Национальный исследовательский университет ИТМО

Факультет информационных технологий и программирования

Прикладная математика и информатика

Отчет по лабораторной работе 2

по дисциплине «**Методы оптимизации в машинном обучении**»

Авторы: Багаутдинов Искандер Ильгизович M3237

Пан Артём Олегович M3237

Преподаватель: Андреев Юрий Александрович

Санкт-Петербург 2024

Цель работы:

Реализовать и исследовать эффективность метода Ньютона путем собственной реализации, его библиотечной реализации в scipy.optimize, а также библиотечных реализаций квазиньютоновских методов между собой, а также с уже изученными методами.

Задачи работы:

- Выбрать функции, на которых эффективность методов будет явно различаться
- Исследовать работу методов в зависимости от способа нахождения производной
- Проиллюстрировать примеры

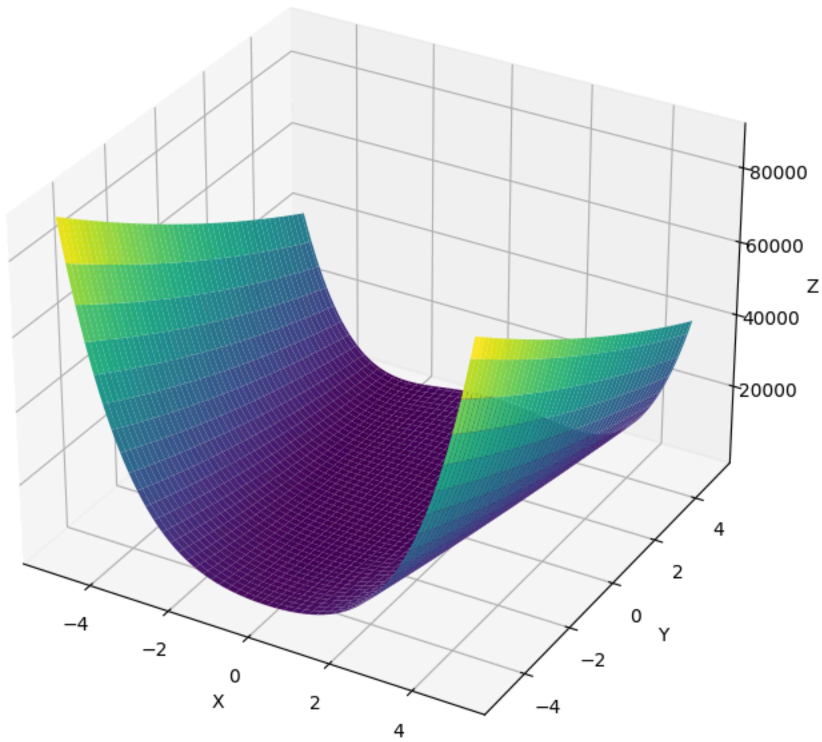
Подготовка. Выбор функций

Для проведения исследования эффективности нами были выбраны 3 стандартные тестовые функции в оптимизации:

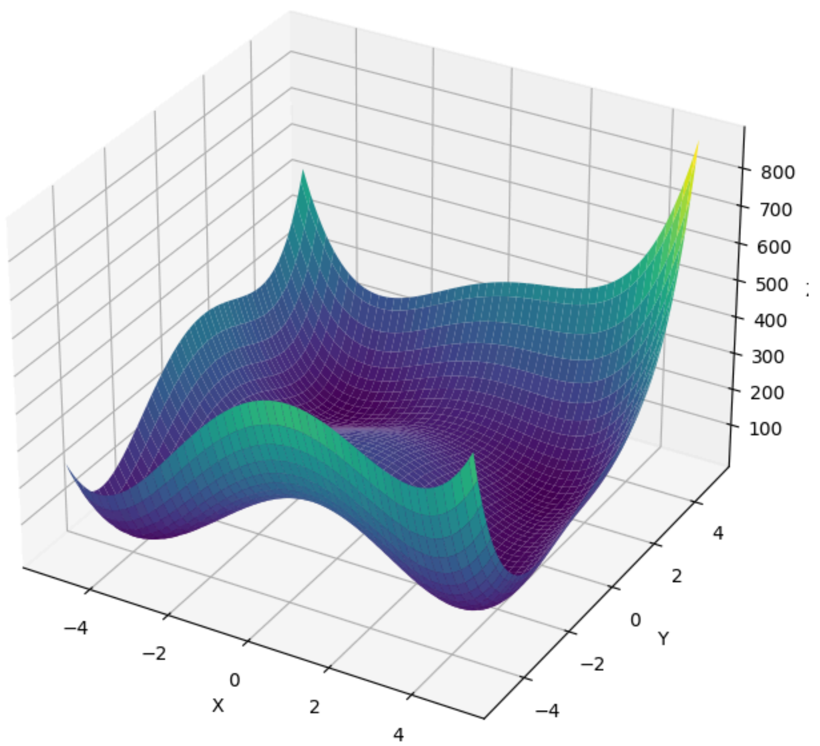
1. Функция Розенброка - функция вида (1-x)**2 + 100*(y-x**2)**2. Имеет единственный глобальный минимум в (1, 1).
2. Функция Химмельблау - (x**2 + y – 11) **2 + (x + y**2 – 7)**2. Она имеет 4 локальных минимума с одинаковым значением - 0 и один глобальный максимум - (-0.270845…, -0.923039…).
3. Функция Растригина для функции двух переменных: 20 + (x**2 - 10(cos(2pi*x)) + (y**2 - 10(cos(2pi*y)). Она имеет крайне нетривиальную поверхность и большое количество локальных минимумов и максимумов.

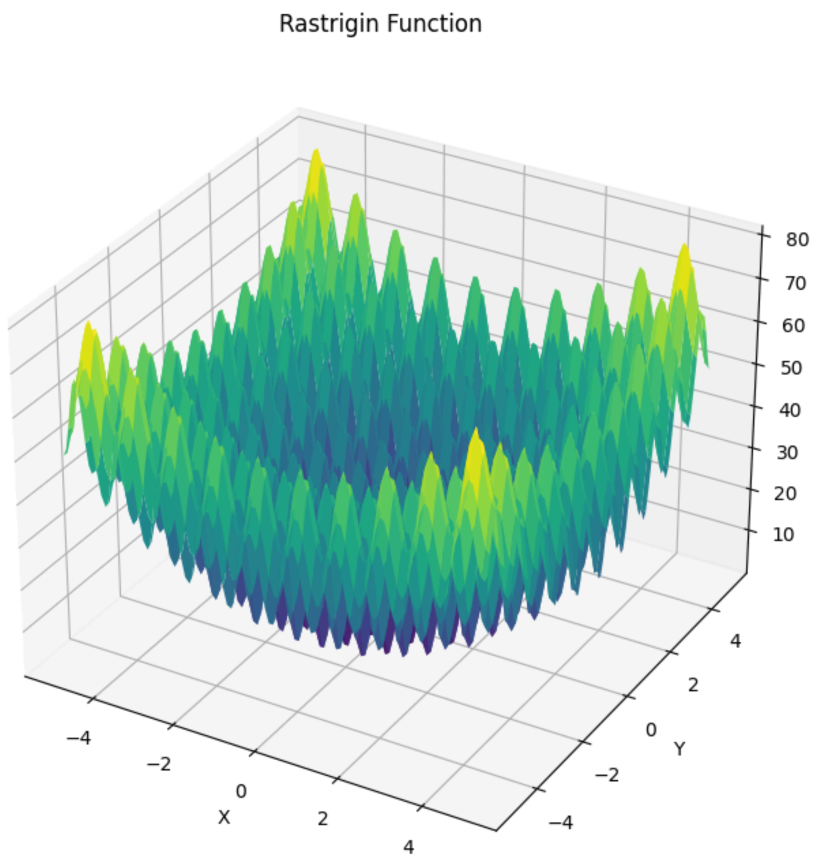
Визуализация функций:

Rosenbrock Function



Himmelblau Function





Подготовка. Реализация методов.

