Методы оптимизации.
Отчет по лабораторной работе 1.
Новицкий Илья, Остриченко Илья, Петров Георгий.
M3234

19.03.2024

Ссылка на GitHub : `https://github.com/Saintghetto17/Optimization_Lab_1`

Постановка задачи: найти минимальное значение функции $f : \mathbb{R}^n \to \mathbb{R}$, используя методы 0-1 порядков.
В нашем случае пришлось работать с квадратичными функциями при $n = 2$. В качестве исследуемых функций мы взяли:

(1) $f_1(x, y) = x^2 + (2x - 4y)^2 + (x - 5)^2$. Точка минимума $(\frac{5}{2}, \frac{5}{4}, \frac{25}{2})$,

(2) $f_2(x, y) = x^2 + y^2 - xy + 2x - 4y + 3$. Точка минимума $(2, 0, -1)$,

(3) Bonus: $f_3(x, y) = 100x^2 + y^2$. Точка минимума $(0, 0, 0)$.

Каждую из функций мы протестировали на двух методах градиентного спуска, а именно: методе с постоянным шагом и методе одномерного поиска. А также на методе Нельдера-Мида, который заключается в том, что поиск минимума происходит засчет отражений, сжатий и растяжений симплекса. Каждый из методов имеет свои собственные преимущества. Например, градиентные методы являются представителями метода 1 порядка, это значит алгоритм поиска использует производную первого порядка исследуемой функции, в то время как Нельдер-Мид является методом 0 порядка.

Nota bene : мы обрезали часть таблицы (последние две колонки, так как они не сильно влияли на результат. Их смысл заключался в критерии остановки метода)

1. Градиентные методы. Постоянный шаг. Реализация алгоритма градиентного спуска на наших функиях работает ожидаемо корректно. Рассмотрим таблицы, в которых отражена работа метода с постоянным шагом.
В качестве возможных постоянных шагов мы взяли 0.1, 0.001. Заметно, что выбор шага очень влияет на конечный результат. Так, в 0-3-6 экспериментах для f_1 минимальное значение так и не достигается, а для бонусной плохо-обусловленной функции 21-24 выявляют тот же результат.
С чем это связано? Как известно, следующий шаг градиентного метода вычисляется по формуле $x_{k+1} = x_k - \alpha \cdot grad(x_k)$. При этом, может случиться такое, что прыжки по вектору градиента устремят точку "вникуда"или просто зациклят. Обычно такое происходит, когда выбранный

коэффициент недостаточно мал. Заметно, что, для f_1 количество итераций наиболее мало, по сравнению с двумя оставшимися. Эта характеристика ни в коем случае не означает, что алгоритм работает плохо для оставшихся функций, так как помимо самого алгоритма важен и выбор начальной точки. В случае с f_1 нам можно сказать повезло. Из таблицы видно, что метод достаточно точно находит минимальное значение. Различие буквально порядка 10^{-4}.

LEARNING RATE

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	LEARNING_RATE	VALUE
0	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(0, 0)	None	0.001	0.1	None
1	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(0, 0)	1794	1e-06	0.001	12.50015799807337
2	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(0, 0)	2161	1e-05	0.001	12.500015800782446
3	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(1, 1)	None	0.001	0.1	None
4	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(1, 1)	1581	1e-06	0.001	12.500158342460495
5	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(1, 1)	1948	1e-05	0.001	12.500015835223284
6	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(2, 2)	None	0.001	0.1	None
7	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(2, 2)	635	1e-06	0.001	12.50015851173671
8	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(2, 2)	1002	1e-05	0.001	12.50001585215195

Рис. 1: Таблица для f_1

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	LEARNING_RATE	VALUE	BY_DOT
9	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(0, 0)	26	0.001	0.1	-0.9958254156446342	False
10	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(0, 0)	3800	1e-06	0.001	-0.9995014475739183	False
11	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(0, 0)	4951	1e-05	0.001	-0.9999501730370577	True
12	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(1, 1)	12	0.001	0.1	-0.9994252563058583	False
13	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(1, 1)	1632	1e-06	0.001	-0.9998347589355459	False
14	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(1, 1)	2015	1e-05	0.001	-0.9999834571911825	True
15	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(2, 2)	26	0.001	0.1	-0.9958254156446342	False
16	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(2, 2)	3800	1e-06	0.001	-0.9995014475739179	False
17	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(2, 2)	4951	1e-05	0.001	-0.9999501730370586	True

