МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Липецкий государственный технический университет

Факультет автоматизации и информатики

Лабораторная работа №2

по математическому программированию

“Методы оптимизации нулевого порядка”

Студент Станиславчук С. М.

Группа АС-21-1

Руководитель Качановский Ю. П.

Липецк 2023 г.

Содержание

2. Задание.

1) Отобразить линии уровня минимизируемой функции

2) С помощью программы MoDS найти минимум функции следующими методами: метод покоординатного спуска, метод сопряженных направлений, метод Хука-Дживса, метод Розенброка, метод Нелдера-Мида.

3) Отобразить траектории нахождения оптимума каждым методом на графике линий уровня функции

4) Представить результаты расчета всеми методами в таблице

5) На основе топологии функции сделать выводы о траекториях оптимизации каждого метода

6) Сделать выводы об эффективности использования предлагаемых методов прямого поиска для заданной функции.

Номер задачи: 4

Номер варианта данной задачи: 2

Задача №4

Студенту специальности АС дали задание сравнить результаты минимизации функции Розенброка различными методами прямого поиска.

Функция имеет вид:

$$f(x) = a(x_2 - {x_1}^2)^2 + (b - x_1)^2$$

Значения коэффициентов:

$$a = 90; \mathrm{b} = 2$$

Пусть $x_1 = x,\ x_2 = y$, тогда функция с подставленными значениями коэффициентов имеет вид:

$$f(x) = 90(y - x^2)^2 + (2 - x)^2$$

3. Цель работы.

Изучение численных методов прямого поиска, предназначенных для оптимизации функции многих переменных.

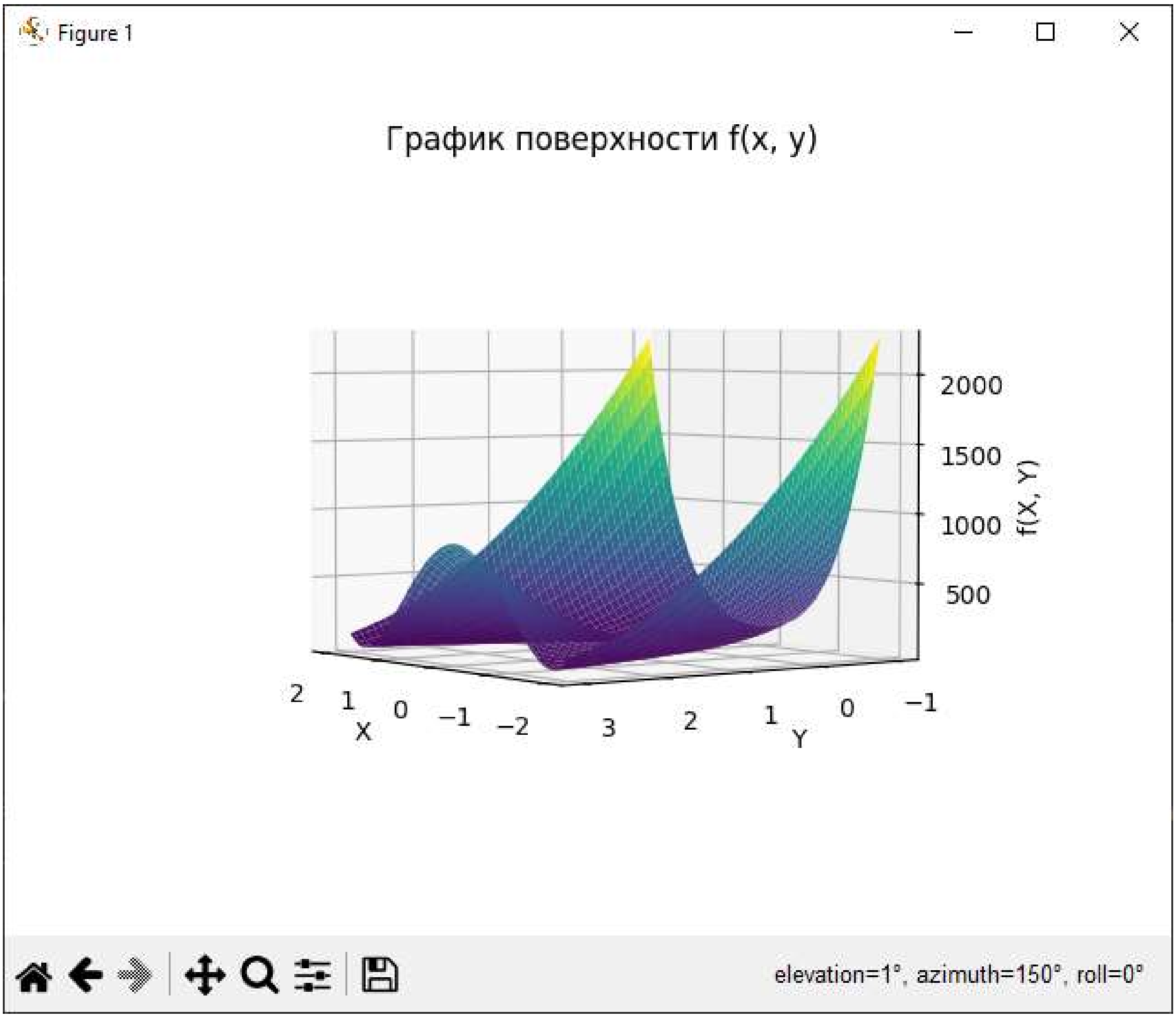
Ход работы

4. Уравнение функции, подлежащей минимизации:

$$f(x, y) = 90(y - x^2)^2 + (2 - x)^2$$

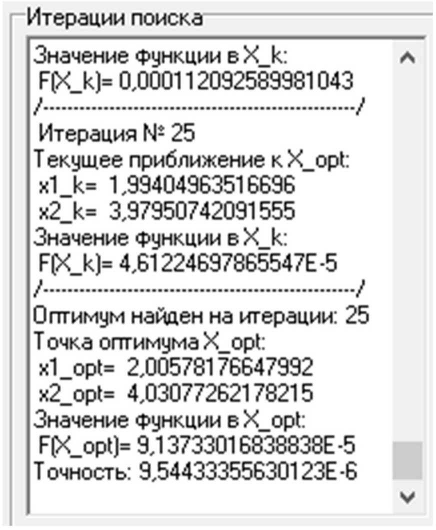
5. График поверхности функции f(x,y):

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.

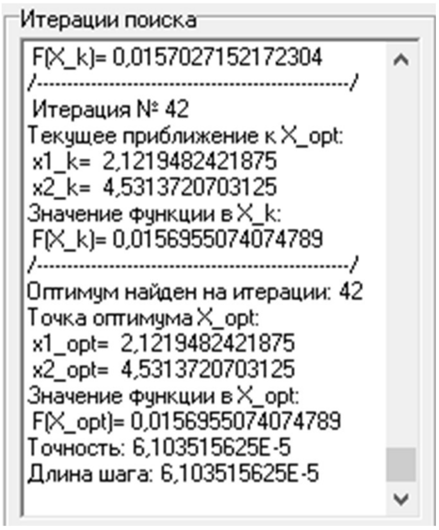


С помощью программы MoDS найдем минимум функции следующими методами:

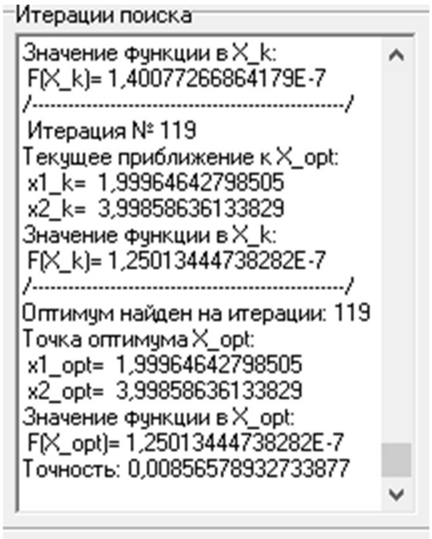
1. Метод Нелдера-Мида:



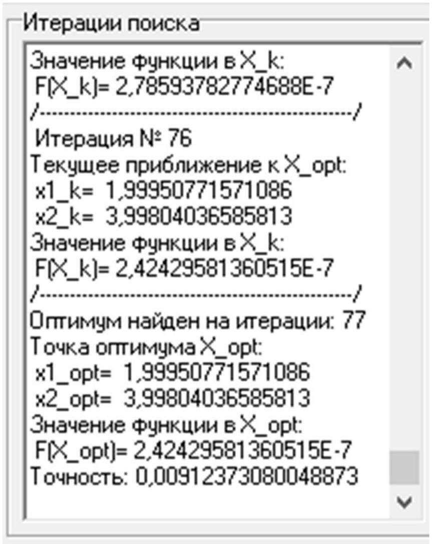
2. Метод Хука-Дживса:



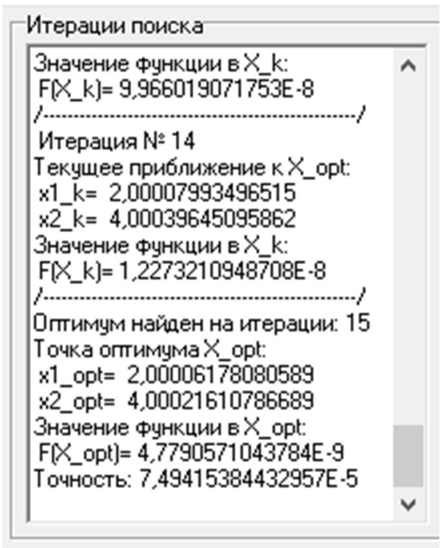
3. Метод покоординатного спуска:



4. Метод Пауэла:



5. Метод Розенброка:

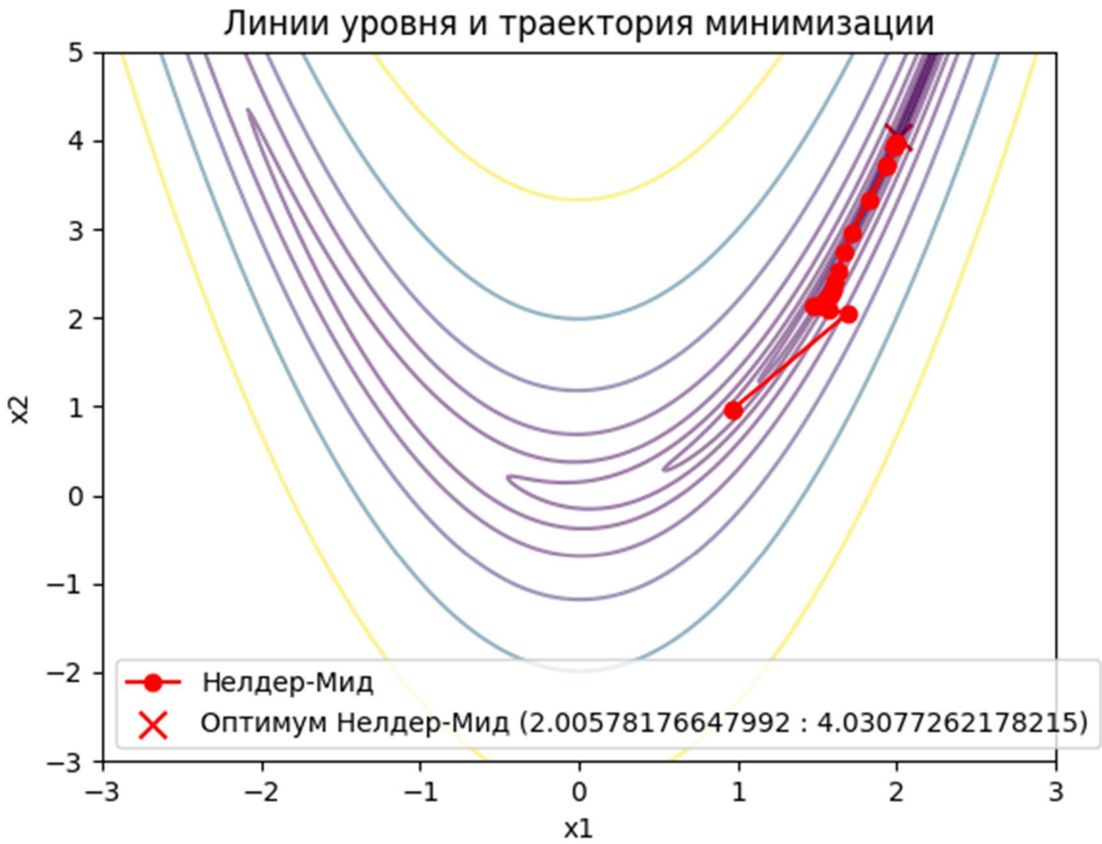


6. Для отображения графиков линий уровня функции с указанными на них траекториями минимизации, напишем Python-программу, которая считает значения x1_k, x2_k, x1_opt, x2_opt из сгенерировавшихся в программе MoDS .txt файлов.

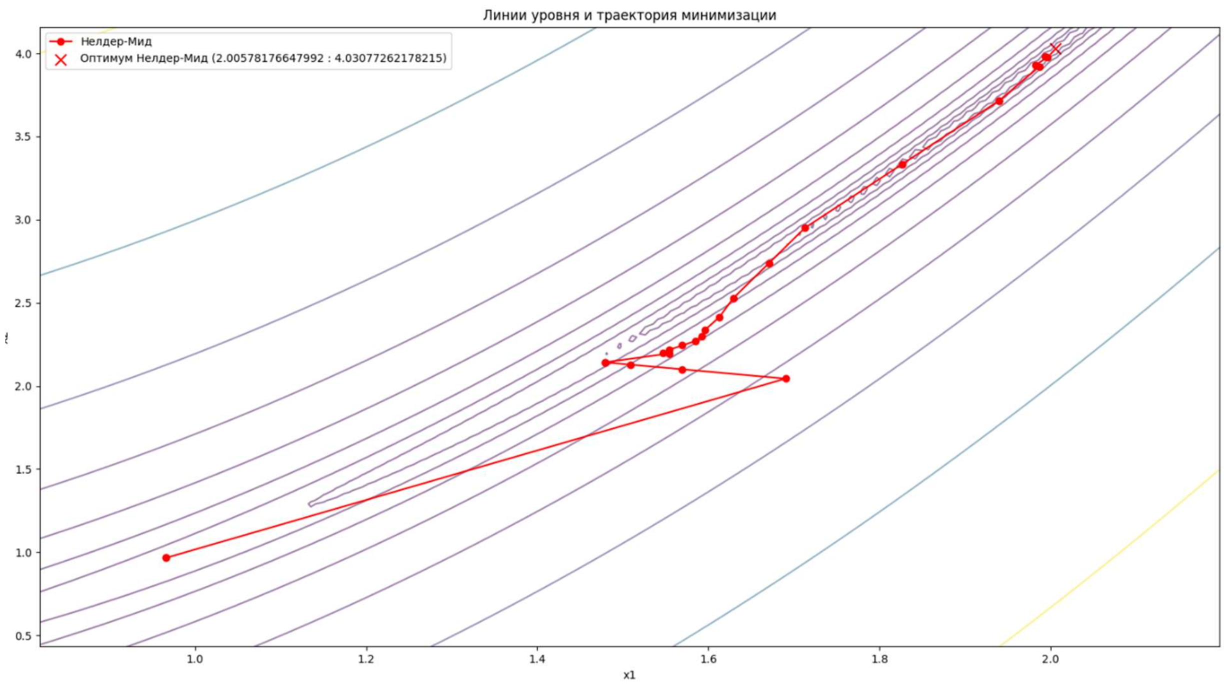
```
import re


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)
```

```python
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()
```

