

Методы оптимизации.
Лабораторная работа №2

Дубровин Антон, Кулешов Егор, Белицкий Андрей

М3236

Команда «Аппроксимирующий многочлен»

1. Постановка задачи

Реализовать алгоритмы:

- Метод градиентного спуска
- Метод наискорейшего спуска
- Метод сопряженных градиентов

Описать, как меняется скорость сходимости, если для поиска величины шага использовать различные методы одномерного поиска.

2. Аналитическое решение

Функция $f(x_1, x_2) = 4x_1^2 + x_2^2 + 2x_1 + 4x_2 + 17$

Возьмём производную по x_1 , приравняем к нулю.

Получим $8x_1 + 2 = 0$

Возьмём производную по x_2 приравняем к нулю.

Получим $2x_2 + 4 = 0$

Решим систему, из этого получаем, что минимум функции достигается в $(x_1, x_2) = (-0.25, -2)$

3. Реализация алгоритма

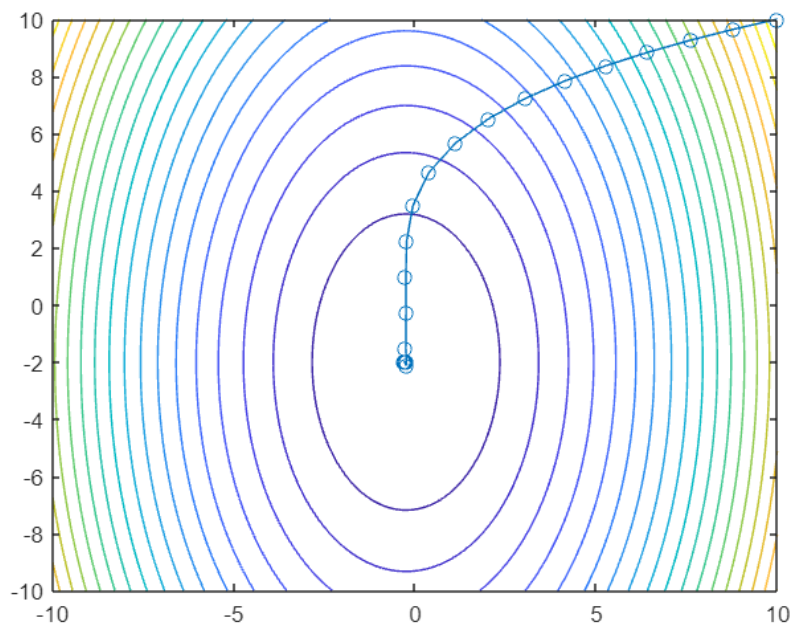
3.1. Методы

3.1.1. Градиентный спуск

- Описание: итеративный алгоритм, который на каждой итерации движется в сторону антиградиента. Для вычисления следующей точки сначала задаётся направление, обратное градиенту функции, после выбирается шаг. На каждом шаге переходим в точку, значение в которой меньше предыдущего. Значение шага передаётся с прошлой итерации. Получив шаг, мы уменьшаем его в 2 раза, пока значение в точке после шага не станет меньше текущего.
- Вычислительная схема метода: на каждой итерации $p^k = -\nabla f(x^k)$. Если $\nabla f(x^k) = 0$, то выполняется условие $(\nabla f(x^k), p^k) < 0$, и вектор p^k направлен в сторону убывания функции, следовательно $\exists a_k > 0$ такое, что $f(x^{k+1}) < f(x^k)$

Пример работы на данной функции:

Итерация	x_1	x_2
0	10	10
1	8.8003	9.6489
2	7.6104	9.266
3	6.4336	8.8443
4	5.2753	8.3745
5	4.1438	7.8433
6	2.0285	6.5165
7	1.1152	5.6631
8	0.3898	4.6451
9	-0.0594	3.4787
10	-0.2316	2.2406
11	-0.2532	0.9907
12	-0.2478	-1.5091
13	-0.2541	-2.1338
14	-0.2331	-1.9943
15	-0.2449	-1.9992
16	-0.2546	-1.9994