1. Метод Ньютона

Как мы знаем, в методе Ньютона приближение к решению осуществляется по следующему правилу:

$$\begin{cases} H(x_i)p_i = -F(x_i) \\ x_{i+1} = x_i + \alpha_i p_i \end{cases}$$, где F - это якобиан исходной функции, а H - гессиан.

p_i вычисляется из первого уравнения решением СЛАУ с помощью numpy.linalg. Функция, вычисляющая α_i передается как аргумент функции newton, которая уже в свою очередь возвращает функцию, в которой задается исследуемая функция f, начальная точка x, максимальное количество итераций maxiter (чтобы не застревать, когда метод не сходится) и значение tol: если на шаге i норма разницы x_{i-1} и x_i меньше tol, то результат метода считается вычисленным.

1.1. С постоянным шагом

Как классический метод Ньютона и предполагает в своей теории, используется $\alpha_i = const = 1$. Может возникнуть некоторая путаница в знаках, поэтому обращу внимание, что речь об α_i в определении уравнений выше.

1.2. С градиентным спуском

Использована реализация градиентного спуска из лабораторной работы №1.

2. Адаптация scipy.optimize

   2.1. Newton-CG

   Параметры: c1 - Armijo condition rule, c2 - curvature condition rule TODO

   2.2 BFGS

   Параметры: norm - порядок нормы, вычисляемой для проверки окончания вычисления метода, поставлен на 2 для консистентности со своей реализацией Ньютона, finite_diff_rel_step, c1, c2 TODO

   2.2 L-BFGS-B

   Параметры: maxcor - количество значений, хранимых алгоритмом для аппроксимации гессиана, оставлено дефолтным (10), maxls - максимальное количество шагов линейного поиска за итерацию, оставлено дефолтным (20), finite_diff_rel_step

Основное задание. Пункт 1: сравнение своей реализации с Newton-CG из scipy.optimize

Для сравнения по скорости будем смотреть на количество вычислений функции, ее градиента и гессиана, так как эти вычисления могут занимать много времени у функций не аналитического происхождения.

Для сравнения по точности будем задавать одинаковый уровень tol. Также будем смотреть является ли результат глобальным минимумом (дискретный показатель).

Функция Розенброка, x0 = (0, 0), tol = 1e-18:

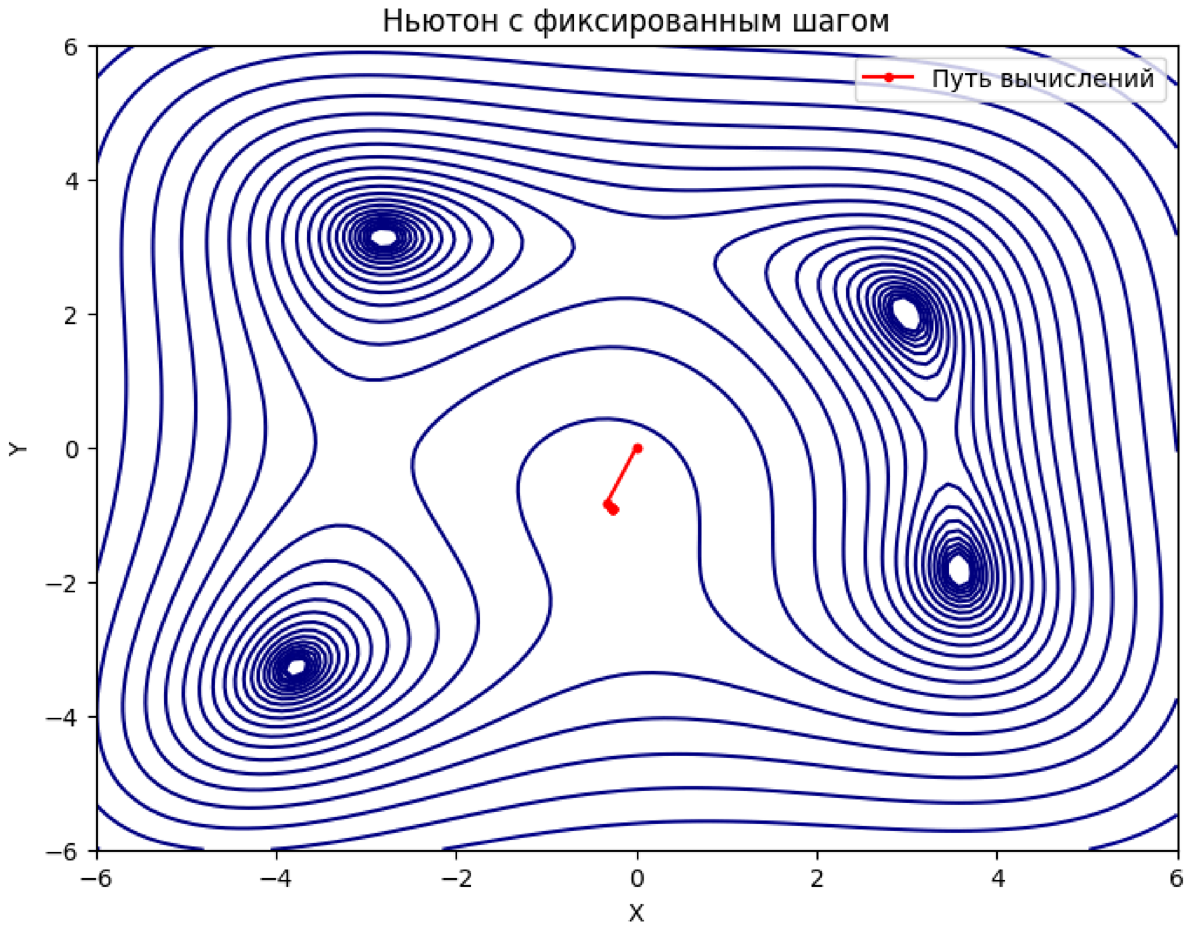
Метод	Предполагаемая x* (округлено)	f(x*) (округлено)	Количество итераций	Количество вычислений f	Количество вычислений градиента	Количество вычислений гессиана
свой Ньютон с фиксированным шагом	(1, 1)	0	3	0	3	3
свой Ньютон с	(1, 1)	18	8	756	18	18

