Методы оптимизации.
Отчет по лабораторной работе 2.

Новицкий Илья, Остриченко Илья, Петров Георгий.
M3234

17.04.2024

Ссылка на GitHub : В этот раз мы просто добавили вас в контрибутеров.

Постановка задачи: найти минимальное значение функции $f : \mathbb{R}^n \to \mathbb{R}$, используя методы 2 порядка.
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) Функция Розенброка: $f_3(x, y) = (1 - x)^2 + 100 \cdot (y - x^2)^2$. Точка минимума $(1, 1, 0)$.

(4) Функция "Подштанники": $f_4(x, y) = x^4 + y^4 - 4xy$.
Точки локального минимума: $(-1, -1, -2), (1, 1, -2)$. Седловая точка: $(0, 0, 0)$.
Данная функция будет приведена в качестве примера для дополнительного задании 2.

(5) Неполиномиальная функция: $f_5(x, y) = x^2 \cdot (cos(x) + 2) + y^2(sin(y) + 12)$

Кратко о том, что мы сделали:

Каждую из функций мы протестировали на 3 методах Ньютона: с постоянным шагом, с шагом, который ищется одномерным поиском и с библиотечной реализацией под названием Newton-CG. Помимо этого, в качестве 3 пункта основного задания мы использовали библиотечную реализацию квазиньютоновского метода с типом подсчета матрицы-направления Бройдена-Флетчера-Гольдфарба-Шанно(BFGS) и типом подсчета производной при помощи 3-ех точечного разностного метода. Из-за собственного интереса в устройстве BFGS мы сделали оба пункта 1 доп. задания, а именно - реализовали BFGS самостоятельно. В дополнительном задании 2 мы подобрали функцию, для которой не работает метод Ньютона, а так же нашли точку, применение из которой разных методов Ньютона приводит к разным результатам. Подробное описание экспериментов будет приведено ниже в соответствующих разделах.

Метод Ньютона. С постоянным шагом и с изменяющимся.
Описание метода: Метод Ньютона является методом 2 порядка. Это значит, что для вычисления следующего шага нужно знать вторые производные. Следующий шаг, как и во всех итерационных методах, имеет вид : $x_{k+1} = x_k + \alpha_k \cdot p_k$. Только теперь $p_k = -\left[\nabla^2 f(x_k)\right]^{-1} \cdot \nabla f(x_k)$. То

есть мы домножаем градиент на обратный гессиан. Почему это работает? Фактически - найти минимум эквивалентно нахождению корня производной, а корень можно найти методом касательных. Именно тут это и происходит, мы сдвигаем точку по направлению корня производной. Более того, гессиан отвечает за оценку кривизны функции в окрестности точки. Именно поэтому хочется, чтоб гессиан был **положительно определен** (определенность можно понять при помощи критерия Сильвестра или проверки того, что все собственные значения гессиана положительны. В силу симметричности все собственные значения функции действительны(доказывали на линале)). Значение функции будет минимальным в выпуклой окрестности функции, и как известно, критерий выпуклости - это положительная определенность матрицы вторых производных. Более того, нам нужна сходимость метода, поэтому вектор должен быть направлен в сторону убывания.
Какова **сходимость** данного метода? Она квадратичная в силу теоремы о сходимости метода Ньютона.
Формулировка теоремы - пусть $x*$ - корень уравнения $f(x) = 0$, а функция $f(x)$ дважды дифференцируема в некоторой окрестности $U_r(x*)$, причем первая произодная нигде не обращается в нуль. Тогда, если $0 < m_1 = \inf_{x \in U_r(x*)} |f'(x)|$, $M_2 = \sup_{x \in U_r(x*)} |f''(x)|$. При выборе начальной точки x_0 из той же окрестности $U_r(x*)$ такого, что $\frac{M_2 |x_0 - x*|}{2m_1} = q < 1$ наша итерационная последовательность будет сходиться, причем сходимость будет квадратичная (при переходе от x_k к x_{k+1} ошибка q_k станет q_k^2) :

$$|x_k - x*| < q^{2^k - 1} |x_0 - x*|$$

Результаты метода:
Рассмотрим таблицы, показывающие результаты метода:

CONSTANT STEP NEWTON

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CONSTANT STEP NEWTON	VALUE	TIME
0	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	58	0.001	0.1	12.5016168378554	0.3759646415710449
1	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	599	0.0001	0.01	12.5019386459480	4.192461013793945
2	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	370	0.001	0.01	12.6934547403896	2.5471689701080322
3	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	44	0.001	0.1	12.5004232074891	0.3089470863342285
4	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	449	0.0001	0.01	12.5005415234430	3.2389581203460693
5	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	220	0.001	0.01	12.5540378593627	1.806838035583496
6	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	55	0.001	0.1	12.5001157673391	0.4200103282928467
7	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	562	0.0001	0.01	12.5001551917894	3.930753231048584
8	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	333	0.001	0.01	12.5154863694213	2.015069007873535

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CONSTANT STEP NEWTON	VALUE	TIME
9	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	56	0.001	0.1	-0.999947487934967	0.3047962188720703
10	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	574	0.0001	0.01	-0.999931718753928	3.269421339035034
11	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	345	0.001	0.01	-0.993186310916526	1.9276669025421143
12	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	56	0.001	0.1	-0.999932484487815	0.2636542320251465
13	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	569	0.0001	0.01	-0.999902928003610	2.8533236980438232
14	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	340	0.001	0.01	-0.990313322615386	1.6625080108642578
15	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	52	0.001	0.1	-0.999930292264760	0.28098249435424805
16	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	529	0.0001	0.01	-0.999903595817647	2.8074212074279785
17	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	300	0.001	0.01	-0.990379962834756	1.6843464374542236

Рис. 1

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CONSTANT STEP NEWTON	VALUE	TIME
18	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	58	0.001	0.1	0.00787500934125580	0.32869482040405273
19	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	598	0.0001	0.01	0.00963413485974150	3.0300557613372803
20	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	369	0.001	0.01	0.961376707354222	1.9164776802062988
21	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	157	0.001	0.1	1.64877243576593e-5	1.43229341506958
22	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	1384	0.0001	0.01	1.95526795064165e-5	12.973082780838013
23	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	318	0.001	0.01	12.0778723421856	2.8232369422912598
24	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	79	0.001	0.1	1.48930163179016e-5	0.5979413986206055
25	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	604	0.0001	0.01	1.98380748514900e-5	4.170583486557007
26	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	362	0.001	0.01	0.00233369150985323	2.769706964492798

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CONSTANT STEP NEWTON	VALUE	TIME
27	x^4 + y^4 - 4xy	-2	(1, 5)	83	0.001	0.1	-1.99959463590084	0.5214364528656006
28	x^4 + y^4 - 4xy	-2	(1, 5)	844	0.0001	0.01	-1.99950652327065	5.2480597496032715
29	x^4 + y^4 - 4xy	-2	(1, 5)	594	0.001	0.01	-1.93503662159249	3.5006000995635986
30	x^4 + y^4 - 4xy	-2	(3, 2)	69	0.001	0.1	-1.99966069689323	0.473186731338501
31	x^4 + y^4 - 4xy	-2	(3, 2)	699	0.0001	0.01	-1.99956118834414	4.211199760437012
32	x^4 + y^4 - 4xy	-2	(3, 2)	449	0.001	0.01	-1.94213130250909	2.603781223297119
33	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.001	0.1	0.0	0.003949880599975586
34	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.0001	0.01	0.0	0.003399372100830078
35	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.001	0.01	0.0	0.006226062774658203

Рис. 2

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CONSTANT STEP NEWTON	VALUE	TIME
36	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	122	0.001	0.1	0.000811476742363963	0.5321071147918701
37	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	1238	0.0001	0.01	0.00114824787824282	5.164440870285034
38	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	776	0.001	0.01	0.117537380261324	3.214707851409912
39	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	154	0.001	0.1	1666.01538568793	0.6162171363830566
40	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	1298	0.0001	0.01	0.000372423327844756	5.603110313415527
41	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	840	0.001	0.01	0.0372267287306106	3.5385169982910156
42	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	2	0.001	0.1	0	0.00615239143371582
43	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	2	0.0001	0.01	0	0.0051174163818359375
44	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	2	0.001	0.01	0	0.010095834732055664

Рис. 3

Все константные значения указаны в колонке CONSTANT STEP NEWTON.

