



Концепция методов адаптивной метрики.  
Метод Ньютона. Квазиньютоновские  
методы

Даниил Меркулов

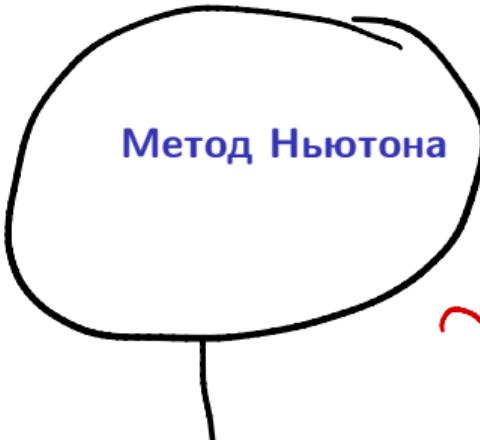
Методы оптимизации. МФТИ

линейка

$f$ -симво  
вып.  
шагкая

$$\sim \left(1 - \frac{\mu}{L}\right)^k$$

$$\min_{x \in \mathbb{R}^n} f(x)$$



GD

$$x_{k+1} = x_k - \Delta_k \nabla f(x_k)$$

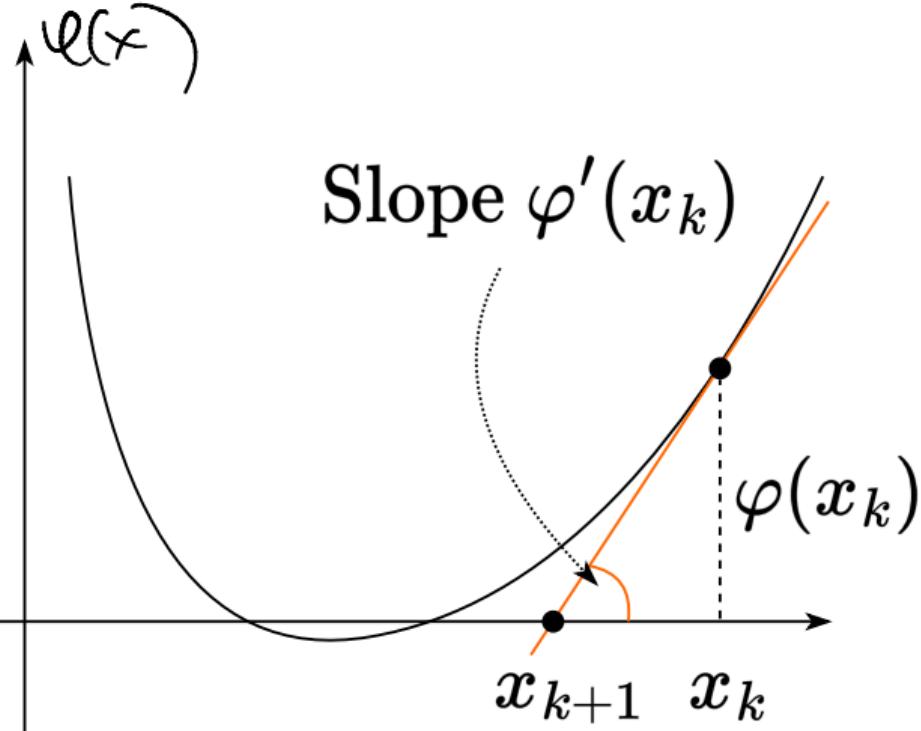
если  $f$ - вып. шагкая

$$\sim f - f^* \sim \frac{1}{k}$$

сублинейный

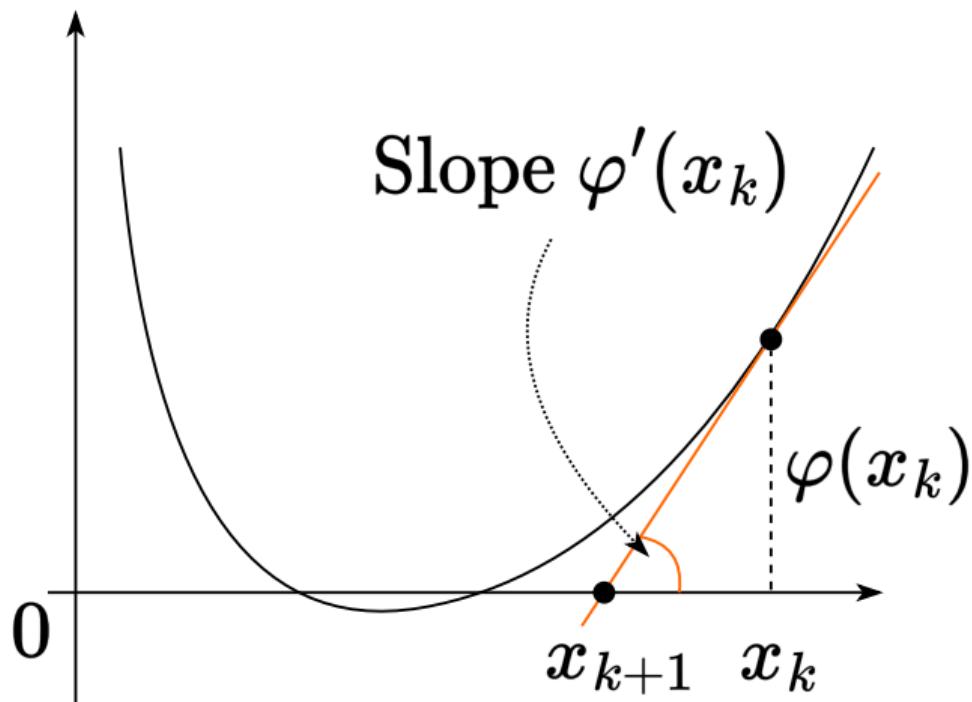
▽ ОЧЕНЬ  
▽ БЫСТРО ▽

## Идея метода Ньютона для нахождения корней функции



Рассмотрим функцию  $\varphi(x) : \mathbb{R} \rightarrow \mathbb{R}$ .

## Идея метода Ньютона для нахождения корней функции



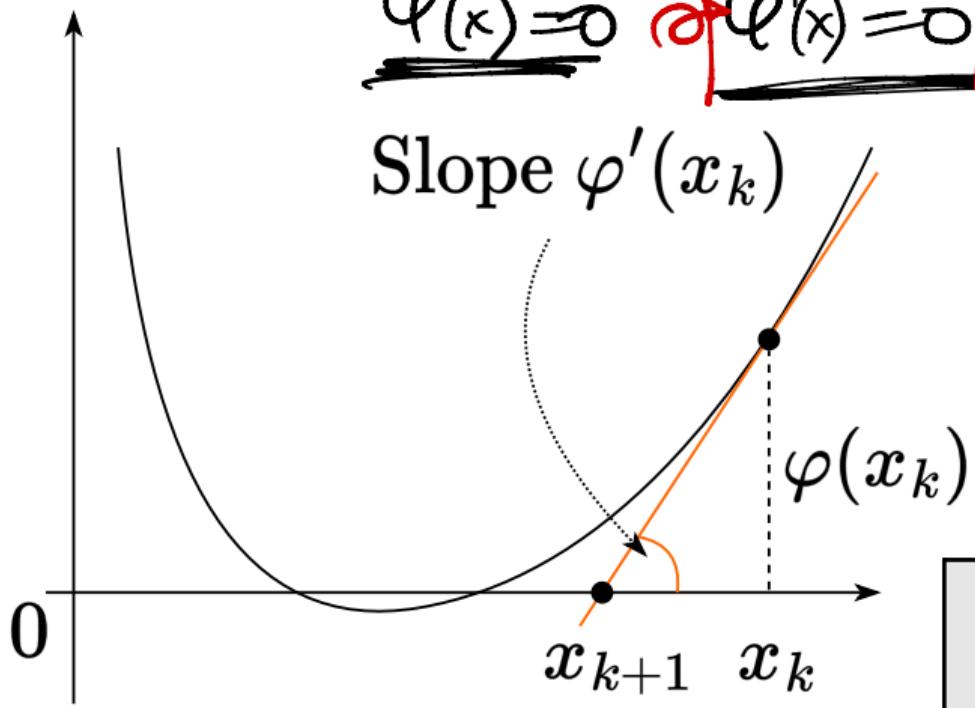
Рассмотрим функцию  $\varphi(x) : \mathbb{R} \rightarrow \mathbb{R}$ .

Основная идея заключается в том, чтобы построить линейное приближение в точке  $x_k$  и найти его корень, который будет новой точкой итерации:

## Идея метода Ньютона для нахождения корней функции

$$\varphi(x) = f(x)$$

~~$$\varphi(x) = 0 \quad \varphi'(x) = 0$$~~



Рассмотрим функцию  $\varphi(x) : \mathbb{R} \rightarrow \mathbb{R}$ .

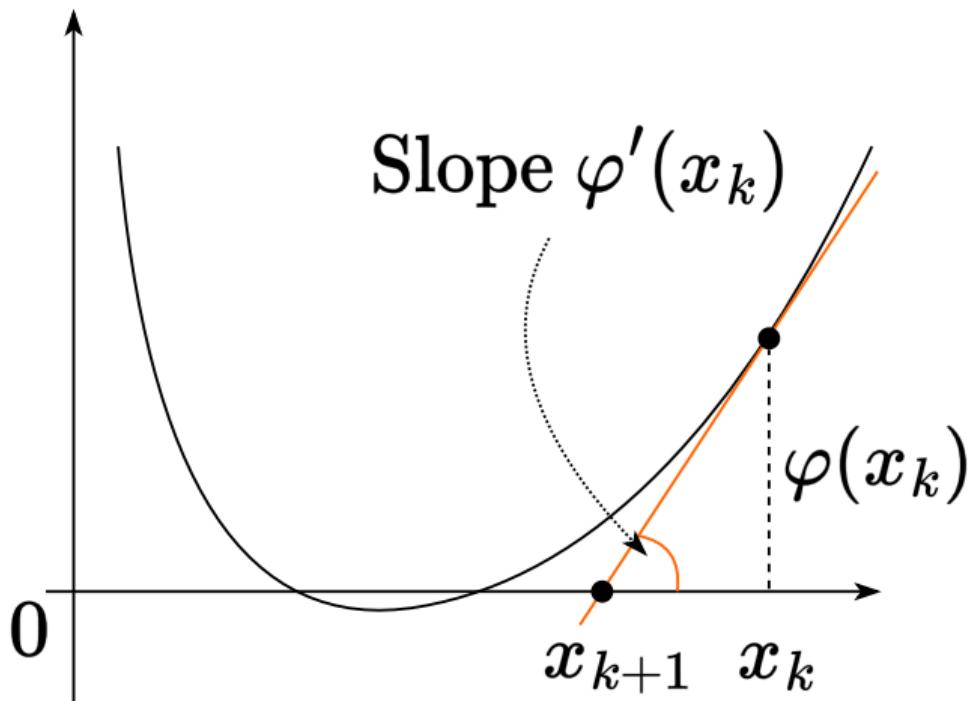
Основная идея заключается в том, чтобы построить линейное приближение в точке  $x_k$  и найти его корень, который будет новой точкой итерации:

$$-\varphi'(x_k) = \frac{\varphi(x_k)}{x_{k+1} - x_k}$$

$$x_{k+1} - x_k = -\frac{\varphi(x_k)}{\varphi'(x_k)}$$

$$x_{k+1} = x_k - \frac{\varphi(x_k)}{\varphi'(x_k)}$$

## Идея метода Ньютона для нахождения корней функции



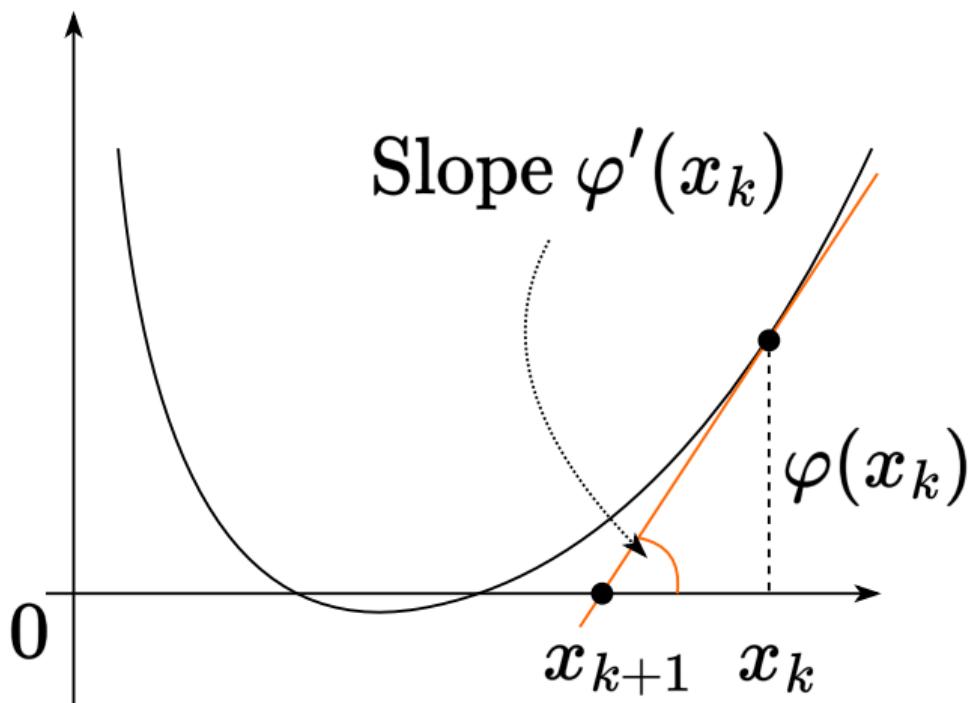
Рассмотрим функцию  $\varphi(x) : \mathbb{R} \rightarrow \mathbb{R}$ .

Основная идея заключается в том, чтобы построить линейное приближение в точке  $x_k$  и найти его корень, который будет новой точкой итерации:

$$\varphi'(x_k) = \frac{\varphi(x_k)}{x_{k+1} - x_k}$$

Мы получаем итерационную схему:

## Идея метода Ньютона для нахождения корней функции



Рассмотрим функцию  $\varphi(x) : \mathbb{R} \rightarrow \mathbb{R}$ .

Основная идея заключается в том, чтобы построить линейное приближение в точке  $x_k$  и найти его корень, который будет новой точкой итерации:

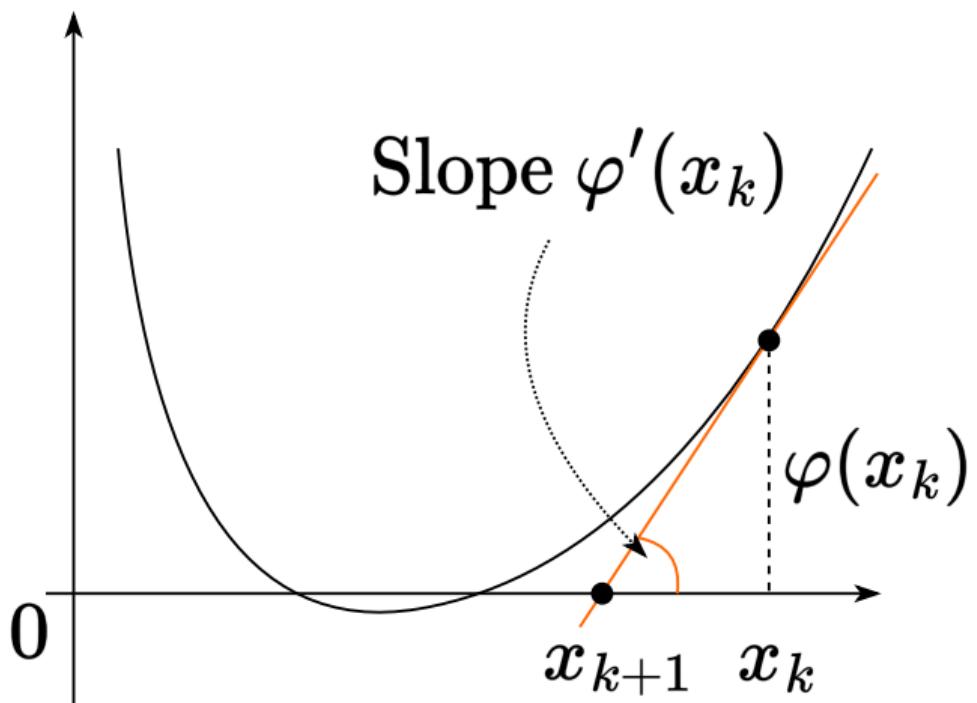
$$\varphi'(x_k) = \frac{\varphi(x_k)}{x_{k+1} - x_k}$$

Мы получаем итерационную схему:

$$x_{k+1} = x_k - \frac{\varphi(x_k)}{\varphi'(x_k)}.$$

<sup>1</sup>Мы фактически решаем задачу нахождения стационарных точек  $\nabla f(x) = 0$

## Идея метода Ньютона для нахождения корней функции



Рассмотрим функцию  $\varphi(x) : \mathbb{R} \rightarrow \mathbb{R}$ .  
Основная идея заключается в том, чтобы построить линейное приближение в точке  $x_k$  и найти его корень, который будет новой точкой итерации:

$$\varphi'(x_k) = \frac{\varphi(x_k)}{x_{k+1} - x_k}$$

Мы получаем итерационную схему:

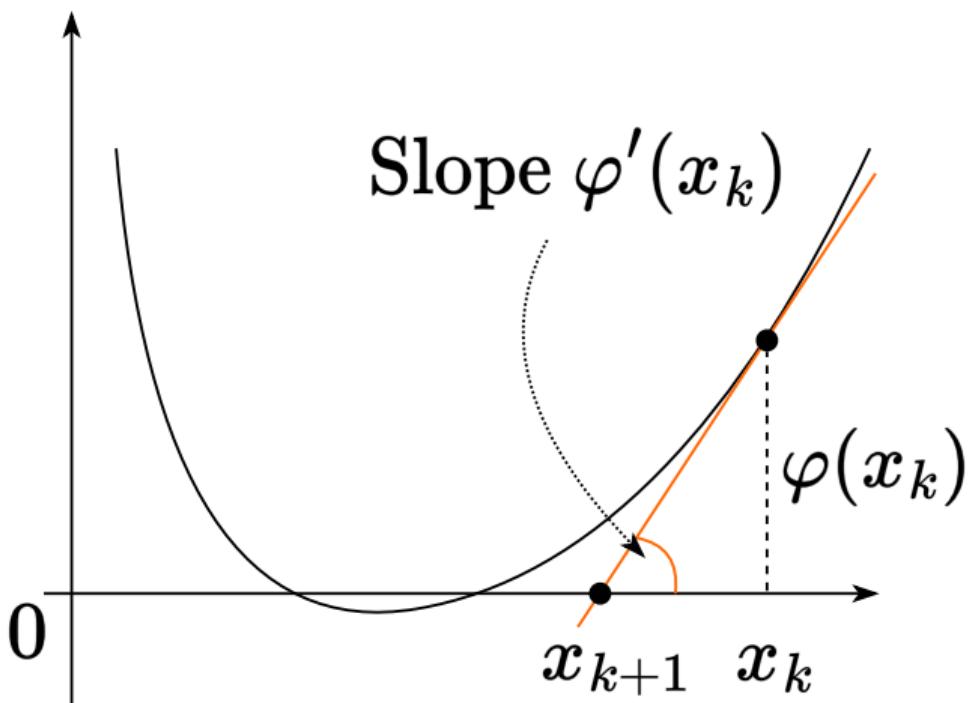
$$x_{k+1} = x_k - \frac{\varphi(x_k)}{\varphi'(x_k)}.$$

Этот метод станет методом оптимизации Ньютона в случае  $f'(x) = \varphi(x)$ <sup>1</sup>:

$$\begin{aligned} x_{k+1} &= x_k - \frac{f'(x_k)}{f''(x_k)} = \\ &= x_k - \frac{\cancel{f'(x_k)}}{\cancel{f''(x_k)}} \end{aligned}$$

<sup>1</sup> Мы фактически решаем задачу нахождения стационарных точек  $\nabla f(x) = 0$

## Идея метода Ньютона для нахождения корней функции



Рассмотрим функцию  $\varphi(x) : \mathbb{R} \rightarrow \mathbb{R}$ .  
Основная идея заключается в том, чтобы построить линейное приближение в точке  $x_k$  и найти его корень, который будет новой точкой итерации:

$$\varphi'(x_k) = \frac{\varphi(x_k)}{x_{k+1} - x_k}$$

Мы получаем итерационную схему:

$$x_{k+1} = x_k - \frac{\varphi(x_k)}{\varphi'(x_k)}.$$

Этот метод станет методом оптимизации Ньютона в случае  $f'(x) = \varphi(x)$ <sup>1</sup>:

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

<sup>1</sup> Мы фактически решаем задачу нахождения стационарных точек  $\nabla f(x) = 0$

## Метод Ньютона как оптимизация локальной квадратичной аппроксимации

Пусть у нас есть функция  $f(x)$  и некоторая точка  $x_k$ . Рассмотрим квадратичное приближение этой функции в окрестности  $x_k$ :

$$f_{x_k}^{\text{II}}(x) = f(x_k) + \underbrace{\langle \nabla f(x_k), x - x_k \rangle}_{\text{линейная часть}} + \frac{1}{2} \langle x - x_k, \nabla^2 f(x_k) (x - x_k) \rangle$$

$$\nabla_x f_{x_k}^{\text{II}}(x) = 0 \quad |_{x=x_{k+1}}$$

$$0 + \nabla f(x_k) + \nabla^2 f(x_k) \cdot (x - x_k) = 0$$

$$x_{k+1} - x_k = - [\nabla^2 f(x_k)]^{-1} \cdot \nabla f(x_k)$$

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

## Метод Ньютона как оптимизация локальной квадратичной аппроксимации

Пусть у нас есть функция  $f(x)$  и некоторая точка  $x_k$ . Рассмотрим квадратичное приближение этой функции в окрестности  $x_k$ :

$$f_{x_k}^{II}(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle.$$

## Метод Ньютона как оптимизация локальной квадратичной аппроксимации

Пусть у нас есть функция  $f(x)$  и некоторая точка  $x_k$ . Рассмотрим квадратичное приближение этой функции в окрестности  $x_k$ :

$$f_{x_k}^{II}(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle.$$

Идея метода заключается в том, чтобы найти точку  $x_{k+1}$ , которая минимизирует функцию  $f_{x_k}^{II}(x)$ , т.е.  
 $\nabla f_{x_k}^{II}(x_{k+1}) = 0$ .

## Метод Ньютона как оптимизация локальной квадратичной аппроксимации

Пусть у нас есть функция  $f(x)$  и некоторая точка  $x_k$ . Рассмотрим квадратичное приближение этой функции в окрестности  $x_k$ :

$$f_{x_k}^{II}(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle.$$

Идея метода заключается в том, чтобы найти точку  $x_{k+1}$ , которая минимизирует функцию  $f_{x_k}^{II}(x)$ , т.е.  $\nabla f_{x_k}^{II}(x_{k+1}) = 0$ .

$$\nabla f_{x_k}^{II}(x_{k+1}) = \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) = 0$$

## Метод Ньютона как оптимизация локальной квадратичной аппроксимации

Пусть у нас есть функция  $f(x)$  и некоторая точка  $x_k$ . Рассмотрим квадратичное приближение этой функции в окрестности  $x_k$ :

$$f_{x_k}^{II}(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle.$$

Идея метода заключается в том, чтобы найти точку  $x_{k+1}$ , которая минимизирует функцию  $f_{x_k}^{II}(x)$ , т.е.  
 $\nabla f_{x_k}^{II}(x_{k+1}) = 0$ .

$$\nabla f_{x_k}^{II}(x_{k+1}) = \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) = 0$$

$$\nabla^2 f(x_k)(x_{k+1} - x_k) = -\nabla f(x_k)$$

## Метод Ньютона как оптимизация локальной квадратичной аппроксимации

Пусть у нас есть функция  $f(x)$  и некоторая точка  $x_k$ . Рассмотрим квадратичное приближение этой функции в окрестности  $x_k$ :

$$f_{x_k}^{II}(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle.$$

Идея метода заключается в том, чтобы найти точку  $x_{k+1}$ , которая минимизирует функцию  $f_{x_k}^{II}(x)$ , т.е.  
 $\nabla f_{x_k}^{II}(x_{k+1}) = 0$ .

$$\nabla f_{x_k}^{II}(x_{k+1}) = \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) = 0$$

$$\nabla^2 f(x_k)(x_{k+1} - x_k) = -\nabla f(x_k)$$

$$[\nabla^2 f(x_k)]^{-1} \nabla^2 f(x_k)(x_{k+1} - x_k) = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

## Метод Ньютона как оптимизация локальной квадратичной аппроксимации

Пусть у нас есть функция  $f(x)$  и некоторая точка  $x_k$ . Рассмотрим квадратичное приближение этой функции в окрестности  $x_k$ :

$$f_{x_k}^{II}(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle.$$

Идея метода заключается в том, чтобы найти точку  $x_{k+1}$ , которая минимизирует функцию  $f_{x_k}^{II}(x)$ , т.е.  
 $\nabla f_{x_k}^{II}(x_{k+1}) = 0$ .

$$\begin{aligned}\nabla f_{x_k}^{II}(x_{k+1}) &= \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) = 0 \\ \nabla^2 f(x_k)(x_{k+1} - x_k) &= -\nabla f(x_k) \\ [\nabla^2 f(x_k)]^{-1} \nabla^2 f(x_k)(x_{k+1} - x_k) &= -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \\ x_{k+1} &= x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k).\end{aligned}$$

## Метод Ньютона как оптимизация локальной квадратичной аппроксимации

Пусть у нас есть функция  $f(x)$  и некоторая точка  $x_k$ . Рассмотрим квадратичное приближение этой функции в окрестности  $x_k$ :

$$f_{x_k}^{II}(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle.$$

Идея метода заключается в том, чтобы найти точку  $x_{k+1}$ , которая минимизирует функцию  $f_{x_k}^{II}(x)$ , т.е.  
 $\nabla f_{x_k}^{II}(x_{k+1}) = 0$ .

$$\begin{aligned}\nabla f_{x_k}^{II}(x_{k+1}) &= \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) = 0 \\ \nabla^2 f(x_k)(x_{k+1} - x_k) &= -\nabla f(x_k) \\ [\nabla^2 f(x_k)]^{-1} \nabla^2 f(x_k)(x_{k+1} - x_k) &= -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \\ x_{k+1} &= x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k).\end{aligned}$$

## Метод Ньютона как оптимизация локальной квадратичной аппроксимации

Пусть у нас есть функция  $f(x)$  и некоторая точка  $x_k$ . Рассмотрим квадратичное приближение этой функции в окрестности  $x_k$ :

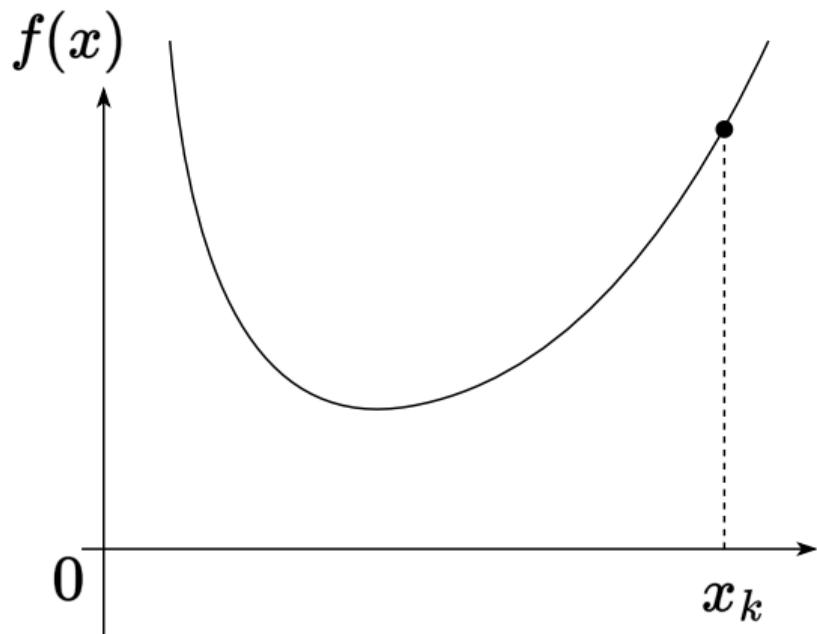
$$f_{x_k}^{II}(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle.$$

Идея метода заключается в том, чтобы найти точку  $x_{k+1}$ , которая минимизирует функцию  $f_{x_k}^{II}(x)$ , т.е.  $\nabla f_{x_k}^{II}(x_{k+1}) = 0$ .

