

Национальный исследовательский университет ИТМО  
Факультет информационных технологий и программирования  
Прикладная математика и информатика

**Методы оптимизации**

Отчет по лабораторной работе №1

**Работу выполнили:**

Захаров Кирилл М32391

Мавлютов Эрвин М32391

Шилкин Артем М32391

**Преподаватель:**

Шохов Максим Евгеньевич

г. Санкт-Петербург

2023 г.

## Оглавление

Задание 1. Градиентный спуск с постоянным шагом	3
Задание 2. Метод дихотомии и градиентного спуска на его основе	4
Задания 3-4. Анализ траекторий градиентного спуска на примере квадратичных функций	5
Первая функция	5
Вторая функция	9
Третья функция	13
Выводы	15
Задание 5. Генератор случайных квадратичных функций	16
Задания 6-7. Исследование зависимости числа итераций	17
Дополнительное задание	18
Выводы по эффективности	22

## Постановка задач:

1. Реализация градиентного спуска с постоянным шагом
2. Реализация метода одномерного поиска (метод дихотомии) и градиентного спуска на его основе.
3. Анализ траекторий градиентного спуска на примере квадратичных функций.
4. Реализация генератора случайных квадратичных функций  $n$  переменных с числом обусловленности  $k$ .
5. Исследование зависимости числа итераций, необходимых градиентному спуску для сходимости в зависимости от размерности пространства и числа обусловленности функции.

Весь проект можно найти по [ссылке](#).

## Задание 1. Градиентный спуск с постоянным шагом

При поиске локального минимума методом градиентного спуска последовательность приближений определяется формулой  $x_{n+1} = x_n - \alpha \nabla f(x_n)$ , где  $\alpha > 0, x_n \in \mathbb{R}^n$ .

Следовательно, необходимо реализовать следующий алгоритм

- 1) Выбрать случайную точку  $x_0$ ;
- 2) Вычислить градиент в текущей позиции;
- 3) Перейти к точке с номером  $k + 1$  по формуле выше, вернуться к 2).

Градиент в точке  $x_k$  можно приближенно вычислять по какой-то из следующих формул:

1. 
$$\nabla f(x_k)_i = \frac{f(x_k + \varepsilon e_i) - f(x_k - \varepsilon e_i)}{2\varepsilon}$$
2. 
$$\nabla f(x_k)_i = \frac{f(x_k + \varepsilon e_i) - f(x_k)}{\varepsilon}$$

Был реализован градиентный спуск с постоянным шагом – [функция](#) `gradient_descent`, принимающая два аргумента:

- $f$  – исследуемая функция, принимающая массив координат;
- `start_pos` – начальная точка, размерность совпадает с размерностью функции;
- `max_count` – максимальное количество итераций при попытке найти точку минимума;
- `draw` – булевское значение, обозначающее необходимость рисования графика

и возвращающая пару из возможной точки минимума функции  $f$  и числа шагов до сходимости (если функция не сошлась, то `max_count + 1`).

## Задание 2. Метод дихотомии и градиентного спуска на его основе

Метод дихотомии позволяет численно найти корень на отрезке, где в начальной и конечной точках функция принимает значения разных знаков.

Идея заключается в следующем: мы строим последовательность отрезков

$$[a_{k+1}, b_{k+1}] = \begin{cases} \left[ a_k, \frac{a_k + b_k}{2} \right], & f\left(\frac{a_k + b_k}{2}\right) > 0 \\ \left[ \frac{a_k + b_k}{2}, b_k \right], & f\left(\frac{a_k + b_k}{2}\right) \leq 0 \end{cases},$$

которая, при непрерывности исходной функции, в пределе будет равна точке со значением 0.

Можно искать минимум гладкой на отрезке функции, если на нем имеется лишь один локальный экстремум, применив метод дихотомии к производной.

Теперь понятно, как можно реализовать градиентный спуск:

- 1) Выбрать случайную точку  $x_0$ ;
- 2) Вычислить градиент в текущей позиции;
- 3) Сузить функцию на прямую вдоль направления градиента:  
 $\varphi_k(t) = f(x_k - t \nabla f(x_k))$ ;
- 4) Найти отрезок  $[0, c]$ , выполняющий требуемые условия для метода дихотомии;
- 5) Найти на этом отрезке локальный минимум  $d \in (0, c)$ ;
- 6) Перейти к точке  $k + 1$ :  $x_{k+1} = x_k - d \nabla f(x_k)$ .

Одним из проблемных пунктов этого метода является поиск  $c$ . Его можно попытаться найти, начав с достаточно малого значения и увеличивая в два раза до успеха.

Был реализован градиентный спуск на основе дихотомии – [функция `gradient\_descent\_advanced`](#), имеет сигнатуру такую же, как и функция в предыдущем пункте.

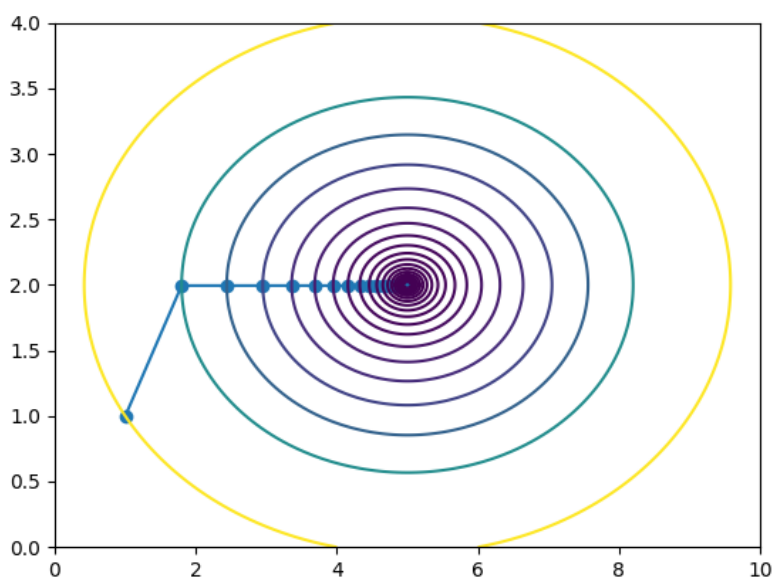
### Задания 3-4. Анализ траекторий градиентного спуска на примере квадратичных функций

Рассмотрим следующие квадратичные функции:

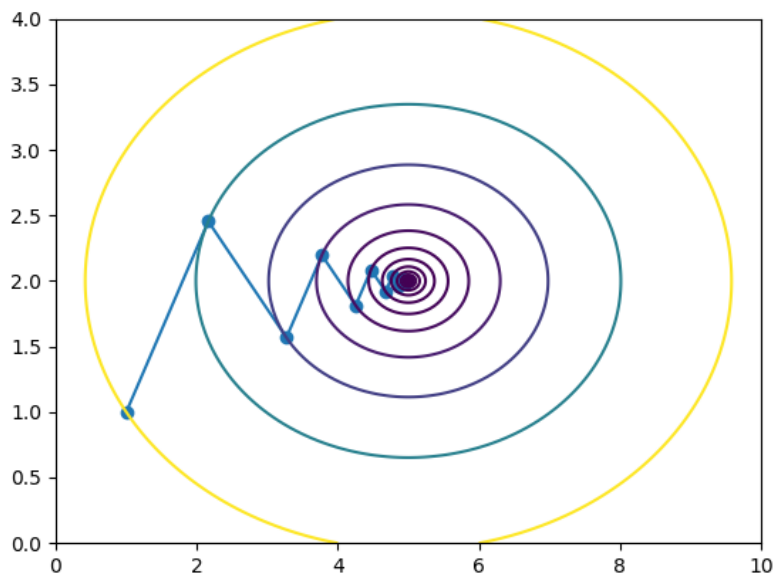
$$\begin{aligned}f_1(x, y) &= (x - 5)^2 + 5(y - 2)^2 \\f_2(x, y) &= 20(x - 5)^2 + (y - 3)^2 \\f_3(x, y) &= (x - 5)^2 + (y - 4)^2\end{aligned}$$

Для каждой из них запустим поиск минимума каждым из алгоритмов, посчитаем количество вычислений градиента и подстановок в функцию, а также рассмотрим влияние нормализации координат на эффективность вычислений.

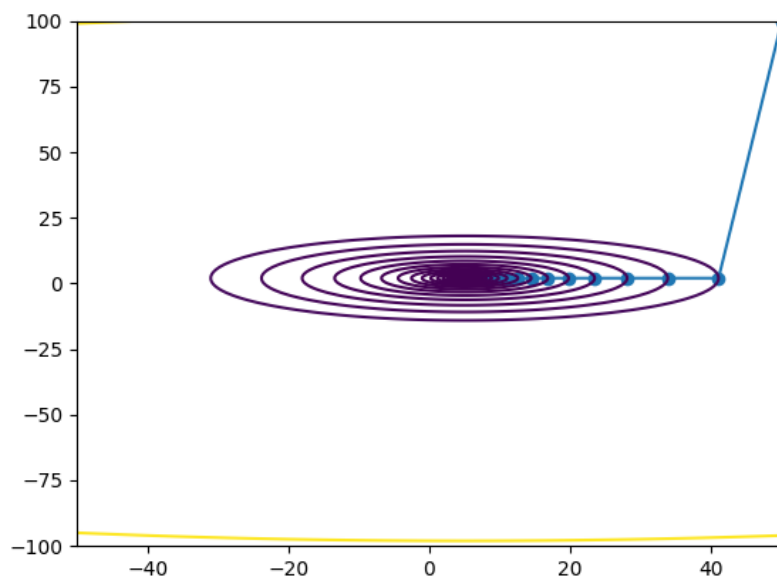
**Первая функция:**  $f_1(x, y) = (x - 5)^2 + 5(y - 2)^2$



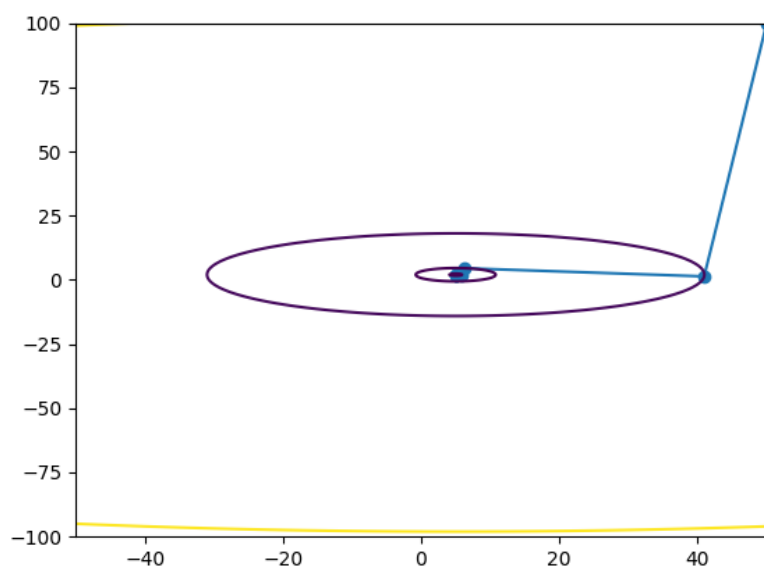
**Рис. 1** — градиентный спуск с постоянным шагом на первом примере из  $(1, 1)$ , вычислений градиента – 21, функции – 84



**Рис. 2** — градиентный спуск с методом дихотомии на первом примере из  $(1, 1)$ , вычислений градиента – 14, функции – 665

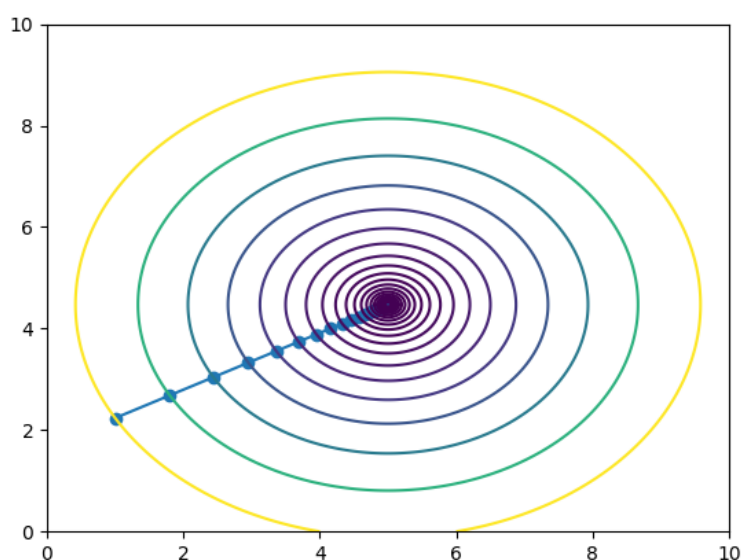


**Рис. 3** — градиентный спуск с постоянным шагом на первом примере из  $(50, 100)$ , вычислений градиента – 32, функции – 128

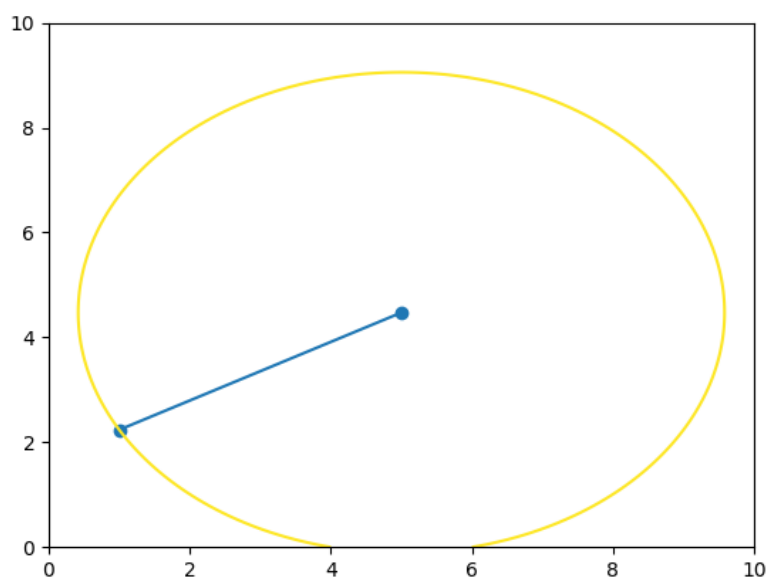


**Рис. 4** — градиентный спуск с методом дихотомии на первом примере из  $(50, 100)$ , вычислений градиента – 7, функции – 335

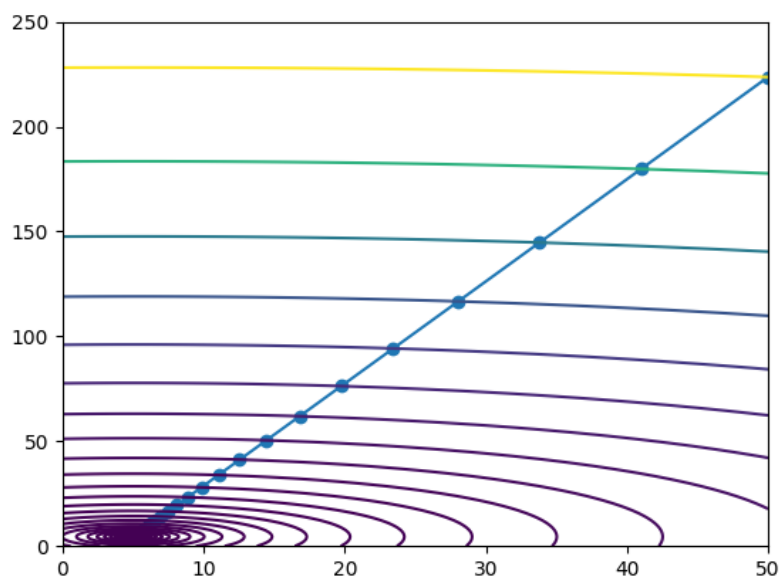
После замены переменных  $(\tilde{x} = x, \tilde{y} = \sqrt{5}y)$  получаем  $f = (\tilde{x} - 5)^2 + \tilde{y}^2 - 4\sqrt{5}\tilde{y} + 20$ ,  $(1, 1)$  переходит в  $(1, \sqrt{5})$ ,  $(50, 100)$  – в  $(50, 100\sqrt{5})$ .



**Рис. 5** — градиентный спуск с постоянным шагом на первом примере после замены переменной из близкой точки, вычислений градиента – 22, функции – 88

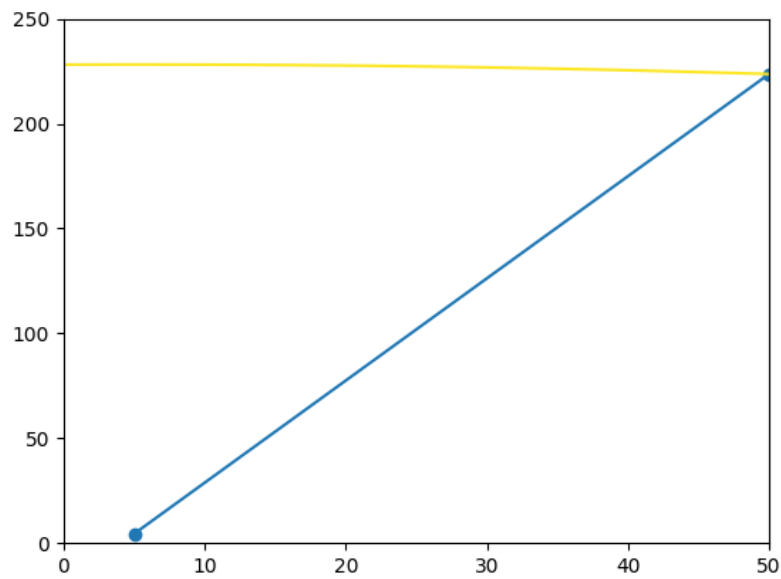


**Рис. 6** — градиентный спуск с методом дихотомии на первом примере после замены переменной из близкой точки, вычислений градиента – 2, функции – 95



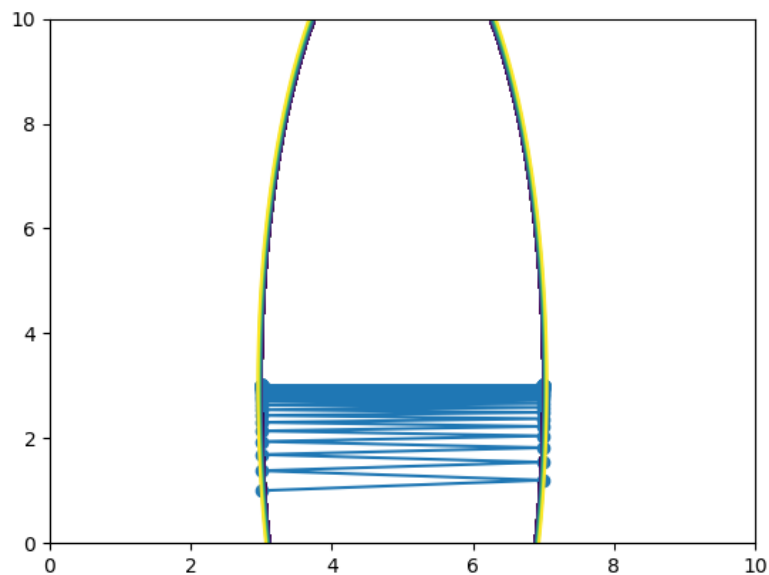
**Рис. 7** — градиентный спуск с постоянным шагом на первом примере после замены переменной из далекой точки, вычислений градиента – 39, функции – 156



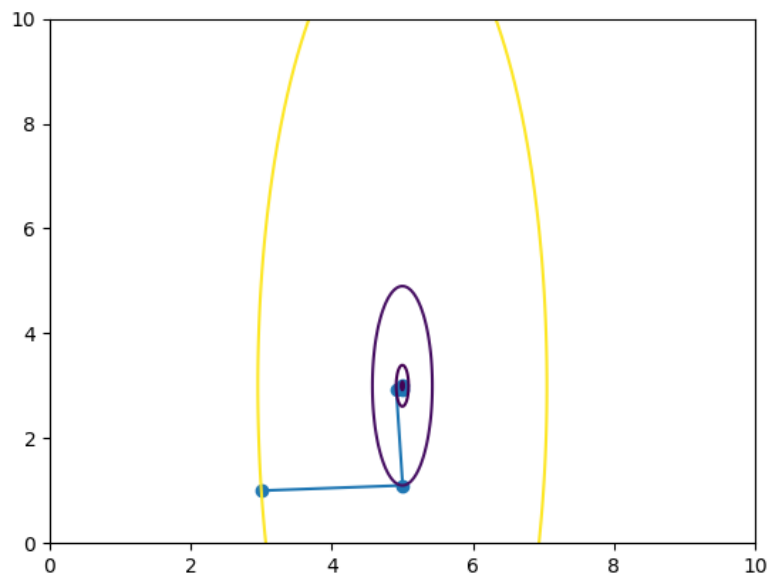


**Рис. 8** — градиентный спуск с методом дихотомии на первом примере после замены переменной из далекой точки, вычислений градиента – 2, функции – 98

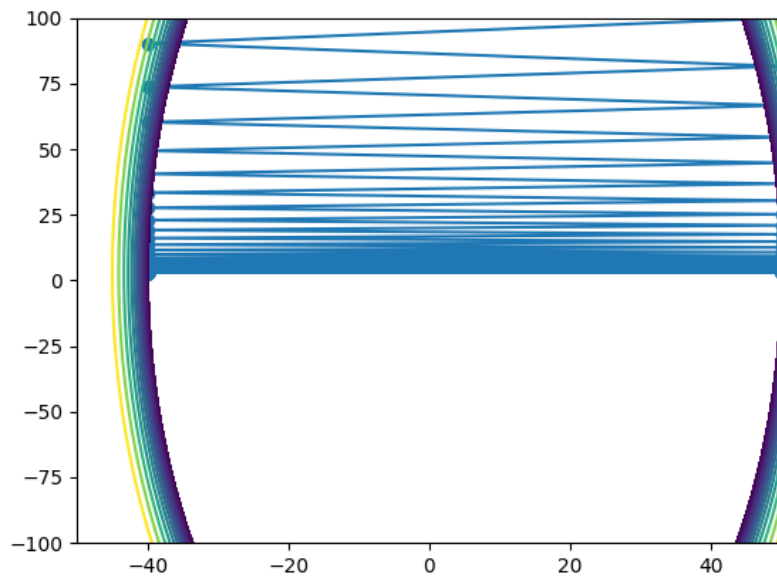
**Вторая функция:**  $f_2(x, y) = 20(x - 5)^2 + (y - 3)^2$



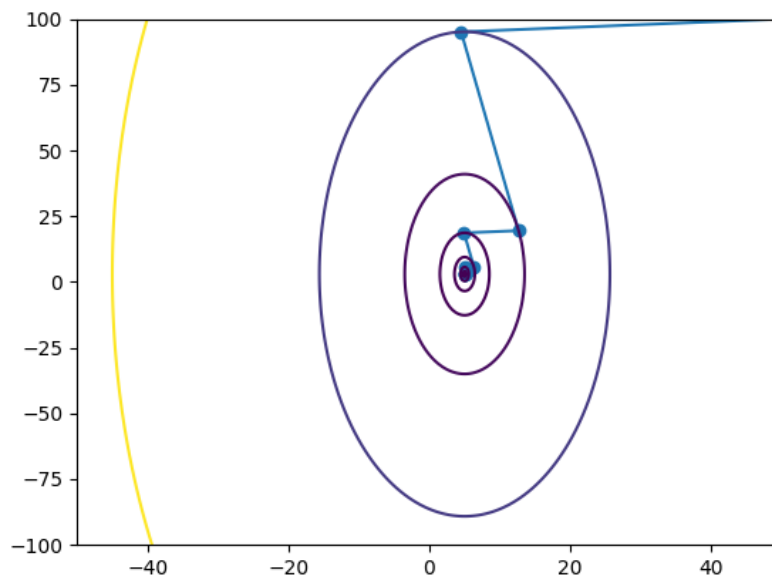
**Рис. 9** — градиентный спуск с постоянным шагом на втором примере из  $(3, 1)$ , вычислений градиента – 200, функции – 796 (не сошлось)



**Рис. 10** — градиентный спуск с методом дихотомии на втором примере из  $(3, 1)$ , вычислений градиента – 5, функции – 241

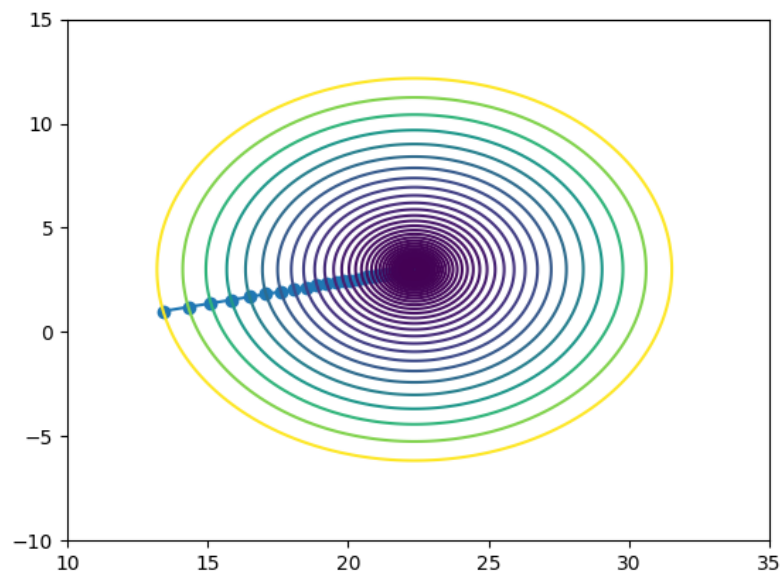


**Рис. 11** — градиентный спуск с постоянным шагом на втором примере из  $(50, 100)$ , вычислений градиента – 200, функции – 796 (не сошлось)

