фМинистерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №2 по дисциплине «**Методы оптимизации**»

Авторы: Ахметов Марсель Ринатович M3237,
Винников Глеб Вячеславович M3237,
Яценко Данил Вячеславович M3236.

Факультет: ИТиП

Группа: M3236-37

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург 2021

Цель работы:

Знакомство с основными методами многомерной оптимизации и изучение их работы на различных квадратичных функциях.

Задания:

Реализовать алгоритмы:

- метод градиентного спуска;
- метод наискорейшего спуска;
- метод сопряженных градиентов.

И проанализировать их работу на двух-трех функциях, на которых работа методов будет заметно отличаться, а также проследить зависимость числа итераций от размерности функции и ее числа обусловленности.

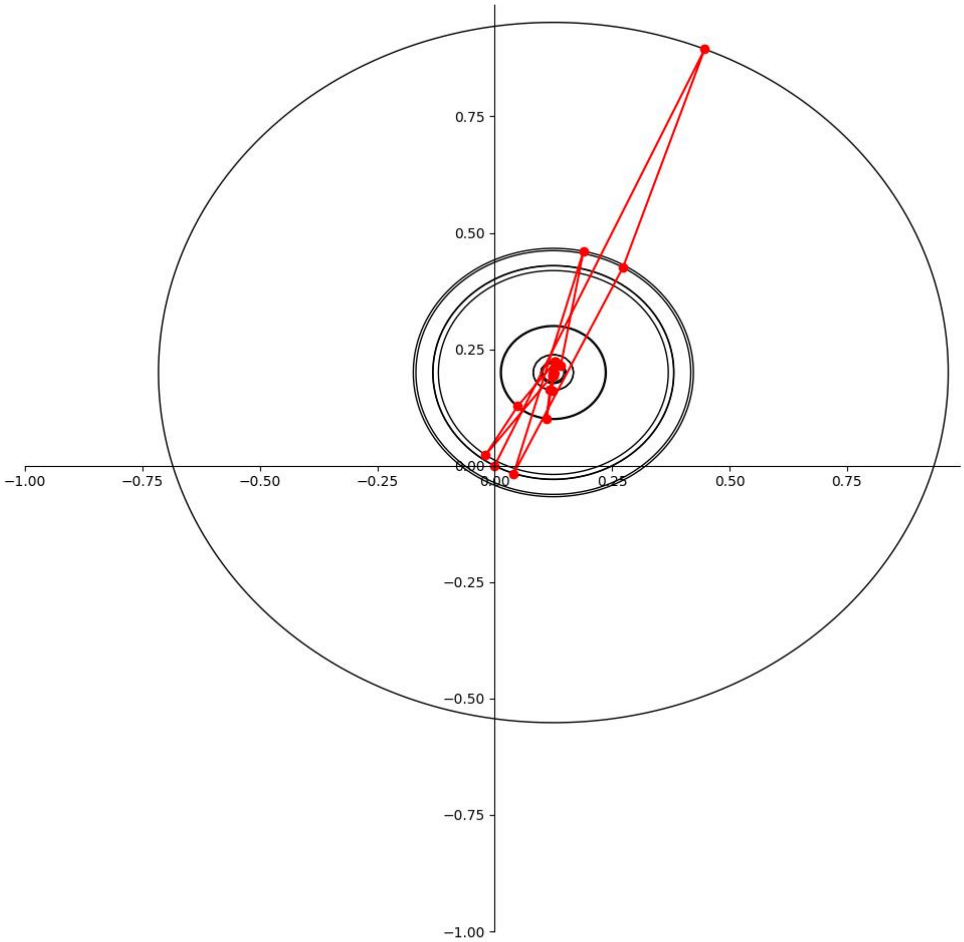
Работа алгоритмов на различных квадратичных функциях.

При исследовании функций критерием остановки было условие $/\!/ \nabla f(xk) /\!/ <= 10^{-5}$.
За начальное приближение бралась точка (0, 0).
Начальное значение λ в градиентном спуске было равно 1.

