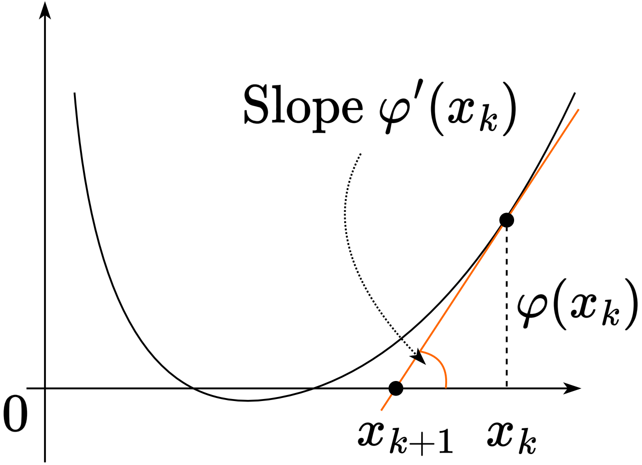
Метод Ньютона

Даня Меркулов

Методы Оптимизации в Машинном Обучении. ФКН ВШЭ

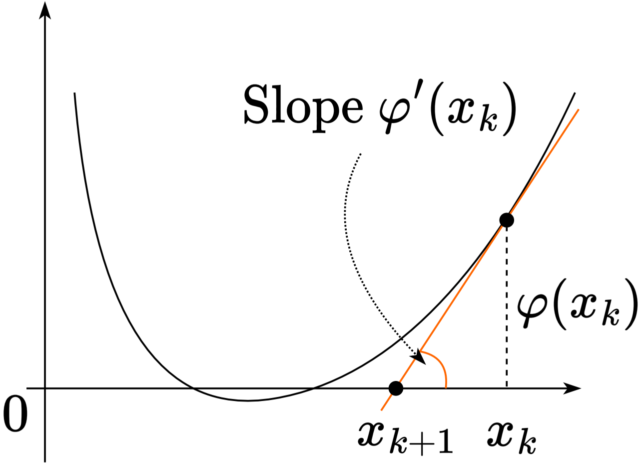
Метод Ньютона

Идея метода Ньютона для нахождения корней функции



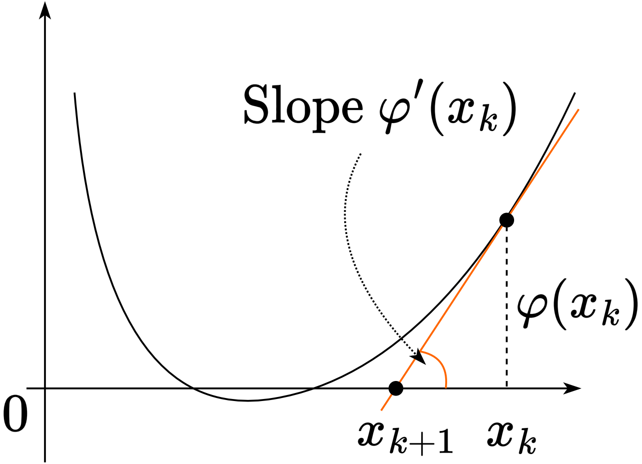
Рассмотрим функцию $\varphi(x) : \mathbb{R} \to \mathbb{R}$.

Идея метода Ньютона для нахождения корней функции



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

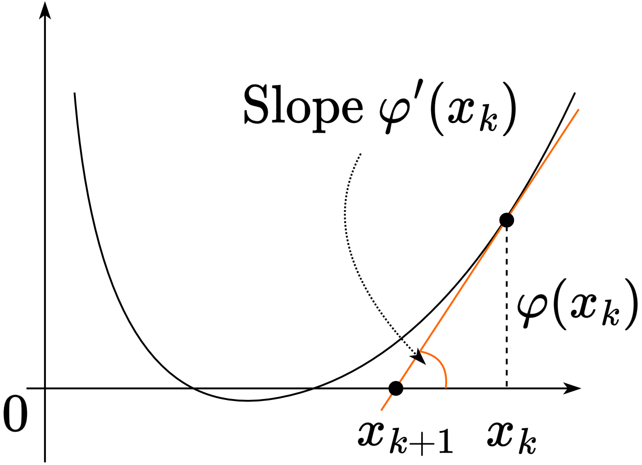


Рассмотрим функцию $\varphi(x) : \mathbb{R} \to \mathbb{R}$.

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

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

Идея метода Ньютона для нахождения корней функции



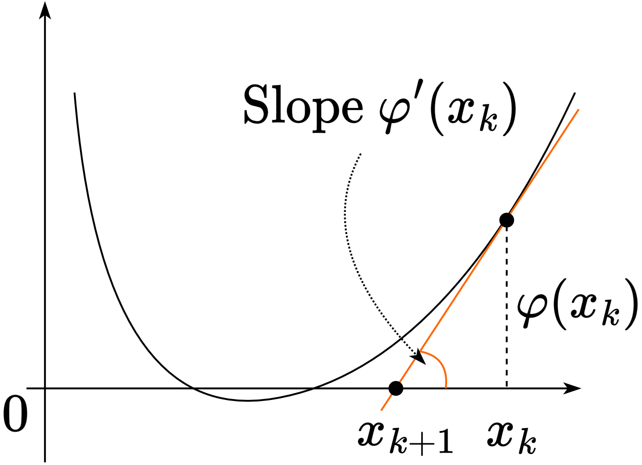
Рассмотрим функцию $\varphi(x) : \mathbb{R} \to \mathbb{R}$.

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

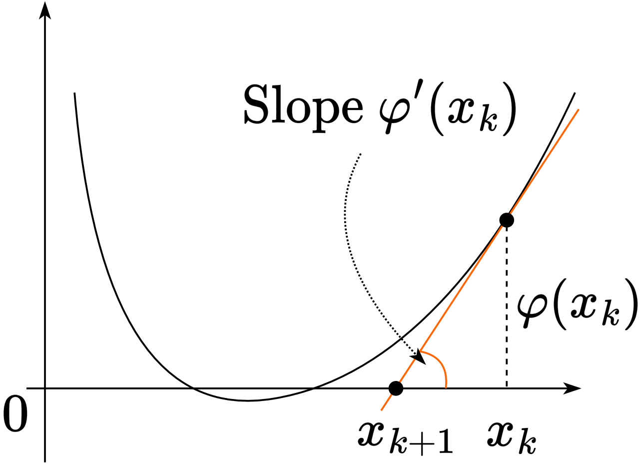
$$\varphi'(x_k) = \frac{\varphi(x_k)}{x_{k+1} - x_k}$$

Мы получаем итерационную схему:

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

[1] Мы фактически решаем задачу нахождения стационарных точек $\nabla f(x) = 0$

Идея метода Ньютона для нахождения корней функции



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

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

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

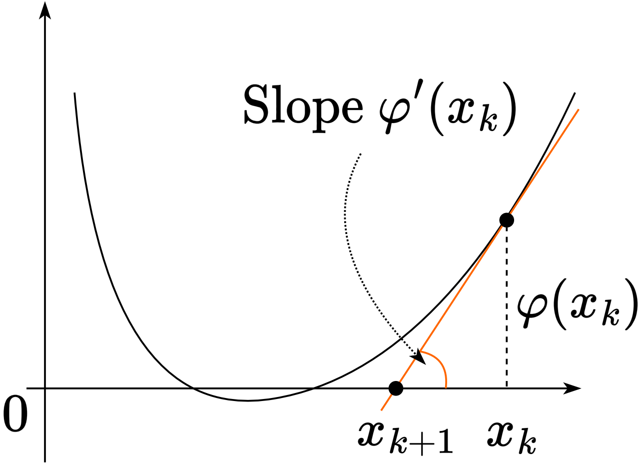
Мы получаем итерационную схему:

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

Этот метод станет методом оптимизации Ньютона в случае $f'(x) = \varphi(x)$[1]:

[1] Мы фактически решаем задачу нахождения стационарных точек $\nabla f(x) = 0$

Идея метода Ньютона для нахождения корней функции



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

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 $f'(x) = \varphi(x)$[1]:

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

[1] 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:

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:

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

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

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:

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

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

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

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:

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

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

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

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:

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

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

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

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:

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

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

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

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:

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

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

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

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:

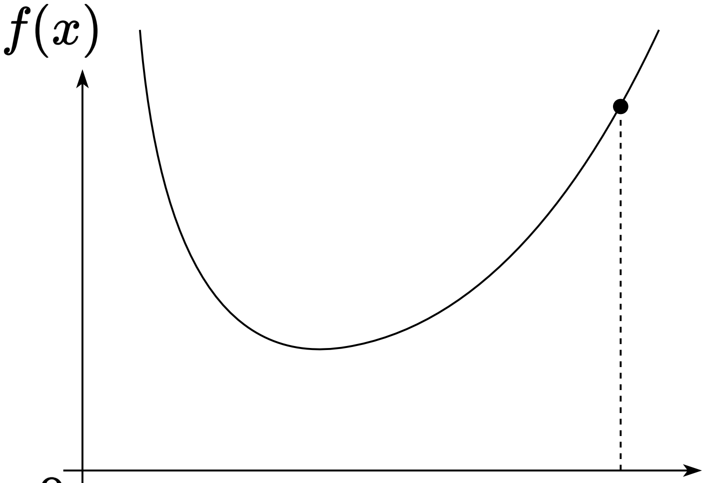
$$f_{x_k}^{II}(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle.$$

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

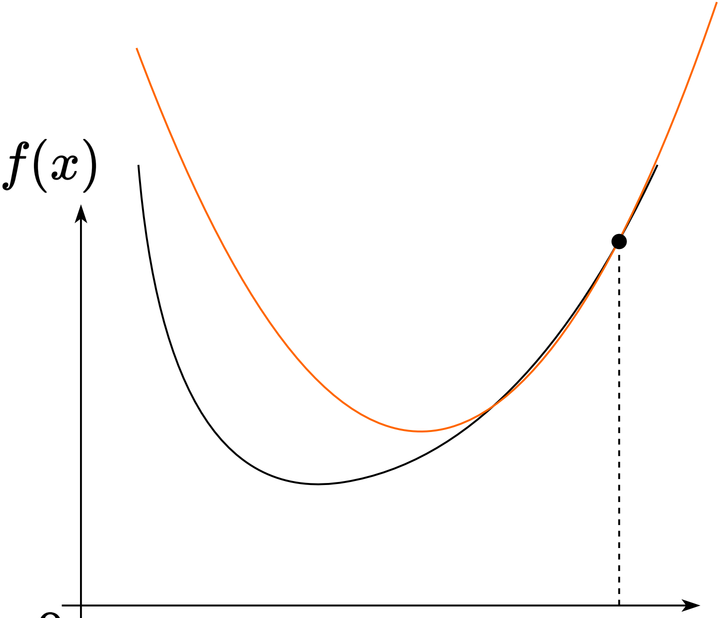
$$\begin{aligned}
\nabla f_{x_k}^{II}(x_{k+1}) &= \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) = 0 \\
\nabla^2 f(x_k)(x_{k+1} - x_k) &= -\nabla f(x_k) \\
\left[\nabla^2 f(x_k)\right]^{-1} \nabla^2 f(x_k)(x_{k+1} - x_k) &= -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k) \\
x_{k+1} &= x_k - \left[\nabla^2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