Теперь для шага с изменяющимся α_k, мы приняли решение изменять его тернарным поиском.

TERNARY STEP NEWTON

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	TERNARY STEP NEWTON	VALUE	TIME
0	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	2	0.001	TERNARY STEP NEWTON	12.5000000000000	0.525947093963623
1	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	2	0.0001	TERNARY STEP NEWTON	12.5000000000000	0.4649007320404053
2	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	2	0.001	TERNARY STEP NEWTON	12.5000000000000	0.4627556800842285
3	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.001	TERNARY STEP NEWTON	12.5000000000000	0.5321767330169678
4	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.0001	TERNARY STEP NEWTON	12.5000000000000	0.4693286418914795
5	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.001	TERNARY STEP NEWTON	12.5000000000000	0.4644201469421387
6	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	2	0.001	TERNARY STEP NEWTON	12.5000000000000	0.44893884658813477
7	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	2	0.0001	TERNARY STEP NEWTON	12.5000000000000	0.3006861209869385
8	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	2	0.001	TERNARY STEP NEWTON	12.5000000000000	0.24950027465820312

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	TERNARY STEP NEWTON	VALUE	TIME
9	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	2	0.001	TERNARY STEP NEWTON	-1.00000000000000	0.28316521644592285
10	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	2	0.0001	TERNARY STEP NEWTON	-1.00000000000000	0.1435532569885254
11	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	2	0.001	TERNARY STEP NEWTON	-1.00000000000000	0.10973954200744629
12	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	2	0.001	TERNARY STEP NEWTON	-1.00000000000000	0.24681663513183594
13	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	2	0.0001	TERNARY STEP NEWTON	-1.00000000000000	0.09966206550598145
14	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	2	0.001	TERNARY STEP NEWTON	-1.00000000000000	0.10149025917053223
15	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	2	0.001	TERNARY STEP NEWTON	-1.00000000000000	0.20395112037658691
16	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	2	0.0001	TERNARY STEP NEWTON	-1.00000000000000	0.1367640495300293
17	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	2	0.001	TERNARY STEP NEWTON	-1.00000000000000	0.0844573974609375

Рис. 4

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	TERNARY STEP NEWTON	VALUE	TIME
18	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	2	0.001	TERNARY STEP NEWTON	1.06856139992844e-25	0.18134737014770508
19	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	2	0.0001	TERNARY STEP NEWTON	1.06856139992844e-25	0.09894156455993652
20	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	2	0.001	TERNARY STEP NEWTON	1.06856139992844e-25	0.07739615440368652
21	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	12	0.001	TERNARY STEP NEWTON	6.38860081505976e-13	3.461925983428955
22	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	13	0.0001	TERNARY STEP NEWTON	3.80491957109996e-24	3.7200005054473877
23	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	12	0.001	TERNARY STEP NEWTON	6.38860081505976e-13	3.180324077606201
24	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	8	0.001	TERNARY STEP NEWTON	1.67722814717378e-17	1.446875810623169
25	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	8	0.0001	TERNARY STEP NEWTON	1.67722814717378e-17	1.459545373916626
26	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	8	0.001	TERNARY STEP NEWTON	1.67722814717378e-17	1.496325969696045

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	TERNARY STEP NEWTON	VALUE	TIME
27	x^4 + y^4 - 4xy	-2	(1, 5)	4	0.001	TERNARY STEP NEWTON	-2.00000000000000	0.5673439502716064
28	x^4 + y^4 - 4xy	-2	(1, 5)	5	0.0001	TERNARY STEP NEWTON	-2.00000000000000	0.5618832111358643
29	x^4 + y^4 - 4xy	-2	(1, 5)	4	0.001	TERNARY STEP NEWTON	-2.00000000000000	0.37877774238586426
30	x^4 + y^4 - 4xy	-2	(3, 2)	4	0.001	TERNARY STEP NEWTON	-2.00000000000000	0.5108892917633057
31	x^4 + y^4 - 4xy	-2	(3, 2)	4	0.0001	TERNARY STEP NEWTON	-2.00000000000000	0.376570463180542
32	x^4 + y^4 - 4xy	-2	(3, 2)	4	0.001	TERNARY STEP NEWTON	-2.00000000000000	0.41791725158691406
33	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.001	TERNARY STEP NEWTON	0.0	0.014600992202758789
34	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.0001	TERNARY STEP NEWTON	0.0	0.013059139251708984
35	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.001	TERNARY STEP NEWTON	0.0	0.013527870178222656

Рис. 5

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	TERNARY STEP NEWTON	VALUE	TIME
36	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	294	0.001	TERNARY STEP NEWTON	3.03614956788766e-24	1.2051889896392822
37	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	294	0.0001	TERNARY STEP NEWTON	3.03614956788766e-24	1.1747856140136719
38	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	294	0.001	TERNARY STEP NEWTON	3.03614956788766e-24	1.1453893184661865
39	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	168	0.001	TERNARY STEP NEWTON	7.44578306322710e-24	0.7680611610412598
40	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	168	0.0001	TERNARY STEP NEWTON	7.44578306322710e-24	0.4057796001434326
41	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	168	0.001	TERNARY STEP NEWTON	7.44578306322710e-24	0.4411966800689697
42	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	42	0.001	TERNARY STEP NEWTON	0	0.0189669132232666
43	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	42	0.0001	TERNARY STEP NEWTON	0	0.014774799346923828
44	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	42	0.001	TERNARY STEP NEWTON	0	0.014425992965698242

Рис. 6

Уже здесь видно, что на 4-ой функции методы работают плохо в начальной точке $(0, 0, 0)$ - они останвливаются в седле, считая, что это локальный минимум.

Метод Ньютона. Scipy.

В методе, заполняющем таблицы вызывается метод optimize.minimize с методом, называемым "Newton-CG".

```
if newton_name == NAME.NEWTON_CG:
    res_optimize_scipy = optimize.minimize(lambda xy: compute_func(xy[0], xy[1]),
                                           np.array([INIT_POINTS[i][0], INIT_POINTS[i][1]]),
                                           method="Newton-CG",
                                           jac=lambda xy: compute_gradient(xy[0], xy[1]))
```

Рис. 7

NEWTON CG METHOD

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	NEWTON CG METHOD	VALUE	TIME
0	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	5	0.001	NEWTON CG METHOD	12.5	0.001035451889038086
1	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	5	0.0001	NEWTON CG METHOD	12.5	0.0009739398956298828
2	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	5	0.001	NEWTON CG METHOD	12.5	0.0008716583251953125
3	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	3	0.001	NEWTON CG METHOD	12.5	0.0009496212005615234
4	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	3	0.0001	NEWTON CG METHOD	12.5	0.0008127689361572266
5	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	3	0.001	NEWTON CG METHOD	12.5	0.0009381771087646484
6	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	2	0.001	NEWTON CG METHOD	12.5	0.0004487037658691406
7	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	2	0.0001	NEWTON CG METHOD	12.5	0.0004818439483642578
8	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	2	0.001	NEWTON CG METHOD	12.5	0.0004656314849853516

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	NEWTON CG METHOD	VALUE	TIME
9	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	3	0.001	NEWTON CG METHOD	-1.0	0.0007283687591552734
10	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	3	0.0001	NEWTON CG METHOD	-1.0	0.0006353855133056641
11	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	3	0.001	NEWTON CG METHOD	-1.0	0.0007791519165039062
12	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	3	0.001	NEWTON CG METHOD	-1.0	0.0006940364837646484
13	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	3	0.0001	NEWTON CG METHOD	-1.0	0.0006101131439208984
14	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	3	0.001	NEWTON CG METHOD	-1.0	0.0006175041198730469
15	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	5	0.001	NEWTON CG METHOD	-1.0	0.0008118152618408203
16	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	5	0.0001	NEWTON CG METHOD	-1.0	0.0008900165557861328
17	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	5	0.001	NEWTON CG METHOD	-1.0	0.0007719993591308594

Рис. 8

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	NEWTON CG METHOD	VALUE	TIME
18	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	40	0.001	NEWTON CG METHOD	3.0785778289605954e-09	0.00687861442565918
19	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	40	0.0001	NEWTON CG METHOD	3.0785778289605954e-09	0.006655693054199219
20	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	40	0.001	NEWTON CG METHOD	3.0785778289605954e-09	0.0068132877349853516
21	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	28	0.001	NEWTON CG METHOD	1.0072784403711728e-12	0.004942655563354492
22	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	28	0.0001	NEWTON CG METHOD	1.0072784403711728e-12	0.004542827606201172
23	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	28	0.001	NEWTON CG METHOD	1.0072784403711728e-12	0.004950284957885742
24	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	31	0.001	NEWTON CG METHOD	3.949981563726443e-16	0.006183147430419922
25	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	31	0.0001	NEWTON CG METHOD	3.949981563726443e-16	0.006239414215087891
26	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	31	0.001	NEWTON CG METHOD	3.949981563726443e-16	0.005311012268066406

