

Методы оптимизации

Отчёт по лабораторной работе №2

Выполнили:

Евсеева Арина М3234

Шульпин Егор М3232

Гаврилюк Виталий М3232

Постановка задачи

1. Метод Ньютона с постоянным шагом;
2. Метод Ньютона с использованием одномерного поиска (dichotomy);
3. Модификации метода Ньютона:
 - Метод Ньютона с использованием метода сопряженных градиентов (Newton-CG);
 - Квазиньютоновский метод (BFGS);
4. Сравнение методов на примере функции Розенброка и не полиномиальной функции:
 - Сравнение своей реализации метода Ньютона с методом Newton-CG из библиотеки `scipy.optimize`;
 - Сравнение эффективности методов нулевого порядка и градиентного спуска из лабораторной работы №1 с методом Ньютона и квазиньютоновскими методами;
 - Сравнение эффективности методов нулевого порядка с квазиньютоновскими методами с вычислением производных разностным методом;
 - Иллюстрация работы методов с помощью графиков и таблиц;
5. Дополнительная реализация метода Ньютона с использованием одномерного поиска по правилу Вольфе и дальнейшее сравнение;
6. Исследование реализованных методов в сложных случаях:
 - Исследование работоспособности метода Ньютона в зависимости от выбранной начальной точки;
 - Исследование результатов каждого из методов на функции со многими точками локального минимума.

Метод Ньютона с постоянным шагом

Описание метода

Метод Ньютона - это итерационный метод нахождения нуля заданной функции, который впервые был предложен Исааком Ньютоном. Данный метод может быть использован для нахождения минимума функции в задачах оптимизации. Известно, что необходимым условием минимума функции в точке x является равенство ее градиента нулю. Таким образом, для поиска минимума мы можем воспользоваться методом Ньютона для решения уравнения вида: $\nabla f(x) = 0$. В стандартной реализации метода Ньютона подразумевается, что исходная функция является дифференцируемой по каждому аргументу (возможно, несколько раз).

Релизация метода

Алгоритм метода Ньютона:

1. Выбор начальной точки $x = (x_0, \dots, x_n)$ - начальное приближение.
2. На каждой итерации вычисляются новые значения аргумента функции:
$$x_{i+1} = x_i - H^{-1}(x_i) \cdot \nabla f(x_i),$$
 где H - матрица Гессе заданной функции.
3. Условие останова: $\|x_{i+1} - x_i\| < \mathcal{E}$ или $\|H^{-1} \cdot \nabla f(x_i)\| < \mathcal{E}$.

Метод Ньютона с использованием одномерного поиска (dichotomy)

Реализация метода Ньютона с использованием дихотомии

Как и в случае с градиентным спуском, использование фиксированного шага может сказаться на работоспособности: замедление приближения к минимуму или его перепрыгивание. Также метод Ньютона очень восприимчив к начальной точке, поэтому при неверно подобранном шаге на каждой итерации может происходить отдаление в неверном направлении. Для поиска оптимального шага можно использовать метод одномерного поиска, например, дихотомии: по каждому направлению подбираем значение на отрезке с границами $l = \mathcal{E}$, $r = 50$.

Реализация метода

Алгоритм очень схож с предыдущим, однако на каждой итерации будем находить (с помощью дихотомии) α - оптимальный шаг. Тогда $x_{i+1} = x_i - \alpha \cdot H^{-1}(x_i) \cdot \nabla f(x_i)$.

Модификации метода Ньютона

Метод Ньютона с использованием метода сопряженных градиентов (Newton-CG)

Описание метода

Главным недостатком стандартной реализации метода Ньютона является необходимость хранения гессиана, который слишком неэффективен по памяти и вычислению при большом количестве данных. Использование метода сопряженных градиентов является модификацией метода Ньютона, позволяющей на каждом шаге выбирать направление с использованием градиентов функции. Данный метод позволяет (приблизительно) обращать гессиан, чем уменьшает сложность вычислений и устраняет проблему с нехваткой памяти.

Реализация метода

Реализация метода Ньютона с использованием метода сопряженных градиентов взята из библиотеки `scipy.optimize`, анализ эффективности которой будет рассмотрен далее.

Квазиньютоновский метод (BFGS)

Описание метода

Метод BFGS (Broyden–Fletcher–Goldfarb–Shanno) - это метод для решения задач оптимизации без ограничений. Данный метод относится к классу квазиньютоновских, которые вычисляют приближенное значение обратного гессиана с помощью информации о градиентах. На каждой итерации метод BFGS обновляет аппроксимацию обратной матрицы Гессе на основе изменения градиентов и значений аргументов. После изменения аппроксимации данный метод использует полученное значение для определения направления.

Реализация метода

Реализация квазиньютоновского метода BFGS взята из библиотеки `scipy.optimize`, анализ эффективности которой будет рассмотрен далее.