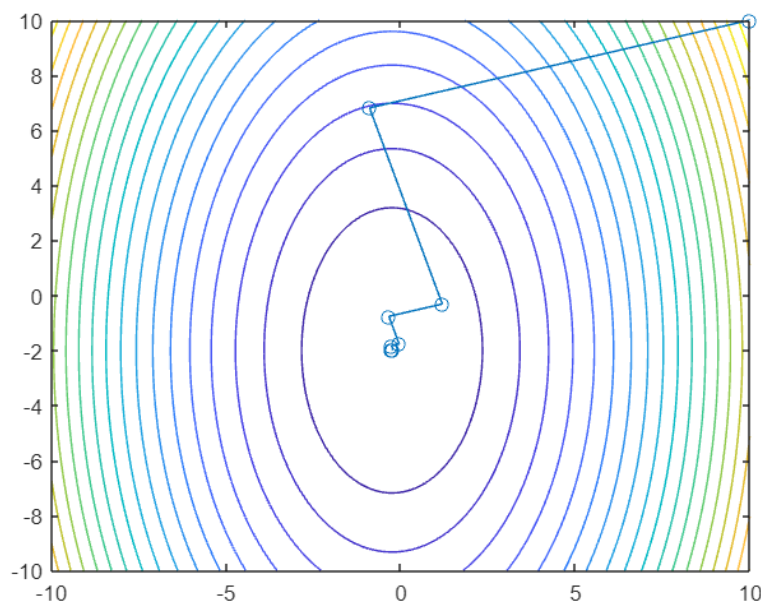
3.1.2. Наискорейший спуск

- Описание: улучшение стандартного градиентного спуска. Движение происходит в направлении антиградиента до тех пор, пока не достигнем минимума функции на этом направлении. Главная идея в оптимальном выборе шага. Теперь мы будем шагать напрямик в минимум на прямой с направлением антиградиента. Для нахождения минимума используются алгоритмы одномерной оптимизации.
- Вычислительная схема метода: на каждой итерации $p^k = -\nabla f(x^k)$. Оптимальная величина шага a_k находится с помощью метода одномерной оптимизации задачи: $\Phi_k(a_k) = f(x^k - a_k \nabla f(x^k))$,

$$\Phi_k(a_k) \rightarrow \min, a_k > 0$$

Пример работы на данной функции:

Итерация	x_1	x_2
0	10	10
1	-0.894	6.8114
2	1.1898	-0.3162
3	-0.3404	-0.7636
4	-0.0478	-1.7632
5	-0.2627	-1.8261
6	-0.2215	-1.9666
7	-0.2517	-1.9755
8	-0.2459	-1.9953
9	-0.2502	-1.9965
10	-0.2494	-1.9993
11	-0.25	-1.9995
12	-0.2499	-1.9998



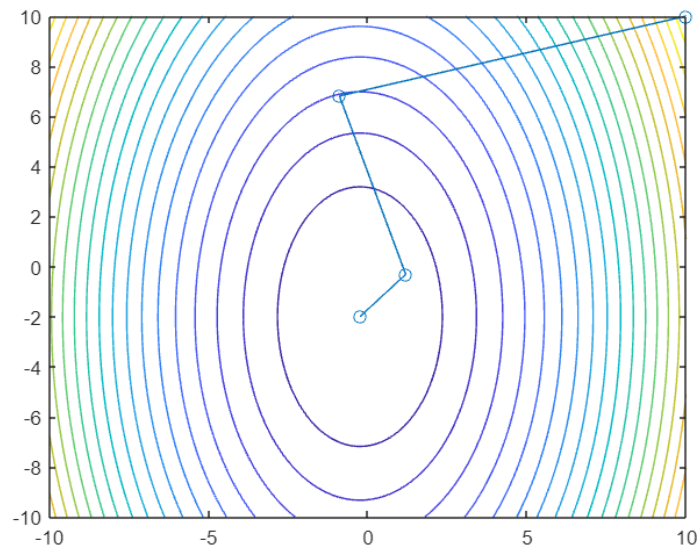
3.1.3. Метод сопряженных градиентов

- Описание: в первых двух методах в качестве направления убывания функции использовался только вектор антиградиента. Однако для плохо обусловленных задач направление антиградиента может значительно отличаться от направления к точке минимума. В отличие от предыдущих двух методов, в методе сопряженных градиентов направления спуска представляют из себя A-ортогональные вектора. Таким образом при очень точных вычислениях это позволяет находить минимум квадратичной формы не более чем за n итераций, где n - размерность пространств.

- Вычислительная схема метода:
$$a_k = \frac{\|\nabla f(x^k)\|^2}{(Ap^k, p^k)}$$
$$p^{k+1} = -\nabla f(x^k) + \beta_k p^k, k = 0, 1 \dots$$
где
$$\beta_k = \frac{\|\nabla f(x^{k+1})\|^2}{\|\nabla f(x^k)\|^2}, k = 1, 2 \dots$$

Пример работы на данной функции:

Итерация	x_1	x_2
0	10	10
1	-0.8947	6.8112
2	1.1905	-0.3134
3	-0.25	-2



3.2. Оценка скорости сходимости

Данные ниже приведены для метода наискорейшего спуска, так как методы одномерного поиска используются только в нём.

3.2.1.1. Метод дихотомии

n/k	2	5	10	20	50	100	1000
10	10	28	51	80	180	340	1931
100	11	30	52	95	192	372	2197
1000	12	31	55	101	210	390	2469
10000	13	34	57	107	219	404	2702

3.2.1.2. Метод золотого сечения

n/k	2	5	10	20	50	100	1000
10	9	27	39	76	184	339	1925
100	11	27	48	92	186	379	2237
1000	12	28	49	99	208	381	2455
10000	13	31	57	104	231	413	2635

3.2.1.3. Метод Фибоначчи

n/k	2	5	10	20	50	100	1000
10	10	25	41	70	174	329	1853
100	11	25	47	77	193	375	2114
1000	12	27	51	97	213	382	2474
10000	13	30	55	103	228	407	2678

3.2.1.4. Метод парабол

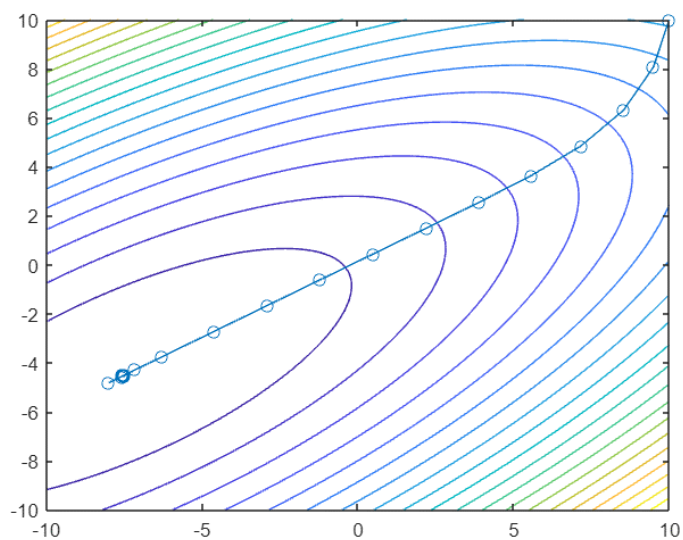
n/k	2	5	10	20	50	100	1000
10	10	24	48	75	173	328	1900
100	11	26	48	85	188	360	2098
1000	12	29	51	88	203	384	2422
10000	13	31	57	105	230	397	2631

3.2.1.5. Метод Брента

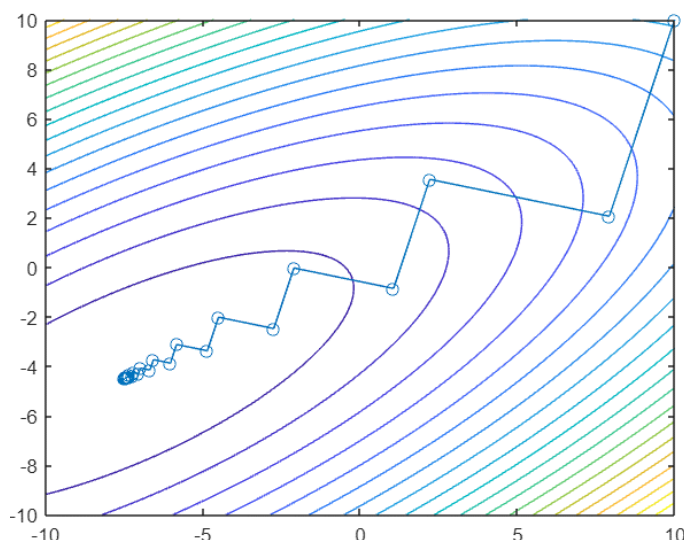
n/k	2	5	10	20	50	100	1000
10	10	22	45	90	168	327	1823
100	11	27	46	94	180	355	2110
1000	12	28	53	95	202	381	2361
10000	13	31	57	103	221	392	2575

3.3. Сравнение работ методов на разных функциях

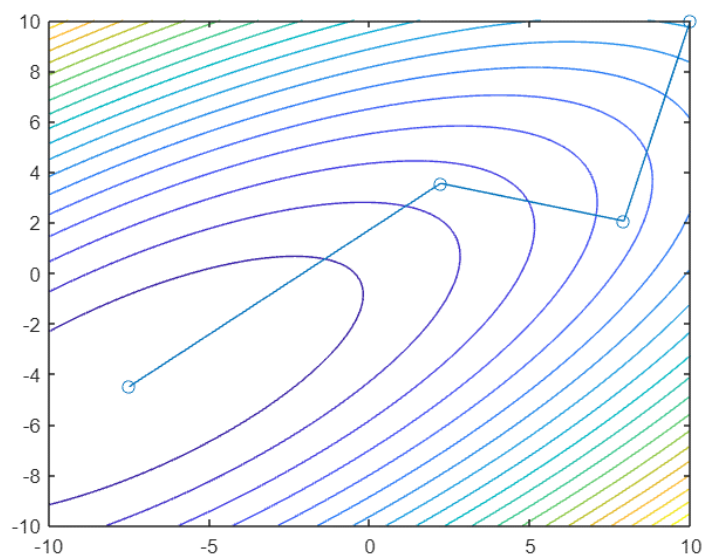
3.3.1. $f(x_1, x_2) = 2x_1^2 + x_2^2 - 2x_1x_2 + 3x_1 + 6x_2 + 1$