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

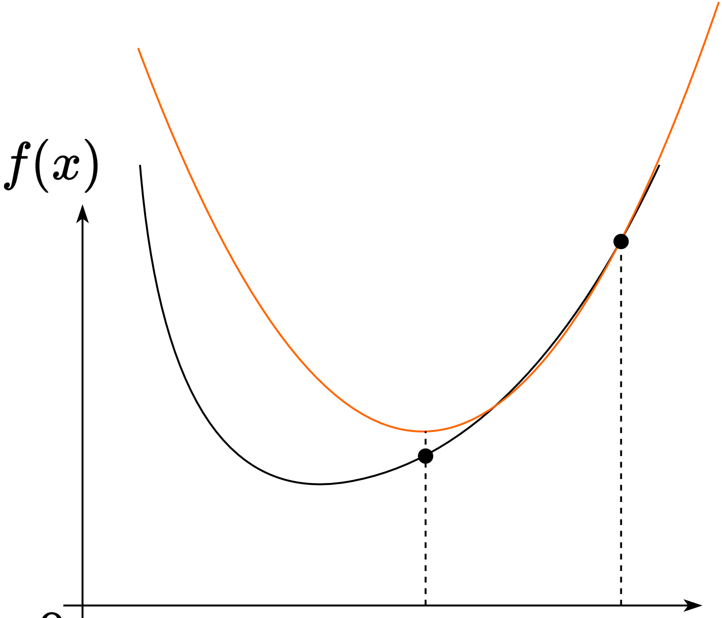
Метод Ньютона как оптимизация локальной квадратичной аппроксимации



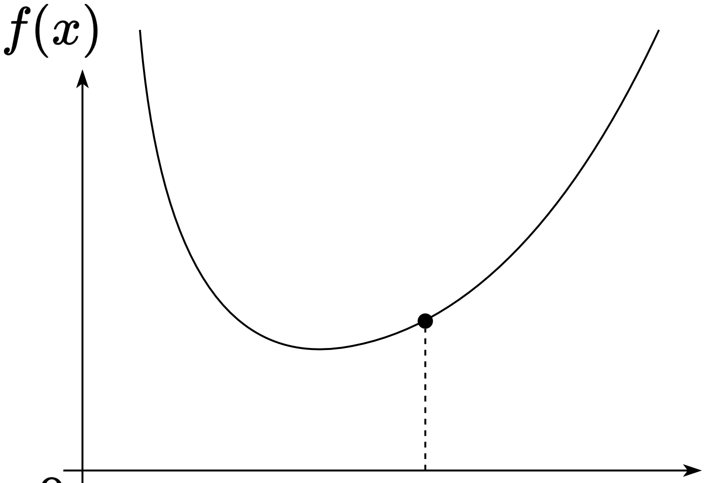
Метод Ньютона как оптимизация локальной квадратичной аппроксимации



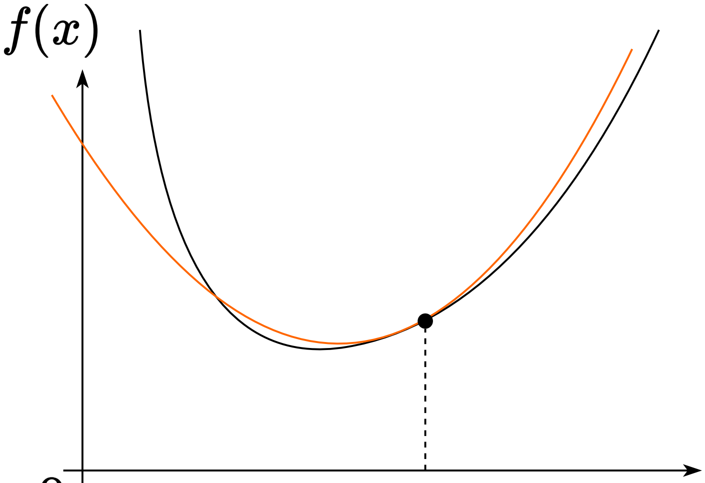
Метод Ньютона как оптимизация локальной квадратичной аппроксимации



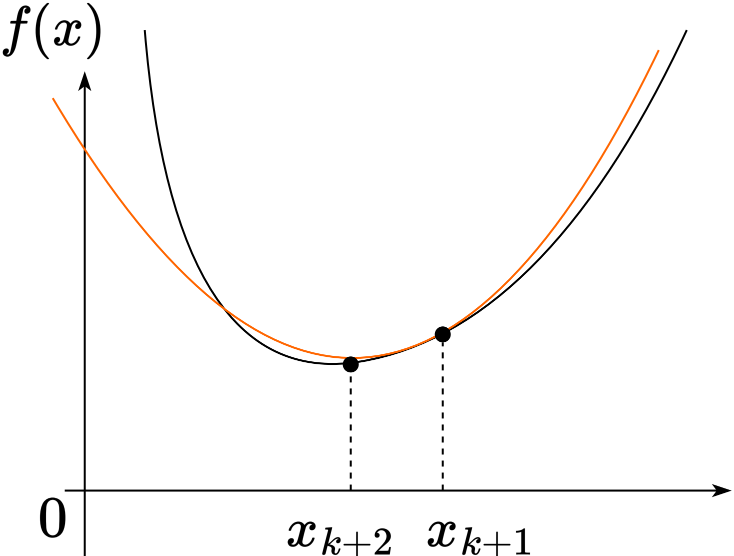
Метод Ньютона как оптимизация локальной квадратичной аппроксимации



Метод Ньютона как оптимизация локальной квадратичной аппроксимации



Метод Ньютона как оптимизация локальной квадратичной аппроксимации



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 $\|x_0 - x^*\| < \frac{2\mu}{3M}$:

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

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

1. 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: $\mu I_n \preceq \nabla^2 f(x) \preceq LI_n$. 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

1. Мы будем использовать формулу Ньютона-Лейбница

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

2. 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: $\mu I_n \preceq \nabla^2 f(x) \preceq LI_n$. 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

1. Мы будем использовать формулу Ньютона-Лейбница

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

2. Мы будем отслеживать расстояние до решения

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

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

1. Мы будем использовать формулу Ньютона-Лейбница

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

2. Мы будем отслеживать расстояние до решения

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

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

Сходимость

3.

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

Сходимость

3.

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

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

Сходимость

3.

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

Сходимость

3.

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

Сходимость

3.
$$
\begin{aligned}
&= \left(I - \left[\nabla^2 f(x_k)\right]^{-1} \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau\right)(x_k - x^*) = \\
&= \left[\nabla^2 f(x_k)\right]^{-1} \left(\nabla^2 f(x_k) - \int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*)) d\tau\right)(x_k - x^*) = \\
&= \left[\nabla^2 f(x_k)\right]^{-1} \left(\int_0^1 \left(\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))d\tau\right)\right)(x_k - x^*) = \\
&= \left[\nabla^2 f(x_k)\right]^{-1} G_k (x_k - x^*)
\end{aligned}
$$

4. Введём:

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

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 $r_k = \|x_k - x^*\|$:

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

$$\leq \int_0^1 \left\| \nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*)) \right\| d\tau \leq \qquad \text{(Липшицевость гессиана)}$$

Сходимость

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

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

$$\leq \int_0^1 \left\| \nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*)) \right\| d\tau \leq \qquad \text{(Липшицевость гессиана)}$$

$$\leq \int_0^1 M\|x_k - x^* - \tau(x_k - x^*)\| d\tau = \int_0^1 M\|x_k - x^*\|(1-\tau) d\tau = \frac{r_k}{2} M,$$

Сходимость

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

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

$$\leq \int_0^1 \left\| \nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*)) \right\| d\tau \leq \qquad \text{(Липшицевость гессиана)}$$

$$\leq \int_0^1 M\|x_k - x^* - \tau(x_k - x^*)\| d\tau = \int_0^1 M\|x_k - x^*\|(1-\tau)d\tau = \frac{r_k}{2}M,$$

6. Получаем:

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

и нам нужно оценить норму обратного гессиана

Сходимость

7. Из липшицевости и симметричности гессиана:

Сходимость

7. Из липшицевости и симметричности гессиана:

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

Сходимость

7. Из липшицевости и симметричности гессиана:

$$
\begin{aligned}
\nabla^2 f(x_k) - \nabla^2 f(x^*) &\succeq -Mr_k I_n \\
\nabla^2 f(x_k) &\succeq \nabla^2 f(x^*) - Mr_k I_n
\end{aligned}
$$

Сходимость

7. Из липшицевости и симметричности гессиана:

$$
\begin{aligned}
\nabla^2 f(x_k) - \nabla^2 f(x^*) &\succeq -Mr_k I_n \\
\nabla^2 f(x_k) &\succeq \nabla^2 f(x^*) - Mr_k I_n \\
\nabla^2 f(x_k) &\succeq \mu I_n - Mr_k I_n
\end{aligned}
$$

Сходимость

7. Из липшицевости и симметричности гессиана:

$$
\begin{aligned}
\nabla^2 f(x_k) - \nabla^2 f(x^*) &\succeq -Mr_k I_n \\
\nabla^2 f(x_k) &\succeq \nabla^2 f(x^*) - Mr_k I_n \\
\nabla^2 f(x_k) &\succeq \mu I_n - Mr_k I_n \\
\nabla^2 f(x_k) &\succeq (\mu - Mr_k) I_n
\end{aligned}
$$

Сходимость

7. Из липшицевости и симметричности гессиана:

$$
\begin{aligned}
\nabla^2 f(x_k) - \nabla^2 f(x^*) &\succeq -Mr_k I_n \\
\nabla^2 f(x_k) &\succeq \nabla^2 f(x^*) - Mr_k I_n \\
\nabla^2 f(x_k) &\succeq \mu I_n - Mr_k I_n \\
\nabla^2 f(x_k) &\succeq (\mu - Mr_k) I_n
\end{aligned}
$$

8. Из сильной выпуклости следует, что $\nabla^2 f(x_k) \succ 0$, i.e. $r_k < \frac{\mu}{M}$.

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

Сходимость

7. Из липшицевости и симметричности гессиана:

$$
\begin{aligned}
\nabla^2 f(x_k) - \nabla^2 f(x^*) &\succeq -Mr_k I_n \\
\nabla^2 f(x_k) &\succeq \nabla^2 f(x^*) - Mr_k I_n \\
\nabla^2 f(x_k) &\succeq \mu I_n - Mr_k I_n \\
\nabla^2 f(x_k) &\succeq (\mu - Mr_k) I_n
\end{aligned}
$$

8. Из сильной выпуклости следует, что $\nabla^2 f(x_k) \succ 0$, i.e. $r_k < \frac{\mu}{M}$.

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

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

$$
\begin{aligned}
\frac{r_k^2 M}{2(\mu - Mr_k)} &< r_k \\
\frac{M}{2(\mu - Mr_k)} r_k &< 1 \\
Mr_k &< 2(\mu - Mr_k) \\
3Mr_k &< 2\mu \\
r_k &< \frac{2\mu}{3M}
\end{aligned}
$$

Сходимость

7. Из липшицевости и симметричности гессиана:

$$
\begin{aligned}
\nabla^2 f(x_k) - \nabla^2 f(x^*) &\succeq -Mr_k I_n \\
\nabla^2 f(x_k) &\succeq \nabla^2 f(x^*) - Mr_k I_n \\
\nabla^2 f(x_k) &\succeq \mu I_n - Mr_k I_n \\
\nabla^2 f(x_k) &\succeq (\mu - Mr_k) I_n
\end{aligned}
$$

8. Из сильной выпуклости следует, что $\nabla^2 f(x_k) \succ 0$, i.e. $r_k < \frac{\mu}{M}$.