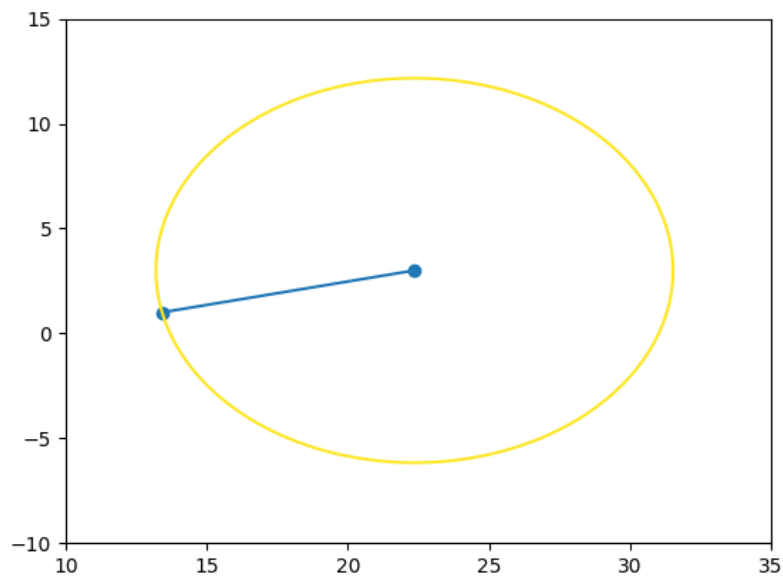


**Рис. 12** — градиентный спуск с методом дихотомии на втором примере из  $(50, 100)$ , вычислений градиента – 11, функции – 532

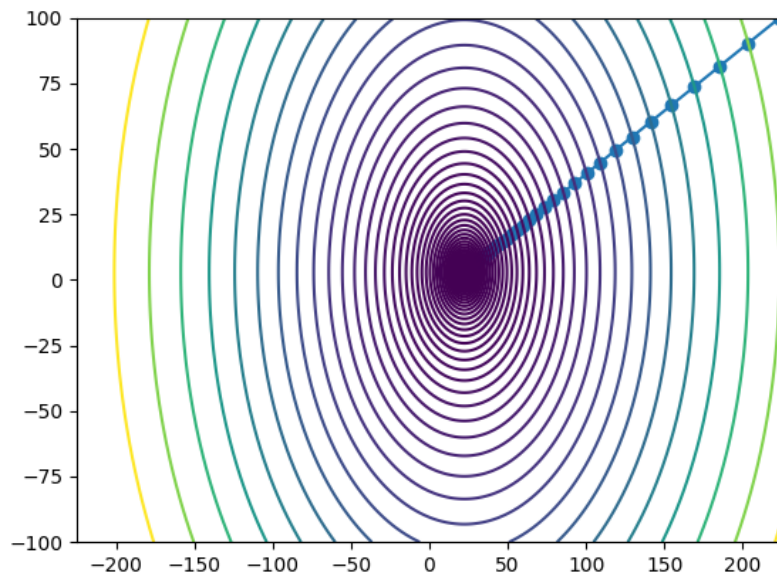
После замены переменных  $(x = \sqrt{20}x, y = y)$  получаем  $f = x^2 - 10\sqrt{20}x + 500 + (y - 3)^2$ ,  $(3, 1)$  переходит в  $(3\sqrt{20}, 1)$ ,  $(50, 100)$  – в  $(50\sqrt{20}, 100)$



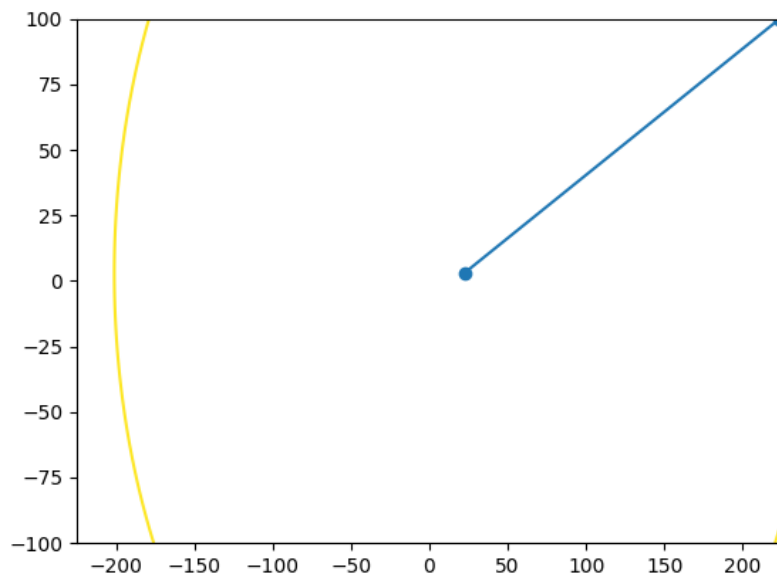
**Рис. 13** — градиентный спуск с постоянным шагом на втором примере после замены координат из близкой точки, вычислений градиента – 44, функции – 176



**Рис. 14** — градиентный спуск с методом дихотомии на втором примере после замены координат из близкой точки, вычислений градиента – 2, функции – 100

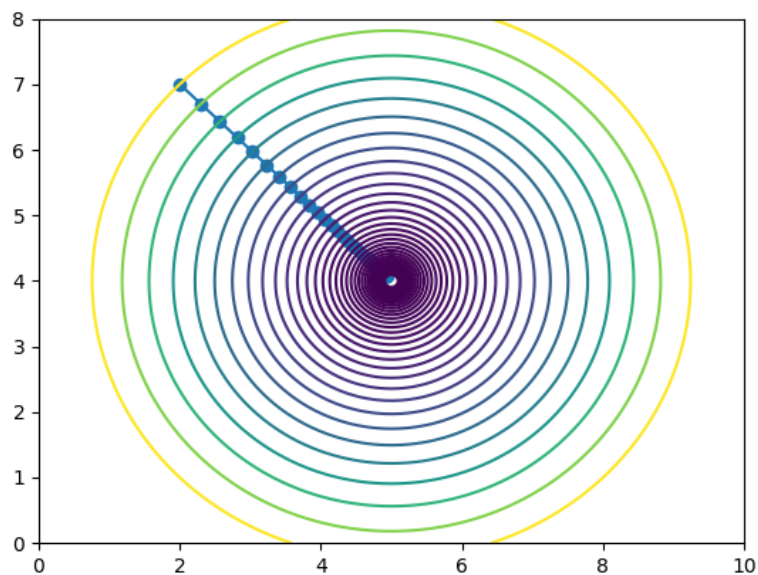


**Рис. 15** — градиентный спуск с постоянным шагом на втором примере после замены координат из дальней точки, вычислений градиента – 75, функции – 300

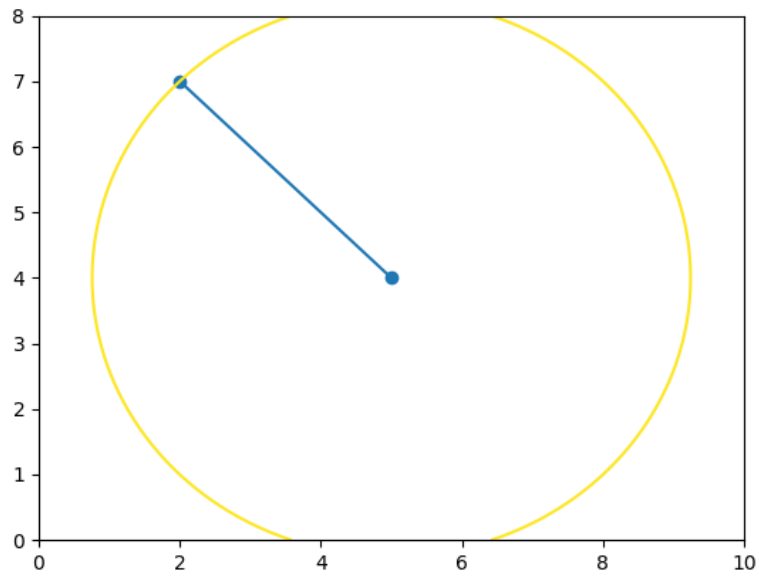


**Рис. 16** — градиентный спуск с методом дихотомии на втором примере после замены переменной из далекой точки, вычислений градиента – 2, функции – 100