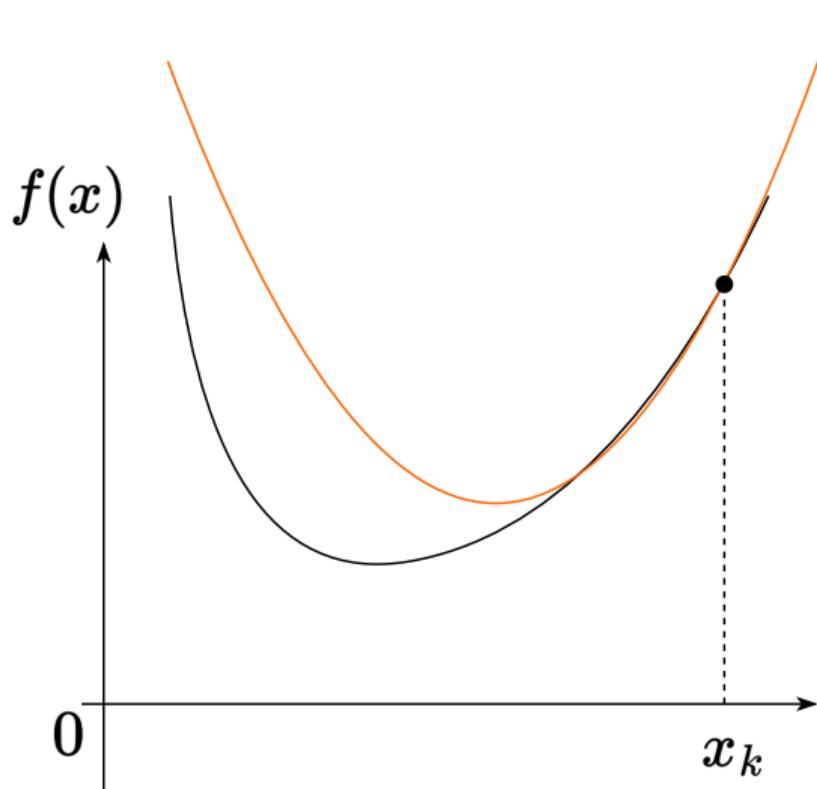
$$\begin{aligned}\nabla f_{x_k}^{II}(x_{k+1}) &= \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) = 0 \\ \nabla^2 f(x_k)(x_{k+1} - x_k) &= -\nabla f(x_k) \\ [\nabla^2 f(x_k)]^{-1} \nabla^2 f(x_k)(x_{k+1} - x_k) &= -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \\ x_{k+1} &= x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k).\end{aligned}$$

Необходимо отметить ограничения, связанные с необходимостью невырожденности (для существования метода) и положительной определенности (для гарантии сходимости) гессиана.

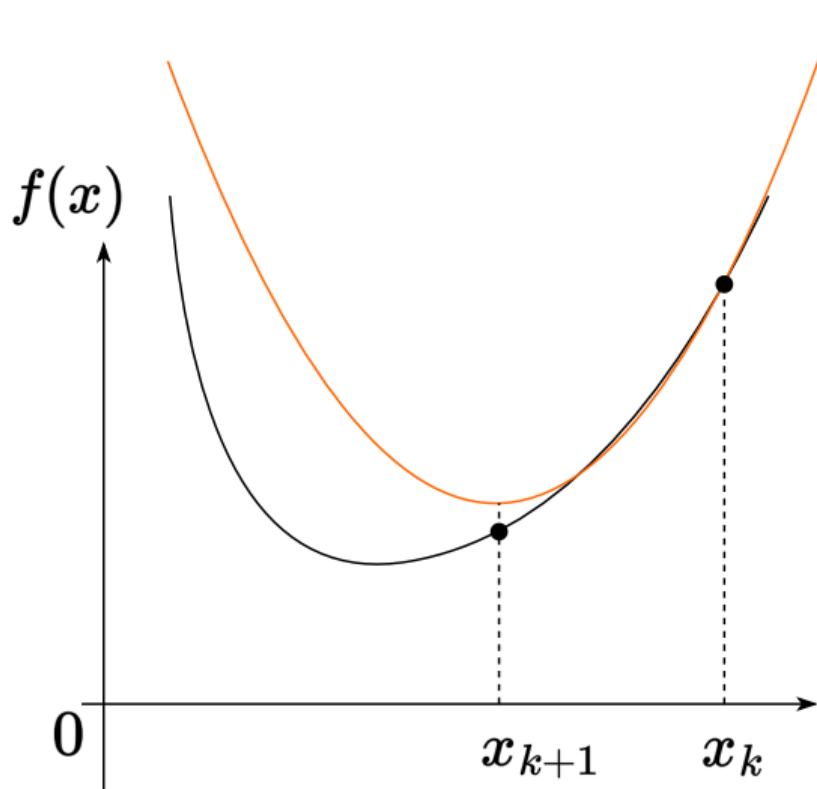
## Метод Ньютона как оптимизация локальной квадратичной аппроксимации



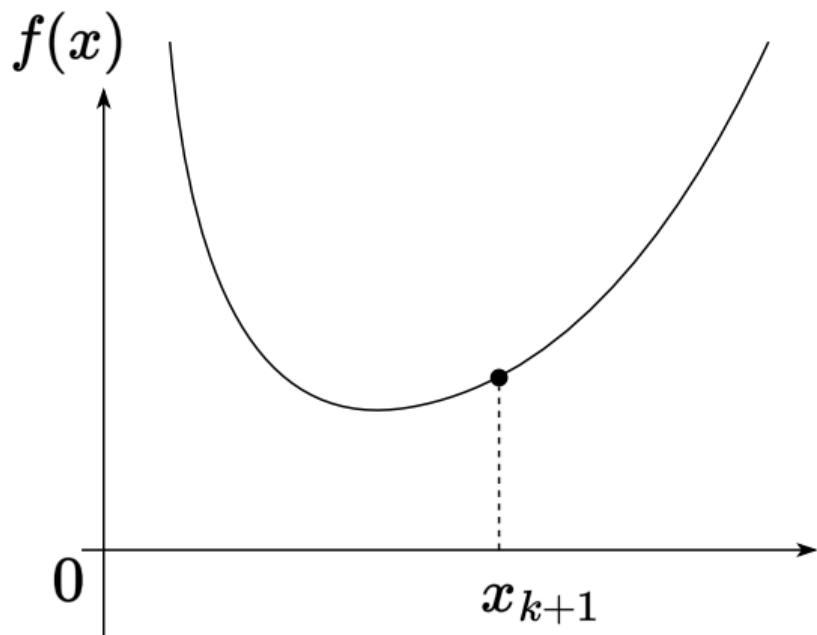
## Метод Ньютона как оптимизация локальной квадратичной аппроксимации



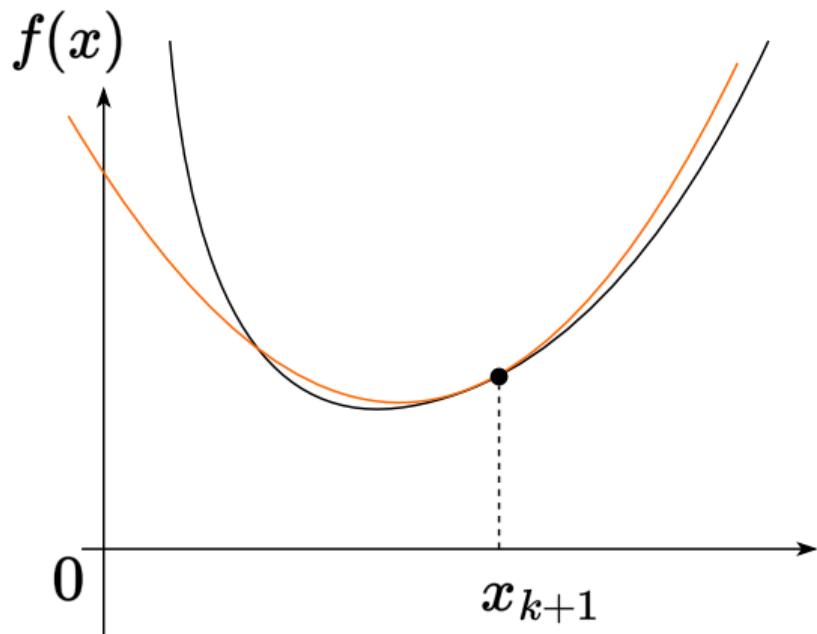
## Метод Ньютона как оптимизация локальной квадратичной аппроксимации



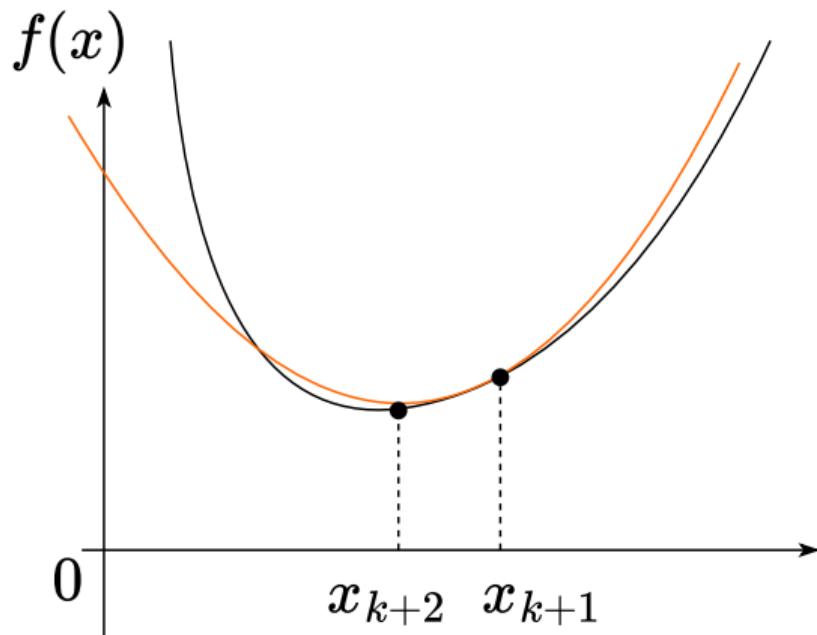
## Метод Ньютона как оптимизация локальной квадратичной аппроксимации



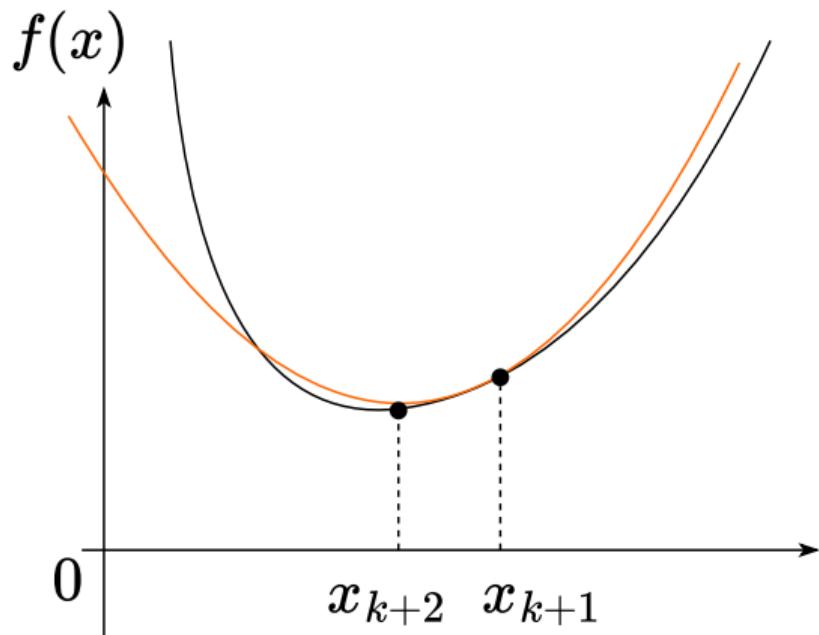
## Метод Ньютона как оптимизация локальной квадратичной аппроксимации



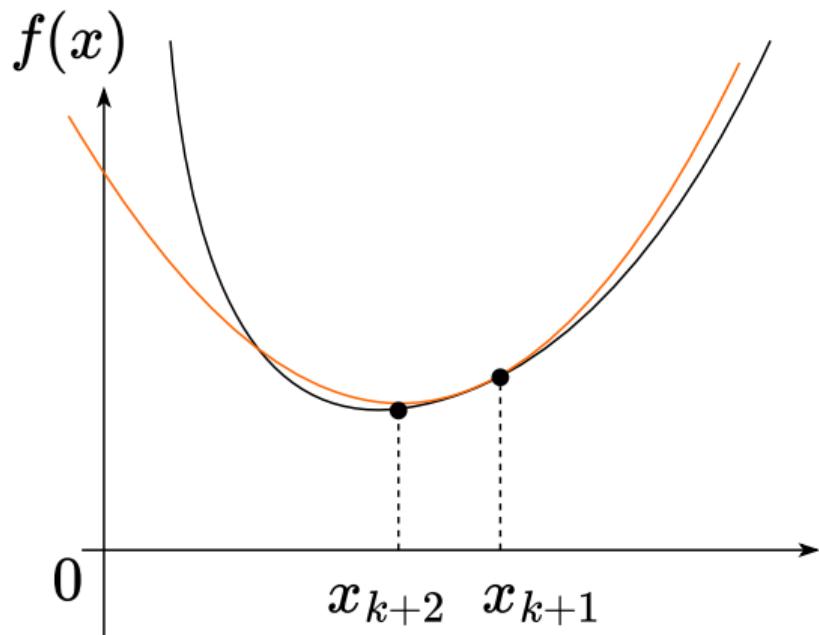
## Метод Ньютона как оптимизация локальной квадратичной аппроксимации



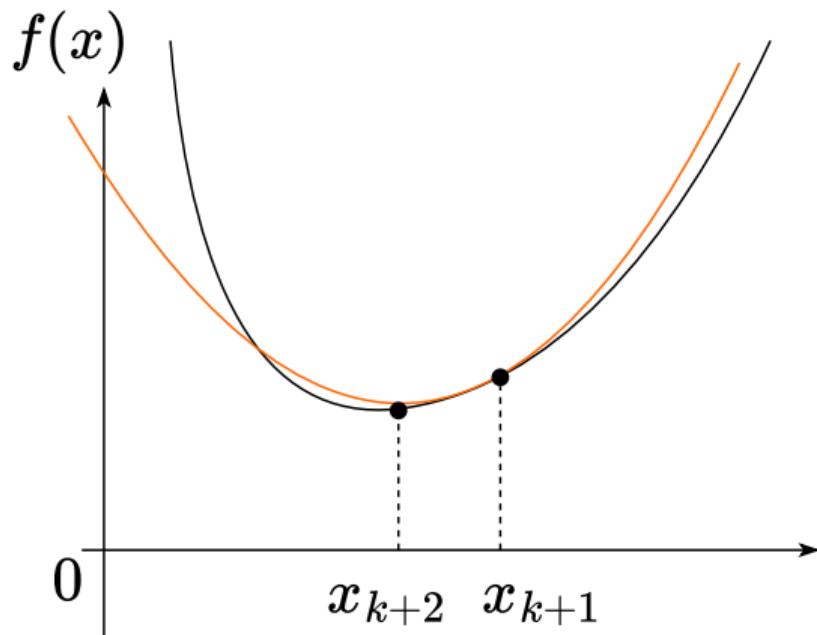
## Метод Ньютона как оптимизация локальной квадратичной аппроксимации



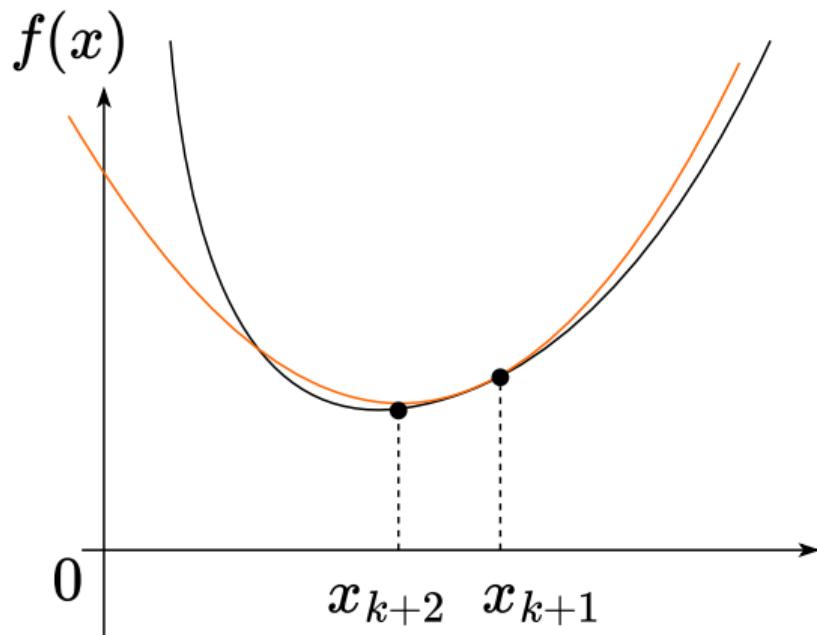
## Метод Ньютона как оптимизация локальной квадратичной аппроксимации



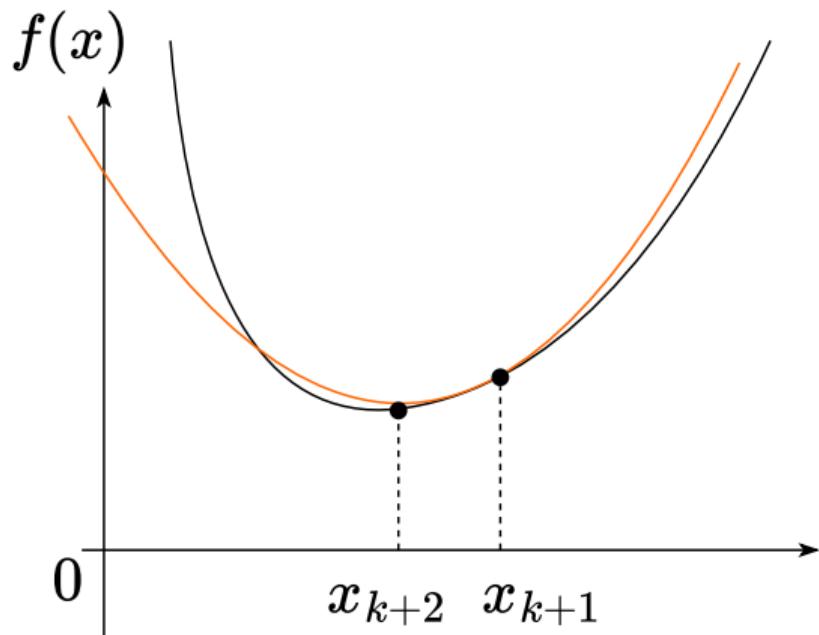
## Метод Ньютона как оптимизация локальной квадратичной аппроксимации



## Метод Ньютона как оптимизация локальной квадратичной аппроксимации



## Метод Ньютона как оптимизация локальной квадратичной аппроксимации



## Сходимость

3.

$$= \left( I - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) =$$

## Сходимость

3.

$$\begin{aligned} &= \left( I - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \nabla^2 f(x_k) - \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \end{aligned}$$

## Сходимость

3.

$$\begin{aligned} &= \left( I - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \nabla^2 f(x_k) - \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau \right) (x_k - x^*) = \end{aligned}$$

## Сходимость

3.

$$\begin{aligned} &= \left( I - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \nabla^2 f(x_k) - \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau \right) (x_k - x^*) = \\ &\qquad\qquad\qquad = [\nabla^2 f(x_k)]^{-1} G_k(x_k - x^*) \end{aligned}$$

## Сходимость

3.

$$\begin{aligned} &= \left( I - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \nabla^2 f(x_k) - \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau \right) (x_k - x^*) = \\ &\qquad\qquad\qquad = [\nabla^2 f(x_k)]^{-1} G_k (x_k - x^*) \end{aligned}$$

4. Введём:

$$G_k = \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau.$$

## Сходимость

5. Попробуем оценить размер  $G_k$  с помощью  $r_k = \|x_k - x^*\|$ :

## Сходимость

5. Попробуем оценить размер  $G_k$  с помощью  $r_k = \|x_k - x^*\|$ :

$$\|G_k\| = \left\| \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau \right\| \leq$$

## Сходимость

5. Попробуем оценить размер  $G_k$  с помощью  $r_k = \|x_k - x^*\|$ :

$$\begin{aligned} \|G_k\| &= \left\| \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau \right\| \leq \\ &\leq \int_0^1 \|\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))\| d\tau \leq \quad (\text{Липшицевость гессиана}) \end{aligned}$$

## Сходимость

5. Попробуем оценить размер  $G_k$  с помощью  $r_k = \|x_k - x^*\|$ :

$$\begin{aligned} \|G_k\| &= \left\| \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau \right\| \leq \\ &\leq \int_0^1 \|\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))\| d\tau \leq \quad (\text{Липшицевость гессиана}) \\ &\leq \int_0^1 M \|x_k - x^* - \tau(x_k - x^*)\| d\tau = \int_0^1 M \|x_k - x^*\| (1 - \tau) d\tau = \frac{r_k}{2} M, \end{aligned}$$

## Сходимость

5. Попробуем оценить размер  $G_k$  с помощью  $r_k = \|x_k - x^*\|$ :

$$\begin{aligned}\|G_k\| &= \left\| \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau \right\| \leq \\ &\leq \int_0^1 \|\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))\| d\tau \leq \quad (\text{Липшицевость гессиана}) \\ &\leq \int_0^1 M \|x_k - x^* - \tau(x_k - x^*)\| d\tau = \int_0^1 M \|x_k - x^*\| (1 - \tau) d\tau = \frac{r_k}{2} M,\end{aligned}$$

6. Получаем:

$$r_{k+1} \leq \left\| [\nabla^2 f(x_k)]^{-1} \right\| \cdot \frac{r_k}{2} M \cdot r_k \leq \frac{M}{2\mu} r_k^2$$

и нам нужно оценить норму обратного гессиана

$$\nu_k \leq \frac{M}{\mu}$$

## Сходимость

7. Из липшицевости и симметричности  
гессиана:

## Сходимость

7. Из липшицевости и симметричности гессиана:

$$-\underline{Mr_k I} \leq \nabla^2 f(x_k) - \nabla^2 f(x^*) \leq \overline{Mr_k I}$$

$$\nabla^2 f(x_k) - \nabla^2 f(x^*) \succeq -Mr_k I_n$$

## Сходимость

7. Из липшицевости и симметричности гессиана:

$$\nabla^2 f(x_k) - \nabla^2 f(x^*) \succeq -Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq \nabla^2 f(x^*) - Mr_k I_n$$