Сравнение эффективности методов

Сравнение метода Ньютона с методом Newton-CG

Выбор исследуемых функций

1) $f = (1 - x)^2 + 100 \cdot (y - x^2)^2$ (функция Розенброка)

Данная функция имеет минимум в точке $(x, y) = (1, 1)$ (при $-1.5 \leq x \leq 4, -3 \leq y \leq 4$), где принимает значение равное 0. Она является хорошим примером тестовой функции для сравнения эффективности работы каждого метода.

2) $f = \sin(x + y) + (x - y)^2 - 1.5 \cdot x + 2.5 \cdot y + 1$ (функция МакКормика)

Данная функция имеет глобальный минимум в точке $(x, y) = (-0.54719, -1.54719)$, где принимает значение равное -1.9133 . Она подойдет для проверки работоспособности каждого метода при различном выборе начальных точек.

Исследование точности и скорости

Начальные параметры:

- $f = (1 - x)^2 + 100 \cdot (y - x^2)^2$;
- $x_0 = (-2, 2)$;
- $\mathcal{E} \in \{10^{-5}, 10^{-10}\}$;
- $l = \mathcal{E}, r = 50$ (при использовании метода дихотомии).

Точность полученных значений:

eps	Newton (constant)	Newton (dichotomy)	Newton-CG
1e -5	f(1, 1) = 0	f(1, 1) = 2.57492e-25	f(1, 1) = 1.02396e-14
1e -10	f(1, 1) = 0	f(1, 1) = 3.51323e-22	f(1, 1) = 1.13311e-18

Количество итераций и обращений к функции при заданной точности:

eps	Newton (constant)	Newton (dichotomy)	Newton-CG
1e -5	6 итер. 0 функ. 6 град. 6 гесс.	15 итер. 900 функ. 15 град. 15 гесс.	277 итер. 300 функ. 993 град. 0 гесс.
1e -10	7 итер. 0 функ. 7 град. 7 гесс.	16 итер. 960 функ. 16 град. 16 гесс.	279 итер. 301 функ. 995 град. 0 гесс.