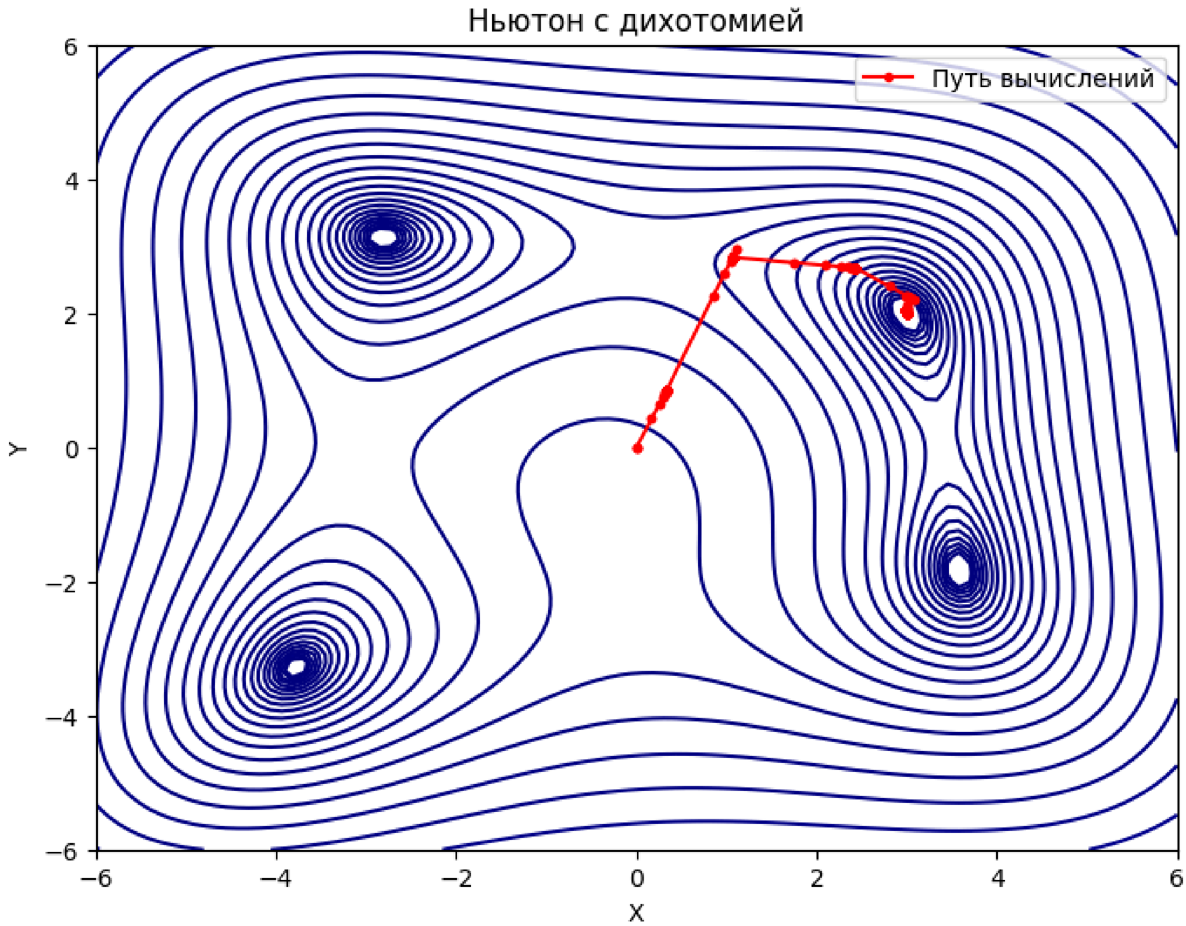
дихотомией						
Newton-CG	(1, 1)	0	36	55	55	36

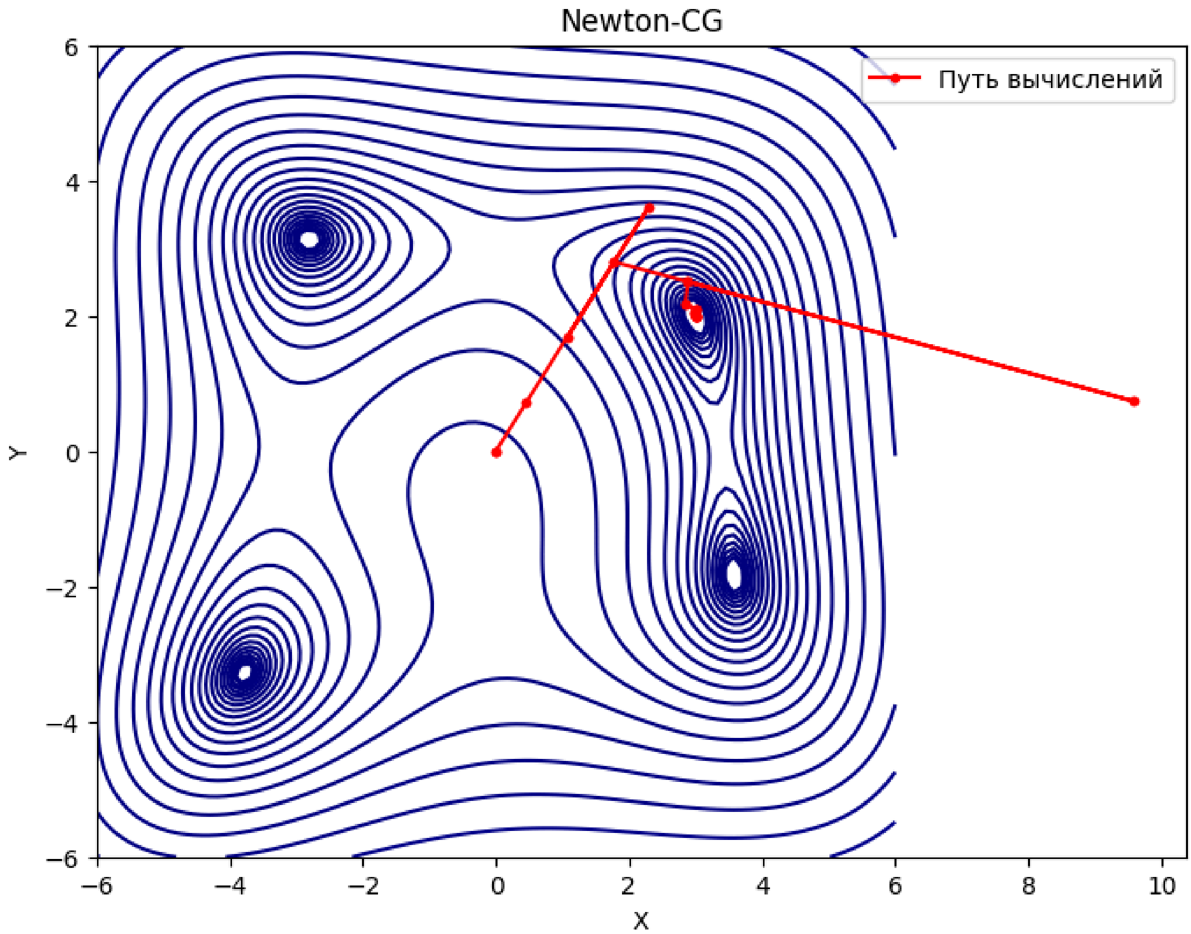
Все методы сошлись к глобальному минимуму.