## Сходимость

7. Из липшицевости и симметричности гессиана:

$$\nabla^2 f(x_k) - \nabla^2 f(x^*) \succeq -Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq \nabla^2 f(x^*) - Mr_k I_n$$

$$\boxed{\nabla^2 f(x_k) \succeq \mu I_n - Mr_k I_n}$$

$$\mu > 0$$

## Сходимость

7. Из липшицевости и симметричности гессиана:

$$\nabla^2 f(x_k) - \nabla^2 f(x^*) \succeq -Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq \nabla^2 f(x^*) - Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq \mu I_n - Mr_k I_n$$

$$\boxed{\nabla^2 f(x_k) \succeq (\mu - Mr_k) I_n}$$

## Сходимость

7. Из липшицевости и симметричности гессиана:

$$\nabla^2 f(x_k) - \nabla^2 f(x^*) \succeq -Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq \nabla^2 f(x^*) - Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq \mu I_n - Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq (\mu - Mr_k) I_n > 0$$

$$\mu - Mr_k > 0$$

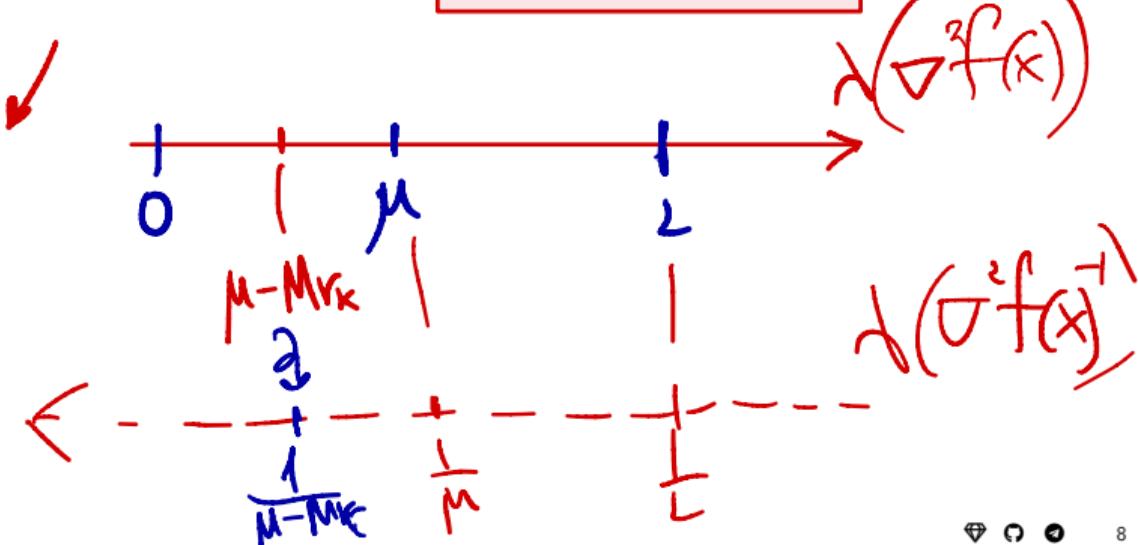
$$r_k < \frac{\mu}{M}$$

8. Из сильной выпуклости следует, что

$$\nabla^2 f(x_k) \succ 0, \text{ т.е. } r_k < \frac{\mu}{M}.$$

$$\|[\nabla^2 f(x_k)]^{-1}\| \leq (\mu - Mr_k)^{-1}$$

$$r_{k+1} \leq \frac{r_k^2 M}{2(\mu - Mr_k)}$$



## Сходимость

7. Из липшицевости и симметричности гессиана:

$$\nabla^2 f(x_k) - \nabla^2 f(x^*) \succeq -Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq \nabla^2 f(x^*) - Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq \mu I_n - Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq (\mu - Mr_k) I_n$$

8. Из сильной выпуклости следует, что

$$\nabla^2 f(x_k) \succ 0, \text{ т.е. } r_k < \frac{\mu}{M}.$$

$$\left\| [\nabla^2 f(x_k)]^{-1} \right\| \leq (\mu - Mr_k)^{-1}$$

$$r_{k+1} \leq \frac{r_k^2 M}{2(\mu - Mr_k)}$$

9. Потребуем, чтобы верхняя оценка на  $r_{k+1}$  была меньше  $r_k$ , учитывая, что  $0 < r_k < \frac{\mu}{M}$ :

$$r_{k+1} = \frac{r_k^2 M}{2(\mu - Mr_k)} < r_k$$

$$\frac{M}{2(\mu - Mr_k)} r_k < 1$$

$$Mr_k < 2(\mu - Mr_k)$$

$$3Mr_k < 2\mu$$

$$r_k < \frac{2\mu}{3M}$$

Было:  
 $\downarrow$   
 $r_k < \frac{\mu}{M}$

Пусть

$$r_{k+1} < r_k$$

## Сходимость

7. Из липшицевости и симметричности гессиана:

$$\nabla^2 f(x_k) - \nabla^2 f(x^*) \succeq -Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq \nabla^2 f(x^*) - Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq \mu I_n - Mr_k I_n$$

$$\nabla^2 f(x_k) \succeq (\mu - Mr_k) I_n$$

8. Из сильной выпуклости следует, что

$$\nabla^2 f(x_k) \succ 0, \text{ т.е. } r_k < \frac{\mu}{M}.$$

$$\left\| [\nabla^2 f(x_k)]^{-1} \right\| \leq (\mu - Mr_k)^{-1}$$

$$r_{k+1} \leq \frac{r_k^2 M}{2(\mu - Mr_k)}$$

9. Потребуем, чтобы верхняя оценка на  $r_{k+1}$  была меньше  $r_k$ , учитывая, что  $0 < r_k < \frac{\mu}{M}$ :

$$\frac{r_k^2 M}{2(\mu - Mr_k)} < r_k$$

$$\frac{M}{2(\mu - Mr_k)} r_k < 1$$

$$Mr_k < 2(\mu - Mr_k)$$

$$3Mr_k < 2\mu$$

$$r_k < \frac{2\mu}{3M}$$

10. Возвращаясь к оценке невязки на  $k + 1$ -ой итерации, получаем:

$$r_{k+1} \leq \frac{r_k^2 M}{2(\mu - Mr_k)} < \frac{3Mr_k^2}{2\mu}$$

Таким образом, мы получили важный результат: метод Ньютона для функции с липшицевым положительно определённым гессианом сходится **квадратично** вблизи решения.

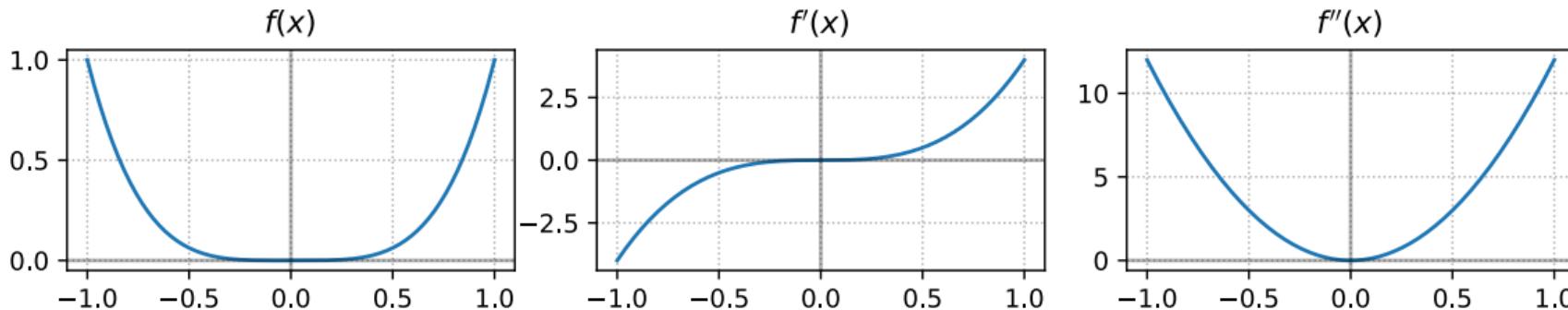
## Свойства метода Ньютона

## Отсутствие квадратичной сходимости, если некоторые предположения нарушаются

i

$x_{k+1} = \dots$

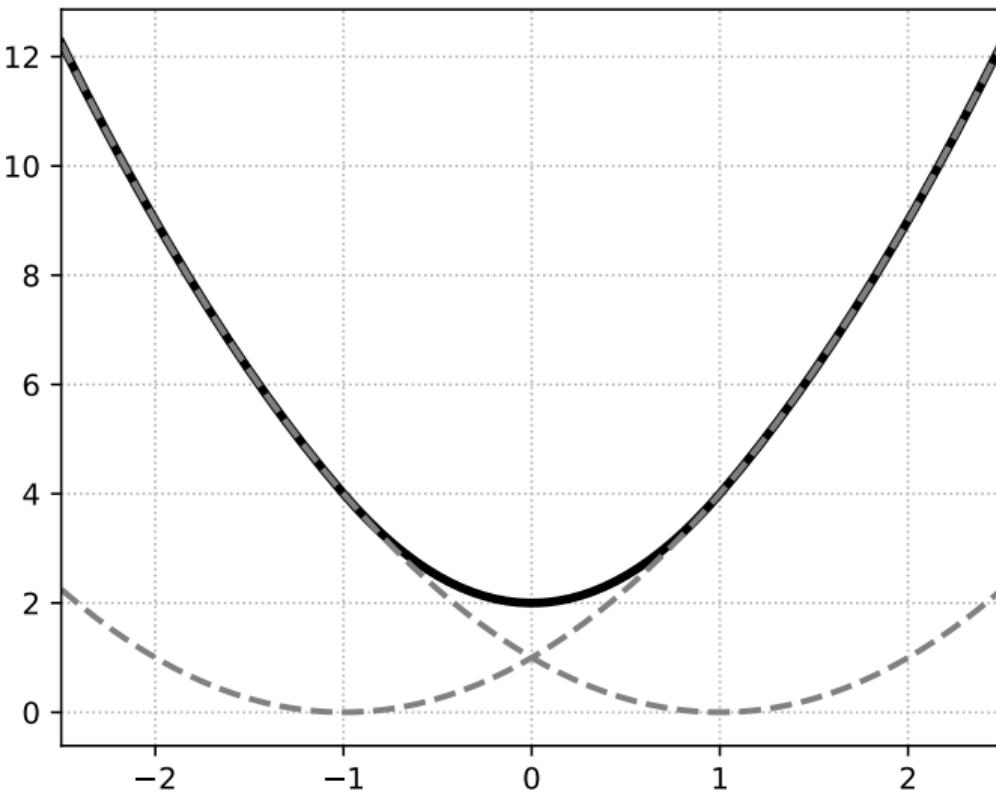
$$f(x) = x^4 \quad f'(x) = 4x^3 \quad f''(x) = 12x^2$$



$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} = x_k - \frac{4x_k^3}{12x_k^2} = x_k - \frac{1}{3}x_k = \frac{2}{3}x_k,$$

сходится к 0, единственному решению задачи, с линейной скоростью.

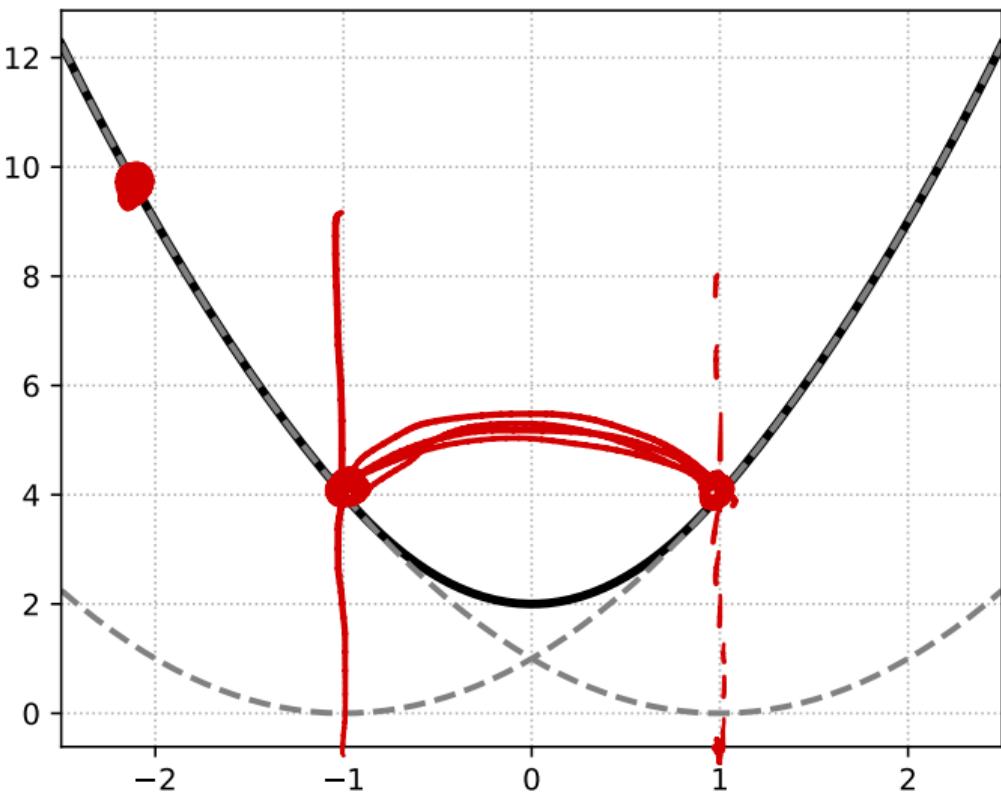
## Локальная сходимость метода Ньютона для гладкой сильно выпуклой $f(x)$



$$f(x) = \begin{cases} (x-1)^2, & x \leq -1 \\ 2x^2 + 2, & -1 < x < 1 \\ (x+1)^2, & x \geq 1 \end{cases}$$

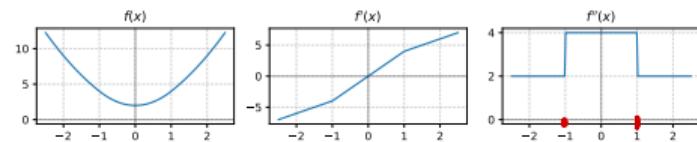
Эта функция сильно выпукла, но вторая производная не является липшицевой.

# Локальная сходимость метода Ньютона для гладкой сильно выпуклой $f(x)$

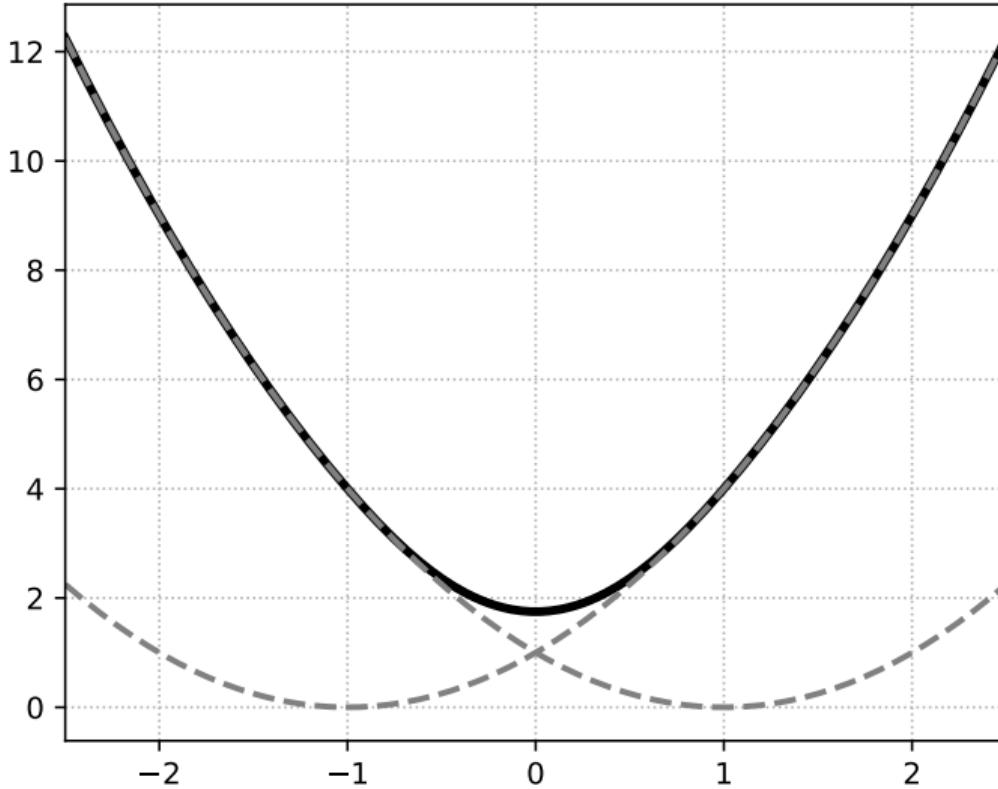


$$f(x) = \begin{cases} (x-1)^2, & x \leq -1 \\ 2x^2 + 2, & -1 < x < 1 \\ (x+1)^2, & x \geq 1 \end{cases}$$

Эта функция сильно выпукла, но вторая производная не является липшицевой.

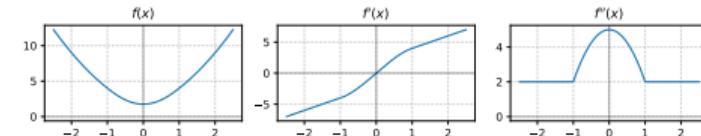


## Локальная сходимость метода Ньютона даже если $\nabla^2 f$ липшицев

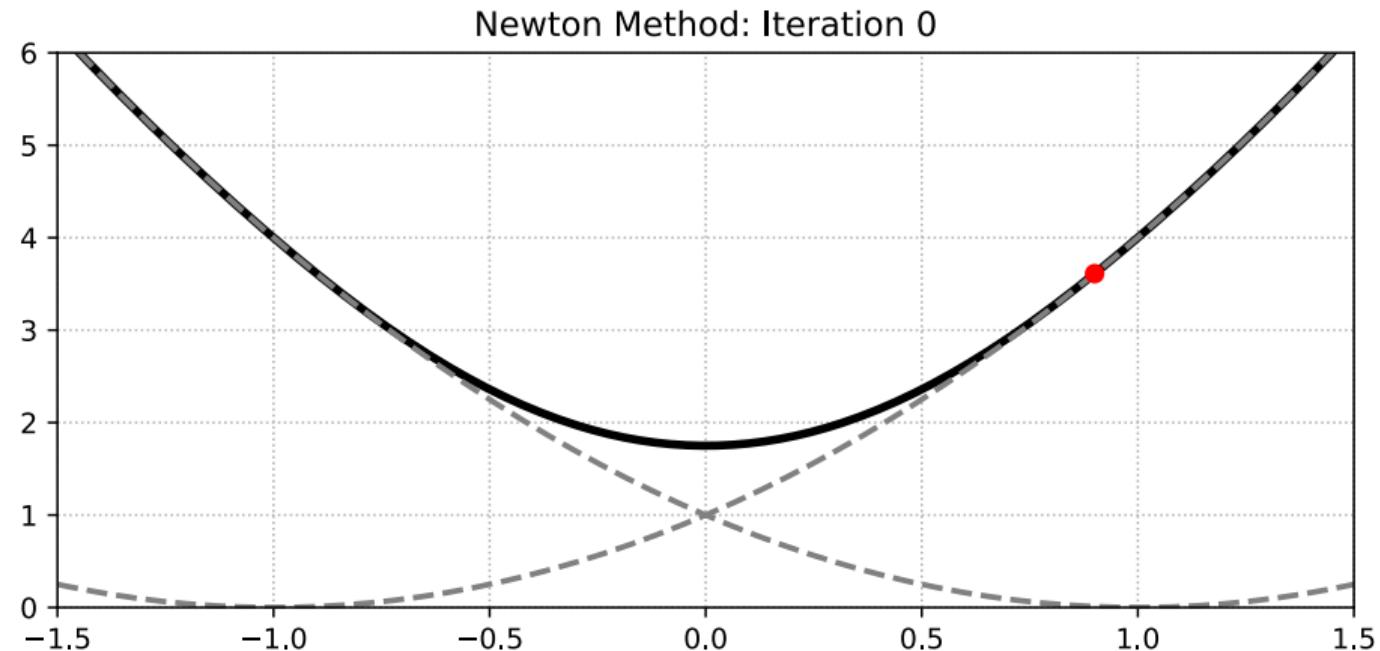


$$f(x) = \begin{cases} (x-1)^2, & x \leq -1 \\ -\frac{1}{4}x^4 + \frac{5}{2}x^2 + \frac{7}{4}, & -1 < x < 1 \\ (x+1)^2, & x \geq 1 \end{cases}$$

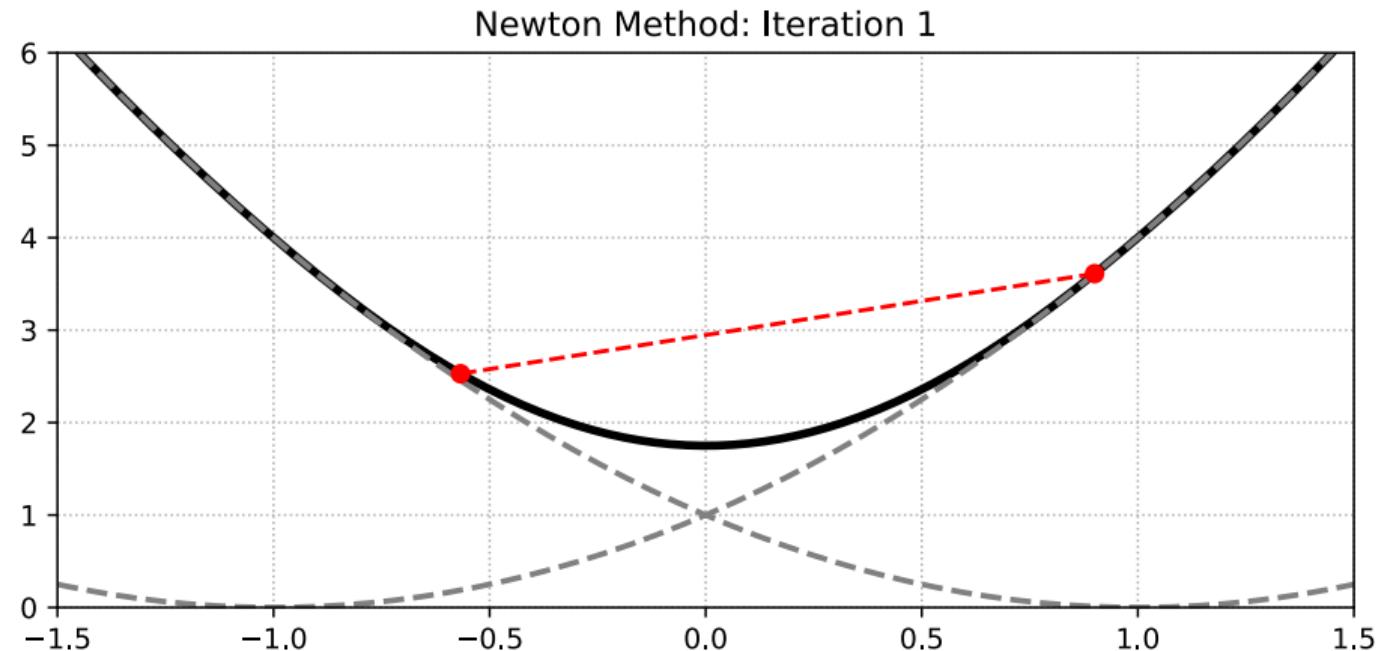
Эта функция сильно выпукла и вторая производная является липшицевой.



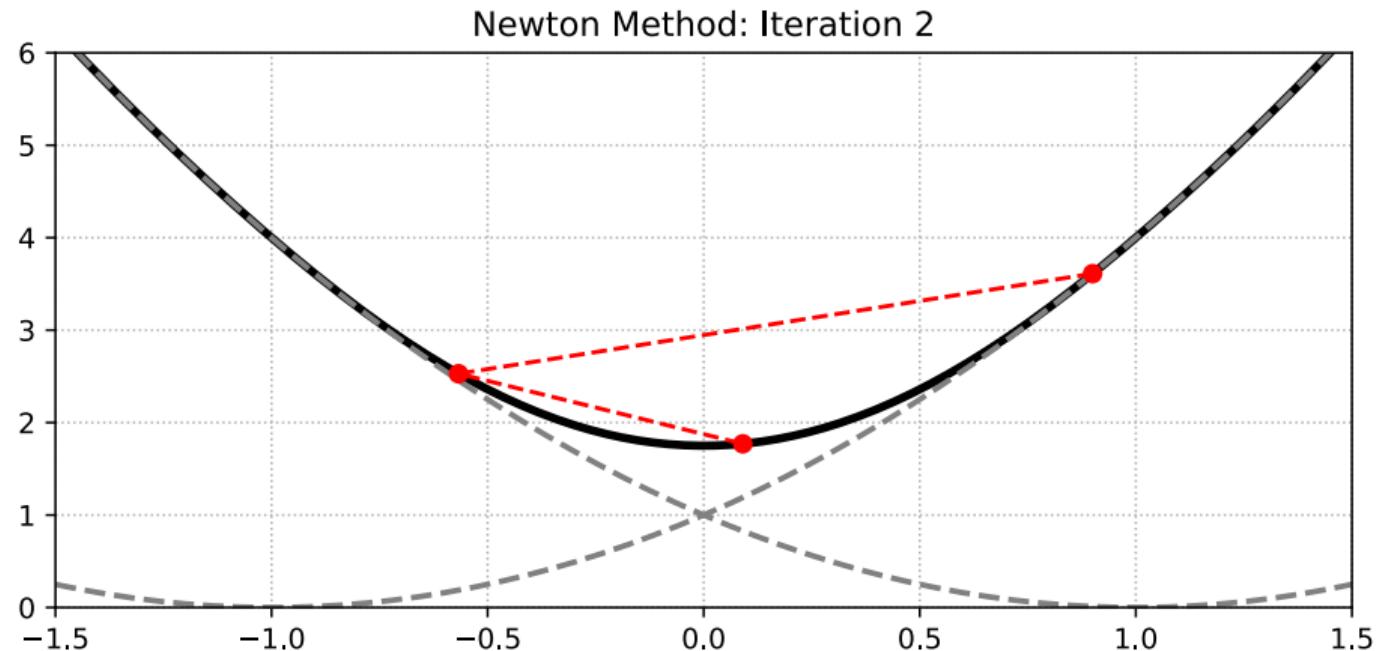
## Локальная сходимость метода Ньютона. Хорошая инициализация



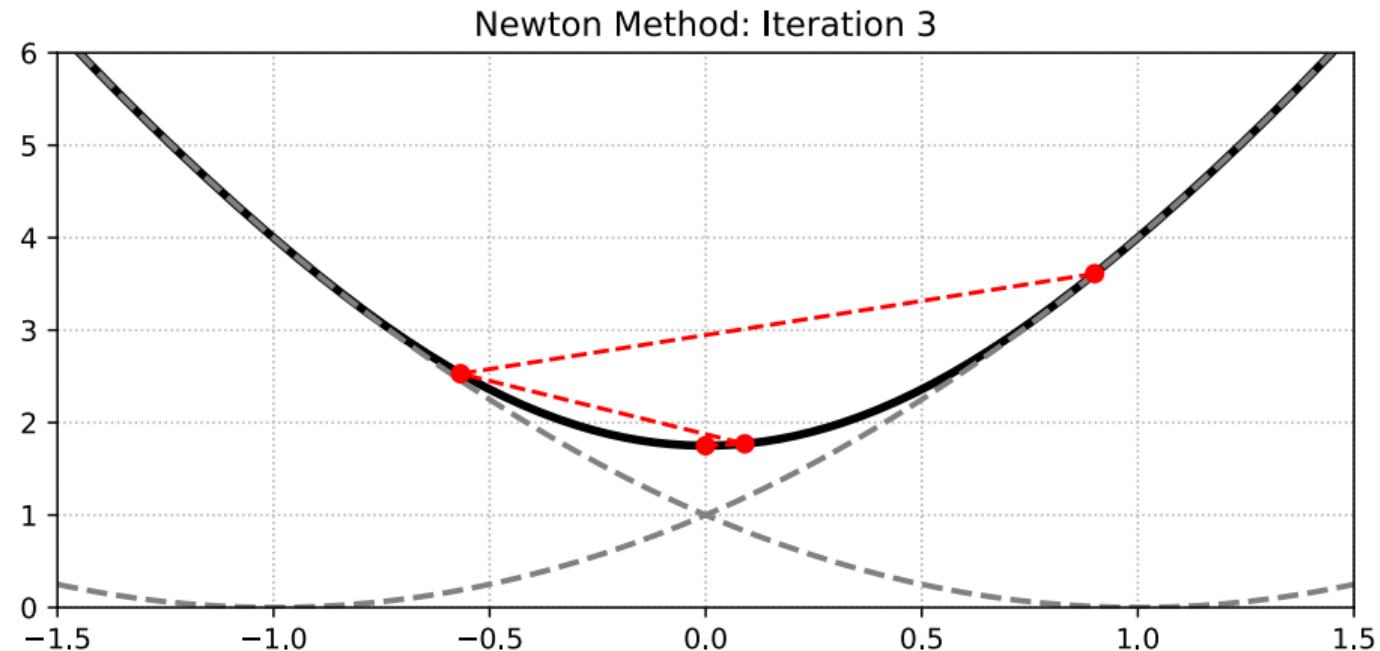
## Локальная сходимость метода Ньютона. Хорошая инициализация



