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

Gradient Alg.	Newton's method
$x \leftarrow x - t \nabla f(x)^T$	$x \leftarrow x - t\, H(f(x))^{-1} \nabla f(x)^T$
# of iterations $\sim \log \frac{1}{\varepsilon}$	# of iterations: $\log\log \frac{1}{\varepsilon}$ if the initial value is close enough to a local solution.

- which algorithm is better?

many iterations for Gradient alg. $\longleftrightarrow$ modest number of iterations for Newton's

Example: 10000 Example: 20

cheap iterations $\longleftrightarrow$ expensive iterations.

- Example: Imagine that $x \in \mathbb{R}^{1000}$

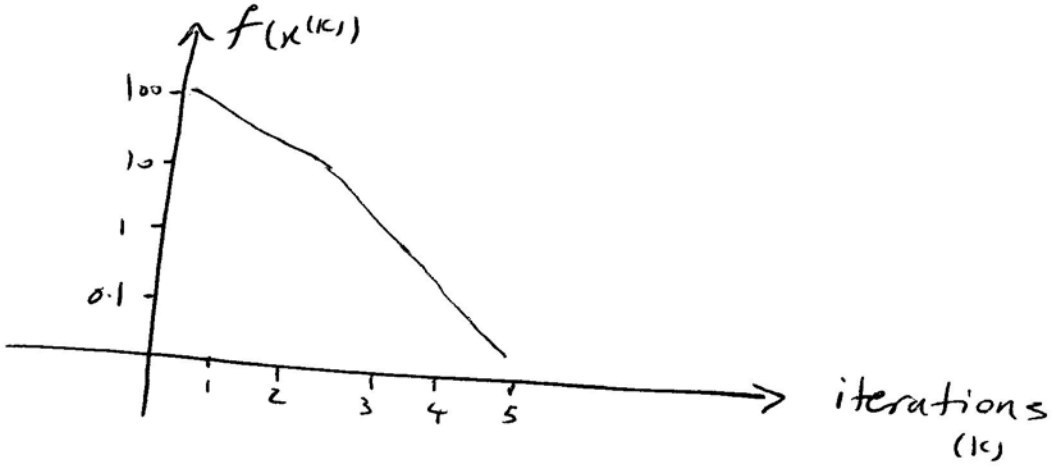
$\Rightarrow$ $H(f(x))$ has 10^6 entries and we need to take its inverse at each iteration.

- we can compute $H(f(x))^{-1} \nabla f(x)^T$ without explicitly computing $H(f(x))^{-1}$, but it's still expensive

Summary :

Gradient algorithm needs many cheap iterations to get a tolerance ε $\longleftrightarrow$ Newton's method needs a few expensive iterations to get to a tolerance ε.

- How to evalute the convergence ? draw a Linear-Log plot where the horizontal axis shows iterations and the vertical axis shows $f(x)$.



- Can we design an algorithm with a convergence rate better than Gradient and Newton's methods ?

- Recall that $\nabla f(x^{(k)}) \Delta x^{(k)}$ accounts for the reduction at every iteration and that's why we look for a descent direction: $\nabla f(x^{(k)}) \Delta x^{(k)} < 0$

- What if we optimize the direction Δx ?

$\min_{\Delta x^{(k)}} \nabla f(x^{(k)}) \Delta x^{(k)}$ $\longrightarrow$ Looks plausible but the minimum would be $-\infty$ corresponding to a $\Delta x^{(k)}$ with unbounded entries.

- Recall that $\Delta x^{(k)}$ accounts for direction and $t^{(k)}$ scales it down or up. So, why not normalize $\Delta x^{(k)}$.

- Consider an arbitrary norm $\|\cdot\|$.

- Example: $\|[1 \; -1 \; 3]\| =$ $\nearrow \sqrt{1^2+(-1)^2+3^2}$, $\rightarrow |1|+|-1|+|3|$, $\searrow \max(|1|, |-1|, |3|)$: different types of norms.

$\Rightarrow \min_{\Delta x^{(k)}} \nabla f(x^{(k)}) \Delta x^{(k)}$ s.t. $\|\Delta x^{(k)}\| = 1$

- This is called steepest descent algorithm.