Рис. 2: Таблица для f_2

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	LEARNING_RATE	VALUE	BY_DOT	BY_FUNC
18	100*x**2 + y**2	0	(0, 0)	2	0.001	0.1	0.0	False	True
19	100*x**2 + y**2	0	(0, 0)	2	1e-06	0.001	0.0	False	True
20	100*x**2 + y**2	0	(0, 0)	1	1e-05	0.001	0.0	True	False
21	100*x**2 + y**2	0	(1, 1)	None	0.001	0.1	None	False	True
22	100*x**2 + y**2	0	(1, 1)	2073	1e-06	0.001	0.00024844151987674437	False	True
23	100*x**2 + y**2	0	(1, 1)	2648	1e-05	0.001	2.485115956624835e-05	True	False
24	100*x**2 + y**2	0	(2, 2)	None	0.001	0.1	None	False	True
25	100*x**2 + y**2	0	(2, 2)	2419	1e-06	0.001	0.0002486673347931632	False	True
26	100*x**2 + y**2	0	(2, 2)	2994	1e-05	0.001	2.4873747427259503e-05	True	False

Рис. 3: Таблица для f_3

Тем не менее такой метод достаточно не конструктивен. Сразу возникает вопрос, а как мы выбрали шаг, который выдаст корректный результат, как мы его подобрали, и что, если для разных функций шаг может варьироваться? Именно поэтому и существует метод градиентного спуска, основанный на одномерном спуске, или как его еще называют "метод наискорейшего спуска".

1. Градиентные методы. Метод наискорейшего спуска..

В чем заключается метод? Будем конструктивно искать такое α, при котором $f(x_k - \alpha \cdot grad(x_k))$ принимает минимальное значение по α. Заметим, что теперь наша функция является функцией одного аргумента, а также является унимодальной (без док-ва). На помощь приходят методы тернарного спуска и дихотомии (бонус 1). Оба метода сужают область поиска на какую-то фиксированную константу. Такой подход поиска шага более универсален, а засчет правильного подбора мы уменьшим количество итераций алгоритма.

CHANGING STEP TERNARY

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CHANGING STEP	VALUE
0	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(0, 0)	22	0.001	TERNARY RATE METHOD	12.501670476617594
1	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(0, 0)	39	1e-06	TERNARY RATE METHOD	12.50000169600781
2	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(0, 0)	33	1e-05	TERNARY RATE METHOD	12.50001931524061
3	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(1, 1)	9	0.001	TERNARY RATE METHOD	12.500311332327318
4	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(1, 1)	15	1e-06	TERNARY RATE METHOD	12.500000342775959
5	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(1, 1)	13	1e-05	TERNARY RATE METHOD	12.500003319695804
6	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(2, 2)	3	0.001	TERNARY RATE METHOD	12.500000001360888
7	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(2, 2)	4	1e-06	TERNARY RATE METHOD	12.5000000000006
8	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(2, 2)	3	1e-05	TERNARY RATE METHOD	12.500000001360888

Рис. 4: Таблица для f_1. Тернарный поиск

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CHANGING STEP	VALUE
9	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(0, 0)	5	0.001	TERNARY RATE METHOD	-0.9999435223647546
10	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(0, 0)	8	1e-06	TERNARY RATE METHOD	-0.9999999305241438
11	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(0, 0)	7	1e-05	TERNARY RATE METHOD	-0.9999993516566339
12	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(1, 1)	2	0.001	TERNARY RATE METHOD	-1.0
13	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(1, 1)	2	1e-06	TERNARY RATE METHOD	-1.0
14	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(1, 1)	2	1e-05	TERNARY RATE METHOD	-1.0
15	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(2, 2)	5	0.001	TERNARY RATE METHOD	-0.999943522364755
16	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(2, 2)	8	1e-06	TERNARY RATE METHOD	-0.9999999305269531
17	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(2, 2)	7	1e-05	TERNARY RATE METHOD	-0.9999993516716925