Функция Химмельблау, x0 = (0, 0), tol = 1e-18:

Метод	Предполагаемая x* (округлено)	f(x*) (округлено)	Количество итераций	Количество вычислений f	Количество вычислений градиента	Количество вычислений гессиана
свой Ньютон с фиксированным шагом	(-0.27084, -0.92304)	181.61652	7	0	7	7
свой Ньютон с дихотомией	(3, 2)	0	10	770	10	10
Newton-CG	(3, 2)	0	9	13	13	9


Ньютон с фиксированным шагом
Путь вычислений
Y
X
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6


Ньютон с дихотомией
Путь вычислений
X
Y
−6
−4
−2
0
2
4
6



Эксперимент показывает недостаток метода Ньютона с фиксированным шагом - он ищет те значения, в которых производная равна 0, поэтому находит как минимумы, так и максимумы: в этом случае метод застрял в максимуме.

В дихотомии и Newton-CG же такая проблема избегается одномерным поиском по направлению (по α_i), который не идет в сторону максимума.

Замечу, что путь вычислений на графиках показывает все точки, в которых вычислялась функция, ее градиент или гессиан, а не значения, на которых остановился x на некоторой итерации. Именно из-за этого мы видим прямые на графике Ньютона с дихотомией и Newton-CG, отвечающие за одномерный поиск.

Функция Химмельблау, x0 = (1, 1), tol = 1e-18:

Метод	Предполагаемая x* (округлено)	f(x*) (округлено)	Количество итераций	Количество вычислений f	Количество вычислений градиента	Количество вычислений гессиана
свой Ньютон с фиксиров	(-3.77931, -3.28319)	0	7	0	7	7

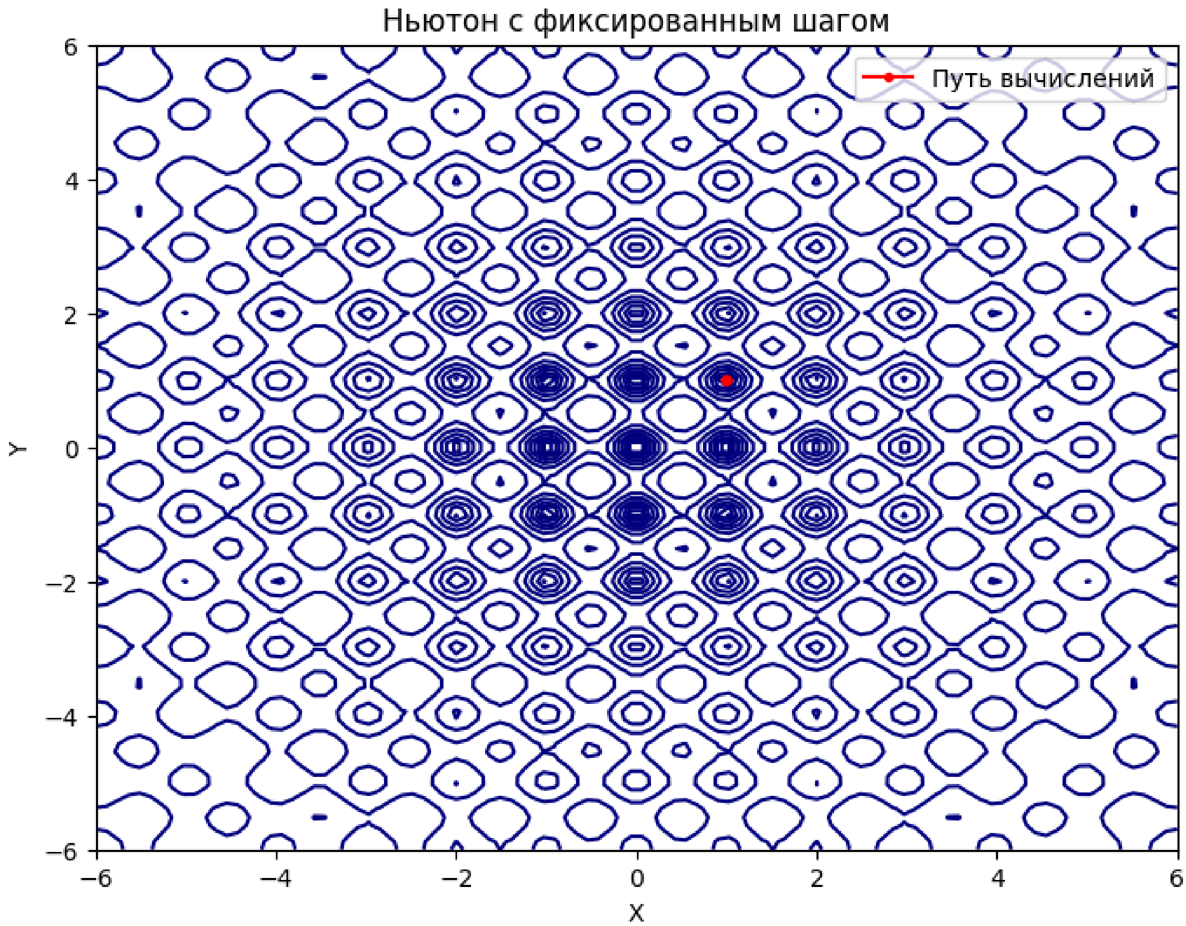
анным шагом						
свой Ньютон с дихотомией	(3, 2)	0	8	336	8	8
Newton-CG	(3, 2)	0	9	10	10	9

В этот раз все методы справились - каждый нашел идентичный локальный минимум.

Функция Растригина, x0 = (1, 1), tol = 1e-18:

Метод	Предполагаемая x* (округлено)	f(x*) (округлено)	Количество итераций	Количество вычислений f	Количество вычислений градиента	Количество вычислений гессиана
свой Ньютон с фиксированным шагом	(0.99496, 0.99496)	1.98992	4	0	4	4
свой Ньютон с дихотомией	(0.99496, 0.99496)	1.98992	7	294	7	7
Newton-CG	(0.99496, 0.99496)	1.98992	3	3	3	3