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

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

$$
\begin{aligned}
\frac{r_k^2 M}{2(\mu - Mr_k)} &< r_k \\
\frac{M}{2(\mu - Mr_k)} r_k &< 1 \\
Mr_k &< 2(\mu - Mr_k) \\
3Mr_k &< 2\mu \\
r_k &< \frac{2\mu}{3M}
\end{aligned}
$$

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

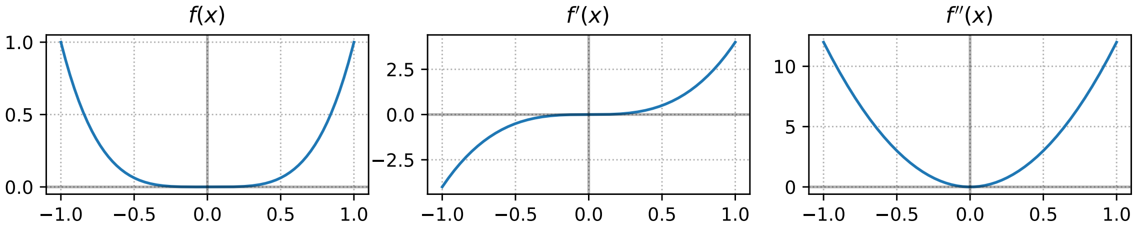
$$
r_{k+1} \leq \frac{r_k^2 M}{2(\mu - Mr_k)} < \frac{3Mr_k^2}{2\mu}
$$

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

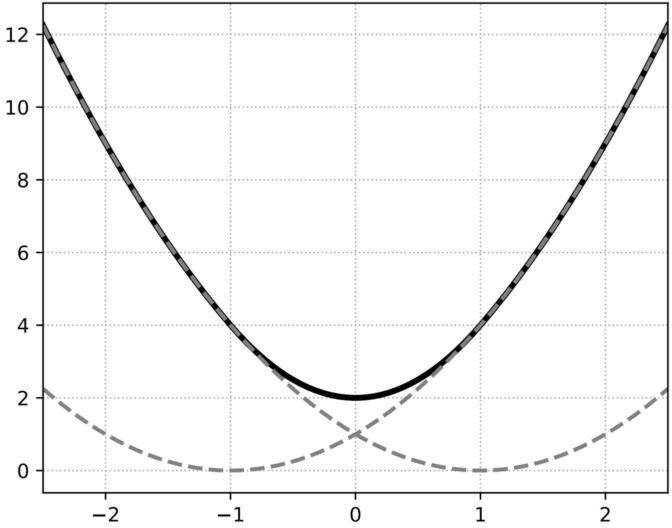
$$f(x) = x^4 \qquad f'(x) = 4x^3 \qquad f''(x) = 12x^2$$



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

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

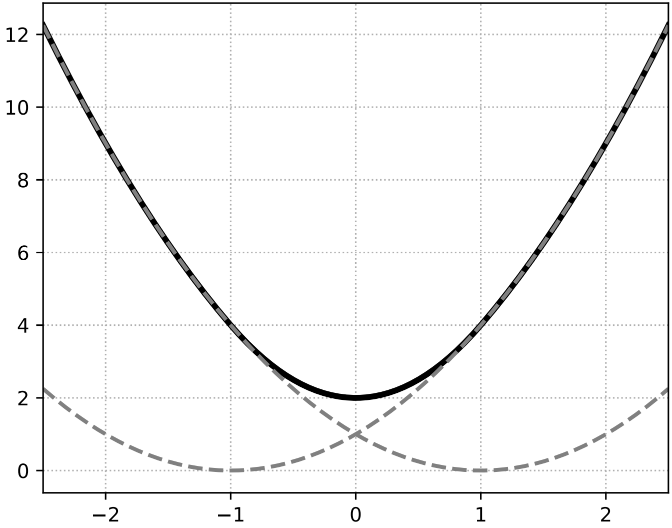
Локальная сходимость метода Ньютона для гладкой сильно выпуклой $f(x)$



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

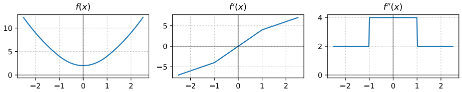
Эта функция сильно выпукла, но вторая производная не является липшицевой.

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

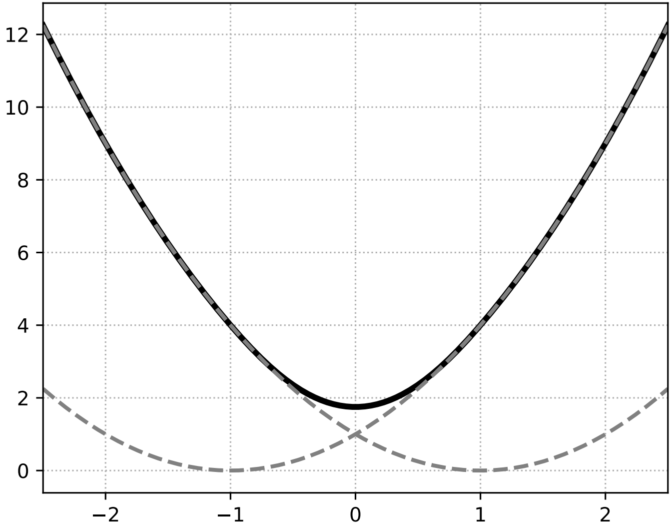


$$
f(x) = \begin{cases} (x-1)^2, & x \leq -1 \\ 2x^2 + 2, & -1 < x < 1 \\ (x+1)^2, & x \geq 1 \end{cases}
$$

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

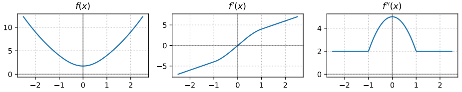


Локальная сходимость метода Ньютона даже если $\nabla^2 f$ липшицев

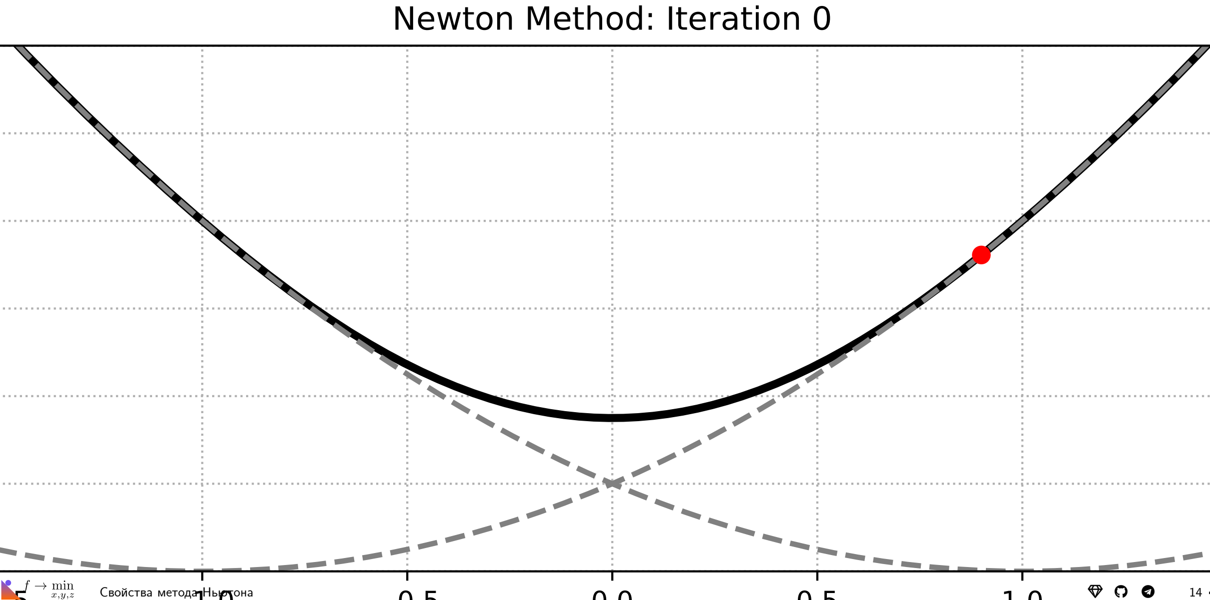


$$
f(x) = \begin{cases} (x-1)^2, & x \leq -1 \\ -\frac{1}{4}x^4 + \frac{5}{2}x^2 + \frac{7}{4}, & -1 < x < 1 \\ (x+1)^2, & x \geq 1 \end{cases}
$$

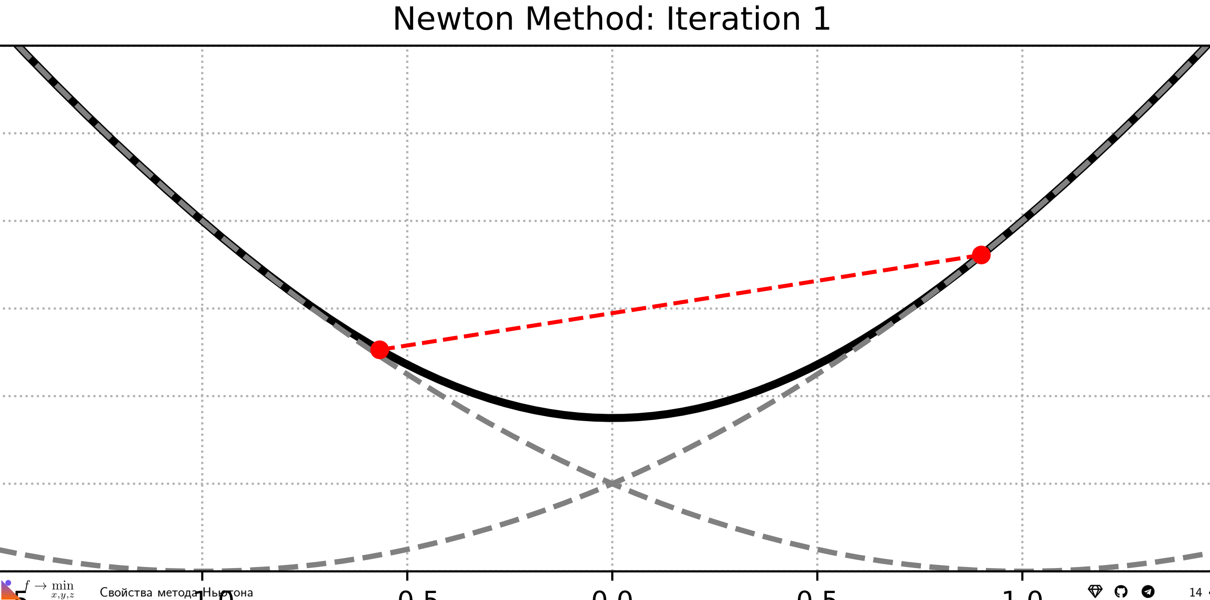
Эта функция сильно выпукла и вторая производная является липшицевой.



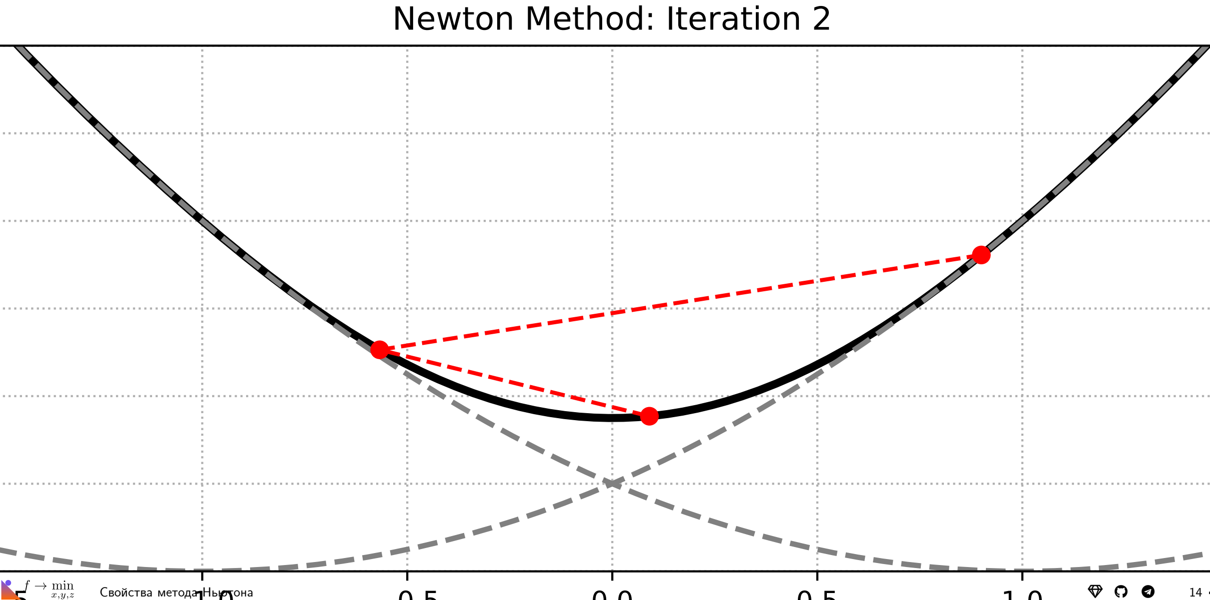
Локальная сходимость метода Ньютона. Хорошая инициализация