Рис. 5: Таблица для f_2. Тернарный поиск

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CHANGING STEP	VALUE
18	100*x**2 + y**2	0	(0, 0)	2	0.001	TERNARY RATE METHOD	0.0
19	100*x**2 + y**2	0	(0, 0)	2	1e-06	TERNARY RATE METHOD	0.0
20	100*x**2 + y**2	0	(0, 0)	2	1e-05	TERNARY RATE METHOD	0.0
21	100*x**2 + y**2	0	(1, 1)	4	0.001	TERNARY RATE METHOD	5.794217434150271e-07
22	100*x**2 + y**2	0	(1, 1)	5	1e-06	TERNARY RATE METHOD	5.62269292366592e-09
23	100*x**2 + y**2	0	(1, 1)	5	1e-05	TERNARY RATE METHOD	5.62269292366592e-09
24	100*x**2 + y**2	0	(2, 2)	4	0.001	TERNARY RATE METHOD	2.3176869736601084e-06
25	100*x**2 + y**2	0	(2, 2)	6	1e-06	TERNARY RATE METHOD	1.7554621463648612e-10
26	100*x**2 + y**2	0	(2, 2)	5	1e-05	TERNARY RATE METHOD	2.249077169466368e-08

Рис. 6: Таблица для f_3. Тернарный поиск

BONUS: DICHOTOMY

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CHANGING STEP	VALUE
0	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(0, 0)	22	0.001	DICHOTOMY RATE METHOD	12.50167053528769
1	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(0, 0)	39	1e-06	DICHOTOMY RATE METHOD	12.500001695400137
2	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(0, 0)	33	1e-05	DICHOTOMY RATE METHOD	12.500019312219717
3	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(1, 1)	9	0.001	DICHOTOMY RATE METHOD	12.500311790951944
4	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(1, 1)	15	1e-06	DICHOTOMY RATE METHOD	12.500000344191495
5	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(1, 1)	13	1e-05	DICHOTOMY RATE METHOD	12.500003330441187
6	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(2, 2)	3	0.001	DICHOTOMY RATE METHOD	12.500000000691724
7	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(2, 2)	4	1e-06	DICHOTOMY RATE METHOD	12.500000000001204
8	x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(2, 2)	3	1e-05	DICHOTOMY RATE METHOD	12.500000000691724

Рис. 7: Таблица для f_1. Бонус 1 - дихотомия

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CHANGING STEP	VALUE
9	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(0, 0)	5	0.001	DICHOTOMY RATE METHOD	-0.9999435213903207
10	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(0, 0)	8	1e-06	DICHOTOMY RATE METHOD	-0.9999999305405627
11	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(0, 0)	7	1e-05	DICHOTOMY RATE METHOD	-0.9999993516405161
12	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(1, 1)	2	0.001	DICHOTOMY RATE METHOD	-0.999999999997911
13	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(1, 1)	2	1e-06	DICHOTOMY RATE METHOD	-0.999999999997911
14	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(1, 1)	2	1e-05	DICHOTOMY RATE METHOD	-0.999999999997911
15	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(2, 2)	5	0.001	DICHOTOMY RATE METHOD	-0.9999435213903212
16	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(2, 2)	8	1e-06	DICHOTOMY RATE METHOD	-0.9999999305302536
17	x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(2, 2)	7	1e-05	DICHOTOMY RATE METHOD	-0.9999993516537664

Рис. 8: Таблица для f_2. Бонус 1 - дихотомия

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CHANGING STEP	VALUE
18	100*x**2 + y**2	0	(0, 0)	2	0.001	DICHOTOMY RATE METHOD	0.0
19	100*x**2 + y**2	0	(0, 0)	2	1e-06	DICHOTOMY RATE METHOD	0.0
20	100*x**2 + y**2	0	(0, 0)	2	1e-05	DICHOTOMY RATE METHOD	0.0
21	100*x**2 + y**2	0	(1, 1)	4	0.001	DICHOTOMY RATE METHOD	1.1883818283144956e-06
22	100*x**2 + y**2	0	(1, 1)	6	1e-06	DICHOTOMY RATE METHOD	1.2890607353081426e-10
23	100*x**2 + y**2	0	(1, 1)	5	1e-05	DICHOTOMY RATE METHOD	1.153348946446996e-08
24	100*x**2 + y**2	0	(2, 2)	4	0.001	DICHOTOMY RATE METHOD	4.753527313257982e-06
25	100*x**2 + y**2	0	(2, 2)	6	1e-06	DICHOTOMY RATE METHOD	5.15624294123257e-10
26	100*x**2 + y**2	0	(2, 2)	5	1e-05	DICHOTOMY RATE METHOD	4.613395785787984e-08