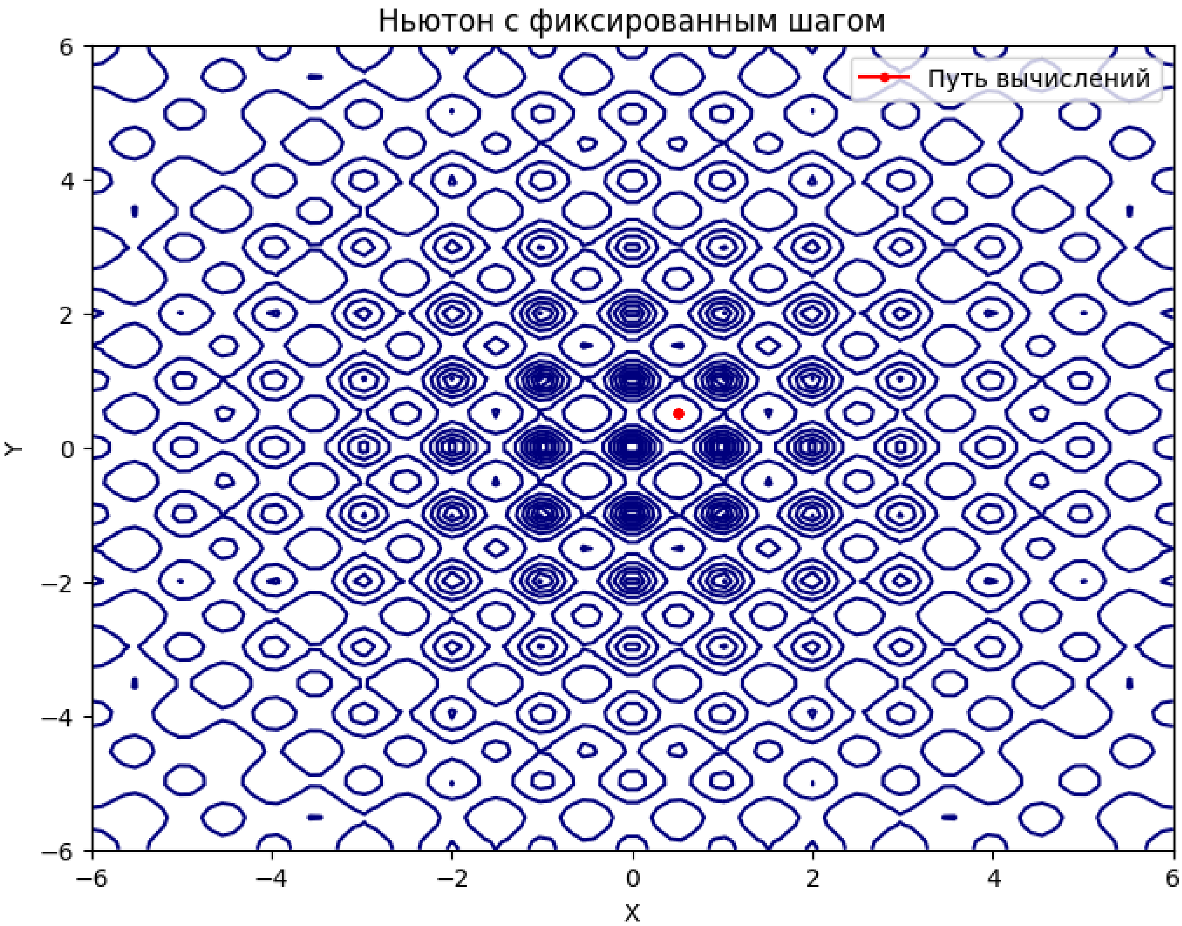
Все методы попали в точку локального минимума, находящегося близко к начальной точке. Неудивительно, что одномерный поиск не спас последние два метода - градиент слишком мал из-за близости локального минимума и выбраться из него не получится. Заметим, что метод Ньютона с фиксированным шагом для этого и не создан - теорема говорит о поиске локальных экстремумов.