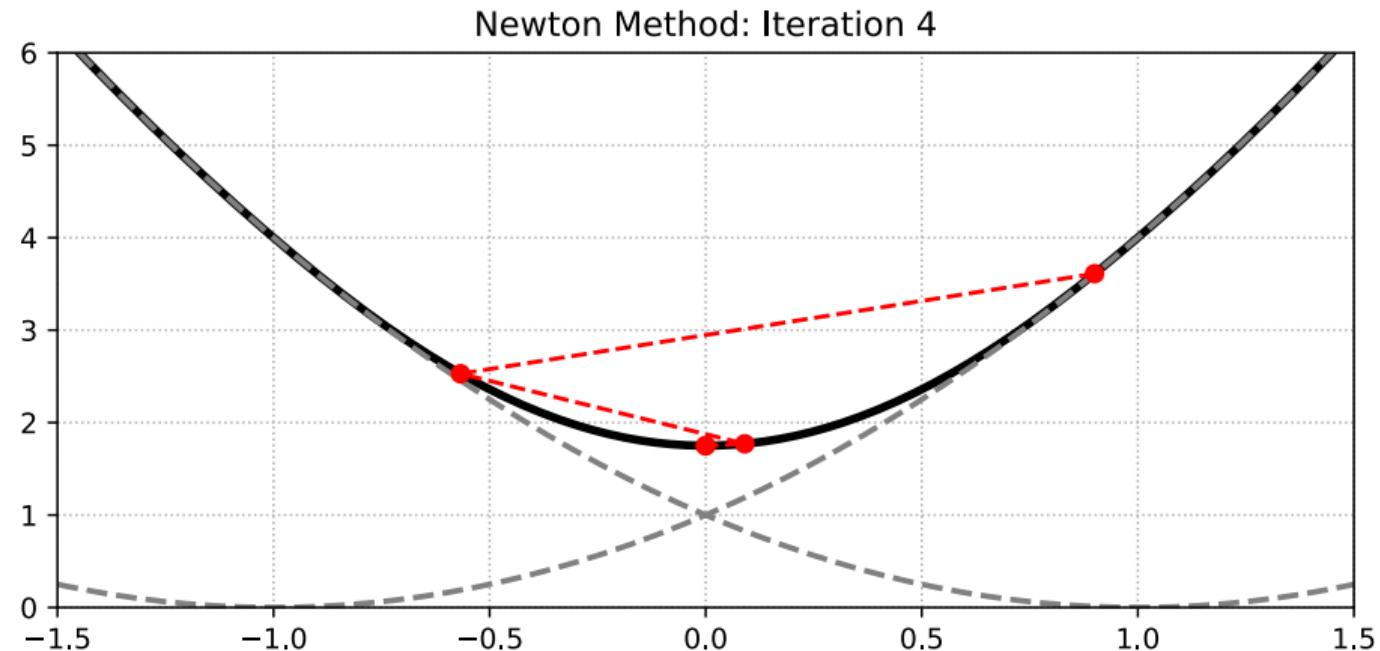
## Локальная сходимость метода Ньютона. Хорошая инициализация



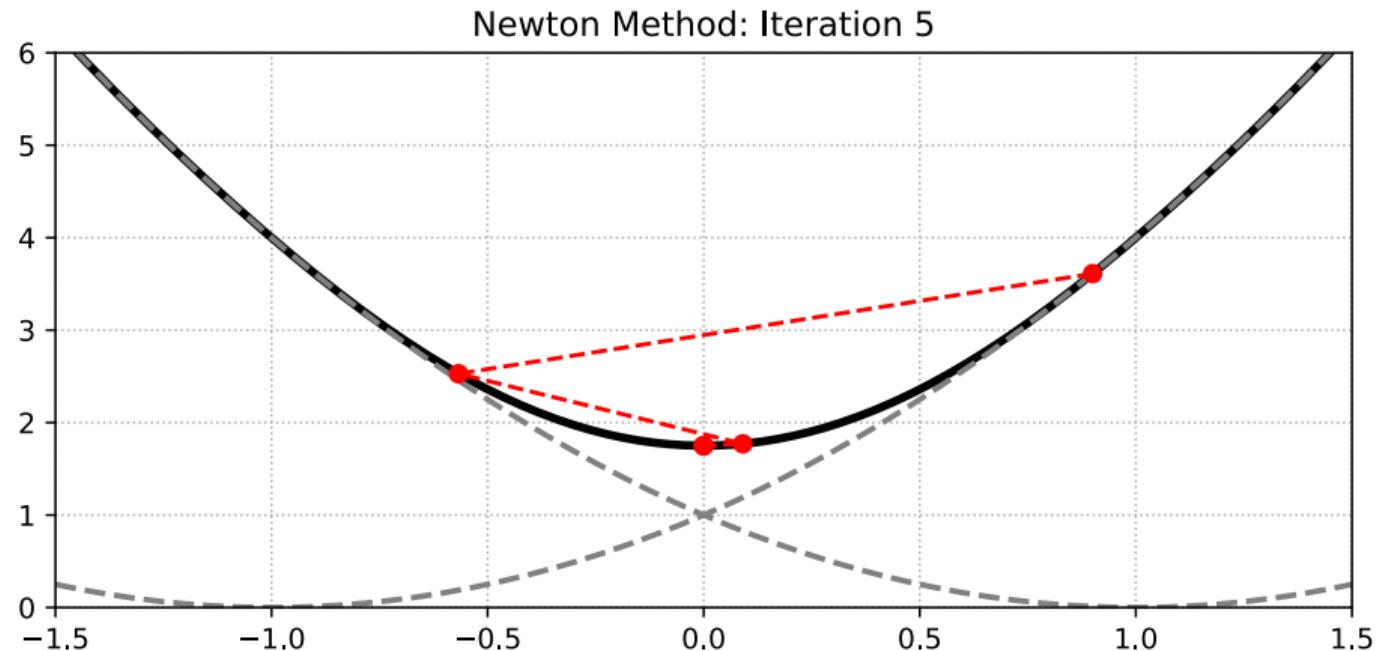
## Локальная сходимость метода Ньютона. Хорошая инициализация



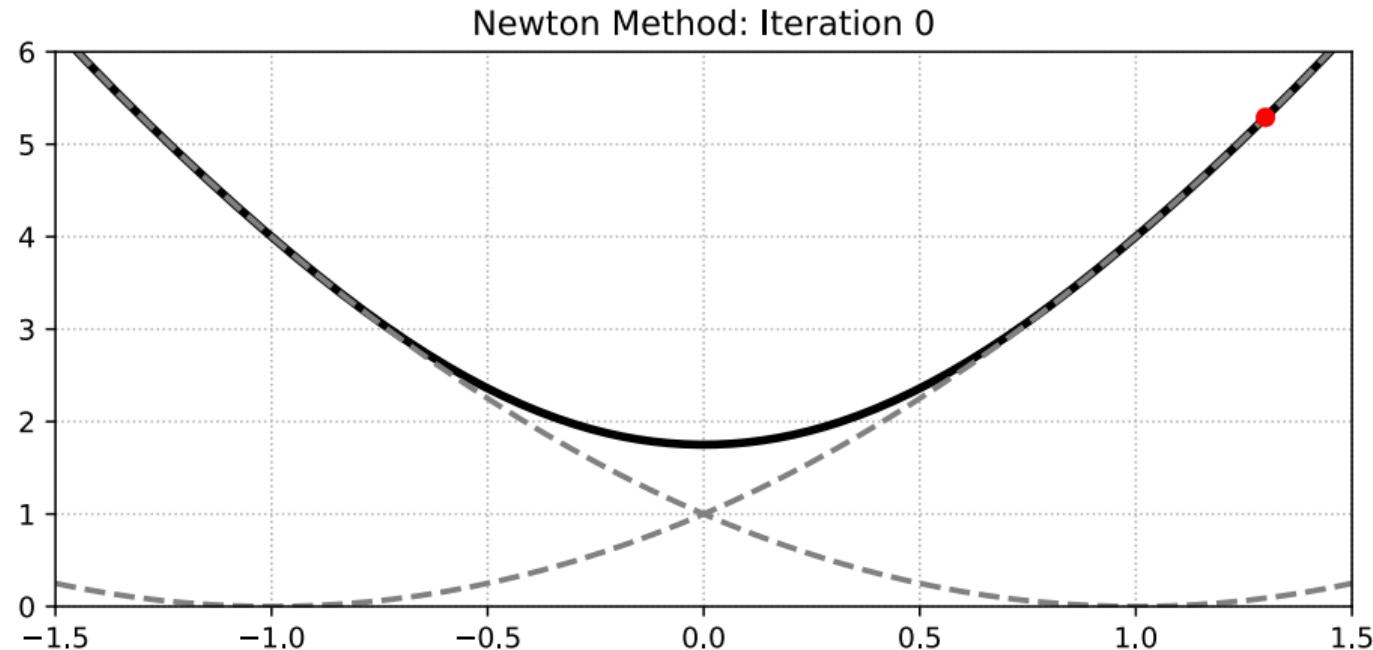
## Локальная сходимость метода Ньютона. Хорошая инициализация



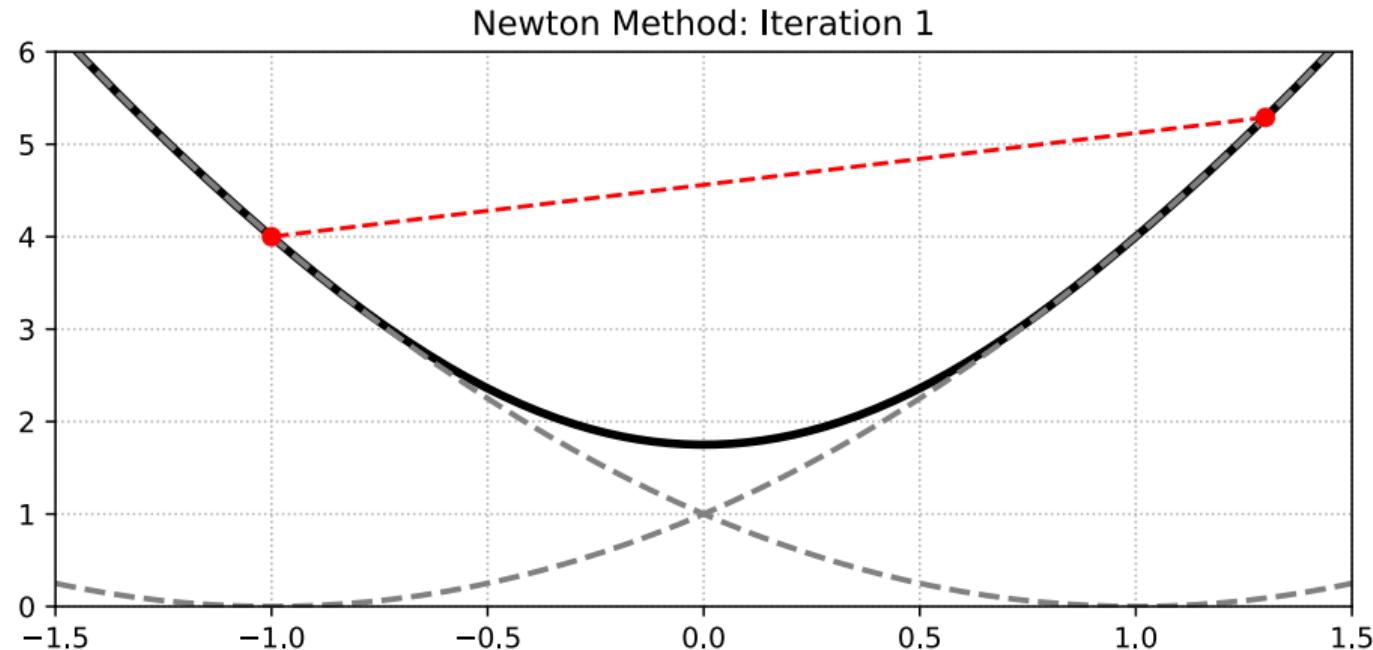
## Локальная сходимость метода Ньютона. Хорошая инициализация



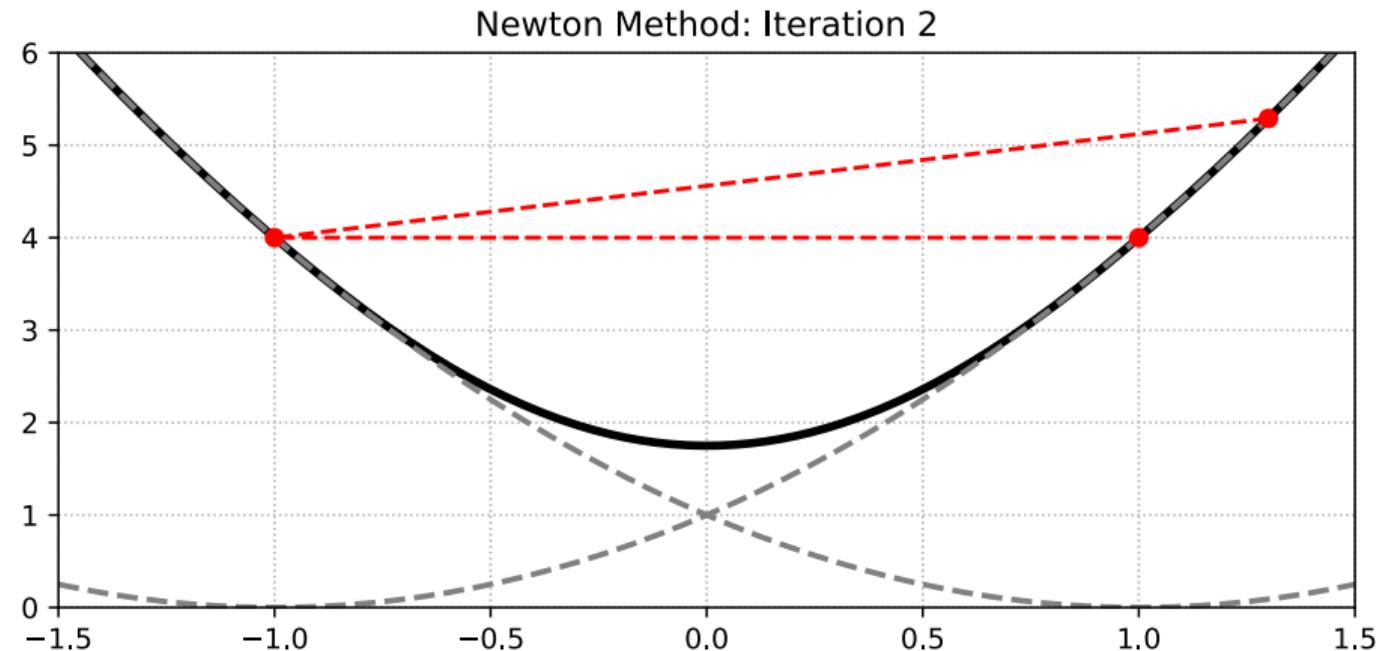
## Локальная сходимость метода Ньютона. Плохая инициализация



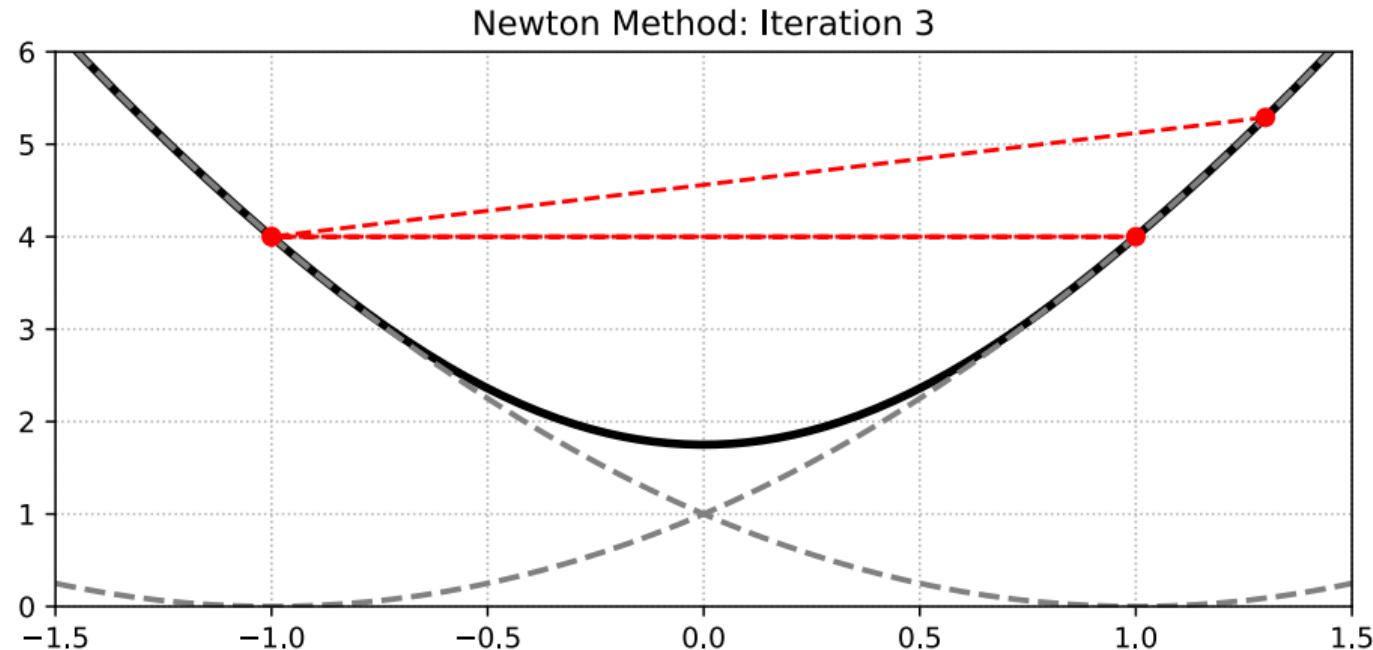
## Локальная сходимость метода Ньютона. Плохая инициализация



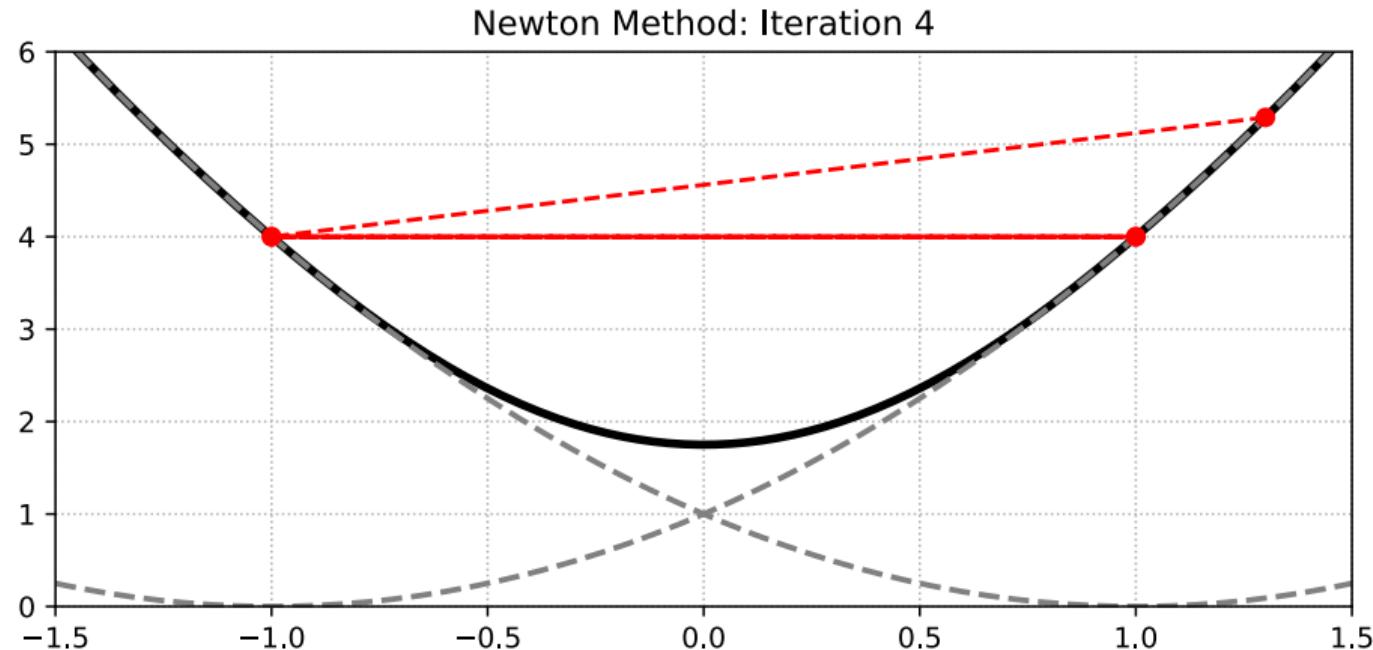
## Локальная сходимость метода Ньютона. Плохая инициализация



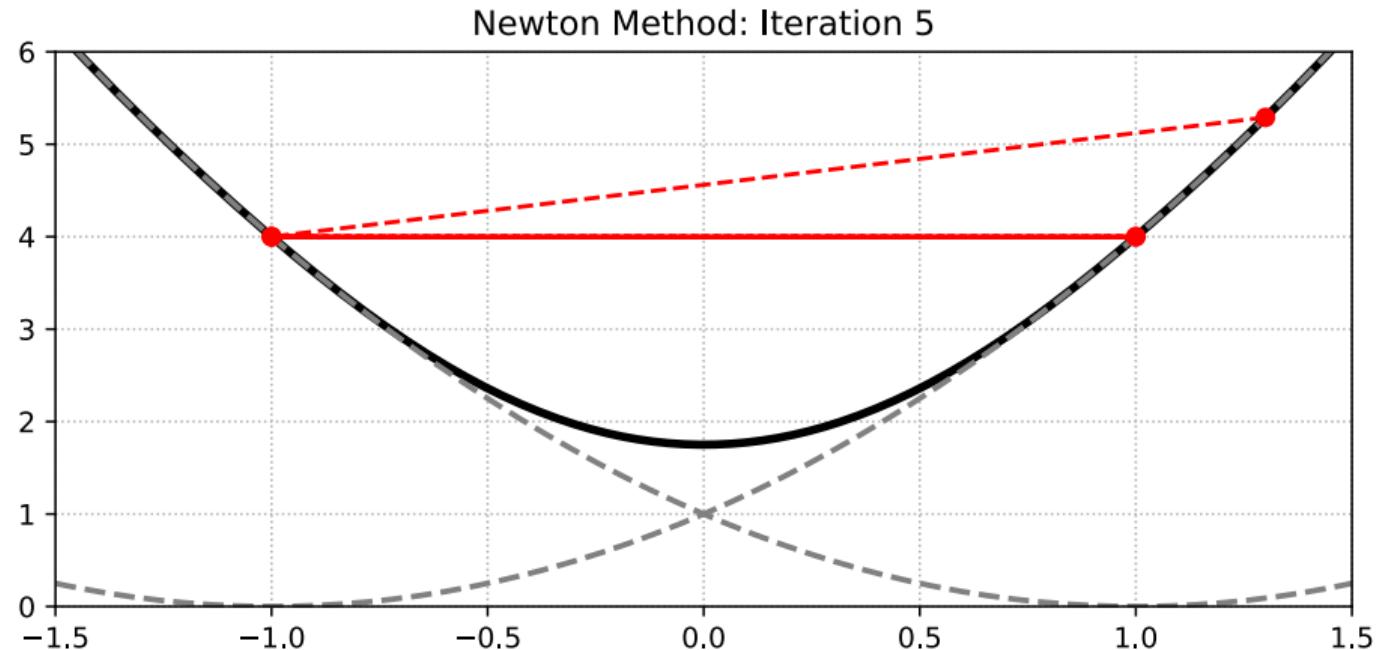
## Локальная сходимость метода Ньютона. Плохая инициализация