Рис. 9: Таблица для f_3. Бонус 1 - дихотомия

Из таблицы видно, что количество итераций уменьшилось аж в 100 раз. И это неудивительно, потому что каждый раз мы выбираем лучший шаг для данной точки и функции. Конечно, внутри мы крутимся внутри функции поиска, но так или иначе, количество итераций уменьшается, а точноссть значений увеличивается до 10^{-8} порядка. Подряд идут фотографии таблиц для тернарного поиска и дихотомии, так как идейно они не сильно отличаются.

Нельдер-Мид.

Как уже было сказано ранее, Нельдер Мид принципиально другой метод с точки зрения идеи. Основная задача все такая же - поиск минимального значения функции. Но данный метод является методом 0 порядка, а значит нам не нужно брать производную функции. Иногда этот метод довольно удобен если мы хотим исследовать не гладкую функцию. Метод устроен примерно так: берем в n-мерном пространстве симплекс размера n+1, сортируем значения функции в точках симплекса, берем две точки, в которых достигаются максимальные значения функции соответственно, а также точку, где достигается минимальное значение. Считаем центр тяжести n+1 точки и в зависимости от некоторых условий отражаем, сжимаем или растягиваем наш симплекс. Но есть и минусы у этого метода. При выборе "плохих"точек симплекса мы можем убежать дальше допустимой окрестности поиска минимума. Рассмотрим таблицы, относящиеся к Нельдеру-Миду.

NELDER-MEAD

FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	VALUE
x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(0, 0)	65	12.500000001737911
x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(1, 1)	43	12.500000004392836
x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(2, 2)	34	12.500000002056694

FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	VALUE
x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(0, 0)	68	-0.9999999991925863
x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(1, 1)	37	-0.9999999984274837
x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(2, 2)	45	-0.9999999992480335

FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	VALUE
100*x**2 + y**2	0	(0, 0)	7	0.0
100*x**2 + y**2	0	(1, 1)	49	2.945793253797094e-09
100*x**2 + y**2	0	(2, 2)	51	4.238519938715909e-09

Рис. 10: Общая таблица по Нельдеру-Миду

Заметно, что библиотичная реализация этой функции в SciPy достаточно эффективна и точна. Количество итераций чуть больше, чем количество итераций в спуске с одномерным поиском, но значения точны до 10^{-8} порядка. Причем видно, что метод достаточно эффективен и для плохо-обусловленной функции.

Функция n аргументов. Бонус 2.1

Мы считаем, что данный бонус ничем не отличается от рассмотренных примеров. Идейно в код нужно добавить точки произвольной размерности и посчитать значения функции с учетом размерности. Код был написан изначально для функции двух аргументов, а принципиально что-то менять для повышения размерности - долго и не представляет научного интереса.

Плохо обусловленная функция. Бонус 2.2

В обоих методах для f_3 результат в 23 и 26 экспериментах(для постоянного шага) и 21 - 26 экспериментах (для наискорейшего спуска) сильно отличается от желаемого. Дело в том, что данная функция является плохо обусловленной, а именно : при малом изменении нормы аргумента, значения функции меняются сильно. Это затрудняет выдачу точных значений для градиентных методов. В таком случае лучше использовать Нельдера-Мида.

Зашумленная функция и Мультимодальная функция. Бонус 2.3

В чем смысл зашумленной функции? Смысл в том, что "поставщик"значений может прислать значения функции с немного некорректными значениями. В общем виде это выглядит так

$f \to f + random(0, EPS)$.
При этом поставщик значений ожидает, что минимальное значение при зашумлении не сильно будет отличаться от минимального значения оригинальной функции. Так вышло, что зашумление достаточно сильно влияет на поиск минимального значения. Это немудрено, так как сама функция становится недифференцируемой и неявной. Рассмотрим таблицы для постоянного шага и для изменяемого.

