операций и точности полученных результатов.