## Локальная сходимость метода Ньютона. Плохая инициализация



## Локальная сходимость метода Ньютона. Плохая инициализация



# Newton

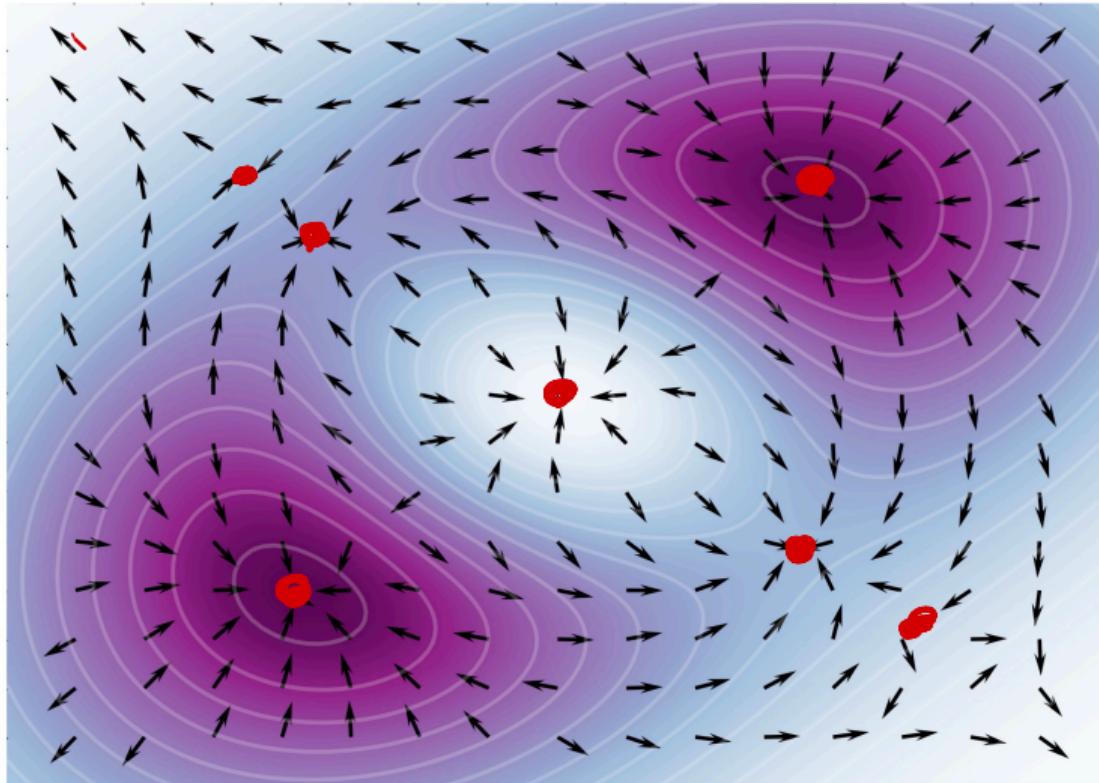
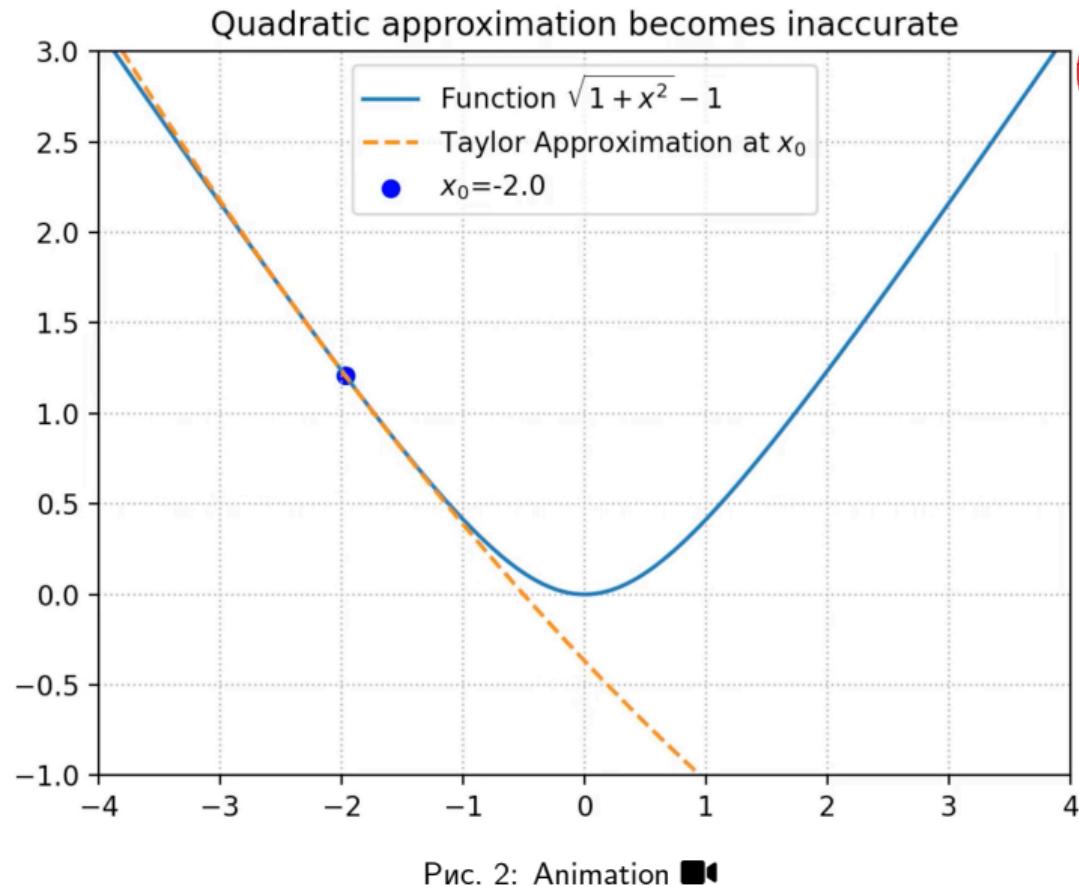


Рис. 1: Animation

## Проблемы метода Ньютона



$$f'(x)=0$$

# Метод Ньютона для квадратичной задачи (линейной регрессии)

$n \approx 10^{12}$

$$x_{k+1} = x_k - \Delta f(x_k)$$

$$x_{k+1} = x_k - [\nabla^2 f(x)]^{-1} \nabla f(x)$$

• ПАМЯТЬ:  $O(n^2)$

• ВРЕМЯ:  $O(n^3)$

$n^3 \approx 10^{36}$

• ПАМЯТЬ:  $O(n)$

• ВРЕМЯ:  $O(n)$

Собр. задачи

DeepSeek GTH

Quinn ~ 220B

Ким K25

ЛТ

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^\top A x - b^\top x, \quad A \in \mathbb{R}^{n \times n}, \quad \lambda(A) \in [\mu; L].$$

Strongly convex quadratics:  $n=60$ , random matrix,  $\mu=1$ ,  $L=10$

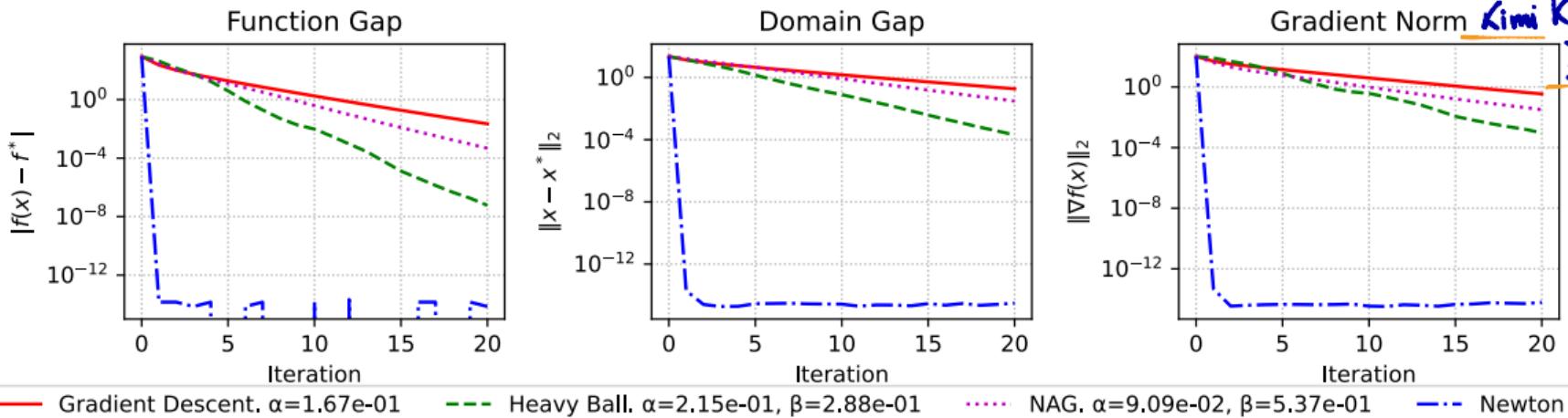


Рис. 3: Так как задача - квадратичная, то метод Ньютона сходится за один шаг.

## Метод Ньютона для квадратичной задачи (линейной регрессии)

$$\nabla^2 f(x_k) [x_{k+1} - x_k] = -\nabla f(x_k)$$

$$x_{k+1} = \dots$$

напр. Newton реш. СЛУ  
получают как

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^\top A x - b^\top x, \quad A \in \mathbb{R}^{n \times n}, \quad \lambda(A) \in [\mu; L].$$

Convex quadratics:  $n=60$ , random matrix,  $\mu=0$ ,  $L=10$

$$\|x_k - x^*\|_2$$

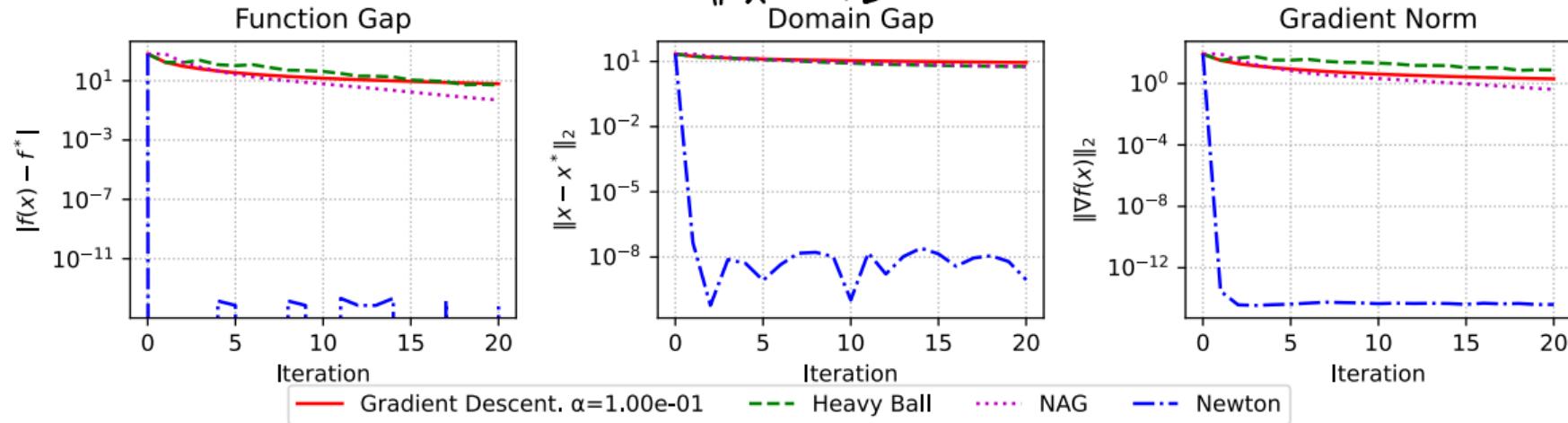


Рис. 4: В этом случае метод Ньютона тоже крайне быстро сходится, однако, отметим, что это происходит благодаря тому, что минимальное собственное число гессиана не 0, а около  $10^{-8}$ . Если применять метод Ньютона в наивной форме с обращением матрицы, то получится ошибка, так как матрица вырождена. На практике все равно можно использовать метод, если для направления итерации решать линейную систему  $\nabla^2 f(x_k) d_k = -\nabla f(x_k)$  методом наименьших квадратов.

# Метод Ньютона для квадратичной задачи (линейной регрессии)

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^\top A x - b^\top x, \quad A \in \mathbb{R}^{n \times n}, \quad \lambda(A) \in [\mu; L].$$

Strongly convex quadratics:  $n=60$ , random matrix,  $\mu=1$ ,  $L=1000$

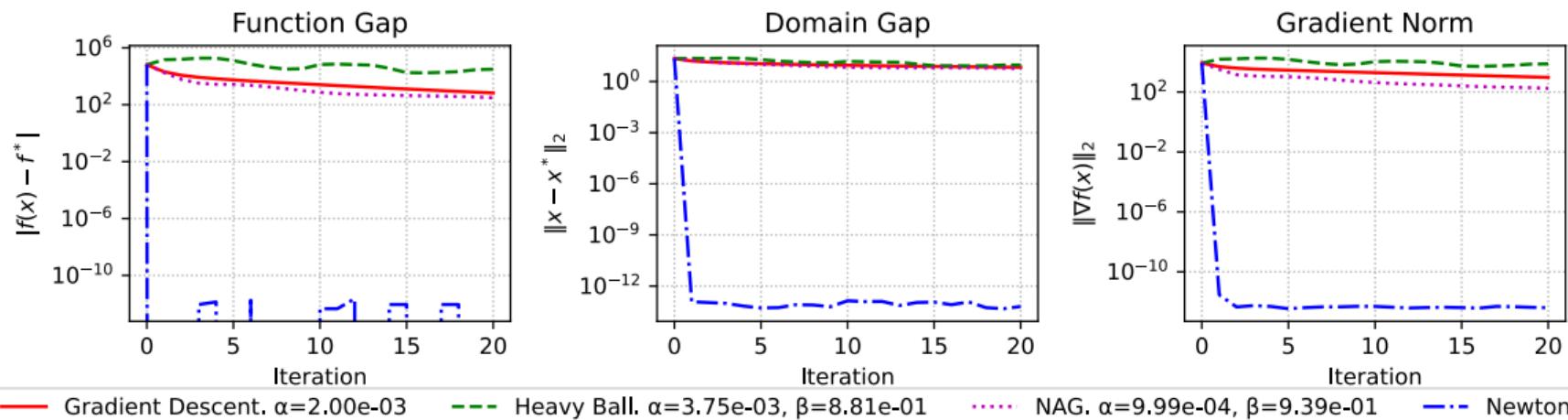


Рис. 5: Здесь число обусловленности гессиана в 1000 раз больше, чем в предыдущем случае, и метод Ньютона сходится за 1 итерацию.

# Метод Ньютона для задачи бинарной логистической регрессии

Convex binary logistic regression.  $\mu=0$ ,  $m=1000$ ,  $n=10$ .

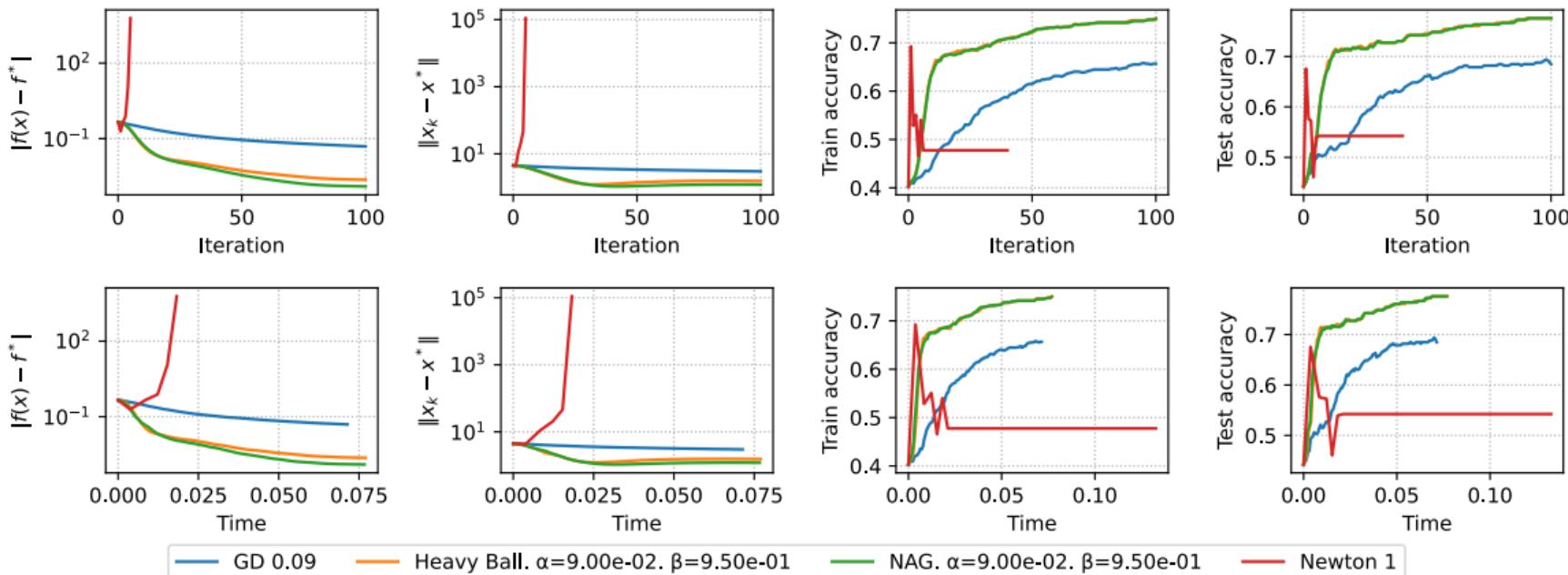


Рис. 6: Наблюдается расходимость метода Ньютона. Сразу отметим, что в задаче нет регуляризации и гарантии сильной выпуклости. А также нет гарантий того, что мы инициализируем метод в окрестности решения.

# Метод Ньютона для задачи бинарной логистической регрессии

Strongly convex binary logistic regression.  $\mu=0.2$ .  $m=1000$ ,  $n=10$ .

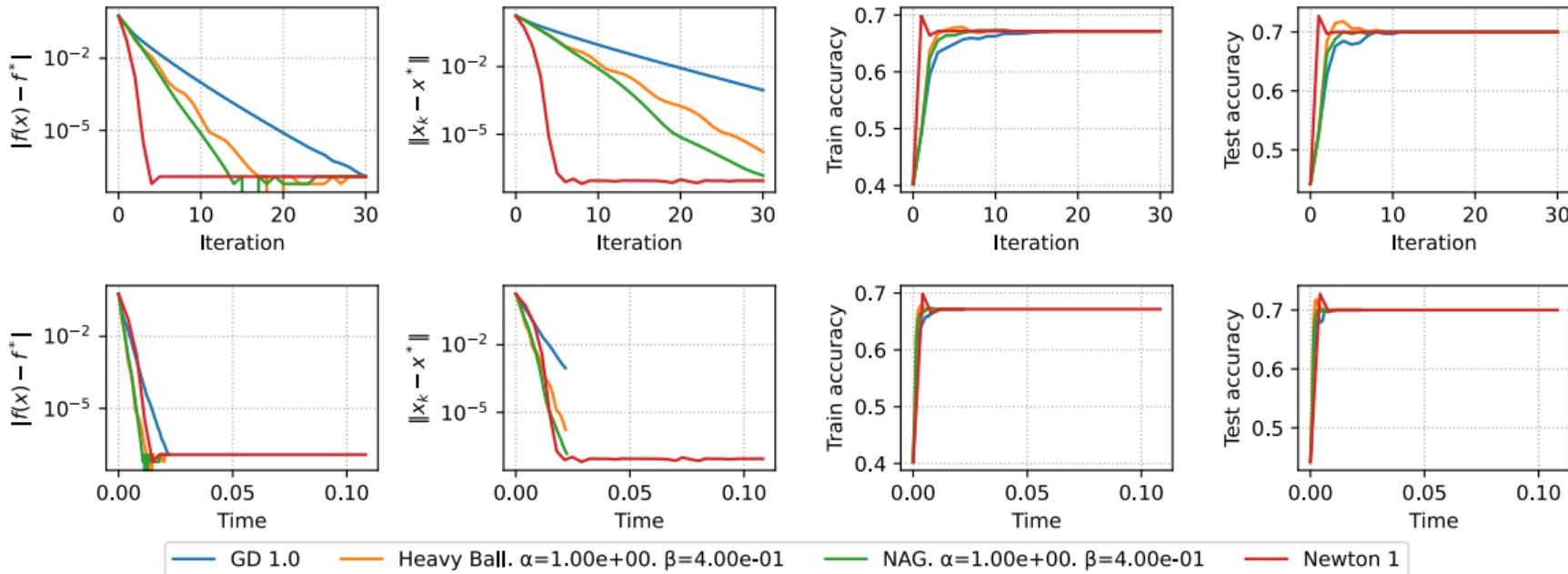


Рис. 7: Добавление регуляризации гарантирует сильную выпуклость, наблюдается сходимость метода Ньютона.

# Метод Ньютона для задачи бинарной логистической регрессии

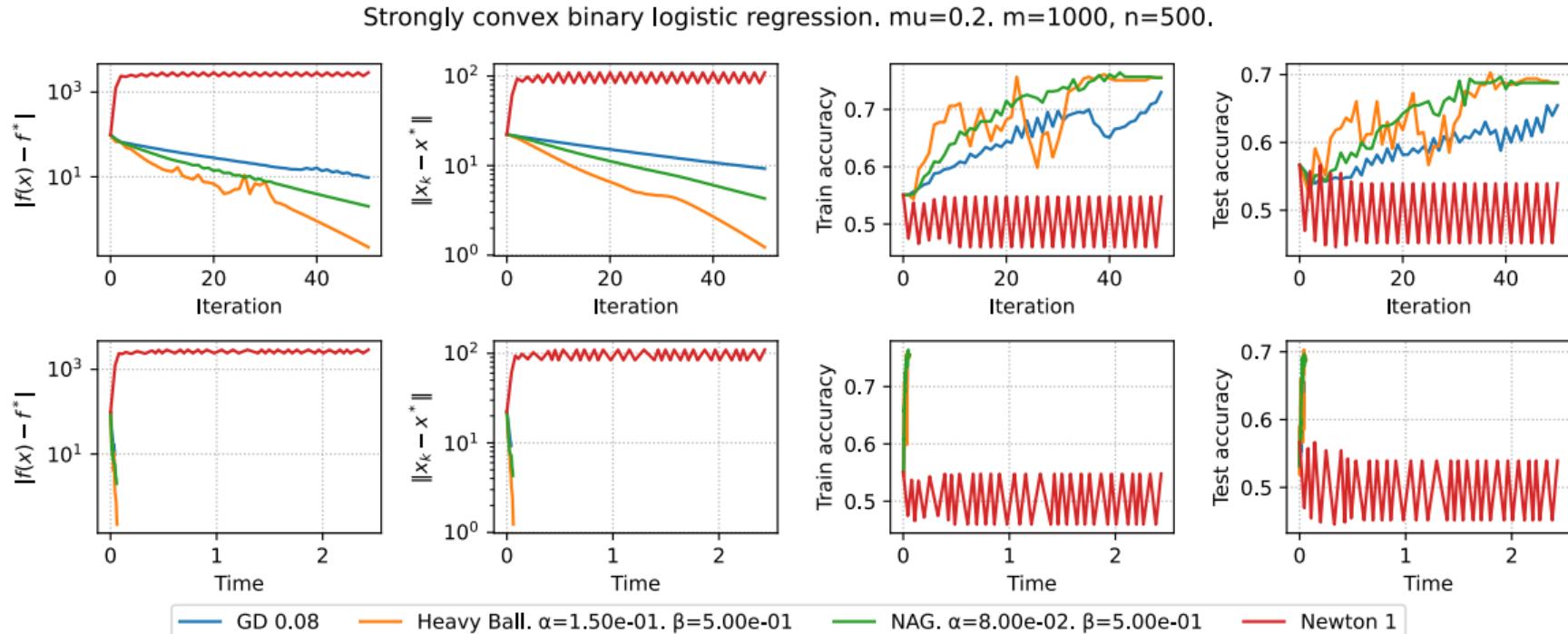


Рис. 8: Увеличим размерность в 50 раз и наблюдаем расходимость метода Ньютона. Это можно связать с тем, что мы инициализируем метод в точке, далекой от решения

# Метод Ньютона для задачи бинарной логистической регрессии

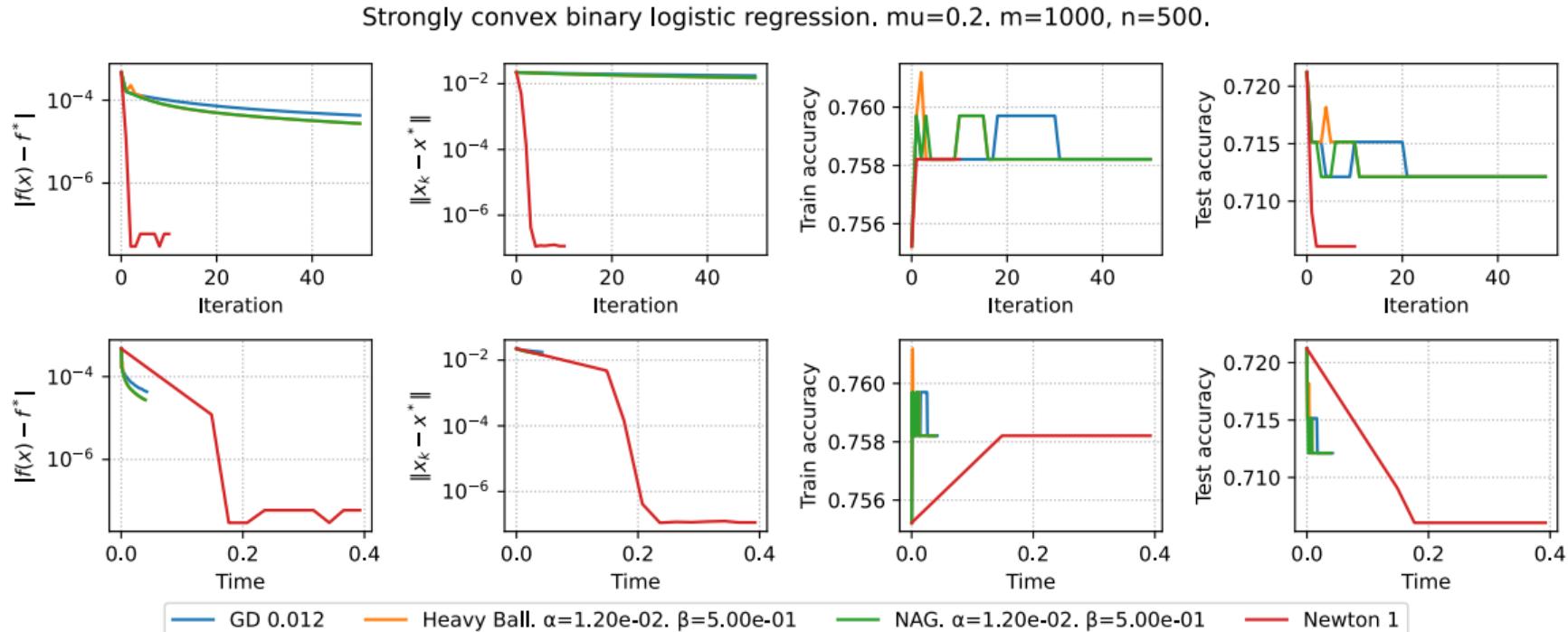


Рис. 9: Не меняя задачу, изменим начальную точку и наблюдаем квадратичную сходимость метода Ньютона. Однако, обратите внимание на время работы. Уже при небольшой размерности, метод Ньютона работает значительно дольше, чем градиентные методы.

## Задача нахождения аналитического центра многогранника

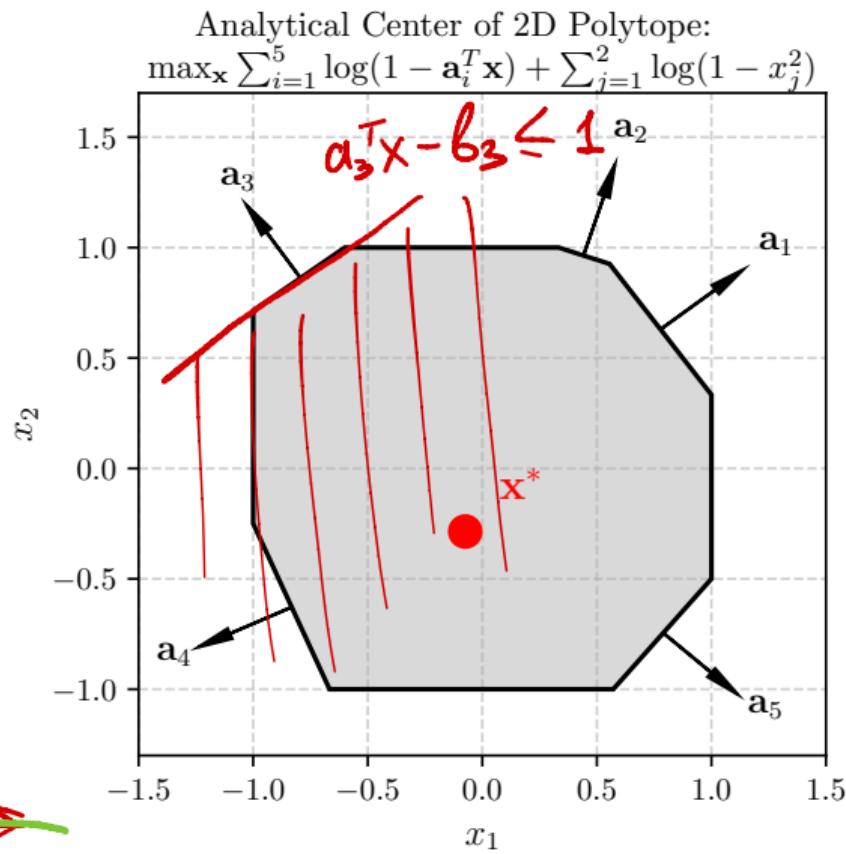
Найти точку  $x \in \mathbb{R}^n$ , которая максимизирует сумму логарифмов расстояний до границ полигонопа:

$$\max_x \sum_{i=1}^m \log(1 - a_i^T x) + \sum_{j=1}^n \log(1 - x_j^2)$$

или, эквивалентно, минимизирует:

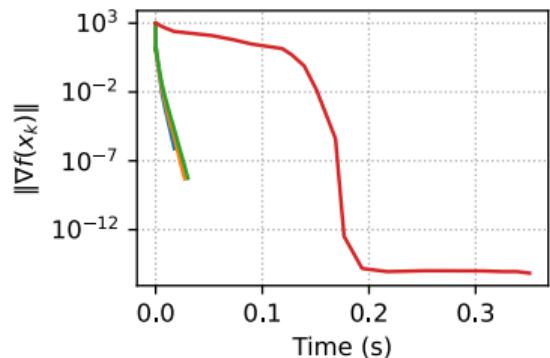
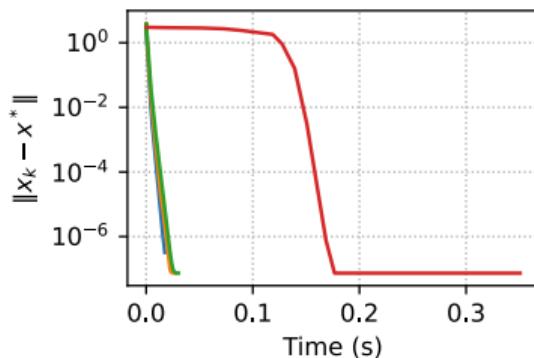
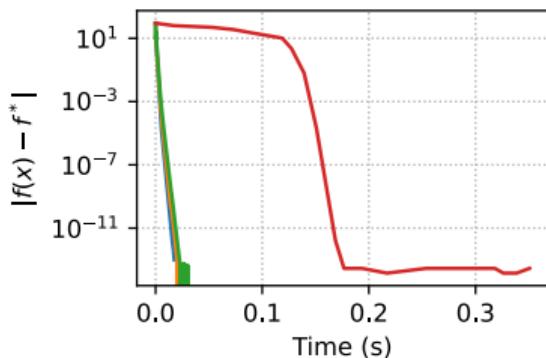
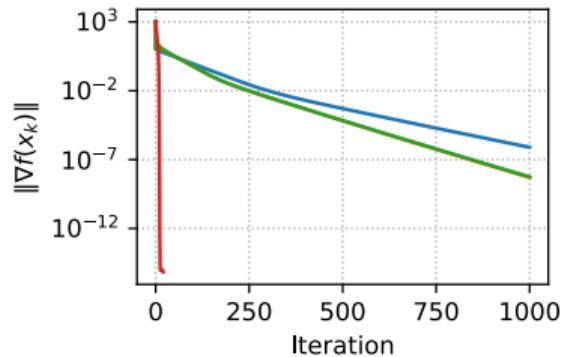
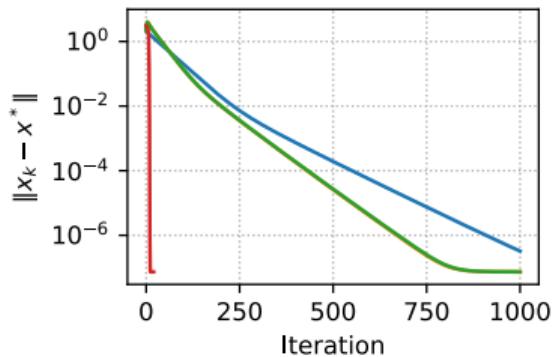
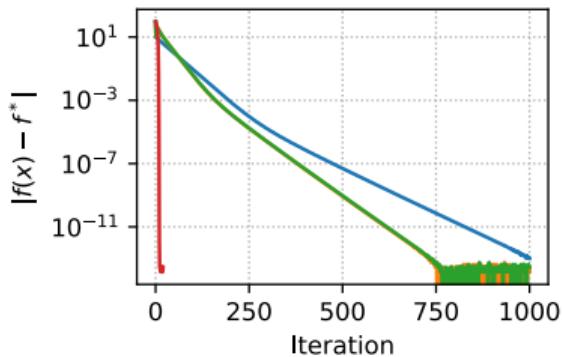
$$\min_x - \sum_{i=1}^m \log(1 - a_i^T x) - \sum_{j=1}^n \log(1 - x_j^2)$$

при ограничениях:  $-a_i^T x < 1$  для всех  $i = 1, \dots, m$ , где  $a_i$  - строки матрицы  $A^T$ ;  $|x_j| < 1$  для всех  $j = 1, \dots, n$   
 Аналитический центр многогранника - это точка, которая максимально удалена от всех границ многогранника в смысле логарифмического барьера. Эта концепция широко используется в методах внутренней точки для выпуклой оптимизации.



# Задача нахождения аналитического центра многогранника

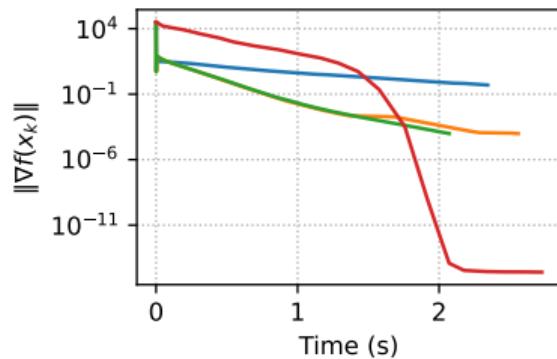
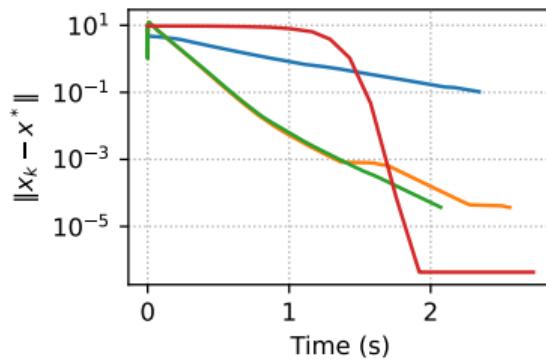
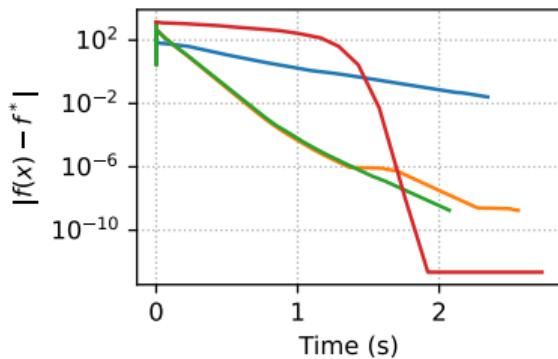
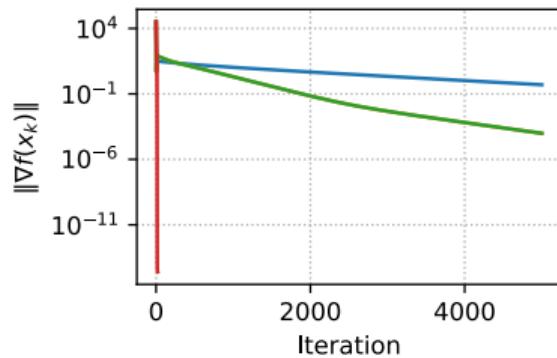
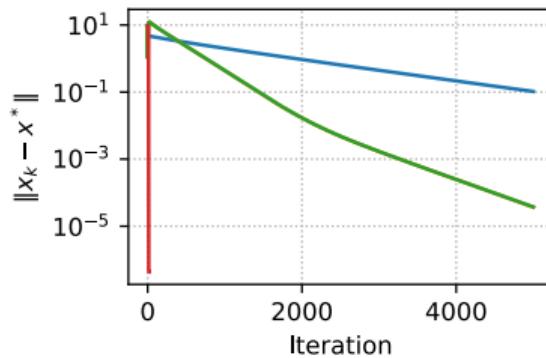
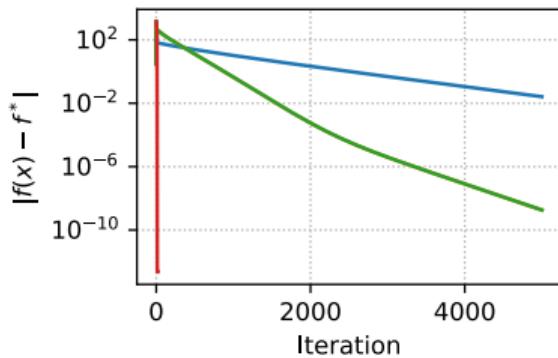
Analytical Center,  $m = 20, n = 100$



— GD,  $\alpha=0.005$    — Heavy Ball,  $\alpha=0.005, \beta=0.33$    — NAG,  $\alpha=0.005, \beta=0.33$    — Newton, damping=1.0

# Задача нахождения аналитического центра многогранника

Analytical Center,  $m = 200$ ,  $n = 1000$



— GD,  $\alpha=0.00015$

— Heavy Ball,  $\alpha=0.00015$ ,  $\beta=0.79$

— NAG,  $\alpha=0.00015$ ,  $\beta=0.79$

— Newton, damping=1.0

## Аффинная инвариантность метода Ньютона

Важным свойством метода Ньютона является **аффинная инвариантность**. Пусть дана функция  $f$  и невырожденная матрица  $A \in \mathbb{R}^{n \times n}$ , пусть  $x = Ay$ , и пусть  $g(y) = f(Ay)$ . Заметим, что  $\nabla g(y) = A^T \nabla f(x)$  и  $\nabla^2 g(y) = A^T \nabla^2 f(x)A$ . Шаги Ньютона на  $q$  выражаются как:

$$y_{k+1} = y_k - (\nabla^2 g(y_k))^{-1} \nabla g(y_k)$$

$$y = A^{-1}x$$

## Аффинная инвариантность метода Ньютона

Важным свойством метода Ньютона является **аффинная инвариантность**. Пусть дана функция  $f$  и невырожденная матрица  $A \in \mathbb{R}^{n \times n}$ , пусть  $x = Ay$ , и пусть  $g(y) = f(Ay)$ . Заметим, что  $\nabla g(y) = A^T \nabla f(x)$  и  $\nabla^2 g(y) = A^T \nabla^2 f(x)A$ . Шаги Ньютона на  $g$  выражаются как:

$$y_{k+1} = y_k - (\nabla^2 g(y_k))^{-1} \nabla g(y_k)$$

Раскрывая это, мы получаем:

$$\begin{aligned} y_{k+1} &= y_k - \underbrace{(A^T \nabla^2 f(Ay_k) A)^{-1}}_{\text{A}^{-1} \cdot (\text{A}^T \nabla^2 f(\text{Ay}_k))^{\text{-1}}} \underbrace{A^T \nabla f(Ay_k)}_{\text{A}^{-1} \cdot (\nabla f(\text{Ay}_k))^{\text{-1}} \cdot \underbrace{\text{A}^T \cdot \text{A}^T \cdot \nabla f(\text{Ay}_k)}_{\text{I}}} \\ &\quad \text{A}^{-1} \cdot (\nabla f(\text{Ay}_k))^{\text{-1}} \cdot \underbrace{\text{A}^T \cdot \text{A}^T \cdot \nabla f(\text{Ay}_k)}_{\text{I}} \end{aligned}$$

## Аффинная инвариантность метода Ньютона

Важным свойством метода Ньютона является **аффинная инвариантность**. Пусть дана функция  $f$  и невырожденная матрица  $A \in \mathbb{R}^{n \times n}$ , пусть  $x = Ay$ , и пусть  $g(y) = f(Ay)$ . Заметим, что  $\nabla g(y) = A^T \nabla f(x)$  и  $\nabla^2 g(y) = A^T \nabla^2 f(x)A$ . Шаги Ньютона на  $g$  выражаются как:

$$y_{k+1} = y_k - (\nabla^2 g(y_k))^{-1} \nabla g(y_k)$$

$$y_{k+1} = y_k - \nabla g(y_k)$$

Раскрывая это, мы получаем:

$$y_{k+1} = y_k - (A^T \nabla^2 f(Ay_k) A)^{-1} A^T \nabla f(Ay_k)$$

$$y_{k+1} = y_k - A^T \nabla f(Ay_k)$$

Используя свойство обратной матрицы  $(AB)^{-1} = B^{-1}A^{-1}$ , это упрощается до:

$$y_{k+1} = y_k - A^{-1} (\nabla^2 f(Ay_k))^{-1} \nabla f(Ay_k)$$

$$x_{k+1} = x_k - A A^T \nabla f(x_k)$$

$$Ay_{k+1} = Ay_k - (\nabla^2 f(Ay_k))^{-1} \nabla f(Ay_k)$$

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

## Аффинная инвариантность метода Ньютона

Важным свойством метода Ньютона является **аффинная инвариантность**. Пусть дана функция  $f$  и невырожденная матрица  $A \in \mathbb{R}^{n \times n}$ , пусть  $x = Ay$ , и пусть  $g(y) = f(Ay)$ . Заметим, что  $\nabla g(y) = A^T \nabla f(x)$  и  $\nabla^2 g(y) = A^T \nabla^2 f(x)A$ . Шаги Ньютона на  $g$  выражаются как:

$$y_{k+1} = y_k - (\nabla^2 g(y_k))^{-1} \nabla g(y_k)$$

Раскрывая это, мы получаем:

$$y_{k+1} = y_k - (A^T \nabla^2 f(Ay_k) A)^{-1} A^T \nabla f(Ay_k)$$

Используя свойство обратной матрицы  $(AB)^{-1} = B^{-1}A^{-1}$ , это упрощается до:

$$y_{k+1} = y_k - A^{-1} (\nabla^2 f(Ay_k))^{-1} \nabla f(Ay_k)$$

$$Ay_{k+1} = Ay_k - (\nabla^2 f(Ay_k))^{-1} \nabla f(Ay_k)$$

Таким образом, правило обновления для  $x$  выглядит так:

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

## Аффинная инвариантность метода Ньютона

Важным свойством метода Ньютона является **аффинная инвариантность**. Пусть дана функция  $f$  и невырожденная матрица  $A \in \mathbb{R}^{n \times n}$ , пусть  $x = Ay$ , и пусть  $g(y) = f(Ay)$ . Заметим, что  $\nabla g(y) = A^T \nabla f(x)$  и  $\nabla^2 g(y) = A^T \nabla^2 f(x)A$ . Шаги Ньютона на  $g$  выражаются как:

$$y_{k+1} = y_k - (\nabla^2 g(y_k))^{-1} \nabla g(y_k)$$

Раскрывая это, мы получаем:

$$y_{k+1} = y_k - (A^T \nabla^2 f(Ay_k) A)^{-1} A^T \nabla f(Ay_k)$$

Используя свойство обратной матрицы  $(AB)^{-1} = B^{-1}A^{-1}$ , это упрощается до:

$$y_{k+1} = y_k - A^{-1} (\nabla^2 f(Ay_k))^{-1} \nabla f(Ay_k)$$

$$Ay_{k+1} = Ay_k - (\nabla^2 f(Ay_k))^{-1} \nabla f(Ay_k)$$

Таким образом, правило обновления для  $x$  выглядит так:

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

Это показывает, что итерация метода Ньютона не зависит от масштаба задачи. У градиентного спуска такого свойства нет!

## Итоги

Плюсы:

- Квадратичная сходимость вблизи решения  $x^*$

Минусы:

## Итоги

Плюсы:

- Квадратичная сходимость вблизи решения  $x^*$
- Аффинная инвариантность

Минусы:

## Итоги

Плюсы:

- Квадратичная сходимость вблизи решения  $x^*$
- Аффинная инвариантность
- Отсутствие параметров у метода

Минусы:

## Итоги

Плюсы:

- Квадратичная сходимость вблизи решения  $x^*$
- Аффинная инвариантность
- Отсутствие параметров у метода
- Сходимость можно сделать глобальной, если использовать демпфированный метод Ньютона (добавить процедуру линейного поиска и шага метода)

Минусы:

$$x_{k+1} = x_k - \alpha_k \begin{bmatrix} \nabla^2 f(x_k) \end{bmatrix}^{-1} \nabla f(x_k)$$

# Итоги

Плюсы:

- Квадратичная сходимость вблизи решения  $x^*$
- Аффинная инвариантность
- Отсутствие параметров у метода
- Сходимость можно сделать глобальной, если использовать демпфированный метод Ньютона (добавить процедуру линейного поиска и шага метода)

Минусы:

- Необходимо хранить (обратный) гессиан на каждой итерации:  $\mathcal{O}(n^2)$  памяти

## Итоги

# Newton-CG

Плюсы:

- Квадратичная сходимость вблизи решения  $x^*$
- Аффинная инвариантность
- Отсутствие параметров у метода
- Сходимость можно сделать глобальной, если использовать демпфированный метод Ньютона (добавить процедуру линейного поиска и шага метода)

Минусы:

- Необходимо хранить (обратный) гессиан на каждой итерации:  $\mathcal{O}(n^2)$  памяти
- Необходимо решать линейные системы:  $\mathcal{O}(n^3)$  операций

# Итоги

Плюсы:

- Квадратичная сходимость вблизи решения  $x^*$
- Аффинная инвариантность
- Отсутствие параметров у метода
- Сходимость можно сделать глобальной, если использовать демпфированный метод Ньютона (добавить процедуру линейного поиска и шага метода)

Минусы:

- Необходимо хранить (обратный) гессиан на каждой итерации:  $\mathcal{O}(n^2)$  памяти
- Необходимо решать линейные системы:  $\mathcal{O}(n^3)$  операций
- Гессиан может быть вырожденным в  $x^*$

# Итоги

Плюсы:

- Квадратичная сходимость вблизи решения  $x^*$
- Аффинная инвариантность
- Отсутствие параметров у метода
- Сходимость можно сделать глобальной, если использовать демпфированный метод Ньютона (добавить процедуру линейного поиска и шага метода)

Минусы:

- Необходимо хранить (обратный) гессиан на каждой итерации:  $\mathcal{O}(n^2)$  памяти
- Необходимо решать линейные системы:  $\mathcal{O}(n^3)$  операций
- Гессиан может быть вырожденным в  $x^*$
- Гессиан может не быть положительно определенным → направление  $-(f''(x))^{-1}f'(x)$  может не быть направлением спуска

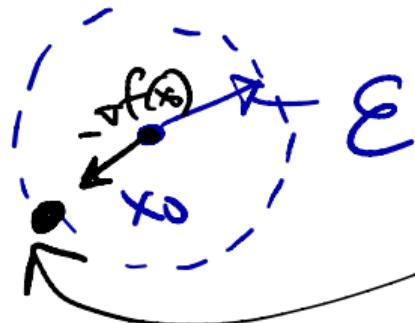
$$\nabla^2 f(x_k) = I \Rightarrow GD$$

### Квазиньютоновские методы

$$x_{k+1} = x_k - d_k \left( \nabla^2 f(x_k) \right)^{-1} \cdot \nabla f(x_k)$$

## Идея методов адаптивной метрики

Пусть дана функция  $f(x)$  и точка  $x_0$ . Определим  $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$  как множество точек с расстоянием  $\varepsilon$  до  $x_0$ . Здесь мы предполагаем существование функции расстояния  $d(x, x_0)$ .



$$\min_{x \in \mathbb{R}^n} f(x)$$
$$\|x - x_0\|_2^2 = \varepsilon^2$$

$$(x - x_0)^T A (x - x_0) = \varepsilon^2$$

## Идея методов адаптивной метрики

Пусть дана функция  $f(x)$  и точка  $x_0$ . Определим  $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$  как множество точек с расстоянием  $\varepsilon$  до  $x_0$ . Здесь мы предполагаем существование функции расстояния  $d(x, x_0)$ .

$$x^* = \arg \min_{x \in B_\varepsilon(x_0)} f(x)$$

## Идея методов адаптивной метрики

Пусть дана функция  $f(x)$  и точка  $x_0$ . Определим  $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$  как множество точек с расстоянием  $\varepsilon$  до  $x_0$ . Здесь мы предполагаем существование функции расстояния  $d(x, x_0)$ .

$$x^* = \arg \min_{x \in B_\varepsilon(x_0)} f(x)$$

Далее, мы можем определить другое направление наискорейшего спуска в терминах минимизатора функции на сфере:

## Идея методов адаптивной метрики

Пусть дана функция  $f(x)$  и точка  $x_0$ . Определим  $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$  как множество точек с расстоянием  $\varepsilon$  до  $x_0$ . Здесь мы предполагаем существование функции расстояния  $d(x, x_0)$ .

$$x^* = \arg \min_{x \in B_\varepsilon(x_0)} f(x)$$

Далее, мы можем определить другое направление наискорейшего спуска в терминах минимизатора функции на сфере:

$$s = \lim_{\varepsilon \rightarrow 0} \frac{x^* - x_0}{\varepsilon}$$

## Идея методов адаптивной метрики

Пусть дана функция  $f(x)$  и точка  $x_0$ . Определим  $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$  как множество точек с расстоянием  $\varepsilon$  до  $x_0$ . Здесь мы предполагаем существование функции расстояния  $d(x, x_0)$ .

$$x^* = \arg \min_{x \in B_\varepsilon(x_0)} f(x)$$

Далее, мы можем определить другое направление наискорейшего спуска в терминах минимизатора функции на сфере:

$$s = \lim_{\varepsilon \rightarrow 0} \frac{x^* - x_0}{\varepsilon}$$

Предположим, что расстояние локально определяется некоторой метрикой  $A$ :

$$d(x, x_0) = (x - x_0)^\top A(x - x_0)$$

## Идея методов аддативной метрики

Пусть дана функция  $f(x)$  и точка  $x_0$ . Определим  $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$  как множество точек с расстоянием  $\varepsilon$  до  $x_0$ . Здесь мы предполагаем существование функции расстояния  $d(x, x_0)$ .

$$x^* = \arg \min_{x \in B_\varepsilon(x_0)} f(x)$$

Далее, мы можем определить другое направление наискорейшего спуска в терминах минимизатора функции на сфере:

$$s = \lim_{\varepsilon \rightarrow 0} \frac{x^* - x_0}{\varepsilon}$$

Предположим, что расстояние локально определяется некоторой метрикой  $A$ :

$$d(x, x_0) = (x - x_0)^\top A(x - x_0)$$

Далее рассмотрим первый порядок аппроксимации функции  $f(x)$  в окрестности точки  $x_0$ :

$$\bullet f \rightarrow \min_{x, y, z} \quad f(x_0 + \delta x) \approx f(x_0) + \nabla f(x_0)^\top \delta x \quad (1)$$

Теперь мы можем сформулировать задачу нахождения  $s$ , как это было сказано выше.

$$\begin{aligned} & \min_{\delta x \in \mathbb{R}^n} f(x_0 + \delta x) \\ \text{s.t. } & \delta x^\top A \delta x = \varepsilon^2 \end{aligned}$$

## Идея методов аддативной метрики

Пусть дана функция  $f(x)$  и точка  $x_0$ . Определим  $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon\}$  как множество точек с расстоянием  $\varepsilon$  до  $x_0$ . Здесь мы предполагаем существование функции расстояния  $d(x, x_0)$ .

$$x^* = \arg \min_{x \in B_\varepsilon(x_0)} f(x)$$

Далее, мы можем определить другое направление наискорейшего спуска в терминах минимизатора функции на сфере:

$$s = \lim_{\varepsilon \rightarrow 0} \frac{x^* - x_0}{\varepsilon}$$

Предположим, что расстояние локально определяется некоторой метрикой  $A$ :

$$d(x, x_0) = (x - x_0)^\top A(x - x_0)$$

Далее рассмотрим первый порядок аппроксимации функции  $f(x)$  в окрестности точки  $x_0$ :

$$f(x_0 + \delta x) \approx f(x_0) + \nabla f(x_0)^\top \delta x$$

Теперь мы можем сформулировать задачу нахождения  $s$ , как это было сказано выше.

$$\min_{\delta x \in \mathbb{R}^n} f(x_0 + \delta x)$$

$$\text{s.t. } \delta x^\top A \delta x = \varepsilon^2$$

по ВАЙБИМ

Используя уравнение 1, получаем:

$$\min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x$$

$$\text{s.t. } \delta x^\top A \delta x = \varepsilon^2$$

$$L = \nabla f(x_0)^\top \delta x + \lambda (\delta x^\top A \delta x - \varepsilon^2)$$

$$\frac{\partial L}{\partial \delta x} = \nabla f(x_0) + 2\lambda A \delta x = 0$$

$$\delta x^\top A \delta x = \varepsilon^2$$

$$A \delta x = -\frac{1}{2\lambda} \nabla f(x_0)$$

$$\delta x = -\frac{1}{2\lambda} A^{-1} \nabla f(x_0)$$

$$\lambda = \dots$$

## Идея методов аддативной метрики

Пусть дана функция  $f(x)$  и точка  $x_0$ . Определим  $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon\}$  как множество точек с расстоянием  $\varepsilon$  до  $x_0$ . Здесь мы предполагаем существование функции расстояния  $d(x, x_0)$ .

$$x^* = \arg \min_{x \in B_\varepsilon(x_0)} f(x)$$

Далее, мы можем определить другое направление наискорейшего спуска в терминах минимизатора функции на сфере:

$$s = \lim_{\varepsilon \rightarrow 0} \frac{x^* - x_0}{\varepsilon}$$

Предположим, что расстояние локально определяется некоторой метрикой  $A$ :

$$d(x, x_0) = (x - x_0)^\top A(x - x_0)$$

Далее рассмотрим первый порядок аппроксимации функции  $f(x)$  в окрестности точки  $x_0$ :

$$f(x_0 + \delta x) \approx f(x_0) + \nabla f(x_0)^\top \delta x \quad (1)$$

Теперь мы можем сформулировать задачу нахождения  $s$ , как это было сказано выше.

$$\begin{aligned} & \min_{\delta x \in \mathbb{R}^n} f(x_0 + \delta x) \\ \text{s.t. } & \delta x^\top A \delta x = \varepsilon^2 \end{aligned}$$

Используя уравнение 1, получаем:

$$\begin{aligned} & \min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x \\ \text{s.t. } & \delta x^\top A \delta x = \varepsilon^2 \end{aligned}$$

Используя метод множителей Лагранжа:

$$\delta x = -\frac{2\varepsilon^2}{\nabla f(x_0)^\top A^{-1} \nabla f(x_0)} A^{-1} \nabla f(x_0)$$

Метод Зеркального Спуска

## Идея методов аддативной метрики

Пусть дана функция  $f(x)$  и точка  $x_0$ . Определим  $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$  как множество точек с расстоянием  $\varepsilon$  до  $x_0$ . Здесь мы предполагаем существование функции расстояния  $d(x, x_0)$ .

$$x^* = \arg \min_{x \in B_\varepsilon(x_0)} f(x)$$

Далее, мы можем определить другое направление наискорейшего спуска в терминах минимизатора функции на сфере:

$$s = \lim_{\varepsilon \rightarrow 0} \frac{x^* - x_0}{\varepsilon}$$

Предположим, что расстояние локально определяется некоторой метрикой  $A$ :

$$d(x, x_0) = (x - x_0)^\top A(x - x_0)$$

Далее рассмотрим первый порядок аппроксимации функции  $f(x)$  в окрестности точки  $x_0$ :

$$\bullet f \rightarrow \min_{x,y,z} \quad f(x_0 + \delta x) \approx f(x_0) + \nabla f(x_0)^\top \delta x \quad (1)$$

Теперь мы можем сформулировать задачу нахождения  $s$ , как это было сказано выше.

$$\begin{aligned} & \min_{\delta x \in \mathbb{R}^n} f(x_0 + \delta x) \\ \text{s.t. } & \delta x^\top A \delta x = \varepsilon^2 \end{aligned}$$

Используя уравнение 1, получаем:

$$\begin{aligned} & \min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x \\ \text{s.t. } & \delta x^\top A \delta x = \varepsilon^2 \end{aligned}$$

Используя метод множителей Лагранжа:

$$\delta x = -\frac{2\varepsilon^2}{\nabla f(x_0)^\top A^{-1} \nabla f(x_0)} A^{-1} \nabla f(x_0)$$

Новое направление наискорейшего спуска:  $A^{-1} \nabla f(x_0)$ .

