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

ОТЧЕТ

по лабораторной работе №3
по дисциплине «**Прикладная математика**».

Методы спуска

Автор: Кирьянов Глеб Дмитриевич

Факультет: ФИТиП

Группа: М32091

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург 2023

Для исследования методов спуска мы используем функции

$$f1 = 2 * x^2 + y^2$$

$$grad\, f1 = 4xi + 2yj$$

$$f2 = 2 * x^2 + y^2 + 2xy$$

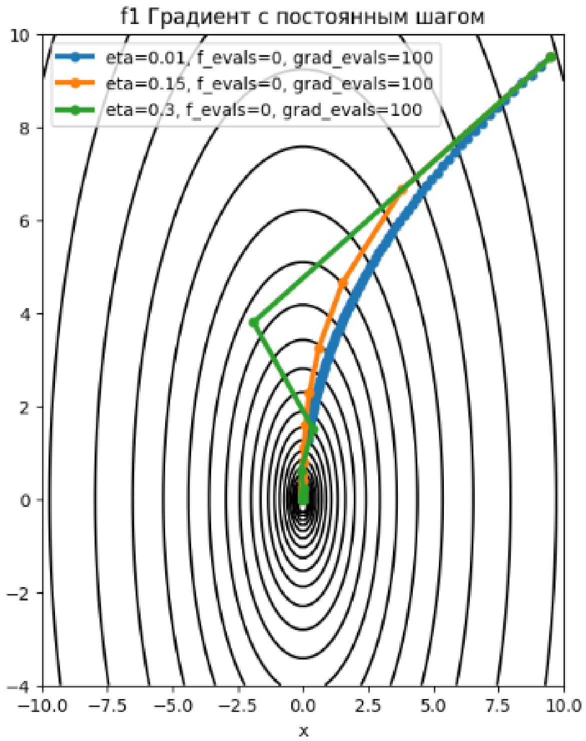
$$grad\, f2 = (4x + 2y)i + (2x + 2y)j$$

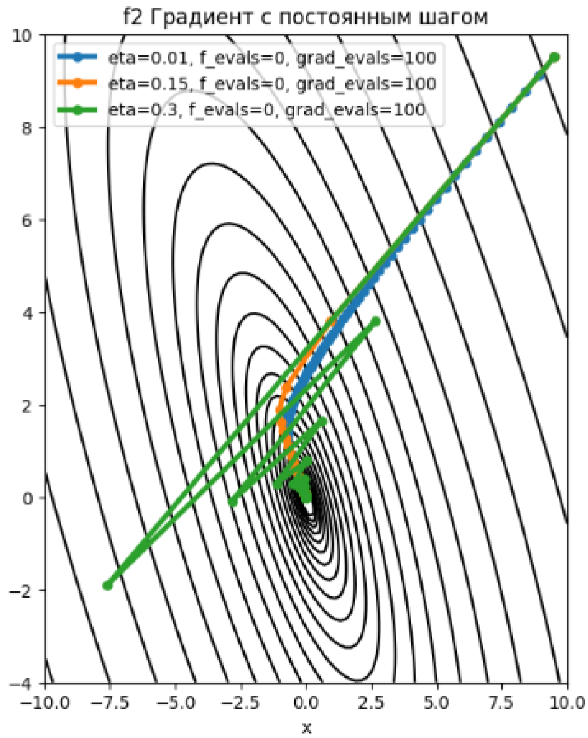
Метод градиентного спуска с постоянным шагом

Основная идея метода заключается в том, чтобы осуществлять оптимизацию в направлении наискорейшего спуска - антиградиента.

$$x^{[k+1]} = x^{[k]} - \lambda^{[k]} \nabla f(x^{[k]})$$

В качестве начала итераций выбирается произвольная точка, а величина шага задается самостоятельно и остается постоянной, но при этом метод может расходиться - при выборе слишком большого шага метод расходится, но при слишком малом сходится медленно и страдает точность метода.