Функция 1:

$$f(x, y) = 16x^2 + 20y^2 - 4x - 8y + 5$$

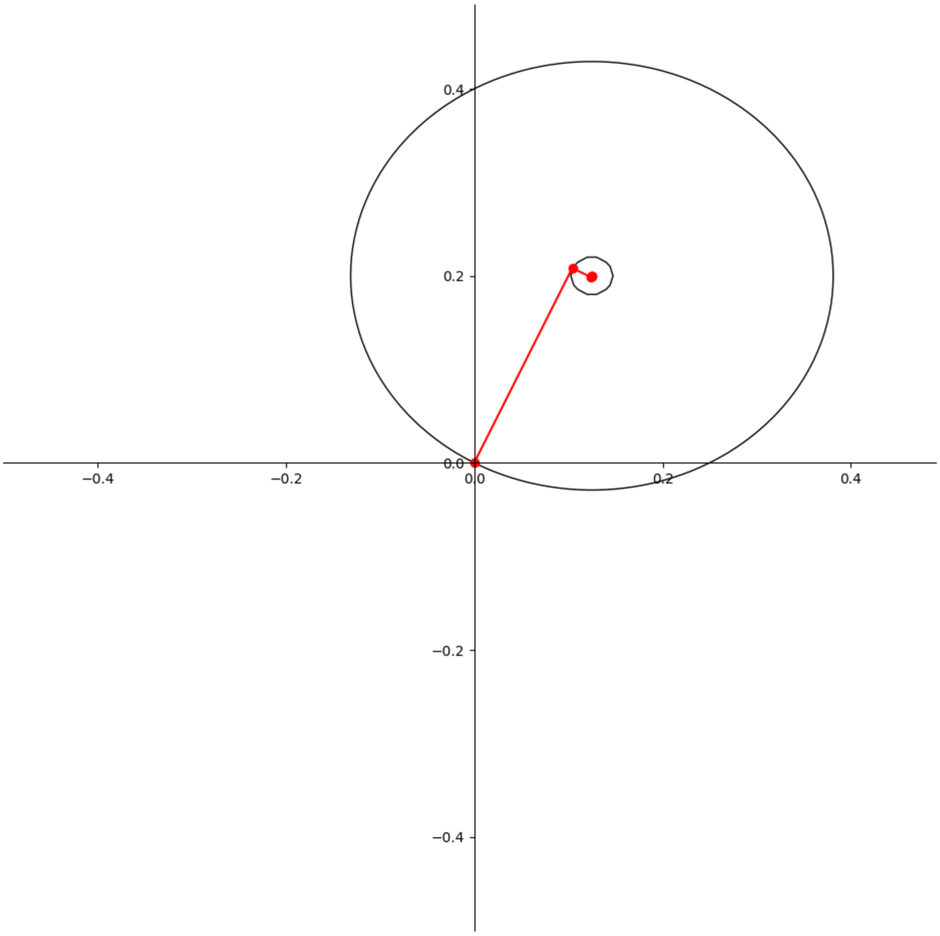
- *Метод градиентного спуска.*



На рисунке видно, что на представленной функции данный метод двигается довольно медленно к точке минимума, при этом периодически совершая скачки на более отдаленные от минимума уровни.