## Идея методов аддативной метрики

Пусть дана функция  $f(x)$  и точка  $x_0$ . Определим  $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$  как множество точек с расстоянием  $\varepsilon$  до  $x_0$ . Здесь мы предполагаем существование функции расстояния  $d(x, x_0)$ .

$$x^* = \arg \min_{x \in B_\varepsilon(x_0)} f(x)$$

Далее, мы можем определить другое направление наискорейшего спуска в терминах минимизатора функции на сфере:

$$s = \lim_{\varepsilon \rightarrow 0} \frac{x^* - x_0}{\varepsilon}$$

Предположим, что расстояние локально определяется некоторой метрикой  $A$ :

$$d(x, x_0) = (x - x_0)^\top A(x - x_0)$$

Далее рассмотрим первый порядок аппроксимации функции  $f(x)$  в окрестности точки  $x_0$ :

$$f(x_0 + \delta x) \approx f(x_0) + \nabla f(x_0)^\top \delta x$$

Теперь мы можем сформулировать задачу нахождения  $s$ , как это было сказано выше.

$$\begin{aligned} & \min_{\delta x \in \mathbb{R}^n} f(x_0 + \delta x) \\ \text{s.t. } & \delta x^\top A \delta x = \varepsilon^2 \end{aligned}$$

Используя уравнение 1, получаем:

$$\begin{aligned} & \min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x \\ \text{s.t. } & \delta x^\top A \delta x = \varepsilon^2 \end{aligned}$$

Используя метод множителей Лагранжа:

$$\delta x = -\frac{2\varepsilon^2}{\nabla f(x_0)^\top A^{-1} \nabla f(x_0)} A^{-1} \nabla f(x_0)$$

Новое направление наискорейшего спуска:  $A^{-1} \nabla f(x_0)$ . Действительно, если пространство изотропно и  $A = I$ , мы сразу получаем формулу градиентного спуска, в то время как метод Ньютона использует локальный гессиан как матрицу метрик.

(1)

## Интуиция квазиньютоновских методов

Для классической задачи безусловной оптимизации  $f(x) \rightarrow \min_{x \in \mathbb{R}^n}$  общий алгоритм итерационного метода записывается как:

$$x_{k+1} = x_k + \alpha_k d_k$$

## Интуиция квазиньютоновских методов

Для классической задачи безусловной оптимизации  $f(x) \rightarrow \min_{x \in \mathbb{R}^n}$  общий алгоритм итерационного метода записывается как:

$$x_{k+1} = x_k + \alpha_k d_k$$

$$d_k = \frac{x_{k+1} - x_k}{\alpha_k}$$

В методе Ньютона направление  $d_k$  (направление Ньютона) устанавливается решением линейной системы на каждом шаге:

$$B_k d_k = -\nabla f(x_k), \quad B_k = \nabla^2 f(x_k)$$

## Интуиция квазиньютоновских методов

Для классической задачи безусловной оптимизации  $f(x) \rightarrow \min_{x \in \mathbb{R}^n}$  общий алгоритм итерационного метода записывается как:

$$x_{k+1} = x_k + \alpha_k d_k$$

В методе Ньютона направление  $d_k$  (направление Ньютона) устанавливается решением линейной системы на каждом шаге:

$$B_k d_k = -\nabla f(x_k), \quad B_k = \nabla^2 f(x_k)$$

т.е. на каждой итерации необходимо **вычислить** гессиан и градиент и **решить** линейную систему.

## Интуиция квазиньютоновских методов

Для классической задачи безусловной оптимизации  $f(x) \rightarrow \min_{x \in \mathbb{R}^n}$  общий алгоритм итерационного метода записывается как:

$$x_{k+1} = x_k + \alpha_k d_k$$

В методе Ньютона направление  $d_k$  (направление Ньютона) устанавливается решением линейной системы на каждом шаге:

$$B_k d_k = -\nabla f(x_k), \quad B_k = \nabla^2 f(x_k)$$

т.е. на каждой итерации необходимо **вычислить** гессиан и градиент и **решить** линейную систему.

Обратите внимание, что если мы возьмем единичную матрицу  $B_k = I_n$  в качестве  $B_k$  на каждом шаге, мы получим в точности метод градиентного спуска.

Общий алгоритм квазиньютоновских методов основан на выборе матрицы  $B_k$  так, чтобы она в некотором смысле стремилась к истинному значению гессиана  $\nabla^2 f(x_k)$  при  $k \rightarrow \infty$ .

## Шаблон квазиньютоновского метода

Пусть  $x_0 \in \mathbb{R}^n$ ,  $B_0 \succ 0$ . Для  $k = 0, 1, 2, \dots$ , повторяем:

1. Решить  $\underline{B_k d_k = -\nabla f(x_k)}$   $\rightarrow d_k = \dots$

## Шаблон квазиньютоновского метода

Пусть  $x_0 \in \mathbb{R}^n$ ,  $B_0 \succ 0$ . Для  $k = 0, 1, 2, \dots$ , повторяем:

1. Решить  $B_k d_k = -\nabla f(x_k)$   $d_k = \cdot \cdot \cdot - -$ .
2. Обновить  $x_{k+1} = x_k + \alpha_k d_k$ .

## Шаблон квазиньютоновского метода

Пусть  $x_0 \in \mathbb{R}^n$ ,  $B_0 \succ 0$ . Для  $k = 0, 1, 2, \dots$ , повторяем:

1. Решить  $B_k d_k = -\nabla f(x_k)$
  2. Обновить  $x_{k+1} = x_k + \alpha_k d_k$
  3. Вычислить  $B_{k+1}$  из  $B_k$
-

## Шаблон квазиньютоновского метода

Пусть  $x_0 \in \mathbb{R}^n$ ,  $B_0 \succ 0$ . Для  $k = 0, 1, 2, \dots$ , повторяем:

1. Решить  $B_k d_k = -\nabla f(x_k)$
2. Обновить  $x_{k+1} = x_k + \alpha_k d_k$
3. Вычислить  $B_{k+1}$  из  $B_k$

## Шаблон квазиньютоновского метода

Пусть  $x_0 \in \mathbb{R}^n$ ,  $B_0 \succ 0$ . Для  $k = 0, 1, 2, \dots$ , повторяем:

1. Решить  $B_k d_k = -\nabla f(x_k)$
2. Обновить  $x_{k+1} = x_k + \alpha_k d_k$
3. Вычислить  $B_{k+1}$  из  $B_k$

$$C_k = (B_k^{-1})$$

Разные квазиньютоновские методы реализуют шаг 3 по-разному. Мы скоро увидим, что обычно мы можем вычислить  $(B_{k+1})^{-1}$  из  $(B_k)^{-1}$ .

## Шаблон квазиньютоновского метода

Пусть  $x_0 \in \mathbb{R}^n$ ,  $B_0 \succ 0$ . Для  $k = 0, 1, 2, \dots$ , повторяем:

1. Решить  $B_k d_k = -\nabla f(x_k)$
2. Обновить  $x_{k+1} = x_k + \alpha_k d_k$
3. Вычислить  $B_{k+1}$  из  $B_k$

Разные квазиньютоновские методы реализуют шаг 3 по-разному. Мы скоро увидим, что обычно мы можем вычислить  $(B_{k+1})^{-1}$  из  $(B_k)^{-1}$ .

**Основная идея:** Поскольку  $B_k$  уже содержит информацию о гессиане, используем подходящее обновление матрицы для формирования  $B_{k+1}$ .

$$B_{k+1} = \dots - \frac{a a^\top}{\dots}$$

## Шаблон квазиньютоновского метода

Пусть  $x_0 \in \mathbb{R}^n$ ,  $B_0 \succ 0$ . Для  $k = 0, 1, 2, \dots$ , повторяем:

1. Решить  $B_k d_k = -\nabla f(x_k)$
2. Обновить  $x_{k+1} = x_k + \alpha_k d_k$
3. Вычислить  $B_{k+1}$  из  $B_k$

Разные квазиньютоновские методы реализуют шаг 3 по-разному. Мы скоро увидим, что обычно мы можем вычислить  $(B_{k+1})^{-1}$  из  $(B_k)^{-1}$ .

**Основная идея:** Поскольку  $B_k$  уже содержит информацию о гессиане, используем подходящее обновление матрицы для формирования  $B_{k+1}$ .

**Разумное требование для  $B_{k+1}$  (вдохновленное методом секущих):**

$$\nabla f(x_{k+1}) - \nabla f(x_k) = B_{k+1}(x_{k+1} - x_k) = B_{k+1}d_k$$

$$\stackrel{\textcolor{red}{\bullet}}{=} \Delta y_k = B_{k+1}\Delta x_k$$

## Шаблон квазиньютоновского метода

Пусть  $x_0 \in \mathbb{R}^n$ ,  $B_0 \succ 0$ . Для  $k = 0, 1, 2, \dots$ , повторяем:

1. Решить  $B_k d_k = -\nabla f(x_k)$
2. Обновить  $x_{k+1} = x_k + \alpha_k d_k$
3. Вычислить  $B_{k+1}$  из  $B_k$

Разные квазиньютоновские методы реализуют шаг 3 по-разному. Мы скоро увидим, что обычно мы можем вычислить  $(B_{k+1})^{-1}$  из  $(B_k)^{-1}$ .

**Основная идея:** Поскольку  $B_k$  уже содержит информацию о гессиане, используем подходящее обновление матрицы для формирования  $B_{k+1}$ .

**Разумное требование для  $B_{k+1}$**  (вдохновленное методом секущих):

$$\begin{aligned}\nabla f(x_{k+1}) - \nabla f(x_k) &= B_{k+1}(x_{k+1} - x_k) = B_{k+1}d_k \\ \Delta y_k &= B_{k+1}\Delta x_k\end{aligned}$$

Помимо уравнения секущей, мы хотим:

- $B_{k+1}$  симметричная

если  $B_k \succ 0 \Rightarrow B_{k+1} \succ 0$

## Шаблон квазиньютоновского метода

Пусть  $x_0 \in \mathbb{R}^n$ ,  $B_0 \succ 0$ . Для  $k = 0, 1, 2, \dots$ , повторяем:

1. Решить  $B_k d_k = -\nabla f(x_k)$
2. Обновить  $x_{k+1} = x_k + \alpha_k d_k$
3. Вычислить  $B_{k+1}$  из  $B_k$

Разные квазиньютоновские методы реализуют шаг 3 по-разному. Мы скоро увидим, что обычно мы можем вычислить  $(B_{k+1})^{-1}$  из  $(B_k)^{-1}$ .

**Основная идея:** Поскольку  $B_k$  уже содержит информацию о гессиане, используем подходящее обновление матрицы для формирования  $B_{k+1}$ .

**Разумное требование для  $B_{k+1}$**  (вдохновленное методом секущих):

$$\nabla f(x_{k+1}) - \nabla f(x_k) = B_{k+1}(x_{k+1} - x_k) = B_{k+1}d_k$$

$$\Delta y_k = B_{k+1}\Delta x_k$$

Хочу, чтобы  $B_{k+1}$  было легко считать

Помимо уравнения секущей, мы хотим:

- $B_{k+1}$  симметричная
- $B_{k+1}$  близка к  $B_k$

## Шаблон квазиньютоновского метода

Пусть  $x_0 \in \mathbb{R}^n$ ,  $B_0 \succ 0$ . Для  $k = 0, 1, 2, \dots$ , повторяем:

1. Решить  $B_k d_k = -\nabla f(x_k)$
2. Обновить  $x_{k+1} = x_k + \alpha_k d_k$
3. Вычислить  $B_{k+1}$  из  $B_k$

Разные квазиньютоновские методы реализуют шаг 3 по-разному. Мы скоро увидим, что обычно мы можем вычислить  $(B_{k+1})^{-1}$  из  $(B_k)^{-1}$ .

**Основная идея:** Поскольку  $B_k$  уже содержит информацию о гессиане, используем подходящее обновление матрицы для формирования  $B_{k+1}$ .

**Разумное требование для  $B_{k+1}$  (вдохновленное методом секущих):**

$$\begin{aligned}\nabla f(x_{k+1}) - \nabla f(x_k) &= B_{k+1}(x_{k+1} - x_k) = B_{k+1}d_k \\ \Delta y_k &= B_{k+1}\Delta x_k\end{aligned}$$

Помимо уравнения секущей, мы хотим:

- $B_{k+1}$  симметричная
- $B_{k+1}$  близка к  $B_k$
- $B_k \succ 0 \Rightarrow B_{k+1} \succ 0$

## Симметричное одноранговое обновление

Попробуем обновление вида:

$$B_{k+1} = B_k + \underbrace{auu^T}_{\text{известно}}$$

$a \in \mathbb{R}$   
 $u \in \mathbb{R}^n$



## Симметричное одноранговое обновление

Попробуем обновление вида:

$$u = \Delta y_k - B_k d_k$$

Уравнение секущей  $B_{k+1}d_k = \Delta y_k$  дает:

$$B_{k+1} = B_k + auu^T$$

$$(au^T d_k) u = \Delta y_k - B_k d_k$$

$\Rightarrow$

$$(B_k + auu^T) d_k = \Delta y_k$$

$$B_k d_k + a u (u^T d_k) \overset{\text{СКАЛЯР}}{=} \Delta y_k$$

$$a = \frac{1}{u^T d_k} =$$

$$= \frac{1}{(\Delta y_k - B_k d_k)^T d_k}$$

$$a \cdot (u^T d_k) \cdot \underline{u} = \underline{\Delta y_k - B_k d_k}$$

## Симметричное одноранговое обновление

Попробуем обновление вида:

$$B_{k+1} = B_k + auu^T$$

Уравнение секущей  $B_{k+1}d_k = \Delta y_k$  дает:

$$(au^T d_k)u = \Delta y_k - B_k d_k$$

Это верно только если  $u$  является кратным  $\Delta y_k - B_k d_k$ . Положив  $u = \Delta y_k - B_k d_k$ , мы решаем уравнение,

$$a = \frac{1}{(\Delta y_k - B_k d_k)^T d_k},$$

## Симметричное одноранговое обновление

Symmetric

RANK-1

Попробуем обновление вида:

$$B_{k+1} = B_k + auu^T$$

Уравнение секущей  $B_{k+1}d_k = \Delta y_k$  дает:

$$(au^T d_k)u = \Delta y_k - B_k d_k$$

Это верно только если  $u$  является кратным  $\Delta y_k - B_k d_k$ . Положив  $u = \Delta y_k - B_k d_k$ , мы решаем уравнение,

$$a = \frac{1}{(\Delta y_k - B_k d_k)^T d_k},$$

SR1

что приводит к

$$B_{k+1} = B_k + \frac{(\Delta y_k - B_k d_k)(\Delta y_k - B_k d_k)^T}{(\Delta y_k - B_k d_k)^T d_k}$$

Это называется симметричным одноранговым (SR1) обновлением.

## Симметричное одноранговое обновление с инверсией

Как мы можем решить

$$B_{k+1}d_{k+1} = -\nabla f(x_{k+1}),$$

чтобы сделать следующий шаг? Помимо обновления  $B_k \rightarrow B_{k+1}$ , будем также обновлять обратную матрицу, т.е.  $C_k = B_k^{-1} \rightarrow C_{k+1} = (B_{k+1})^{-1}$ .

Формула Шермана-Моррисона:

Формула Шермана-Моррисона утверждает:

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}$$

Таким образом, для SR1 обновления, обратная матрица также легко обновляется:

$$C_{k+1} = C_k + \frac{(d_k - C_k \Delta y_k)(d_k - C_k \Delta y_k)^T}{(d_k - C_k \Delta y_k)^T \Delta y_k}$$

В общем, SR1 прост и дешев, но у него есть ключевой недостаток: он не сохраняет положительную определенность.

## Обновление Давидона-Флетчера-Паузлла

Мы могли бы продолжить ту же идею для обновления обратной матрицы  $C$ :

$$C_{k+1} = C_k + auu^T + bvv^T.$$

$$a \in \mathbb{R}$$

$$b \in \mathbb{R}$$

$$u, v \in \mathbb{R}^n, \mathbb{R}^s$$

$$d_k = C_k u \Delta y_k$$

$$d_k = (C_k + a uu^T + b vv^T) \Delta y_k$$

$$d_k = C_k \Delta y_k + a u \underline{u^T \Delta y_k} + b v \underline{v^T \Delta y_k}$$

$$\underbrace{a \cdot u^T \Delta y_k \cdot u}_{} + \underbrace{b v^T \Delta y_k \cdot v}_{} = d_k - C_k \Delta y_k$$

$$u = d_k$$

$$v = C_k \Delta y_k$$

## Обновление Давидона-Флетчера-Пауэлла

Мы могли бы продолжить ту же идею для обновления обратной матрицы  $C$ :

$$C_{k+1} = C_k + auu^T + bvv^T.$$

Умножая на  $\Delta y_k$ , используя уравнение секущей  $d_k = C_k \Delta y_k$  и решая для  $a, b$ , получаем:

$$C_{k+1} = C_k - \frac{C_k \Delta y_k \Delta y_k^T C_k}{\Delta y_k^T C_k \Delta y_k} + \frac{d_k d_k^T}{\Delta y_k^T d_k}$$

### Применение формулы Вудбери

Вудбери показывает:

$$B_{k+1} = \left( I - \frac{\Delta y_k d_k^T}{\Delta y_k^T d_k} \right) B_k \left( I - \frac{d_k \Delta y_k^T}{\Delta y_k^T d_k} \right) + \frac{\Delta y_k \Delta y_k^T}{\Delta y_k^T d_k}$$

Это обновление Давидона-Флетчера-Пауэлла (DFP). Также дешево:  $O(n^2)$ , но сохраняет положительную определенность. Не так популярно, как BFGS.

## Обновление Брайдена-Флетчера-Гольдштейна-Шенно

Попробуем теперь двухранговое обновление:

$$B_{k+1} = B_k + auu^T + bvv^T.$$

## Обновление Бройдена-Флетчера-Гольдштейна-Шенно

Попробуем теперь двухранговое обновление:

$$B_{k+1} = B_k + auu^T + bvv^T.$$

Уравнение секущей  $\Delta y_k = B_{k+1}d_k$  дает:

$$\underbrace{\Delta y_k - \underline{B_k d_k}}_{=} = (au^T d_k)u + (bv^T d_k)v$$

$a = \dots$   
 $b = \dots$

## Обновление Бройдена-Флетчера-Гольдштейна-Шенно

$$B_k > 0$$

$$C_k > 0$$

Попробуем теперь двухранговое обновление:

$$B_{k+1} = B_k + auu^T + bvv^T.$$

Уравнение секущей  $\Delta y_k = B_{k+1}d_k$  дает:

$$\Delta y_k - B_k d_k = (au^T d_k)u + (bv^T d_k)v$$

Положив  $u = \Delta y_k$ ,  $v = B_k d_k$  и решая для  $a$ ,  $b$ , получаем:

$$B_{k+1} = B_k - \frac{B_k d_k d_k^T B_k}{d_k^T B_k d_k} + \frac{\Delta y_k \Delta y_k^T}{d_k^T \Delta y_k}$$

Это обновление Бройдена-Флетчера-Гольдштейна-Шенно (BFGS).

## Обновление Бройдена-Флетчера-Гольдштейна-Шенно с инверсией

---

### Формула Вудбери

Формула Вудбери, обобщение формулы Шермана-Моррисона, дается как:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

# Обновление Бройдена-Флетчера-Гольдштейна-Шенно с инверсией

## Формула Вудбери

Формула Вудбери, обобщение формулы Шермана-Моррисона, дается как:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

Применяя её к нашему случаю, получаем двухранговое обновление обратной матрицы  $C$ :

$$C_{k+1} = C_k + \frac{(d_k - C_k \Delta y_k) d_k^T}{\Delta y_k^T d_k} + \frac{d_k (d_k - C_k \Delta y_k)^T}{\Delta y_k^T d_k} - \frac{(d_k - C_k \Delta y_k)^T \Delta y_k}{(\Delta y_k^T d_k)^2} d_k d_k^T$$

$$C_{k+1} = \left( I - \frac{d_k \Delta y_k^T}{\Delta y_k^T d_k} \right) C_k \left( I - \frac{\Delta y_k d_k^T}{\Delta y_k^T d_k} \right) + \frac{d_k d_k^T}{\Delta y_k^T d_k}$$

Эта формулировка обеспечивает, что обновление BFGS, оставаясь достаточно общим, сохраняет вычислительную эффективность и требует  $O(n^2)$  операций. Важно, что обновление BFGS сохраняет положительную определенность:  $B_k \succ 0 \Rightarrow B_{k+1} \succ 0$ . Эквивалентно,  $C_k \succ 0 \Rightarrow C_{k+1} \succ 0$ .

## Алгоритм BFGS

Перейдём к стандартным обозначениям:  $s_k = \Delta x_k = x_{k+1} - x_k$  — шаг,  $y_k = \Delta y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$  — разность градиентов,  $\rho_k = \frac{1}{y_k^\top s_k}$ .

## Алгоритм BFGS

Перейдём к стандартным обозначениям:  $s_k = \Delta x_k = x_{k+1} - x_k$  — шаг,  $y_k = \Delta y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$  — разность градиентов,  $\rho_k = \frac{1}{y_k^\top s_k}$ .

Вход:  $x_0 \in \mathbb{R}^n$ ,  $C_0 = I_n$

Для  $k = 0, 1, 2, \dots$ :

$$d_k = -C_k \nabla f(x_k) \quad \text{МАТВЕК} \quad O(n^2)$$

$$\alpha_k = \text{LineSearch}(f, x_k, d_k) \quad (\text{условия Вольфа})$$

$$x_{k+1} = x_k + \alpha_k d_k$$

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

$$\rho_k = \frac{1}{y_k^\top s_k}$$

$$C_{k+1} = (I - \rho_k s_k y_k^\top) C_k (I - \rho_k y_k s_k^\top) + \rho_k s_k s_k^\top$$

| BFGS

## Условия Вольфа

Для корректности обновления BFGS необходимо, чтобы  $y_k^\top s_k > 0$  (иначе положительная определенность  $C_{k+1}$  нарушается). Это гарантируется условиями Вольфе для линейного поиска.

## Условия Вольфа

Для корректности обновления BFGS необходимо, чтобы  $y_k^\top s_k > 0$  (иначе положительная определенность  $C_{k+1}$  нарушается). Это гарантируется условиями Вольфе для линейного поиска.

### 🔥 Условия Вольфе

Шаг  $\alpha_k$  выбирается так, чтобы выполнялись два условия:

1. **Достаточное убывание** (условие Армихо):

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^\top d_k, \quad c_1 \in (0, 1)$$

Типичные значения:  $c_1 = 10^{-4}$ ,  $c_2 = 0.9$ .

## Условия Вольфе

Для корректности обновления BFGS необходимо, чтобы  $y_k^\top s_k > 0$  (иначе положительная определенность  $C_{k+1}$  нарушается). Это гарантируется условиями Вольфе для линейного поиска.