- If $\|\cdot\|$ = length of vector $\Rightarrow \Delta x^{(k)} = \dfrac{-\nabla f(x^{(k)})}{\|\nabla f(x^{(k)})\|}$

↓ Normalized Gradient

↓ Gradient method.

- If $\|y\| = y^T M y$ for some fixed matrix M,

$\Rightarrow$ The optimal $\Delta x^{(k)}$ corresponds to Newton's method.

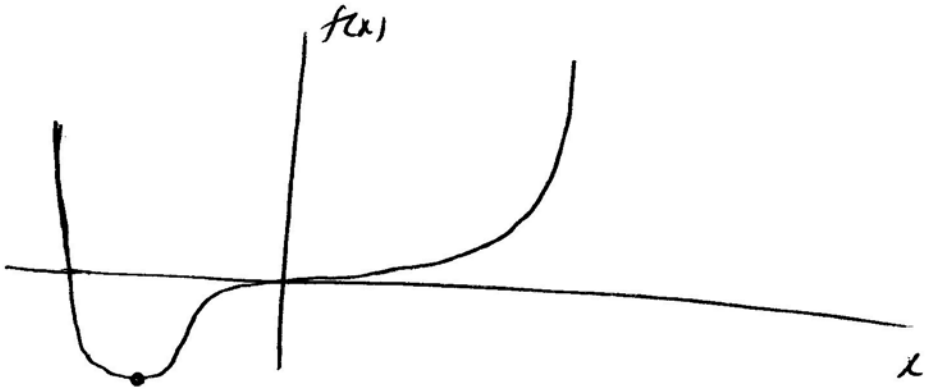
$\Rightarrow$ steepest descent includes Gradient and Newton's method.

- Assume we use one of the previous algorithms and generates a sequence:

$$x^{(0)} \longrightarrow x^{(1)} \longrightarrow x^{(2)} \cdots \longrightarrow x^{(k)} \longrightarrow \cdots$$

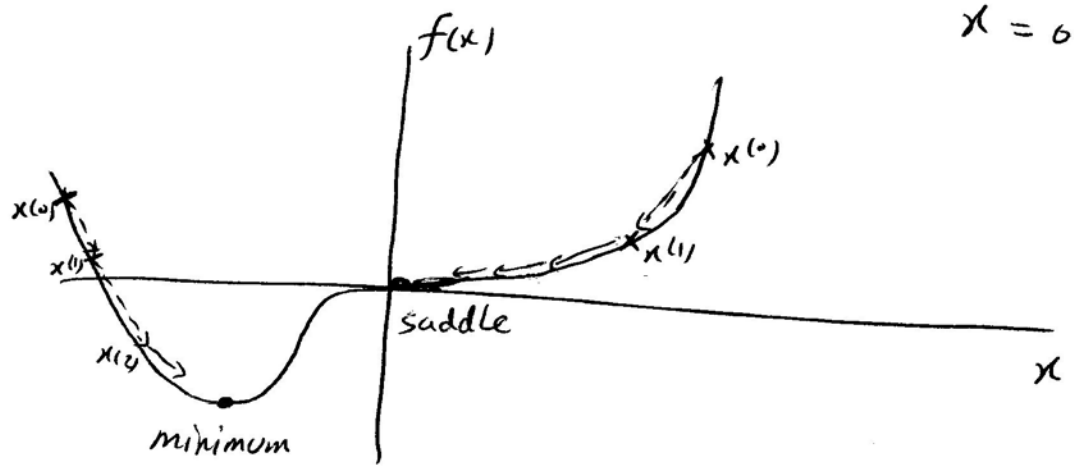
- Question: where does it converge to (if any)?
  Local min or saddle point.

- Example:



If $x^{(0)} < 0 \Rightarrow$ it converges to a minimum

If $x^{(0)} \geq 0 \Rightarrow$ it converges to the saddle point $x = 0$



Theorem: Consider a local minimum x^*.