Итоговое количество итераций - *44*

- *Метод наискорейшего спуска.*



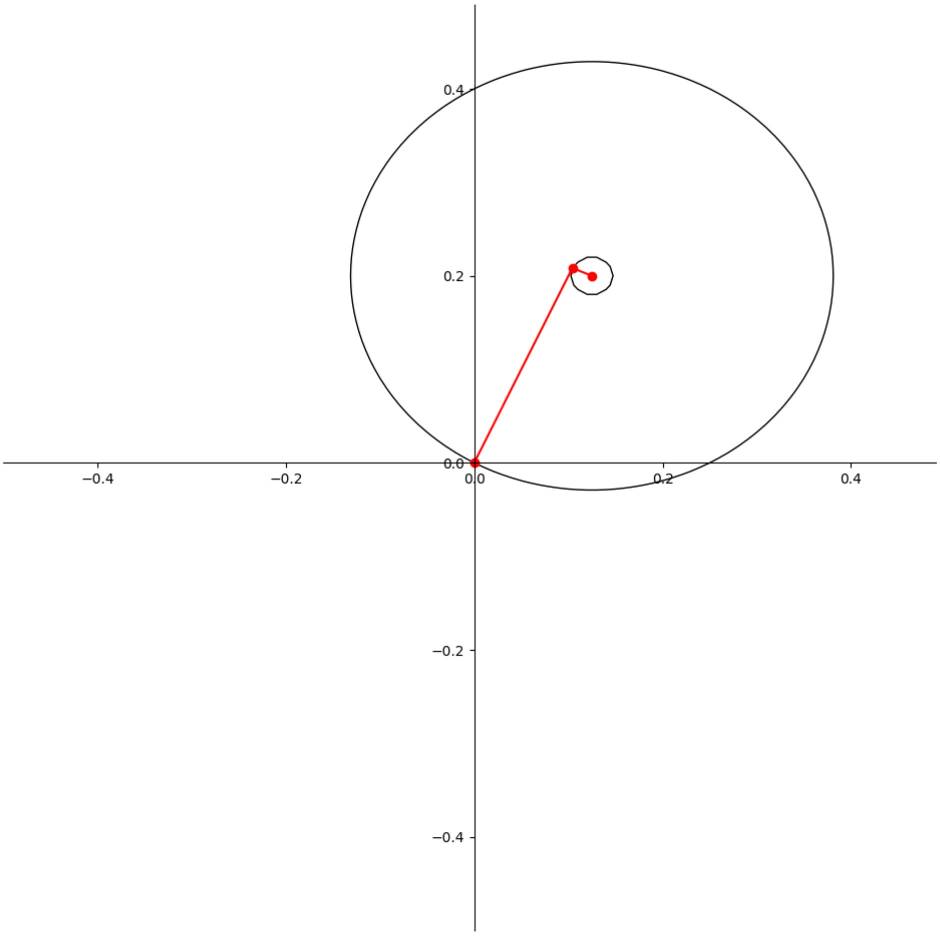
Благодаря использованию методов одномерной оптимизации данный метод довольно быстро находит точку минимума.

Количество итераций методов при использовании различных методов одномерной оптимизации.

Одномерный метод	Количество итераций
Дихотомии	6
Золотого сечения	6
Фибоначчи	6
Параболы	6
Брента	6

<u>Итоговое количество итераций: *2.*</u>

- *Метод сопряженных градиентов.*



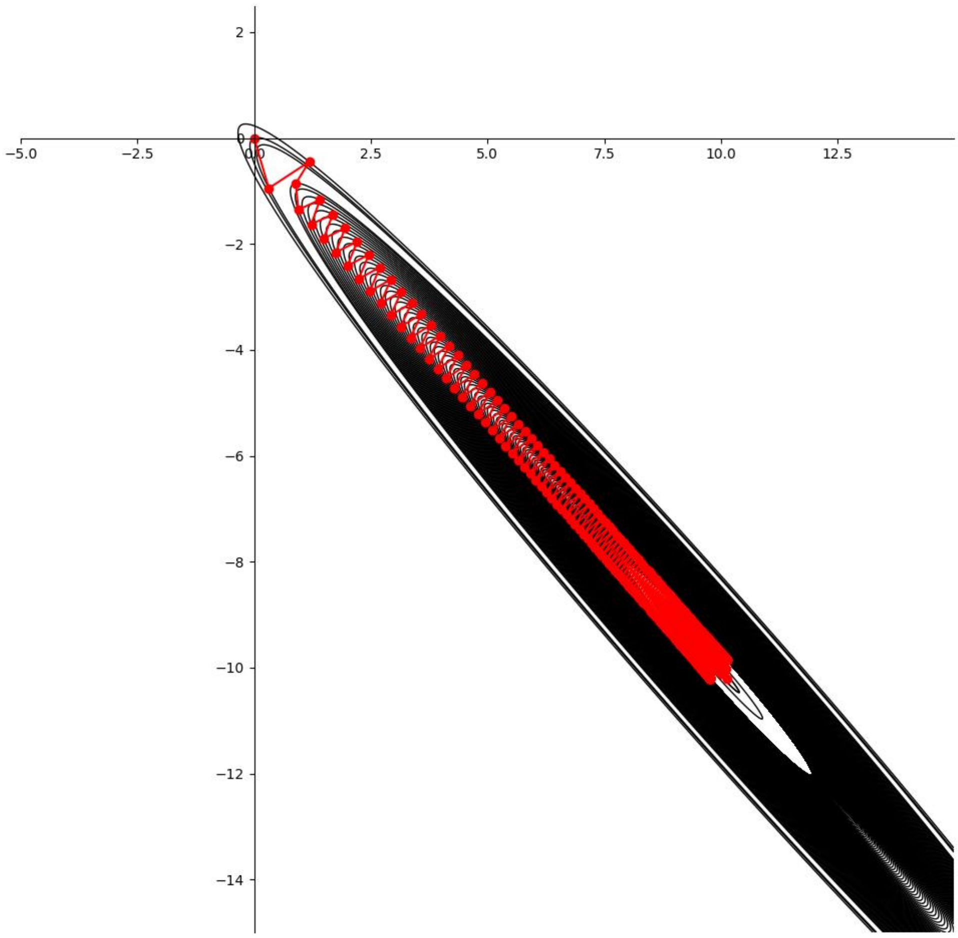
Благодаря использованию A-ортогональных направлений при спуске, алгоритм находит искомый минимум очень быстро.