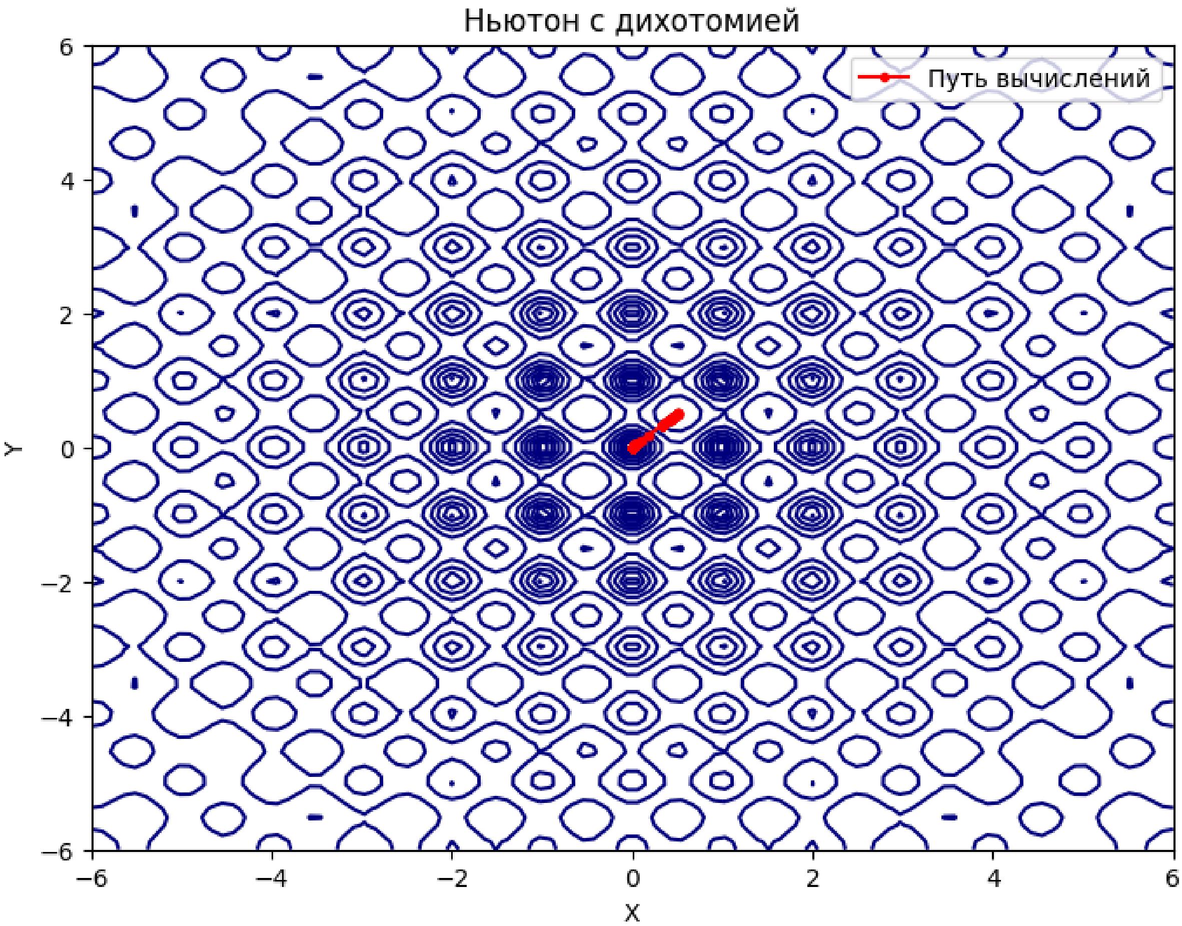


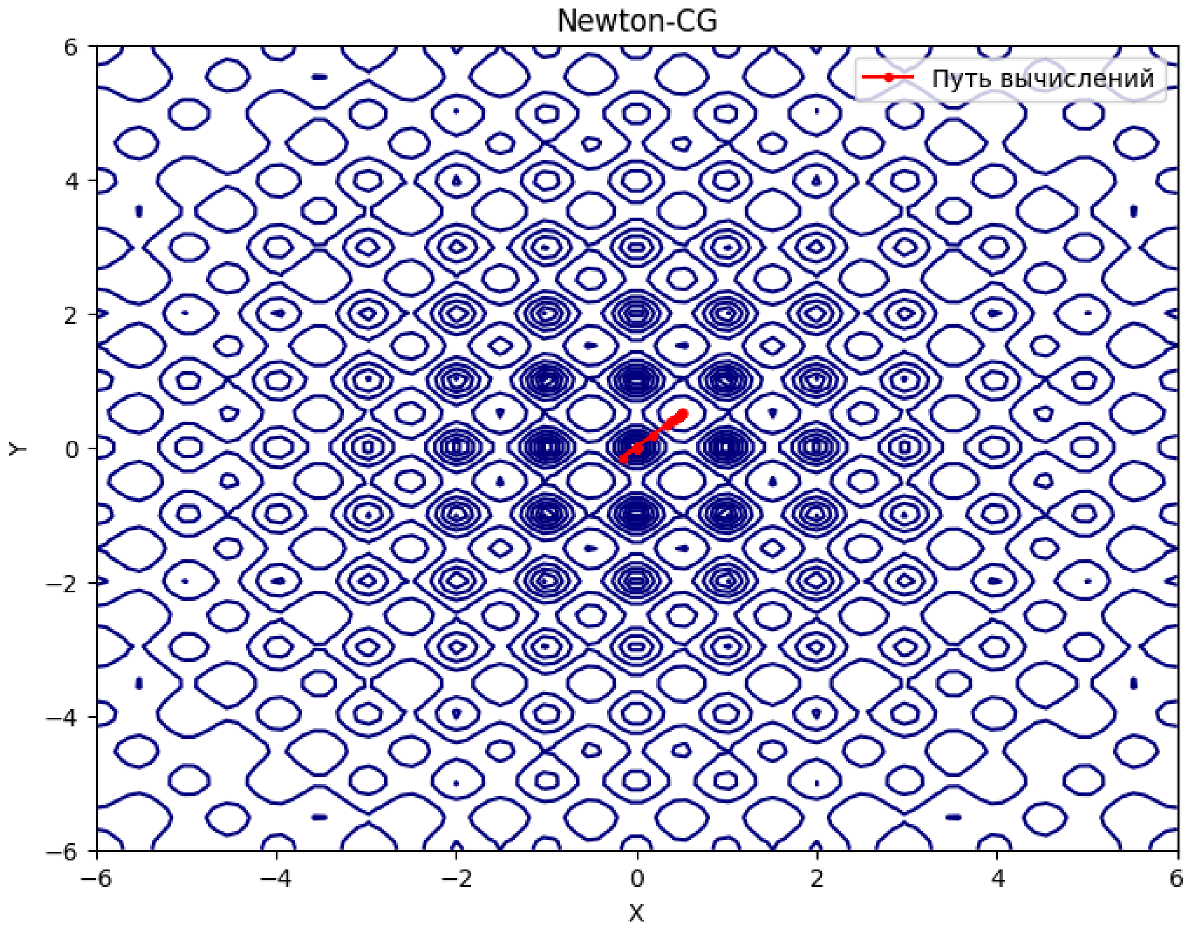
Другие графики не приведены, так как выглядят аналогично.

Функция Растригина, x0 = (0.5, 0.5), tol = 1e-18:

Метод	Предполагаемая x* (округлено)	f(x*) (округлено)	Количество итераций	Количество вычислений f	Количество вычислений градиента	Количество вычислений гессиана
свой Ньютон с фиксированным шагом	(0.50255, 0.50255)	40.50255	4	0	4	4
свой Ньютон с дихотомией	(0, 0)	0	11	462	11	11
Newton-CG	(0, 0)	0	2	42	30	3


Ньютон с фиксированным шагом
Путь вычислений
X
Y
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6


Ньютон с дихотомией
Путь вычислений
X
Y
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6



По сказанному ранее, неудивительно, что метод Ньютона с фиксированным шагом застревает в максимуме около начальной точки.

Основное задание. Пункт 2. Сравнение методов нулевого порядка и градиентного спуска, метода Ньютона и квазиньютоновских методов

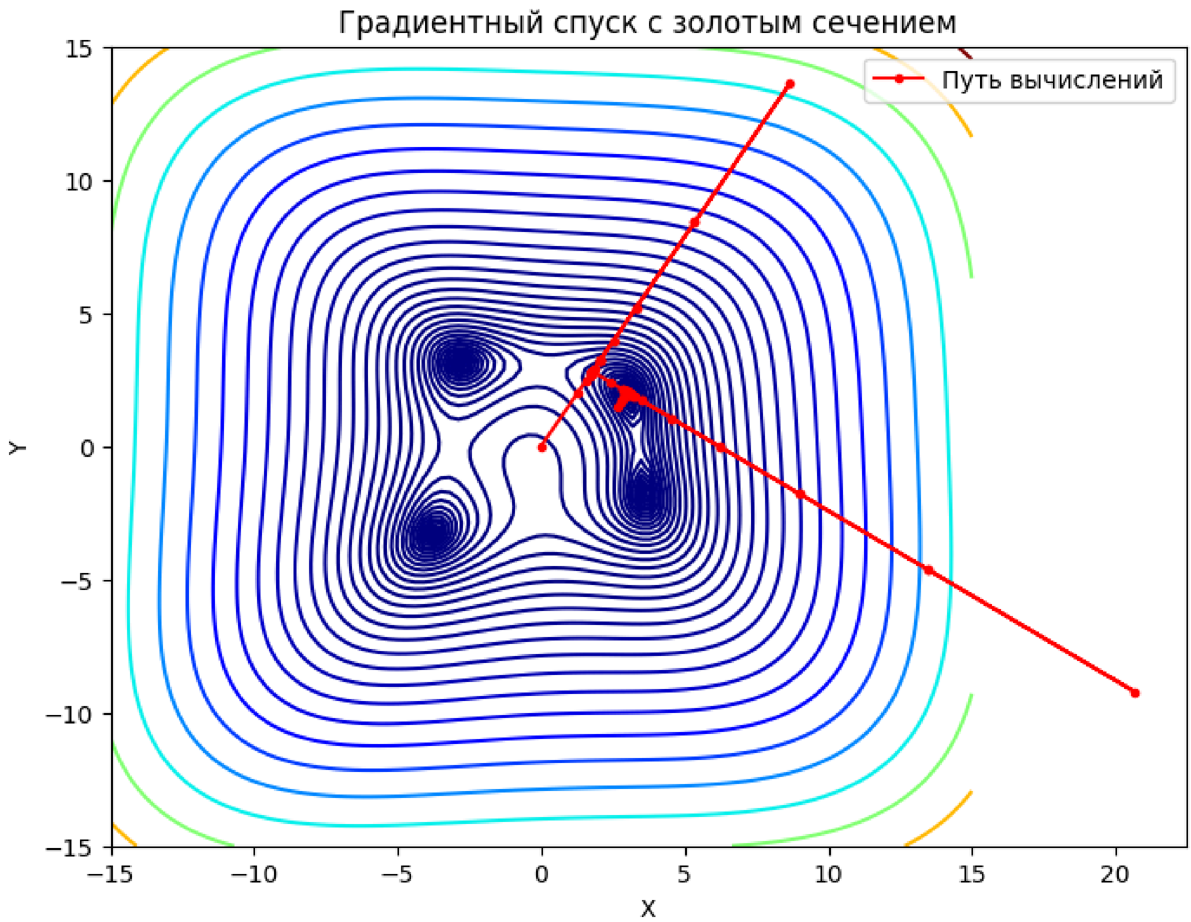
В качестве метода нулевого порядка в комбинации с градиентным спуском было выбрано золотое сечение.

В качестве квазиньютоновских методов были выбраны алгоритм Бройдена — Флетчера — Гольдфарба — Шанно (BFGS из scipy.optimize), а также его версия с ограниченной памятью (L-BFGS-B).

Функция Химмельблау, x0 = (0, 0), tol = 1e-18:

Метод	Предполагаемая x* (округлено)	f(x*) (округлено)	Количество итераций	Количество вычислений f	Количество вычислений градиента	Количество вычислений гессиана
градиентный спуск	(3, 2)	0	49	2842	49	0

с золотым сечением						
свой Ньютон с дихотомией	(3, 2)	0	10	420	10	10
BFGS	(3, 2)	0	15	21	21	0
L-BFGS	(3, 2)	0	12	18	18	0



Как мы видим все методы успешно пришли к правильному результату, но градиентному спуску для достижения требуемой точности понадобилось почти на порядок больше вычислений значения функции в точке.

Функция Растригина, x0 = (0.5, 0.5), tol = 1e-18:

Метод	Предполагаемая x* (округлено)	f(x*) (округлено)	Количество итераций	Количество вычислений f	Количество вычислений градиента	Количество вычислений гессиана
градиентн	(0, 0)	0	1000	58000	1000	0

ый спуск с золотым сечением						
свой Ньютон с дихотомией	(0, 0)	0	11	462	11	11
BFGS	(0, 0)	0	3	49	37	0
L-BFGS	(0, 0)	0	3	50	50	0

Аналогично предыдущему эксперименту замечаем разницу в 2 порядка в количестве вычислений для градиентного спуска в сравнение с остальными методами.

Метод Ньютона с дихотомией также показывает результат значительно хуже в сравнении с библиотечными реализациями квазиньютоновских методов.

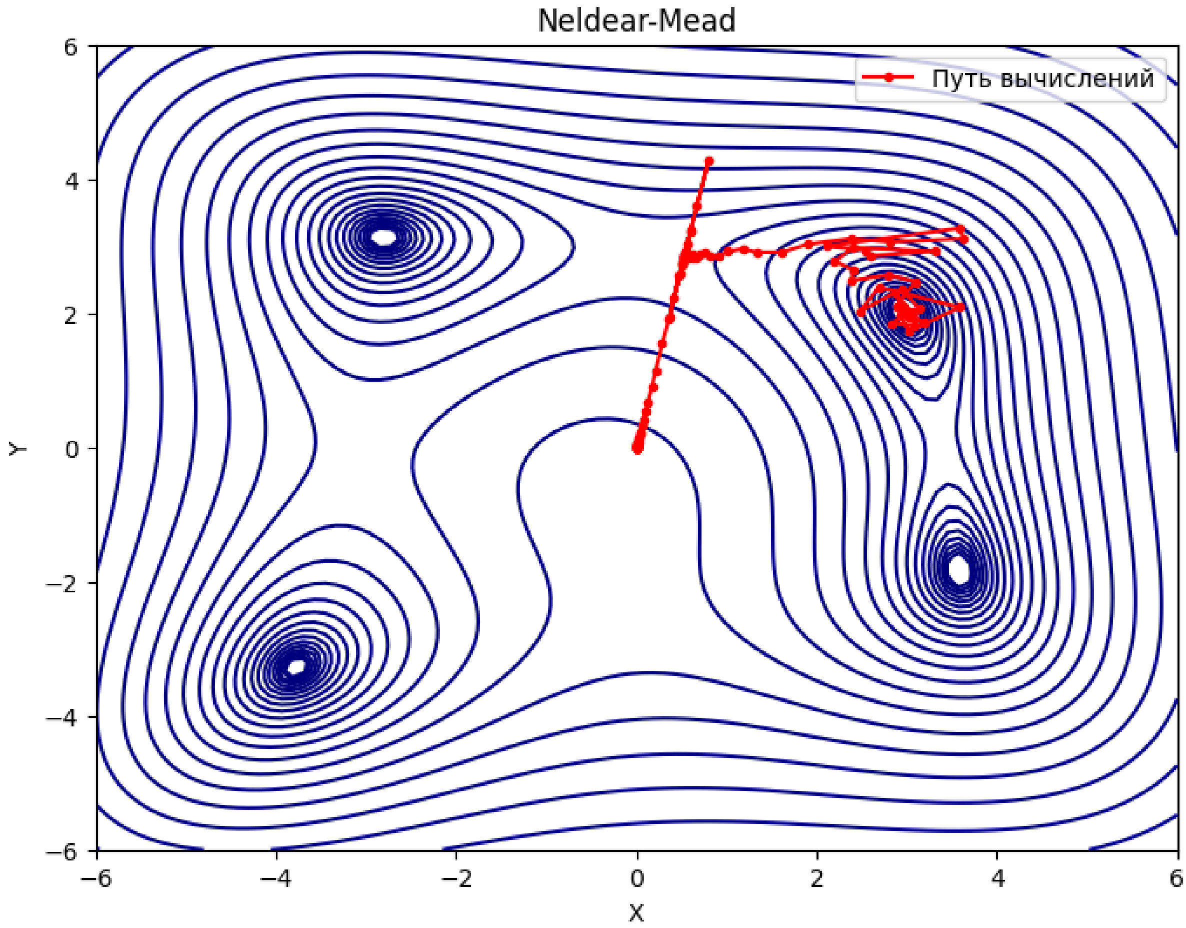
Основное задание. Пункт 3. Сравнение методов нулевого порядка с квазиньютоновскими методами (производная по разностной схеме)