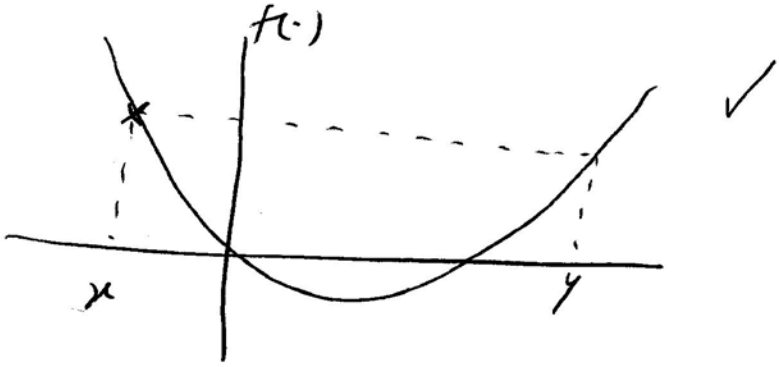
If $x^{(0)}$ is close enough to x^*, the sequence converges to x^*.

- I don't know the solution, how can I figure out my initial guess $x^{(0)}$ is good enough?

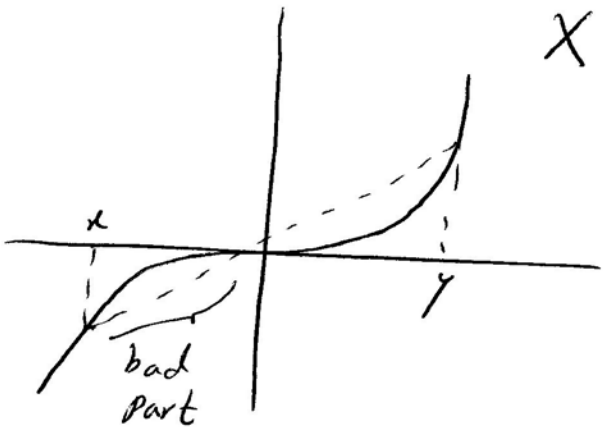
- Can we come up with a classes of functions for which $x^{(0)}$ doesn't matter?

- Yes, this is about Convex optimization.

- A function $f(x): \mathbb{R}^n \to \mathbb{R}$ is convex if for every two points x and y, the line connecting $(x, f(x))$ to $(y, f(y))$ is above the function. (segment)



✓



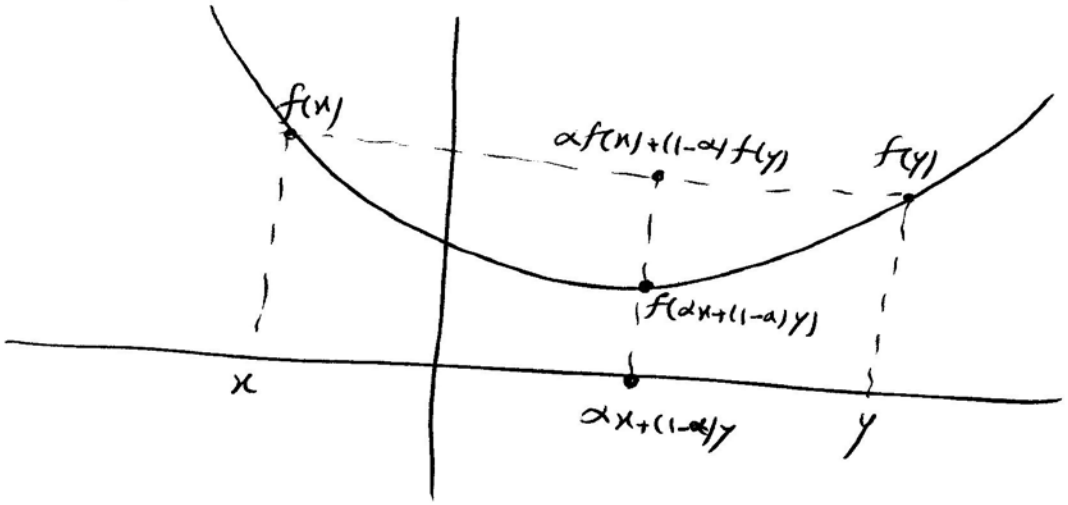
X

Mathematical definition:

$f(\cdot): \mathbb{R}^n \to \mathbb{R}$ is convex if for every $x, y \in \mathbb{R}^n$ and $\alpha \in [0,1]$, we have:

$$f(\alpha x + (1-\alpha) y) \leq \alpha f(x) + (1-\alpha) f(y)$$

- Illustration:



- Examples of convex functions in $\mathbb{R}$: