

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРОГРАММИРОВАНИЯ  
ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

---

## МЕТОДЫ ОПТИМИЗАЦИИ

---

### ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 1

Черкашин Николай, Медведев Иван, Леонтьев Тарас  
Преподаватель: Шохов М.Е

# 1. Градиентный спуск с постоянным шагом

---

## 1.1 Идея

Основная идея метода – на каждом шаге вычислять направление  $p_k$  и двигаться в его направлении с константным шагом, где направление  $p_k$  – это антиградиент  $-\nabla f$ ,

$k + 1$  точка находится из  $k$ -ой по формуле:

$$x^{k+1} = x^k - \alpha \cdot \nabla f(x^k)$$

$\alpha$  – коэффициент обучения.

Ясно, что не любая  $\alpha$  нам гарантирует сходимость градиентного спуска.

## 1.2 Реализация

Данная реализация градиентного спуска останавливается, когда изменение  $x^{k+1}$  и  $x^k$  становится несущественным.

Критерий остановки:

$$|x^{k+1} - x^k| < \varepsilon$$

Мы реализовали две функции:

1. `def gradient_descent(function, start, alpha, eps, points)`
2. `def gradient(vector).`

`gradient_descent` – реализация градиентного спуска.

- принимает функцию – `f`,
- начальную точку – `start`,
- коэффициент обучения – `learn_rate`,
- эпсилон – `eps`,
- `points` – некий массив, заполняемый точками  $x^k$ , он нужен для вывода изображения.

`gradient` – функция, которая находит приближенное значение градиента в точке.

- принимает `vector` – точка в пространстве

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$

### 1.3 Результаты

Пример нашего градиентного спуска с постоянным шагом для функции

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

Аналитически видно, что  $\arg \min f(x, y) = (0, 0)$ . Применив алгоритм, мы получили результат, близкий к истине; в данном примере мы начинаем в точке  $x^0 = (2.2, 2.2)$ ,  $\varepsilon = 10^{-6}$

iteration	point
0	(2.2, 2.2)
1	(1.759, 1.759)
2	(1.126, 1.126)
$\vdots$	$\vdots$
58	$(4.76 \cdot 10^{-6}, 4.76 \cdot 10^{-6})$
59	$(3.71 \cdot 10^{-6}, 3.71 \cdot 10^{-6})$

Таблица 1

Из таблицы 1 видно, что конечное значение близко к локальному минимуму.

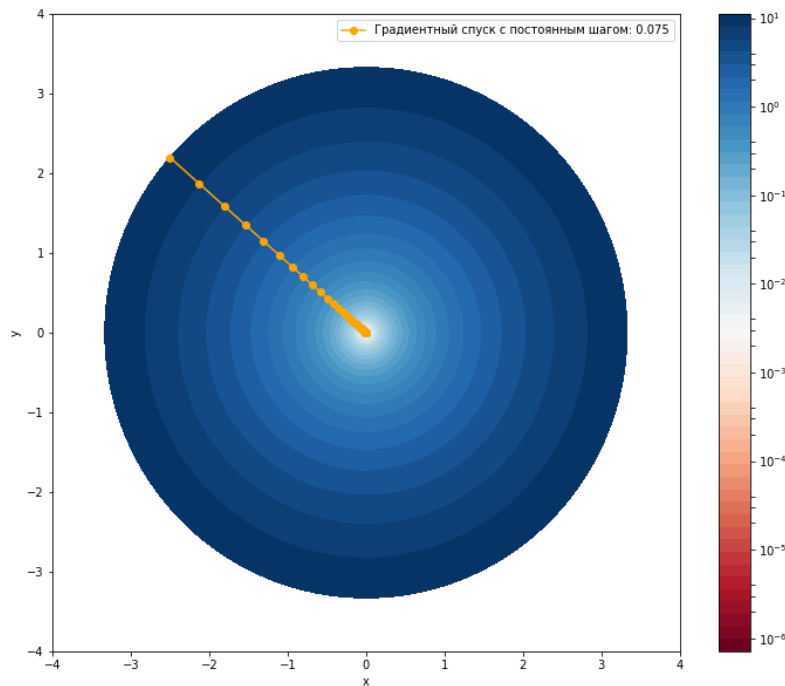


Figure 1: График  $f(x, y)$  с иллюстрацией работы градиентного спуска

На изображении 1 оранжевые точки – это  $x^k$ , для наглядности мы их соединили прямыми, чтобы хорошо было видно, как именно мы приближаемся к локальному минимуму.

## 2. Градиентный спуск на основе линейного поиска

---

### 2.1 Идея

В отличие от градиентного спуска с постоянным шагом, мы будем выбирать  $\alpha_k$  таким, что  $\phi_k(\alpha) = f(x^k - \alpha f'(x^k))$  - минимальна.

$$\alpha_k = \arg \min_{\alpha \geq 0} (f(x^k - \alpha f'(x^k)))$$

На каждом шаге решается задача одномерной оптимизации - поиска минимума  $\phi_k(\alpha)$ . Для этого была реализована дихотомия.

### 2.2 Реализация

Аналогично было реализован критерий остановки.

Мы реализовали следующие функции:

1. `def gradient_descent_with_dichotomy(f, start, eps, points)`
2. `def dichotomy(a, b, eps)`

`gradient_descent_with_dichotomy` - реализация градиентного спуска, аналогично градиентному спуску из п.1, только этот не принимает `learn_rate`

`dichotomy` - функция, которая находит приближенное значение, при котором переданная функция минимальна

- принимает `a` и `b` - соответственно левый и правый конец отрезка, на котором ищется минимум
- `eps` - аналогично

### 2.3 Результаты

Пример градиентного спуска, основанного на дихотомии для функции

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

Аналитически видно, что  $\arg \min f(x, y) = (1, 3)$ . Применив алгоритм, мы получили результат, близкий к истине; в данном примере мы начинаем в точке  $x^0 = (3, -2.5)$ ,  $\varepsilon = 10^{-6}$

iteration	point
0	(3, -2.5)
1	(4.40, -0.22)
2	(1.532, 1.538)
$\vdots$	$\vdots$
18	(1.0004, 2.9994)
19	(1.0001, 2.9999)

Таблица 2

Из таблицы 2 видно, что конечное значение близко к локальному минимуму.

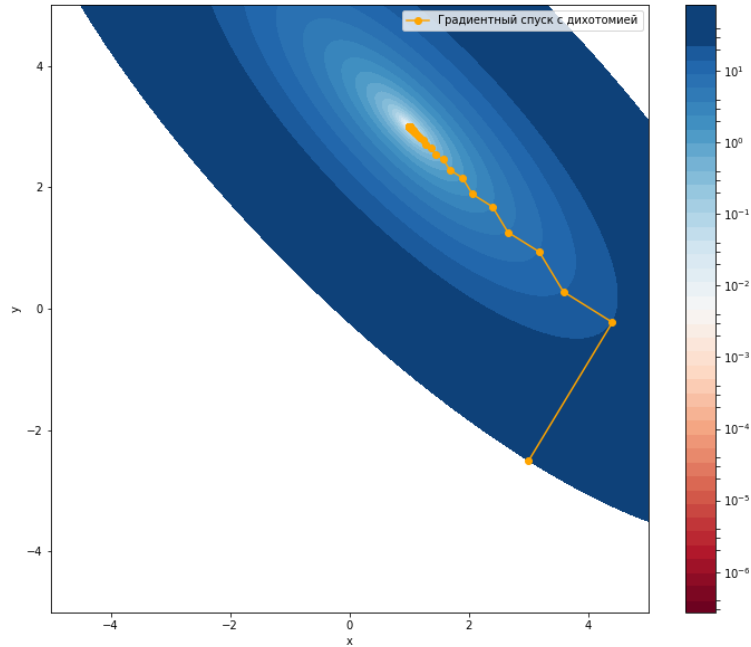


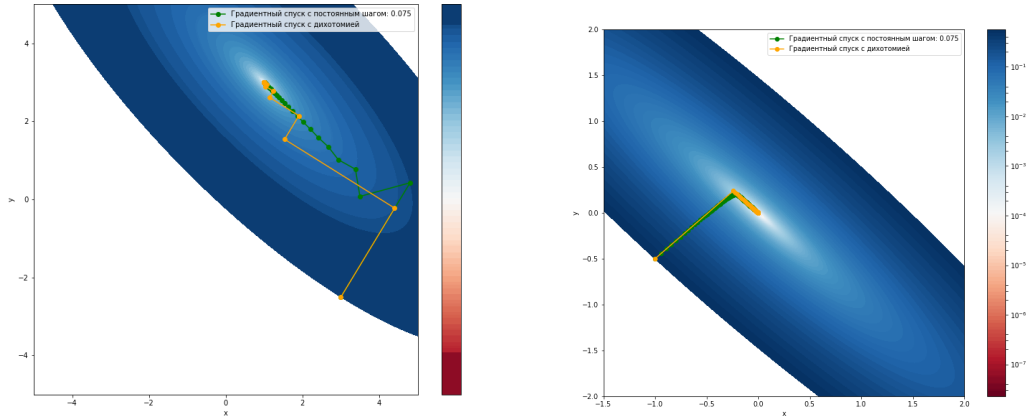
Figure 2: График  $f(x, y)$  с иллюстрацией работы градиентного спуска на основе одномерного поиска

На изображении 2 синие точки – это  $x^k$ , для наглядности мы их соединили прямыми, чтобы хорошо было видно, как именно мы приближаемся к локальному минимуму.

При данном запуске дихотомия находила значения при  $\alpha \in [0, 2]$

### 3. Анализ траекторий методов градиентного спуска

Проанализируем траектории для функций  $f(x, y)$  и  $g(x, y)$



(a)  $f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$

(b)  $g(x, y) = 0.26(x^2 + y^2) + 0.48xy$

Figure 3: Сравнение двух реализаций градиентного спуска

На данных функциях градиентный спуск с постоянным шагом был запущен с  $\alpha = 0.075$  и  $\varepsilon = 10^{-6}$ . Дихотомия была запущена с тем же  $\varepsilon$  и находила значения при  $\alpha \in [0, 2]$

function	$f(x, y)$	$g(x, y)$
const	55	672
dichotomy	20	72

Таблица 3. Количество итераций в зависимости от метода

Из таблицы 3 можно сделать вывод, что градиентный спуск на основе дихотомии на практике работает быстрее, чем с постоянным шагом.

## 4. Исследование методов градиентного спуска в зависимости от параметров

### 4.1 Исследование сходимости градиентного спуска с постоянным шагом

Будем исследовать функции:

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

$$g(x, y) = 0.26(x^2 + y^2) + 0.48xy$$

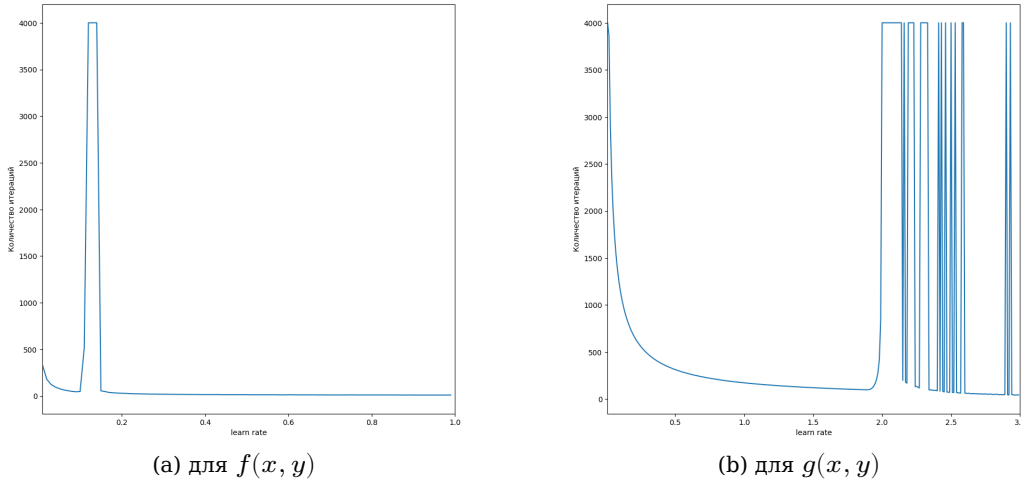


Figure 4: Количество итераций градиентного спуска с постоянным шагом в зависимости от шага

Рисунок 2 показывает нам, что произвольная  $\alpha$  не гарантирует сходимость градиентного спуска, однако можно заметить, что на некотором интервале при увеличении  $\alpha$  уменьшается количество итераций. На графиках пиками со значениями в  $4 \cdot 10^3$  показаны  $\alpha$  при которых градиент спуск расходится.

### 4.2 Исследование эффективности градиентного спуска с использованием одномерного поиска с точки зрения количества вычислений минимизируемой функции и ее градиентов

Рассмотрим две функции  $f(x, y)$  и  $g(x, y)$

$$f(x, y) = x^2 + 16y^2 - xy$$

$$g(x, y) = 0.26(x^2 + y^2) + 0.48xy$$

Посчитаем количество вычислений градиента в процессе выполнения градиентного спуска и количество вычислений одномерной оптимизации в градиентном спуске, основанном на дихотомии.

	спуск с постоянным шагом, градиент	дихотомия, градиент	дихотомия, одномерная оптимизация
$f(x, y)$	80	43	$1.5 \cdot 10^3$
$g(x, y)$	2000	1170	$4.2 \cdot 10^4$

Таблица 4. Сравнение количества вычислений градиента

В двумерном случае преимущество спуска на основе дихотомии несущественно, однако при увеличении размерности преимущество будет расти.

### 4.3 Исследование в зависимости от выбора начальной точки

Исследуем зависимость количества итераций от выбора начальной точки. Пробегам для  $x$  и  $y$  с шагом 0.5 и считаем  $f(x, y) = x^2 + 16y^2 - xy$ .

$y/x$	-2.0	-1.5	-1.0	-0.5	0.0	0.5	1.0	1.5	2.0
-2.0	14	14	17	29	45	15	9	6	7
-1.5	11	14	16	24	43	11	6	7	5
-1.0	9	12	12	18	41	9	5	4	5
-0.5	10	9	9	11	37	5	5	5	5
0.0	10	10	10	10	1	3	5	5	5
0.5	9	9	9	12	26	5	5	5	5
1.0	9	10	11	16	30	9	10	5	5
1.5	12	12	15	19	32	13	7	7	7
2.0	12	13	16	24	34	19	9	7	7

Таблица 5. Дихотомия

$y/x$	-2.0	-1.5	-1.0	-0.5	0.0	0.5	1.0	1.5	2.0
-2.0	74	74	74	74	74	74	74	74	73
-1.5	71	71	71	71	71	71	71	71	71
-1.0	68	68	67	67	67	67	67	67	66
-0.5	62	61	61	61	60	60	60	59	59
0.0	41	38	34	27	1	28	34	38	41
0.5	59	59	60	60	60	61	61	61	62
1.0	66	67	67	67	67	67	67	68	68
1.5	71	71	71	71	71	71	71	71	71
2.0	73	74	74	74	74	74	74	74	74

Таблица 6. Градиентный спуск с постоянным шагом

Можно заметить, что нет явной зависимости между начальной точкой и числом итераций.



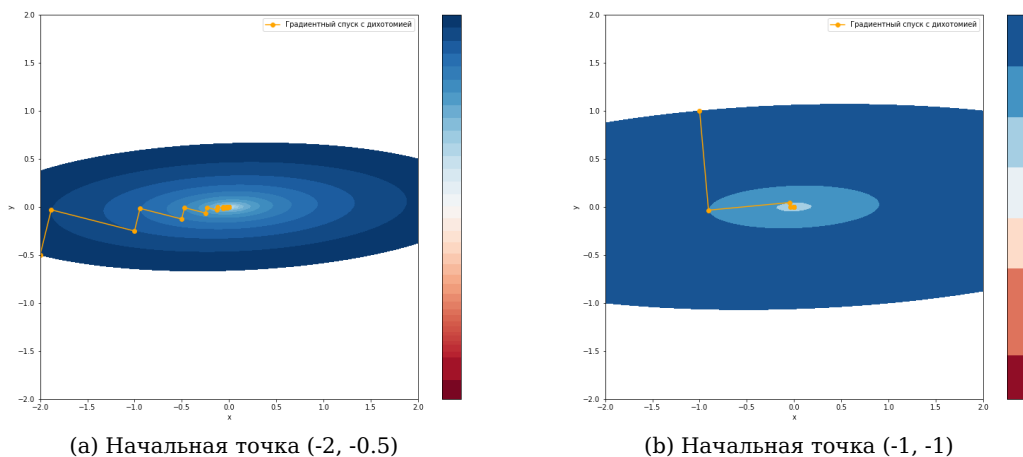


Figure 5: Градиентный спуск на основе Дихотомии

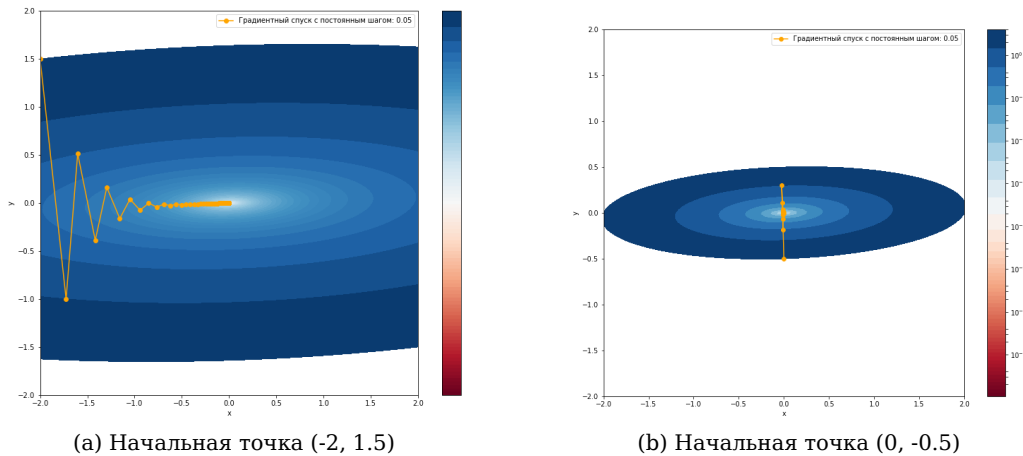


Figure 6: Градиентный спуск с постоянным шагом

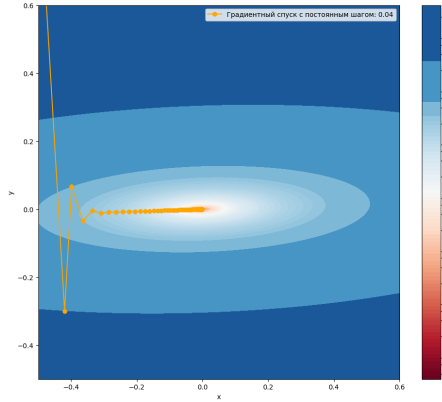
#### 4.4 Исследование влияния нормализации на сходимость на примере масштабирования осей плохо обусловленной функции

Исследуем на примере функции

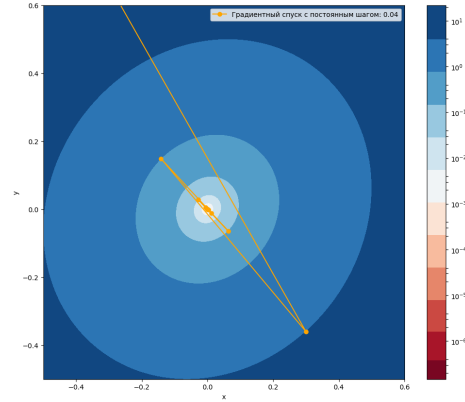
$$f(x, y) = -xy + x^2 + 16y^2$$

$$f_{scaled}(x, y) = -4xy + 16x^2 + 16y^2$$

Нетрудно заметить, что нормализация позволяет уменьшить количество итераций обеих реализаций градиентного спуска. Действительно, нормализация плохо обусловленной функции позволяет выбирать более точное направление спуска, отчего существенно уменьшается количество итераций.

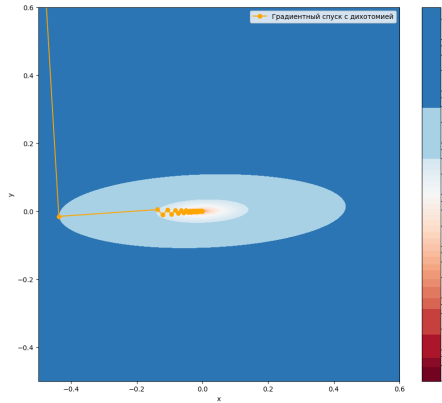


(a) Без нормализации, 74 итерации

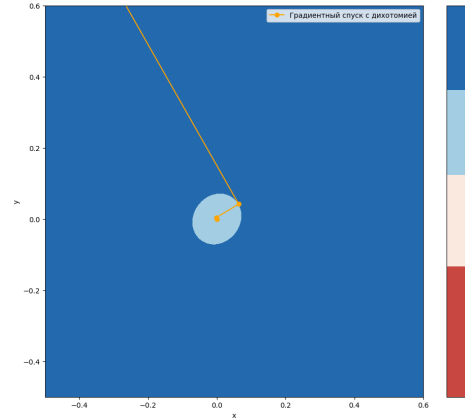


(b) С нормализацией, 13 итераций

Figure 7: Градиентный спуск с постоянным шагом



(a) Без нормализации, 30 итераций



(b) С нормализацией, 5 итераций

Figure 8: Градиентный спуск на основе дихотомии

## 5. Генератор случайных квадратичных функций $n$ переменных с числом обусловленности $k$

Результатом работы является генератор квадратичных симметричных матриц, которые однозначно, с точностью до константы задают квадратичную форму.

Мы создаём случайную диагональную матрицу  $S$ , в которой задаём максимальный и минимальный элемент на диагонали так, что выполняется  $\frac{\max}{\min} = k$ , где  $k$  - данное нам число обусловленности.

Дальше мы пользуемся спектральным разложением матрицы:

Если  $A$  - ортогональная матрица, то  $ASA^{-1} = B$ , где  $B$  - симметричная матрица, у которой собственные числа равны собственным числам  $S$ .

Чтобы перевести в функцию, достаточно сложить все перемножения  $B_{ij}$  на  $v[i] \cdot v[j]$ , где  $v$  это  $n$ -мерный вектор значений.

**6. Зависимость числа итераций  $T(n, k)$ , необходимых градиентному спуску для сходимости в зависимости от размерности пространства  $2 \leq n \leq 10^3$  и числа обусловленности оптимизируемой функции  $1 \leq k \leq 10^3$**

Проверим влияние числа обусловленности на количество итераций.

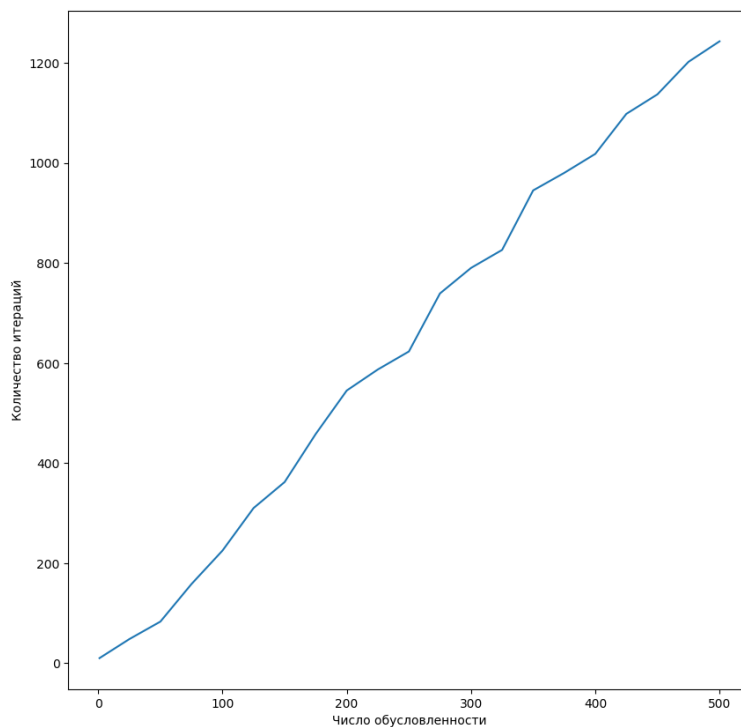


Figure 9

На графике наблюдается линейная зависимость.

Проверим влияние размерности пространства на количество итераций.

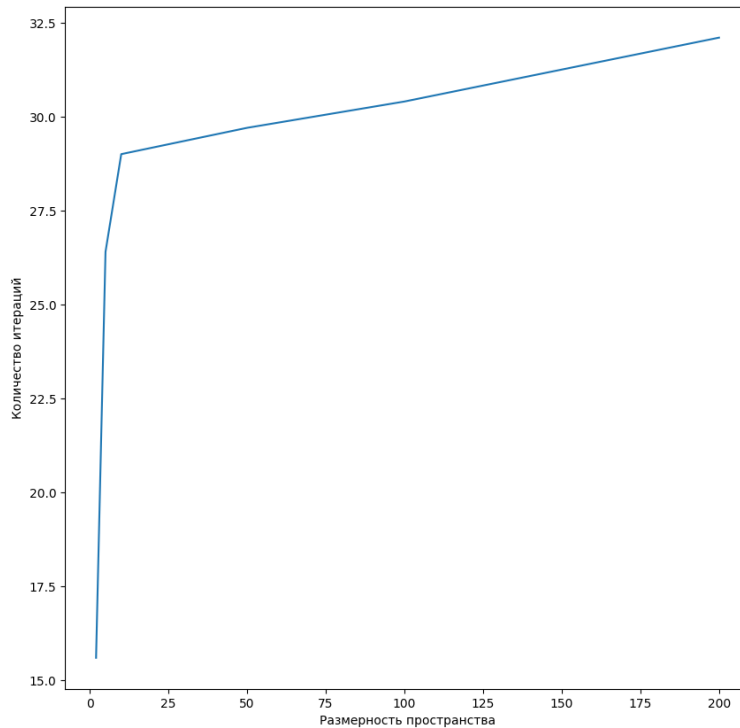


Figure 10

До небольших чисел(порядка 10) виден рост, но дальше число итераций не растёт. Т.е зависимости от размерности нет.

Данные измерения проводились усреднением 100 экспериментов при заданных параметрах.

## Градиентный спуск на основе одномерного поиска с учетом условий Вольфа

### Идея

Для подбора подходящих  $\alpha$  используются (сильные) условия Вольфа:

$$f(x^k + \alpha_k \cdot p_k) \leq f(x^k) + c_1 \cdot \alpha_k \cdot \nabla f_k^T \cdot p_k$$

$$|\nabla f(x^k + \alpha \cdot p_k)^T p_k| \leq c_2 \cdot |\nabla f_k^T \cdot p_k|$$

где  $0 < c_1 < c_2 < 1$

Первое условие накладывает ограничения на  $\alpha$  такие, что интервал в правой части неравенства выше чем то, где мы хотим оказаться. Иначе говоря, функция должна уменьшиться достаточно соответствующе шагу. Второе условие говорит, что градиент в следующей точке должен быть

больше, чем в текущей в некоторое количество раз. Усиленное условие Вольфа разрешает области, в которых градиент меняет знак.

### Реализация

Реализация отличается от реализации градиентного спуска, основанного на дихотомии, реализацией поиска  $\alpha$ , удовлетворяющего усиленным условиям Вольфа: изначально  $\alpha$  увеличивается с целью найти подходящий шаг или интервал, который может быть уменьшен до некоторой приемлемой длины шага без потери выполнения условий Вольфа, который затем и уменьшается.

### Результаты

Рассмотрим функцию:  $f(x, y) = 0.26(x^2 + y^2) + 0.48xy$

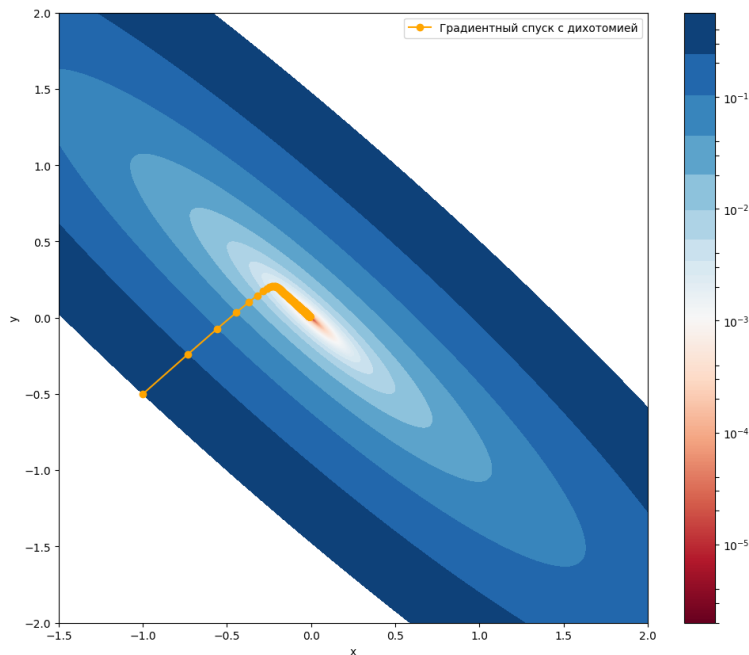


Figure 11: Результат градиентного спуска с дихотомией для  $f$

градиентный спуск	дихотомия	постоянный шаг	условия Вольфа
количество итераций	254	867	79