**Третья функция:**  $f_3(x, y) = (x - 5)^2 + (y - 4)^2$



**Рис. 17** — градиентный спуск с постоянным шагом на третьем примере из  $(2, 7)$ , вычислений градиента – 37, функции – 148



**Рис. 18** — градиентный спуск с методом дихотомии на третьем примере из  $(2, 7)$ , вычислений градиента – 2, функции – 100

	фикс. шаг, градиент	дихотомия, градиент	фикс. шаг, функция	дихотомия, функция
$f_1$ , близкая т.	21	14	84	665
$f_1$ , далекая т.	32	7	128	335
$f_1$ , норм., близкая т.	22	2	88	95
$f_1$ , норм., далекая т.	39	2	156	98
$f_2$ , близкая т.	200	5	796	241
$f_2$ , далекая т.	200	11	796	532
$f_2$ , норм., близкая т.	44	2	176	100

$f_2$ , норм., далекая т.	75	2	300	100
$f_3$ , близкая т.	37	2	148	100

### Выводы:

- a) С увеличением расстояния от начальной точки до точки минимума количество вычислений для спуска с постоянным шагом ожидаемо растет, на функции с большим числом обусловленности метод с постоянным шагом может не сойтись.
- b) В большинстве случаев метод дихотомии делает меньше вычислений, благодаря тому, что делает более длинные шаги, в то время как метод с фиксированным шагом каждую итерацию считает градиент, который может не сильно отличаться от предыдущего. Однако если искомая точка близко, метод дихотомии наоборот, больше раз вычисляет значение функции.
- c) Понятно, что из далеких точек идти дальше и функция будет вычисляться большее количество раз, при этом для плохо обусловленной функции разница более существенна. Также результат сильно зависит от начальной точки: градиентный спуск с дихотомией для первой функции намного больше раз вычислял функцию для близкой точки, из-за того, что пришлось ходить зигзагами.
- d) Хорошая обусловленность функции позволяет спуску с методом дихотомии сразу получить направление на точку минимума, после чего алгоритм сразу сходится. На спуск с фиксированным шагом влияние не такое сильное, за исключением случая, когда алгоритм не сходится для исходной функции.

## Задание 5. Генератор случайных квадратичных функций

По определению квадратичной называется функция  $f(x) = x^T A x$ , где  $A$  – симметричная матрица. Функция имеет минимум, если квадратичная форма является положительно определенной.

Число обусловленности диагональной матрицы  $D$  может быть подсчитано по формуле  $\text{cond}(D) = \lambda_{\max}/\lambda_{\min}$ . По теореме о спектральном разложении матрицы, симметричная матрица  $A$  может быть представлено в виде  $A = Q D Q^T$ , где  $Q$  – ортогональная,  $D$  – диагональная. Поскольку домножение на ортогональную матрицу не меняет норму, то оно и не меняет числа обусловленности. Тогда матрица  $D = \text{diag}(1, \lambda_2, \dots, \lambda_{n-1}, k), i < j \Rightarrow \lambda_i < \lambda_j$  имеет  $\text{cond}(D) = k$ . Тогда легко получить случайную симметричную матрицу – достаточно сгенерировать  $Q$ . Для простоты ее генерации можно найти  $QR$ -разложение случайной матрицы.

Ясно, как считать градиент квадратичной формы:

$$\nabla f(x) = 2Ax$$

На основе приведенного алгоритма была реализована [функция](#) `generate_function`, принимающая два аргумента:

- $n$  – число аргументов;
- $k$  – число обусловленности функции

и возвращающая пару из лямбды, принимающей вектор длины  $n$  и возвращающей результат вызова функции в типе `float`, и лямбды, вычисляющей градиент в точке по формуле выше.



### Задания 6-7. Исследование зависимости числа итераций $T(n, k)$

Введем функцию  $T(n, k)$  – среднее число итераций, необходимых градиентному спуску для сходимости случайной функции из точки  $[10, 10, \dots, 10]$  в зависимости от числа переменных  $n$  и числа обусловленности  $k$ .

Функции `gradient_descent_*` возвращают пару из точки и числа шагов до сходимости. Для наглядности и информативности, будем строить графики  $T(n, k)$  для фиксированных  $n$ . Функцию для вычисления  $T$  можно найти по [ссылке](#).