<u>Итоговое количество итераций: *2.*</u>

Функция 2:

$$f(x, y) = 64x^2 + 64y^2 + 126xy - 10x + 30y + 13$$

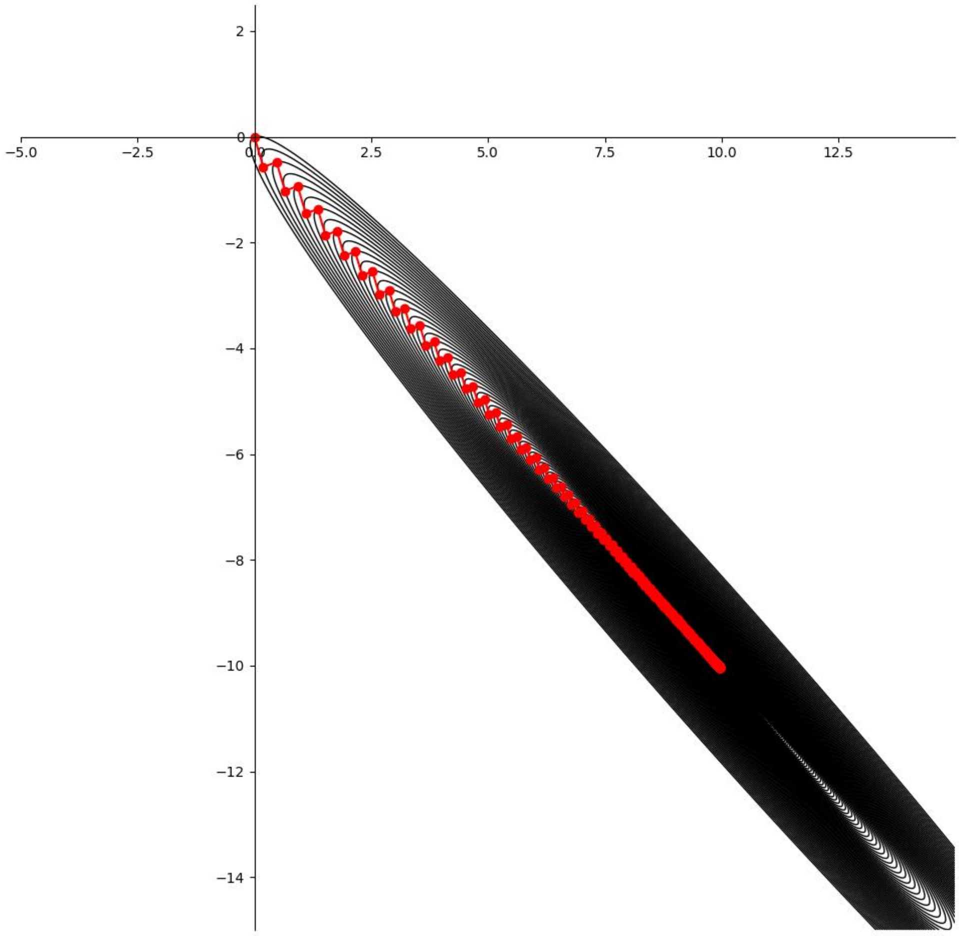
- *Метод градиентного спуска.*



Из-за овражного характера функции метод градиентного спуска совершает множество “зигзагов”, медленно приближаясь к точке минимума. Из-за того, что длина шага не меняется на протяжение многих итераций, частота “зигзагов” увеличивается.