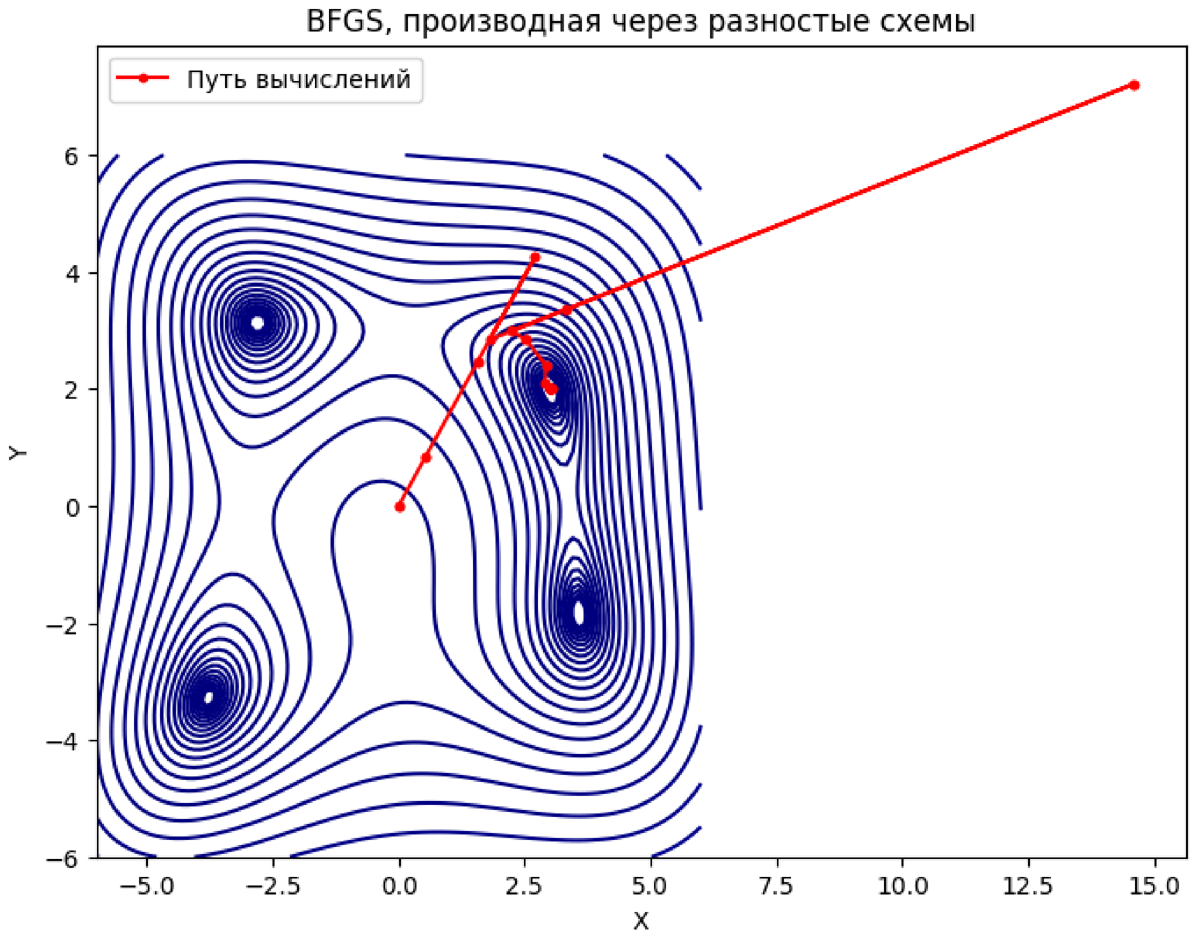
Способ вычисления производной задается в функции scipy.optimize minimize - jac.

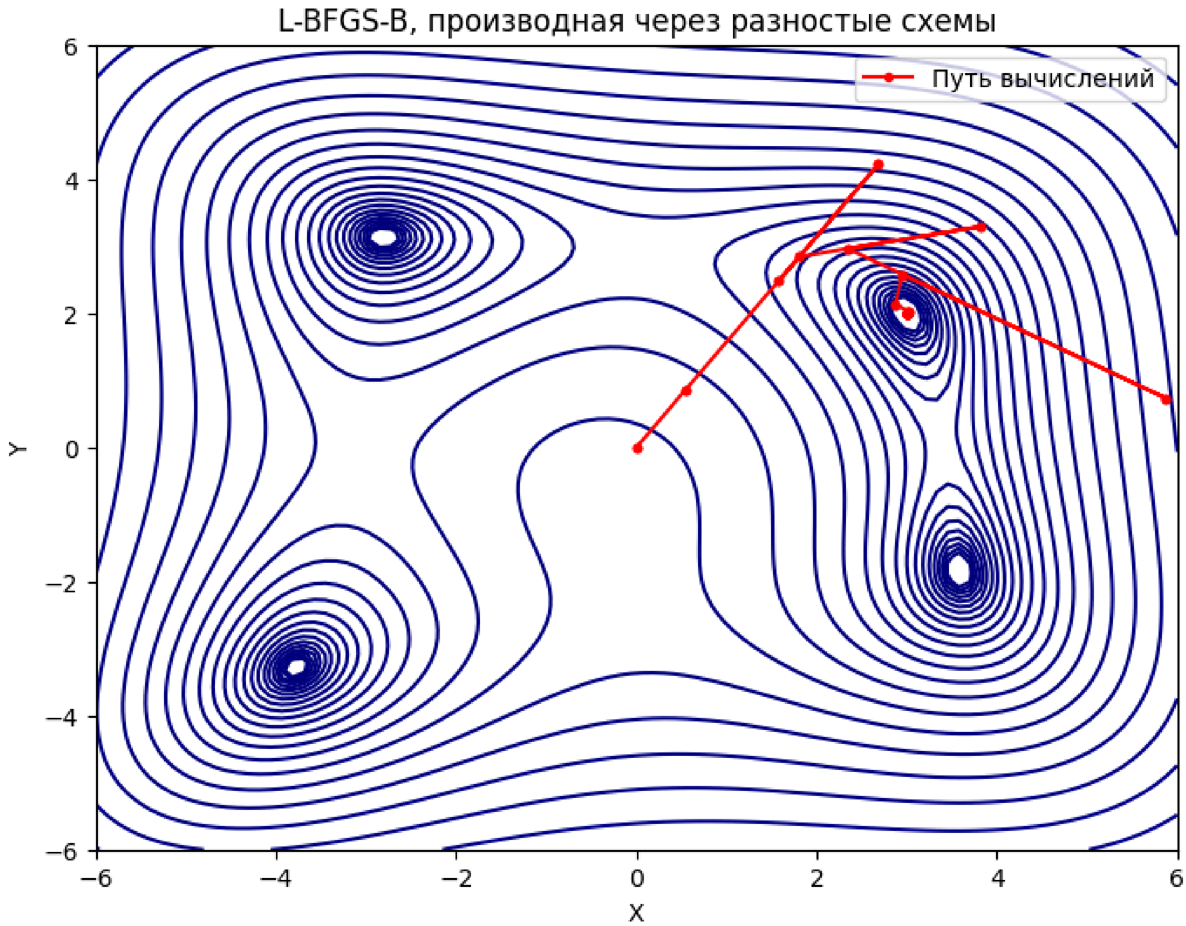
Функция Химмельблау, x0 = (0, 0), tol = 1e-18:

Метод	Предполагаемая x* (округлено)	f(x*) (округлено)	Количество итераций	Количество вычислений f
Nelder-Mead	(3, 2)	0	168	327
BFGS	(3, 2)	0	13	152
L-BFGS	(3, 2)	0	13	60

Все методы приходят в один локальный минимум.


Neldear-Mead
Путь вычислений
Y
X
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6


BFGS, производная через разностые схемы
Путь вычислений
Y
X
6
4
2
0
−2
−4
−6
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0



Значительное различие в количестве вычисляемых значений, нужных Нелдер-Миду в сравнении с квазиньютоновскими методами, пусть и с аппроксимированным градиентом, вполне отражается на графиках.