### 🔥 Условия Вольфе

Шаг  $\alpha_k$  выбирается так, чтобы выполнялись два условия:

1. **Достаточное убывание** (условие Армихо):

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^\top d_k, \quad c_1 \in (0, 1)$$

2. **Условие кривизны:**

$$\nabla f(x_k + \alpha_k d_k)^\top d_k \geq c_2 \nabla f(x_k)^\top d_k, \quad c_2 \in (c_1, 1)$$

Типичные значения:  $c_1 = 10^{-4}$ ,  $c_2 = 0.9$ .

## Условия Вольфа

Для корректности обновления BFGS необходимо, чтобы  $y_k^\top s_k > 0$  (иначе положительная определенность  $C_{k+1}$  нарушается). Это гарантируется условиями Вольфе для линейного поиска.

### 🔥 Условия Вольфе

Шаг  $\alpha_k$  выбирается так, чтобы выполнялись два условия:

1. **Достаточное убывание** (условие Армихо):

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^\top d_k, \quad c_1 \in (0, 1)$$

2. **Условие кривизны:**

$$\nabla f(x_k + \alpha_k d_k)^\top d_k \geq c_2 \nabla f(x_k)^\top d_k, \quad c_2 \in (c_1, 1)$$

Типичные значения:  $c_1 = 10^{-4}$ ,  $c_2 = 0.9$ .

## Условия Вольфера

Для корректности обновления BFGS необходимо, чтобы  $y_k^\top s_k > 0$  (иначе положительная определенность  $C_{k+1}$  нарушается). Это гарантируется условиями Вольфера для линейного поиска.

### 🔥 Условия Вольфера

Шаг  $\alpha_k$  выбирается так, чтобы выполнялись два условия:

1. **Достаточное убывание** (условие Армихо):

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^\top d_k, \quad c_1 \in (0, 1)$$

2. **Условие кривизны:**

$$\nabla f(x_k + \alpha_k d_k)^\top d_k \geq c_2 \nabla f(x_k)^\top d_k, \quad c_2 \in (c_1, 1)$$

Типичные значения:  $c_1 = 10^{-4}$ ,  $c_2 = 0.9$ .

Из условия кривизны непосредственно следует:

$$y_k^\top s_k = (\nabla f(x_{k+1}) - \nabla f(x_k))^\top s_k \geq (c_2 - 1) \nabla f(x_k)^\top d_k \cdot \alpha_k > 0.$$

## Сходимость BFGS

Теорема. Глобальная сходимость BFGS.

Пусть  $f \in C^1(\mathbb{R}^n)$  — выпуклая функция с липшицевым градиентом, и множество уровня  $\mathcal{L}_0 = \{x : f(x) \leq f(x_0)\}$  ограничено. Тогда метод BFGS с линейным поиском, удовлетворяющим условиям Вольфа, сходится глобально:

$$\lim_{k \rightarrow \infty} \nabla f(x_k) = 0.$$

## Сходимость BFGS

Теорема. Глобальная сходимость BFGS.

Пусть  $f \in C^1(\mathbb{R}^n)$  — выпуклая функция с липшицевым градиентом, и множество уровня  $\mathcal{L}_0 = \{x : f(x) \leq f(x_0)\}$  ограничено. Тогда метод BFGS с линейным поиском, удовлетворяющим условиям Вольфе, сходится глобально:

$$\lim_{k \rightarrow \infty} \nabla f(x_k) = 0.$$

Более того, вблизи решения  $x^*$ , где  $\nabla^2 f(x^*) \succ 0$  и гессиан липшицев, метод BFGS демонстрирует **локальную суперлинейную сходимость** (Dennis, Moré, 1974):  
*сверхлинейную*

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \rightarrow 0 \quad \text{при } k \rightarrow \infty.$$

## Сходимость BFGS

■ Теорема. Глобальная сходимость BFGS.

Пусть  $f \in C^1(\mathbb{R}^n)$  — выпуклая функция с липшицевым градиентом, и множество уровня  $\mathcal{L}_0 = \{x : f(x) \leq f(x_0)\}$  ограничено. Тогда метод BFGS с линейным поиском, удовлетворяющим условиям Вольфе, сходится глобально:

$$\lim_{k \rightarrow \infty} \nabla f(x_k) = 0.$$

Более того, вблизи решения  $x^*$ , где  $\nabla^2 f(x^*) \succ 0$  и гессиан липшицев, метод BFGS демонстрирует **локальную суперлинейную сходимость** (Dennis, Moré, 1974):

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \rightarrow 0 \quad \text{при } k \rightarrow \infty.$$

■ Суперлинейная сходимость BFGS — промежуточный результат между линейной сходимостью градиентного спуска и квадратичной сходимостью метода Ньютона. При этом стоимость одной итерации BFGS составляет  $O(n^2)$  вместо  $O(n^3)$  у метода Ньютона. Заметим, что последовательность  $B_k$  не обязана сходиться к  $\nabla^2 f(x^*)$  — достаточно выполнения условия Дениса-Море.

# Сравнение квазиньютоновских обновлений

Ньютон Full

$O(n^3)$

Метод	Обновление	Сохраняет $\succ 0$	Стоимость
SR1	Ранг 1	Нет	$O(n^2)$
DFP	Ранг 2 (на $C$ )	Да	$O(n^2)$
BFGS	Ранг 2 (на $B$ )	Да	$O(n^2)$

# Сравнение квазиньютоновских обновлений

Метод	Обновление	Сохраняет $\succ 0$	Стоимость
SR1	Ранг 1	Нет	$O(n^2)$
DFP	Ранг 2 (на $C$ )	Да	$O(n^2)$
BFGS	Ранг 2 (на $B$ )	Да	$O(n^2)$

Источник:

На практике BFGS значительно превосходит DFP по устойчивости. Причина: DFP обновляет непосредственно  $C_k$ , и при неточном линейном поиске ошибки в  $C_k$  могут быстро накапливаться. BFGS обновляет  $B_k$ , а  $C_k$  получается через формулу Вудбери, что даёт лучшую численную устойчивость.

*limited memory*

L-BFGS: метод BFGS с ограниченной памятью

## Мотивация L-BFGS

Метод BFGS требует хранения матрицы  $C_k \in \mathbb{R}^{n \times n}$ , что означает  $O(n^2)$  памяти. Для задач большой размерности (например,  $n = 10^6$  в машинном обучении) это неприемлемо.

## Мотивация L-BFGS

Метод BFGS требует хранения матрицы  $C_k \in \mathbb{R}^{n \times n}$ , что означает  $O(n^2)$  памяти. Для задач большой размерности (например,  $n = 10^6$  в машинном обучении) это неприемлемо.

🔥 Ключевая идея L-BFGS

~~матрица~~  $s_i, y_i$  вектор

Вместо хранения полной матрицы  $C_k$  храним только  $m$  последних пар  $(s_i, y_i)$  и вычисляем произведение  $C_k \nabla f(x_k)$  неявно. Это снижает требования к памяти до  $O(mn)$  при типичном  $m \in [3, 20]$ .

$$\begin{aligned} C_k \cdot \nabla f(x_k) &= \\ (C_{k-2} + s_3 y_3 y_3^\top + s_1 y_1 y_1^\top + s_2 y_2 y_2^\top) \nabla f &= \\ &= \end{aligned}$$

## Мотивация L-BFGS

Метод BFGS требует хранения матрицы  $C_k \in \mathbb{R}^{n \times n}$ , что означает  $O(n^2)$  памяти. Для задач большой размерности (например,  $n = 10^6$  в машинном обучении) это неприемлемо.

### 🔥 Ключевая идея L-BFGS

Вместо хранения полной матрицы  $C_k$  храним только  $m$  последних пар  $(s_i, y_i)$  и вычисляем произведение  $C_k \nabla f(x_k)$  неявно. Это снижает требования к памяти до  $O(mn)$  при типичном  $m \in [3, 20]$ .

Рекурсивная структура обновления BFGS позволяет записать  $C_k$  через последовательные применения формулы обновления:

$$C_k = V_{k-1}^\top C_{k-1} V_{k-1} + \rho_{k-1} s_{k-1} s_{k-1}^\top,$$

где  $V_i = I - \rho_i y_i s_i^\top$ . Раскрывая рекурсию на  $m$  шагов назад:

$$\begin{aligned} C_k &= (V_{k-1}^\top \cdots V_{k-m}^\top) C_{k-m}^0 (V_{k-m} \cdots V_{k-1}) \\ &\quad + \rho_{k-m} (V_{k-1}^\top \cdots V_{k-m+1}^\top) s_{k-m} s_{k-m}^\top (V_{k-m+1} \cdots V_{k-1}) \\ &\quad + \dots + \rho_{k-1} s_{k-1} s_{k-1}^\top \end{aligned}$$

## Мотивация L-BFGS

Метод BFGS требует хранения матрицы  $C_k \in \mathbb{R}^{n \times n}$ , что означает  $O(n^2)$  памяти. Для задач большой размерности (например,  $n = 10^6$  в машинном обучении) это неприемлемо.

### 🔥 Ключевая идея L-BFGS

Вместо хранения полной матрицы  $C_k$  храним только  $m$  последних пар  $(s_i, y_i)$  и вычисляем произведение  $C_k \nabla f(x_k)$  неявно. Это снижает требования к памяти до  $O(mn)$  при типичном  $m \in [3, 20]$ .

Рекурсивная структура обновления BFGS позволяет записать  $C_k$  через последовательные применения формулы обновления:

$$C_k = V_{k-1}^\top C_{k-1} V_{k-1} + \rho_{k-1} s_{k-1} s_{k-1}^\top,$$

где  $V_i = I - \rho_i y_i s_i^\top$ . Раскрывая рекурсию на  $m$  шагов назад:

$$\begin{aligned} C_k &= (V_{k-1}^\top \cdots V_{k-m}^\top) C_{k-m}^0 (V_{k-m} \cdots V_{k-1}) \\ &\quad + \rho_{k-m} (V_{k-1}^\top \cdots V_{k-m+1}^\top) s_{k-m} s_{k-m}^\top (V_{k-m+1} \cdots V_{k-1}) \\ &\quad + \dots + \rho_{k-1} s_{k-1} s_{k-1}^\top \end{aligned}$$

Начальное приближение  $C_{k-m}^0$  обычно выбирается как  $C_k^0 = \gamma_k I$ , где  $\gamma_k = \frac{s_{k-1}^\top y_{k-1}}{y_{k-1}^\top y_{k-1}}$ .

## Алгоритм двух циклов (two-loop recursion)

Этот алгоритм вычисляет  $r = C_k \nabla f(x_k)$  без явного построения матрицы  $C_k$ , используя только хранимые пары  $(s_i, y_i)_{i=k-m}^{k-1}$ .

## Алгоритм двух циклов (two-loop recursion)

Этот алгоритм вычисляет  $r = C_k \nabla f(x_k)$  без явного построения матрицы  $C_k$ , используя только хранимые пары  $(s_i, y_i)_{i=k-m}^{k-1}$ .

$$q = \nabla f(x_k)$$

**Цикл 1 (от новых к старым):**

Для  $i = k-1, k-2, \dots, k-m$  :

$$\alpha_i = \rho_i s_i^\top q$$

$$q = q - \alpha_i y_i$$

$$r = C_k^0 q \quad (\text{где } C_k^0 = \gamma_k I)$$

**Цикл 2 (от старых к новым):**

Для  $i = k-m, k-m+1, \dots, k-1$  :

$$\beta = \rho_i y_i^\top r$$

$$r = r + (\alpha_i - \beta) s_i$$

$$d_k = -r$$

# Свойства L-BFGS

**Вычислительная сложность:**

- Память:  $O(mn)$  вместо  $O(n^2)$

## Свойства L-BFGS

**Вычислительная сложность:**

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$

## Свойства L-BFGS

Вычислительная сложность:

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20

## Свойства L-BFGS

Вычислительная сложность:

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

## Свойства L-BFGS

Вычислительная сложность:

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

## Свойства L-BFGS

**Вычислительная сложность:**

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

**Практические рекомендации:**

- $m = 10$  — хорошее значение по умолчанию

# Свойства L-BFGS

**Вычислительная сложность:**

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

**Практические рекомендации:**

- $m = 10$  — хорошее значение по умолчанию
- Увеличение  $m$  улучшает аппроксимацию гессиана, но замедляет итерацию

# Свойства L-BFGS

Вычислительная сложность:

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

Практические рекомендации:

- $m = 10$  — хорошее значение по умолчанию
- Увеличение  $m$  улучшает аппроксимацию гессиана, но замедляет итерацию
- Инициализация  $C_k^0 = \gamma_k I$  адаптирует масштаб на каждой итерации

# Свойства L-BFGS

Вычислительная сложность:

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

Практические рекомендации:

- $m = 10$  — хорошее значение по умолчанию
- Увеличение  $m$  улучшает аппроксимацию гессиана, но замедляет итерацию
- Инициализация  $C_k^0 = \gamma_k I$  адаптирует масштаб на каждой итерации

# Свойства L-BFGS

## Вычислительная сложность:

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

## Практические рекомендации:

- $m = 10$  — хорошее значение по умолчанию
- Увеличение  $m$  улучшает аппроксимацию гессиана, но замедляет итерацию
- Инициализация  $C_k^0 = \gamma_k I$  адаптирует масштаб на каждой итерации

## Сходимость:

- Глобальная сходимость с условиями Вольфе (те же условия, что для BFGS)

# Свойства L-BFGS

## Вычислительная сложность:

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

## Практические рекомендации:

- $m = 10$  — хорошее значение по умолчанию
- Увеличение  $m$  улучшает аппроксимацию гессиана, но замедляет итерацию
- Инициализация  $C_k^0 = \gamma_k I$  адаптирует масштаб на каждой итерации

## Сходимость:

- Глобальная сходимость с условиями Вольфе (те же условия, что для BFGS)
- При  $m \ll n$ : линейная скорость сходимости (в отличие от суперлинейной у полного BFGS)

# Свойства L-BFGS

## Вычислительная сложность:

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

## Практические рекомендации:

- $m = 10$  — хорошее значение по умолчанию
- Увеличение  $m$  улучшает аппроксимацию гессиана, но замедляет итерацию
- Инициализация  $C_k^0 = \gamma_k I$  адаптирует масштаб на каждой итерации

## Сходимость:

- Глобальная сходимость с условиями Вольфе (те же условия, что для BFGS)
- При  $m \ll n$ : линейная скорость сходимости (в отличие от суперлинейной у полного BFGS)
- При  $m \rightarrow n$ : восстанавливается суперлинейная сходимость

# Свойства L-BFGS

## Вычислительная сложность:

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

## Практические рекомендации:

- $m = 10$  — хорошее значение по умолчанию
- Увеличение  $m$  улучшает аппроксимацию гессиана, но замедляет итерацию
- Инициализация  $C_k^0 = \gamma_k I$  адаптирует масштаб на каждой итерации

## Сходимость:

- Глобальная сходимость с условиями Вольфе (те же условия, что для BFGS)
- При  $m \ll n$ : линейная скорость сходимости (в отличие от суперлинейной у полного BFGS)
- При  $m \rightarrow n$ : восстанавливается суперлинейная сходимость
- На практике сходится быстрее градиентного спуска и сопоставимо с полным BFGS

# Свойства L-BFGS

## Вычислительная сложность:

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

## Практические рекомендации:

- $m = 10$  — хорошее значение по умолчанию
- Увеличение  $m$  улучшает аппроксимацию гессиана, но замедляет итерацию
- Инициализация  $C_k^0 = \gamma_k I$  адаптирует масштаб на каждой итерации

## Сходимость:

- Глобальная сходимость с условиями Вольфе (те же условия, что для BFGS)
- При  $m \ll n$ : линейная скорость сходимости (в отличие от суперлинейной у полного BFGS)
- При  $m \rightarrow n$ : восстанавливается суперлинейная сходимость
- На практике сходится быстрее градиентного спуска и сопоставимо с полным BFGS

# Свойства L-BFGS

## Вычислительная сложность:

- Память:  $O(mn)$  вместо  $O(n^2)$
- Умножение  $C_k \nabla f(x_k)$ :  $O(mn)$  вместо  $O(n^2)$
- Типичные значения  $m$ : от 3 до 20
- Стоимость итерации сопоставима с градиентным спуском

## Практические рекомендации:

- $m = 10$  — хорошее значение по умолчанию
- Увеличение  $m$  улучшает аппроксимацию гессиана, но замедляет итерацию
- Инициализация  $C_k^0 = \gamma_k I$  адаптирует масштаб на каждой итерации

## Сходимость:

- Глобальная сходимость с условиями Вольфе (те же условия, что для BFGS)
- При  $m \ll n$ : линейная скорость сходимости (в отличие от суперлинейной у полного BFGS)
- При  $m \rightarrow n$ : восстанавливается суперлинейная сходимость
- На практике сходится быстрее градиентного спуска и сопоставимо с полным BFGS

## Сравнение методов:

	GD	BFGS	L-BFGS	Newton
Память	$O(n)$	$O(n^2)$	$O(mn)$	$O(n^2)$
Итерация	$O(n)$	$O(n^2)$	$O(mn)$	$O(n^3)$
Сходимость	Лин.	Суперлин.	Лин.	Квадр.

Сходимость  $\forall x, y \in \mathbb{R}^n \quad \|\nabla^2 f(x) - \nabla^2 f(y)\| \leq M \cdot \|x - y\|$

i Theorem

Пусть  $f(x)$  — сильно выпуклая дважды непрерывно дифференцируемая функция на  $\mathbb{R}^n$ , для второй производной которой выполняются неравенства:  $\mu I_n \preceq \nabla^2 f(x) \preceq L I_n$ . Пусть также гессиан функции  $M$ -липшицев. Тогда метод Ньютона сходится локально к решению с квадратичной скоростью, т.е. при  $\|x_0 - x^*\| < \frac{2\mu}{3M}$ :

$$\|x_{k+1} - x^*\| \leq \frac{3M}{2\mu} \|x_k - x^*\|^2$$

$$r_{k+1} \leq \dots r_k^2$$

$$\mu I_n \preceq \nabla^2 f(x) \preceq L I$$

гладкость

сильно вып.

# Сходимость

## Theorem

Пусть  $f(x)$  — сильно выпуклая дважды непрерывно дифференцируемая функция на  $\mathbb{R}^n$ , для второй производной которой выполняются неравенства:  $\mu I_n \preceq \nabla^2 f(x) \preceq L I_n$ . Пусть также гессиан функции  $M$ -липшицев. Тогда метод Ньютона сходится локально к решению с квадратичной скоростью, т.е. при  $\|x_0 - x^*\| < \frac{2\mu}{3M}$ :

$$\|x_{k+1} - x^*\| \leq \frac{3M}{2\mu} \|x_k - x^*\|^2$$

## Доказательство

# Сходимость

## Theorem

Пусть  $f(x)$  — сильно выпуклая дважды непрерывно дифференцируемая функция на  $\mathbb{R}^n$ , для второй производной которой выполняются неравенства:  $\mu I_n \preceq \nabla^2 f(x) \preceq L I_n$ . Пусть также гессиан функции  $M$ -липшицев. Тогда метод Ньютона сходится локально к решению с квадратичной скоростью, т.е. при  $\|x_0 - x^*\| < \frac{2\mu}{3M}$ :

$$\|x_{k+1} - x^*\| \leq \frac{3M}{2\mu} \|x_k - x^*\|^2$$

## Доказательство

- Мы будем использовать формулу Ньютона-Лейбница

$$\nabla f(x_k) - \nabla f(x^*) = \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*))(x_k - x^*) d\tau$$

# Сходимость

## Theorem

Пусть  $f(x)$  — сильно выпуклая дважды непрерывно дифференцируемая функция на  $\mathbb{R}^n$ , для второй производной которой выполняются неравенства:  $\mu I_n \preceq \nabla^2 f(x) \preceq L I_n$ . Пусть также гессиан функции  $M$ -липшицев. Тогда метод Ньютона сходится локально к решению с квадратичной скоростью, т.е. при  $\|x_0 - x^*\| < \frac{2\mu}{3M}$ :

$$\|x_{k+1} - x^*\| \leq \frac{3M}{2\mu} \|x_k - x^*\|^2$$

## Доказательство

- Мы будем использовать формулу Ньютона-Лейбница

$$\nabla f(x_k) - \nabla f(x^*) = \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*))(x_k - x^*) d\tau$$

- Мы будем отслеживать расстояние до решения

# Сходимость

## Theorem

Пусть  $f(x)$  — сильно выпуклая дважды непрерывно дифференцируемая функция на  $\mathbb{R}^n$ , для второй производной которой выполняются неравенства:  $\mu I_n \preceq \nabla^2 f(x) \preceq L I_n$ . Пусть также гессиан функции  $M$ -липшицев. Тогда метод Ньютона сходится локально к решению с квадратичной скоростью, т.е. при  $\|x_0 - x^*\| < \frac{2\mu}{3M}$ :

$$\|x_{k+1} - x^*\| \leq \frac{3M}{2\mu} \|x_k - x^*\|^2$$

## Доказательство

- Мы будем использовать формулу Ньютона-Лейбница

$$\nabla f(x_k) - \nabla f(x^*) = \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*))(x_k - x^*) d\tau$$

- Мы будем отслеживать расстояние до решения

$$x_{k+1} - x^* = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) - x^* = x_k - x^* - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) =$$

$$r_{k+1} = r_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

# Сходимость

## Theorem

Пусть  $f(x)$  — сильно выпуклая дважды непрерывно дифференцируемая функция на  $\mathbb{R}^n$ , для второй производной которой выполняются неравенства:  $\mu I_n \preceq \nabla^2 f(x) \preceq L I_n$ . Пусть также гессиан функции  $M$ -липшицев. Тогда метод Ньютона сходится локально к решению с квадратичной скоростью, т.е. при  $\|x_0 - x^*\| < \frac{2\mu}{3M}$ :

$$\|x_{k+1} - x^*\| \leq \frac{3M}{2\mu} \|x_k - x^*\|^2$$

## Доказательство

- Мы будем использовать формулу Ньютона-Лейбница

$$\nabla f(x_k) - \nabla f(x^*) = \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*))(x_k - x^*) d\tau$$

~~$\nabla f(x_k) - \nabla f(x^*) = \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*))(x_k - x^*) d\tau$~~

- Мы будем отслеживать расстояние до решения

$$\begin{aligned} x_{k+1} - x^* &= x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) - x^* = x_k - x^* - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) = \\ &= x_k - x^* - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*))(x_k - x^*) d\tau \end{aligned}$$

## Сходимость

3.

$$= \left( I - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) =$$

## Сходимость

3.

$$\begin{aligned} &= \left( I - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \nabla^2 f(x_k) - \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \end{aligned}$$

## Сходимость

3.

$$\begin{aligned} &= \left( I - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \nabla^2 f(x_k) - \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau \right) (x_k - x^*) = \\ &\quad \nabla^2 f(x_k) = \int_0^1 \nabla^2 f(x_k) d\tau = \nabla^2 f(x_k) \underbrace{\int_0^1 d\tau}_{1} \end{aligned}$$

## Сходимость

3.

$$\begin{aligned} &= \left( I - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \nabla^2 f(x_k) - \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \underbrace{\int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau}_{G_k} \underbrace{(x_k - x^*)} = \\ &= \underbrace{[\nabla^2 f(x_k)]^{-1}}_{G_k} \underbrace{G_k(x_k - x^*)} \end{aligned}$$

## Сходимость

3.

$$\begin{aligned} &= \left( I - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \nabla^2 f(x_k) - \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right) (x_k - x^*) = \\ &= [\nabla^2 f(x_k)]^{-1} \left( \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau \right) (x_k - x^*) = \\ &\quad = [\nabla^2 f(x_k)]^{-1} G_k (x_k - x^*) \end{aligned}$$

4. Введём:

$$G_k = \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))) d\tau .$$

## Сходимость

5. Попробуем оценить размер  $G_k$  с помощью  $r_k = \|x_k - x^*\|$ :

## Сходимость

5. Попробуем оценить размер  $G_k$  с помощью  $r_k = \|x_k - x^*\|$ :

$$\|G_k\| = \left\| \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right\| \leq$$

---

## Сходимость

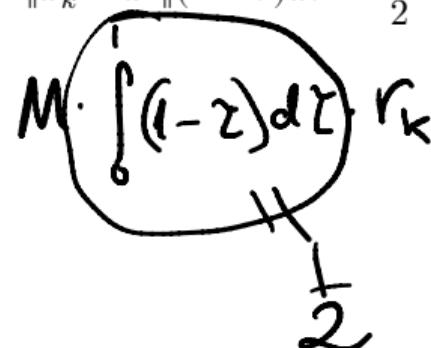
5. Попробуем оценить размер  $G_k$  с помощью  $r_k = \|x_k - x^*\|$ :

$$\begin{aligned}\|G_k\| &= \left\| \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right\| \leq \\ &\leq \int_0^1 \|\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))\| d\tau \leq \quad \underbrace{\text{(Липшицевость гессиана)}}_{\text{—}}\end{aligned}$$

## Сходимость

5. Попробуем оценить размер  $G_k$  с помощью  $r_k = \|x_k - x^*\|$ :

$$\begin{aligned}\|G_k\| &= \left\| \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right\| \leq \\ &\leq \int_0^1 \|\nabla^2 f(\underline{x_k}) - \nabla^2 f(\underline{x^* + \tau(x_k - x^*)})\| d\tau \leq \quad (\text{Липшицевость гессиана}) \\ &\leq \int_0^1 M \|x_k - x^* - \tau(x_k - x^*)\| d\tau = \int_0^1 M \|x_k - x^*\| (1 - \tau) d\tau = \frac{r_k}{2} M,\end{aligned}$$



## Сходимость

5. Попробуем оценить размер  $G_k$  с помощью  $r_k = \|x_k - x^*\|$ :

$$\begin{aligned}\|G_k\| &= \left\| \int_0^1 (\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau \right\| \leq \\ &\leq \int_0^1 \|\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))\| d\tau \leq \quad (\text{Липшицевость гессиана}) \\ &\leq \int_0^1 M \|x_k - x^* - \tau(x_k - x^*)\| d\tau = \int_0^1 M \|x_k - x^*\| (1 - \tau) d\tau = \frac{r_k}{2} M,\end{aligned}$$

6. Получаем:

$$r_{k+1} \leq \left\| [\nabla^2 f(x_k)]^{-1} \right\| \cdot \frac{r_k}{2} M \cdot r_k$$

и нам нужно оценить норму обратного гессиана

$$\frac{1}{M}$$

# Код

- Открыть в Colab

## Код

- Открыть в Colab
- Сравнение квазиньютоновских методов

# Код

- Открыть в Colab
- Сравнение квазиньютоновских методов
- Некоторые практические замечания о методе Ньютона