Градиентный спуск

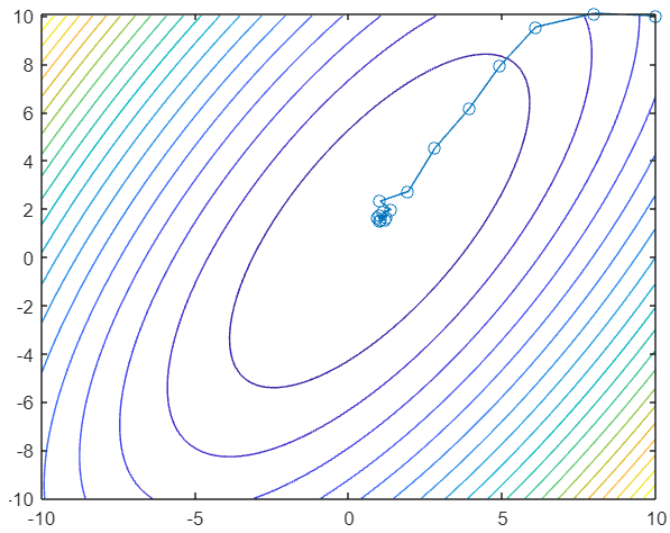


Наискорейший спуск

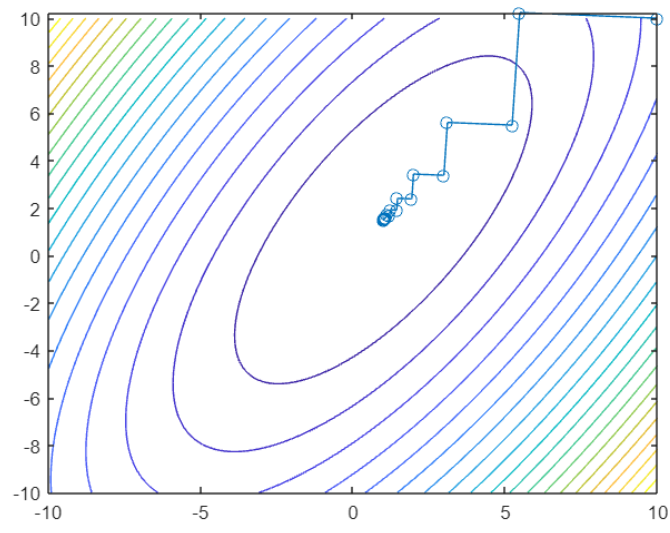


Метод сопряженных градиентов

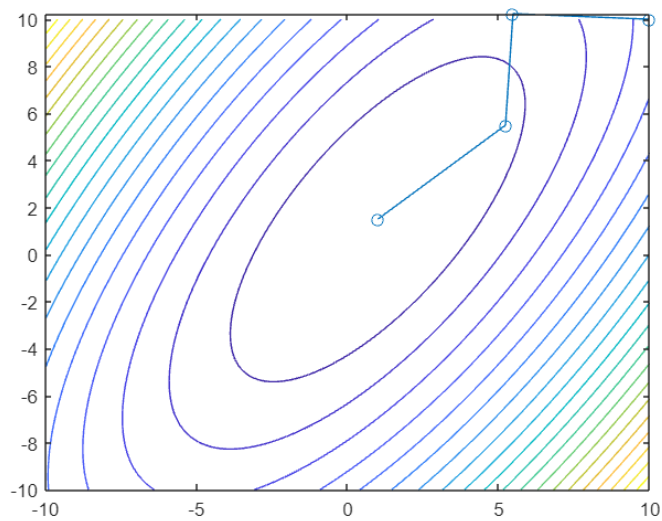
3.3.2. $f(x_1, x_2) = x_1^2 + 2x_2^2 - 2x_1x_2 - x_1 - x_2 + 4$



Градиентный спуск



1Наискорейший спуск



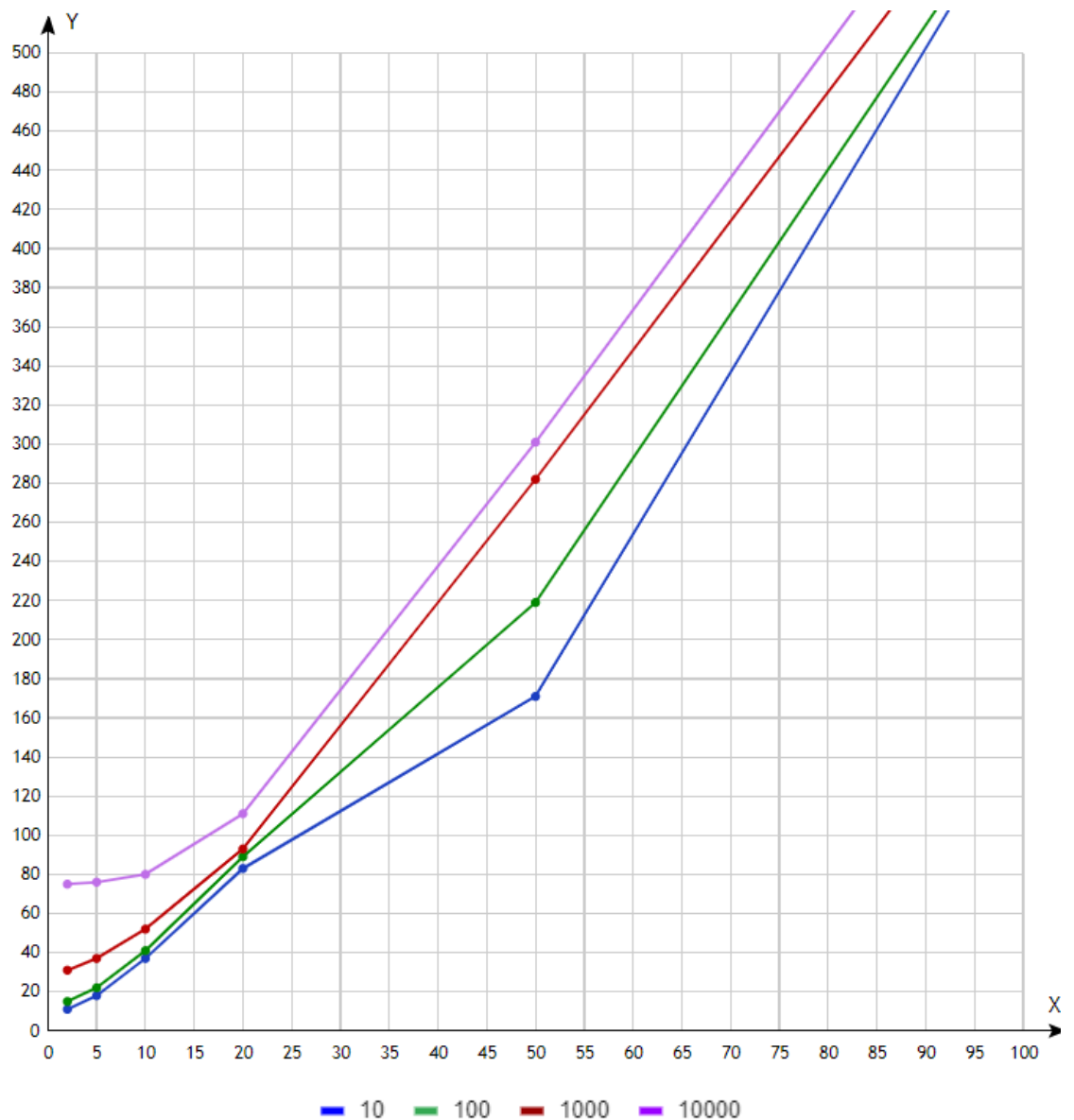
Метод сопряженных градиентов

4. Исследование зависимости числа итераций от параметров

В этом пункте показаны зависимости количества итераций от размерности пространства n и чисел обусловленности k . Все методы вычисляют минимум с точностью 5 знаков после запятой.