Итоговое количество итераций - *711*

- *Метод наискорейшего спуска.*

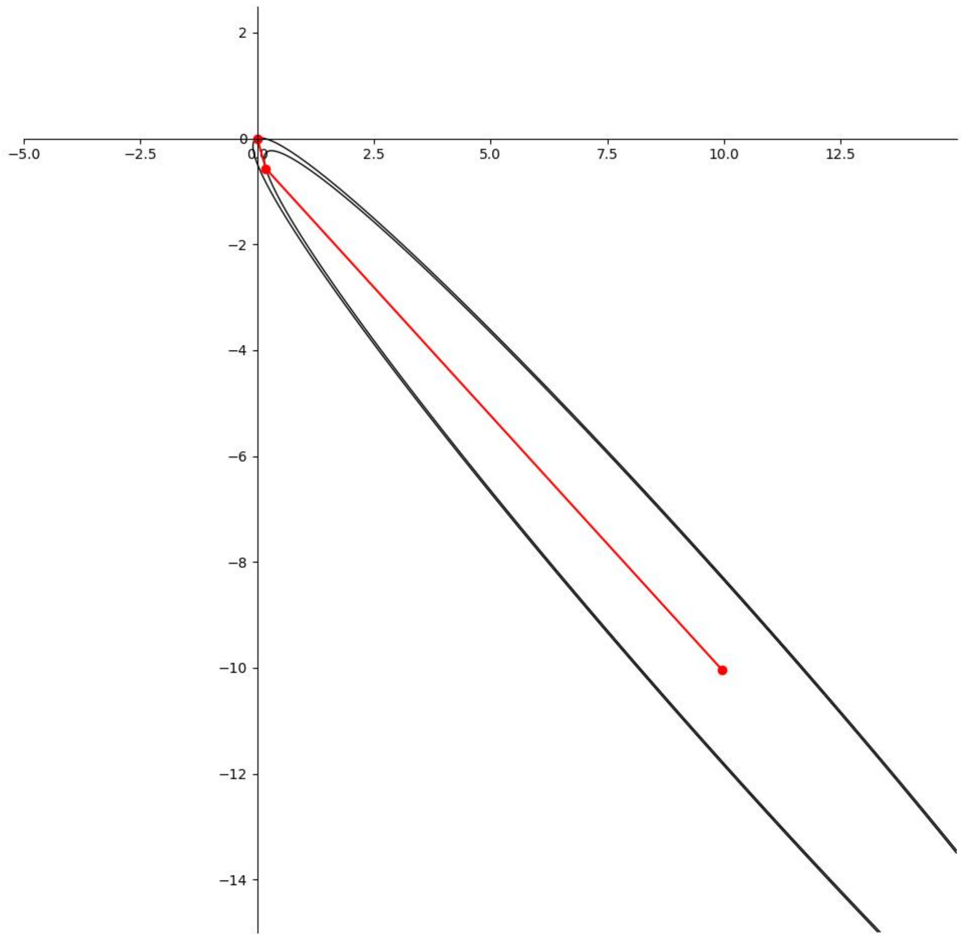


Использование методов одномерной оптимизации заметно ускоряет нахождение минимума примерно в 1.5 раза.

Количество итераций методов при использовании различных методов одномерной оптимизации...

Одномерный метод	Количество итераций
Дихотомии	448
Золотого сечения	424
Фибоначчи	420
Параболы	426
Брента	460

- *Метод сопряженных градиентов.*



Благодаря использованию А-ортогональных направлений при спуске, алгоритм находит искомый минимум очень быстро даже на овражной функции.

<u>Итоговое количество итераций: *2.*</u>

Вывод:

Для двумерных функций наиболее предпочтительным является метод сопряженных градиентов. Он лучше остальных методов показывает себя на овражных функциях и находит точку минимума за гарантированное число итераций. При этом самым невыгодным является метод градиентного спуска, хоть его и выделяет среди остальных относительная простота реализации. Метод наискорейшего спуска хоть и дает заметное ускорение, однако требует использования методов одномерной оптимизации, что усложняет его реализацию и замедляет время работы.

Исследование зависимости числа итераций, необходимое методам для сходимости, от следующих двух параметров:

а) числа обусловленности $k \geq 1$ оптимизируемой функции;
б) размерности пространства n оптимизируемых переменных

Для этого для заданных параметров n и k матрицы следующего вида:

1) a[1] = 1,
2) a[i] = random (1, k), i = 2...n-1
3) a[n] = k.