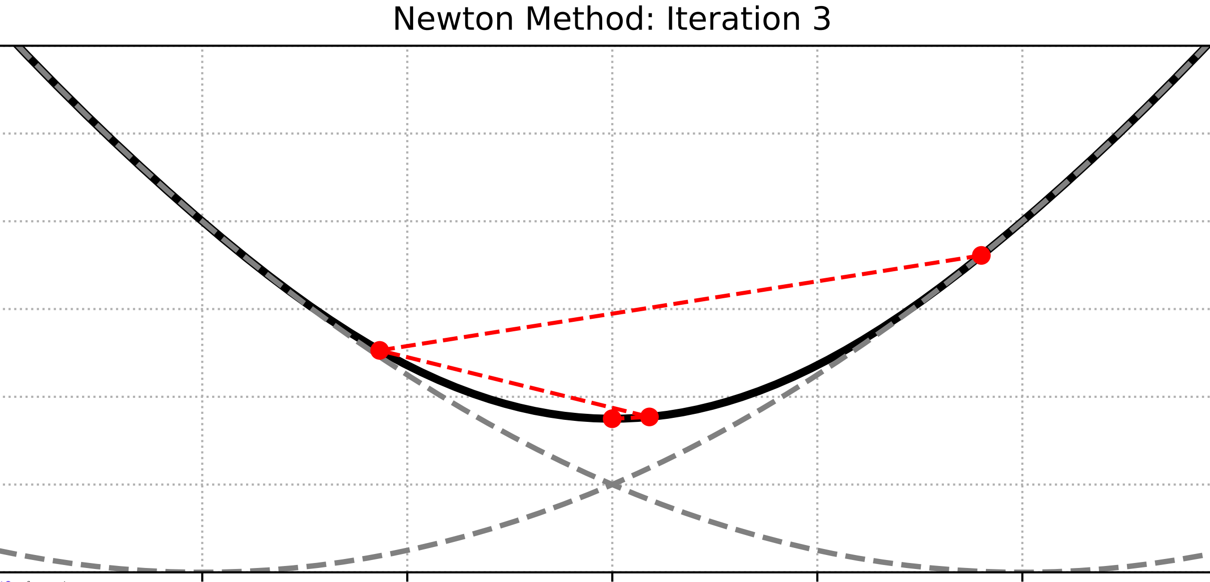
Локальная сходимость метода Ньютона. Хорошая инициализация



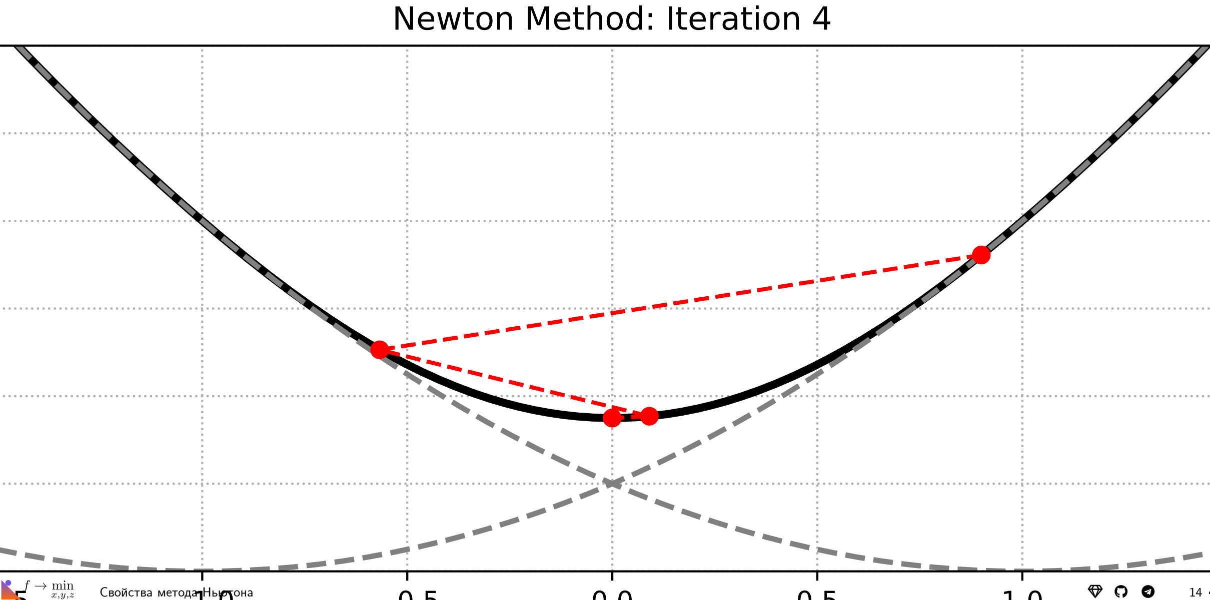
Локальная сходимость метода Ньютона. Хорошая инициализация



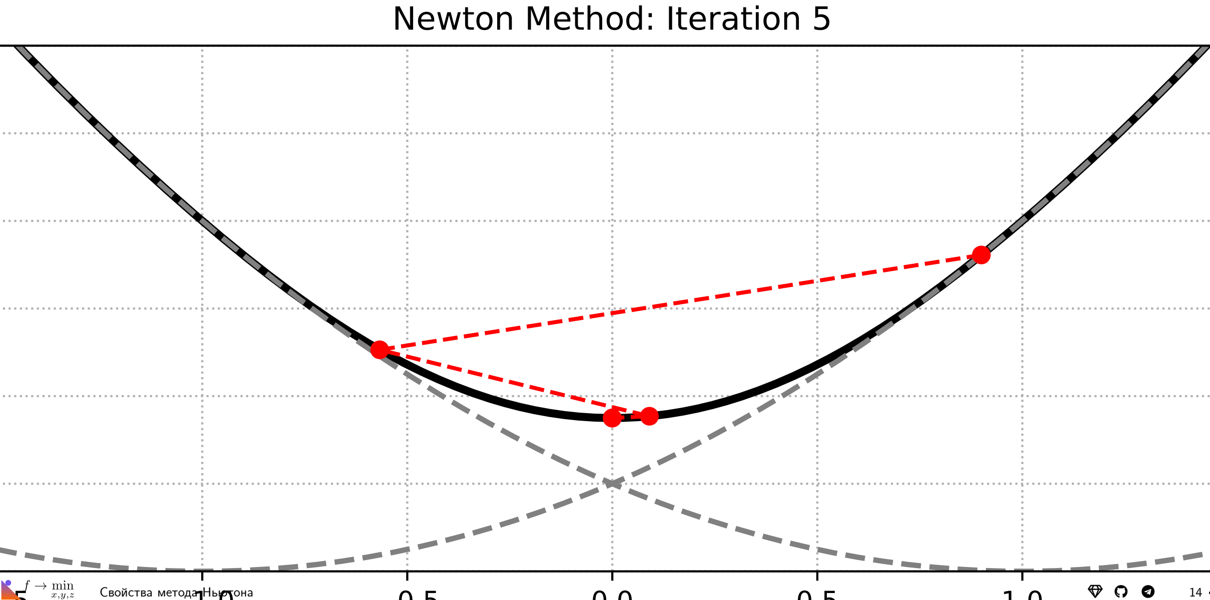
Локальная сходимость метода Ньютона. Хорошая инициализация



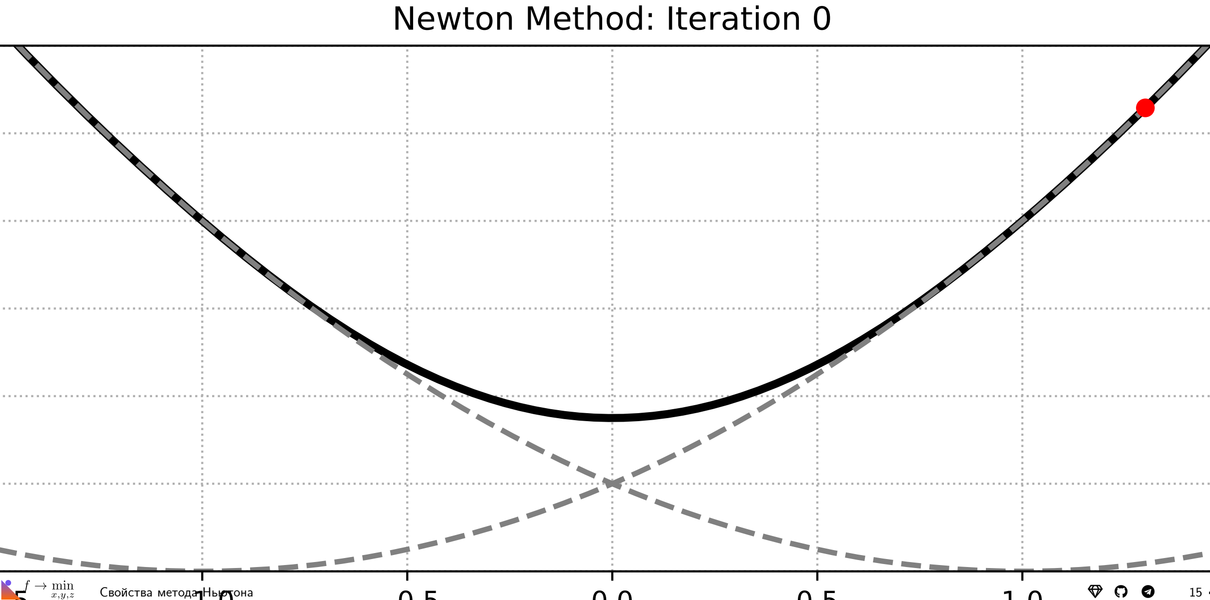
Локальная сходимость метода Ньютона. Хорошая инициализация



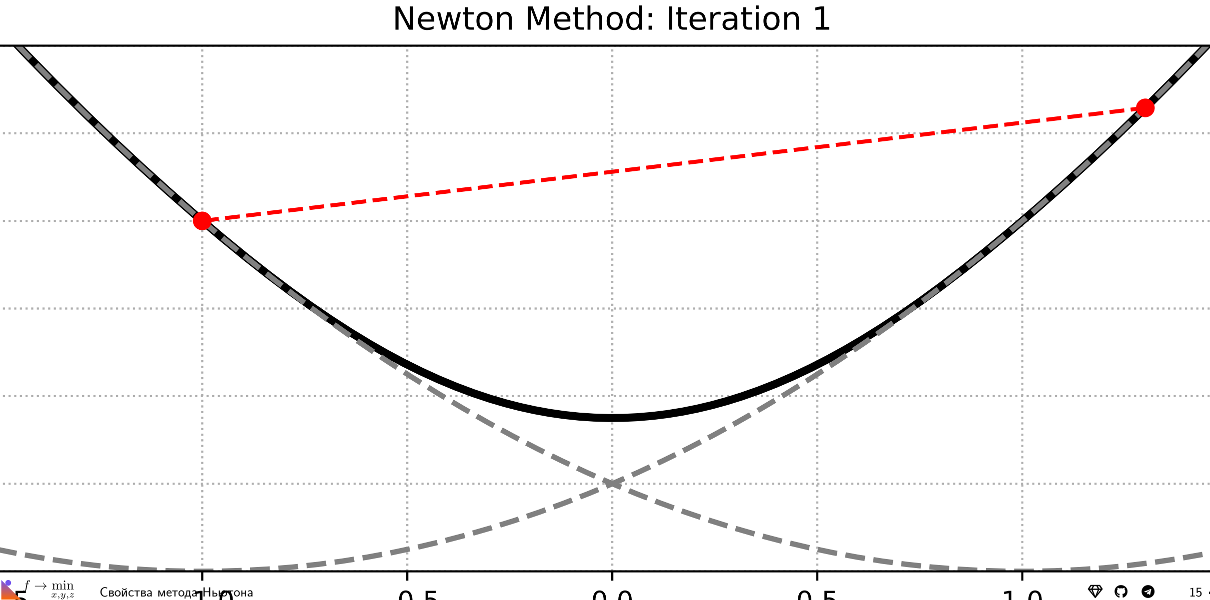
Локальная сходимость метода Ньютона. Хорошая инициализация