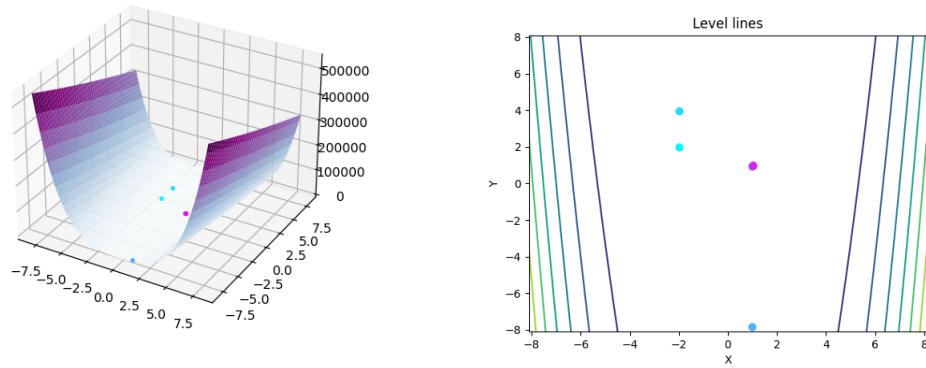


Рис. 1: Метод Ньютона

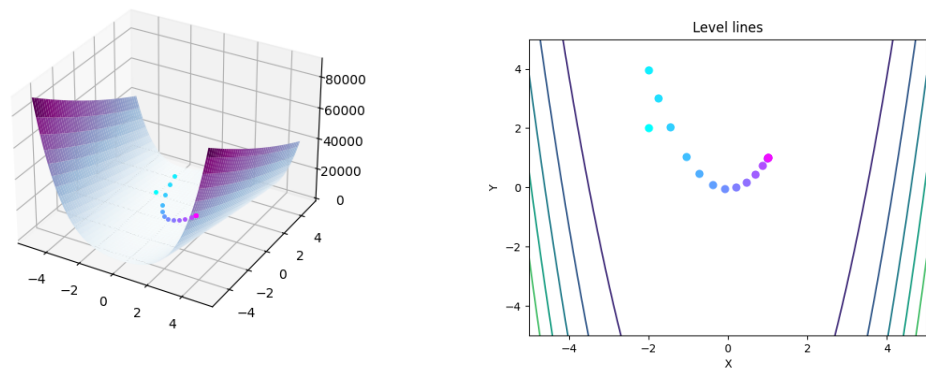


Рис. 2: Метод Ньютона с использованием дихотомии

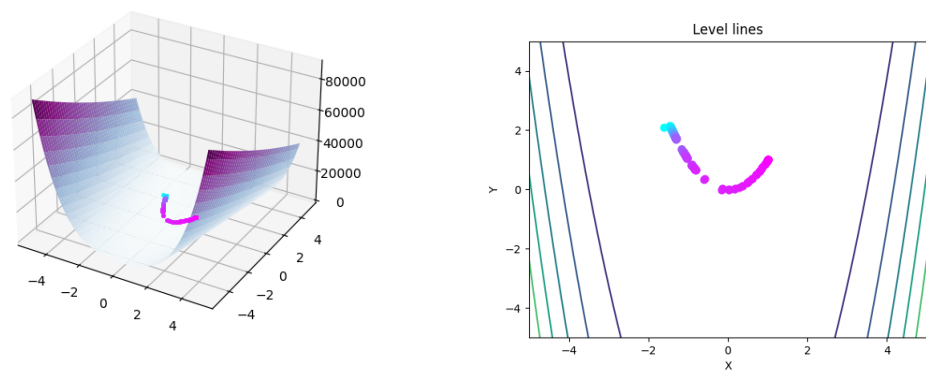


Рис. 3: Метод Ньютона с использованием метода сопряженных градиентов

Исследование работоспособности в зависимости от выбранной начальной точки

Начальные параметры:

- $f = \sin(x + y) + (x - y)^2 - 1.5 \cdot x + 2.5 \cdot y + 1$;
- $x_0 \in \{(0, -3), (2.7, -1.5)\}$;
- $\mathcal{E} = 10^{-7}$;
- $l = \mathcal{E}, r = 100$ (при использовании метода дихотомии).