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	LEARNING_RATE	VALUE	BY_DOT	BY_FUNC
36	noisy	12.5	(0, 0)	3	0.001	0.1	24.0	False	True
37	noisy	12.5	(0, 0)	5	1e-06	0.001	24.0	False	True
38	noisy	12.5	(0, 0)	1	1e-05	0.001	26.0	True	False
39	noisy	12.5	(1, 1)	12123	0.001	0.1	22.546888316483496	False	True
40	noisy	12.5	(1, 1)	157	1e-06	0.001	19.94293451427555	False	True
41	noisy	12.5	(1, 1)	107	1e-05	0.001	20.943132619939362	True	False
42	noisy	12.5	(2, 2)	1856	0.001	0.1	21.544740162433932	False	True
43	noisy	12.5	(2, 2)	192	1e-06	0.001	27.646259447617638	False	True
44	noisy	12.5	(2, 2)	101	1e-05	0.001	27.64799574623453	True	False

Рис. 11: Зашумленная функция. Постоянный шаг. Бонус 2.3

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CHANGING STEP	VALUE
36	noisy	12.5	(0, 0)	3	0.001	TERNARY RATE METHOD	25.0
37	noisy	12.5	(0, 0)	6	1e-06	TERNARY RATE METHOD	26.0
38	noisy	12.5	(0, 0)	2	1e-05	TERNARY RATE METHOD	24.0
39	noisy	12.5	(1, 1)	31	0.001	TERNARY RATE METHOD	20.83336525471613
40	noisy	12.5	(1, 1)	751	1e-06	TERNARY RATE METHOD	19.833334375143025
41	noisy	12.5	(1, 1)	874	1e-05	TERNARY RATE METHOD	19.833333339757115
42	noisy	12.5	(2, 2)	42	0.001	TERNARY RATE METHOD	19.83333989038131
43	noisy	12.5	(2, 2)	2847	1e-06	TERNARY RATE METHOD	20.833333363380408
44	noisy	12.5	(2, 2)	414	1e-05	TERNARY RATE METHOD	21.833339795083706

Рис. 12: Зашумленная функция. Изменяемый шаг. Бонус 2.3

Видно, что количество итераций может достигать больших значений. Да и вообще, нельзя рассчитывать на хоть сколько-то корректные значения с неявной и недифференцируемой функцией.

Рассмотрим мультимодальную функцию. В качестве примера мы взяли функцию Экли. В чем сложность работы с мультимодальными функциями? Мультимодальная функция — это функция с более чем одной “модой” или оптимумом (например долиной на графике). А значит, градиентный метод может забрести не в ту впадину и мы получим не тот ответ, который ожидаем. Мы ожидаем в качестве минимального значения выбранной функции 0. Рассмотрим таблицы.

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	LEARNING_RATE	VALUE	BY_DOT	BY_FUNC
36	мм	0	(0, 0)	2	0.001	0.1	1.7182818284590446	False	True
37	мм	0	(0, 0)	2	1e-06	0.001	1.7182818284590446	False	True
38	мм	0	(0, 0)	1	1e-05	0.001	1.7182818284590446	True	False
39	мм	0	(1, 1)	1771	0.001	0.1	1.7173885360741252	False	True
40	мм	0	(1, 1)	157	1e-06	0.001	5.418707755169727	False	True
41	мм	0	(1, 1)	107	1e-05	0.001	5.4184920065230955	True	False
42	мм	0	(2, 2)	111	0.001	0.1	3.5938976823111553	False	True
43	мм	0	(2, 2)	157	1e-06	0.001	8.390901291065479	False	True
44	мм	0	(2, 2)	101	1e-05	0.001	8.390587534133259	True	False

Рис. 13: Функция Экли. Постоянный шаг. Бонус 2.3

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CHANGING STEP	VALUE
36	мм	0	(0, 0)	2	0.001	TERNARY RATE METHOD	1.7182818284590446
37	мм	0	(0, 0)	2	1e-06	TERNARY RATE METHOD	1.7182818284590446
38	мм	0	(0, 0)	2	1e-05	TERNARY RATE METHOD	1.7182818284590446
39	мм	0	(1, 1)	2	0.001	TERNARY RATE METHOD	0.9062583296075921
40	мм	0	(1, 1)	2	1e-06	TERNARY RATE METHOD	0.9062583296075921
41	мм	0	(1, 1)	2	1e-05	TERNARY RATE METHOD	0.9062583296075921
42	мм	0	(2, 2)	2	0.001	TERNARY RATE METHOD	0.9062583296081712
43	мм	0	(2, 2)	2	1e-06	TERNARY RATE METHOD	0.9062583296081712
44	мм	0	(2, 2)	2	1e-05	TERNARY RATE METHOD	0.9062583296081712