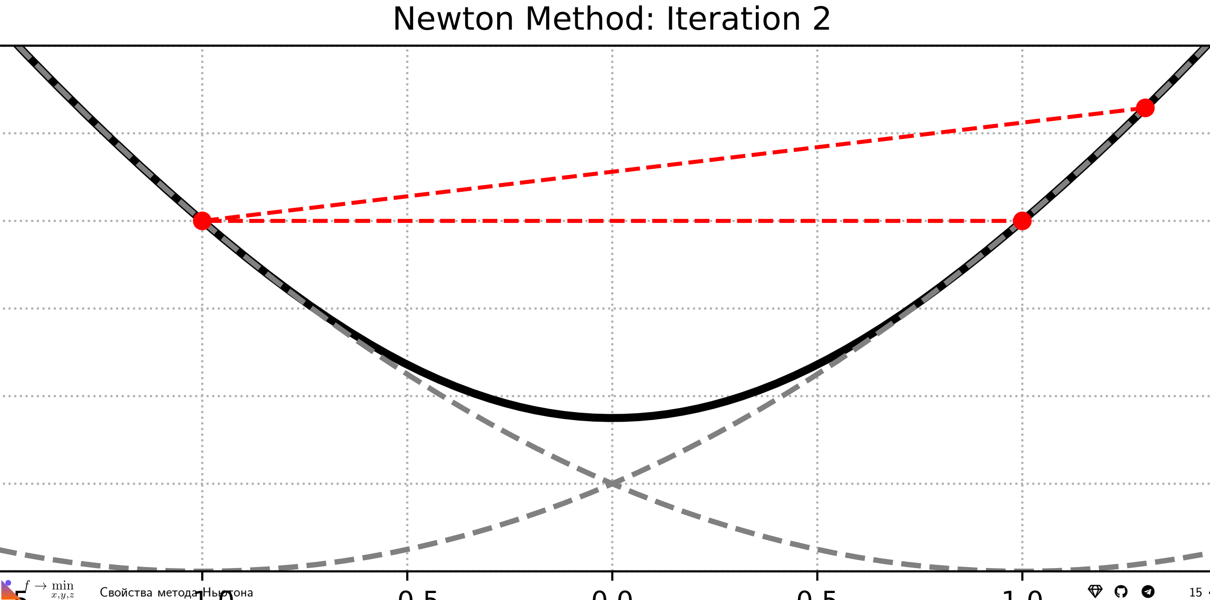
Локальная сходимость метода Ньютона. Плохая инициализация



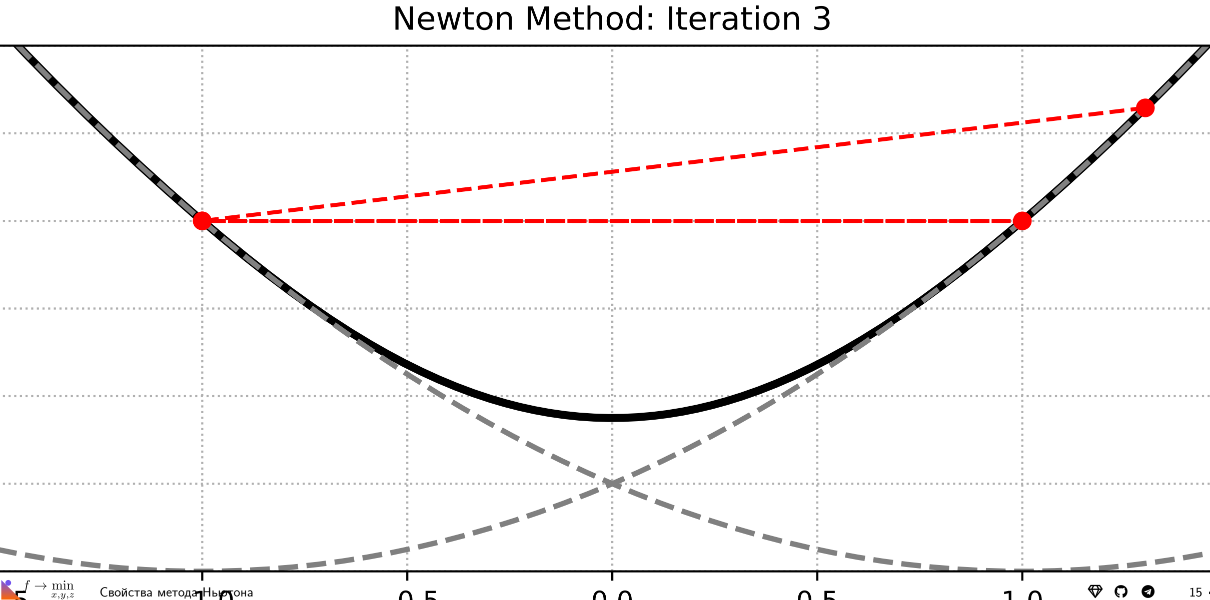
Локальная сходимость метода Ньютона. Плохая инициализация



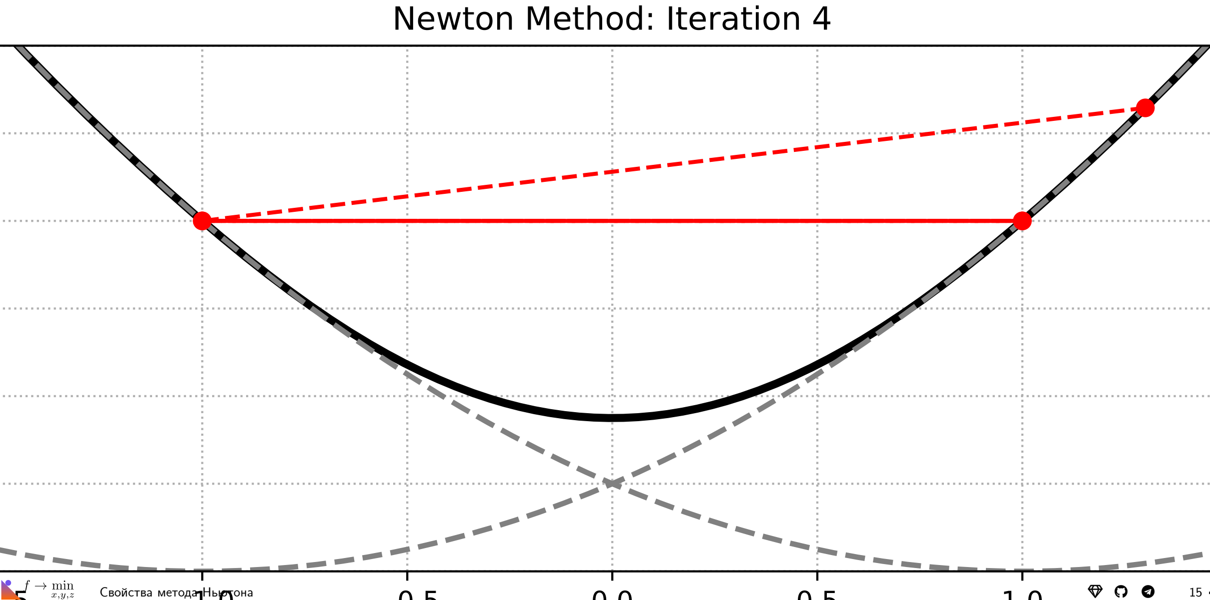
Локальная сходимость метода Ньютона. Плохая инициализация



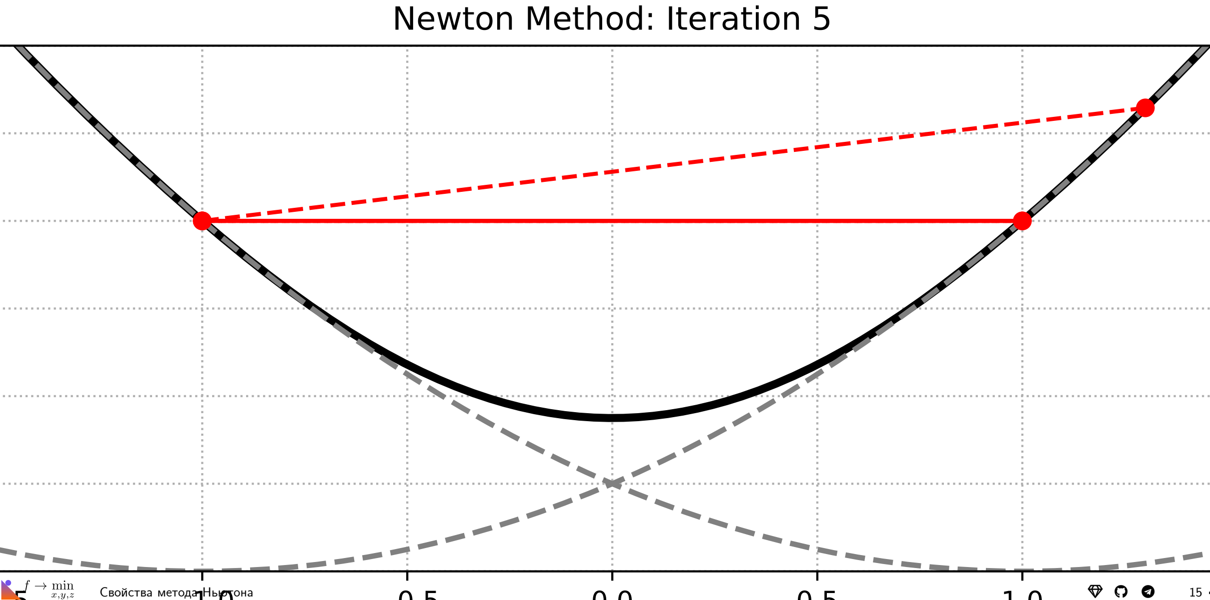
Локальная сходимость метода Ньютона. Плохая инициализация