Методы оптимизации запускались в точке (1, 1..., 1), т. к. в точке (0, 0..., 0) всегда находился минимум этой матрицы и алгоритмы находили этот минимум за одну итерацию.

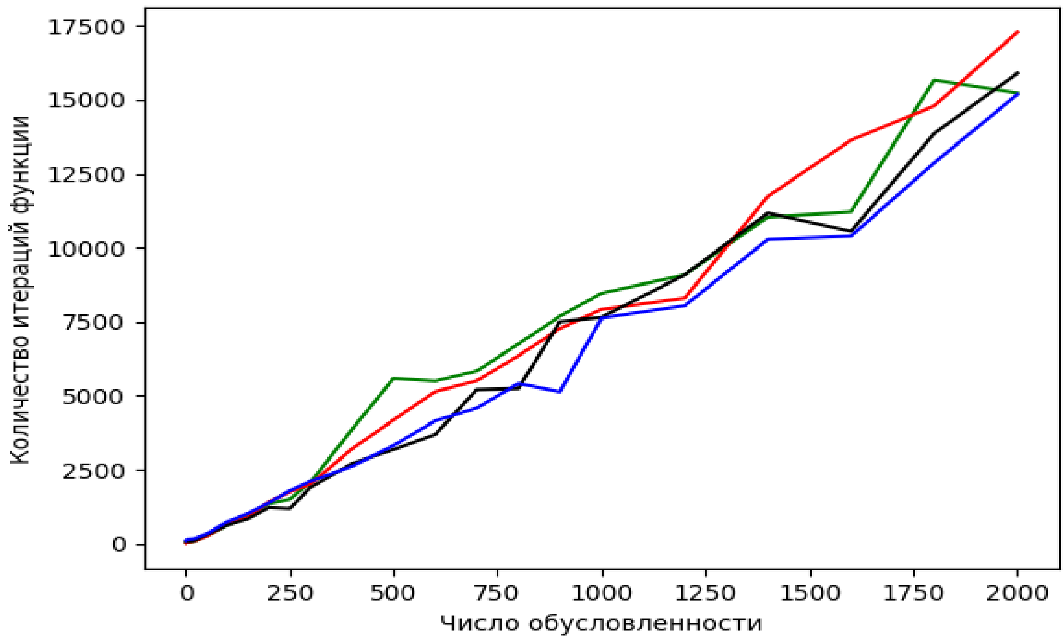
Точность - 10^{-4}

Чтобы наши данные были более общими, для каждой пары n и k генерировалось пять матриц. После чего на каждой из них мы считали необходимое число итераций и брали среднее между полученными числами.

Цвет линии - размерность:
зеленый - 10,
красный - 100,
черный - 1000,
синий - 10000.

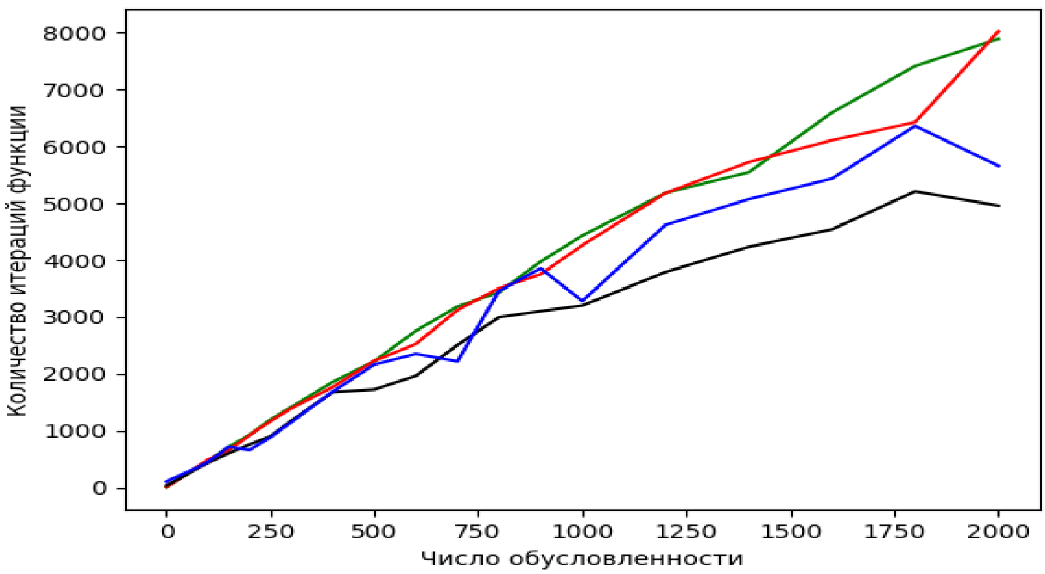
1) **Метод градиентного спуска**

Для метода градиентного спуска явно просматривается линейная зависимость числа итераций от числа обусловленности. Что удивительно, размерность не оказывает большого влияния на число итераций