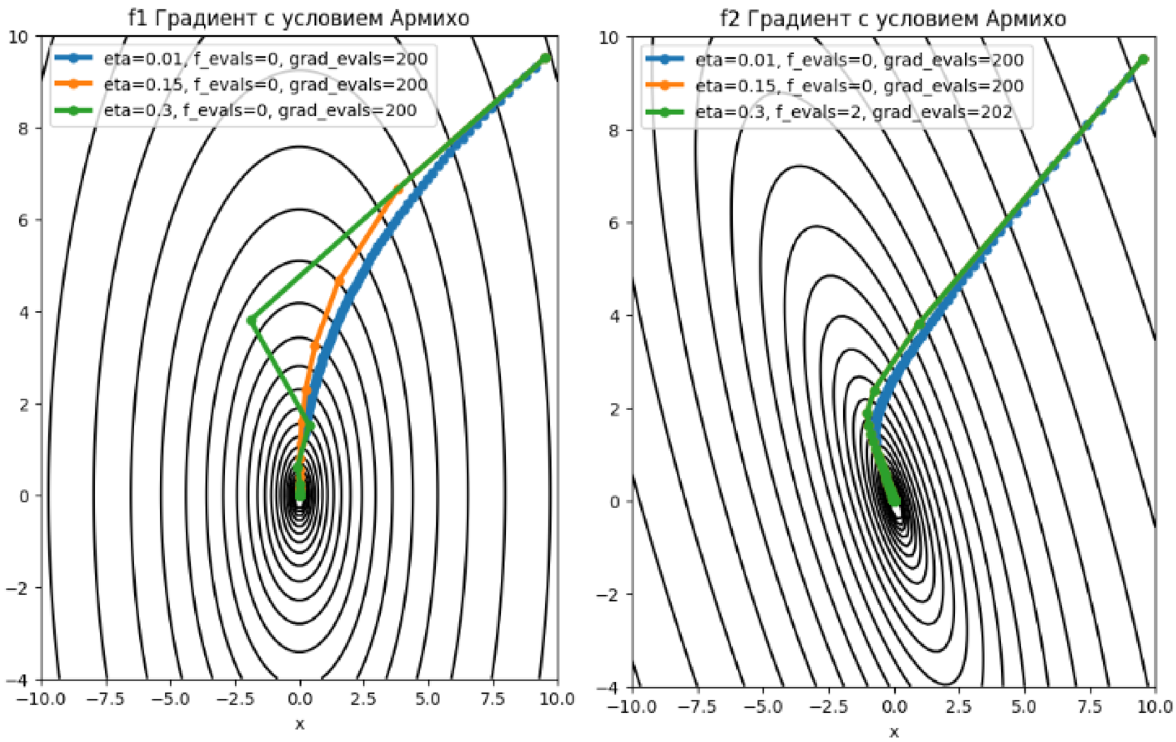


Алгоритм спуска с дроблением шага

В этом варианте градиентного метода величина шага на каждой итерации выбирается из условия выполнения неравенства

$$f(x^{[k+1]}) = f(x^{[k]} - \lambda^{[k]} f'(x^{[k]})) \leq f(x^{[k]}) - \epsilon \lambda^{[k]} \| f'(x^{[k]}) \|^2,$$

Процедуру нахождения такого $\lambda^{[k]}$ обычно оформляют так. Выбирается число и некоторый начальный шаг $\lambda^{[0]}$. Теперь для каждого k полагают $\lambda^{[k]}=\lambda^{[0]}$ и делают шаг градиентного метода. Если с таким $\lambda^{[k]}$ условие выполняется, то переходят к следующему k. Если же не выполняется, то умножают $\lambda^{[k]}$ на дробящий шаг и повторяют эту процедуру до тех пор пока неравенство не будет выполняться. В условиях теоремы 1 эта процедура для каждого k за конечное число шагов приводит к нужному $\lambda^{[k]}$.
Данный алгоритм линейно сходится, а описанный алгоритм избавляет нас от проблемы выбора $\lambda^{[k]}$ на каждом шаге и помогает избегать слишком малых или больших шагов, которые могут замедлить сходимость алгоритма. Объем вычислений разумеется возрастает, но не сильно, поскольку в основном вычислительные затраты ложатся на вычисление градиента.



Метод наискорейшего спуска

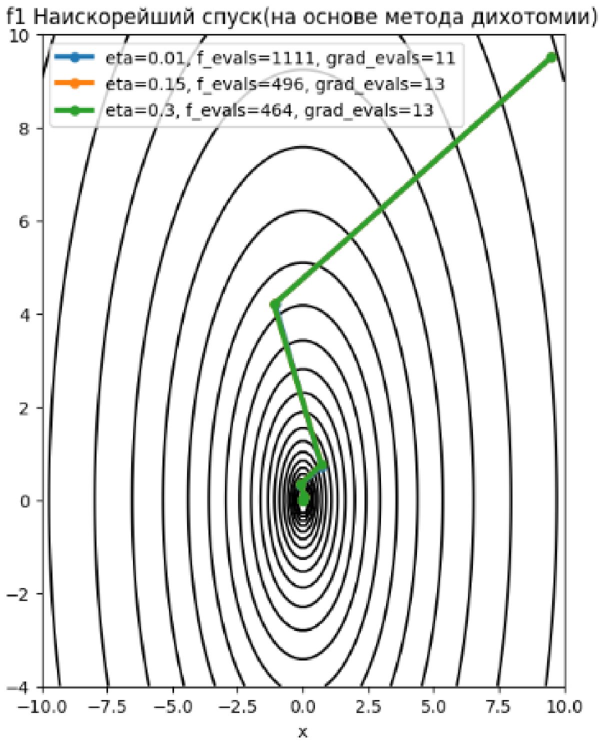
Этот вариант градиентного метода основывается на выборе шага из следующего соображения. Из точки $x^{[k]}$ будем двигаться в направлении антиградиента до тех пор пока не достигнем минимума функции f на этом направлении, т. е. на луче

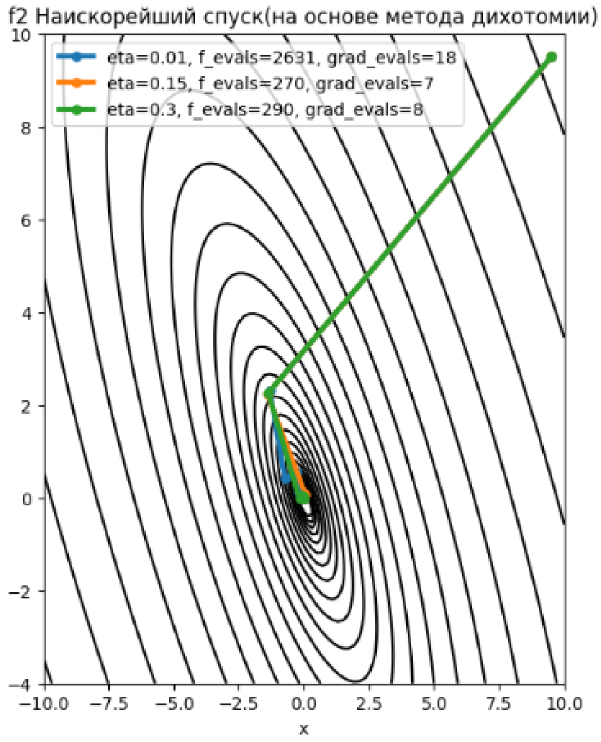
$$L = \left\{ x = x^{[k]} - \lambda f'(x^{[k]});\ \lambda \leq 0 \right\}$$

$$\lambda^{[k]} = \arg \min_{\lambda \in [0, \infty)} f(x^{[k]} - \lambda f'(x^{[k]}))$$

Каждый раз, $\lambda^{[k]}$ выбирается так, чтобы следующая итерация было точкой минимума функции f на луче L.

Однако, данный метод требует решения на каждом шаге задачи одномерной оптимизации - в нашем случае метода дихотомии. Теоретическая скорость сходимости не выше скорости сходимости градиентного метода с постоянным шагом, но на практике часто этот метод требует меньшего числа операций.