Локальная сходимость метода Ньютона. Плохая инициализация



Локальная сходимость метода Ньютона. Плохая инициализация



Проблемы метода Ньютона

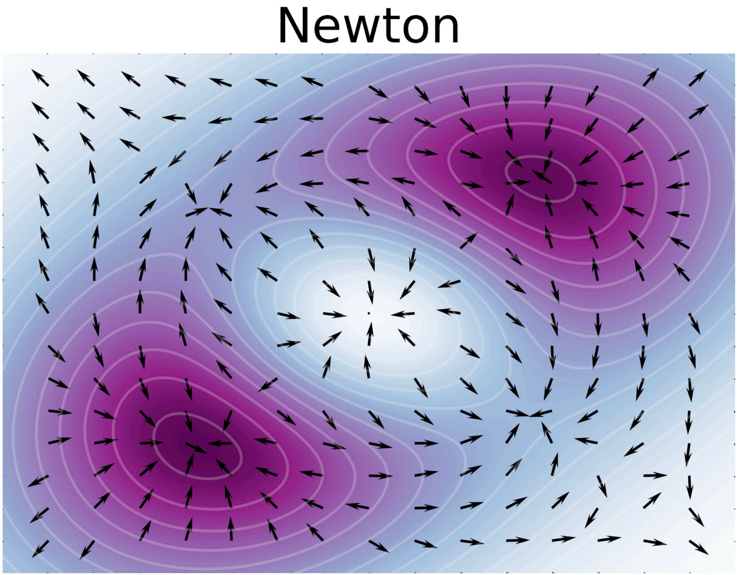


Рисунок 1: Animation

Проблемы метода Ньютона

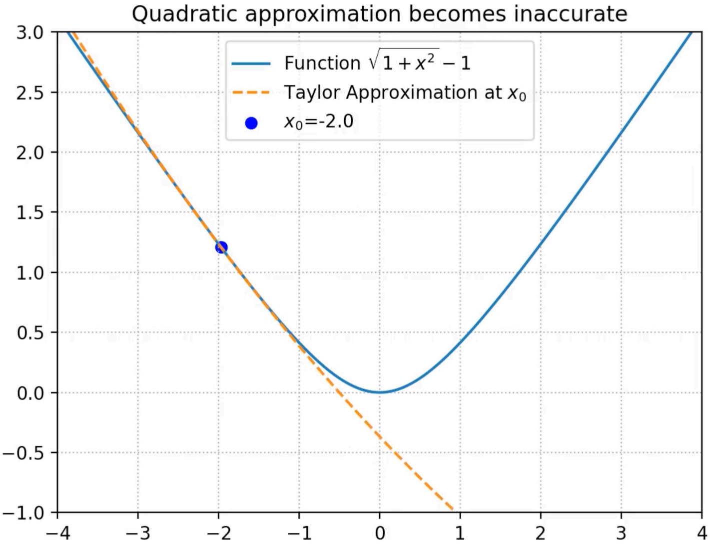


Рисунок 2: Animation

Метод Ньютона для квадратичной задачи (линейной регрессии)

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

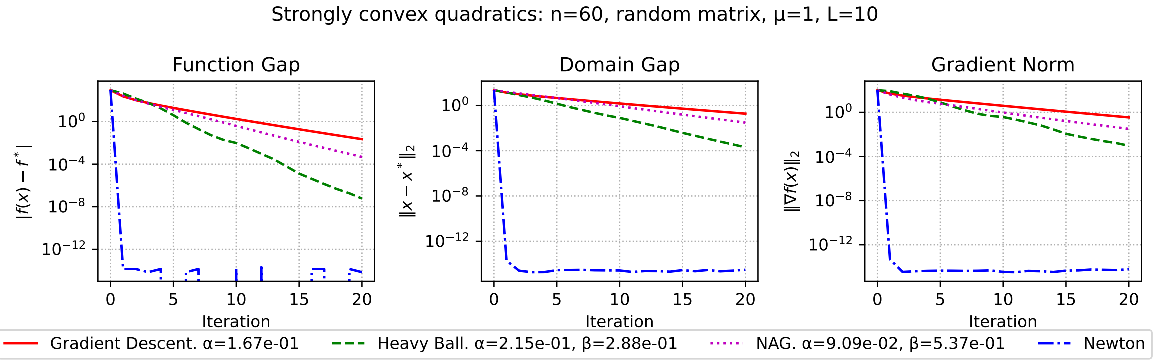


Рисунок 3: Так как задача - квадратичная, то метод Ньютона сходится за один шаг.

Метод Ньютона для квадратичной задачи (линейной регрессии)

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

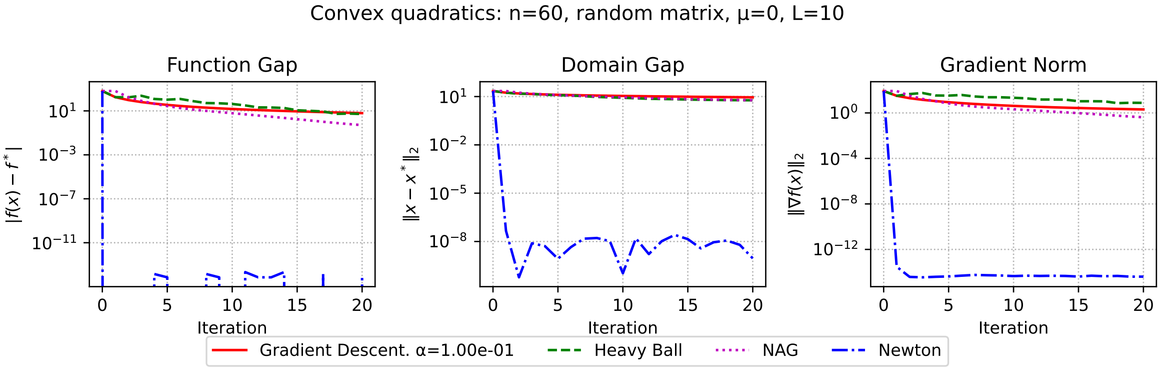


Рисунок 4: В этом случае метод Ньютона тоже крайне быстро сходится, однако, отметим, что это происходит благодаря тому, что минимальное собственное число гессиана не 0, а около 10^{-8}. Если применять метод Ньютона в наивной форме с обращением матрицы, то получится ошибка, так как матрица вырождена. 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)

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

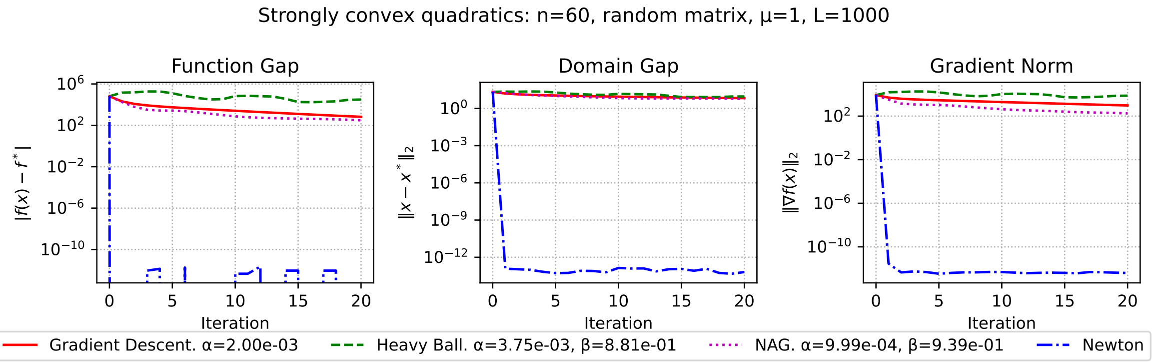


Рисунок 5: Здесь число обусловленности гессиана в 1000 раз больше, чем в предыдущем случае, и метод Ньютона сходится за 1 итерацию.

Метод Ньютона для задачи бинарной логистической регрессии

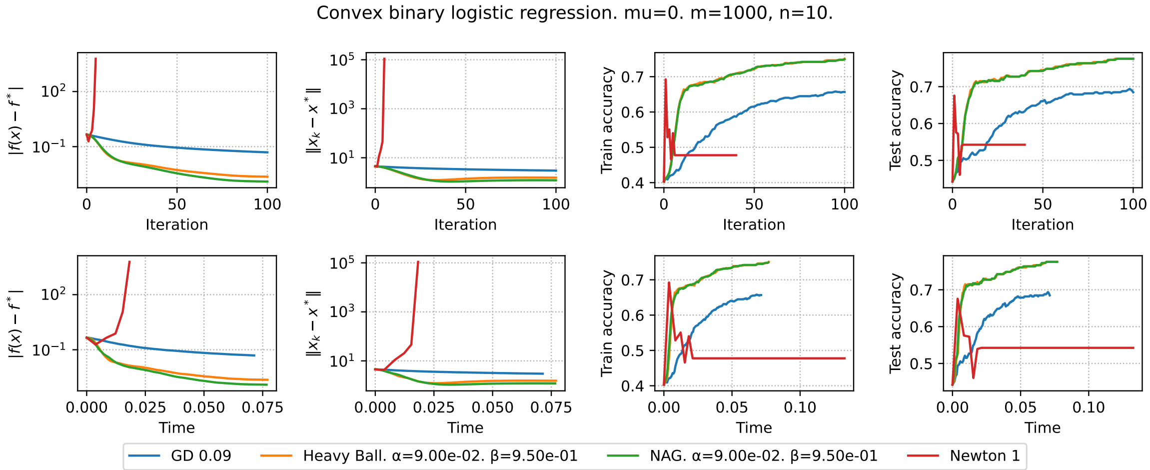


Рисунок 6: Наблюдается расходимость метода Ньютона. Сразу отметим, что в задаче нет регуляризации и гарантии сильной выпуклости. А также нет гарантий того, что мы инициализируем метод в окрестности решения.

Метод Ньютона для задачи бинарной логистической регрессии

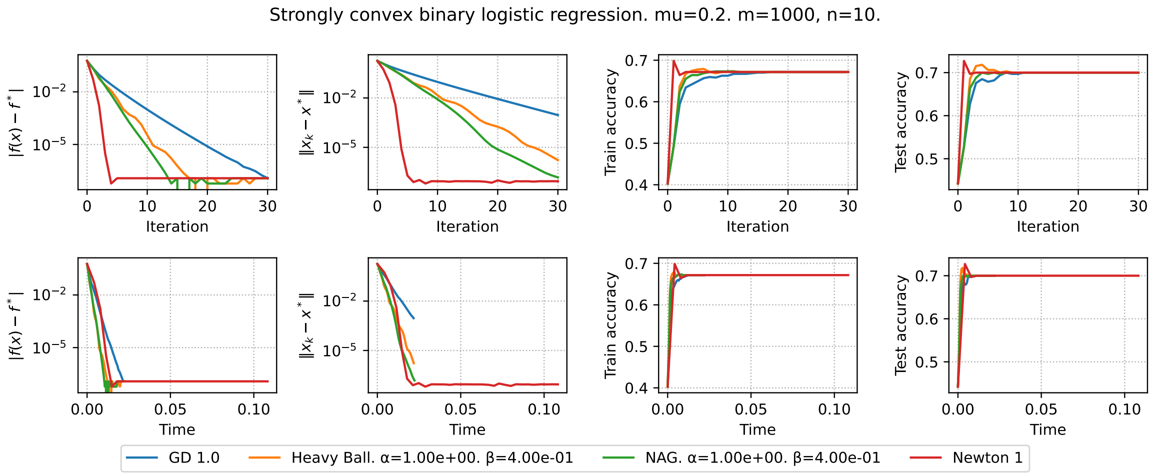


Рисунок 7: Добавление регуляризации гарантирует сильную выпуклость, наблюдается сходимость метода Ньютона.