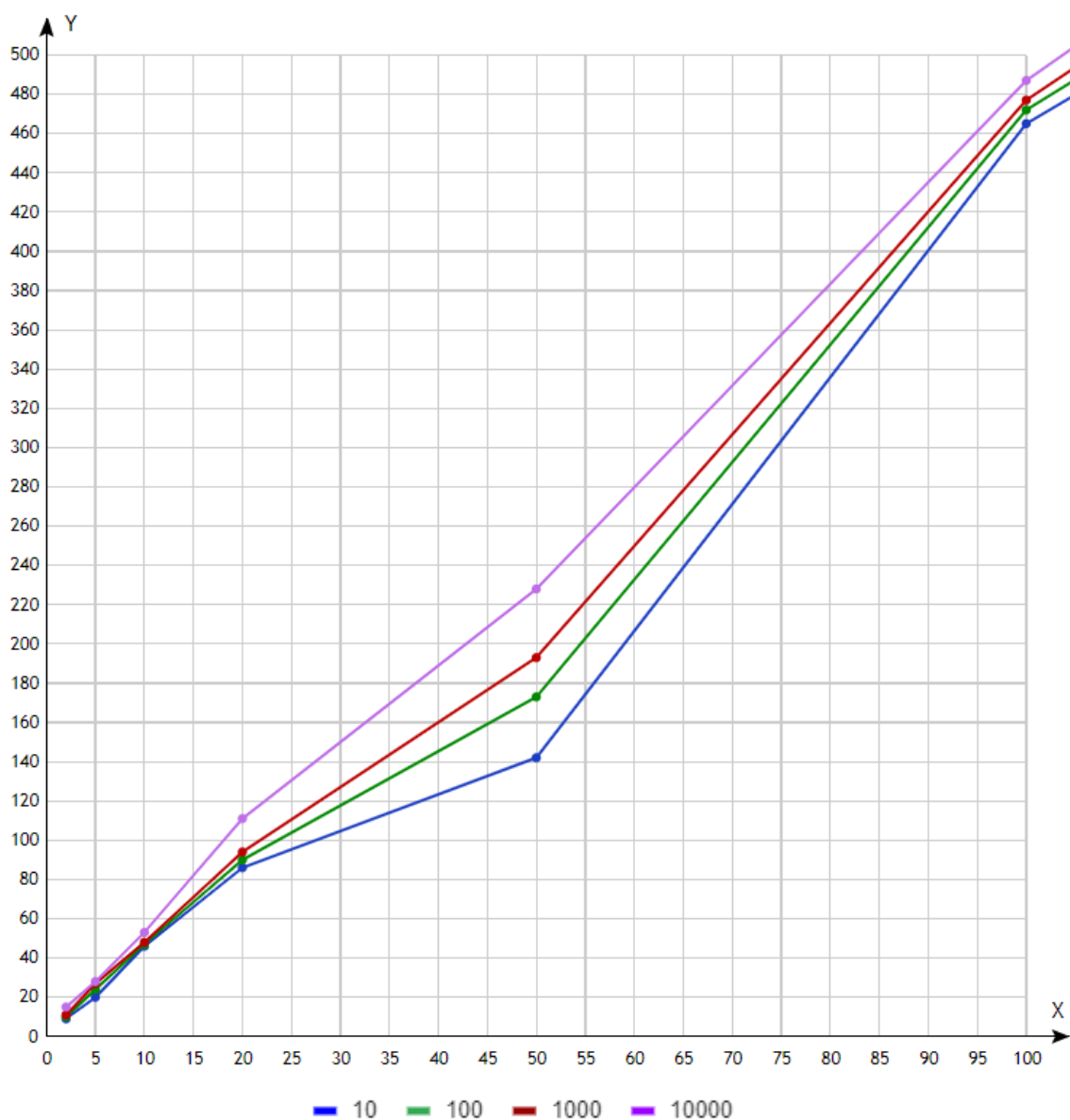
4.1. Градиентный спуск

n/k	2	5	10	20	50	100	1000
10	11	18	37	83	171	585	4712
100	15	22	41	89	219	588	4715
1000	31	37	52	93	282	612	4734
10000	75	76	80	111	301	639	4808



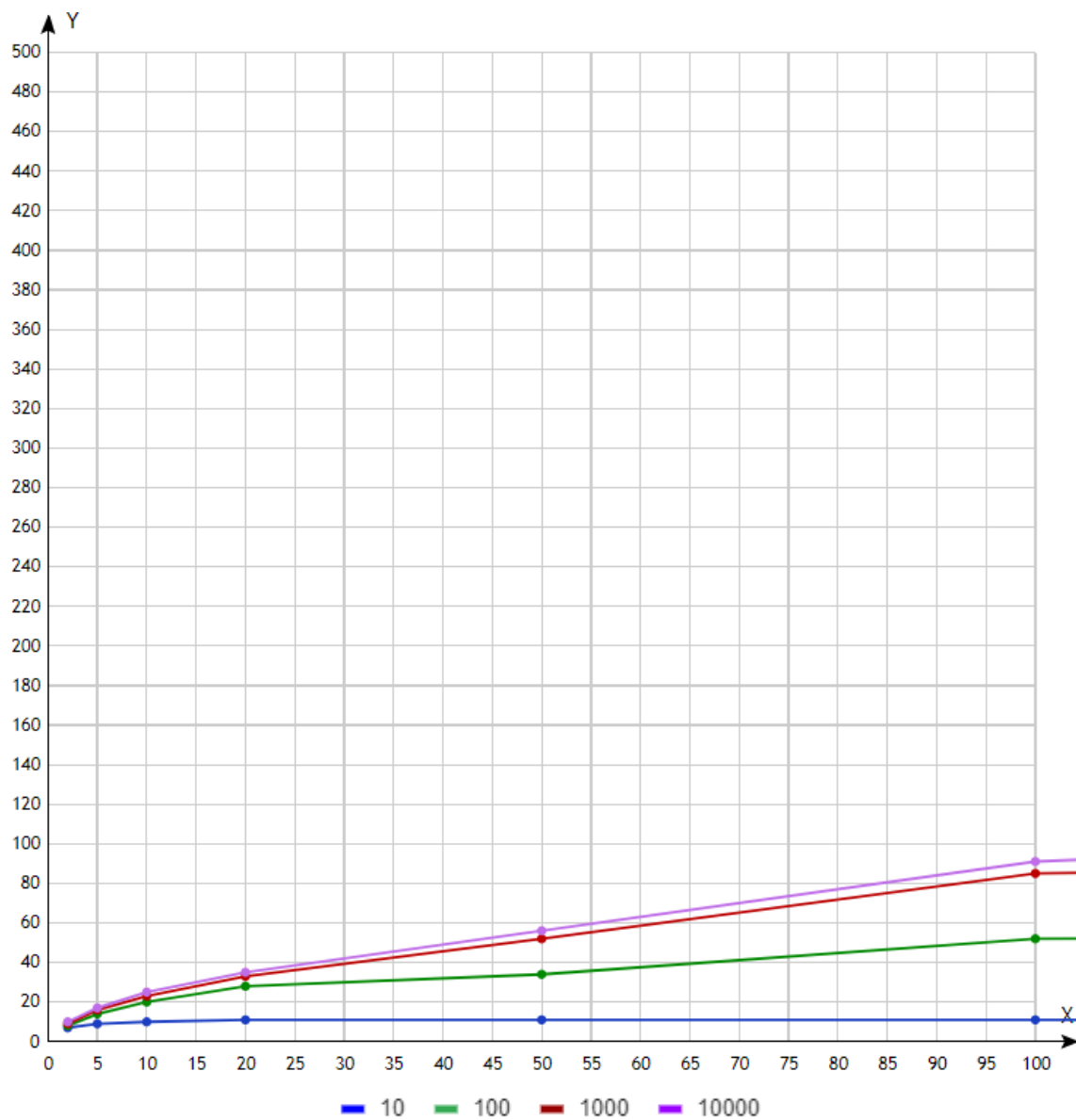
4.2.Наискорейший спуск

n/k	2	5	10	20	50	100	1000
10	9	20	46	86	142	465	3118
100	10	24	47	90	173	472	3245
1000	11	27	48	94	193	477	3488
10000	15	28	53	111	228	487	3627



4.3.Метод сопряженных градиентов

n/k	2	5	10	20	50	100	1000
10	7	9	10	11	11	11	11
100	8	14	20	28	34	52	67
1000	9	16	23	33	52	85	144
10000	10	17	25	35	56	91	277



5. Выводы

5.1. Оценка скорости сходимости

Мы пришли к выводу, что количество итераций, необходимых для сходимости, практически не зависит от выбранного одномерного метода поиска. Табличная погрешность обусловлена случайностью генерируемых квадратичных функций.

5.2. Сравнение работы методов на разных функциях

Мы практическим путем убедились в том, что существуют функции, на которых работа методов различается, и привели пару примеров, на которых видно, что метод сопряжённых градиентов показал наименьшее число итераций до сходимости.

5.3. Исследование зависимости числа итераций от параметров

Проанализировав таблицы, мы пришли к выводу, что количество итераций возрастает пропорционально размерности пространства n и обусловленности k . Скорость возрастания количества итераций функции в градиентном спуске довольно высокая, в то время как в методе сопряженных градиентов количество итераций возрастает плавно, даже при самых больших n и k метод показывает быстрый результат. Наискорейший спуск показывает средние результаты.

6. Код:

<https://github.com/SirDratuti/OptimizationMethods>

Диаграмма классов:

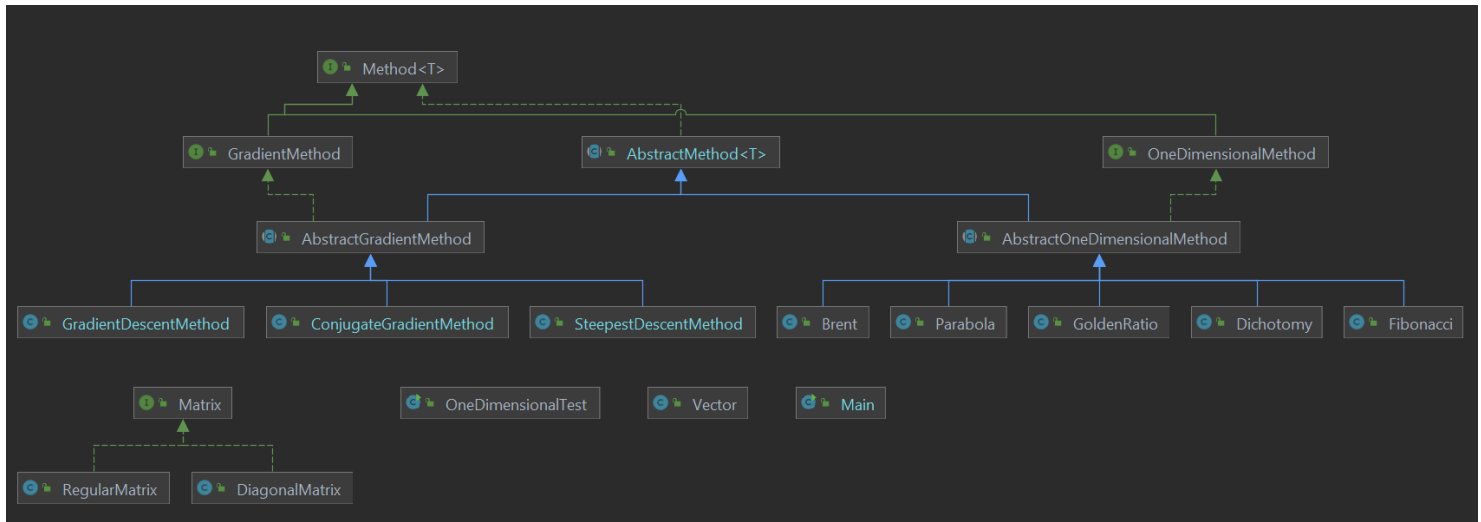
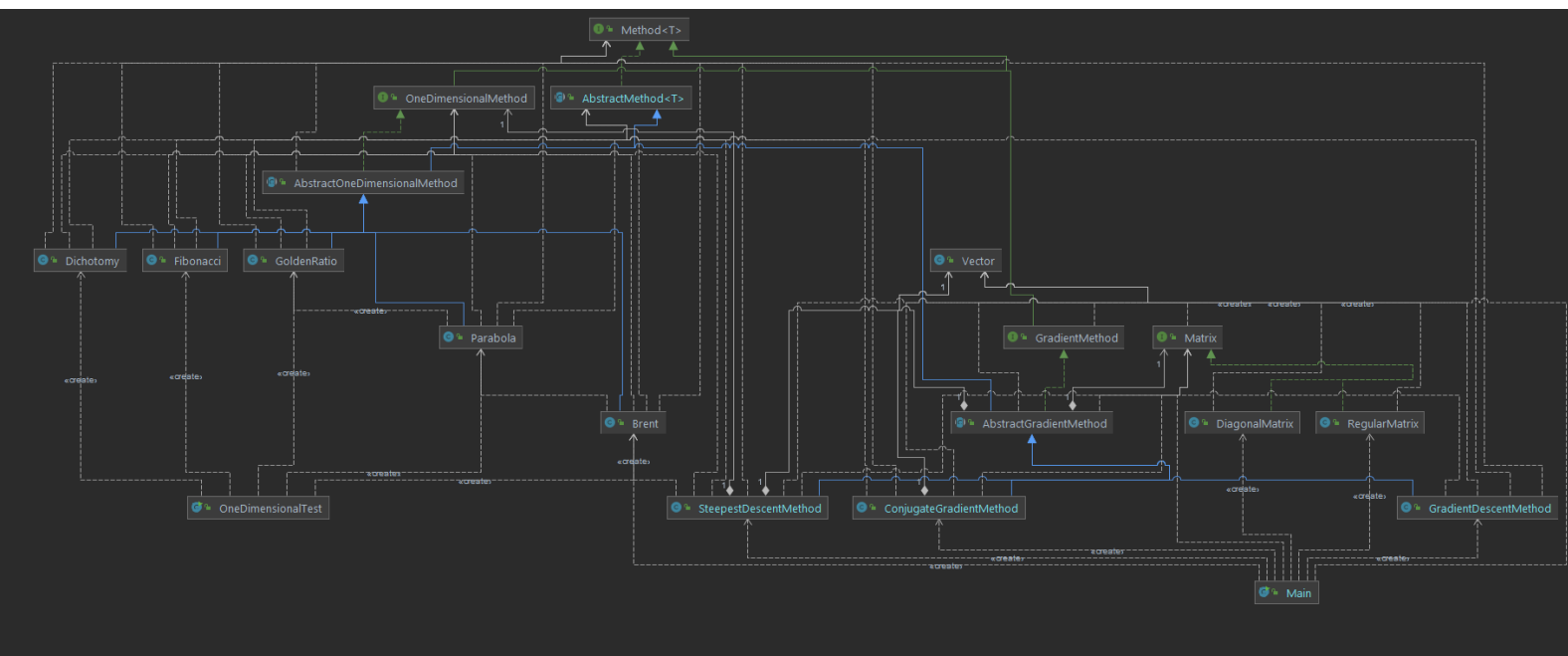


Диаграмма классов с зависимостями:



Javadoc:

<https://drive.google.com/drive/folders/1gd7JBaZfOnQjeJovSMDvyVJteowLLJ5X?usp=sharing>