Рис. 14: Функция Экли. Тернарный поиск. Бонус 2.3

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CHANGING STEP	VALUE
36	мм	0	(0, 0)	2	0.001	DICHOTOMY RATE METHOD	1.7182818284590446
37	мм	0	(0, 0)	2	1e-06	DICHOTOMY RATE METHOD	1.7182818284590446
38	мм	0	(0, 0)	2	1e-05	DICHOTOMY RATE METHOD	1.7182818284590446
39	мм	0	(1, 1)	62	0.001	DICHOTOMY RATE METHOD	18.84286402472968
40	мм	0	(1, 1)	62	1e-06	DICHOTOMY RATE METHOD	18.84286402472968
41	мм	0	(1, 1)	62	1e-05	DICHOTOMY RATE METHOD	18.84286402472968
42	мм	0	(2, 2)	9	0.001	DICHOTOMY RATE METHOD	18.8428640247201
43	мм	0	(2, 2)	9	1e-06	DICHOTOMY RATE METHOD	18.8428640247201
44	мм	0	(2, 2)	9	1e-05	DICHOTOMY RATE METHOD	18.8428640247201

Рис. 15: Функция Экли. Дихотомия. Бонус 2.3

Видно, что метод тернарного поиска для этой мультимодальной функции преуспел. Из этого следует, что для мультимодальных методов градиентный метод не подходит в силу нескольких оптимумов.

Картинки:

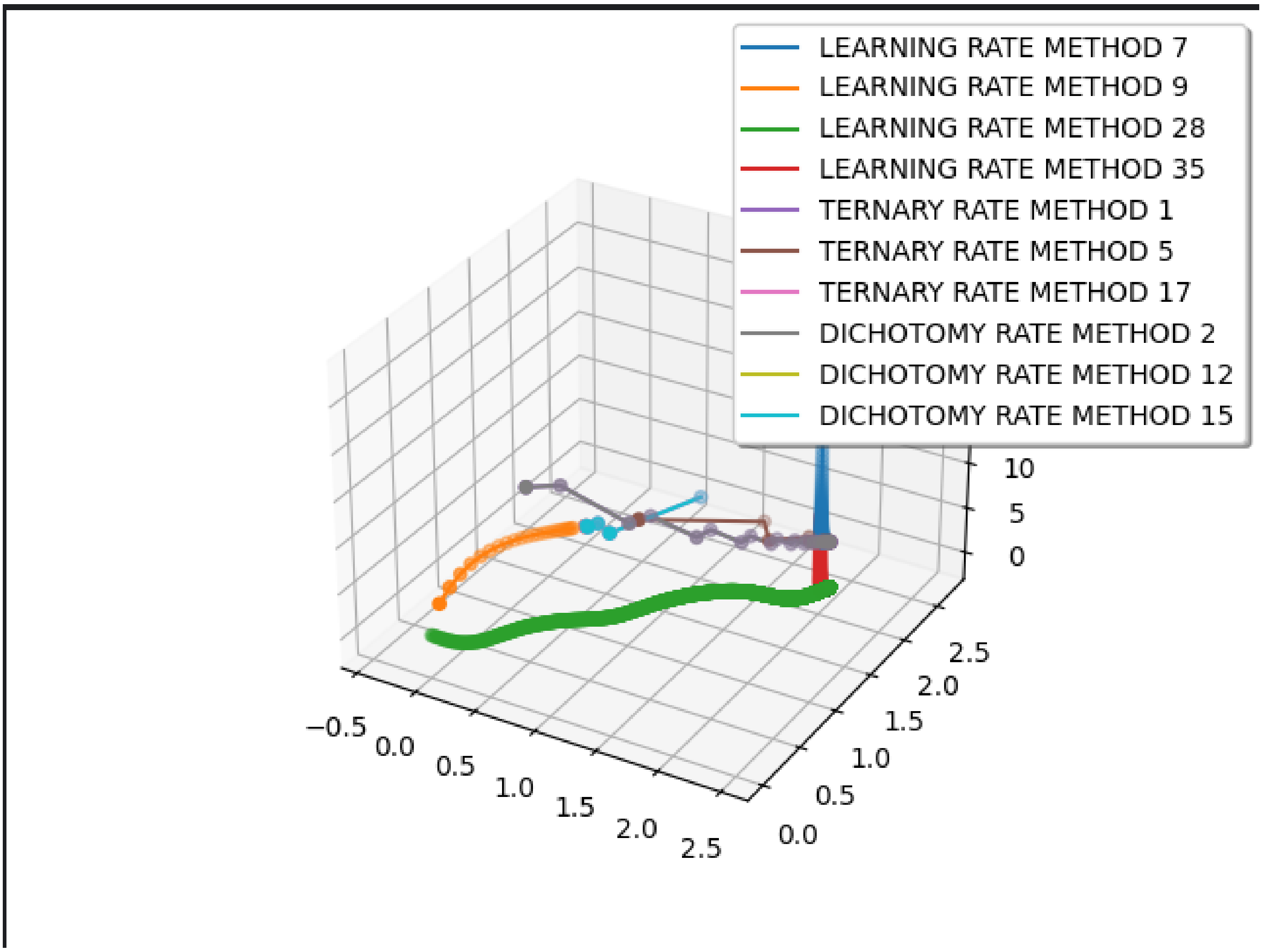

Рис. 16: Блуждания градиента

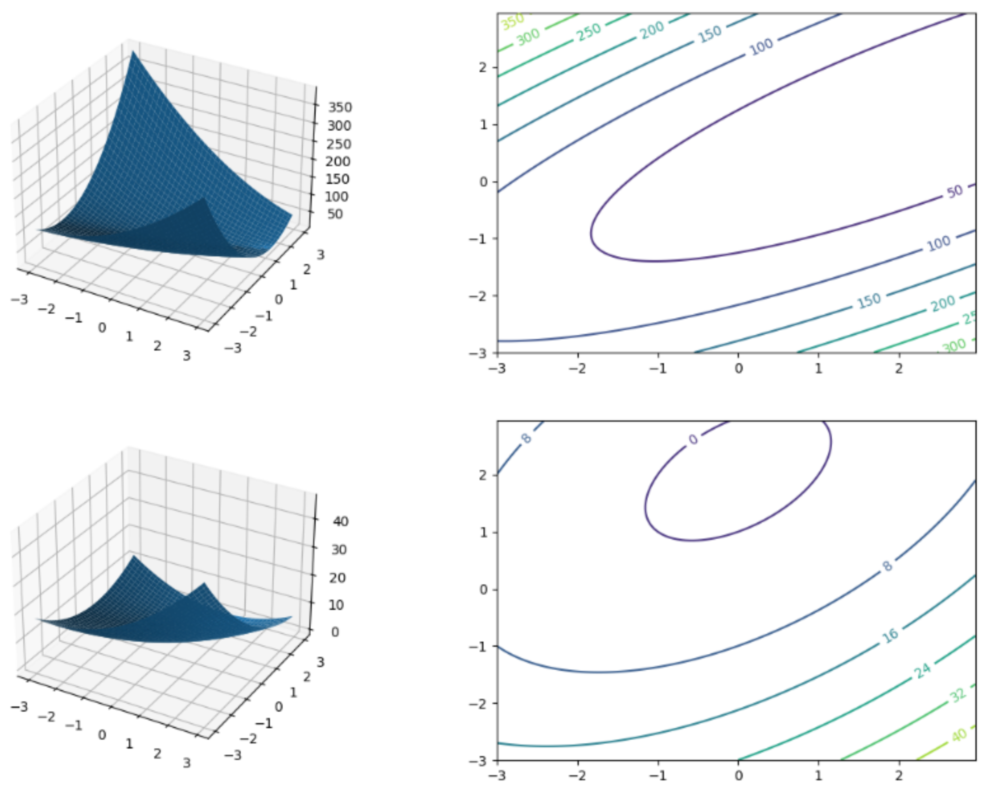

Рис. 17: Функция и линии уровня

Итог:
В ходе лабораторной работы мы узнали много удивительных фактов, а также поработали с разными типами функций, численно нашли значение минимумов и оценили отклонения.