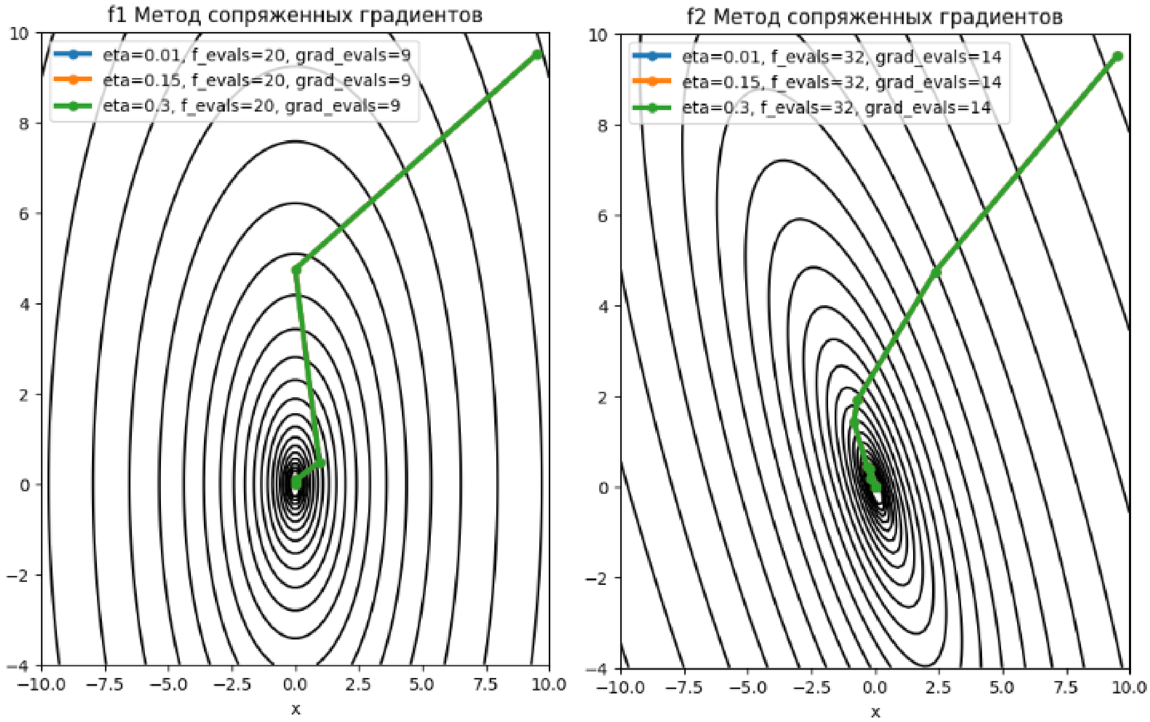




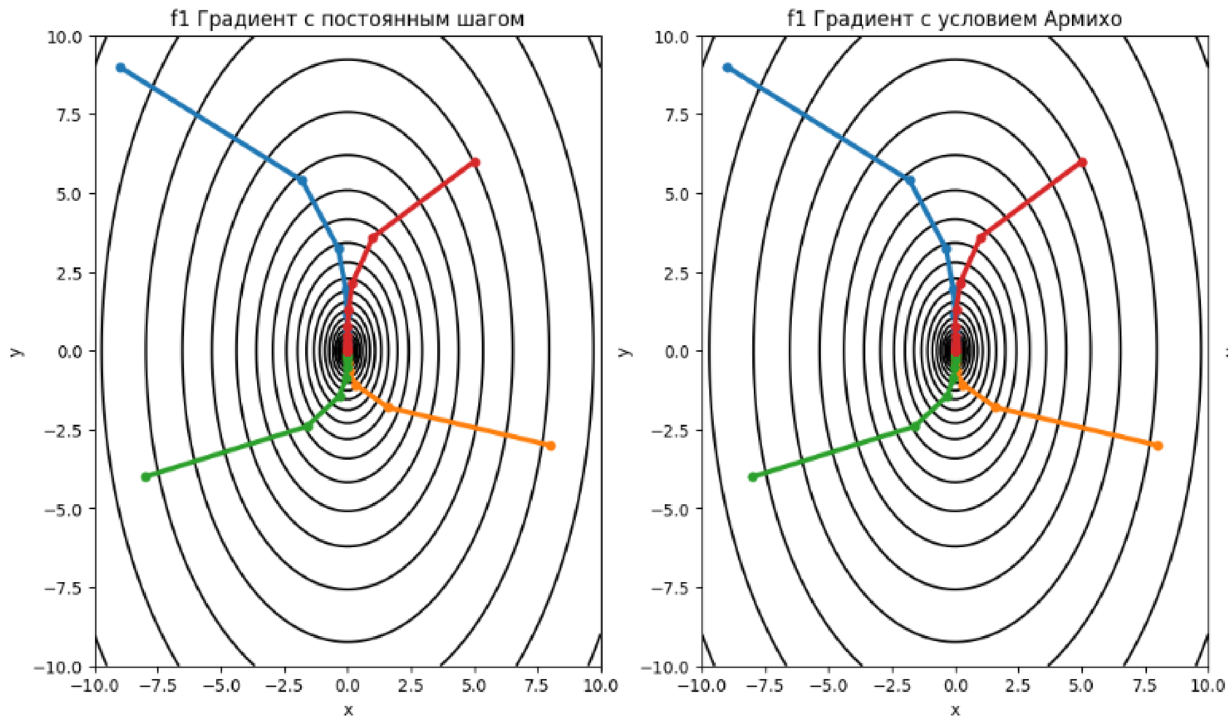
Метод сопряженных градиентов

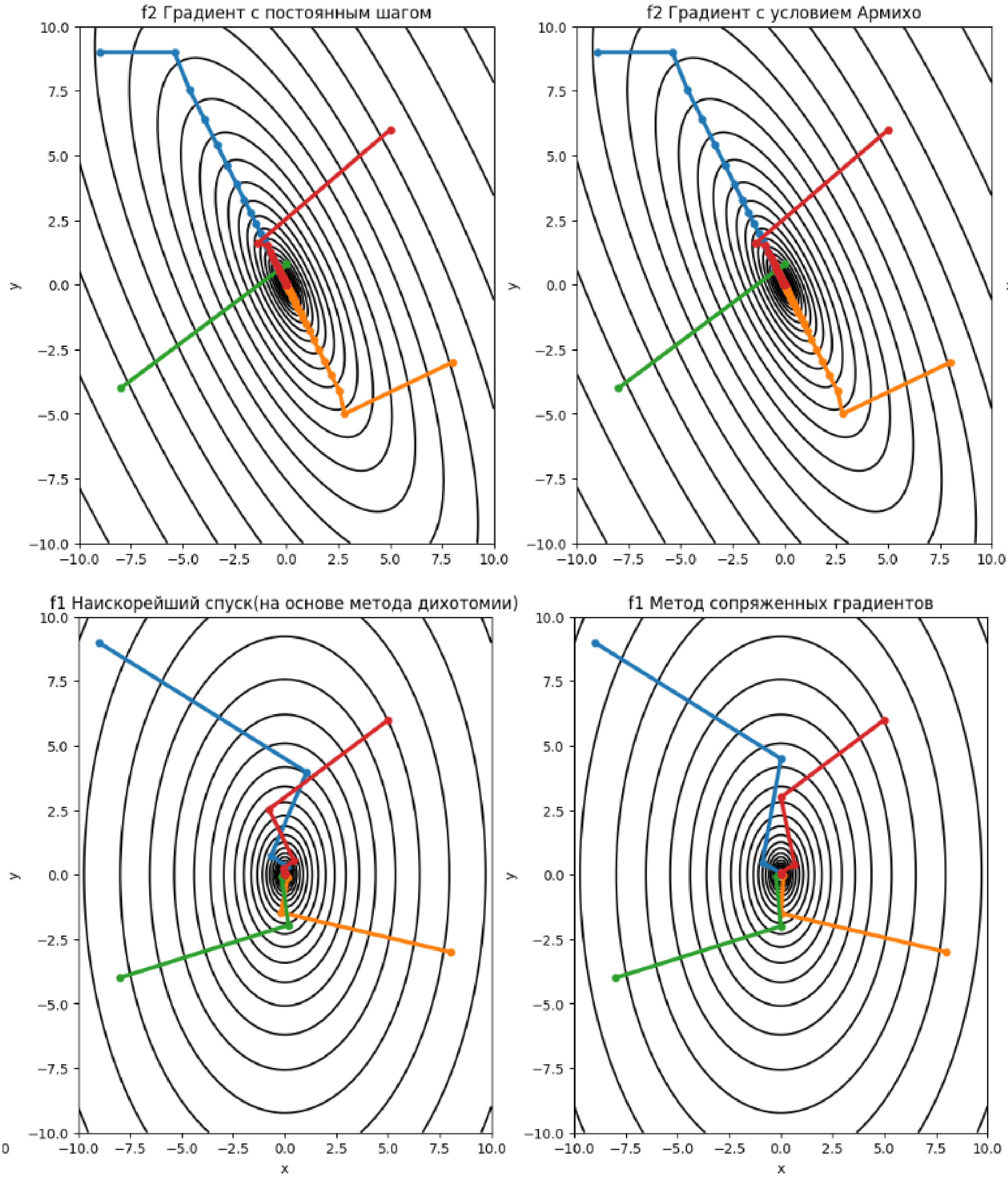
Итерационный метод для безусловной оптимизации в многомерном пространстве, является дальнейшим развитием метода наискорейшего спуска. Основное достоинство - он решает квадратичную задачу оптимизации за конечное число шагов.
Идея его заключается в том, чтобы выбрать направление спуска, принимая в расчет предыдущий шаг, отчего метод и получил свое название.