Метод Ньютона для задачи бинарной логистической регрессии

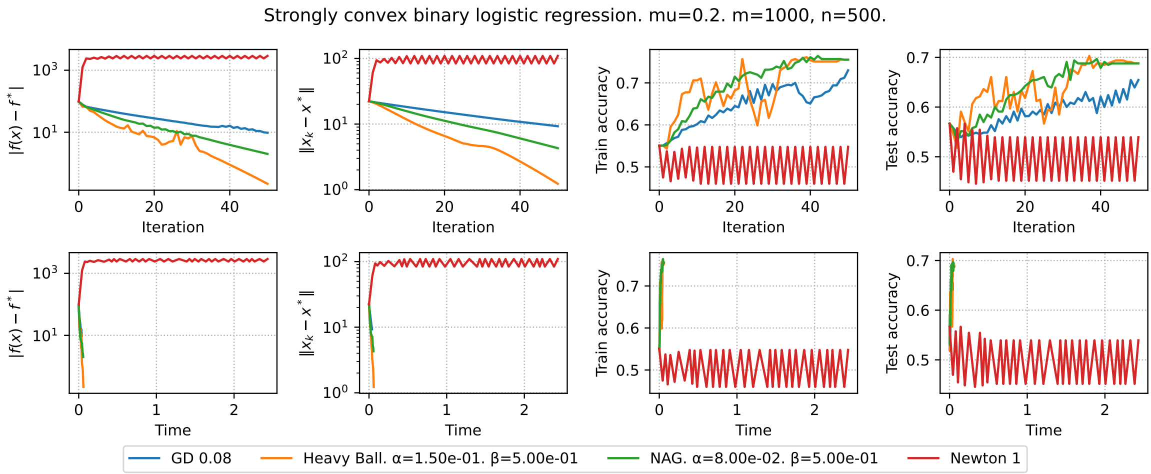


Рисунок 8: Увеличим размерность в 50 раз и наблюдаем расходимость метода Ньютона. Это можно связать с тем, что мы инициализируем метод в точке, далекой от решения

Метод Ньютона для задачи бинарной логистической регрессии

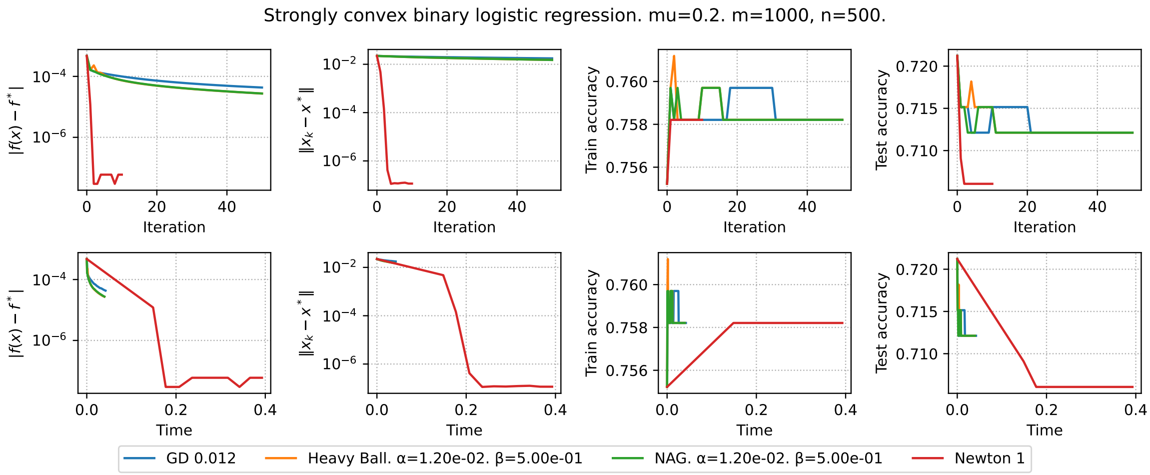


Рисунок 9: Не меняя задачу, изменим начальную точку и наблюдаем квадратичную сходимость метода Ньютона. Однако, обратите внимание на время работы. Уже при небольшой размерности, метод Ньютона работает значительно дольше, чем градиентные методы.

Задача нахождения аналитического центра многогранника

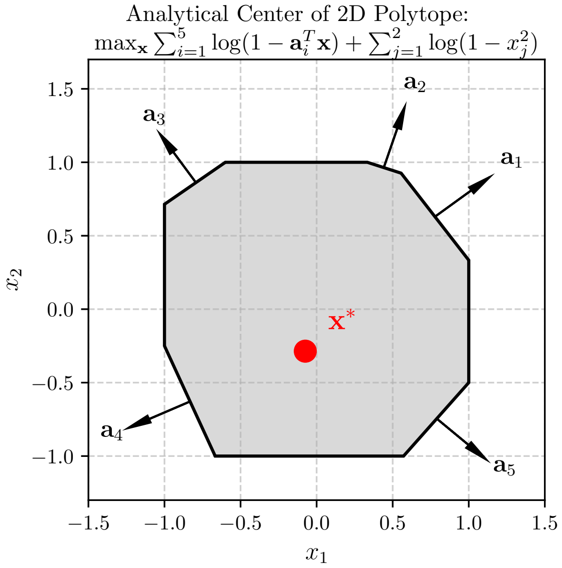
Найти точку $x \in \mathbb{R}^n$, которая максимизирует сумму логарифмов расстояний до границ политопа:

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

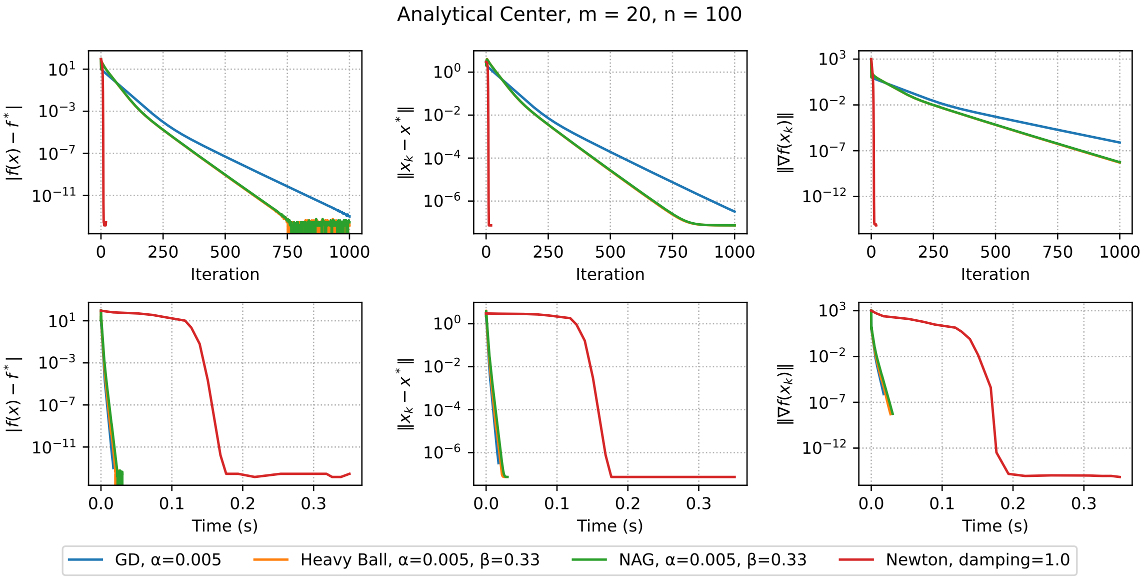
или, эквивалентно, минимизирует:

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

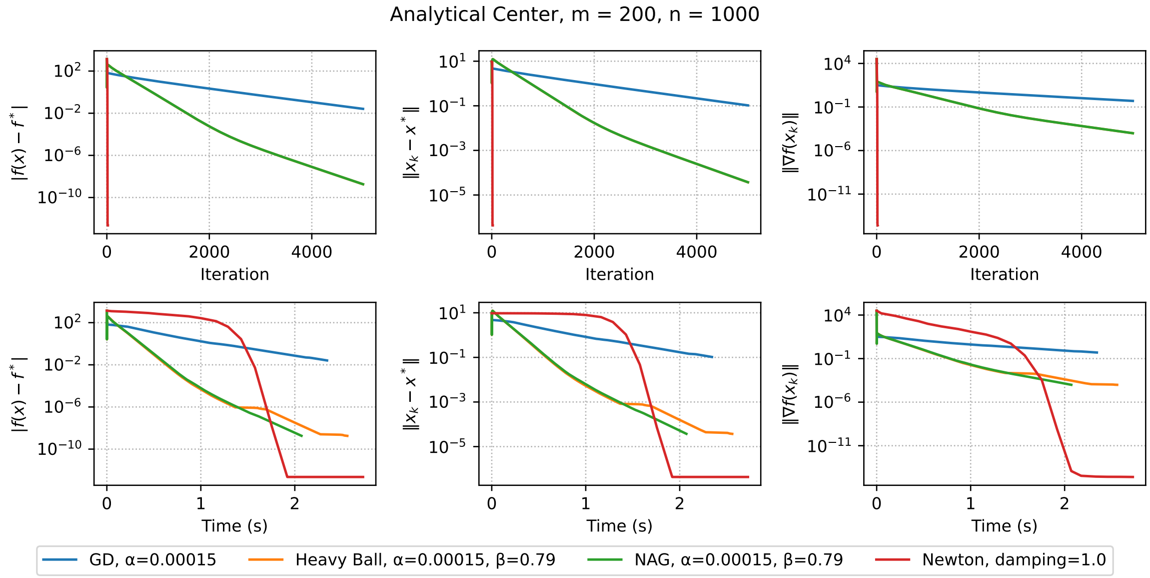
при ограничениях: - $a_i^T x < 1$ для всех $i = 1, \dots, m$, где a_i - строки матрицы A^T - $|x_j| < 1$ для всех $j = 1, \dots, n$
Аналитический центр многогранника - это точка, которая максимально удалена от всех границ многогранника в смысле логарифмического барьера. Эта концепция широко используется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



Задача нахождения аналитического центра многогранника



Задача нахождения аналитического центра многогранника



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:

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

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:

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

Раскрывая это, мы получаем:

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

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:

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

Раскрывая это, мы получаем:

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

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

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

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:

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

Раскрывая это, мы получаем:

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

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

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

Таким образом, правило обновления для x выглядит так:

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

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:

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

Раскрывая это, мы получаем:

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

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

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

Таким образом, правило обновления для x выглядит так:

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

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

Summary

Плюсы:

- Квадратичная сходимость вблизи решения x^*

Минусы:

Summary

Плюсы:

- Квадратичная сходимость вблизи решения x^*
- Аффинная инвариантность

Минусы:

Summary

Плюсы:

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

Минусы:

Summary

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:

Summary

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

Summary

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

Summary

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^*

Summary

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 $\to$ направление $-(f''(x))^{-1}f'(x)$ может не быть направлением спуска

Quasi-Newton methods

The idea of adapive metrics

Given $f(x)$ and a point x_0. Define $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$ as the set of points with distance ε to x_0. Here we presume the existence of a distance function $d(x, x_0)$.

The idea of adapive metrics

Given $f(x)$ and a point x_0. Define $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$ as the set of points with distance ε to x_0. Here we presume the existence of a distance function $d(x, x_0)$.

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

The idea of adapive metrics

Given $f(x)$ and a point x_0. Define $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$ as the set of points with distance ε to x_0. Here we presume the existence of a distance function $d(x, x_0)$.

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

Then, we can define another steepest descent direction in terms of minimizer of function on a sphere:

The idea of adapive metrics

Given $f(x)$ and a point x_0. Define $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$ as the set of points with distance ε to x_0. Here we presume the existence of a distance function $d(x, x_0)$.

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

Then, we can define another steepest descent direction in terms of minimizer of function on a sphere:

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

The idea of adapive metrics

Given $f(x)$ and a point x_0. Define $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$ as the set of points with distance ε to x_0. Here we presume the existence of a distance function $d(x, x_0)$.

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

Then, we can define another steepest descent direction in terms of minimizer of function on a sphere:

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

Let us assume that the distance is defined locally by some metric A:

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

The idea of adapive metrics

Given $f(x)$ and a point x_0. Define $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$ as the set of points with distance ε to x_0. Here we presume the existence of a distance function $d(x, x_0)$.

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

Then, we can define another steepest descent direction in terms of minimizer of function on a sphere:

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

Let us assume that the distance is defined locally by some metric A:

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

Let us also consider first order Taylor approximation of a function $f(x)$ near the point x_0:

$$
f(x_0 + \delta x) \approx f(x_0) + \nabla f(x_0)^\top \delta x \tag{1}
$$

Now we can explicitly pose a problem of finding s, as it was stated above.

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

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

The idea of adapive metrics

Given $f(x)$ and a point x_0. Define $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$ as the set of points with distance ε to x_0. Here we presume the existence of a distance function $d(x, x_0)$.

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

Then, we can define another steepest descent direction in terms of minimizer of function on a sphere:

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

Let us assume that the distance is defined locally by some metric A:

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

Let us also consider first order Taylor approximation of a function $f(x)$ near the point x_0:

$$
f(x_0 + \delta x) \approx f(x_0) + \nabla f(x_0)^\top \delta x \tag{1}
$$

Now we can explicitly pose a problem of finding s, as it was stated above.