Метод Ньютона

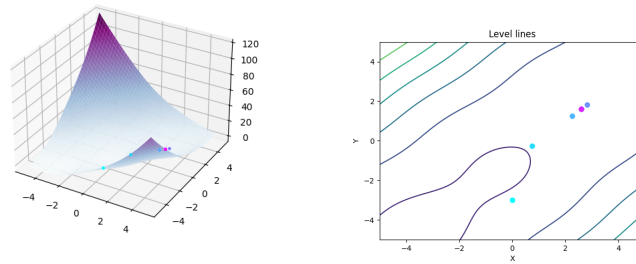


Рис. 4: Некорректное значение при $x_0 = (0, -3)$

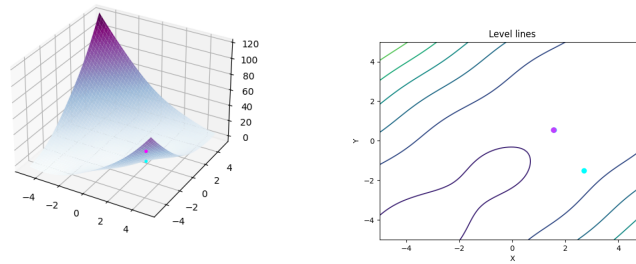


Рис. 5: Некорректное значение при $x_0 = (2.7, -1.5)$

Метод Ньютона с использованием дихотомии

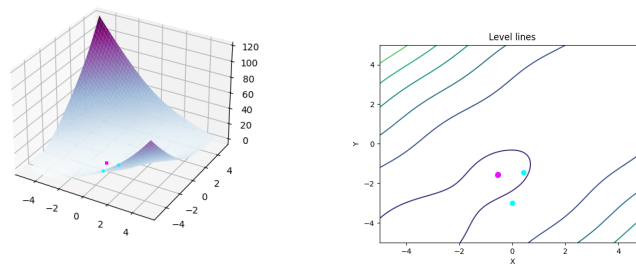


Рис. 6: Корректное значение при $x_0 = (0, -3)$

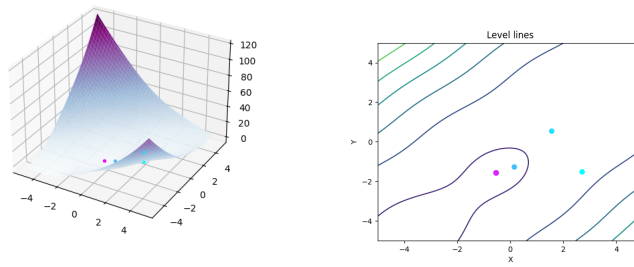


Рис. 7: Корректное значение при $x_0 = (2.7, -1.5)$

Метод Ньютона с использованием метода сопряженных градиентов

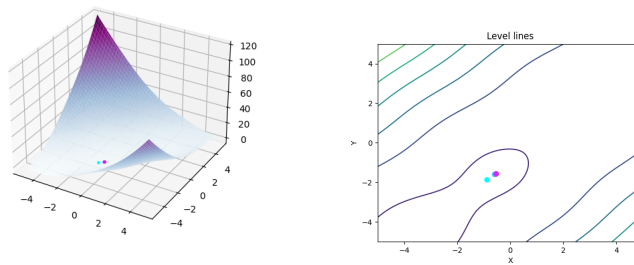


Рис. 8: Корректное значение при $x_0 = (0, -3)$

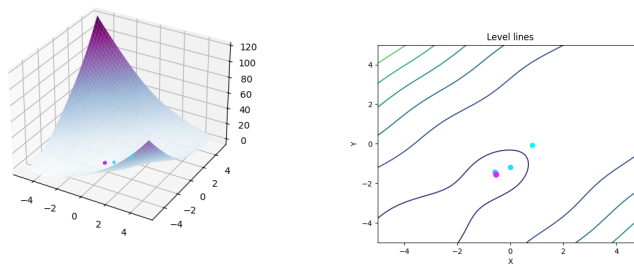


Рис. 9: Корректное значение при $x_0 = (2.7, -1.5)$

Полученные результаты

Из проведенных сравнений можно сделать вывод, что метод Ньютона довольно быстро находит минимум функции, однако его работоспособность очень зависит от выбора начальной точки, что и показало второе сравнение. Использование дихотомии, как и было сказано ранее, решает данную проблему, однако, в некоторых случаях, может требовать большего количества итераций алгоритма. Также стоит упомянуть, что данный метод все еще использует гессиан, вычисление и хранение которого очень затратно. Таким образом, исходя из полученных результатов, можно сказать, что использование сопряженных градиентов является выходом из этой ситуации, так как он сравним с использованием дихотомии по работоспособности и использует меньшее количество ресурсов.

Сравнение эффективности методов: метода Нелдера-Мида, метод градиентного спуска, метод Ньютона и квазиньютоновских методов

При рассмотрении метода Ньютона и метода градиентного спуска будем использовать дихотомию, так как в первой лабораторной работе мы нашли точку, начиная с которой, метод сходится.

Выбор исследуемой функции

$f = -\cos(x) \cdot \cos(y) \cdot \exp(-((x - \pi)^2 + (y - \pi)^2))$ (функция Изома).

Данная функция имеет минимум в точке $f(\pi, \pi) = -1$.

Начальные параметры:

- $f = -\cos(x) \cdot \cos(y) \cdot \exp(-((x - \pi)^2 + (y - \pi)^2))$;
- $(x_0, y_0) = (1.71, 1.71)$;
- $\mathcal{E} = 10^{-5}$;
- $l = \mathcal{E}, r = 30$.

Количество итераций и обращений к функции:

eps	Gradient descent	Newton	Needer-Mead	BFGS
10e -5	4 итер. 184 функ. 8 град. 0 гесс.	Не находит!	48 итер. 95 функ. 0 град. 0 гесс.	2 итер. 12 функ. 12 град. 0 гесс.