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	NEWTON CG METHOD	VALUE	TIME
27	x^4 + y^4 - 4xy	-2	(1, 5)	10	0.001	NEWTON CG METHOD	-2.0	0.00157928466796875
28	x^4 + y^4 - 4xy	-2	(1, 5)	10	0.0001	NEWTON CG METHOD	-2.0	0.0012645721435546875
29	x^4 + y^4 - 4xy	-2	(1, 5)	10	0.001	NEWTON CG METHOD	-2.0	0.0012311935424804688
30	x^4 + y^4 - 4xy	-2	(3, 2)	8	0.001	NEWTON CG METHOD	-2.0	0.0011053085327148438
31	x^4 + y^4 - 4xy	-2	(3, 2)	8	0.0001	NEWTON CG METHOD	-2.0	0.0010628700256347656
32	x^4 + y^4 - 4xy	-2	(3, 2)	8	0.001	NEWTON CG METHOD	-2.0	0.0014998912811279297
33	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.001	NEWTON CG METHOD	0.0	0.00031256675720214844
34	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.0001	NEWTON CG METHOD	0.0	0.0003483295440673828
35	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.001	NEWTON CG METHOD	0.0	0.00033664703369140625

Рис. 9

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	NEWTON CG METHOD	VALUE	TIME
36	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	7	0.001	NEWTON CG METHOD	1.083076886790746e-21	0.0017075538635253906
37	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	7	0.0001	NEWTON CG METHOD	1.083076886790746e-21	0.001280069351196289
38	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	7	0.001	NEWTON CG METHOD	1.083076886790746e-21	0.0012445449829101562
39	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	7	0.001	NEWTON CG METHOD	7.042093481955985e-33	0.0014040470123291016
40	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	7	0.0001	NEWTON CG METHOD	7.042093481955985e-33	0.001425027847290039
41	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	7	0.001	NEWTON CG METHOD	7.042093481955985e-33	0.0013644695281982422
42	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	1	0.001	NEWTON CG METHOD	0.0	0.0003712177276611328
43	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	1	0.0001	NEWTON CG METHOD	0.0	0.0003597736358642578
44	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	1	0.001	NEWTON CG METHOD	0.0	0.000263214111328125

Рис. 10

Заметим, что метод работает некорректно для функции "подштанники"на некоторых начальных значениях.

Наш Ньютон с ternary-шагом:

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	TERNARY STEP NEWTON	VALUE	TIME
0	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	2	0.001	TERNARY STEP NEWTON	12.5000000000000	0.3371100425720215
1	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	2	0.0001	TERNARY STEP NEWTON	12.5000000000000	0.27778196334838867
2	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	2	0.001	TERNARY STEP NEWTON	12.5000000000000	0.2772250175476074
3	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.001	TERNARY STEP NEWTON	12.5000000000000	0.2938239574432373
4	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.0001	TERNARY STEP NEWTON	12.5000000000000	0.30898094177246094
5	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.001	TERNARY STEP NEWTON	12.5000000000000	0.27796316146850586

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	TERNARY STEP NEWTON	VALUE	TIME
9	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	2	0.001	TERNARY STEP NEWTON	-1.00000000000000	0.16210675239562988
10	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	2	0.0001	TERNARY STEP NEWTON	-1.00000000000000	0.06010103225708008
11	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	2	0.001	TERNARY STEP NEWTON	-1.00000000000000	0.05917692184448242
12	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	2	0.001	TERNARY STEP NEWTON	-1.00000000000000	0.14541006088256836
13	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	2	0.0001	TERNARY STEP NEWTON	-1.00000000000000	0.056118011474609375
14	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	2	0.001	TERNARY STEP NEWTON	-1.00000000000000	0.05574512481689453

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	TERNARY STEP NEWTON	VALUE	TIME
18	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	2	0.001	TERNARY STEP NEWTON	1.06856139992844e-25	0.10238289833068848
19	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	2	0.0001	TERNARY STEP NEWTON	1.06856139992844e-25	0.04445624351501465
20	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	2	0.001	TERNARY STEP NEWTON	1.06856139992844e-25	0.04435300827026367
21	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	12	0.001	TERNARY STEP NEWTON	6.38774576689944e-13	2.019995927810669
22	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	13	0.0001	TERNARY STEP NEWTON	3.74167821935401e-24	2.0649819374084473
23	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	12	0.001	TERNARY STEP NEWTON	6.38774576689944e-13	1.908888816833496

Теперь давайте сравним версии метода Ньютона, которые мы использовали

Библиотечный Ньютон

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	NEWTON CG METHOD	VALUE	TIME
0	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	5	0.001	NEWTON CG METHOD	12.5	0.003047943115234375
1	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	5	0.0001	NEWTON CG METHOD	12.5	0.0005471706390380859
2	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	5	0.001	NEWTON CG METHOD	12.5	0.0006480216979980469
3	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	3	0.001	NEWTON CG METHOD	12.5	0.0005600452423095703
4	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	3	0.0001	NEWTON CG METHOD	12.5	0.0006339550018310547
5	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	3	0.001	NEWTON CG METHOD	12.5	0.0004680156707763672

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	NEWTON CG METHOD	VALUE	TIME
9	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	3	0.001	NEWTON CG METHOD	-1.0	0.0006709098815917969
10	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	3	0.0001	NEWTON CG METHOD	-1.0	0.0005850791931152344
11	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	3	0.001	NEWTON CG METHOD	-1.0	0.00042319297790527344
12	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	3	0.001	NEWTON CG METHOD	-1.0	0.001447916030883789
13	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	3	0.0001	NEWTON CG METHOD	-1.0	0.0003540515899658203
14	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	3	0.001	NEWTON CG METHOD	-1.0	0.00030803680419921875

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	NEWTON CG METHOD	VALUE	TIME
18	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	36	0.001	NEWTON CG METHOD	6.518261713360773e-14	0.0038499832153320312
19	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	36	0.0001	NEWTON CG METHOD	6.518261713360773e-14	0.00317120552062988 3
20	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	36	0.001	NEWTON CG METHOD	6.518261713360773e-14	0.003155946731567383
21	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	28	0.001	NEWTON CG METHOD	1.3109536103429905e-16	0.002442598342895508
22	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	28	0.0001	NEWTON CG METHOD	1.3109536103429905e-16	0.002429962158203125
23	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	28	0.001	NEWTON CG METHOD	1.3109536103429905e-16	0.002416849136352539

Сравнение наших Ньютонов и библиотечных

Наш Ньютон с постоянным шагом:

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CONSTANT STEP NEWTON	VALUE	TIME
0	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	58	0.001	0.1	12.5016168378554	0.3594980239868164
1	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	599	0.0001	0.01	12.5019386459480	2.352436065673828
2	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	370	0.001	0.01	12.6934547403896	1.2967369556427002
3	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	44	0.001	0.1	12.5004232074891	0.1648712158203125
4	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	449	0.0001	0.01	12.5005415234430	1.6890439987182617
5	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	220	0.001	0.01	12.5540378593627	0.7978041172027588

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CONSTANT STEP NEWTON	VALUE	TIME
9	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	56	0.001	0.1	-0.999947487934967	0.1570289134979248
10	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	574	0.0001	0.01	-0.999931718753928	1.572364091873169
11	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	345	0.001	0.01	-0.993186310916526	0.9595599174499512
12	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	56	0.001	0.1	-0.999932484487815	0.1455228328704834
13	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	569	0.0001	0.01	-0.999902928003610	1.490664005279541
14	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	340	0.001	0.01	-0.990313322615386	0.8503851890563965

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CONSTANT STEP NEWTON	VALUE	TIME
18	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	58	0.001	0.1	0.00787500934125695	0.3970658779144287
19	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	598	0.0001	0.01	0.00963413485974150	2.091766834259033
20	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	369	0.001	0.01	0.961376707354222	1.5229699611663818
21	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	157	0.001	0.1	1.64877243577159e-5	1.0629220008850098
22	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	1384	0.0001	0.01	1.95526795064899e-5	6.684416055679321
23	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	318	0.001	0.01	12.0778723421856	1.5785191059112549

Подведем итоги. Как и ожидалась, Ньютон с постоянным шагом работает намного хуже, чем Ньютон с одномерным поиском. А вот что интересно, так это то, что наш Ньютон с одномерным поиском работает быстрее и точнее, чем библиотечный Ньютон. |

Квазиньютоновские методы.