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

Using equation (1 it can be written as:

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

The idea of adapive metrics

Given $f(x)$ and a point x_0. Define $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$ as the set of points with distance ε to x_0. Here we presume the existence of a distance function $d(x, x_0)$.

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

Then, we can define another steepest descent direction in terms of minimizer of function on a sphere:

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

Let us assume that the distance is defined locally by some metric A:

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

Let us also consider first order Taylor approximation of a function $f(x)$ near the point x_0:

$$
f(x_0 + \delta x) \approx f(x_0) + \nabla f(x_0)^\top \delta x \tag{1}
$$

Now we can explicitly pose a problem of finding s, as it was stated above.

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

Using equation (1 it can be written as:

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

Using Lagrange multipliers method, we can easily conclude, that the answer is:

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

The idea of adapive metrics

Given $f(x)$ and a point x_0. Define $B_\varepsilon(x_0) = \{x \in \mathbb{R}^n : d(x, x_0) = \varepsilon^2\}$ as the set of points with distance ε to x_0. Here we presume the existence of a distance function $d(x, x_0)$.

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

Then, we can define another steepest descent direction in terms of minimizer of function on a sphere:

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

Let us assume that the distance is defined locally by some metric A:

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

Let us also consider first order Taylor approximation of a function $f(x)$ near the point x_0:

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

Now we can explicitly pose a problem of finding s, as it was stated above.

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

Using equation (1 it can be written as:

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

Using Lagrange multipliers method, we can easily conclude, that the answer is:

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

Which means, that new direction of steepest descent is nothing else, but $A^{-1}\nabla f(x_0)$.

. . . Indeed, if the space is isotropic and $A = I$, we immediately have gradient descent formula, while Newton method uses local Hessian as a metric matrix.

Quasi-Newton methods intuition

For the classic task of unconditional optimization $f(x) \to \min\limits_{x \in \mathbb{R}^n}$ the general scheme of iteration method is written as:

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

Quasi-Newton methods intuition

For the classic task of unconditional optimization $f(x) \to \min\limits_{x \in \mathbb{R}^n}$ the general scheme of iteration method is written as:

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

In the Newton method, the d_k direction (Newton's direction) is set by the linear system solution at each step:

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

Quasi-Newton methods intuition

For the classic task of unconditional optimization $f(x) \to \min\limits_{x \in \mathbb{R}^n}$ the general scheme of iteration method is written as:

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

In the Newton method, the d_k direction (Newton's direction) is set by the linear system solution at each step:

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

i.e. at each iteration it is necessary to **compute** hessian and gradient and **solve** linear system.

Quasi-Newton methods intuition

For the classic task of unconditional optimization $f(x) \to \min\limits_{x \in \mathbb{R}^n}$ the general scheme of iteration method is written as:

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

In the Newton method, the d_k direction (Newton's direction) is set by the linear system solution at each step:

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

i.e. at each iteration it is necessary to **compute** hessian and gradient and **solve** linear system.

Note here that if we take a single matrix of $B_k = I_n$ as B_k at each step, we will exactly get the gradient descent method.

The general scheme of quasi-Newton methods is based on the selection of the B_k matrix so that it tends in some sense at $k \to \infty$ to the truth value of the Hessian $\nabla^2 f(x_k)$.

Quasi-Newton Method Template

Let $x_0 \in \mathbb{R}^n$, $B_0 \succ 0$. For $k = 1, 2, 3, \ldots$, repeat:

1. Solve $B_k d_k = -\nabla f(x_k)$

Quasi-Newton Method Template

Let $x_0 \in \mathbb{R}^n$, $B_0 \succ 0$. For $k = 1, 2, 3, ...$, repeat:

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

Quasi-Newton Method Template

Let $x_0 \in \mathbb{R}^n$, $B_0 \succ 0$. For $k = 1, 2, 3, \ldots$, repeat:

1. Solve $B_k d_k = -\nabla f(x_k)$
2. Update $x_{k+1} = x_k + \alpha_k d_k$
3. Compute B_{k+1} from B_k

Quasi-Newton Method Template

Let $x_0 \in \mathbb{R}^n$, $B_0 \succ 0$. For $k = 1, 2, 3, \ldots$, repeat:

1. Solve $B_k d_k = -\nabla f(x_k)$
2. Update $x_{k+1} = x_k + \alpha_k d_k$
3. Compute B_{k+1} from B_k

Quasi-Newton Method Template

Let $x_0 \in \mathbb{R}^n$, $B_0 \succ 0$. For $k = 1, 2, 3, \dots$, repeat:

1. Solve $B_k d_k = -\nabla f(x_k)$
2. Update $x_{k+1} = x_k + \alpha_k d_k$
3. Compute B_{k+1} from B_k

Different quasi-Newton methods implement Step 3 differently. As we will see, commonly we can compute $(B_{k+1})^{-1}$ from $(B_k)^{-1}$.

Quasi-Newton Method Template

Let $x_0 \in \mathbb{R}^n$, $B_0 \succ 0$. For $k = 1, 2, 3, \ldots$, repeat:

1. Solve $B_k d_k = -\nabla f(x_k)$
2. Update $x_{k+1} = x_k + \alpha_k d_k$
3. Compute B_{k+1} from B_k

Different quasi-Newton methods implement Step 3 differently. As we will see, commonly we can compute $(B_{k+1})^{-1}$ from $(B_k)^{-1}$.

Basic Idea: As B_k already contains information about the Hessian, use a suitable matrix update to form B_{k+1}.

Quasi-Newton Method Template

Let $x_0 \in \mathbb{R}^n$, $B_0 \succ 0$. For $k = 1, 2, 3, \ldots$, repeat:

1. Solve $B_k d_k = -\nabla f(x_k)$
2. Update $x_{k+1} = x_k + \alpha_k d_k$
3. Compute B_{k+1} from B_k

Different quasi-Newton methods implement Step 3 differently. As we will see, commonly we can compute $(B_{k+1})^{-1}$ from $(B_k)^{-1}$.

Basic Idea: As B_k already contains information about the Hessian, use a suitable matrix update to form B_{k+1}.

Reasonable Requirement for B_{k+1} (motivated by the secant method):

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

Quasi-Newton Method Template

Let $x_0 \in \mathbb{R}^n$, $B_0 \succ 0$. For $k = 1, 2, 3, \ldots$, repeat:

1. Solve $B_k d_k = -\nabla f(x_k)$
2. Update $x_{k+1} = x_k + \alpha_k d_k$
3. Compute B_{k+1} from B_k

Different quasi-Newton methods implement Step 3 differently. As we will see, commonly we can compute $(B_{k+1})^{-1}$ from $(B_k)^{-1}$.

Basic Idea: As B_k already contains information about the Hessian, use a suitable matrix update to form B_{k+1}.

Reasonable Requirement for B_{k+1} (motivated by the secant method):

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

In addition to the secant equation, we want:

- B_{k+1} to be symmetric

Quasi-Newton Method Template

Let $x_0 \in \mathbb{R}^n$, $B_0 \succ 0$. For $k = 1, 2, 3, \dots$, repeat:

1. Solve $B_k d_k = -\nabla f(x_k)$
2. Update $x_{k+1} = x_k + \alpha_k d_k$
3. Compute B_{k+1} from B_k

Different quasi-Newton methods implement Step 3 differently. As we will see, commonly we can compute $(B_{k+1})^{-1}$ from $(B_k)^{-1}$.

Basic Idea: As B_k already contains information about the Hessian, use a suitable matrix update to form B_{k+1}.

Reasonable Requirement for B_{k+1} (motivated by the secant method):

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

In addition to the secant equation, we want:

- B_{k+1} to be symmetric
- B_{k+1} to be "close" to B_k

Quasi-Newton Method Template

Let $x_0 \in \mathbb{R}^n$, $B_0 \succ 0$. For $k = 1, 2, 3, \ldots$, repeat:

1. Solve $B_k d_k = -\nabla f(x_k)$
2. Update $x_{k+1} = x_k + \alpha_k d_k$
3. Compute B_{k+1} from B_k

Different quasi-Newton methods implement Step 3 differently. As we will see, commonly we can compute $(B_{k+1})^{-1}$ from $(B_k)^{-1}$.

Basic Idea: As B_k already contains information about the Hessian, use a suitable matrix update to form B_{k+1}.

Reasonable Requirement for B_{k+1} (motivated by the secant method):

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

In addition to the secant equation, we want:

- B_{k+1} to be symmetric
- B_{k+1} to be "close" to B_k
- $B_k \succ 0 \Rightarrow B_{k+1} \succ 0$

Symmetric Rank-One Update

Let's try an update of the form:

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

Symmetric Rank-One Update

Let's try an update of the form:

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

The secant equation $B_{k+1}d_k = \Delta y_k$ yields:

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

Symmetric Rank-One Update

Let's try an update of the form:

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

The secant equation $B_{k+1}d_k = \Delta y_k$ yields:

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

This only holds if u is a multiple of $\Delta y_k - B_k d_k$. Putting $u = \Delta y_k - B_k d_k$, we solve the above,

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

Symmetric Rank-One Update

Let's try an update of the form:

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

The secant equation $B_{k+1} d_k = \Delta y_k$ yields:

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

This only holds if u is a multiple of $\Delta y_k - B_k d_k$. Putting $u = \Delta y_k - B_k d_k$, we solve the above,

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

which leads to

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

called the symmetric rank-one (SR1) update or Broyden method.

Symmetric Rank-One Update with inverse

How can we solve

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

in order to take the next step? In addition to propagating B_k to B_{k+1}, let's propagate inverses, i.e., $C_k = B_k^{-1}$ to $C_{k+1} = (B_{k+1})^{-1}$.

Sherman-Morrison Formula:

The Sherman-Morrison formula states:

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

Thus, for the SR1 update, the inverse is also easily updated:

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

In general, SR1 is simple and cheap, but it has a key shortcoming: it does not preserve positive definiteness.

Davidon-Fletcher-Powell Update

We could have pursued the same idea to update the inverse C:

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

Davidon-Fletcher-Powell Update

We could have pursued the same idea to update the inverse C:

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

Multiplying by Δy_k, using the secant equation $d_k = C_k \Delta y_k$, and solving for a, b, yields:

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

Woodbury Formula Application

Woodbury then shows:

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

This is the Davidon-Fletcher-Powell (DFP) update. Also cheap: $O(n^2)$, preserves positive definiteness. Not as popular as BFGS.

Broyden-Fletcher-Goldfarb-Shanno update

Let's now try a rank-two update:

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

Broyden-Fletcher-Goldfarb-Shanno update

Let's now try a rank-two update:

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

The secant equation $\Delta y_k = B_{k+1} d_k$ yields:

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

Broyden-Fletcher-Goldfarb-Shanno update

Let's now try a rank-two update:

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

The secant equation $\Delta y_k = B_{k+1} d_k$ yields:

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

Putting $u = \Delta y_k$, $v = B_k d_k$, and solving for a, b we get:

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

called the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update.

Broyden-Fletcher-Goldfarb-Shanno update with inverse

Woodbury Formula

The Woodbury formula, a generalization of the Sherman-Morrison formula, is given by:

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

Broyden-Fletcher-Goldfarb-Shanno update with inverse

Woodbury Formula

The Woodbury formula, a generalization of the Sherman-Morrison formula, is given by:

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

Applied to our case, we get a rank-two update on the inverse C:

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

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

This formulation ensures that the BFGS update, while comprehensive, remains computationally efficient, requiring $O(n^2)$ operations. Importantly, BFGS update preserves positive definiteness. Recall this means $B_k \succ 0 \Rightarrow B_{k+1} \succ 0.$ Equivalently, $C_k \succ 0 \Rightarrow C_{k+1} \succ 0$

Code

- Open In Colab

Code

- Open In Colab
- Comparison of quasi Newton methods

Code

- Open In Colab
- Comparison of quasi Newton methods
- Some practical notes about Newton method