Метод градиентного спуска

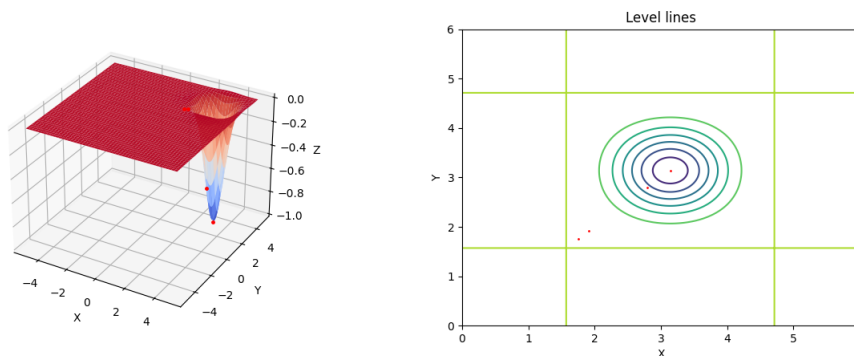


Рис. 10

Метод Ньютона

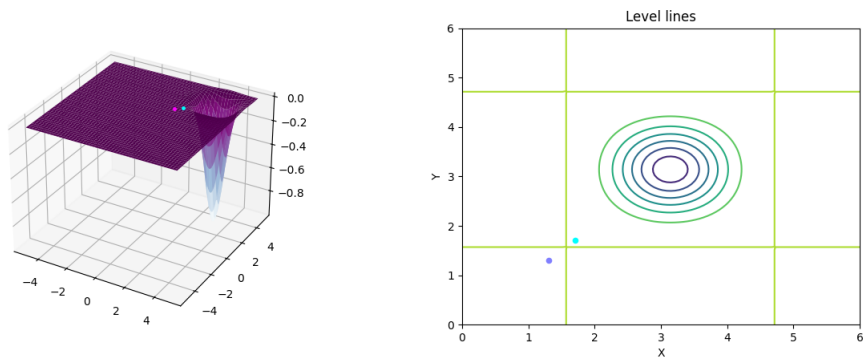


Рис. 11

Метод Нелдера-Мида

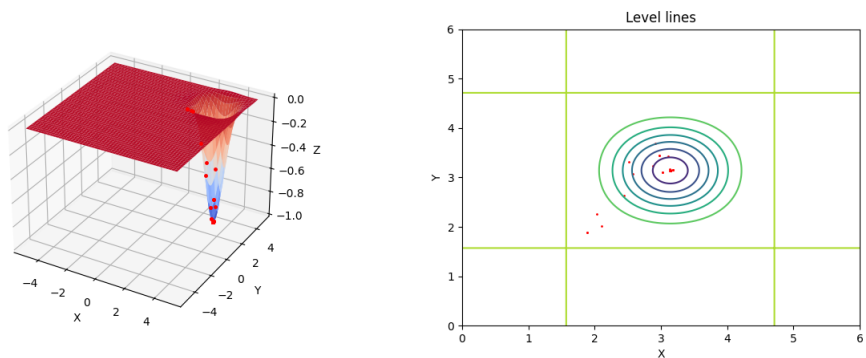


Рис. 12

BFGS

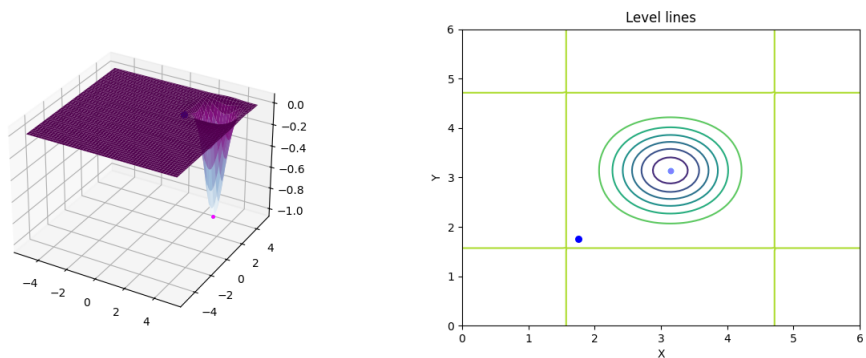


Рис. 13

Полученные результаты

На данном примере видно, что метод Ньютона не смог найти минимум, так как начальное приближение оказалось не достаточно близким. При этом градиентный спуск был относительно

легким в вычислении, метод Нелдера-Мида не использовал градиент, а BFGS потребовал меньше всего итераций. Плюсом этих методов стало и то, что они не использовали гессиан.

Сравнение эффективности методов нулевого порядка с квазиньютоновскими методами с вычислением производных разностным методом

При большом количестве данных возникает проблема с получением, хранением и вычислением градиентов функции. В реализации квазиньютоновского метода BFGS из библиотеки `scipy.optimize` сделана возможность вычисления производных с помощью разностных методов, например:

- $f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$ (2-point);
- $f'(x) \approx \frac{-f(x+2h) + 4f(x+h) - 3f(x)}{2h}$ (3-point).

Таким образом, мы смогли избавиться от явного получения и вычисления градиента функции, а значит, данный подход сравним с методом Нелдера-Мида из лабораторной работы №1.

Выбор исследуемой функции

Хорошей функцией для сравнения данных методов будет функция типа Розенброка. В общем виде она определена так: $f = (a-x)^2 + b(y-x^2)^2$ и имеет глобальный минимум в точке $f(a, a^2) = 0$. Возьмем $a = 3, b = 20$, тогда минимум находится в точке $(3, 9)$. Сравним количество итераций каждого метода при начальной точке, которая довольно удалена от минимума.

Начальные параметры:

- $f = (3-x)^2 + 20 \cdot (y-x^2)^2$;
- $(x_0, y_0) = (-2.5, -7.5)$;
- $\mathcal{E} = 10^{-7}$.