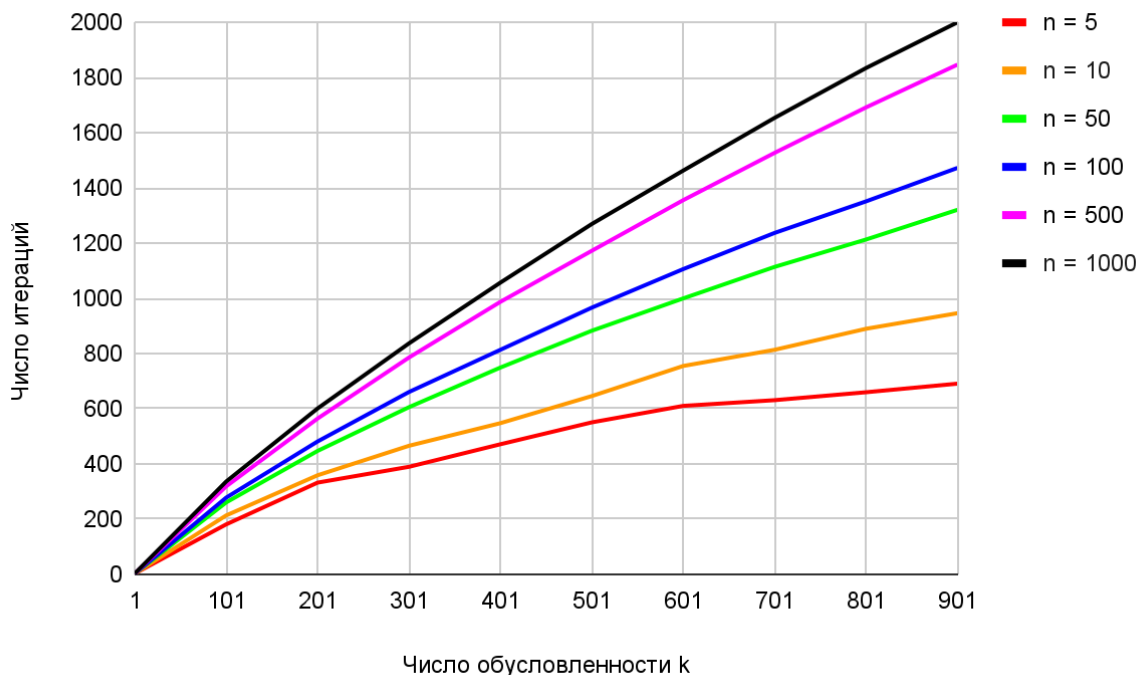


Рис. 19 — График зависимости  $T(n, k)$

Как несложно заметить из графика,  $T(n, k)$  растет при увеличении  $k$ . Это объясняется тем, что плохо обусловленные функции медленнее сходятся к минимуму. Также заметен рост по  $n$  – это связано с тем, что увеличивается норма начальной точки, т.е. увеличивается расстояние от исходной точки до точки минимума, находящейся в нуле.

### Дополнительное задание

Градиентный спуск с линейным поиском в условиях Вольфе, использующий библиотечную функцию *line\_search*.

Эта функция выполняет поиск  $\alpha$  в сильных условиях Вольфе:

$$\begin{cases} f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k \\ |\nabla f(x_k + \alpha_k p_k)^T p_k| \leq c_2 |\nabla f_k^T p_k| \end{cases}$$

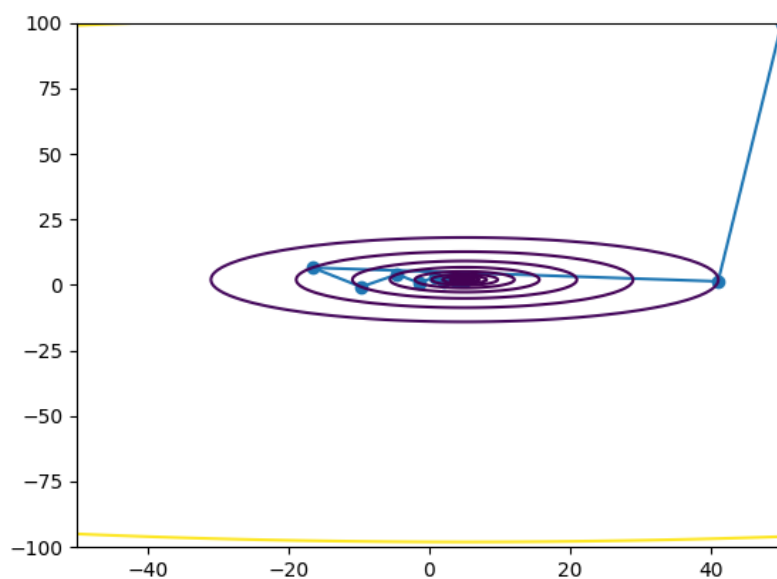
Константы  $0 < c_1 < c_2 < 1$  подбираются таким образом. При этом  $c_1$  выбирается в окрестности 0, а  $c_2$  – в окрестности 1, это означает, что проекция градиента в новом приближении должна либо изменить направление, либо уменьшится по модулю. Однако наша функция может не удовлетворять усиленным условиям Вольфе, поэтому в этом случае находится  $\alpha$ , через деление  $\alpha$  на 2 и взятием такого, что выполняются слабые условия Вольфе:

$$\begin{cases} f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k \\ \nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k \end{cases}$$

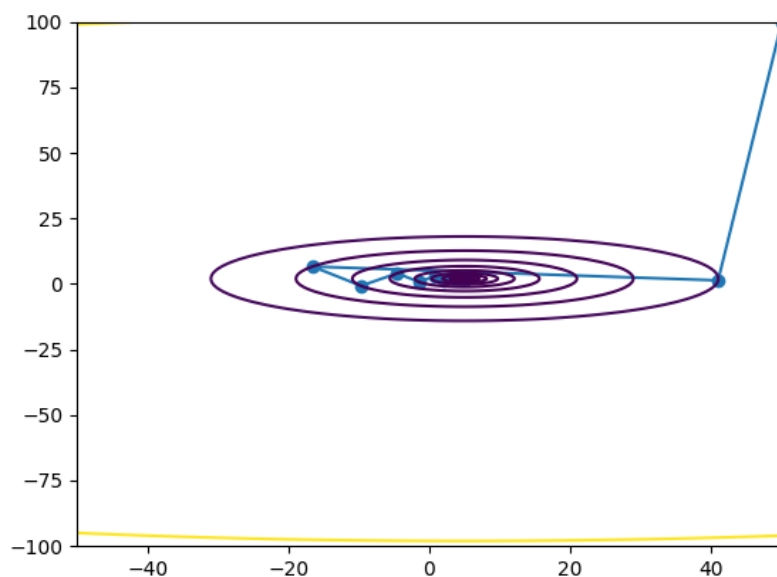
Теперь проведем исследование метода линейного поиска, основанного на условии Вольфе, на рассмотренных ранее примерах. Замечено наблюдение, что такой градиентный спуск считает большее число градиентов, нежели чем обычный градиентный спуск, поэтому мы можем использовать функцию, использующее меньшее число точек для вычисления градиента.

Если сравнивать с другими методами градиентного спуска, то можно пронаблюдать увеличение количества вычисления градиентов в 2-3 раза, при этом функцию мы вычисляем гораздо меньшее число раз (около 4 раз), что является хорошим результатом.

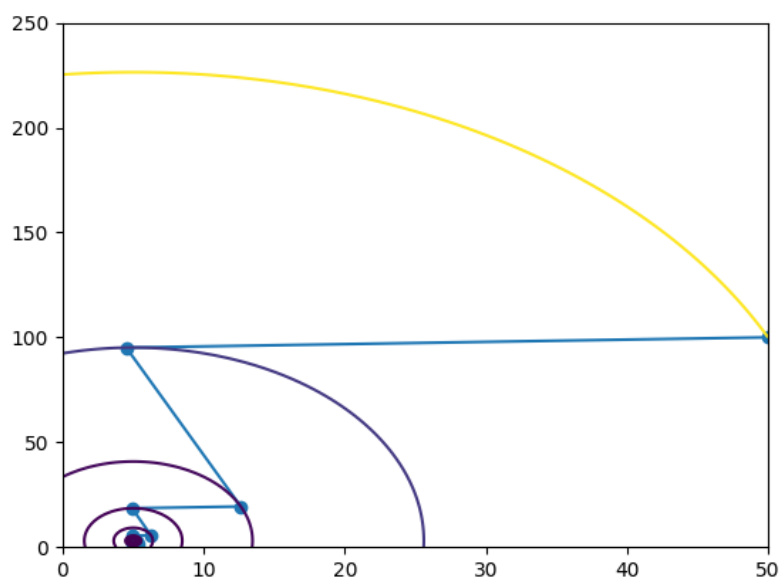
Реализацию градиентного спуска с учетом условий Вольфе можно найти по [ссылке](#).