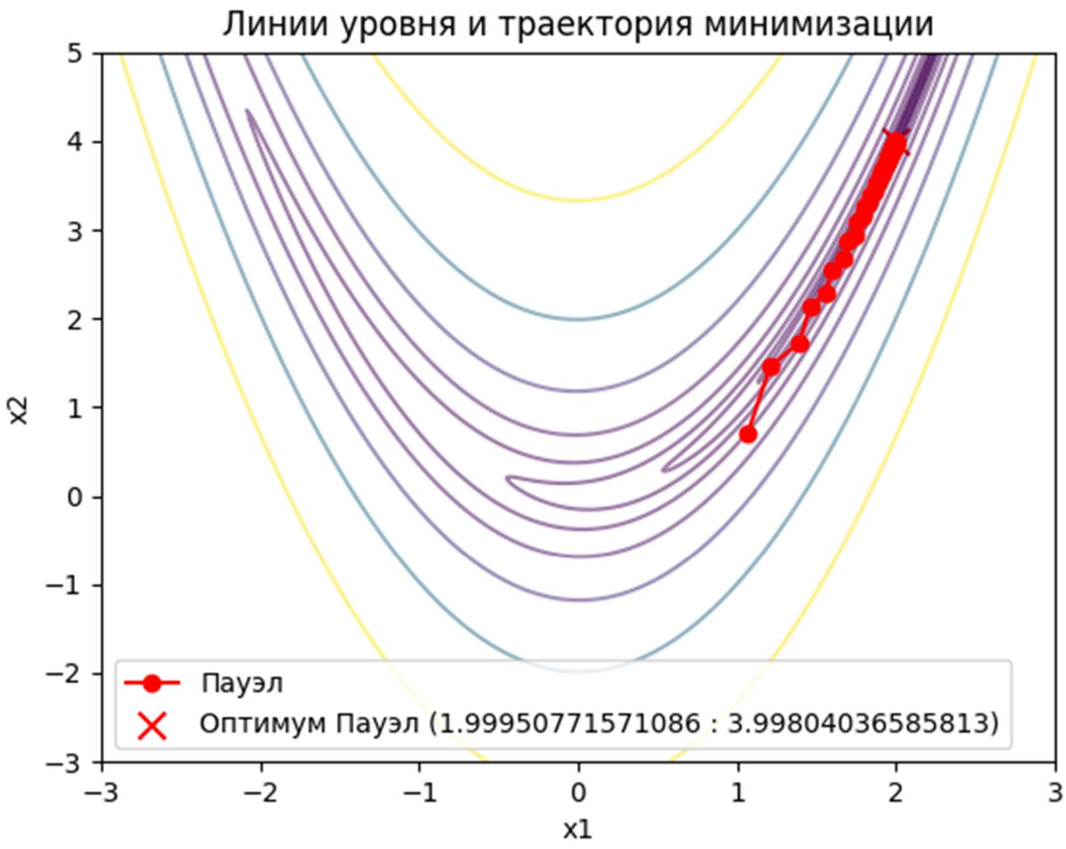
Результаты:



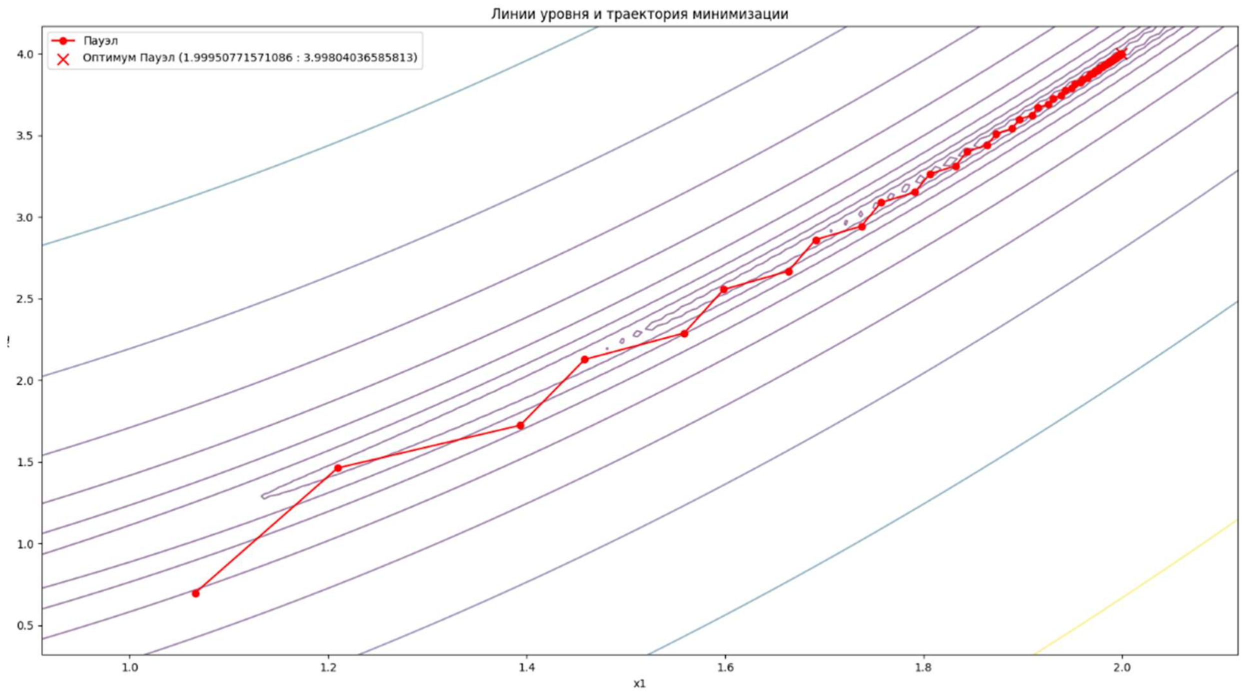
Линии уровня и траектория минимизации методом **Нелдера-Мида**



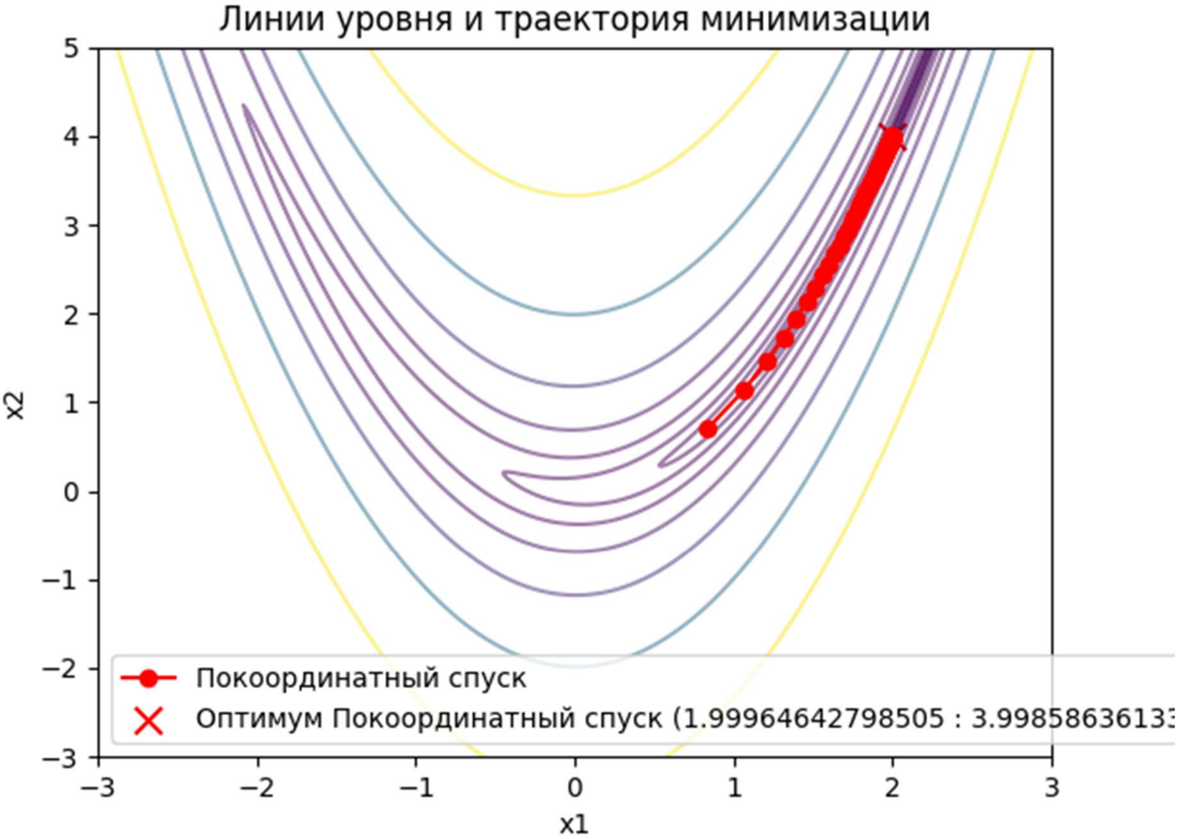
Линии уровня и траектория минимизации методом **Нелдера-Мида** (приближено)



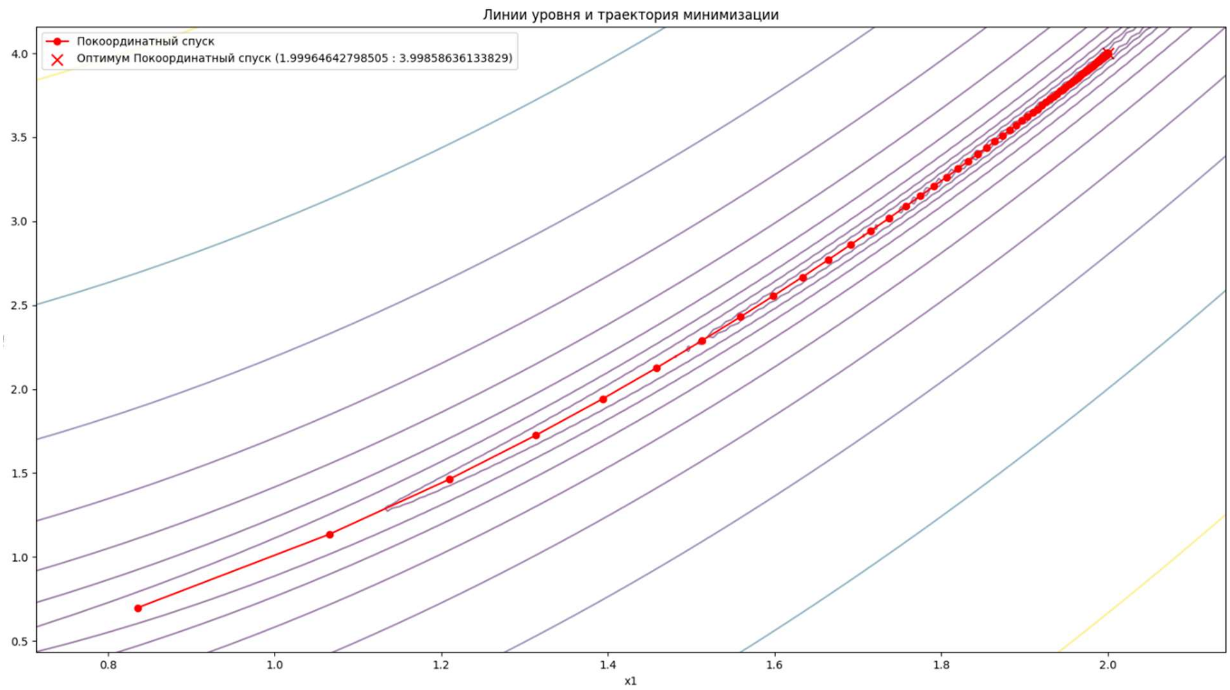
Линии уровня и траектория минимизации методом **Пауэла**



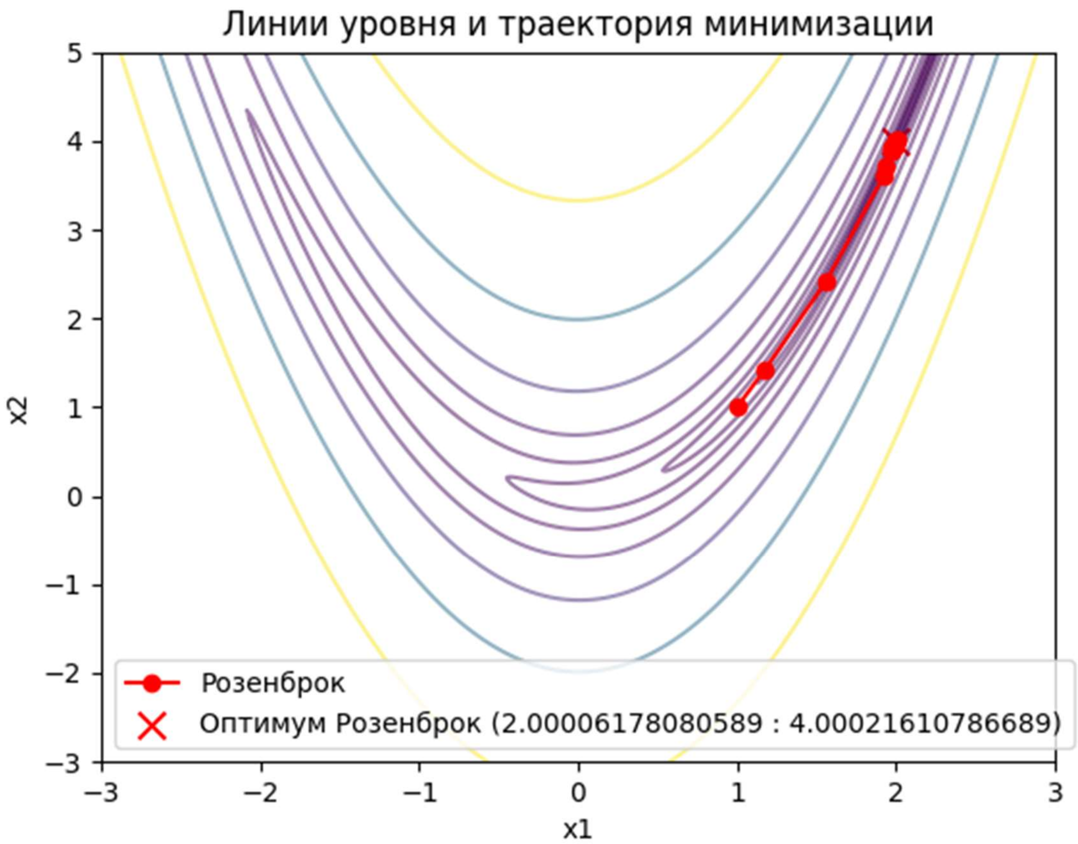
Линии уровня и траектория минимизации методом **Пауэла** (приближено)