В квазиньютоновских методах направление спуска d_k полагается равным $-H_k \nabla f(x_k)$, где $H_k \in \mathbb{S}^n_{++}$ — некоторая матрица. Если бы матрица H_k в точности равнялась обратному гессиану $[\nabla^2 f(x_k)]^{-1}$, то получился бы метод Ньютона. Вместо этого в квазиньютоновских методах матрица H_k всего лишь «аппроксимирует» обратный гессиан — отсюда и название *квази*-ньютоновские методы.

Матрицы H_k в квазиньютоновских методах строятся следующим образом. На самой первой итерации выбирается некоторая начальная матрица $H_0 \in \mathbb{S}^n_{++}$ (обычно $H_0 = I_n$). Далее матрицы H_k обновляются в итерациях — на каждой итерации метода

(a) вычисляется новая точка $x_{k+1} := x_k - \alpha_k H_k \nabla f(x_k)$;

(b) выполняется обновление матрицы: $H_k \to H_{k+1}$.

Все квазиньютоновские методы отличаются между собой лишь способом обновления матрицы H_k. Напомним, что, согласно наложенным выше ограничениям, процедура обновления матрицы H_k должна быть достаточно эффективной: сложность этой процедуры должна составлять $O(n^2)$, и при этом не должно быть никаких вычислений гессиана. Таким образом, процедура обновления матрицы H_k должна каким-то эффективным образом «подгрузить» новую информацию о гессиане, при этом не вычисляя сам гессиан. Для этого используются градиенты $\nabla f(x_k)$ и $\nabla f(x_{k+1})$, а также следующее правило.

Квазиньютоновское правило

Выбрать новую матрицу H_{k+1} таким образом, чтобы выполнялось *уравнение секущей*:

$$H_{k+1}[\nabla f(x_{k+1}) - \nabla f(x_k)] = x_{k+1} - x_k. \tag{1.2}$$

Схема Бройдена–Флетчера–Гольдфарба–Шанно (BFGS):

$$H_{k+1} = \left(I_n - \frac{s_k y_k^T}{\langle y_k, s_k\rangle}\right) H_k \left(I_n - \frac{y_k s_k^T}{\langle y_k, s_k\rangle}\right) + \frac{s_k s_k^T}{\langle y_k, s_k\rangle}.$$

Общие выводы:

— Итерации BFGS вычислительно проще итераций метода Ньютона и не требуют вычисления гессиана;

— По скорости сходимости BFGS уступает методу Ньютона, но все равно является достаточно быстрым;

— По прежнему требуется $O(d^2)$ памяти, что по-прежнему вызывает проблемы при большой размерности ($10^4 - 10^5$).

— Время выполнения итерации $O(d^2)$ гораздо лучше, чем $O(d^3)$ метода Ньютона, но всё ещё оставляет желать лучшего.

Пример работы Квазиньютоновских методов.

QUASI-NEWTON METHOD

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-NEWTON METHOD	VALUE	TIME
0	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	2	0.001	QUASI-NEWTON METHOD	12.5	0.007157087326049805
1	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	2	0.0001	QUASI-NEWTON METHOD	12.5	0.007081031799316406
2	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	2	0.001	QUASI-NEWTON METHOD	12.5	0.006824970245361328
3	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.001	QUASI-NEWTON METHOD	12.5	0.006644010543823242
4	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.0001	QUASI-NEWTON METHOD	12.5	0.006587982177734375
5	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.001	QUASI-NEWTON METHOD	12.5	0.007187843322753906
6	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	2	0.001	QUASI-NEWTON METHOD	12.5	0.007564067840576172
7	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	2	0.0001	QUASI-NEWTON METHOD	12.5	0.006981849670410156
8	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	2	0.001	QUASI-NEWTON METHOD	12.5	0.006701946258544922

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-NEWTON METHOD	VALUE	TIME
9	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	3	0.001	QUASI-NEWTON METHOD	-1.0	0.00966191291809082
10	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	3	0.0001	QUASI-NEWTON METHOD	-1.0	0.009257793426513672
11	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	3	0.001	QUASI-NEWTON METHOD	-1.0	0.009350776672363281
12	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	3	0.001	QUASI-NEWTON METHOD	-1.0	0.009166955947875977
13	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	3	0.0001	QUASI-NEWTON METHOD	-1.0	0.009050130844116211
14	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	3	0.001	QUASI-NEWTON METHOD	-1.0	0.009132862091064453
15	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	3	0.001	QUASI-NEWTON METHOD	-1.0	0.009377002716064453
16	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	3	0.0001	QUASI-NEWTON METHOD	-1.0	0.010059118270874023
17	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	3	0.001	QUASI-NEWTON METHOD	-1.0	0.00924992561340332

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-NEWTON METHOD	VALUE	TIME
18	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	23	0.001	QUASI-NEWTON METHOD	8.654302687290901e-12	0.09782290458679199
19	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	23	0.0001	QUASI-NEWTON METHOD	8.654302687290901e-12	0.08607602119445801
20	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	23	0.001	QUASI-NEWTON METHOD	8.654302687290901e-12	0.08741378784179688
21	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	55	0.001	QUASI-NEWTON METHOD	1.536223031816008e-10	0.19609713554382324
22	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	56	0.0001	QUASI-NEWTON METHOD	5.147817401746769e-12	0.19911718368530273
23	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	55	0.001	QUASI-NEWTON METHOD	1.536223031816008e-10	0.20284700393676758
24	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	18	0.001	QUASI-NEWTON METHOD	7.750964550677903e-11	0.08083581924438477
25	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	19	0.0001	QUASI-NEWTON METHOD	7.11455747502704e-17	0.0750267505645752
26	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	18	0.001	QUASI-NEWTON METHOD	7.750964550677903e-11	0.07273101806640625

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-NEWTON METHOD	VALUE	TIME
27	x^4 + y^4 - 4xy	-2	(1, 5)	12	0.001	QUASI-NEWTON METHOD	-1.9999999994913187	0.035241127014160156
28	x^4 + y^4 - 4xy	-2	(1, 5)	12	0.0001	QUASI-NEWTON METHOD	-1.9999999994913187	0.035382986068725586
29	x^4 + y^4 - 4xy	-2	(1, 5)	12	0.001	QUASI-NEWTON METHOD	-1.9999999994913187	0.035032033920288086
30	x^4 + y^4 - 4xy	-2	(3, 2)	8	0.001	QUASI-NEWTON METHOD	-1.9999999987196535	0.02506399154663086
31	x^4 + y^4 - 4xy	-2	(3, 2)	9	0.0001	QUASI-NEWTON METHOD	-1.999999999999987	0.02667999267578125
32	x^4 + y^4 - 4xy	-2	(3, 2)	8	0.001	QUASI-NEWTON METHOD	-1.9999999987196535	0.02350592613220215
33	x^4 + y^4 - 4xy	-2	(0, 0)	0	0.001	QUASI-NEWTON METHOD	0	0.0009579658508300781
34	x^4 + y^4 - 4xy	-2	(0, 0)	0	0.0001	QUASI-NEWTON METHOD	0	0.0010292530059814453
35	x^4 + y^4 - 4xy	-2	(0, 0)	0	0.001	QUASI-NEWTON METHOD	0	0.0009317398071289062

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-NEWTON METHOD	VALUE	TIME
36	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	6	0.001	QUASI-NEWTON METHOD	1.2428112594694937e-09	0.026246070861816406
37	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	7	0.0001	QUASI-NEWTON METHOD	7.291582007476773e-13	0.02987504005432129
38	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	6	0.001	QUASI-NEWTON METHOD	1.2428112594694937e-09	0.0262069702148437 5
39	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	6	0.001	QUASI-NEWTON METHOD	2.8291950466518396e-15	0.027562856674194336
40	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	6	0.0001	QUASI-NEWTON METHOD	2.8291950466518396e-15	0.027172088623046875
41	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	6	0.001	QUASI-NEWTON METHOD	2.8291950466518396e-15	0.026768922805786133
42	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	0	0.001	QUASI-NEWTON METHOD	0.0	0.0014662742614746094
43	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	0	0.0001	QUASI-NEWTON METHOD	0.0	0.0013356208801269531
44	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	0	0.001	QUASI-NEWTON METHOD	0.0	0.0012350082397460938