Метод Нелдера-Мида

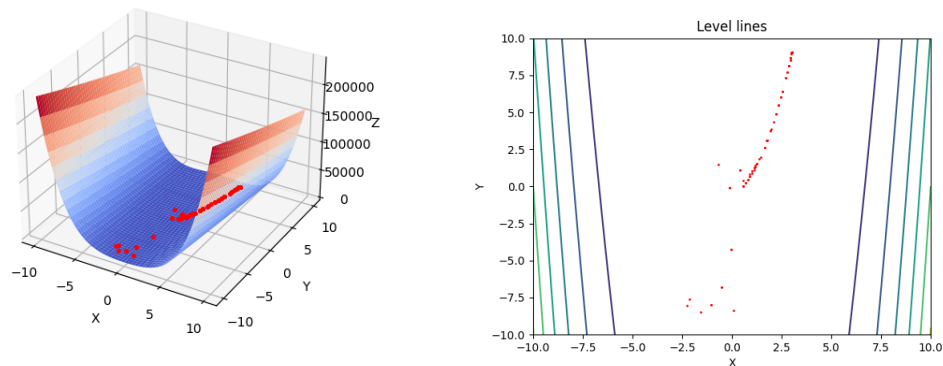


Рис. 14

Метод BFGS с вычислением производных разностным методом

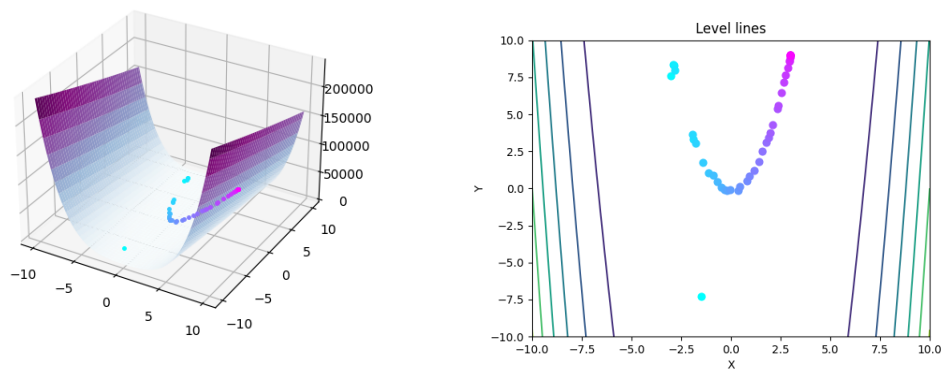


Рис. 15: 2-point

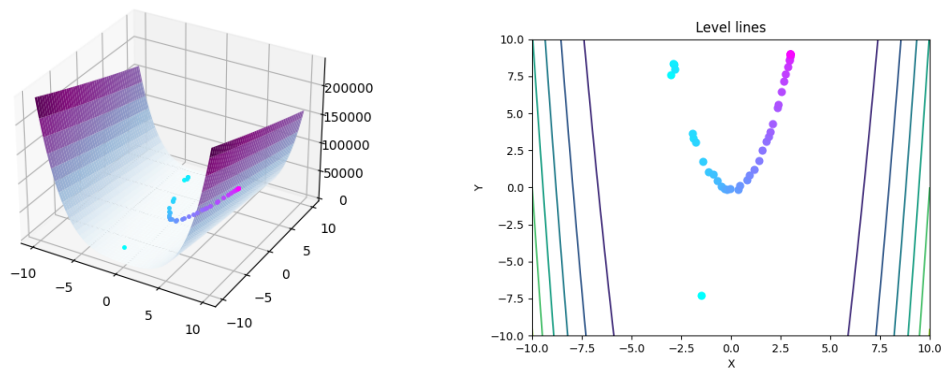


Рис. 16: 3-point

Количество итераций и обращений к функции:

eps	BFGS (2-point)	BFGS (3-point)	Needler-Mead
10e -7	41 итер. 159 функ.	41 итер. 265 функ.	97 итер. 182 функ.

Полученные результаты

Из представленного сравнения можно сделать вывод, что оба метода требуют больше итераций, чем метод Ньютона, использующий гессиан. Однако можно выделить огромное преимущество, заключающееся в том, что метод Нелдера-Мида и метод BFGS (с вычислением производных разностным методом) работают без вычисленных ранее градиентов. Тем самым, мы полностью можем избавиться от проблемы большой затраты памяти.

Дополнительное задание 1

Описание метода

Одномерный поиск по правилу Вольфе - это метод численной оптимизации, используемый для нахождения шага при минимизации или максимизации функции. Он основан на сочетании условий убывания функции (например, условия Армихо) и условий кривизны (по формулировке Вольфа). Этот метод обеспечивает эффективное движение к оптимальной точке, учитывая как направление убывания, так и кривизну функции. Условия Вольфе используются для выбора оптимального шага, удовлетворяющего двум условиям:

- Условие Армихо: новое значение функции должно быть достаточно малым по сравнению со значением функции в текущей точке и должно быть больше, чем значение функции на предыдущем шаге.
- Условие кривизны гарантирует, что градиент в новой точке не слишком маленький по сравнению с градиентом в старой точке. Это помогает избежать слишком маленьких шагов и ускоряет сходимость градиентного спуска к оптимальному решению.

$$f(x_k + \alpha \cdot p_k) \leq f(x_k) + c_1 \cdot \alpha \cdot \text{grad}(f_k^T) \cdot p_k, \text{ где } 0 < c_1 < 1$$

$$\text{grad}(f(x_k + \alpha \cdot p_k)^T) \cdot p_k \geq c_2 \text{grad}(f_k^T) \cdot p_k, \text{ где } c_1 < c_2 < 1$$

Метод Ньютона на основе метода Вольфе

Метод Ньютона использует гессиан (матрицу вторых производных функции) для определения направления поиска, что позволяет более точно двигаться к точке минимума. Однако, прямое использование метода Ньютона может быть неэффективным из-за вычисления и обращения гессиана на каждом шаге, а также из-за чувствительности к выбору начальной точки и размеру шага.