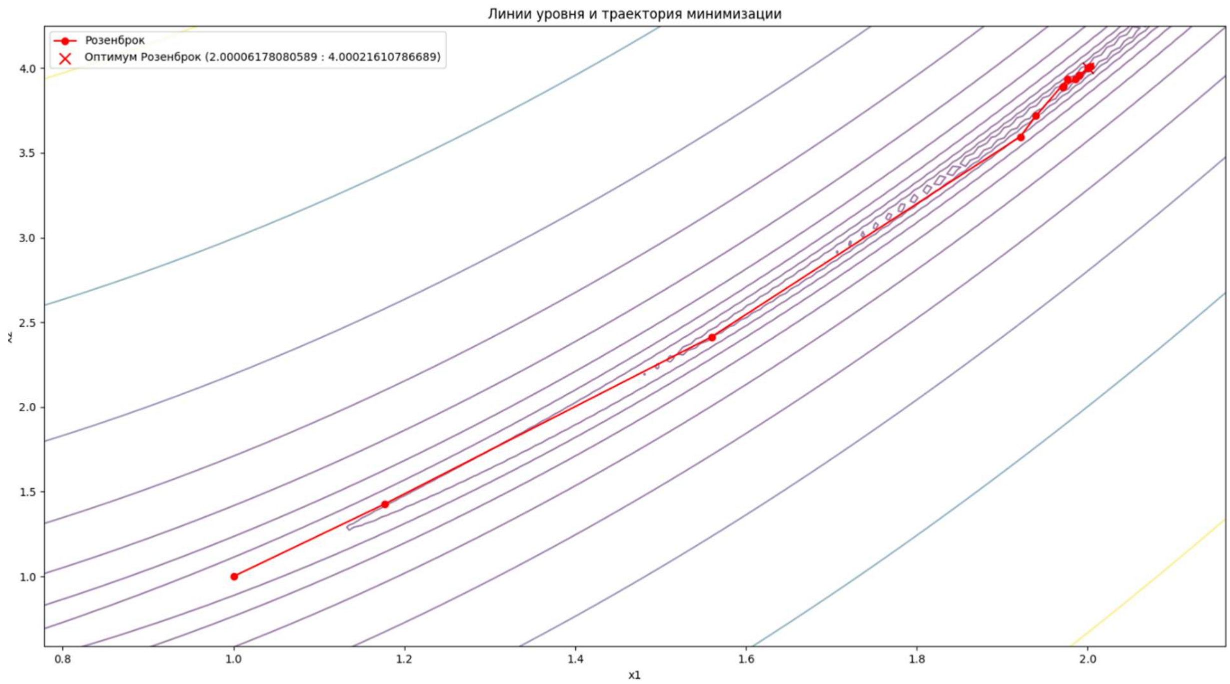
Линии уровня и траектория минимизации методом **покоординатного спуска**