QUASI-SCIPY

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-SCIPY	VALUE	TIME
0	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	9	0.001	QUASI-SCIPY	12.5	0.0008759498596191406
1	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	9	0.0001	QUASI-SCIPY	12.5	0.000701904296875
2	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	9	0.001	QUASI-SCIPY	12.5	0.0006499290466308594
3	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.001	QUASI-SCIPY	12.5	0.00029015541076660156
4	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.0001	QUASI-SCIPY	12.5	0.00028395652770996094
5	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.001	QUASI-SCIPY	12.5	0.0002849102020263672
6	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	4	0.001	QUASI-SCIPY	12.500000000001673	0.00033092498779296875
7	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	4	0.0001	QUASI-SCIPY	12.500000000001673	0.00033593177795410156
8	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	4	0.001	QUASI-SCIPY	12.500000000001673	0.0004911422729492188

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-SCIPY	VALUE	TIME
9	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	6	0.001	QUASI-SCIPY	-0.9999999999999849	0.0005247592926025391
10	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	6	0.0001	QUASI-SCIPY	-0.9999999999999849	0.0004658699035644531
11	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	6	0.001	QUASI-SCIPY	-0.9999999999999849	0.0004589557647705078
12	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	5	0.001	QUASI-SCIPY	-0.9999999999999671	0.0003998279571533203
13	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	5	0.0001	QUASI-SCIPY	-0.9999999999999671	0.00041103363037109375
14	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	5	0.001	QUASI-SCIPY	-0.9999999999999671	0.00039505958557128906
15	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	5	0.001	QUASI-SCIPY	-0.9999999999999929	0.00039124488830566406
16	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	5	0.0001	QUASI-SCIPY	-0.9999999999999929	0.00040912628173828125
17	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	5	0.001	QUASI-SCIPY	-0.9999999999999929	0.0005166530609130859

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-SCIPY	VALUE	TIME
18	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	35	0.001	QUASI-SCIPY	5.134728361159098e-13	0.002471923828125
19	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	35	0.0001	QUASI-SCIPY	5.134728361159098e-13	0.002403736114501953
20	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	35	0.001	QUASI-SCIPY	5.134728361159098e-13	0.002680063247680664
21	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	43	0.001	QUASI-SCIPY	1.0577699776892852e-14	0.0033180713653564453
22	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	43	0.0001	QUASI-SCIPY	1.0577699776892852e-14	0.003323078155517578
23	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	43	0.001	QUASI-SCIPY	1.0577699776892852e-14	0.003114938735961914
24	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	19	0.001	QUASI-SCIPY	7.84435581094168e-13	0.00139617919921875
25	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	19	0.0001	QUASI-SCIPY	7.84435581094168e-13	0.0015101432800292969
26	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	19	0.001	QUASI-SCIPY	7.84435581094168e-13	0.0014138221740722656

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-SCIPY	VALUE	TIME
27	x^4 + y^4 - 4xy	-2	(1, 5)	16	0.001	QUASI-SCIPY	-1.9999999999994948	0.001071929931640625
28	x^4 + y^4 - 4xy	-2	(1, 5)	16	0.0001	QUASI-SCIPY	-1.9999999999994948	0.001071929931640625
29	x^4 + y^4 - 4xy	-2	(1, 5)	16	0.001	QUASI-SCIPY	-1.9999999999994948	0.0010759830474853516
30	x^4 + y^4 - 4xy	-2	(3, 2)	13	0.001	QUASI-SCIPY	-1.9999999999999414	0.0010449886322021484
31	x^4 + y^4 - 4xy	-2	(3, 2)	13	0.0001	QUASI-SCIPY	-1.9999999999999414	0.0009849071502685547
32	x^4 + y^4 - 4xy	-2	(3, 2)	13	0.001	QUASI-SCIPY	-1.9999999999999414	0.0009379386901855469
33	x^4 + y^4 - 4xy	-2	(0, 0)	0	0.001	QUASI-SCIPY	0.0	8.0108642578125e-05
34	x^4 + y^4 - 4xy	-2	(0, 0)	0	0.0001	QUASI-SCIPY	0.0	7.390975952148438e-05
35	x^4 + y^4 - 4xy	-2	(0, 0)	0	0.001	QUASI-SCIPY	0.0	7.200241088867188e-05

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-SCIPY	VALUE	TIME
36	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	7	0.001	QUASI-SCIPY	3.0014014380431847e-13	0.0006589889526367188
37	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	7	0.0001	QUASI-SCIPY	3.0014014380431847e-13	0.0006330013275146484
38	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	7	0.001	QUASI-SCIPY	3.0014014380431847e-13	0.0006251335144042969
39	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	7	0.001	QUASI-SCIPY	3.6592213519708e-16	0.0006072521209716797
40	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	7	0.0001	QUASI-SCIPY	3.6592213519708e-16	0.0005917549133300781
41	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	7	0.001	QUASI-SCIPY	3.6592213519708e-16	0.0005879402160644531
42	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	0	0.001	QUASI-SCIPY	0.0	7.915496826171875e-05
43	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	0	0.0001	QUASI-SCIPY	0.0	7.700920104980469e-05
44	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	0	0.001	QUASI-SCIPY	0.0	7.772445678710938e-05

В данных таблицах метод под названием QUASI-NEWTON METHOD - это наша реализация (доп. задание 1.2), а QUASI-SCIPY - библиотечная реализация.

Сравним эффективность Нелдера-Меда(метод нулевого порядка) с Квазиньютоновским методом, где производная вычиляется разностым методом.

Нельдер-Мид:

FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	VALUE
x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(1, 5)	53	12.50000000448824
x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(3, 2)	37	12.500000001436103
x**2 + (x - 5)**2 + (2*x - 4*y)**2	12.5	(0, 0)	65	12.500000001737911

FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	VALUE
x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(1, 5)	45	-0.9999999991220854
x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(3, 2)	48	-0.9999999998121147
x**2 - x*y + 2*x + y**2 - 4*y + 3	-1	(0, 0)	68	-0.9999999991925863

FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	VALUE
(1 - x)**2 + 100*(-x**2 + y)**2	0	(1, 5)	46	7.688182664813061e-09
(1 - x)**2 + 100*(-x**2 + y)**2	0	(3, 2)	55	2.2008132703793614e-09
(1 - x)**2 + 100*(-x**2 + y)**2	0	(0, 0)	7	0.0

Квазиньютоновский метод:

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-SCIPY	VALUE
0	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	9	0.001	QUASI-SCIPY	12.5
1	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	9	0.0001	QUASI-SCIPY	12.5
2	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	9	0.001	QUASI-SCIPY	12.5
3	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.001	QUASI-SCIPY	12.5
4	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.0001	QUASI-SCIPY	12.5
5	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	2	0.001	QUASI-SCIPY	12.5
6	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	4	0.001	QUASI-SCIPY	12.500000000001673
7	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	4	0.0001	QUASI-SCIPY	12.500000000001673
8	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	4	0.001	QUASI-SCIPY	12.500000000001673

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-SCIPY	VALUE
9	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	6	0.001	QUASI-SCIPY	-0.9999999999999849
10	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	6	0.0001	QUASI-SCIPY	-0.9999999999999849
11	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	6	0.001	QUASI-SCIPY	-0.9999999999999849
12	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	5	0.001	QUASI-SCIPY	-0.9999999999999671
13	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	5	0.0001	QUASI-SCIPY	-0.9999999999999671
14	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	5	0.001	QUASI-SCIPY	-0.9999999999999671
15	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	5	0.001	QUASI-SCIPY	-0.9999999999999929
16	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	5	0.0001	QUASI-SCIPY	-0.9999999999999929
17	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	5	0.001	QUASI-SCIPY	-0.9999999999999929

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	QUASI-SCIPY	VALUE
18	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	35	0.001	QUASI-SCIPY	5.134728361159098e-13
19	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	35	0.0001	QUASI-SCIPY	5.134728361159098e-13
20	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	35	0.001	QUASI-SCIPY	5.134728361159098e-13
21	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	43	0.001	QUASI-SCIPY	1.0577699776892852e-14
22	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	43	0.0001	QUASI-SCIPY	1.0577699776892852e-14
23	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	43	0.001	QUASI-SCIPY	1.0577699776892852e-14
24	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	19	0.001	QUASI-SCIPY	7.84435581094168e-13
25	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	19	0.0001	QUASI-SCIPY	7.84435581094168e-13
26	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	19	0.001	QUASI-SCIPY	7.84435581094168e-13

Как можно заметить, Нелдер-Мид опять “в отстающих” |

Дополнительное задание 1.1

Пример работы метода Ньютона с условием Вольфе.

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	WOLFE CONDITION	VALUE	TIME
0	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	589	0.001	WOLFE CONDITION	12.5003221918625	0.9930770397186279
1	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	798	0.0001	WOLFE CONDITION	12.5000023773563	1.3642418384552002
2	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(1, 5)	589	0.001	WOLFE CONDITION	12.5003221918625	0.9839041233062744
3	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	475	0.001	WOLFE CONDITION	12.5000642261462	0.8812508583068848
4	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	665	0.0001	WOLFE CONDITION	12.5000007404771	1.094264030456543
5	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(3, 2)	475	0.001	WOLFE CONDITION	12.5000642261462	0.7754368782043457
6	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	570	0.001	WOLFE CONDITION	12.5000191561943	0.8354811668395996
7	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	760	0.0001	WOLFE CONDITION	12.5000002208559	1.1157660484313965
8	x^2 + (2x - 4y)^2 + (x-5)^2	12.5	(0, 0)	570	0.001	WOLFE CONDITION	12.5000191561943	0.826740026473999

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	WOLFE CONDITION	VALUE	TIME
9	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	570	0.001	WOLFE CONDITION	-0.999989272531215	0.7070770263671875
10	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	779	0.0001	WOLFE CONDITION	-0.999999920845251	0.8740830421447754
11	x^2 + y^2 - xy + 2x - 4y + 3	-1	(1, 5)	570	0.001	WOLFE CONDITION	-0.999989272531215	0.7347612380981445
12	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	570	0.001	WOLFE CONDITION	-0.999986207540131	0.6615831851959229
13	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	760	0.0001	WOLFE CONDITION	-0.999999840983764	0.8069820404052734
14	x^2 + y^2 - xy + 2x - 4y + 3	-1	(3, 2)	570	0.001	WOLFE CONDITION	-0.999986207540131	0.644089937210083
15	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	532	0.001	WOLFE CONDITION	-0.999985034223234	0.5893082618713379
16	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	741	0.0001	WOLFE CONDITION	-0.999999889572059	0.859961986541748
17	x^2 + y^2 - xy + 2x - 4y + 3	-1	(0, 0)	532	0.001	WOLFE CONDITION	-0.999985034223234	0.5778319835662842

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	WOLFE CONDITION	VALUE	TIME
18	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	589	0.001	WOLFE CONDITION	0.00156927543385221	0.7379400730133057
19	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	798	0.0001	WOLFE CONDITION	1.15792089237786e-5	1.035240888595581
20	(1 - x)^2 + 100(y - x^2)^2	0	(1, 5)	589	0.001	WOLFE CONDITION	0.00156927543385221	0.6879091262817383
21	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	1653	0.001	WOLFE CONDITION	2.75995620535741e-6	3.429793119430542
22	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	1843	0.0001	WOLFE CONDITION	3.24648406195806e-8	3.8032760620117188
23	(1 - x)^2 + 100(y - x^2)^2	0	(3, 2)	1653	0.001	WOLFE CONDITION	2.75995620535741e-6	3.4804608821868896
24	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	912	0.001	WOLFE CONDITION	2.72845192496434e-6	1.4229750633239746
25	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	1102	0.0001	WOLFE CONDITION	3.20906024035004e-8	1.5960609912872314
26	(1 - x)^2 + 100(y - x^2)^2	0	(0, 0)	912	0.001	WOLFE CONDITION	2.72845192496434e-6	1.3797399997711182

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	WOLFE CONDITION	VALUE	TIME
27	x^4 + y^4 - 4xy	-2	(1, 5)	836	0.001	WOLFE CONDITION	-1.99992655420777	0.9237251281738281
28	x^4 + y^4 - 4xy	-2	(1, 5)	1045	0.0001	WOLFE CONDITION	-1.99999945414909	1.199570894241333
29	x^4 + y^4 - 4xy	-2	(1, 5)	836	0.001	WOLFE CONDITION	-1.99992655420777	0.9101059436798096
30	x^4 + y^4 - 4xy	-2	(3, 2)	703	0.001	WOLFE CONDITION	-1.99993549465423	0.8454482555389404
31	x^4 + y^4 - 4xy	-2	(3, 2)	912	0.0001	WOLFE CONDITION	-1.99999952068299	1.0268151760101318
32	x^4 + y^4 - 4xy	-2	(3, 2)	703	0.001	WOLFE CONDITION	-1.99993549465423	0.769355058670044
33	x^4 + y^4 - 4xy	-2	(0, 0)	3	0.001	WOLFE CONDITION	0.0	0.0027921199798583984
34	x^4 + y^4 - 4xy	-2	(0, 0)	3	0.0001	WOLFE CONDITION	0.0	0.0021970272064208984
35	x^4 + y^4 - 4xy	-2	(0, 0)	3	0.001	WOLFE CONDITION	0.0	0.002580881118774414

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	WOLFE CONDITION	VALUE	TIME
36	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	627	0.001	WOLFE CONDITION	0.000134040747617467	1.0800039768218994
37	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	817	0.0001	WOLFE CONDITION	1.54643933404689e-6	1.3994660377502441
38	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(1, 5)	627	0.001	WOLFE CONDITION	0.000134040747617467	1.0904700756072998
39	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	613	0.001	WOLFE CONDITION	3.40144523291417e-5	1.0817320346832275
40	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	803	0.0001	WOLFE CONDITION	3.92156054016982e-7	1.3586997985839844
41	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(3, 2)	613	0.001	WOLFE CONDITION	3.40144523291417e-5	1.0233683586120605
42	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	3	0.001	WOLFE CONDITION	0	0.004296064376831055
43	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	3	0.0001	WOLFE CONDITION	0	0.003293752670288086
44	x^2 * (cos(x) + 2) + y^2 * (sin(y) + 12)	0	(0, 0)	3	0.001	WOLFE CONDITION	0	0.003001689910888672

Дополнительное задание 1.2

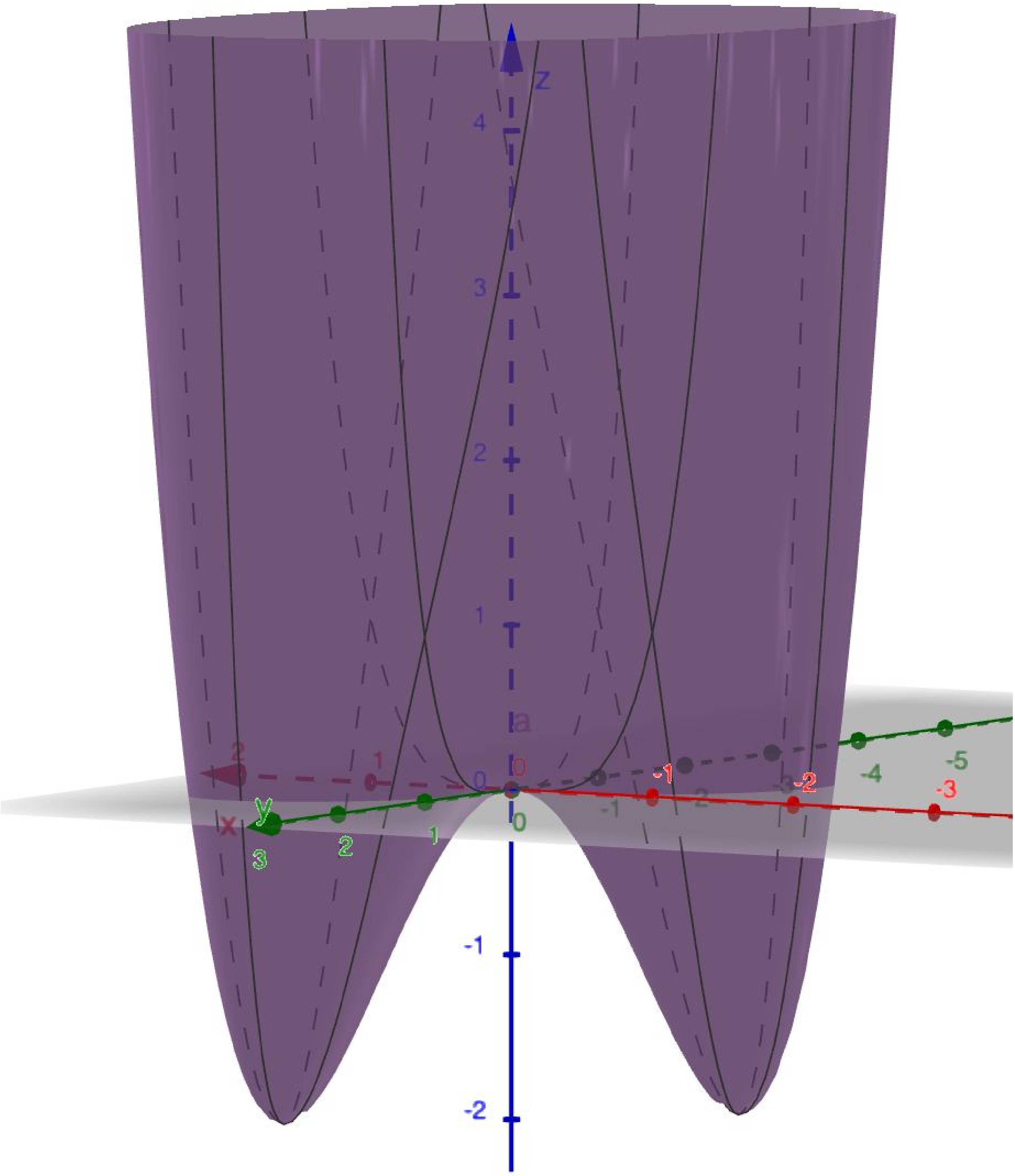
В качестве квазиньютоновского метода мы реализовали BFGS, пример и принцип его работы приведены в части, посвященной этому методу.

Дополнительное задание 2.1

В качестве примера приведем функцию $f_4(x, y) = x^4 + y^4 - 4xy$.

Данная функция имеет точки локального минимума: $(-1, -1, -2), (1, 1, -2)$ и седловую точку: $(0, 0, 0)$.

В силу того, что в седловой точке градиент равен 0, метод Ньютона останавливается, в результате чего не приходит в минимум.



№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	NEWTON CG METHOD	VALUE	TIME
27	x^4 + y^4 - 4xy	-2	(1, 5)	10	0.001	NEWTON CG METHOD	-2.0	0.00041604042053222656
28	x^4 + y^4 - 4xy	-2	(1, 5)	10	0.0001	NEWTON CG METHOD	-2.0	0.000415802001953125
29	x^4 + y^4 - 4xy	-2	(1, 5)	10	0.001	NEWTON CG METHOD	-2.0	0.00041294097900390625
30	x^4 + y^4 - 4xy	-2	(3, 2)	8	0.001	NEWTON CG METHOD	-2.0	0.0003399848937988281
31	x^4 + y^4 - 4xy	-2	(3, 2)	8	0.0001	NEWTON CG METHOD	-2.0	0.00033473968505859375
32	x^4 + y^4 - 4xy	-2	(3, 2)	8	0.001	NEWTON CG METHOD	-2.0	0.0003387928009033203
33	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.001	NEWTON CG METHOD	0.0	6.723403930664062e-05
34	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.0001	NEWTON CG METHOD	0.0	6.604194641113281e-05
35	x^4 + y^4 - 4xy	-2	(0, 0)	1	0.001	NEWTON CG METHOD	0.0	6.127357482910156e-05

Очевидно, что и обычный градиентный спуск не будет работать в данной ситуации.

№	FUNCTION	GLOBAL_MIN	INIT_POINT	ITERATIONS	EPS	CHANGING STEP	VALUE
9	x**4 - 4*x*y + y**4	-2	(1, 5)	4	0.001	TERNARY RATE METHOD	-1.9998939749962212
10	x**4 - 4*x*y + y**4	-2	(1, 5)	8	1e-06	TERNARY RATE METHOD	-1.9999999842087912
11	x**4 - 4*x*y + y**4	-2	(1, 5)	7	1e-05	TERNARY RATE METHOD	-1.9999998573195912
12	x**4 - 4*x*y + y**4	-2	(3, 2)	5	0.001	TERNARY RATE METHOD	-1.999951838644537 2
13	x**4 - 4*x*y + y**4	-2	(3, 2)	8	1e-06	TERNARY RATE METHOD	-1.9999999573441825
14	x**4 - 4*x*y + y**4	-2	(3, 2)	7	1e-05	TERNARY RATE METHOD	-1.9999995566301787
15	x**4 - 4*x*y + y**4	-2	(0, 0)	2	0.001	TERNARY RATE METHOD	0.0
16	x**4 - 4*x*y + y**4	-2	(0, 0)	2	1e-06	TERNARY RATE METHOD	0.0
17	x**4 - 4*x*y + y**4	-2	(0, 0)	2	1e-05	TERNARY RATE METHOD	0.0

Дополнительное задание 2.2

В этом пункте используется та же функция, что и в предыдущем. Для того, чтобы попасть в разные точки мы должны были попасть в разные "окрестности притяжения точек". В нашем случае функция работащая с условием Вольфе пришла в седловую точку, а та, что работала на тернарном поиске нашла точку минимума.

```
print("Result for Newton with step changing ternary:")
print(newton( initial_dot: (0, 0.0002),  eps: 0.0001, FUNCTION.FUNC_4, REGIME.CHANGING_STEP_TERNARY, CONSTANT_STEPS[0])[0])
print("Result for Newton with wolfe condition:")
print(newton( initial_dot: (0, 0.0002),  eps: 0.0001, FUNCTION.FUNC_4, REGIME.WOLFE_CONDITION, CONSTANT_STEPS[0])[0])
```

```
Result for Newton with step changing ternary:
-2.00000000000000
Result for Newton with wolfe condition:
2.45760000000002e-16
```

Сравнение результатов NELDER-MEAD, GRADIENT DESCENT(ternary), NEWTON(scipy) и QUASI-NEWTON

Подопытные:
1) x**2 + (x - 5)**2 + (2*x - 4*y)**2|
2) x**2 - x*y + 2*x + y**2 - 4*y + 3
3) (1 - x) ** 2 + 100 * (y - x ** 2) ** 2

Eps = 0.001
Начальная точка (1, 5).
Порядок погрешности результата:

	NELDER-MEAD	GRADIENT DESCENT	NEWTON	QUASI-NEWTON
1	10**(-9)	10**(-7)	0	0
2	10**(-1)	10**(-1)	0	0
3	-	-	10**(-13)	10**(-11)

Количество итераций алгоритма:

	NELDER-MEAD	GRADIENT DESCENT	NEWTON	QUASI-NEWTON
1	53	9	5	2
2	45	7	3	3
3	-	-	36	23

Начальная точка (3, 2).
Порядок погрешности результата:

	NELDER-MEAD	GRADIENT DESCENT	NEWTON	QUASI-NEWTON
1	10**(-9)	10**(-4)	0	0
2	10**(-1)	10**(-1)	0	0
3	-	-	10**(-16)	10**(-10)

Количество итераций алгоритма:

	NELDER-MEAD	GRADIENT DESCENT	NEWTON	QUASI-NEWTON
1	37	6	5	2
2	48	6	3	3
3	-	-	28	55

Слабости метода Ньютона
Во-первых, мы имеем квадратичную скорость сходимости только в окрестности оптимума. А если мы стартуем из произвольно удалённой точки, то нам, как и в случае градиентного спуска, требуется подбор шага α_k при помощи линейного поиска.