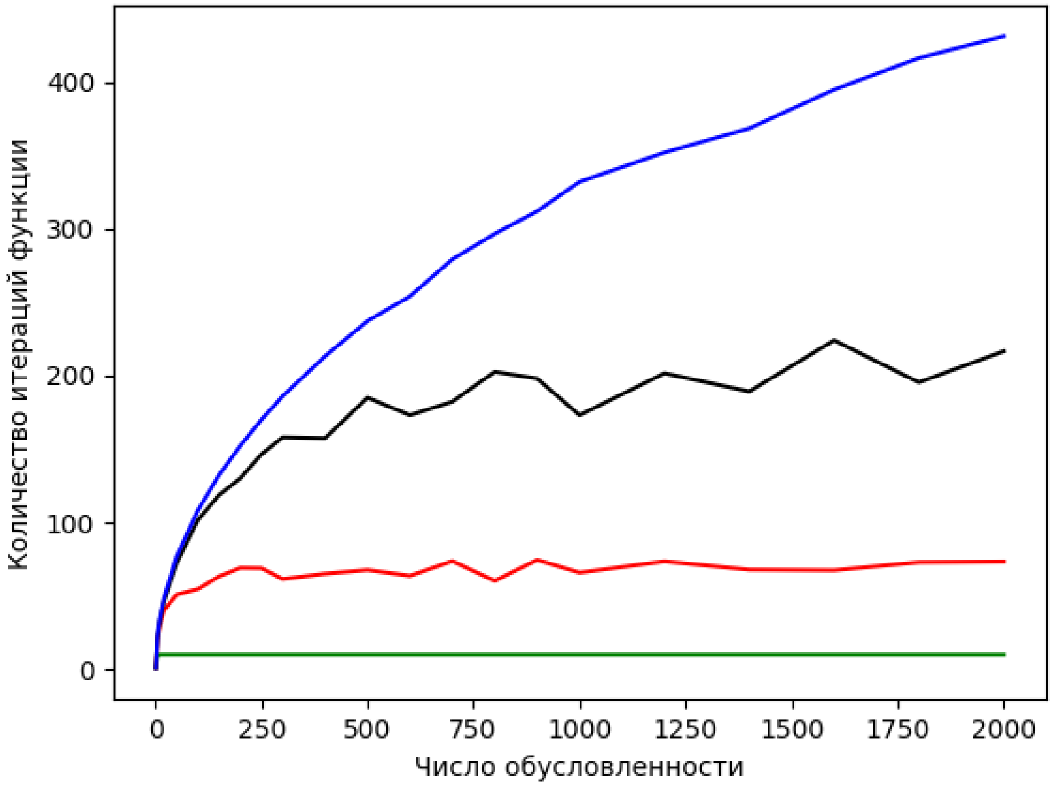
2) Метод наискорейшего спуска

Для метода наискорейшего спуска так же мы видим линейную зависимость числа итераций от числа обусловленности. По числу итераций этот метод быстрее, чем метод градиентного спуска, однако тут не учитывается время на нахождение минимума одномерной функции. На практике этот метод искал минимум на сгенерированных матрицах гораздо дольше



3) Метод сопряженных градиентов

Этот метод показал себя лучше предыдущих, на сгенерированных матрицах число итераций не превысило размерность матрицы. Число итераций логарифмически стремится к размерности генерируемой матрицы. Примечание - при другом способе генерации исследуемой матрицы не было зависимости количества итераций ни от размерности, ни от числа обусловленности.



Вывод:

С точки зрения производительности во всех планах лучше метод сопряженных градиентов. Он дает самую быструю сходимость при самых различных входных данных. При оптимизации функций общего вида метод сопряженных градиентов сходится в несколько раз быстрее остальных методов. В ходе работы мы изучили методы многомерной оптимизации и сравнили друг с другом