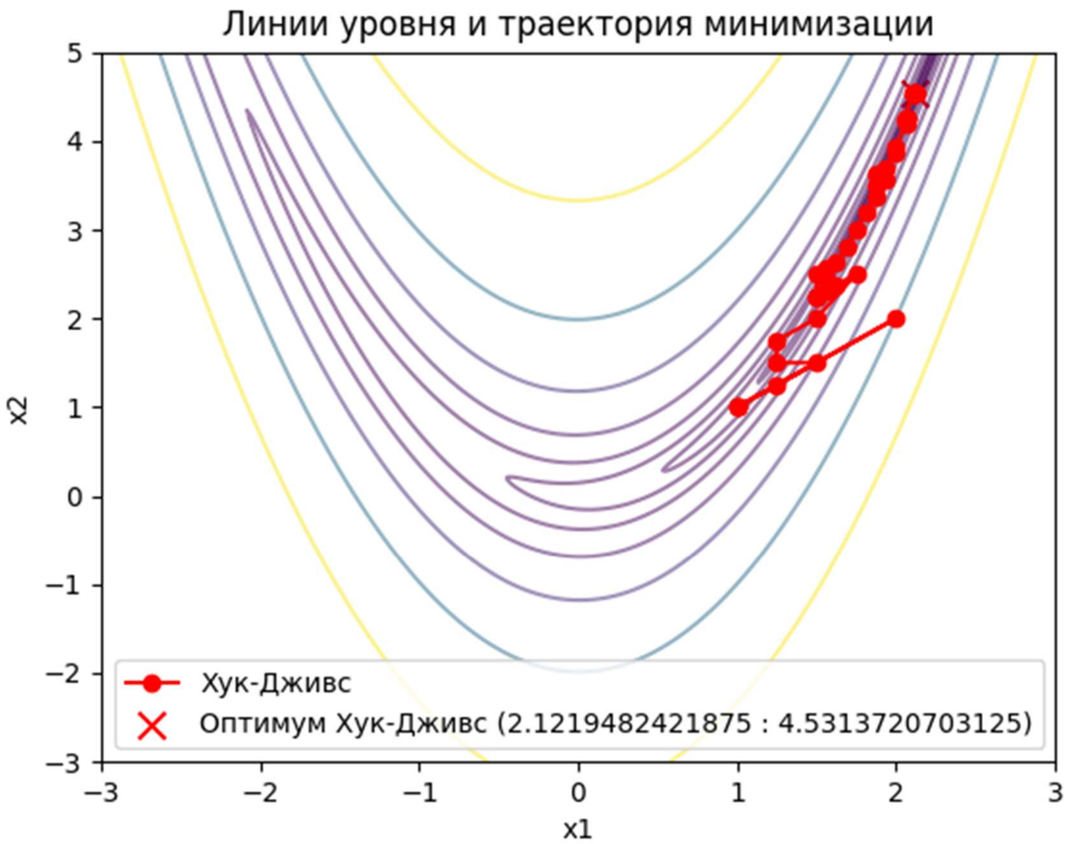
Линии уровня и траектория минимизации методом **покоординатного спуска** (приближено)



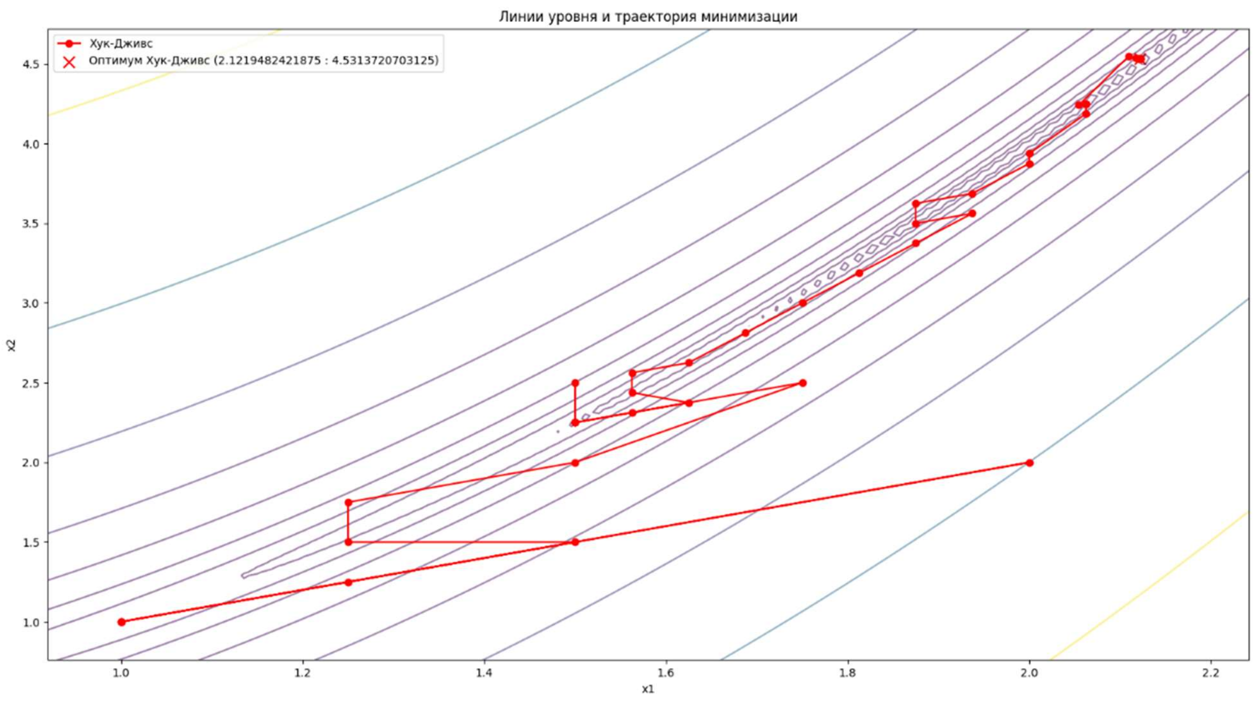
Линии уровня и траектория минимизации методом Розенброка



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.