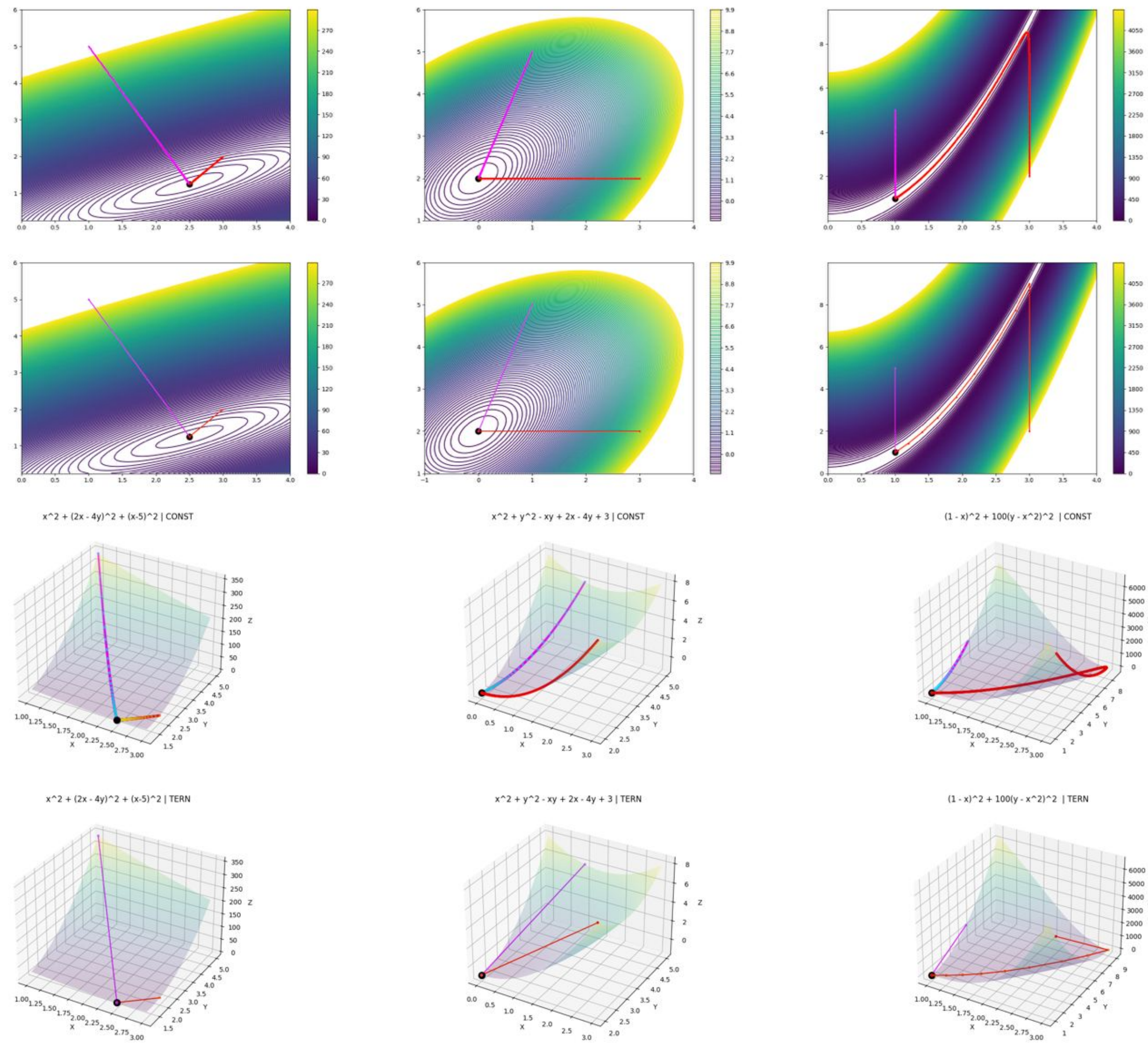
Другая проблема кроется в формуле пересчета следующей итерации: *вычисление и обращение гессиана*. Конечно, вместо обращения гессиана можно честно решать систему линейных уравнений, но асимптотика остается прежней: $O(d^3)$, а от затрат памяти на хранение матрицы $O(d^2)$ вообще некуда деться. А это значит, что, например, решать линейную регрессию с ~10000 признаками методом Ньютона попросту невозможно.

Есть и третья, малозаметная проблема: дословно метод Ньютона не работает для невыпуклых задач, поскольку $\nabla^2 f(x)$ не будет положительно опредленной и Δx перестанет быть направлением спуска. Для решения этой проблемы можно немного «подпортить» нашу аппроксимацию и рассмотреть матрицу вида $B_k = \nabla^2 f(x_k) + \Delta_k$, такую что B_k станет положительно определенной, и уже её подставлять в нашу квадратичную модель. Идея подмены гессиана на что-то более подходящее – это главная идея квазиньютоновских методов, обсуждаемых далее.

Итак, общие выводы:

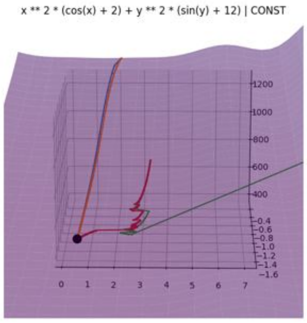
— Метод Ньютона – теоретически оптимальный метод, который автоматически улавливает кривизну функции в окрестности оптимума.
— Для размерности $d > 1000$ он уже не является эффективным, поскольку требует вычисления и хранения гессиана, а также решения системы линейных уравнений с его участием (что может быть в общем случае очень дорого).

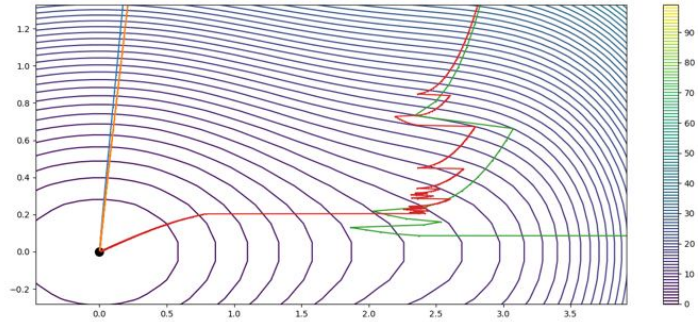
Графики

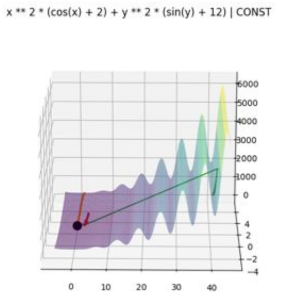


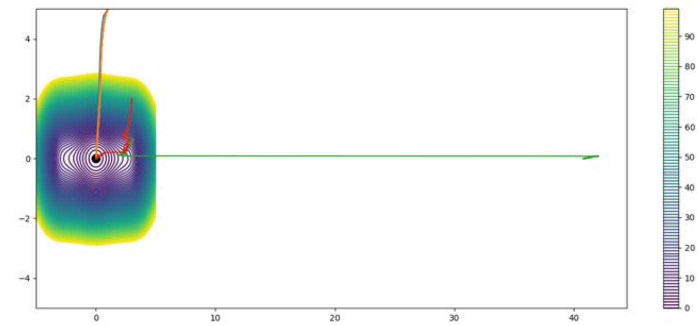
x^2 + (2x - 4y)^2 + (x-5)^2 | CONST
x^2 + y^2 - xy + 2x - 4y + 3 | CONST
(1 - x)^2 + 100(y - x^2)^2 | CONST
x^2 + (2x - 4y)^2 + (x-5)^2 | TERN
x^2 + y^2 - xy + 2x - 4y + 3 | TERN
(1 - x)^2 + 100(y - x^2)^2 | TERN

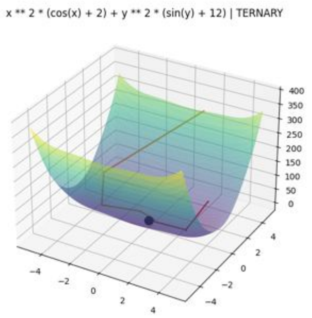
Неполиномиальные функции

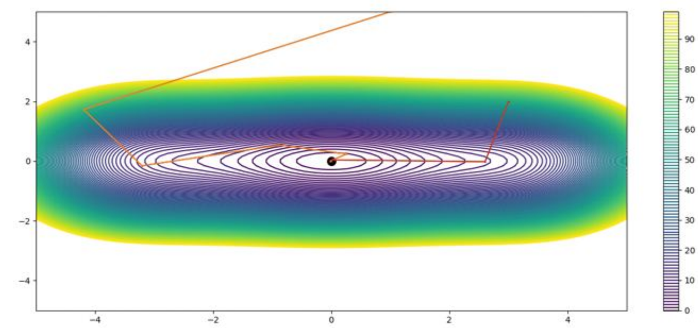












x ** 2 * (cos(x) + 2) + y ** 2 * (sin(y) + 12) | CONST