Реализация метода

На каждой итерации метода Ньютона мы используем одномерный поиск по правилу Вольфе для определения оптимального шага α . Затем обновляем текущую точку по формуле:

$$x_{i+1} = x_i - \alpha \cdot H^{-1}(x_i) \cdot \nabla f(x_i)$$

Начальные параметры:

- $f = (1 - x)^2 + 100 \cdot (y - x^2)^2$;
- $(x_0, y_0) = (-1, -1)$;
- $\mathcal{E} = 10^{-5}$;

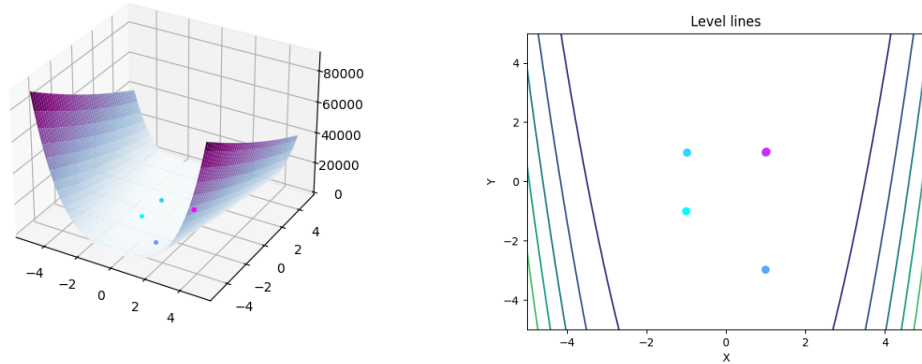


Рис. 17: Метод Ньютона с использованием метода Вольфе на функции Розенброка

Полученные результаты

Метод Ньютона на основе метода Вольфе адаптирует размер шага α на каждой итерации с использованием одномерного поиска, что позволяет более точно и эффективно находить минимум функции, учитывая условия достаточного уменьшения и кривизны. Это улучшает сходимость метода и делает его более устойчивым к выбору начальной точки и размеру шага, в отличие от традиционного метода Ньютона, который может быть неэффективным из-за необходимости вычисления и обращения гессиана на каждом шаге.

Метод Ньютона с использованием сопряженных градиентов (Newton-CG) минимизирует необходимость прямого вычисления и обращения гессиана, используя приближенное решение системы линейных уравнений для определения направления поиска. Это делает Newton-CG особенно подходящим для задач большой размерности, где вычисление и хранение гессиана становится вычислительно затратным.

В сравнении, метод Ньютона на основе метода Вольфе предлагает баланс между точностью и вычислительной эффективностью, адаптируя размер шага для улучшения сходимости без необходимости обращения к методам приближенного решения, как в Newton-CG.

Дополнительное задание 2

1) В основной части исследования (пункт 2) мы рассмотрели функцию Изома, которая является примером и для этого пункта. Все графики и выводы можно увидеть [тут](#).

2) Рассмотрим функция Химмельблау $f = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$. Она имеет 4 локальных минимума:

- $f(3, 2) = 0$;
- $f(-2,805118..., 3,131312...) = 0$;
- $f(-3,779310..., -3,283186...) = 0$;
- $f(3,584428..., -1,848126...) = 0$.

Начальные параметры:

- $f = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$;
- $(x_0, y_0) = (-4, 0)$;

- $\mathcal{E} = 10^{-5}$;
- $l = \mathcal{E}, r = 30$.