Таблица 7. Кол-во итераций в зависимости от реализации градиентного спуска  $f$

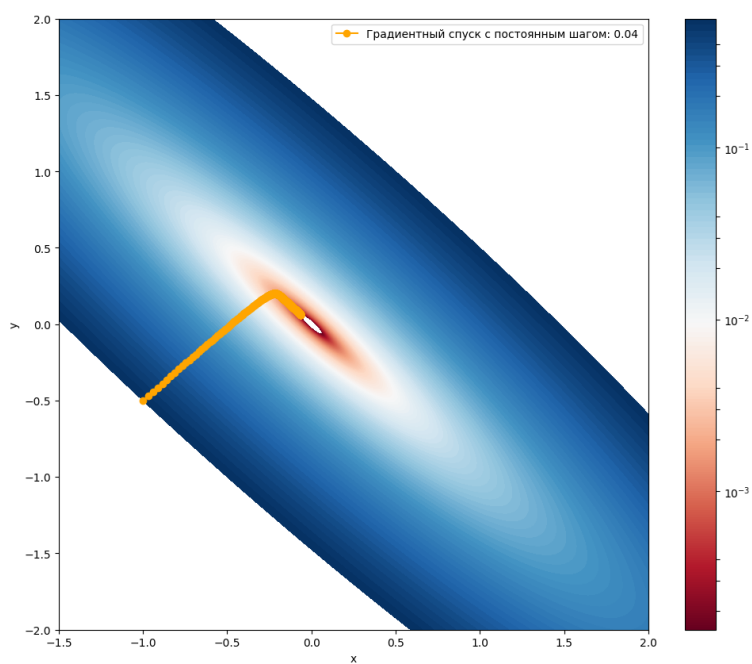


Figure 12: Результат градиентного спуска с постоянным шагом для  $f$

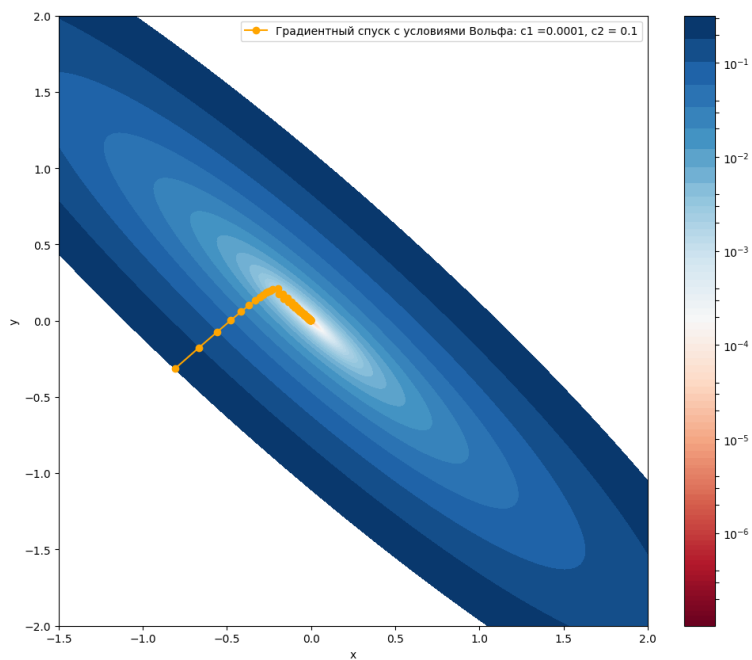


Figure 13: Результат градиентного спуска с условиями Вольфа для  $f$

Рассмотрим функцию:  $g(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$

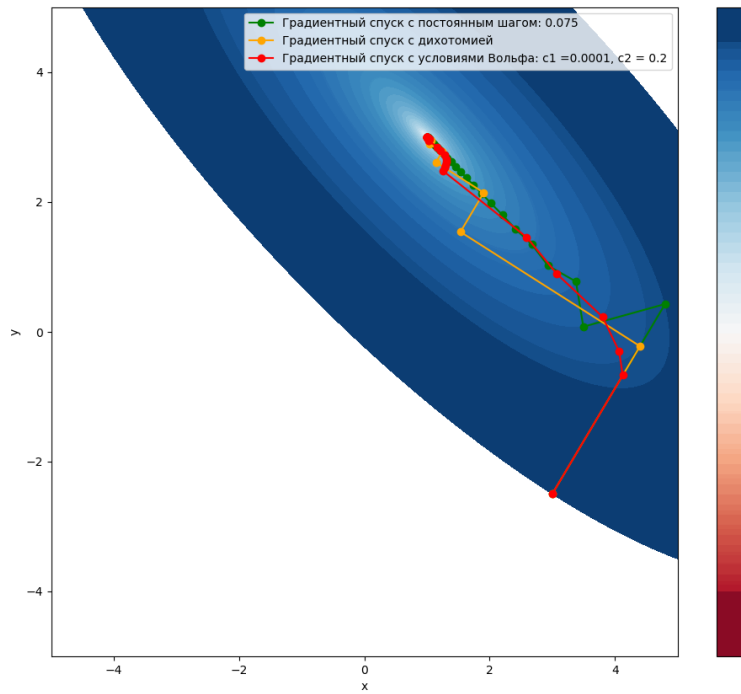


Figure 14: Результат градиентного спуска для  $g$

градиентный спуск	дихотомия	постоянный шаг	условия Вольфа
количество итераций	20	55	36

Таблица 8. Кол-во итераций в зав-ти от реализации спуска  $g$

Можно заметить, что градиентный спуск, основанный на одномерном поиске с учетом условий Вольфа доходит до критерия остановки за меньшее число итераций, чем ранее реализованные версии градиентного спуска.

## Глобальные выводы

- Было исследовано поведение градиентного спуска с постоянным шагом, а также градиентного спуска, основанного на дихотомии на квадратичных функциях.
- Было выявлено более быстрое достижение условия остановки именно вторым методом.
- Также было замечено, что изначально плохие входные условия существенно увеличивают количество итераций (неудачная начальная точка, плохая обусловленность).
- Был исследован метод упрощения вычислений (scaling), помогающий находить локальный минимум за меньшее количество итераций.
- Также был рассмотрен метод на условиях Вольфа, развивающий метод градиентного спуска на дихотомии, он показал наиболее

быстрое достижение условия остановки, но также были найдены случаи, когда дихотомия оказалась эффективнее.