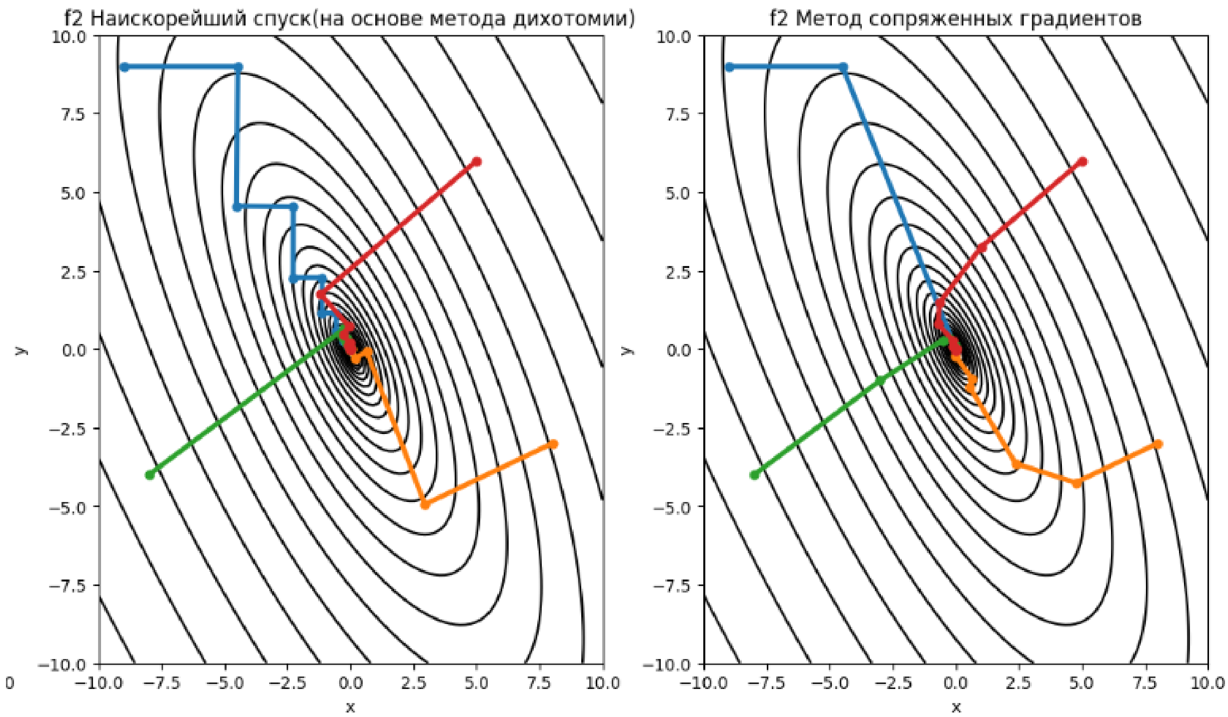
В случае квадратичной функции в R^n минимум находится не более чем за n шагов.



Исследуем алгоритмы в зависимости от начальной точки




f2 Градиент с постоянным шагом
f2 Градиент с условием Армихо
f1 Наискорейший спуск(на основе метода дихотомии)
f1 Метод сопряженных градиентов
x
y
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0



Вывод:

В общем случае метод постоянного шага не может гарантировать нам сходимость, но если шаг выбран правильно, то он сходится к оптимальному решению, поэтому для его работы необходимо выбирать шаг так, чтобы он был достаточно маленьким для обеспечения сходимости, но при этом не слишком маленьким, чтобы обеспечить точность и скорость сходимости. Но даже при выборе оптимального шага, данный метод показывает себя наименее эффективно относительно других рассмотренных нами методов.

Самым же эффективным методом стал метод сопряженных градиентов, хотя на рассмотренных нами функциях метод наискорейшего спуска работает лишь чуть менее эффективно.

Также мы выяснили, что выбор начальной точки сильно влияет на эффективность методов, а иногда и на их сходимость. Но в наших примерах мы такого не встретили.

Код и рассчеты:
https://github.com/ElderEv1l/applied-math/tree/main/lab-3/Code