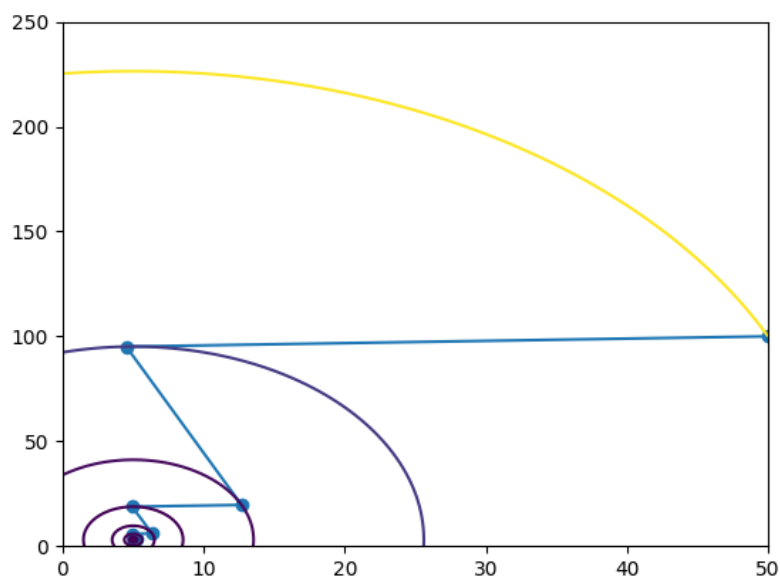
**Рис. 20** — градиентный спуск на 1 примере из точки (50, 100), с использованием взятие градиента через взятие производной по 2 формуле, число вычислений градиента – 60, число вычислений функции – 180.



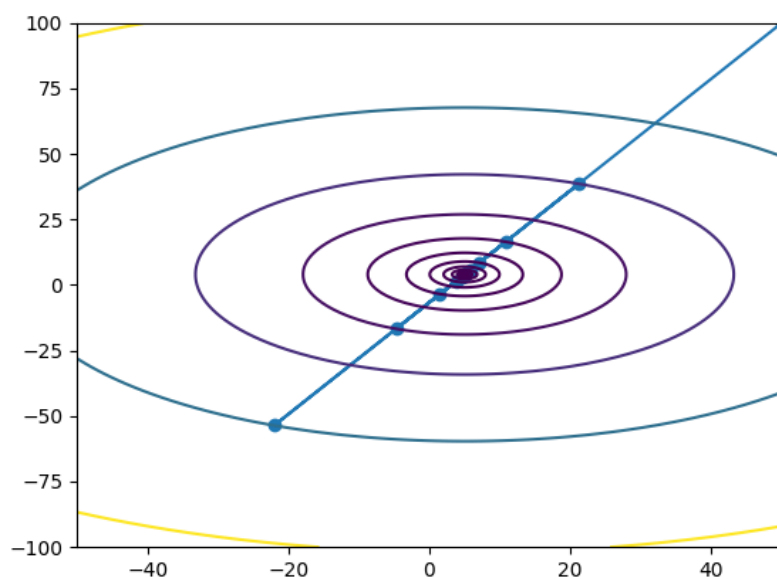
**Рис. 21** — градиентный спуск на 1 примере из точки (50, 100), с использованием взятие градиента через взятие производной по 1 формуле, число вычислений градиента – 67, число вычислений функции – 222.



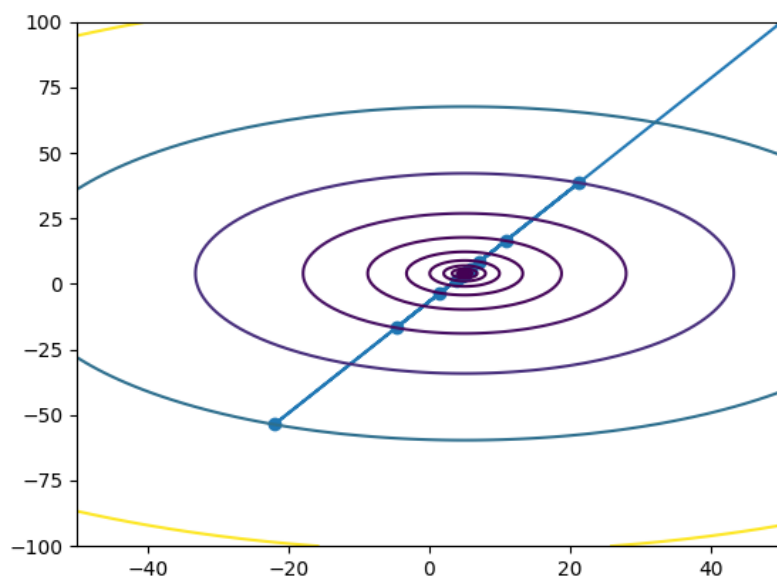
**Рис. 22** — градиентный спуск на 2 примере из точки (50, 100), с использованием взятие градиента через взятие производной по 2 формуле, число вычислений градиента – 38, число вычислений функции – 114.



**Рис. 23** — градиентный спуск на 2 примере из точки (50, 100), с использованием взятие градиента через взятие производной по 1 формуле, число вычислений градиента – 39, число вычислений функции – 129.



**Рис. 24** – градиентный спуск на 3 примере из точки (50, 100), с использованием взятие градиента через взятие производной по 2 формуле, число вычислений градиента – 54, число вычислений функции – 162.



**Рис. 25** – градиентный спуск на 3 примере из точки (50, 100), с использованием взятие градиента через взятие производной по 1 формуле, число вычислений градиента – 60, число вычислений функции – 199.

## **Выводы по эффективности**

Градиентный спуск с постоянным шагом имеет ряд недостатков: 1) он может не сходиться, 2) зачастую функции имеет большое число обусловленности, из-за чего сходимость может оказаться медленной. Но при этом у него есть одно главное преимущество: если начальная точка находится рядом с точкой минимума, то сходимость оказывается быстрой и не приходится пересчитывать значения функции и градиента по несколько раз, что в общем случае может оказаться дорогой операцией.

При этом градиентный спуск, реализованный на линейном поиске с учетом условий Вольфе, показал свою эффективность с точки зрения итоговых вычислений значений функции, поэтому данный метод можно применять наравне с методом дихотомии.

Проанализировав график зависимости числа итераций от числа обусловленности и размерности квадратичной функции, можно прийти к выводу, что зависимость монотонна по обоим аргументам, причем размерность оказывает меньшее влияние, нежели чем число обусловленности.