Метод градиентного спуска

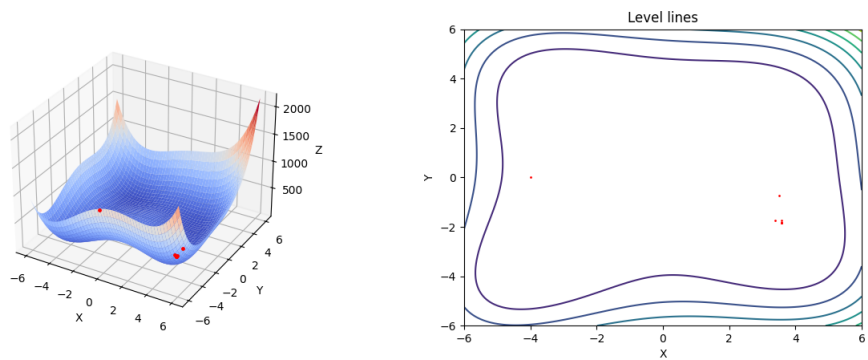


Рис. 18: $f(3.58437, -1.84809) = 0$

Метод Ньютона

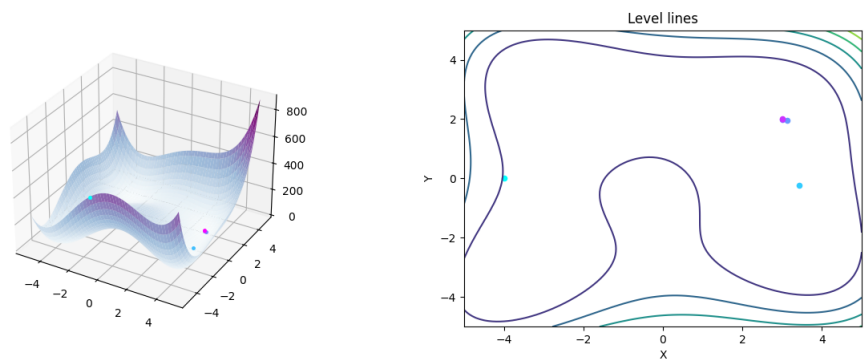


Рис. 19: $f(3, 2) = 0$

BFGS

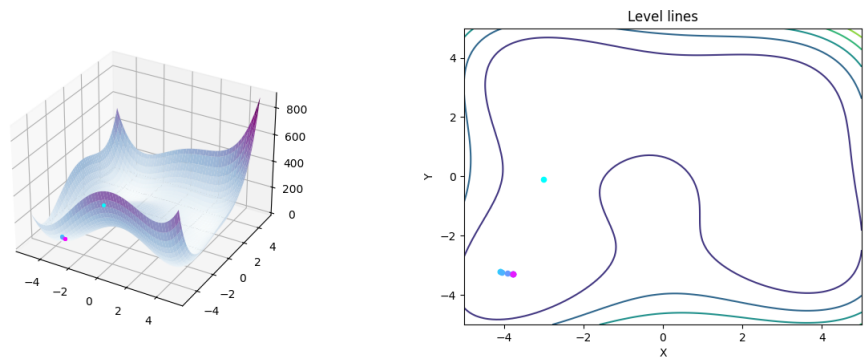


Рис. 20: $f(-3.779310..., -3.283186...) = 0$

Метод Нелдера-Мида

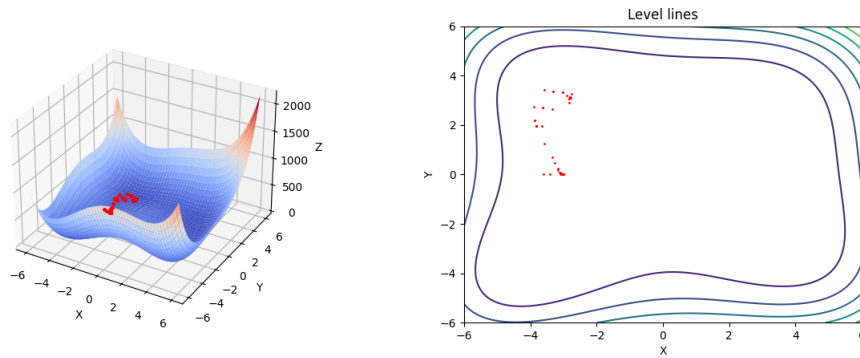


Рис. 21: $f(-2,805118..., 3,131312...) = 0$

Полученные результаты

Метод Ньютона использует точное значение гессиана функции или его аппроксимацию для определения направления спуска. Он может быть более быстрым в сходимости к минимуму, чем градиентный спуск, но он также более чувствителен к выбору начальной точки и может застревать в локальных минимумах или точках седловых.

Градиентный спуск, с другой стороны, использует только информацию о градиенте функции, который указывает направление наискорейшего убывания. Это более простой метод, который может быть менее чувствителен к выбору начальной точки, но он может потребовать большего количества итераций для достижения минимума, особенно если функция имеет узкие долины или сильно вытянутые формы. Он требует меньше памяти, чем метод Ньютона.

Метод BFGS обновляет оценку обратного гессиана по мере продвижения по итерациям, используя информацию о градиентах. Он приближает гессиан итеративно и адаптивно, что делает его более эффективным для оптимизации в случаях, когда точное вычисление гессиана затруднительно или дорого. Он также обычно требует меньше памяти, чем метод Ньютона.

Метод Нелдера-Мида не использует информацию о градиенте или гессиане функции и оперирует только значениями функции в различных точках. Это позволяет ему исследовать пространство параметров более равномерно и гибко, что может помочь в обходе локальных минимумов, но при этом он может потребовать больше итераций для сходимости.